

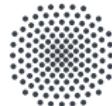
Numerische Grundlagen (HM4)

SoSe 2018

Prof. Dr. Dominik Göddeke

Anna Rörich Gabriele Santin

Anne Bernhardt Annegret Dieterich Julia Kühnert Laura Voggesberger



Universität Stuttgart



0. Einführung

Gliederung Kapitel 0

- Organisatorisches
- Einige Beispiele aus der Vorlesung
- Anwendungen jenseits der Vorlesung
- Bedeutung der numerischen Mathematik für Ihre Studiengänge



0. Einführung

Dozent:

Prof. Dr. Dominik Götdeke

Sprechstunde:

AMR5b, Raum E.040

Dienstag, 16:00-17:00 Uhr
und nach Vereinbarung

Assistenz:

Dr. Gabriele Santin

Sprechstunde:

NWZII (PWR 57), Raum 7.118

Montag, 9:00-10:00 Uhr

M.Sc. Anna Rörich

Sprechstunde:

AMR 5b, Raum 1.043

Mittwoch, 10:00-11:00 Uhr

Tutorinnen:

Anne Bernhardt, Annegret Dieterich

Julia Kühnert, Laura Voggesberger

ViPLab-Sprechstunden

Termine in ILIAS



0. Einführung

Zeiten und Räume:

Vorlesung: Di, wöchentlich, 4. Block, V 53.01

Vortragsübung: MI, 14-tägig, 1. Block, V 7.02 (ernen, fmt, mawi, bau)
(gedoppelt) FR, 14-tägig, 3. Block, V 38.01 (mach, tema, verf)

Tutorien: s. ILIAS, 0.711+0.712 NWZII

Zentrale Plattform: ILIAS-Kурсseite (Beitritt über C@MPUS)

- https://ilias3.uni-stuttgart.de/goto_Uni_Stuttgart_crs_1427875.html



0. Einführung

Konzept der Vorlesung:

- Kompletter Neuentwurf nach berechtigter Kritik aus MaBau letztes Semester
- Weit mehr als ein „Rundumschlag“ durch die numerische Mathematik
- Vielmehr: Anwendungsübergreifender Überblick über Methoden und Algorithmen zur numerischen Approximation von Lösungen für Probleme, die Modellcharakter für Ihre Studiengänge aufweisen
- Und: Konsolidierung der HM123 in Bezug auf praktische Umsetzung, Bereitstellung eines methodischen „Werkzeugkastens“

Ziel der Vorlesung

- Erarbeitung aller notwendiger Komponenten zur numerischen Lösung von Differentialgleichungen, wenigstens für zentrale Modellprobleme



0. Einführung

Anwendung und Erweiterung dieser Vorlesung in den Fachcurricula

- Technische Mechanik IV
- Computergestützte Materialwissenschaft
- Technische Thermodynamik, Technische Strömungslehre
- Regelungs- und Steuerungstechnik
- Und ca. 3458435345 Master-Vorlesungen mit Aspekten der Modellierung, Simulation und Optimierung



0. Einführung

Umsetzung des Konzepts:

- Entscheidungshilfen und ein erstes „Bauchgefühl“ zur Auswahl geeigneter Methoden für gegebene Aufgabenstellungen
- In jeder Vorlesung: Anwendungsbeispiele als „roter Faden“, kurze Rechenbeispiele, ausführliche Diskussion und Zusammenfassung
- Als Hausaufgabe: Umsetzung mit Papier & Bleistift und am Computer
 - ▶ Dazu: Erlernen / Auffrischen mathematischer Programmierung
 - ▶ Details gleich
- Viel stärkere Verzahnung von Vorlesung, Vortragsübung und Übung als früher



0. Einführung

Themen der Vorlesung: (1 Thema entspricht 1 Vorlesung)

Teil A (Basics): Konsolidierung HM123, Verfahren für lineare Probleme

- ① Einführung
- ② Direkte Lösungsverfahren für lineare Gleichungssysteme (auch heute)
- ② Störungstheorie, iterative Lösung linearer Gleichungssysteme
- ③ Iterationsverfahren für Eigenwertprobleme



0. Einführung

Themen der Vorlesung: (1 Thema entspricht 1 Vorlesung)

Teil B (Foundations): Werkzeugkasten zentraler fortgeschrittener Methoden

- ④ Kontinuierliche Optimierung
- ⑤ Iterative Verfahren für nichtlineare Gleichungssysteme
- ⑥ Interpolation mit Polynomen und Splines
- ⑦ Numerische Integration: Quadraturverfahren



0. Einführung

Themen der Vorlesung: (1 Thema entspricht 1 Vorlesung)

Teil C (Goal): Numerik für Differentialgleichungen

- ⑧ Gewöhnliche Differentialgleichungen
- ⑨ Partielle Differentialgleichungen: Finite Differenzen Methode im Ort
- ⑩ PDEs: Finite Differenzen in Ort und Zeit, nichtlinearer Fall
- ⑪ PDEs: schwache Lösungen und Finite Elemente Methode
- ⑫ PDEs: Finite Elemente in der Praxis



0. Einführung

Begleitung der Veranstaltung in ILIAS

- Vorlesungskalender, konkrete (!) Termine für Vortragsübungen und Tutorien
- Prüfungsmodalitäten, Modulzugehörigkeiten (unverbindlich!)
- Kein ausformuliertes Skript, sondern ausführliche Folien
- Folien jeweils **vor** der Vorlesung: „mitdenken statt mitschreiben“
- Videomitschnitte der Vorlesungen (Beamerbild+Audio) als besonderer Service zur Nachbereitung und Prüfungsvorbereitung
- Ergänzende Demos, Implementierungen, Visualisierungen, Bonusmaterial, . . .
- Materialien zu Vortragsübungen, zu ViPLab-Übungen, alte Klausuren, . . .



0. Einführung

Vortragsübungen

- Ausführlichere Rechenbeispiele als in der Vorlesung, Tipps und Tricks, deshalb hochgradig Klausur-relevant
- Erfahrung über die letzten Semester (Zitat Modulbefragung MaBau):

Die Vortragsübungen sind immens wichtig gewesen. Ich habe in keinem HM bisher Vortragsübung für die Prüfungsvorbereitung verwendet. Für Numerik jedoch ausschließlich mit den VÜ gelernt.

Achtung: gilt so nur für Bestehen, gute Note erfordert Vorlesung!

- Aufgabenblätter mit Rechenaufgaben (R-Blätter): Ausgabe 1.5 Wochen vor Besprechung passend zur ersten zugehörigen Vorlesung, Umfang jeweils ca. zwei Vorlesungen, Empfehlung: jede Woche $\frac{1}{2}$ R-Blatt bearbeiten
- Erstes Blatt nächste Woche, Besprechung FR 27.4. und MI 2.5.
- Keine Abgabe, keine Korrektur
- Keine Musterlösung, kein Videomitschnitt (aus didaktischem Prinzip)



0. Einführung

ViPLab-Übungen, Tutorien und (Programmier-) Sprechstunde

- Ein Übungsblatt (V-Blatt) pro Vorlesung, eng verzahnte Mischung aus Rechen- und Programmieraufgaben: praktische Anwendung der VL-Inhalte
- Bearbeitungszeit eine Woche: DI 16 Uhr bis DI 13 Uhr
- Automatische Korrektur, Einreichung via ViPLab in ILIAS
- Programmiersprache: Octave (Matlab)
 - ▶ <http://www.tik.uni-stuttgart.de/forschung/projekte/vip>
 - ▶ Viel Flexibilität: Bearbeitung auf eigenem Rechner oder im Browser
 - ▶ Einführung in den ersten Vortragsübungen diese Woche, Folien bereits online
 - ▶ V-Blatt 0 (bereits online): unverbindliches Herumspielen und Warmwerden
- Tutorien für individuelle Unterstützung
 - ▶ Diese Woche: MI 2.3.4., DO 2.3.4., FR 1.4., DI 2. Block, NWZII, 0.711+0.712
 - ▶ Achtung: first-come-first-serve, keine „Übungsgruppeneinteilung“!



0. Einführung

Prüfung und Modulabschluss

- Prüfungsstoff
 - ▶ Klausur-relevant: alle Vorlesungen und alle Vortragsübungen (außer VL 0)
 - ▶ Klausur-relevant: alle ViPLab-Übungen OHNE Programmieraspekte
 - ▶ Besonderer Service: Sammlung älterer Klausuraufgaben als Angebot für zusätzliches Lernmaterial, inkrementell für Teil A,B,C in ILIAS
 - ▶ Zwei Massensprechstunden: letzte Vorlesungswoche und in der Woche vor der Klausur in den Sommerferien
- Ewige Termine (unverbindlich): erste Septemberwoche und (neu!) Februar
 - ▶ Schriftliche Klausur, 90 Minuten
 - ▶ Anmeldung über C@MPUS wie üblich



0. Einführung

Modulabschluss „Numerische Grundlagen“

- 31740 Die benotete Studienleistung (BSL) setzt sich aus 10% Übungsleistung (ViPLab) und 90% Prüfungsleistung zusammen. Für die Prüfung gibt es keine Zulassungsbedingung (außer C@MPUS Anmeldung).
- 12180 Die unbenotete Studienleistung (USL) ist bestanden, wenn im Mittel aus 10% Übungsleistung (ViPLab) und 90% Prüfungsleistung eine 4.0 oder besser erreicht wurde. Für die Prüfung gibt es keine Zulassungsbedingung (außer C@MPUS Anmeldung).

Eine 1.0 ist ohne ViPLab-Punkte möglich. Erfahrungswert: Bonuspunkte helfen (teilweise extrem) für Bestehen: 12% (40%) Durchfallquote mit Bonuspunkten im SS (WS) 2017, 30% (60%) ohne.

Beispiel: 10/36 Klausurpunkte, nicht bestanden bei 1/3 Grenze. Bei 50/100 Übungspunkten (2/4 Bonuspunkten) Summe 12 Punkte, also bestanden



0. Einführung

Modulabschluss „Höhere Mathematik 4 (Numerik)“

11020 Die Prüfungsleistung (PL) besteht aus der Prüfung. Als Vorleistung (V) zur Prüfung müssen die ViPLab-Übungen erfolgreich absolviert werden (50% der Maximalzahl der ViPLab-Punkte).

Hinweis: PL und Bonusregelung sind leider rechtlich inkompatibel.

Teilnehmer vergangener Semester

Für Teilnehmer, die die Veranstaltung bereits gehört, aber das entsprechende Modul noch nicht bestanden haben, werden **keine** Leistungen aus den vergangenen Semestern anerkannt. Ausnahme: erteilte Zulassungen bei MaWi.

Grund: Kompletter inhaltlicher und organisatorischer Umbau



0. Einführung

Literaturempfehlungen

- M. Bollhöfer, V. Mehrmann: *Numerische Mathematik*, Vieweg (2004)
- W. Dahmen, A. Reusken: *Numerik für Ingenieure und Naturwissenschaftler*, Springer (2006)
- Peter Deuflhard und Andreas Hohmann, *Numerische Mathematik I: Eine algorithmisch orientierte Einführung*, de Gruyter (2008)
- R. Plato: *Numerische Mathematik kompakt*, Vieweg (2004)
- C.-D. Munz, T. Westermann: *Numerische Behandlung gewöhnlicher und partieller Differentialgleichungen - Ein interaktives Lehrbuch für Ingenieure*, Springer
- Mathematik-Online: www.mathematik-online.org (Skripte, Übungsaufgaben, Tests)
- Matlab/Octave/ViP Lab: Hinweise im Crashkurs



Einige Beispiele aus dieser Vorlesung

„Es macht wenig Sinn [...] Numerische Mathematik als Selbstzweck zu präsentieren. Wo ist der Sinn von Interpolation, Approximation und der Lösung linearer Systeme, wenn man nicht weiß, in welch vielfältigen Problemen diese Techniken anwendbar sind?“

Thomas Sonar (Mathematikdidaktiker, Braunschweig)

0. Einführung

Der Stuttgarter Fernsehturm in ViPLab

Die Schwingung des Stuttgarter Fernsehturms kann durch folgende Differentialgleichung beschrieben werden:

$$mx''(t) + cx(t) = F(t)$$
$$x(0) = 0 \quad x'(0) = 0$$

$x(t)$: Auslenkung in die Richtung, in der die (Wind-) Kraft F wirkt.

m, c : Masse und Biegesteifigkeit.



Fragestellung

Wie stark schwankt die Besucherplattform abhängig vom Wind?

Erste Antwort: ViPLab-Demo:

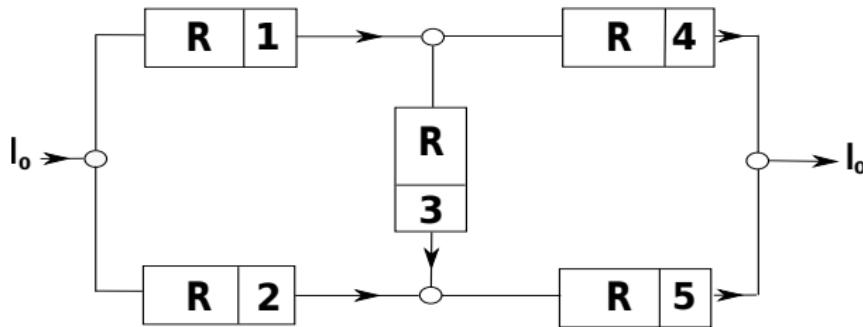
<http://www.tik.uni-stuttgart.de/forschung/projekte/vip>

Details & Relevanz: ViPLab ab jetzt; PDEs ab VL 8



0. Einführung

Elektrische Netzwerke



Knoten

R_i Widerstand i der (festen) Stärke $R > 0$

I_o Gegebene Ein-/ Ausgangsstromstärke

> Orientierung (beliebig, aber fest)

Fragestellung

Wie sehen die Stromstärken I_i in den Einzelleitungen zwischen je zwei Knoten aus?



0. Einführung

Die gesuchten Stromstärken genügen einem linearen Gleichungssystem:

$$\underbrace{\begin{pmatrix} -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ R & -R & R & 0 & 0 \\ 0 & 0 & R & -R & R \end{pmatrix}}_{=:A} \underbrace{\begin{pmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{pmatrix}}_{=:x} = \underbrace{\begin{pmatrix} -I_0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{=:b}$$

Es gilt $\det(A) = 8R^2 \neq 0$, das LGS ist eindeutig lösbar gemäß HM123, das Modell ist also **wohlgestellt**. Für große Netzwerke ist das LGS **dünn besetzt**.

Fragestellung

Wie können wir LGS effizienter als mit der Gauß-Elimination aus der Schule lösen?

Effiziente Lösung: VL 1+2



0. Einführung

Bewegungsgleichungen und Partikelsysteme



<https://www.youtube.com/watch?v=5p60AEVKw-0>

Die Evolution der Partikel genügt einem Anfangswertproblem gewöhnlicher Differentialgleichungen (den sog. Newton'schen Bewegungsgleichungen):

$$\begin{aligned} m_n \frac{\partial^2}{\partial t^2} \mathbf{x}_n(t) &= -\frac{\partial}{\partial \mathbf{x}_n} P(\mathbf{x}_1(t), \dots, \mathbf{x}_N(t)) && \text{für } n = 1, \dots, N \\ \mathbf{x}_n(0) = \mathbf{x}_n^0 & & \frac{\partial}{\partial t} \mathbf{x}_n(0) = \mathbf{v}_n^0 & \end{aligned}$$

Dabei: Potentialfeld P , N Partikel mit Massen $m_n > 0$, zum Zeitpunkt $t = 0$ Startpositionen $\mathbf{x}_n^0 \in \mathbb{R}^d$ und Startgeschwindigkeiten $\mathbf{v}_n^0 \in \mathbb{R}^d$.

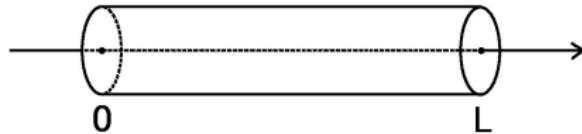
Fragestellung

Wo befinden sich die Partikel zu Zeitpunkten $t > 0$?



0. Einführung

Raclette in 1D



Fragestellung

Wie sieht die Temperaturverteilung $U = U(x, t)$ in einem Stab der Länge $L > 0$ aus, der an beiden Enden auf die Temperatur $U = 1$ geheizt ist?

Das mathematische Modell ist die sog. Wärmeleitungsgleichung, ein Anfangs-Randwertproblem in Ort und Zeit:

$$\begin{aligned}(\dot{U} := \frac{\partial}{\partial t} U) \quad \dot{U}(x, t) &= \alpha \frac{\partial^2}{\partial x^2} U(x, t) \quad \text{in } (0, L) \times (0, \infty) \\ U(x, 0) &= U_0(x) \quad \text{für } x \in (0, L) \\ U(0, t) &= U(L, t) = 1 \quad \text{für } t \in (0, \infty)\end{aligned}$$

Dabei: \dot{U} zeitliche Änderung der Temperatur, $\frac{\partial^2}{\partial x^2} U$ Wärmediffusion im Ort, $\alpha > 0$ Wärmeleitfähigkeit, U_0 Anfangswerte im Ort zum Zeitpunkt $t = 0$.



0. Einführung

Wir werden sehen:

Die numerische Approximation der Lösung ergibt sich als Lösung eines hochdimensionalen schwach gekoppelten (dünn besetzten) gewöhnlichen Anfangswertproblems:

$$\begin{pmatrix} \dot{U}_1(t) \\ \dot{U}_2(t) \\ \vdots \\ \dot{U}_{N-1}(t) \\ \dot{U}_N(t) \end{pmatrix} = \frac{\alpha}{h_x^2} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \begin{pmatrix} U_1(t) \\ U_2(t) \\ \vdots \\ U_{N-1}(t) \\ U_N(t) \end{pmatrix} + \frac{\alpha}{h_x^2} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

Die unbekannten Funktionen \dot{U}_i hängen nicht mehr vom Ort ab. Hinzu kommen die Anfangswerte.

Details: ab VL 8



0. Einführung

Realistischeres Raclette

Fragestellung

Existiert eine Gleichgewichtslösung, d. h. ein stationärer Zustand?

Konkret: Schmilzt der Käse immer gleich schnell, wenn einmal alles aufgeheizt ist?

Die numerische Lösung ergibt sich als Lösung eines dünnbesetzten LGS:

$$\underbrace{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}}_{=: \mathbf{A}} = \frac{\alpha}{h_x^2} \underbrace{\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}}_{=: \mathbf{x}} \underbrace{\begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_{N-1} \\ U_N \end{pmatrix}}_{=: \mathbf{b}} + \underbrace{\frac{\alpha}{h_x^2} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}}_{=: -\mathbf{b}}$$

Details: ab VL 8

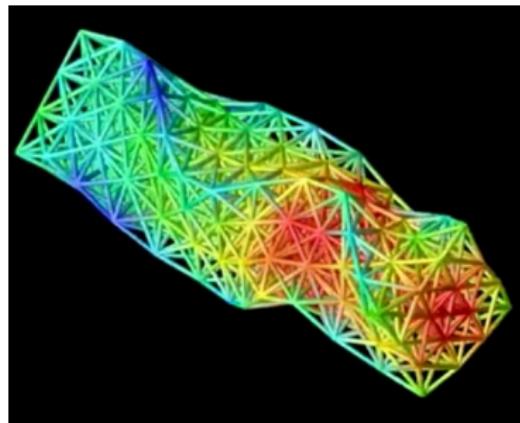


Weiterführende Beispiele jenseits der Vorlesung

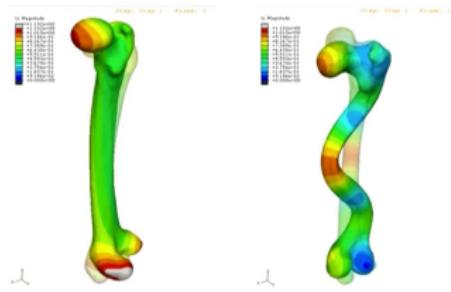


0. Einführung

Schwingung eines Tragwerks und Eigenmoden eines Knochens



www.youtube.com/watch?v=R9EWUI1IMFw

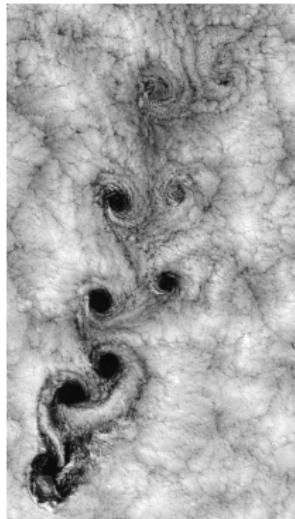


www.youtube.com/watch?v=gZAutaeAf7s



0. Einführung

Strömungssimulationen



Umströmung einer (hohen) Insel, Wirbelschleppen hinter Flugzeugen

NASA public domain



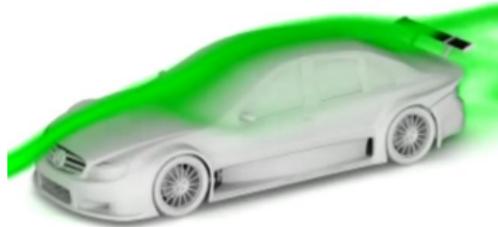
0. Einführung

Virtueller Windkanal für Fahrzeugumströmungen



https://www.youtube.com/watch?v=sV_6E1Lh7yo

<https://www.youtube.com/watch?v=LfJ0nQnu3yI>

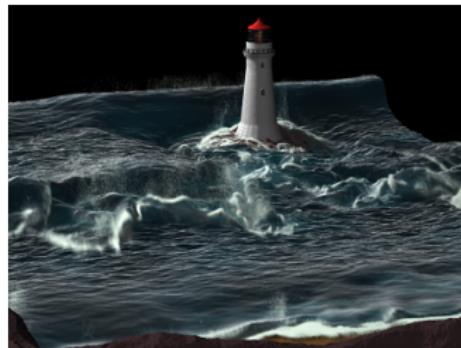


<https://www.youtube.com/watch?v=THx5LUFyP-8>

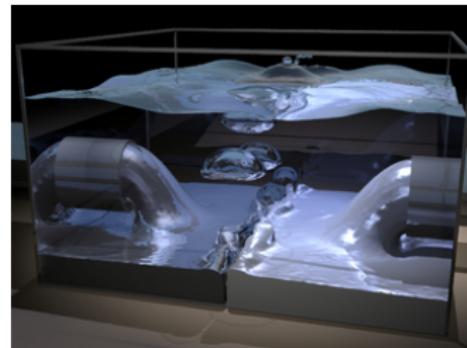


0. Einführung

Partikelsysteme



<http://physbam.stanford.edu/~fedkiw/animations/lighthouse.avi>



[.../chemical_reaction.avi](#)



Numerische Mathematik



0. Einführung

Schwerpunkte der Numerischen Mathematik

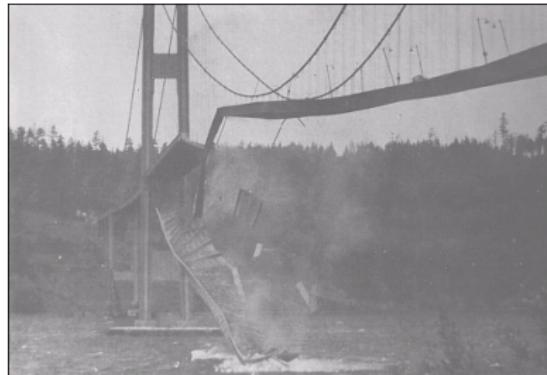
- Herleitung von Methoden zur Approximation von Lösungen von mathematischen Modellen wie (nicht-)lineare Gleichungssysteme, Eigenwert-Probleme, gewöhnliche und partielle Differentialgleichungen, ...
- Analytische Untersuchungen von Eigenschaften: Approximationsgüte, Fehlerschranken, Konvergenz, Konvergenzordnung, ...
- Entwicklung von Kriterien zur (Nicht-) Anwendbarkeit der Methoden
- Implementierung, rechentechnischer Nachweis der Eigenschaften
- Starke Bezüge zu reiner Mathematik, Informatik und Anwendungsgebieten, deshalb „HM4“



0. Einführung

Folgen von Unverständnis oder ungültiger Anwendung

- 1996: Ariane 5 Rakete explodiert (Typkonversion)
- 1991: Patriot Rakete versagt (Fehlerfortpflanzung)
- 1991: Sleipner Bohr-Insel-Basis gesunken (Fehler in Finite Elemente Analysis)
- 1940: Tacoma Brücke kollabiert (Wind-induzierte Vibrationen)



<http://ta.twi.tudelft.nl/users/vuik/wi211/disasters.html>



0. Einführung

Zusammenfassung

- Erst das Zusammenspiel von Theorie, Experiment und Simulation bringt in vielen Bereichen neue Erkenntnisse.
- Die numerische Mathematik ist die Grundlage der Simulationstechnik, und damit vieler Teilgebiete Ihrer Studiengänge.
- In diesem Modul: erster (intensiver) Kontakt, Fokus gleichermaßen auf Grundlagen / Basismethoden und Anwendung / Anwendbarkeit
- Realisiert durch sehr enge Verzahnung von VL, VÜ und Hausaufgaben



1. Lineare Gleichungssysteme I



Ich habe das Problem der Elimination in einer Weise
gelöst, die nichts zu wünschen übrig lässt.

C.F. Gauß, 1798



Gliederung Kapitel 1

- LR-Zerlegung als strukturierte Form der Gauß-Elimination zur Lösung linearer Gleichungssysteme
- Anwendungen der LR-Zerlegung
- Erste algorithmische Aspekte der Numerik



Die LR-Zerlegung



1. Lineare Gleichungssysteme I

Wir betrachten eine quadratische Koeffizientenmatrix

$$\mathbf{A} = (a_{mn})_{m,n=1}^N = \begin{pmatrix} a_{11} & \dots & a_{1N} \\ \vdots & & \vdots \\ a_{N1} & \dots & a_{NN} \end{pmatrix} \in \mathbb{R}^{N \times N},$$

und eine rechte Seite

$$\mathbf{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_N \end{pmatrix} \in \mathbb{R}^N,$$

und wollen das lineare Gleichungssystem (LGS) $\mathbf{Ax} = \mathbf{b}$ lösen, d. h. wir suchen $\mathbf{x} \in \mathbb{R}^N$, so dass

$$\mathbf{Ax} = \mathbf{b} \tag{1.1}$$

erfüllt ist.



1. Lineare Gleichungssysteme I

Wir erinnern uns zuerst an ein zentrales Resultat aus HM12:

Satz 1.1 (Lösbarkeit von LGS)

Ein lineares Gleichungssystem gemäß (1.1) hat genau dann eine eindeutige Lösung x , wenn $\det(\mathbf{A}) \neq 0$, d. h. genau dann, wenn \mathbf{A} invertierbar (regulär) ist.

Beispiel: elektrisches Netzwerk aus der Einleitung

Im Verlauf dieser Vorlesungseinheit beantworten wir folgende Fragen:

- Für welche \mathbf{A} ist das LGS „einfach“ lösbar?
- Wie können wir das mit der Gauß-Elimination aus der Schule und der HM123 verbinden?
- Welchen Mehrwert haben wir durch die damit verbundene Vereinheitlichung?



1. Lineare Gleichungssysteme I

Wir betrachten zunächst eine spezielle Klasse besonders einfach zu lösender LGS:

Definition 1.2 (Dreiecksmatrix)

$R \in \mathbb{R}^{N \times N}$ heißt **obere Dreiecksmatrix**, falls $r_{mn} = 0$ für $m > n$.

$L \in \mathbb{R}^{N \times N}$ heißt **untere Dreiecksmatrix**, falls $\ell_{mn} = 0$ für $m < n$.

$$R = \begin{pmatrix} r_{11} & \dots & r_{1N} \\ & \ddots & \vdots \\ \mathbf{0} & & r_{NN} \end{pmatrix} \quad L = \begin{pmatrix} \ell_{11} & & \mathbf{0} \\ \vdots & \ddots & \\ \ell_{N1} & \dots & \ell_{NN} \end{pmatrix}$$

Verständnisübung: Indices in der Definition genau überprüfen.



1. Lineare Gleichungssysteme I

Für obere Dreiecksmatrizen ist die Lösung von (LGS) besonders einfach. Die Gauß-Elimination (HM12) reduziert sich auf reines Rückwärtseinsetzen:

Algorithmus 1.3 : Rückwärtseinsetzen

input : $R \in \mathbb{R}^{N \times N}$ obere Dreiecksmatrix mit $\det(R) \neq 0$ und $b \in \mathbb{R}^N$

output : $x \in \mathbb{R}^N$ Lösung von $Rx = b$

1 $x_N = \frac{b_N}{r_{NN}}$;

2 **for** $m = N - 1$ **to** 1 **do**

3 $x_m =$

$$\frac{1}{r_{mm}} \left(b_m - \sum_{n=m+1}^N r_{mn} x_n \right);$$

$$\begin{pmatrix} r_{11} & \cdots & r_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & & r_{NN} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_N \end{pmatrix}$$

4 **end**

Das ist analog auch für $Lx = b$ mit einer unteren Dreiecksmatrix L durchführbar (Vorwärtseinsetzen). Für ein Beispiel gedulden wir uns noch einen Moment.



1. Lineare Gleichungssysteme I

Rechenaufwand:

$$x_N = \frac{b_N}{r_{NN}} \quad \text{und} \quad x_m = \frac{1}{r_{mm}} \left(b_m - \sum_{n=m+1}^N r_{mn}x_n \right) \quad \text{für } m = N-1, \dots, 1$$

- Aufwand in der m -ten Matrixzeile: $N - m$ Additionen und Multiplikationen, also insgesamt arithmetische Summe über alle Zeilen:

$$\sum_{m=1}^{N-1} (N - m) = \frac{N(N - 1)}{2} = \mathcal{O}(N^2)$$

- Zusätzlich: N Divisionen
- Gesamtaufwand: $\mathcal{O}(N^2)$ arithmetische Operationen
- \mathcal{O} -Notation: $\mathcal{O}(N^2) := C N^2$ mit $C \neq C(N)$ und $N \rightarrow \infty$



1. Lineare Gleichungssysteme I

Wie kann man im allgemeinen Fall $\mathbf{A}\mathbf{x} = \mathbf{b}$ vorgehen?

Beobachtung

Ist $\mathbf{B} \in \mathbb{R}^{N \times N}$ regulär, so gilt:

$$\mathbf{x} \text{ ist Lösung von } \mathbf{A}\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad \mathbf{x} \text{ ist Lösung von } (\mathbf{B}\mathbf{A})\mathbf{x} = \mathbf{B}\mathbf{b}$$

Beweis: elementare Eigenschaften linearer bijektiver Abbildungen, s. HM123.

Idee

Finde $\mathbf{L} \in \mathbb{R}^{N \times N}$ mit $\det(\mathbf{L}) \neq 0$, so dass

$$\mathbf{R} = \mathbf{L}^{-1}\mathbf{A}$$

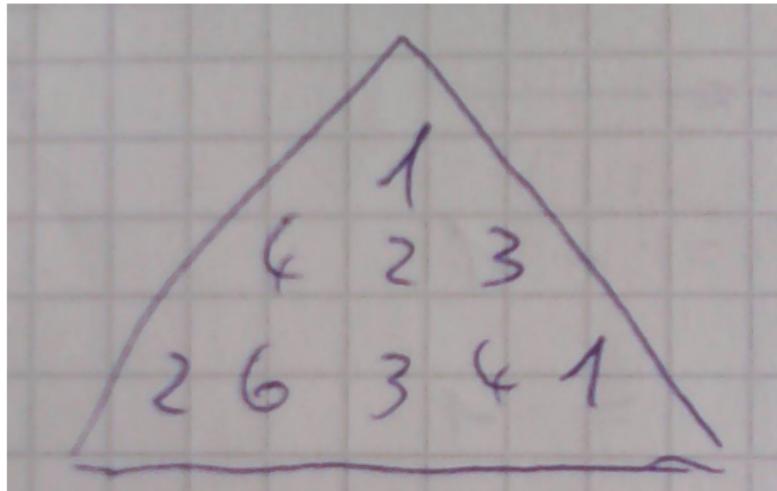
eine obere Dreiecksmatrix ist. Dann ist das System einfach per Rückwärtseinsetzen lösbar. Man spricht von einer **LR-Zerlegung** wegen

$$\mathbf{A} = \mathbf{L}\mathbf{R} \quad \Leftrightarrow \quad \mathbf{L}^{-1}\mathbf{A} = \mathbf{R}.$$

1. Lineare Gleichungssysteme I



Beispiel aus einer früheren Klausur: Überführen Sie die gegebene Matrix in untere Dreiecksgestalt.



Das können wir besser!



1. Lineare Gleichungssysteme I

Beispiel 1.4 (Gauß-Elimination)

Wir betrachten das LGS $\mathbf{A}\mathbf{x} = \mathbf{b}$ mit

$$\mathbf{A} = \begin{pmatrix} 2 & 4 & 6 \\ 1 & 3 & 5 \\ 1 & 4 & 9 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Gauß-Elimination aus HM1/Schule liefert im ersten Schritt:

$$\left| \begin{array}{ccc|cc} 2 & 4 & 6 & 1 & (\frac{1}{2}) \\ 1 & 3 & 5 & 0 & \\ 1 & 4 & 9 & 0 & \end{array} \right| \xrightarrow{\text{Subtraktions-Schritte}} \left| \begin{array}{ccc|cc} 2 & 4 & 6 & 1 & \\ 0 & 1 & 2 & -\frac{1}{2} & \\ 0 & 2 & 6 & -\frac{1}{2} & \end{array} \right|$$

Im zweiten Schritt erhalten wir:

$$\left| \begin{array}{ccc|cc} 2 & 4 & 6 & 1 & \\ 0 & 1 & 2 & -\frac{1}{2} & | (2) \\ 0 & 2 & 6 & -\frac{1}{2} & \end{array} \right| \xrightarrow{\text{Subtraktions-Schritte}} \left| \begin{array}{ccc|cc} 2 & 4 & 6 & 1 & \\ 0 & 1 & 2 & -\frac{1}{2} & \\ 0 & 0 & 2 & \frac{1}{2} & \end{array} \right|$$



1. Lineare Gleichungssysteme I

Die LR-Zerlegung ist nun eine „strukturierte“ Form der Gauß-Elimination:

Algorithmus 1.5 : LR-Zerlegung

input : $A \in \mathbb{R}^{N \times N}$

output : LR-Zerlegung $LR = A$ mit L untere Dreiecksmatrix mit 1-Diagonale und R obere Dreiecksmatrix, bzw. Abbruch, falls nicht durchführbar

```
1  $R = A, L = I;$ 
2 for  $k = 1$  to  $N - 1$  do
3   if  $r_{kk} = 0$  then
4     | % Fehlermeldung, LR-Zerlegung nicht durchführbar;
5   else
6     |  $\ell_{mk} = r_{mk} / r_{kk}$            Schleife  $m = k + 1, \dots, N;$ 
7     |  $r_{mk} = 0$                    Schleife  $m = k + 1, \dots, N;$ 
8     |  $r_{mn} = r_{mn} - \ell_{mk} r_{kn}$  Schleife  $m, n = k + 1, \dots, N;$ 
9   end
10 end
```



1. Lineare Gleichungssysteme I

Bemerkungen:

- Auf Englisch: LU-Zerlegung (für lower-upper statt links-rechts)
- In \mathbf{L} werden die Eliminationsfaktoren gespeichert.
- Gilt $\det(\mathbf{A}) = 0$, so bricht der Algorithmus für ein $k < N$ ab.
- Existiert eine Zerlegung von \mathbf{A} in ein Produkt $\mathbf{A} = \mathbf{L}\mathbf{R}$ aus unterer und oberer Dreiecksmatrix, so ist diese i. A. nicht eindeutig. Die **Normierung** $\ell_{kk} = 1$ macht die Zerlegung eindeutig, das kann man beweisen.
- Man kann die LR-Zerlegung ohne zusätzlichen Speicheraufwand in \mathbf{A} speichern, sofern die Eins-Diagonale von \mathbf{L} nicht gespeichert wird.
- Rechenaufwand: $\approx \sum_{k=1}^{N-1} (N - k)^2 = \mathcal{O}(N^3)$



1. Lineare Gleichungssysteme I

Sei \mathbf{L} und \mathbf{R} die Ausgabe von Algorithmus 1.5 mit Eingabe \mathbf{A} . Damit können wir die Lösung von $\mathbf{Ax} = \mathbf{b}$ wie folgt berechnen:

$$(\mathbf{LR})\mathbf{x} = \mathbf{Ax} = \mathbf{b} \quad \Leftrightarrow \quad \mathbf{Rx} = \mathbf{L}^{-1}\mathbf{b} \quad \Leftrightarrow \quad \mathbf{Ly} = \mathbf{b} \text{ und } \mathbf{Rx} = \mathbf{y}$$

Algorithmus 1.6 : Gauß-LR

input : $\mathbf{A} \in \mathbb{R}^{N \times N}$, die eine LR-Zerlegung besitzt, und $\mathbf{b} \in \mathbb{R}^N$

output : Lösung $\mathbf{x} \in \mathbb{R}^N$ von $\mathbf{Ax} = \mathbf{b}$

- 1 Berechne die LR-Zerlegung $\mathbf{LR} = \mathbf{A}$ mit Algorithmus 1.5;
 - 2 Berechne die Lösung $\mathbf{y} \in \mathbb{R}^N$ von $\mathbf{Ly} = \mathbf{b}$ durch Vorwärtseinsetzen;
 - 3 Berechne die Lösung \mathbf{x} von $\mathbf{Rx} = \mathbf{y}$ durch Rückwärtseinsetzen;
-

Gesamtaufwand: $\mathcal{O}(N^3) + \mathcal{O}(N^2) + \mathcal{O}(N^2) = \mathcal{O}(N^3)$



1. Lineare Gleichungssysteme I

Beispiel 1.4 cont.

$$\begin{pmatrix} 2 & 4 & 6 \\ 1 & 3 & 5 \\ 1 & 4 & 9 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Gauß-Elimination aus HM1 liefert im ersten Schritt:

$$\left| \begin{array}{ccc|cc} 2 & 4 & 6 & 1 & (\frac{1}{2}) \\ 1 & 3 & 5 & 0 & \\ 1 & 4 & 9 & 0 & \end{array} \right| \xrightarrow{\begin{array}{l} -1 \cdot \text{Zeile 1} + \text{Zeile 2} \\ -1 \cdot \text{Zeile 1} + \text{Zeile 3} \end{array}} \left| \begin{array}{ccc|cc} 2 & 4 & 6 & 1 & \\ 0 & 1 & 2 & -\frac{1}{2} & \\ 0 & 2 & 6 & -\frac{1}{2} & \end{array} \right|$$

Das ist genau der erste Schritt der LR-Zerlegung und wir speichern:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{pmatrix} \quad \text{und} \quad R = \begin{pmatrix} 2 & 4 & 6 \\ 0 & 1 & 2 \\ 0 & 2 & 6 \end{pmatrix}$$



1. Lineare Gleichungssysteme I

Gauß-Elimination aus HM1 liefert im zweiten Schritt:

$$\begin{array}{ccc|c} 2 & 4 & 6 & 1 \\ 0 & 1 & 2 & -\frac{1}{2} \\ 0 & 2 & 6 & -\frac{1}{2} \end{array} \xrightarrow{(2)} \begin{array}{ccc|c} & & & 1 \\ 0 & 1 & 2 & -\frac{1}{2} \\ 0 & 0 & 2 & \frac{1}{2} \end{array}$$

Dies ist der zweite Schritt der LR-Zerlegung und wir speichern:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & 2 & 1 \end{pmatrix} \quad \text{und} \quad R = \begin{pmatrix} 2 & 4 & 6 \\ 0 & 1 & 2 \\ 0 & 0 & 2 \end{pmatrix}$$

Das ist bereits die LR-Zerlegung von \mathbf{A} .



1. Lineare Gleichungssysteme I

Vorwärtseinsetzen liefert den Hilfsvektor \mathbf{y} als Lösung von $\mathbf{Ly} = \mathbf{b}$:

$$\begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & 2 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \Leftrightarrow \mathbf{y} = \begin{pmatrix} 1 \\ -\frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$$

Rückwärtseinsetzen in $\mathbf{Rx} = \mathbf{y}$ liefert schließlich die gesuchte Lösung:

$$\mathbf{Rx} = \mathbf{L}^{-1}\mathbf{b} = \mathbf{y} \Leftrightarrow \begin{pmatrix} 2 & 4 & 6 \\ 0 & 1 & 2 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ -\frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \Leftrightarrow \mathbf{x} = \begin{pmatrix} \frac{7}{4} \\ -1 \\ \frac{1}{4} \end{pmatrix}$$

Dies ist die eindeutige Lösung von $\mathbf{Ax} = \mathbf{b}$.

Weitere Beispiele: Vortragsübung und Programmierblatt 1, Ende der VL



1. Lineare Gleichungssysteme I

Beispiel 1.7 (Motivation der Pivotisierung)

Wir betrachten das LGS $\mathbf{A}\mathbf{x} = \mathbf{b}$ mit

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 2 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \text{und} \quad \mathbf{b} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Es gilt $\det(\mathbf{A}) = -1 \neq 0$.

Wir versuchen, die LR-Zerlegung zu bestimmen, und erhalten im ersten Eliminationsschritt:

$$\left| \begin{array}{ccc|cc} 1 & 0 & 2 & 1 & (1) \\ 1 & 0 & 1 & 0 & \leftarrow \\ 0 & 1 & 0 & 0 & \end{array} \right|^- \quad \left| \begin{array}{ccc|c} 1 & 0 & 2 & 1 \\ 0 & \color{red}{0} & -1 & -1 \\ 0 & 1 & 0 & 0 \end{array} \right|$$

Danach bricht die LR-Zerlegung ab, da das **Pivot-Element** r_{22} im nächsten Schritt verschwindet.



1. Lineare Gleichungssysteme I

Wir erinnern uns an die HM123 und Folie 43: Äquivalenzumformungen ändern die Lösungsmenge (bei uns: die Lösung) eines LGS nicht.

Die Idee der **Pivotisierung** ist nun, durch Äquivalenzumformungen sicherzustellen, dass der Algorithmus zur LR-Zerlegung fortgesetzt werden kann. Eine besonders einfache Äquivalenzumformung ist der Zeilentausch.

Im Beispiel liefert die Vertauschung der zweiten und dritten Zeile

$$\begin{array}{ccc|c} 1 & 0 & 2 & 1 \\ 0 & \textcolor{red}{0} & -1 & -1 \\ 0 & 1 & 0 & 0 \end{array} \quad \begin{array}{l} \text{Vertauschung} \\ \longrightarrow \\ \text{2-te/3-te Zeile} \end{array} \quad \begin{array}{ccc|c} 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \end{array}$$

sofort eine obere Dreiecksmatrix R .

Wichtig: Ein Zeilentausch mit einer bereits behandelten Zeile ist verboten!



1. Lineare Gleichungssysteme I

Algorithmus 1.8 : LR-Zerlegung mit Zeilenpivotisierung

input : $A \in \mathbb{R}^{N \times N}$ mit $\det(A) \neq 0$

output : LR-Zerlegung $LR = PA$ mit L untere Dreiecksmatrix mit 1-Diagonale, R obere Dreiecksmatrix und p Vektor zur Protokollierung der Zeilenvertauschungen

```
1  $R = A$ ,  $L = I$ ,  $p = [1, \dots, N]^T$ ;  
2 for  $k = 1$  to  $N - 1$  do  
3   if  $r_{kk} = 0$  then  
4     wähle  $m_* \geq k$  so, dass  $|r_{m*,k}| = \max_{m \geq k} |r_{mk}|$ ;  
5      $r_{kn} \leftrightarrow r_{m*,n}$  für  $n = k, \dots, N$ ;          % Zeilenvertauschung in R  
6      $\ell_{kn} \leftrightarrow \ell_{m*,n}$  für  $n = 1, \dots, k - 1$ ;    % Zeilenvertauschung in L  
7     vertausche  $p_k$  und  $p_{m_*}$ ;                  % Protokoll der Vertauschung  
8   end  
9    $\ell_{mk} = r_{mk}/r_{kk}$                           $m = k + 1, \dots, N$ ;  
10   $r_{mk} = 0$                                      $m = k + 1, \dots, N$ ;  
11   $r_{mn} = r_{mn} - \ell_{mk} r_{kn}$                  $m, n = k + 1, \dots, N$ ;  
12 end
```



1. Lineare Gleichungssysteme I

Bemerkungen:

- Prinzipiell kann als **Pivotelement** ein beliebiger Nichtnull-Eintrag der aktuellen Spalte unterhalb der Diagonale gewählt werden. Die Wahl des Betrags-Maximums minimiert zusätzliche Gleitkommafehler → VL2.
- Algorithmus 1.8 liefert nicht die LR-Zerlegung von \mathbf{A} , sondern die LR-Zerlegung von \mathbf{PA} , dabei ist \mathbf{P} die Permutationsmatrix, welche die Zeilenvertauschungen (protokolliert in Zeile 7 von Algorithmus 1.8) beschreibt. Beispiele werden in der Vortragsübung betrachtet, in einer Implementierung und mit Papier und Bleistift wird man \mathbf{P} typischerweise nicht als volle Matrix protokollieren.
- Eine Permutationsmatrix für den Tausch von Zeile i und j ist die Einheitsmatrix entsprechender Dimension, wobei Zeile i und j getauscht sind. Die Permutationsmatrix \mathbf{P} für den gesamten Algorithmus ergibt sich als Produkt der elementaren Permutationsmatrizen.



1. Lineare Gleichungssysteme I

Bemerkungen:

- Ist $\det(\mathbf{A}) \neq 0$, so terminiert Algorithmus 1.8 mit einer LR-Zerlegung von \mathbf{PA} . Insbesondere existiert deshalb für jede reguläre Matrix \mathbf{A} eine LR-Zerlegung mit Pivotisierung.
- Auch Spaltenpivotisierung und vollständige Pivotisierung (simultane Zeilen- und Spaltenvertauschungen) sind möglich. Dies leistet die ViPLab/Octave-Funktion $[L, R, P] = lu(A)$.
- Aus den Matrizen L, R, P erhält man die Lösung des LGS $\mathbf{Ax} = \mathbf{b}$ analog zu eben unter Berücksichtigung der Tauschoperationen:

$$\begin{aligned}\mathbf{Ax} = \mathbf{b} &\Leftrightarrow \mathbf{PAx} = \mathbf{Pb} \\ &\Leftrightarrow \underbrace{\mathbf{L} \mathbf{Rx}}_y = \mathbf{Pb} \\ &\Leftrightarrow \mathbf{Ly} = \mathbf{Pb} \text{ und } \mathbf{Rx} = y\end{aligned}$$



1. Lineare Gleichungssysteme I

Bemerkungen:

- Gauß bzw. LR-Zerlegung mit Vorwärts-Rückwärts-Einsetzen liefert die Lösung eines LGS in endlich vielen Schritten. Deshalb spricht man von einem **direkten** Verfahren im Gegensatz zu **iterativen** Verfahren im nächsten Kapitel.
- Ist \mathbf{A} symmetrisch und positiv definit, so existiert eine sogenannte **Cholesky-Zerlegung**, d. h. $\mathbf{A} = \mathbf{C}\mathbf{C}^T$ mit \mathbf{C} untere Dreiecksmatrix (aber i. A. keine 1-Diagonale, dies ist also nicht die LR-Zerlegung). Der Rechenaufwand halbiert sich gegenüber der LR-Zerlegung, und die Cholesky-Zerlegung hat dieselben Einsatzmöglichkeiten wie LR-Zerlegung \rightarrow gleich.



Anwendungen und Vorteile der LR-Zerlegung



1. Lineare Gleichungssysteme I

Wir betrachten nun einige Beispiele, in denen die systematische LR-Zerlegung große Vorteile gegenüber der ad-hoc Gauß-Elimination aufweist:

LGS mit vielen rechten Seiten: $\mathbf{Ax} = \mathbf{b}$, $\mathbf{Ay} = \mathbf{c}$, $\mathbf{Az} = \mathbf{d}$, ...

- ① Bestimme einmalig die LR-Zerlegung von \mathbf{A} : Aufwand $\mathcal{O}(N^3)$
- ② Löse jedes der k Systeme durch Vorwärts- und Rückwärtseinsetzen, Aufwand pro System (bei existierender LR-Zerlegung): $\mathcal{O}(N^2)$
- ③ Gesamtaufwand im Vergleich zur k -fachen Gauß-Elimination: $\mathcal{O}(N^3 + kN^2)$ statt $\mathcal{O}(kN^3)$, also für $k < N$ deutlich effizienter



1. Lineare Gleichungssysteme I

Matrixinvertierung, explizite Bestimmung von \mathbf{A}^{-1}

- ① Bestimme die LR-Zerlegung $\mathbf{L}\mathbf{R} = \mathbf{A} \in \mathbb{R}^{N \times N}$: Aufwand $\mathcal{O}(N^3)$
- ② Löse die N Gleichungssysteme

$$(\mathbf{L}\mathbf{R})\mathbf{x}_n = \mathbf{A}\mathbf{x}_n = \mathbf{e}_n \quad n = 1, \dots, N$$

mittels Vorwärts- und Rückwärtseinsetzen: Aufwand $N \cdot \mathcal{O}(2N^2) = \mathcal{O}(N^3)$
Hierbei sind $\mathbf{e}_1, \dots, \mathbf{e}_N$ die Einheitsvektoren in \mathbb{R}^N .

- ③ Es gilt dann:

$$\mathbf{A}^{-1} = \begin{pmatrix} | & \cdots & | \\ \mathbf{x}_1 & \cdots & \mathbf{x}_N \\ | & \cdots & | \end{pmatrix}.$$

Der Gesamtaufwand ist also nur um einen von N unabhängigen Faktor höher als die Bestimmung der LR-Zerlegung und identisch zum Gauß-Jordan Verfahren.



1. Lineare Gleichungssysteme I

Determinantenberechnung

Die Berechnung von $\det(\mathbf{A})$ mit der Leibniz-Formel

$$\det(\mathbf{A}) = \sum_{\pi \in \Pi_N} \text{sign}(\pi) a_{\pi(1)1} \cdot a_{\pi(2)2} \cdots \cdots a_{\pi(N)N}.$$

Dabei ist Π_N die Menge aller Permutationen von $\{1, \dots, N\}$, und sign das Vorzeichen einer Permutation.

erfordert $\mathcal{O}(NN!)$ Operationen. Das ist verboten teuer wegen der Fakultät!

Man kann zeigen: Determinanten von Dreiecksmatrizen berechnen sich als Produkt der Hauptdiagonaleinträge. Dies liefert für die LR-Zerlegung:

$$\det(\mathbf{A}) = \det(\mathbf{LR}) = \underbrace{\det(\mathbf{L})}_{=1} \cdot \det(\mathbf{R}) = \det(\mathbf{R}) = r_{11} \cdots \cdots r_{NN}.$$

Der Rechenaufwand ist $\mathcal{O}(N^3)$, bzw. $\mathcal{O}(N)$ bei existierender Zerlegung.



Zusammenfassung

- Die LR-Zerlegung ist eine strukturierte Form der Gauß-Elimination, und erfordert den gleichen arithmetischen Aufwand.
- Sobald die LR-Zerlegung einmal berechnet ist, ist die Lösung eines LGS einfach durch Vorwärts- und Rückwärtseinsetzen möglich.
- Mit Pivotisierung (Zeilentausch) können auch LR-Zerlegungen berechnet werden, bei denen die Gauß-Elimination fehlschlägt.
- Die LR-Zerlegung ist effizienter als Standardverfahren bei der Determinantenberechnung und bei der simultanen Lösung mehrerer LGS.
- Für „große“ Matrizen ist die LR-Zerlegung zu teuer, hier sollten iterative Verfahren verwendet werden → nächste VL.



Hausaufgaben

- Besuchen Sie die nullte Vortragsübung, und üben Sie den Umgang mit ViPLab ein. Bei Problemen und Fragen helfen die Tutorinnen in den Programmiersprechstunden gerne weiter. In ILIAS (Ordner ViPLab-Übung) sind die Termine für die nächsten 7 Tage zusammengefasst.
- Lösen Sie im Vorgriff auf die Vortragsübung die Beispielaufgaben auf den folgenden Seiten.
- Verinnerlichen Sie für die Vorlesung nächste Woche die \mathcal{O} -Notation und die damit verbundene Asymptotik. Überlegen Sie sich insbesondere, warum nach der initialen Berechnung der LR-Zerlegung zahlreiche Eigenschaften (wie Determinanten etc.) faktisch umsonst bestimmbar sind.



Beispieldaufgaben



1. Lineare Gleichungssysteme I

Wichtig

Die folgenden Aufgaben dienen als Vorbereitung und Ergänzung der Vortragsübung. Sie sind typischerweise etwas „einfacher“ als die Aufgaben aus der Vortragsübung, und fokussieren einzelne Teilschritte.

Auch wenn die Bearbeitung freiwillig ist, wird sie dringend empfohlen!

Musterlösungen werden nicht zur Verfügung gestellt, stattdessen gibt es Lösungshinweise und hoffentlich korrekte Endergebnisse.



LR-Zerlegung

Berechnen Sie die LR-Zerlegung ohne Pivotisierung für die folgende Matrix:

$$\mathbf{A} = \begin{pmatrix} 5 & 4 & 0 \\ 10 & 10 & 2 \\ -5 & 0 & 7 \end{pmatrix}$$

Bestimmen Sie dann die Determinante von \mathbf{A} .



1. Lineare Gleichungssysteme I

Lösungshinweise: Hier genügt es, Algorithmus 1.5 anzuwenden, d.h. eine strukturierte Gauß-Elimination mit Speicherung der Eliminationsfaktoren in \mathbf{L} und des Ergebnisses in \mathbf{R} durchzuführen. Die Determinante kann danach faktisch abgelesen werden.

Ergebnis:

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 2 & 1 \end{pmatrix} \quad \mathbf{R} = \begin{pmatrix} 5 & 4 & 0 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{pmatrix} \quad \det(\mathbf{A}) = 30$$

Kreative Verständnisübungen (ohne Angabe von Ergebnissen): Bestimmen Sie mit Hilfe der LR-Zerlegung zusätzlich \mathbf{A}^{-1} . Lösen Sie das LGS für die rechte Seite $\mathbf{b} = (1 \ 1 \ 1)^T$.



1. Lineare Gleichungssysteme I

LR-Gauß

Lösen Sie das folgende lineare Gleichungssystem:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 0 & 3 & 1 & 0 \\ 0 & -2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 4 & 0 \\ 0 & 1 & 4 & 0 \\ 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$



1. Lineare Gleichungssysteme I

Lösungshinweise: Scharfes Hinsehen ergibt, dass die beiden Matrizen eine LR-Zerlegung darstellen: Wir lesen die Dreiecksgestalt ab, und beachten die Einheitsdiagonale der linken unteren Dreiecksmatrix. Es reicht also, zunächst das System $\mathbf{Ly} = \mathbf{b}$ durch Vorwärtseinsetzen, und dann das System $\mathbf{Rx} = \mathbf{y}$ durch Rückwärtseinsetzen zu lösen.

Ergebnis:

$$\mathbf{x} = (-4 \quad -7 \quad 2 \quad 3)^T$$

Kreative Verständnisübungen (ohne Angabe von Ergebnissen): Berechnen Sie $\mathbf{A} = \mathbf{LR}$ durch Matrix-Matrix-Multiplikation, und bestimmen Sie ausgehend von \mathbf{A} die LR-Zerlegung. Berechnen Sie weiterhin die Inverse und die Determinante von \mathbf{A} .



1. Lineare Gleichungssysteme I

Pivotisierung

Gegeben sei die Matrix

$$\mathbf{A} = \begin{pmatrix} 2 & 5 & 3 \\ 4 & \beta & 8 \\ 1 & 4.5 & 9.5 \end{pmatrix}$$

mit einem Parameter $\beta \in \mathbb{R}$. Für welchen Wert von β kann die LR-Zerlegung von \mathbf{A} nur mit Pivotisierung bestimmt werden?



1. Lineare Gleichungssysteme I

Lösungshinweise: Der pragmatische Lösungsansatz besteht darin, einfach solange Algorithmus 1.5 durchzurechnen mit dem Parameter β , bis einmal durch Null geteilt würde. In diesem Schritt des Algorithmus kann man dann ablesen, für welchen Wert von β die Division durch Null auftritt.

Ergebnis:

$$\beta = 10$$

Kreative Verständnisübungen (ohne Angabe von Ergebnissen): Bestimmen Sie die LR-Zerlegung mit Zeilenpivotisierung.



Ergänzungen



Wichtig

Die verbleibenden Folien dieses Kapitels sind nicht prüfungsrelevant.

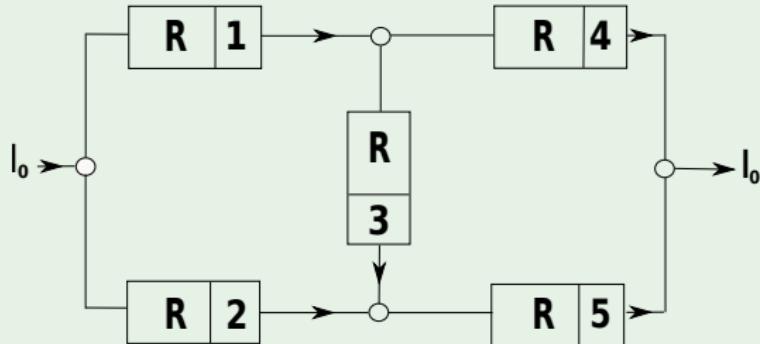
Sie dienen stattdessen als Ergänzung der Aspekte, die in der VL sehr kurz behandelt wurden.

Fragen zu diesen Folien werden in den Sprechstunden beantwortet.



1. Lineare Gleichungssysteme I

Elektrische Netzwerke



Knoten

R_i Widerstand i der (festen) Stärke $R > 0$

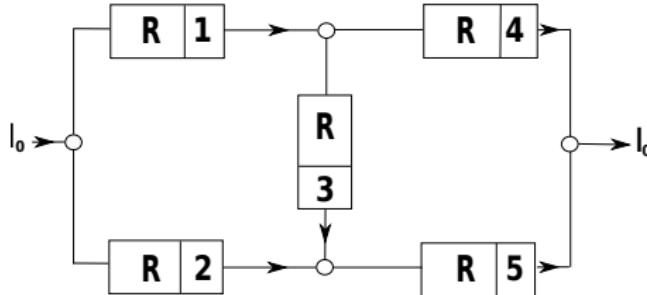
I_0 Gegebene Ein-/ Ausgangsstromstärke

> Orientierung (beliebig, aber fest)

Wie sehen die Stromstärken I_i in den Einzelleitungen zwischen 2 Knoten aus?



1. Lineare Gleichungssysteme I



Für $i = 1, \dots, 5$ gilt:

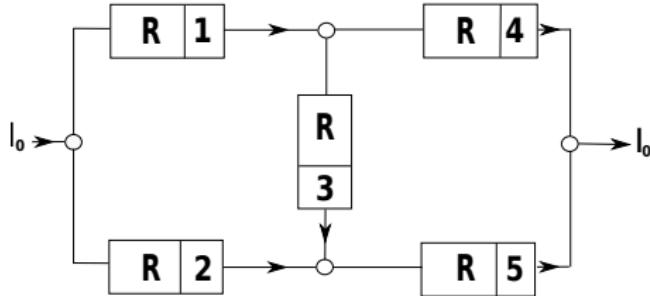
1. Kirchhoff-Gesetz: In jedem Knoten addieren sich die gerichteten Stromstärken I_i zu Null.

2. Kirchhoff-Gesetz: In jeder Masche (Umlauf/Zyklus) addieren sich die (gerichteten) Spannungen U_i zu Null.

Ohm'sches Gesetz: $U_i = R I_i$



1. Lineare Gleichungssysteme I



1. Kirchhoff-Gesetz: In jedem Knoten addieren sich die gerichteten Stromstärken I_i zu Null.

Beispiel:

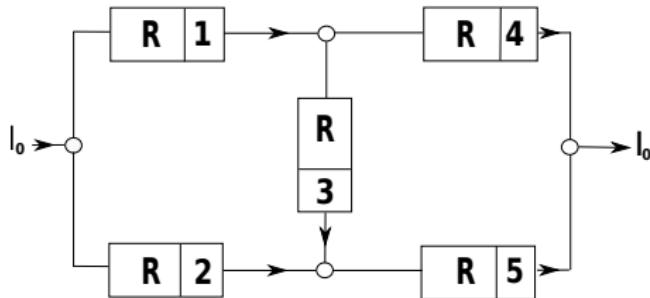
$$0 = I_0 - I_1 - I_2$$

Äquivalent:

$$-I_1 - I_2 = -I_0$$



1. Lineare Gleichungssysteme I



2. Kirchhoff-Gesetz: In jeder Masche (Umlauf/Zyklus) addieren sich die (gerichteten) Spannungen U_i zu Null.

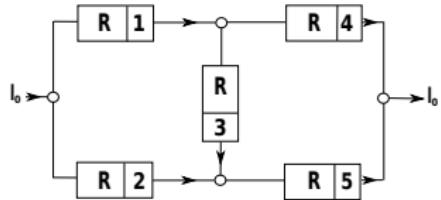
Beispiel:

$$0 = U_1 + U_3 - U_2$$



1. Lineare Gleichungssysteme I

Zusammen: (1. Kirchhoff-Gesetz)



$$\begin{array}{lcl} -I_1 & -I_2 & = -I_0 \\ I_1 & -I_3 & -I_4 & = 0 \\ I_2 & +I_3 & -I_5 & = 0 \\ I_4 & +I_5 & = I_0 \end{array}$$

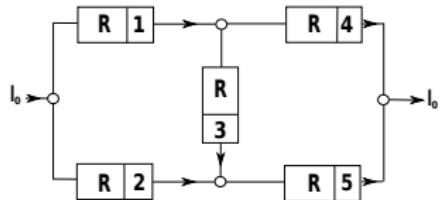
Das ist linear abhängig, denn Addition der ersten drei Gleichungen ergibt „minus“ die letzte Gleichung. Wir reduzieren das Problem also auf drei Gleichungen:

$$\begin{array}{lcl} -I_1 & -I_2 & = -I_0 \\ I_1 & -I_3 & -I_4 & = 0 \\ I_2 & +I_3 & -I_5 & = 0 \end{array}$$



1. Lineare Gleichungssysteme I

Das System ist unterbestimmt mit 3 Gleichungen in 5 Unbekannten. Um das System zu schließen, ergänzen wir mit dem 2. Kirchhoff-Gesetz:



$$\begin{array}{ccc|c} -I_1 & -I_2 & & = & -I_0 \\ I_1 & & -I_3 & -I_4 & = & 0 \\ I_2 & +I_3 & & -I_5 & = & 0 \end{array}$$

$$\begin{array}{ccccc|c} U_1 & -U_2 & +U_3 & & & = & 0 \\ & U_3 & -U_4 & +U_5 & & = & 0 \end{array}$$

Insgesamt haben wir die nötigen 5 linearen Gleichungen, die aber noch nicht zusammenpassen.



1. Lineare Gleichungssysteme I

$$\begin{array}{ccccc} -I_1 & -I_2 & & & = -I_0 \\ I_1 & & -I_3 & -I_4 & = 0 \\ & I_2 & +I_3 & & = 0 \\ U_1 & -U_2 & +U_3 & & = 0 \\ & & U_3 & -U_4 & +U_5 = 0 \end{array}$$

Mit dem Ohm'schen Gesetz übersetzen wir U_i in I_i (Erinnerung: R ist konstant in diesem Beispiel, zur Vereinfachung):

$$\begin{array}{ccccc} -I_1 & -I_2 & & & = -I_0 \\ I_1 & & -I_3 & -I_4 & = 0 \\ & I_2 & +I_3 & -I_5 & = 0 \\ RI_1 & -RI_2 & +RI_3 & & = 0 \\ & & RI_3 & -RI_4 & +RI_5 = 0 \end{array}$$

Somit haben wir 5 lineare Gleichungen für die 5 gesuchten Stromstärken.



1. Lineare Gleichungssysteme I

$$\begin{array}{ccccc} -I_1 & -I_2 & & & = -I_0 \\ I_1 & & -I_3 & -I_4 & = 0 \\ & I_2 & +I_3 & & = 0 \\ RI_1 & -RI_2 & +RI_3 & & = 0 \\ & RI_3 & -RI_4 & +RI_5 & = 0 \end{array}$$

Wir fassen die Gleichungen in einem LGS für die unbekannten Größen (Stromstärken) I_1, \dots, I_5 zusammen, und erhalten:

$$\begin{pmatrix} -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ R & -R & R & 0 & 0 \\ 0 & 0 & R & -R & R \end{pmatrix} \begin{pmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{pmatrix} = \begin{pmatrix} -I_0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Verständnisübung: Matrix-Vektor-Multiplikation dieses LGS durchführen, um das Modell wiederzufinden.

Es gilt $\det(\mathbf{A}) = 8R^2 \neq 0$, das LGS ist eindeutig lösbar gemäß HM123.

HM123 ist ein Platzhalter für die bisherigen HM-VLen.



2. Lineare Gleichungssysteme II



Ich empfehle Ihnen diesen Modus zur Nachahmung. Schwerlich werden Sie je wieder direkt eliminieren, wenigstens nicht, wenn Sie mehr als zwei Unbekannte haben. Das indirekte Verfahren lässt sich halb im Schlaf ausführen oder man kann während desselben an andere Dinge denken.

C.F. Gauß in einem Brief an Gerling, 1823



2. Lineare Gleichungssysteme II

Inhalte und Ziele dieser Vorlesungseinheit

- Tieferes Verständnis der Probleme bei der Lösung mathematischer Aufgaben am Computer, am Beispiel der Lösung linearer Gleichungssysteme
- Beispiele für die Quantifizierung von **Fehlern**, also eines zentralen Konzepts der Numerik
- Bereitstellung und Wiederholung zentralen Basisvokabulars „unterwegs“: Normen, Skalarprodukte, Eigenwerte, ...
- Lösungsverfahren für dünn besetzte LGS, die effizienter als die LR-Zerlegung (die Gauß-Elimination) sind



Störungstheorie und Einfluss von Rundungsfehlern



2. Lineare Gleichungssysteme II

Beispiel 2.1 (Motivation der Störungstheorie I)

Wir betrachten das harmlos aussehende lineare Gleichungssystem

$$\begin{pmatrix} 1.2969 & 0.8648 \\ 0.2161 & 0.1441 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0.8642 \\ 0.1440 \end{pmatrix}.$$

Octave/ViPLab liefert in Standardgenauigkeit (double precision)

$(x, y)^T = (2, -2)^T$, Papier+Bleistift auch. Die Lösung ist also vermutlich korrekt.

Wir stören nun die rechte Seite minimal:

$$\begin{pmatrix} 1.2969 & 0.8648 \\ 0.2161 & 0.1441 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0.86419999 \\ 0.14400001 \end{pmatrix}$$

Octave/ViPLab liefert: $(x, y)^T = (0.9911, -0.4870)^T$, Papier und Bleistift auch.
Das ist unintuitiv.



2. Lineare Gleichungssysteme II

Beispiel 2.2 (Motivation der Störungstheorie II)

Wir betrachten wieder das lineare Gleichungssystem

$$\begin{pmatrix} 1.2969 & 0.8648 \\ 0.2161 & 0.1441 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0.8642 \\ 0.1440 \end{pmatrix}$$

und verwenden **einfache Genauigkeit**, also nur 8 statt 16 signifikante Stellen:

```
A=[1.2969 0.8648; 0.2161 0.1441]    b=[0.8642;0.1440]    A\b  
As=single(A)                            bs=single(b)          As\bs
```

Matlab liefert das Ergebnis $(x, y)^T = (1.3332, -1.0000)^T$ und Octave/ViPLab das Ergebnis $(x, y)^T = (0.4613, 0.3076)^T$. Beides ist sehr falsch.



2. Lineare Gleichungssysteme II

Fragestellung in diesem Unterkapitel

Warum resultieren kleine Störungen in den Daten (in der Eingabe) in so großen Änderungen im Ergebnis? Können wir vielleicht sogar vorab beurteilen und sogar quantifizieren, ob wir den Ergebnissen des Computers trauen dürfen?

Wir betrachten dazu eine reguläre Matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, die rechte Seite $\mathbf{b} \in \mathbb{R}^N$, und die eindeutige Lösung \mathbf{x} von $\mathbf{Ax} = \mathbf{b}$. Der Gauß-LR Algorithmus 1.6 berechnet auf einem Computer die Lösung $\tilde{\mathbf{x}}$. Im Allgemeinen gilt $\mathbf{x} \neq \tilde{\mathbf{x}}$, weil $\tilde{\mathbf{x}}$ mit **Rundungsfehlern** behaftet ist: Der Computer rechnet mit einer endlich-dimensionalen Gleitkomma-Repräsentation reeller Zahlen. Wir interessieren uns für den Fehler in $\tilde{\mathbf{x}}$ (den „Abstand“ zwischen \mathbf{x} und $\tilde{\mathbf{x}}$).



2. Lineare Gleichungssysteme II

Wir definieren einen verallgemeinerten „Abstand“ im \mathbb{R}^N mit Hilfe des aus HM123 bekannten Skalarprodukts:

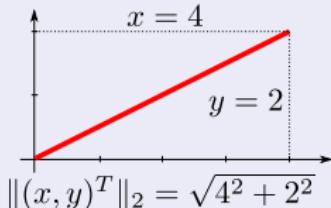
Definition 2.3 (Euklidisches Skalarprodukt und Euklidische Norm)

Das Euklidische Skalarprodukt zweier Vektoren $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ ist

$$(\mathbf{x}, \mathbf{y})_2 := \mathbf{x}^T \mathbf{y} = \sum_{n=1}^N x_n y_n.$$

Die Euklidische Norm (Länge) eines Vektors $\mathbf{x} \in \mathbb{R}^N$ ist

$$\|\mathbf{x}\|_2 := \sqrt{(\mathbf{x}, \mathbf{x})_2} = \left(\sum_{n=1}^N x_n^2 \right)^{\frac{1}{2}}.$$



Norm und Satz des Pythagoras

Es gilt $\|\mathbf{x}\|_2 > 0$ genau dann wenn $\mathbf{x} \neq \mathbf{0}$, das entspricht unserer Intuition.



2. Lineare Gleichungssysteme II

Im Vorgriff auf später definieren wir analog:

Definition 2.4 (p -Normen)

Die p -Normen auf \mathbb{R}^N sind definiert durch

$$\|\mathbf{y}\|_p := \begin{cases} \sqrt[p]{|y_1|^p + \cdots + |y_N|^p} & 1 \leq p < \infty, \\ \max\{|y_1|, \dots, |y_N|\} & p = \infty. \end{cases}$$

Wichtige p -Normen sind:

$$p = 1: \quad \|\mathbf{y}\|_1 = |y_1| + \cdots + |y_N| \quad (\text{Betragssummennorm})$$

$$p = 2: \quad \|\mathbf{y}\|_2 = \sqrt{|y_1|^2 + \cdots + |y_N|^2} \quad (\text{Euklidische Norm})$$

$$p = \infty: \quad \|\mathbf{y}\|_\infty = \max\{|y_1|, \dots, |y_N|\} \quad (\text{Maximumsnorm})$$



2. Lineare Gleichungssysteme II

Mit Hilfe von Normen können wir **Fehler** quantifizieren:

Definition 2.5 (Absoluter und relativer Fehler)

Sei \mathbf{x} die exakte eindeutige Lösung von $\mathbf{A}\mathbf{x} = \mathbf{b}$ und sei $\tilde{\mathbf{x}}$ eine gestörte Lösung, die beispielsweise vom Computer berechnet wurde. Dann heißt

$$\|\mathbf{x} - \tilde{\mathbf{x}}\|_2$$

absoluter Fehler, und für $\mathbf{x} \neq \mathbf{0}$ ist

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2}$$

der **relative Fehler**.

Analog sind Fehler in den Daten definiert, z. B. $\|\mathbf{b} - \tilde{\mathbf{b}}\|_2$ oder $\frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|_2}{\|\mathbf{b}\|_2}$; und ebenso für andere zugrundeliegende Normen wie die Maximumsnorm $\|\cdot\|_\infty$



2. Lineare Gleichungssysteme II

Ein gängiger Trick bei der (ersten) Diskussion von Fehlern ist die Betrachtung eines Problems mit bekannter Lösung, d. h. wir wählen \mathbf{x} und setzen $\mathbf{b} := \mathbf{A}\mathbf{x}$:

Beispiel 2.6 (Stationäres Raclette ohne Gitterweite h)

Wir berechnen $\tilde{\mathbf{x}}$ als Lösung von $\mathbf{A}\tilde{\mathbf{x}} = \mathbf{b}$ in doppelter Genauigkeit mit ViPLab, für

$$\mathbf{A}_T = \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix} \Rightarrow \mathbf{b} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}.$$

Alle vorkommenden Zahlen in der Eingabe sind exakt in Gleitkomma repräsentierbar. Für den absoluten Fehler erhalten wir:

N	$\ \mathbf{x} - \tilde{\mathbf{x}}\ _2$
10	1.11e-16
20	1.76e-15
100	7.73e-14
1000	4.14e-12
10000	9.19e-10



2. Lineare Gleichungssysteme II

Beispiel 2.7 (Hilbert-Matrix)

$$\mathbf{A}_H = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \dots \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \dots \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix} \Rightarrow \quad \mathbf{b} = \begin{pmatrix} \sum a_{1n} \\ \vdots \\ \sum a_{Nn} \end{pmatrix}.$$

Hier sind nicht alle vorkommenden Werte exakt in Gleitkomma repräsentierbar.
Für den absoluten Fehler erhalten wir in doppelter Genauigkeit mit ViPLab:

N		5		10		20		100		500
$\ \mathbf{x} - \tilde{\mathbf{x}}\ _2$		7.81e-13		5.24e-04		8.25e+01		6.82e+02		1.25e+05

ViPLab-Demo: VI02 in ILIAS

Für das winzige Problem mit $N = 10$ stimmen nur die ersten drei Nachkommastellen! Das ist absolut inakzeptabel!



Grundlagen der Störungstheorie



2. Lineare Gleichungssysteme II

Wieso liefert die Gauß-Elimination katastrophale Näherungen?

Beim Vergleich aller vier Beispiele fällt auf, dass (akkumulierte) Rundungsfehler nicht allein für große Fehler verantwortlich sein können: Scheinbar ist nur das dritte Beispiel „einfach“ vom Computer lösbar.

Konditionszahlen messen die „Schwierigkeit“ eines Problem

Allgemein ist die Konditionszahl ein Maß für die Sensitivität eines Problems unter dem Einfluss von Störungen. Gründe für Störungen sind bspw. endliche Arithmetik (Rundungsfehler etc.), aber auch Daten- und Messfehler. In Beispiel 2.1 haben wir Datenfehler durch Änderung von \mathbf{b} emuliert.



2. Lineare Gleichungssysteme II

Auch Koeffizientenmatrizen sind „Daten“ und können fehlerbehaftet sein. Wir erweitern deshalb Definition 2.4 (p-Normen):

Definition 2.8 (Induzierte Matrixnorm)

Sei $\|\cdot\|$ eine (Vektor-) Norm auf \mathbb{R}^N und $\mathbf{A} \in \mathbb{R}^{N \times N}$. Die zugeordnete induzierte Matrixnorm für $\mathbf{A} \in \mathbb{R}^{N \times N}$ ist

$$\|\mathbf{A}\| := \sup_{\mathbf{y} \in \mathbb{R}^N \setminus \{\mathbf{0}\}} \frac{\|\mathbf{Ay}\|}{\|\mathbf{y}\|} = \sup\{\|\mathbf{Ay}\| \mid \mathbf{y} \in \mathbb{R}^N, \|\mathbf{y}\| = 1\}.$$

Es gilt für beliebiges $\mathbf{y} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$ die **Submultiplikativität**:

$$\|\mathbf{Ay}\| = \frac{\|\mathbf{Ay}\|}{\|\mathbf{y}\|} \|\mathbf{y}\| \leq \|\mathbf{A}\| \|\mathbf{y}\|$$

Das folgt direkt aus der ersten Supremums-Definition.



2. Lineare Gleichungssysteme II

Damit können wir die Konditionszahl definieren. Sie quantifiziert die „Fehleranfälligkeit“ (Sensitivität) eines LGS mit gegebener Koeffizientenmatrix.

Definition 2.9 (Konditionszahl von Matrizen)

Sei $\mathbf{A} \in \mathbb{R}^{N \times N}$ regulär und $\|\cdot\|$ eine beliebige Matrixnorm. Dann heißt der Ausdruck

$$\kappa(\mathbf{A}) := \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$$

Konditionszahl von \mathbf{A} .

Um die Konditionszahl berechnen zu können, müssen wir also Matrixnormen ausrechnen, was wegen des Supremums in der Definition etwas unhandlich erscheint. Dazu betrachten wir drei Beispiele:



Für $p = 1$ und $p = \infty$ lässt sich die Matrixnorm fast ablesen:

Satz 2.10 (Zeilen- und Spaltensummennorm)

Sei $\mathbf{A} \in \mathbb{R}^{N \times N}$. Dann sind die zu $p = 1$ und $p = \infty$ gehörigen Matrixnormen explizit berechenbar als **Spaltensummennorm** und **Zeilensummennorm**:

$$\|\mathbf{A}\|_1 = \max_{n=1, \dots, N} \left(\sum_{m=1}^N |a_{mn}| \right) \quad \|\mathbf{A}\|_\infty = \max_{m=1, \dots, N} \left(\sum_{n=1}^N |a_{mn}| \right)$$



2. Lineare Gleichungssysteme II

Quiz:

$$\mathbf{A} = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & \ddots & \ddots & \ddots \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{pmatrix}$$

- $\|\mathbf{A}\|_1 = ?$ für alle $N \geq 3$
- $\|\mathbf{A}\|_\infty = ?$ für alle $N \geq 3$



2. Lineare Gleichungssysteme II

Für die zur Euklid-Norm ($p = 2$) gehörende Matrixnorm ist das nicht so einfach.
Wir können aber einen wichtigen Querbezug zur HM123 nutzen:

Definition 2.11 (Eigenwert und Eigenvektor)

Sei $\mathbf{A} \in \mathbb{R}^{N \times N}$. $\lambda \in \mathbb{C}$ heißt **Eigenwert (EW)** von \mathbf{A} , falls ein $\mathbf{y} \in \mathbb{C}^N \setminus \{\mathbf{0}\}$ existiert mit $\mathbf{A}\mathbf{y} = \lambda\mathbf{y}$. Der Vektor \mathbf{y} heißt **Eigenvektor (EV) zum Eigenwert λ** . Die Menge aller Eigenwerte

$$\text{sp}(\mathbf{A}) := \{\lambda \in \mathbb{C} \mid \lambda \text{ ist Eigenwert von } \mathbf{A}\}$$

heißt **Spektrum** von \mathbf{A} . Der Wert

$$\rho(\mathbf{A}) := \max\{ |\lambda| \mid \lambda \in \text{sp}(\mathbf{A})\}$$

ist der **Spektralradius** von \mathbf{A} .



2. Lineare Gleichungssysteme II

Eigenwerte erlauben es nun, die Euklid-Matrixnorm explizit zu berechnen:

Satz 2.12 (Spektralnorm)

Die zur Euklid-Vektornorm gehörige Matrixnorm heißt auch **Spektralnorm**, sie ist definiert durch

$$\begin{aligned}\|\mathbf{A}\|_2 &= \max\{\sqrt{|\lambda|} \mid \lambda \text{ ist Eigenwert von } \mathbf{A}^T \mathbf{A}\} \\ &= \max\{\sqrt{|\lambda|} \mid \lambda \in \text{sp}(\mathbf{A}^T \mathbf{A})\}.\end{aligned}$$

Ist \mathbf{A} symmetrisch, vereinfacht sich das deutlich:

$$\begin{aligned}\|\mathbf{A}\|_2 &= \max\{|\lambda| \mid \lambda \text{ ist Eigenwert von } \mathbf{A}\} \\ &= \max\{|\lambda| \mid \lambda \in \text{sp}(\mathbf{A})\} \\ &= \rho(\mathbf{A})\end{aligned}$$

Erinnerung an HM123: Eigenwerte lassen sich über das charakteristische Polynom berechnen. Beispiele werden in der Vortragsübung betrachtet.



Der Störungssatz



2. Lineare Gleichungssysteme II

Anwendung auf die Berechnung von Konditionszahlen:

Für allgemeine Matrizen \mathbf{A} gilt $\kappa = \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$. Wir können also beispielsweise κ_1 oder κ_∞ einfach ausrechnen, sobald wir \mathbf{A}^{-1} kennen.

Der Vorteil der Spektralnorm wird für symmetrische Matrizen deutlich: Mit den betragsmäßig größten bzw. kleinsten Eigenwerten $\lambda_{\max}(\mathbf{A})$ und $\lambda_{\min}(\mathbf{A})$ gilt nämlich:

$$\kappa_2(\mathbf{A}) = \frac{|\lambda_{\max}(\mathbf{A})|}{|\lambda_{\min}(\mathbf{A})|}.$$

Dies folgt mit ein wenig Erinnerung an HM123 aus

$$\|\mathbf{A}\|_2 = |\lambda_{\max}| \quad \text{und} \quad \|\mathbf{A}^{-1}\|_2 = |\lambda_{\max}(\mathbf{A}^{-1})| = |\lambda_{\min}(\mathbf{A})|^{-1}.$$

Wir benötigen hier also die Inverse nicht. Nächste Woche: Numerische Berechnung von Eigenwerten.



2. Lineare Gleichungssysteme II

Beispiel 2.13 (Konditionszahlen der Beispielmatrizen)

Für das 2×2 Problem aus der Motivation gilt: $\kappa_2(\mathbf{A}) \approx 10^8$. Für die Tridiagonalmatrix \mathbf{A}_T aus Beispiel 2.6 und die Hilbertmatrix \mathbf{A}_H aus Beispiel 2.7 ergibt sich:

N	$\kappa_2(\mathbf{A}_T)$	$\det \mathbf{A}_T$
5	3.93	6
20	178.06	21
100	$4.13 \cdot 10^3$	101
500	$1.02 \cdot 10^5$	501

N	$\kappa_2(\mathbf{A}_H)$	$\det \mathbf{A}_H$
5	$4.77 \cdot 10^5$	$3.75 \cdot 10^{-12}$
10	$1.60 \cdot 10^{13}$	$2.16 \cdot 10^{-53}$
20	$1.85 \cdot 10^{18}$	$-1.10 \cdot 10^{-195}$

Wir sehen für die Hilbertmatrix, dass die Determinante schon für sehr kleine N faktisch verschwindet. Deshalb nennen wir die Hilbertmatrix schon für kleine N **numerisch singulär**, dies ist ein Extrembeispiel.

Für alle Matrizen korreliert die Konditionszahl mit dem Fehler. Dies wollen wir nun im Hauptresultat dieses Unterkapitels quantifizieren.



2. Lineare Gleichungssysteme II

Satz 2.14 (Fehlerverstärkung bei LGS)

Sei $\mathbf{A}\mathbf{x} = \mathbf{b}$ ein LGS mit exakter Lösung \mathbf{x} . Wir betrachten eine Störung der rechten Seite, d. h. wir untersuchen das gestörte LGS $\mathbf{A}\tilde{\mathbf{x}} = \mathbf{b} + \Delta\mathbf{b}$. Für den relativen Fehler gilt dann

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \kappa(\mathbf{A}) \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|},$$

unabhängig davon, mit welchem Algorithmus $\tilde{\mathbf{x}}$ bestimmt wurde. κ muss dabei mit der von $\|\cdot\|$ induzierten Matrixnorm definiert sein.

Die Konditionszahl ist also ein Maß dafür, wie stark Fehler in der rechten Seite **maximal** beim Lösen eines LGS verstärkt werden (obere Schranke). Ähnliches gilt für Eingangsfehler in \mathbf{A} , wenn wir das gestörte System $(\mathbf{A} + \Delta\mathbf{A})\tilde{\mathbf{x}} = \mathbf{b}$ betrachten. Der Fehler kann quantitativ durchaus unterschiedlich sein, je nachdem welche Norm wir betrachten.



2. Lineare Gleichungssysteme II

Beweis: Wir setzen $\Delta x = x - \tilde{x}$.

- ① Die Submultiplikativität (2.2) liefert nach Übergang zur Norm:

$$b = Ax \quad \Rightarrow \quad \|b\| = \|Ax\| \leq \|A\| \|x\|,$$

Das können wir einfach umstellen zu

$$\frac{1}{\|x\|} \leq \frac{\|A\|}{\|b\|}.$$

- ② Einsetzen der Definition von \tilde{x} und Ausmultiplizieren liefert

$$\Delta x = x - \tilde{x} = x - A^{-1}(b - \Delta b) \quad \Rightarrow \quad \Delta x = \underbrace{x - A^{-1}b}_{=0} - A^{-1}\Delta b,$$

woraus wir durch Übergang zur Norm mit der Submultiplikativität erhalten:

$$\|\Delta x\| \leq \|A^{-1}\| \|\Delta b\|$$

- ③ Fassen wir beide Schritte zusammen, so erhalten wir

$$\frac{\|x - \tilde{x}\|}{\|x\|} = \frac{\|\Delta x\|}{\|x\|} \leq \|A\| \|A^{-1}\| \frac{\|\Delta b\|}{\|b\|} = \kappa(A) \frac{\|\Delta b\|}{\|b\|}. \quad \square$$



2. Lineare Gleichungssysteme II

Beispiel 2.15 (exakte Arithmetik, vgl. Beispiel 2.1)

Für das LGS

$$\begin{pmatrix} 1.2969 & 0.8648 \\ 0.2161 & 0.1441 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0.8642 \\ 0.1440 \end{pmatrix}$$

liefern Octave/ViPLab/Papier&Bleistift die Lösung $(x, y)^T = (2, -2)^T$, und für das gestörte System

$$\begin{pmatrix} 1.2969 & 0.8648 \\ 0.2161 & 0.1441 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0.86419999 \\ 0.14400001 \end{pmatrix}$$

die Lösung $(x, y)^T = (0.9911, -0.4870)^T$.

Wir errechnen: $\kappa_2(\mathbf{A}) \approx 10^8$; sowie $\|\mathbf{b}\|_2 \approx 1$ und $\|\Delta\mathbf{b}\|_2 \approx 10^{-8}$, also
 $\frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2} \approx 10^{-8}$.

Der Satz liefert also einen relative Änderung der Lösung von ≈ 1 , es wird also eine Änderung in (maximal) der ersten Nachkommastelle postuliert. Das sehen wir, die Schranke zur Verstärkung ist also hinreichend scharf.



Motivation iterativer Verfahren für LGS



2. Lineare Gleichungssysteme II

Aus VL 1 wissen wir für die Gauß- und Gauß-LR-Algorithmen:

- Die Rechenkomplexität ist kubisch, d. h. $\mathcal{O}(N^3)$ arithmetische Operationen für Koeffizientenmatrizen $\mathbf{A} \in \mathbb{R}^{N \times N}$.

Aus der Störungstheorie wissen wir:

- Die Algorithmen liefert nur theoretisch die exakte Lösung \mathbf{x} .
- Rundungsfehler führen praktisch immer auf eine Approximation $\tilde{\mathbf{x}}$.
- Die Güte der Approximation hängt von der **Konditionszahl** $\kappa(\mathbf{A})$ ab.

Ziel #1: Es macht Sinn, von Anfang an nur Approximationen mit möglichst kontrollierbarer Genauigkeit zu berechnen.



2. Lineare Gleichungssysteme II

Wir schauen uns zwei Beispielmatrizen aus dieser VL aus der Vogelperspektive an:

$$\begin{pmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ \ddots & \ddots & \ddots & \\ & -1 & 2 & -1 \\ & & -1 & 2 \end{pmatrix} \quad \begin{pmatrix} -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ R & -R & R & 0 & 0 \\ 0 & 0 & R & -R & R \end{pmatrix}$$

Diese Koeffizientenmatrizen bestehen aus ziemlich vielen Nullen. In der Gauß-Elimination bzw. bei der Bestimmung der LR-Zerlegung multiplizieren wir also sehr oft einen Eintrag mit Null, oder schlimmer noch eine Null mit einer Null. Das ist eine Verschwendug von Rechenressourcen.

Ziel #2: Es macht Sinn, nur Operationen durchzuführen, bei denen das Nullergebnis nicht vorher klar ist.



2. Lineare Gleichungssysteme II

Wir definieren dazu, was intuitiv klar ist:

Definition 2.16 (dünn besetzte Matrix)

Eine Matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ heißt **dünn besetzt**, falls die Anzahl an Nicht-Nulleinträgen K pro Zeile im Mittel über alle Zeilen $K \ll N$ erfüllt. Eine dünnbesetzte Matrix besitzt also insgesamt $\mathcal{O}(KN)$ statt N^2 Einträge. Hierbei darf K nicht von N abhängen.

Für die Tridiagonalmatrix gilt $K = 3$ unabhängig von N . Für die Netzwerkmatrix gilt $K = 3$ für $N = 5$. Wir erinnern uns, dass diese Matrix für realistisch große Netzwerke immer dünner besetzt wird.

Solchen Matrizen werden wir im Verlauf der Veranstaltung häufig begegnen. Die Tridiagonalmatrix aus Beispiel 2.6 stammt bspw. aus der Diskretisierung der Wärmeleitungsgleichung, vgl. VL 0 und ab VL 8.



2. Lineare Gleichungssysteme II

Für dünn besetzte Matrizen existieren Speicherformate mit $\mathcal{O}(N)$ Speicher und $\mathcal{O}(KN)$ arithmetischen Operationen für die Matrix-Vektor Multiplikation.

Wir betrachten als Beispiel ein moderat großes System mit $N = 10^6$ und $K = 10$, das ist ungefähr die Problemgröße für die Wärmeleitungsgleichung in 3D bei $100 \times 100 \times 100$ Auswertungspunkten. Zur Speicherung nutzen wir doppelte Genauigkeit (8 Byte pro Wert):

	voll besetzte Matrix	dünn besetzte Matrix
Speicheraufwand	$8 \cdot 10^{12}$ Byte ≈ 8 TeraByte	$\approx 80 \cdot 10^6$ Byte ≈ 80 MegaByte
Rechenaufwand $\mathbf{A}\mathbf{x}$	≈ 1 TeraFLOP	≈ 1 MegaFLOP

Mega= 10^6 , Tera= 10^{12} , FLOP=floating point operations

Dünn besetzte Matrizen dieser Größe lassen sich auf jedem Telefon speichern und behandeln, voll besetzte Matrizen nicht!



2. Lineare Gleichungssysteme II

Die LR-Zerlegung besitzt noch einen weiteren gravierenden Nachteil:

Beispiel 2.17 (fill-in bei der LR-Zerlegung)

Die Faktoren L und R der dünn besetzten Matrix A sind i.A. dicht besetzt:

$$\underbrace{\begin{pmatrix} 1 & 1 & \dots & \dots & 1 \\ 1 & 2 & 0 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & 0 & \ddots & & \vdots \\ \vdots & & \ddots & & 0 \\ 1 & 0 & \dots & 0 & 2^{N-1} \end{pmatrix}}_{A \in \mathbb{R}^{N \times N}} = \underbrace{\begin{pmatrix} 1 & & & & \\ * & 1 & & & \\ \vdots & \ddots & \ddots & & \\ \vdots & & \ddots & \ddots & \\ * & * & \dots & * & 1 \end{pmatrix}}_L \underbrace{\begin{pmatrix} * & * & \dots & \dots & * \\ * & * & \dots & & * \\ \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & & \vdots \\ & & & & * \end{pmatrix}}_R$$

ViPLab-Demo: VI02 in ILIAS

Ziel #3: Wir wollen fill-in vermeiden, d. h. die Lösung des LGS ohne zusätzlichen Speicherbedarf berechnen.



Klassische iterative Verfahren



2. Lineare Gleichungssysteme II

Grundidee iterativer Verfahren:

Sei $\mathbf{A}\mathbf{x} = \mathbf{b}$ ein dünnes LGS, und $\mathbf{x}^{(0)} \in \mathbb{R}^N$ ein Startwert, bspw. $\mathbf{x}^{(0)} = \mathbf{0}$.

Klassische iterative Verfahren bestimmen sukzessiv Lösungsapproximationen gemäß der Vorschrift

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \text{update}(\mathbf{x}^{(k)}) \quad k = 0, 1, \dots$$

Hierbei ist $\text{update}(\mathbf{x}^{(k)})$ eine Platzhalter-Funktion, die eine Verbesserung der aktuellen Iterierten liefern soll. Wir fordern wenig verblüffend zudem

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}.$$

Wir überlegen uns nun verschiedene Möglichkeiten für die update-Funktion.



2. Lineare Gleichungssysteme II

Um einfach berechenbare update-Funktionen zu erhalten, sollten wir bekannte Größen heranziehen:

Definition 2.18 (Residuum und Defekt)

Für $\mathbf{y} \in \mathbb{R}^N$ sind das **Residuum** $\mathbf{r} = \mathbf{r}(\mathbf{y}) \in \mathbb{R}^N$ und der **Defekt** $\mathbf{d} = \mathbf{d}(\mathbf{y}) \in \mathbb{R}^N$ von $\mathbf{A}\mathbf{x} = \mathbf{b}$ die **berechenbaren** Größen

$$\mathbf{r}(\mathbf{y}) = \mathbf{A}\mathbf{y} - \mathbf{b} \qquad \qquad \mathbf{d}(\mathbf{y}) = \mathbf{b} - \mathbf{A}\mathbf{y}$$

Achtung: In der Literatur oft andersherum definiert.

Es gilt offensichtlich mit der exakten Lösung \mathbf{x} :

$$\mathbf{d}(\mathbf{y}) = \mathbf{0} \qquad \Leftrightarrow \qquad \mathbf{r}(\mathbf{y}) = \mathbf{0} \qquad \Leftrightarrow \qquad \mathbf{y} = \mathbf{x}$$

Beide Größen sind also ein Maß, wie gut \mathbf{y} das LGS erfüllt. Ihre Berechnung erfordert eine Matrix-Vektor Multiplikation, dies ist effizient gemäß unserer Ziele.



2. Lineare Gleichungssysteme II

Damit können wir eine erste prototypische update-Vorschrift definieren:

Definition 2.19 (Klassische Iteration, Defektkorrektur)

Sei $\mathbf{B} \in \mathbb{R}^{N \times N}$ mit $\det \mathbf{B} \neq 0$ gegeben. Zu einem gegebenen Startwert $\mathbf{x}^{(0)} \in \mathbb{R}^N$ heißt die Rechenvorschrift

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{B}^{-1} \mathbf{r}^{(k)} = \mathbf{x}^{(k)} - \underbrace{\mathbf{B}^{-1}(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b})}_{=: \text{update}(\mathbf{x}^{(k)})} = \mathbf{x}^{(k)} + \mathbf{B}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)})$$

Defektkorrektur-Verfahren zur Lösung des LGS $\mathbf{Ax} = \mathbf{b}$

Diese Iterationsvorschrift erfüllt das Ziel #1, es wird iterativ eine Approximation berechnet. Über die Güte der Approximation haben wir noch keine Aussage, und die Ziele #2 und #3 hängen von der Wahl von \mathbf{B} ab.



2. Lineare Gleichungssysteme II

Wir betrachten zwei Extrembeispiele zur Wahl von \mathbf{B} in der Iteration

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{B}^{-1} \mathbf{r}^{(k)} = \mathbf{x}^{(k)} - \mathbf{B}^{-1} (\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}).$$

Mit $\mathbf{B} := \mathbf{A}$ erhalten wir aus $\mathbf{x} = \mathbf{A}^{-1} \mathbf{b}$ die Identität

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \mathbf{A}^{-1} (\mathbf{A}\mathbf{x}^{(0)} - \mathbf{b}) = \mathbf{x}^{(0)} - \mathbf{x}^{(0)} + \mathbf{x},$$

d. h. wir erhalten im ersten Schritt die exakte Lösung, und zwar unabhängig vom Startvektor $\mathbf{x}^{(0)}$. Diese Wahl ist jedoch unpraktikabel, weil die Berechnung von \mathbf{A}^{-1} zu teuer ist und i. A. die Ziele #2 und #3 verletzt. Faktisch ist dies ein direktes Verfahren, tatsächlich sogar nur eine dumme Art, die LR-Zerlegung um überflüssige Rechnungen zu erweitern.



2. Lineare Gleichungssysteme II

Das andere Extrem ist die Wahl der Einheitsmatrix:

Verfahren 2.20 (Richardson-Iteration)

Mit $B_{RI} := I \in \mathbb{R}^{N \times N}$ und somit $B_{RI}^{-1} = I$ erhalten wir die **Richardson-Iteration**:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{r}^{(k)} = \mathbf{x}^{(k)} - (\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b})$$

Jeder Schritt erfordert eine Matrix-Vektor Multiplikation sowie zwei Vektor-Vektor Operationen. Das Verfahren erfüllt also alle drei Ziele. Es konvergiert aber nur sehr langsam, wenn überhaupt, und ist somit nicht praxisrelevant.

ViPLab-Demo: VI02 in ILIAS

Unser Ziel ist es jetzt, bessere Verfahren als Kompromiss zwischen diesen Extremen zu entwickeln: \mathbf{B} sollte „nahe“ \mathbf{A} und \mathbf{B}^{-1} einfach auszuwerten sein, d. h. $\mathbf{B}^{-1}\mathbf{y}$ sollte in etwa den gleichen Aufwand wie $\mathbf{A}\mathbf{y}$ erfordern.



2. Lineare Gleichungssysteme II

Wir zerlegen dazu die Matrix **additiv** in $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{R}$:

$$\mathbf{A} = \begin{pmatrix} 0 & & & \\ a_{2,1} & 0 & & \\ \vdots & & \ddots & \\ a_{N,1} & \dots & a_{N,N-1} & 0 \end{pmatrix} + \begin{pmatrix} a_{1,1} & & & \\ & a_{2,2} & & \\ & & \ddots & \\ & & & a_{N,N} \end{pmatrix} + \begin{pmatrix} 0 & a_{1,2} & \dots & a_{1,N} \\ & \ddots & & \\ & 0 & a_{N-1,N} & \\ & & & 0 \end{pmatrix}$$

Dabei ist \mathbf{D} die Diagonale von \mathbf{A} , und \mathbf{L} und \mathbf{R} sind die Anteile von \mathbf{A} unterhalb bzw. oberhalb der Diagonale. Wir nehmen an, dass $\det(\mathbf{D}) \neq 0$, d. h. \mathbf{D}^{-1} existiert.

Die Matrizen \mathbf{L} und \mathbf{R} sind **nicht** diejenigen aus der LR-Zerlegung!
Nicht jede reguläre Matrix \mathbf{A} erfüllt $\det(\mathbf{D}) \neq 0$. Es gibt allerdings eine Permutation von Zeilen und Spalten von \mathbf{A} , so dass die Diagonale der permutierten Matrix invertierbar ist, vergleiche VL 1.



2. Lineare Gleichungssysteme II

Mit der additiven Zerlegung $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{R}$ können wir zu einem gegebenen Startwert $\mathbf{x}^{(0)} \in \mathbb{R}^N$ zwei wichtige Verfahren definieren:

Verfahren 2.21 (Gesamtschrittverfahren, Jacobi-Verfahren)

Für $\mathbf{B}_{GSV} := \mathbf{D}$ erhalten wir das **Gesamtschrittverfahren (GSV)**

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{D}^{-1}(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}).$$

Verfahren 2.22 (Einzelschrittverfahren, Gauß-Seidel-Verfahren)

Für $\mathbf{B}_{ESV} := \mathbf{L} + \mathbf{D}$ erhalten wir das **Einzelschrittverfahren (ESV)**

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\mathbf{L} + \mathbf{D})^{-1}(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}).$$



2. Lineare Gleichungssysteme II

In der Praxis vermeiden wir die explizite Invertierung von \mathbf{B} durch Lösen eines Hilfs-LGS. Für das Gesamtschrittverfahren ist dies besonders einfach:

Algorithmus 2.23 : GSV-Verfahren

input : $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{R} \in \mathbb{R}^{N \times N}$ mit $\det(\mathbf{A}) \neq 0$ und $\det(\mathbf{D}) \neq 0$,
 $\mathbf{b} \in \mathbb{R}^N$, $\mathbf{x}^{(0)} \in \mathbb{R}^N$

output : $\tilde{\mathbf{x}} \in \mathbb{R}^N$ als Approximation an die Lösung \mathbf{x} von $\mathbf{Ax} = \mathbf{b}$

```
1  $\tilde{\mathbf{x}} = \mathbf{x}^{(0)}$ ;
2 for  $k = 1, 2, \dots$  do
3   berechne  $\mathbf{r} = \mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}$ ;           % i. W. eine MatVec,  $\mathcal{O}(KN)$ 
4   „löse“  $\mathbf{Dy} = \mathbf{r}$ ;                 %  $N$  Divisionen
5   aktualisiere  $\tilde{\mathbf{x}} = \tilde{\mathbf{x}} - \mathbf{y}$ ;    %  $N$  Subtraktionen
6    $k = k + 1$ ;
7 end
8 return  $\tilde{\mathbf{x}}$ ;
```

Der zusätzliche Speicherbedarf beträgt einen Vektor, jeder Schritt ist effizient.



2. Lineare Gleichungssysteme II

Das Einzelschrittverfahren ist nur unwesentlich komplizierter:

Algorithmus 2.24 : ESV-Verfahren

input : $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{R} \in \mathbb{R}^{N \times N}$ mit $\det(\mathbf{A}) \neq 0$ und $\det(\mathbf{D}) \neq 0$,
 $\mathbf{b} \in \mathbb{R}^N$, $\mathbf{x}^{(0)} \in \mathbb{R}^N$

output : $\tilde{\mathbf{x}} \in \mathbb{R}^N$ als Approximation an die Lösung \mathbf{x} von $\mathbf{Ax} = \mathbf{b}$

```
1  $\tilde{\mathbf{x}} = \mathbf{x}^{(0)}$ ;  
2 for  $k = 1, 2, \dots$  do  
3   berechne  $\mathbf{r} = \mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}$ ;           % i.W. eine MatVec,  $\mathcal{O}(KN)$   
4   löse  $(\mathbf{L} + \mathbf{D})\mathbf{y} = \mathbf{r}$ ;        % Vorwärtseinsetzen,  $\mathcal{O}(KN)$   
5   aktualisiere  $\tilde{\mathbf{x}} = \tilde{\mathbf{x}} - \mathbf{y}$ ;    %  $N$  Subtraktionen  
6    $k = k + 1$ ;  
7 end  
8 return  $\tilde{\mathbf{x}}$ ;
```

Der zusätzliche Speicherbedarf beträgt einen Vektor, jeder Schritt ist effizient.



2. Lineare Gleichungssysteme II

Zur Verdeutlichung des Unterschieds beider Verfahren betrachten wir die komponentenweise Formulierung des LGS:

$$\mathbf{Ax} = \mathbf{b} \quad \Leftrightarrow \quad \sum_{n=1}^N a_{mn}x_n = b_m \quad m = 1, \dots, N$$

Mit unserer Annahme $a_{mm} \neq 0$, d. h. falls die Diagonalelemente nicht verschwinden, können wir die m -te Gleichung nach der Unbekannten x_m auflösen:

$$x_m = \frac{1}{a_{mm}} \left(b_m - \sum_{n < m} a_{mn}x_n - \sum_{n > m} a_{mn}x_n \right) \quad m = 1, \dots, N$$

Das sollte man in der Nachbereitung genau nachvollziehen!



2. Lineare Gleichungssysteme II

$$x_m = \frac{1}{a_{mm}} \left(b_m - \sum_{n < m} a_{mn}x_n - \sum_{n > m} a_{mn}x_n \right) \quad m = 1, \dots, N$$

Beim Gesamtschrittverfahren (Jacobi-Verfahren) verwenden wir auf der rechten Seite nur alte Komponenten der Iterierten:

$$\textcolor{red}{x}_m^{(k+1)} = \frac{1}{a_{mm}} \left(b_m - \sum_{n < m} a_{mn}\textcolor{red}{x}_n^{(k)} - \sum_{n > m} a_{mn}\textcolor{red}{x}_n^{(k)} \right) \quad m = 1, \dots, N,$$

In Matrix-Schreibweise lautet diese Iteration:

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \mathbf{D}^{-1} \left(\mathbf{b} - (\mathbf{A} - \mathbf{D})\mathbf{x}^{(k)} \right) \\ &= \mathbf{D}^{-1}\mathbf{b} - \mathbf{D}^{-1}(\mathbf{A} - \mathbf{D})\mathbf{x}^{(k)} \\ &= \mathbf{D}^{-1}\mathbf{b} - \mathbf{D}^{-1}\mathbf{A}\mathbf{x}^{(k)} + \mathbf{x}^{(k)} \\ &= \mathbf{x}^{(k)} - \mathbf{D}^{-1}(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}) \end{aligned}$$

Das ist gerade die eben definisierte Iterationsvorschrift.



2. Lineare Gleichungssysteme II

$$x_m = \frac{1}{a_{mm}} \left(b_m - \sum_{n < m} a_{mn}x_n - \sum_{n > m} a_{mn}x_n \right) \quad m = 1, \dots, N$$

Analog erhalten wir beim Einzelschrittverfahren (Gauß-Seidel Verfahren) durch Verwendung aller bereits aktualisierter Lösungskomponenten die komponentenweise Iterationsvorschrift

$$x_m^{(k+1)} = \frac{1}{a_{mm}} \left(b_m - \sum_{n < m} a_{mn}x_n^{(k+1)} - \sum_{n > m} a_{mn}x_n^{(k)} \right) \quad m = 1, \dots, N$$

und nach ähnlicher Rechnung wie beim GSV die Matrix-Formulierung der Iterationsvorschrift aus obiger Definition. Das ist eine schöne Übungsaufgabe.

Durch die Verwendung aller bereits aktualisierter Komponenten erwarten wir schnellere Konvergenz.



2. Lineare Gleichungssysteme II

Beispiel 2.25 (Konvergenz der GSV- und ESV-Verfahren)

Wir betrachten für $N = 10$ Beispiel 2.6, d. h.

$$\mathbf{A} = \text{tridiag}(-1, 2, -1), \quad \mathbf{b} = (1, 0, \dots, 0, 1)^T \quad \Rightarrow \quad \mathbf{x} = (1, \dots, 1)^T.$$

Das Gesamtschrittverfahren konvergiert hier, allerdings sehr langsam:

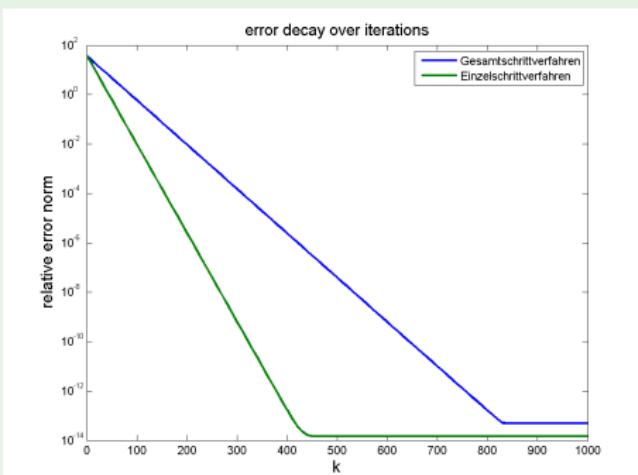
k	0	1	10	100	1000
$\mathbf{x}^{(k)}$	0	0.5	0.7549	0.9943	1
	0	0	0.5508	0.9891	1
	0	0	0.3555	0.9847	1
	0	0	0.2480	0.9816	1
	0	0	0.1748	0.9800	1
	0	0	0.1748	0.9800	1
	0	0	0.2480	0.9816	1
	0	0	0.3555	0.9847	1
	0	0	0.5508	0.9891	1
	0	0.5	0.7549	0.9943	1
$\ \mathbf{x} - \mathbf{x}^{(k)}\ _2$	3.16	2.92	1.96	4.75e-02	2.67e-15



2. Lineare Gleichungssysteme II

Beispiel 2.25 (Konvergenz der GSV- und ESV-Verfahren cont.)

Der Vergleich beider Verfahren ergibt für den relativen Fehler:



Für dieses Beispiel konvergieren beide Verfahren monoton bis zur Maschinengenauigkeit. Das ESV konvergiert wie erwartet schneller als das Jacobi-Verfahren (GSV). Die Richardson-Iteration konvergiert hier nicht.



2. Lineare Gleichungssysteme II

Beispiel 2.25 (Konvergenz der GSV- und ESV-Verfahren cont.)

Für das gleiche Modellproblem erhalten wir in Abhängigkeit von N :

Relativer Fehler des Gesamtschrittverfahrens

k	$N = 50$	$N = 100$	$N = 150$	$N = 200$
100	$8.3 \cdot 10^{-1}$	$9.5 \cdot 10^{-1}$	$9.8 \cdot 10^{-1}$	$9.9 \cdot 10^{-1}$
1000	$1.5 \cdot 10^{-1}$	$6.2 \cdot 10^{-1}$	$8.0 \cdot 10^{-1}$	$8.8 \cdot 10^{-1}$
10000	$5.7 \cdot 10^{-9}$	$7.9 \cdot 10^{-3}$	$1.1 \cdot 10^{-1}$	$2.9 \cdot 10^{-1}$
50000	$2.4 \cdot 10^{-14}$	$3.1 \cdot 10^{-11}$	$2.0 \cdot 10^{-5}$	$2.2 \cdot 10^{-3}$

Relativer Fehler des Einzelschrittverfahrens

k	$N = 50$	$N = 100$	$N = 150$	$N = 200$
100	$6.8 \cdot 10^{-1}$	$9.1 \cdot 10^{-1}$	$9.6 \cdot 10^{-1}$	$9.8 \cdot 10^{-1}$
1000	$2.2 \cdot 10^{-2}$	$3.8 \cdot 10^{-1}$	$6.5 \cdot 10^{-1}$	$7.8 \cdot 10^{-1}$
10000	$6.0 \cdot 10^{-15}$	$6.3 \cdot 10^{-5}$	$1.3 \cdot 10^{-2}$	$8.7 \cdot 10^{-2}$
50000	$6.0 \cdot 10^{-15}$	$3.0 \cdot 10^{-15}$	$4.0 \cdot 10^{-10}$	$5.0 \cdot 10^{-6}$



Konvergenzanalyse



2. Lineare Gleichungssysteme II

Auch wenn das Verfahren „anwendbar“ ist, muss es nicht unbedingt eine Folge von verbessernden Lösungsapproximationen ergeben:

Beispiel 2.26 (Divergenz des Gesamtschrittverfahrens)

Wir betrachten das LGS

$$\mathbf{A} = \begin{pmatrix} 1 & 10 \\ 10 & 1 \end{pmatrix}, \quad \det(\mathbf{A}) = -99 \neq 0, \quad \mathbf{b} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \left(\Rightarrow \mathbf{x} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right).$$

Die ersten Iterierten des Gesamtschrittverfahrens sind:

$$\mathbf{x}^{(0)} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{x}^{(1)} = \begin{pmatrix} 0 \\ -10 \end{pmatrix}, \quad \mathbf{x}^{(2)} = \begin{pmatrix} 100 \\ 0 \end{pmatrix}, \quad \mathbf{x}^{(3)} = \begin{pmatrix} 0 \\ -1000 \end{pmatrix}, \dots$$

Die Iteration divergiert, obwohl $\det(\mathbf{D}) \neq 0$.

Offene Fragen: Wann konvergieren für $k \rightarrow \infty$ die Iterierten $\mathbf{x}^{(k)} \rightarrow \mathbf{x}$ für die klassische Iteration (Definition 2.19) und alle Derivate wie Richardson, ESV und GSV? Gibt es dazu allgemeine Kriterien, um dies vorab zu beurteilen? Können wir die Konvergenzgeschwindigkeit sogar quantifizieren?



2. Lineare Gleichungssysteme II

Um dies zu beantworten, gehen wir zurück zur allgemeinen Iteration

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{B}^{-1}(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}), \quad (2.2)$$

verschieben den Iterationsindex k auf $k - 1$, und stellen den **Fehler** $\mathbf{x}^{(k)} - \mathbf{x}$ zur exakten Lösung \mathbf{x} mit Hilfe der Iterationsvorschrift dar:

$$\begin{aligned}\mathbf{x}^{(k)} - \mathbf{x} &= \mathbf{x}^{(k-1)} - \mathbf{B}^{-1}(\mathbf{A}\mathbf{x}^{(k-1)} - \mathbf{b}) - \mathbf{x} \\ &= \mathbf{x}^{(k-1)} - \mathbf{x} - \mathbf{B}^{-1}\mathbf{A}\mathbf{x}^{(k-1)} - \mathbf{B}^{-1}\mathbf{b} \\ &= \mathbf{x}^{(k-1)} - \mathbf{x} - \mathbf{B}^{-1}\mathbf{A}\mathbf{x}^{(k-1)} - \mathbf{B}^{-1}\mathbf{A}\mathbf{x} \\ &= (\mathbf{x}^{(k-1)} - \mathbf{x}) - \mathbf{B}^{-1}\mathbf{A}(\mathbf{x}^{(k-1)} - \mathbf{x}) \\ &= (\mathbf{I} - \mathbf{B}^{-1}\mathbf{A})(\mathbf{x}^{(k-1)} - \mathbf{x})\end{aligned}$$

Hierbei haben wir wieder $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ genutzt.



2. Lineare Gleichungssysteme II

Ausgehend von der Fehlerdarstellung

$$\mathbf{x}^{(k)} - \mathbf{x} = (\mathbf{I} - \mathbf{B}^{-1}\mathbf{A})(\mathbf{x}^{(k-1)} - \mathbf{x})$$

definieren wir die **Iterationsmatrix** $\mathbf{T} := \mathbf{I} - \mathbf{B}^{-1}\mathbf{A}$, und erhalten sukzessive:

$$\mathbf{x}^{(k)} - \mathbf{x} = \mathbf{T}(\mathbf{x}^{(k-1)} - \mathbf{x}) = \mathbf{T}^2(\mathbf{x}^{(k-2)} - \mathbf{x}) = \cdots = \mathbf{T}^k(\mathbf{x}^{(0)} - \mathbf{x})$$

Dabei ist $\mathbf{T}^k = \mathbf{T} \cdots \mathbf{T}$ (k -mal).

Die Konvergenz $\mathbf{x}^{(k)} \rightarrow \mathbf{x}$ hat also etwas mit dem Verhalten von \mathbf{T}^k für $k \rightarrow \infty$ zu tun. Dies präzisieren wir im folgenden Satz.



2. Lineare Gleichungssysteme II

Satz 2.27 (Konvergenz der klassischen Iteration)

Wenn für die Iterationsmatrix $\mathbf{T} = \mathbf{I} - \mathbf{B}^{-1}\mathbf{A}$ des allgemeinen Verfahrens (2.2)

$$\|\mathbf{T}\|_2 = \|\mathbf{I} - \mathbf{B}^{-1}\mathbf{A}\|_2 < 1$$

gilt, dann konvergiert das Verfahren für alle Startwerte $\mathbf{x}^{(0)}$ gegen die exakte Lösung $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, d. h. $\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}$. Je kleiner $\|\mathbf{T}\|_2$, desto schneller konvergiert das Verfahren.

Beweis: Aus der Submultiplikativität folgt $\|\mathbf{T}^k\|_2 \leq \|\mathbf{T}\|_2^k$, und deshalb für den Fehler mit erneuter Ausnutzung der Submultiplikativität

$$\|\mathbf{x}^{(k)} - \mathbf{x}\|_2 = \|\mathbf{T}^k(\mathbf{x}^{(0)} - \mathbf{x})\|_2 \leq \|\mathbf{T}^k\|_2 \|\mathbf{x}^{(0)} - \mathbf{x}\|_2 \leq \|\mathbf{T}\|_2^k \|\mathbf{x}^{(0)} - \mathbf{x}\|_2.$$

Wegen $\|\mathbf{T}\|_2 < 1$ folgt die Behauptung für alle Startwerte $\mathbf{x}^{(0)}$. □



2. Lineare Gleichungssysteme II

Bemerkungen:

- Die Konvergenzaussage des vorherigen Satzes gilt für den Fehler, und stellt somit eine theoretische Rechtfertigung des Verfahrens dar.
- Man kann auch die Rückrichtung zeigen: Die notwendige Bedingung $\|\mathbf{T}\|_2 < 1$ ist auch hinreichend. Insbesondere sind die Verfahren divergent, wenn einige Eigenwerte im Betrag kleiner und einige größer Eins sind.
- In der Praxis ist die Norm des Residuums eine bessere Größe um die Konvergenz zu beurteilen → gleich. Hier lässt sich mit etwas mehr Aufwand eine ähnliche Aussage zeigen.
- Wir erinnern uns an Definition 2.12 (Spektralnorm): Um die Konvergenz der abgeleiteten Verfahren wie GSV und ESV für eine gegebene Koeffizientenmatrix eines LGS $\mathbf{A}\mathbf{x} = \mathbf{b}$ beurteilen zu können, müssen wir also zuerst die Iterationsmatrix aufstellen und dann ihren Spektralradius bestimmen. Dazu betrachten wir nun einige Beispiele.



2. Lineare Gleichungssysteme II

Beispiel 2.28 (Beispiel 2.26 cont.)

Für die Koeffizientenmatrix $\mathbf{A} = \begin{pmatrix} 1 & 10 \\ 10 & 1 \end{pmatrix}$ lautet die GSV-Iterationsmatrix:

$$\begin{aligned}\mathbf{T}_{\text{GSV}} &= \mathbf{I} - \mathbf{D}^{-1} \mathbf{A} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 10 \\ 10 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 10 \\ 10 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & -10 \\ -10 & 0 \end{pmatrix}\end{aligned}$$

Kurze Rechnung liefert die Eigenwerte ± 10 , also konvergiert das GSV-Verfahren **nicht**.

Verständnisübungen: (1) Durchführen der Rechnung bei der Nachbereitung; (2) Konvergenz des Einzelschrittverfahrens?



2. Lineare Gleichungssysteme II

Beispiel 2.29 (Konvergenz des GSV)

Für $N = 3$ betrachten wir unser Standardbeispiel, d. h. die Tridiagonalmatrix aus Beispiel 2.6. Wir erhalten nach kurzer Rechnung

$$\mathbf{A} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \quad \Rightarrow \quad \mathbf{T}_{\text{GSV}} = \mathbf{I} - \mathbf{D}^{-1} \mathbf{A} = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 \end{pmatrix}$$

und mit HM123 die Eigenwerte $\lambda_{1,2} = \pm \frac{\sqrt{2}}{2} < \pm 1$, $\lambda_3 = 0$ der Iterationsmatrix. Das GSV ist für diese Beispielmatrix also konvergent für beliebige Startwerte und rechte Seiten.

Verständnisübung: Konvergenz des Einzelschrittverfahrens?



2. Lineare Gleichungssysteme II

Beispiel 2.30 (Konvergenz des ESV)

Allgemein lautet die Iterationsmatrix des Einzelschrittverfahrens:

$$\boldsymbol{T}_{\text{ESV}} = \boldsymbol{I} - (\boldsymbol{L} + \boldsymbol{D})^{-1} \boldsymbol{A} = -(\boldsymbol{L} + \boldsymbol{D})^{-1} \boldsymbol{R}$$

Als Beispiel betrachten wir

$$\boldsymbol{A} = \begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{pmatrix}$$

und somit nach einer kurzen Rechnung:

$$\boldsymbol{T}_{\text{ESV}} = \boldsymbol{I} - (\boldsymbol{D} + \boldsymbol{L})^{-1} \boldsymbol{A} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ \frac{1}{2} & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{pmatrix} = \begin{pmatrix} 0 & -\frac{1}{2} \\ 0 & \frac{1}{4} \end{pmatrix}$$

Wir berechnen wegen der fehlenden Symmetrie $\boldsymbol{T}_{\text{ESV}}^T \boldsymbol{T}_{\text{ESV}}$, und finden
 $\|\boldsymbol{T}_{\text{ESV}}^T \boldsymbol{T}_{\text{ESV}}\|_2 = \max \sqrt{|\lambda|} = \frac{\sqrt{5}}{4}$. Also konvergiert das ESV für dieses Beispiel mit jedem Anfangswert.



2. Lineare Gleichungssysteme II

Mit dem Spektralradius kann also die Konvergenz für jede gegebene Matrix a priori qualitativ und quantitativ untersucht werden. Dies ist allerdings recht aufwändig und mühsam, weil alle Eigenwerte bestimmt werden müssen.

In der Praxis sind daher einfacher zu überprüfende Kriterien wichtig.

Satz 2.31 (Konvergenzkriterien für GSV und ESV)

Die Gesamt- und Einzelschrittverfahren sind konvergent, wenn eine der folgenden Bedingungen erfüllt ist:

- (1) \mathbf{A} oder \mathbf{A}^T ist stark diagonaldominant ist, d. h. $|a_{mm}| > \sum_{n \neq m} |a_{mn}|$ für alle $m = 1, \dots, N$.
- (2) \mathbf{A} oder \mathbf{A}^T ist schwach diagonaldominant (d. h. \geq in obiger Eigenschaft) und irreduzibel ist, d. h. es existiert keine Folge von Zeilen- und Spaltentausch-Operationen, so dass \mathbf{A} in die Form $\begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}$ überführt werden kann.

Die zweite Bedingung wird von unserer Tridiagonal-Beispielmatrix erfüllt.



Gedämpfte Verfahren



2. Lineare Gleichungssysteme II

ESV und GSV konvergieren i.A. langsam. Gedämpfte (relaxierte) Verfahren verbessern durch einen frei zu wählenden **Dämpfungsparameter** ω das Spektrum der Iterationsmatrix. Als Beispiel betrachten wir die gedämpfte Variante des ESV:

Verfahren 2.32 (SOR-Verfahren, Successive OverRelaxation)

Mit $B_{SOR(\omega)} = (\mathbf{L} + \omega^{-1}\mathbf{D})$ erhalten wir das gedämpfte Einzelschrittverfahren

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\mathbf{L} + \omega^{-1}\mathbf{D})^{-1}(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}).$$

Für $\omega = 1$ erhalten wir das (ungedämpfte) Einzelschrittverfahren, das ist klar. Epische Rechnung ergibt die Iterationsmatrix des SOR-Verfahrens:

$$T_{SOR(\omega)} = (\omega\mathbf{L} + \mathbf{D})^{-1}((1 - \omega)\mathbf{D} - \omega\mathbf{R})$$

Hiermit kann nun eine Eigenwert-Analyse durchgeführt werden. Es gelten für $0 < \omega < 2$ weiterhin die einfacher zu überprüfenden Konvergenzkriterien.



2. Lineare Gleichungssysteme II

Beispiel 2.33 (Vergleich ESV, GSV und SOR-Verfahren)

Wir betrachten wieder das Standardbeispiel 2.6 dieser Vorlesung, d. h. das Tridiagonalmatrix-Problem mit bekannter exakter Lösung.

Der experimentell bestimmte optimale Dämpfungsparameter lautet $\omega_{opt} = 1.5604$ für das SOR-Verfahren. Für die relativen Fehler erhalten wir die Ergebnisse der folgenden Tabelle.

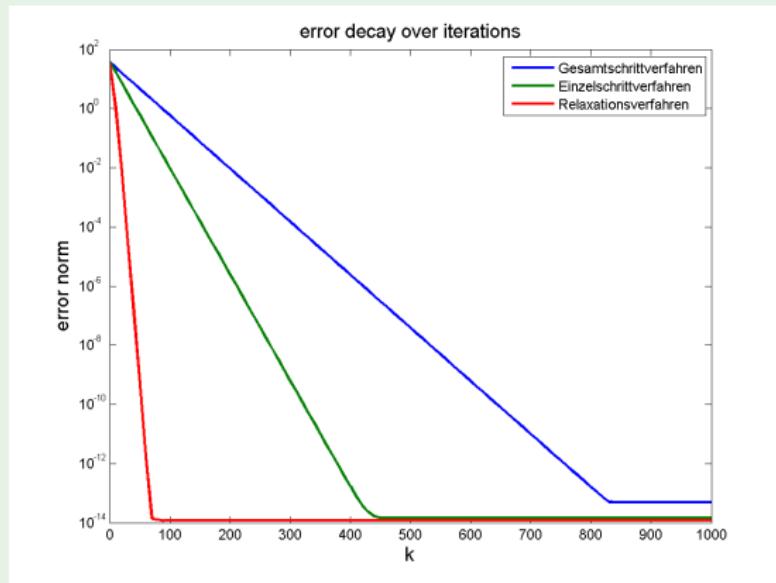
k	0	1	10	100	1000
$\ \mathbf{x} - \mathbf{x}_{GSV}^{(k)}\ _2 / \ \mathbf{x}\ _2$	1.00	9.59e-01	6.66e-01	1.60e-02	1.33e-15
$\ \mathbf{x} - \mathbf{x}_{ESV}^{(k)}\ _2 / \ \mathbf{x}\ _2$	1.00	9.24e-01	4.49e-01	2.64e-04	4.01e-15
$\ \mathbf{x} - \mathbf{x}_{SOR}^{(k)}\ _2 / \ \mathbf{x}\ _2$	1.00	8.02e-01	2.76e-02	3.18e-16	3.18e-16



2. Lineare Gleichungssysteme II

Beispiel 2.33 (Vergleich ESV, GSV und SOR-Verfahren cont.)

Wir verdeutlichen die Ergebnisse der Tabelle zusätzlich grafisch:



Der Vorteil der Dämpfung ist offensichtlich.



Praktische Aspekte



2. Lineare Gleichungssysteme II

Die effiziente Lösung der Hilfssysteme bei den betrachteten Verfahren (vgl. Algorithmus 2.23 und 2.24) ist für die Umsetzung in der Praxis obligatorisch, zugeschnitten auf das jeweilige Speicherformat für dünn besetzte Koeffizientenmatrizen.

Die Konvergenztheorie basiert auf der Betrachtung des Fehlers gegen die exakte Lösung, den wir natürlich nicht kennen. Es fehlt also noch ein **Abbruchkriterium**, d. h. eine berechenbare Entscheidung, wann eine Näherungslösung „gut genug“ ist.

Beispiele für praktikable Abbruchkriterien

- ① Cauchy-Kriterium: $\|\mathbf{y}\| = \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \leq \text{TOL}$
- ② Residual-Kriterium: $\|\mathbf{r}\| = \|\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}\| \leq \text{TOL}$
- ③ Beschränkung der maximalen Anzahl von Iterationen: $k \leq k_{\max}$



2. Lineare Gleichungssysteme II

Algorithmisch lassen sich die Abbruchkriterien folgendermaßen in ein iteratives Verfahren einbauen:

Algorithmus 2.34 : Iteratives Verfahren mit Residual- und Iterationskriterium

input : $\mathbf{A} \in \mathbb{R}^{N \times N}$ mit $\det(\mathbf{A}) \neq 0$, $\mathbf{b} \in \mathbb{R}^N$, $\mathbf{B} \in \mathbb{R}^{N \times N}$ mit $\det(\mathbf{B}) \neq 0$,
 $\mathbf{x}^{(0)} \in \mathbb{R}^N$, $\text{TOL} > 0$ und $k_{\max} > 0$

output : $\tilde{\mathbf{x}} \in \mathbb{R}^N$ als Approximation an die Lösung \mathbf{x} von $\mathbf{Ax} = \mathbf{b}$

- 1 $k = 0$, $\tilde{\mathbf{x}} = \mathbf{x}^{(0)}$, $\mathbf{r} = \mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}$;
 - 2 **while** $k \leq k_{\max}$ **and** $\|\mathbf{r}\|_2 > \text{TOL}$ **do**
 - 3 löse $\mathbf{By} = \mathbf{r}$;
 - 4 $\tilde{\mathbf{x}} = \tilde{\mathbf{x}} - \mathbf{y}$;
 - 5 $k = k + 1$;
 - 6 $\mathbf{r} = \mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}$;
 - 7 **end**
 - 8 **return** $\tilde{\mathbf{x}}$;
-



Zusammenfassung I

- Die Konditionszahl der Koeffizientenmatrix \mathbf{A} eines LGS $\mathbf{Ax} = \mathbf{b}$ bestimmt die unvermeidbare Fehlerverstärkung bei der numerischen Lösung.
- Zur Berechnung der Konditionszahl benötigen wir (Matrix-) Normen.
- Die von der Betragssummen- und Maximumsnorm induzierten Spalten- und Zeilensummennormen sind einfach berechenbar.
- Die Spektralnorm basiert auf maximalen Eigenwerten.
- Der Störungssatz quantifiziert, mit welcher Fehlerverstärkung wir bei Störungen der rechten Seite des LGS rechnen müssen. Dieser Einfluss ist a priori berechenbar.



Zusammenfassung II

- Iterative Verfahren sind Pflicht bei dünn besetzten LGS. Konvergente iterative Verfahren erlauben die Verbesserung der Güte der Lösungsapproximation durch Erhöhen der Iterationszahl.
- Beispiele konkreter Verfahren sind GSV, ESV und SOR, wobei letzteres einen freien Tuningparameter enthält.
- Die Konvergenz der Verfahren kann für eine gegebene Koeffizientenmatrix **A** a priori anhand des Spektrums der zugehörigen Iterationsmatrix beurteilt werden.
- In der Praxis sind speziellere Kriterien wie die Diagonaldominanz einfacher zu überprüfen.



Hausaufgaben

- Verinnerlichen Sie das Konzept einer Norm und einer Matrixnorm.
- Bearbeiten Sie für ein minimales Verständnis die folgenden Beispielaufgaben, die auf die Vortragsübungen vorbereiten. Besuchen Sie bei Problemen die Tutorien.
- Durchdenken Sie für ein vertieftes Verständnis die jeweils kleingedruckten Hinweise zu Verständnisübungen, und bemühen Sie sich, die ViPLab-Beispiele nachzuvollziehen.
- Vollziehen Sie für ein optimales Verständnis die Beweise und Umformungen dieser VL-Einheit nach.
- **BESUCHEN SIE DIE NÄCHSTE VORTRAGSÜBUNG!**



Beispieldaufgaben



Matrixnormen

Berechnen Sie die Zeilensummen-, Spaltensummen- und Spektralnormen von

$$\mathbf{A} = \frac{1}{5} \begin{pmatrix} 3 & -4 & 0 \\ -4 & -3 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$



2. Lineare Gleichungssysteme II

Lösungshinweise: Das ist eine reine Rechenaufgabe, es muss lediglich Satz 2.10 für die Zeilen- und Spaltensummennorm, bzw. Satz 2.12 für die Spektralnorm angewendet werden. Letzeres benötigt HM123 zur Bestimmung des betragsmäßig größten Eigenwerts, weil die Matrix symmetrisch ist.

Ergebnis:

$$\|\mathbf{A}\|_{\infty} = 1.4$$

$$\|\mathbf{A}\|_1 = 1.4$$

$$\|\mathbf{A}\|_2 = 1.0$$



Störungstheorie

Gegeben sei die Matrix

$$\mathbf{A} = \begin{pmatrix} 4 & 1 \\ 3 & -1 \end{pmatrix}$$

- ① Welchen relativen Fehler muss man in der Lösung \mathbf{x} eines Gleichungssystems $\mathbf{Ax} = \mathbf{b}$ maximal erwarten, wenn der relative Fehler in den Daten (der rechten Seite \mathbf{b}) 5 % beträgt?
- ② Wie groß darf der relative Fehler in den Daten (der rechten Seite \mathbf{b}) höchstens sein, damit der relative Fehler in der Lösung kleiner 10 % ist?



2. Lineare Gleichungssysteme II

Lösungshinweise: Dies ist ein Beispiel für eine schlecht gestellte Aufgabe. Zur Anwendung von Satz 2.14 dürfen wir uns also eine Norm aussuchen. Die Musterlösung basiert auf $\kappa_2(\mathbf{A})$ und damit der Spektralnorm als Matrixnorm, die zur Euklid-Vektornorm gehört. In der Zeilen- bzw. Spaltensummennorm sind die Rechnungen deutlich einfacher.

Ergebnisse: (in der Klausur sind die Zahlen netter)

- Bestimmen der Konditionszahl von \mathbf{A} :

$$\kappa_2(\mathbf{A}) = \sqrt{\frac{\frac{27 + \sqrt{533}}{2}}{\frac{27 - \sqrt{533}}{2}}} \approx 3.5776$$

- Einsetzen in den Störungssatz und Umformen:

$$① \frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2} \leq 17.89\%$$

$$② \frac{\|\Delta \mathbf{b}\|_2}{\|\mathbf{b}\|_2} \leq 2,8\%$$



2. Lineare Gleichungssysteme II

GSV und ESV

Es soll das Gleichungssystem $\mathbf{A}\mathbf{x} = \mathbf{b}$ mit

$$\mathbf{A} = \begin{pmatrix} 1 & -0.9 \\ -5 & -10 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} -5.5 \\ -45 \end{pmatrix}, \quad \mathbf{x}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

mit einem iterativen Verfahren gelöst werden. Dazu ist der Startvektor $\mathbf{x}^{(0)}$ wie oben gegeben. Wie viele Iterationen benötigt das Jacobi-Verfahren (GSV-Verfahren), bis der Fehler gegen die exakte Lösung $\mathbf{x}^* = (-1, 5)^\top$ echt kleiner 10^{-6} ist?



2. Lineare Gleichungssysteme II

Lösungshinweise: Der Schlüssel zu dieser Aufgabe ist Satz 2.27. Wenn wir die gegebenen Zahlwerte in den Satz einsetzen, bleibt eine Formel übrig, die wir mit Hilfe des Logarithmus auflösen können.

Ergebnisse: (in der Klausur sind die Zahlen netter)

- Einsetzen liefert $\|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2 \leq \|\mathbf{T}\|_2^k \|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2$ mit $\|\mathbf{T}\|_2 = 0.9$
- Auflösen nach k (Logarithmus!) liefert $k \geq 147$
- Tipp: Kann mit ViPLab überprüft werden (siehe VÜ und ViPÜ); man sieht, dass die gewünschte Fehlerschranke bereits früher erreicht wird



3. Eigenwertprobleme

3. Eigenwertprobleme

THE \$25.000.000.000⁰ EIGENVECTOR THE LINEAR ALGEBRA BEHIND GOOGLE

KRISTIN BRYAN AND TATIANA LEISE

Abstract. Google's success is based on a large part on its PageRank algorithm, which ranks the importance of webpages according to an eigenvector of a large link matrix. Analysis of the PageRank formula provides a mathematical explanation of how Google's search engine works. This paper is intended for students who have completed one or two courses presenting the material with enough background from the review. The authors have included a few exercises at the end of each section to encourage students to work with the material. Some of the exercises require knowledge of linear algebra, while others require knowledge of numerical analysis. <http://www.concordia.edu/~klye/>

Key words: linear algebra, PageRank, eigenvalues, stochastic matrices

AMS subject classifications: 15A12, 15A51, 68M10

1. Introduction. When Google went online in the late 1990's, one thing that set it apart from other search engines was that its search result listings always seemed to deliver the "good stuff" up front. In other words, Google's search results were more likely to be relevant to the search query than to irrelevant web pages that just happened to match the search text. Part of the magic behind Google is its PageRank algorithm, which quantitatively rates the importance of each page on the web, so that the most important pages are the ones that appear at the top of the search results (and typically most relevant and helpful) page list.

Understanding how to calculate PageRank is essential for anyone designing a web page that they want to be found by Google. If a page is listed first in a Google search, it's likely people looking at your page. Indeed, due to Google's prevalence as a search engine, its ranking system has become the standard for determining the relevance of web pages, and thus which kinds of information and services get accessed most frequently. One goal in this paper is to explain one of the core ideas behind how Google calculates web page rankings. This turns out to be a delightful application of linear algebra.

Search engines such as Google have to do three basic things:

ODER: The \$25.000.000.000 eigenvector: the linear algebra behind google

K. Bryan & T. Leise, SIAM Review 48 (2006)



Motivation

Google Page Rank



3. Eigenwertprobleme

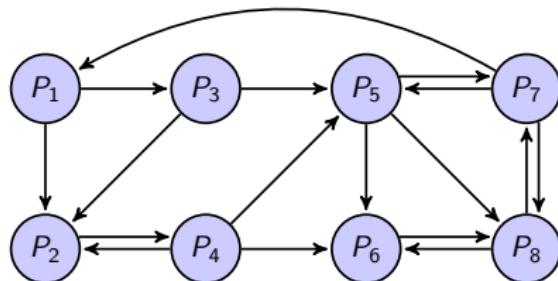
Suchmaschinen im Netz arbeiten (stark vereinfacht) folgendermaßen:

- ① Erstelle eine Datenbank aller öffentlich erreichbarer Seiten.
- ② Indiziere die Seiten in der Datenbank, so dass diese effizient nach Schlüsselwörtern und Phrasen durchsucht werden können.
- ③ Bewerte die Wichtigkeit jeder Seite in der Datenbank, um nach einer Suche die gefundenen Seiten nach Bedeutung geordnet anzuzeigen. Dies ist offenbar der wichtigste Schritt.

Warum war Google Anfang des Jahrtausends so erfolgreich? Schritt 1 und 2 sind Informatik und Informationstheorie. Die Antwort für Schritt 3 lautet plakativ: Brin und Page wussten, was ein **Eigenwert** ist.



3. Eigenwertprobleme



Zur Verdeutlichung betrachten wir ein Netzwerk von N (verlinkten) Webseiten P_1, \dots, P_N . Hierbei sind:

x_n : zu bestimmende Bedeutung der Seite P_n

ℓ_n : Anzahl Links, die von P_n ausgehen

S_n : Menge der Seitennummern der Seiten, die auf P_n verweisen:

$$S_n := \{m \in \{1, \dots, N\} \mid P_m \text{ hat Link auf } P_n\}$$

Im obigen Netzwerk haben wir bspw. für die Seite P_5 : $\ell_5 = 3$, $S_5 = \{3, 4, 7\}$



3. Eigenwertprobleme

Grundidee von Google (vereinfacht)

Die gesuchte Bedeutung x_n der Seite P_n ist die gewichtete Summe der Bedeutungen all der Seiten, die auf P_n verweisen:

$$x_n = \sum_{m \in \mathcal{S}_n} \frac{1}{\ell_m} x_m \quad \text{für } n = 1, \dots, N$$

Das ist ein Silicon Valley Argument: je größer je besser.

Klar ist:

- N ist gigantisch groß, selbst wenn nur die Webseiten der Uni Stuttgart durchsucht werden sollen.
- $|\mathcal{S}_n|$ kann absolut groß sein, ist aber relativ zu N meist klein.



3. Eigenwertprobleme

Die Beobachtung von Google war nun, leicht vereinfacht:

Beispiel 3.1 (PageRank Eigenwertproblem)

Die sogenannte **PageRank Matrix** $\mathbf{A} \in \mathbb{R}^{N \times N}$ besitzt die Einträge

$$a_{nm} = \begin{cases} \ell_m^{-1} & m \in \mathcal{S}_n, \\ 0 & m \notin \mathcal{S}_n. \end{cases}$$

Die gesuchten Bedeutungen $\mathbf{x} = (x_1, \dots, x_N)^\top \in \mathbb{R}^N$ der Seiten P_1, \dots, P_N sind gegeben als Eigenvektor zum Eigenwert 1 der Matrix \mathbf{A} . Wir müssen also ein \mathbf{x} finden mit $\mathbf{Ax} = 1 \cdot \mathbf{x}$, um die Bedeutungen direkt ablesen zu können. Wegen der Überlegungen auf der letzten Folie ist die Matrix typischerweise extrem dünn besetzt.

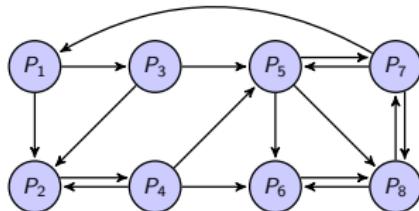
Ziel im Folgenden: Begründung dieses Beispielproblems



3. Eigenwertprobleme

Die PageRank Matrix für unser Beispiel-Internet lautet:

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{3} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 1 & \frac{1}{3} & 0 \end{pmatrix}$$



$$\ell_7 = 3, S_1 = \{7\}$$

$$S_5 = \{3, 4, 7\},$$

$$\ell_3 = 2, \ell_4 = 3, \ell_7 = 3$$

$$\sum = 1 \text{ per Def. von } A \text{ und } S_n$$

Verständnisübung: Nachvollziehen der anderen Einträge.

Bereits für dieses Beispiel ist die PageRank Matrix extrem dünn besetzt.



3. Eigenwertprobleme

Mit den Methoden aus dieser Vorlesungseinheit errechnen wir den Eigenvektor

$$\mathbf{x} \approx \left(\underbrace{0.1400}_{x_1}, 0.1576, 0.0700, 0.1576, 0.2276, 0.4727, 0.4201, \underbrace{0.6886}_{x_N} \right)^T.$$

Dies ergibt die Seitenrangfolge

$$P_8 \rightarrow P_6 \rightarrow P_7 \rightarrow P_5 \rightarrow P_2 \rightarrow P_4 \rightarrow P_1 \rightarrow P_3.$$

Alle Beispiele dieser VL können mit dem Coderahmen der zugehörigen ViPLab-Übungen nachvollzogen werden.

Neben der effizienten Berechnung dieses Eigenvektors interessiert uns in dieser VL:

- ① Wohlgestelltheit: Hat die PageRank Matrix \mathbf{A} immer den EW 1? Funktioniert dieses Verfahren also überhaupt für jedes Netzwerk?
- ② Wenn ja, ist dies der größte EW?



Motivation

Weitere Beispiele aus früheren VLern



3. Eigenwertprobleme

Als zweites Beispiel erinnern wir uns an die Störungstheorie aus VL 2: Für ein LGS $\mathbf{A}\mathbf{x} = \mathbf{b}$ stellt die Konditionszahl $\kappa_2(\mathbf{A}) := \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2$ eine obere Schranke für die Fehlerverstärkung in der Lösung durch Fehler und Änderungen in \mathbf{A} und \mathbf{b} dar.

Beispiel 3.2 (Berechnung der Spektralkonditionszahl)

Zur Berechnung von $\kappa_2(\mathbf{A})$ müssen wir im allgemeinen Fall (die Wurzeln der) betragsmäßig größten und kleinsten Eigenwerte von $\mathbf{A}^T \mathbf{A}$ bestimmen.

Ist \mathbf{A} symmetrisch ergibt sich $\kappa_2(\mathbf{A})$ als Quotient des betragsgrößten und des betragskleinsten Eigenwerts von \mathbf{A} .

Eine berechtigte Frage lautet also: Was haben wir von den Resultaten aus VL 2, falls wir die Eigenwerte nicht kennen? Optimistischer formuliert, wie erhalten wir effizient Näherungen für die benötigten Eigenwerte, da der Zugang über die Nullstellen charakteristischer Probleme höchst uneffizient ist?



3. Eigenwertprobleme

Als drittes Beispiel erinnern wir uns an die Konvergenztheorie einfacher iterativer Verfahren aus VL 2 zur Lösung eines LGS $\mathbf{A}\mathbf{x} = \mathbf{b}$, d. h. an Verfahren der Bauart

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{B}^{-1}(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b})$$

mit einer regulären Matrix \mathbf{B} wie in den Gesamtschritt-, Einzelschritt- und SOR-Verfahren.

Beispiel 3.3 (Konvergenz einfacher iterativer LGS-Löser)

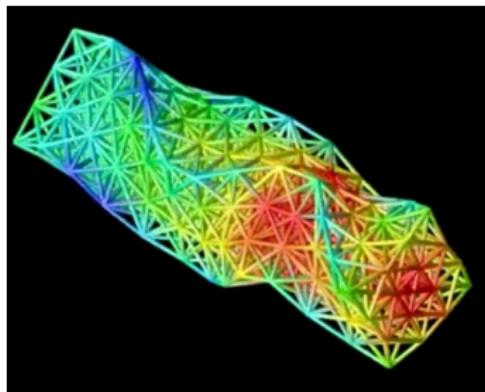
Um die Konvergenz eines iterativen Verfahrens zu beurteilen, müssen wir die Spektralnorm der Iterationsmatrix $\mathbf{I} - \mathbf{B}^{-1}\mathbf{A}$ bestimmen, d. h. im symmetrischen Fall ihren betragsmäßig größten Eigenwert und sonst (die Wurzel des) betragsmäßig größten Eigenwerts von $(\mathbf{I} - \mathbf{B}^{-1}\mathbf{A})^T(\mathbf{I} - \mathbf{B}^{-1}\mathbf{A})$.

Die Herausforderungen sind offenbar dieselben wie im vorherigen Beispiel.

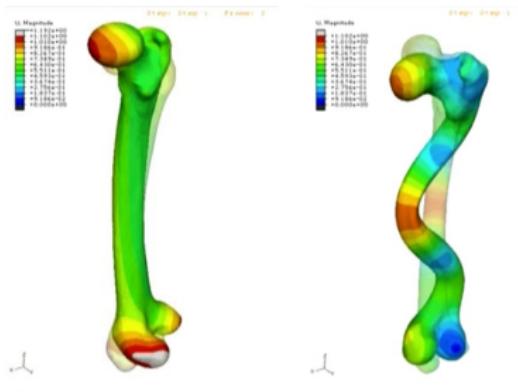


3. Eigenwertprobleme

Aus VL 0 stammt das vierte Beispiel: Simulation der Schwingung eines Tragwerks oder der Deformation eines Knochens:



www.youtube.com/watch?v=R9EWUI1IMFw



www.youtube.com/watch?v=gZAutaeAf7s

Die Deformationsfelder setzen sich aus Eigenvektoren der Systemmatrix zu den Eigenwerten zusammen, welche der (Eigen-) Schwingungs-Frequenz entsprechen. Technisch relevant sind sogenannte Eigenschwingungen, vgl. Ihre Studiengänge. Ein drastisches Beispiel ist die Tacoma-Narrows Brücke aus VL 0.



Wiederholung: Eigenwerte und Eigenvektoren



3. Eigenwertprobleme

Definition 3.4 (Eigenwert und Eigenvektor)

Sei $\mathbf{A} \in \mathbb{R}^{N \times N}$. $\lambda \in \mathbb{C}$ heißt **Eigenwert (EW)** von \mathbf{A} , falls ein $\mathbf{x} \in \mathbb{C}^N \setminus \{\mathbf{0}\}$ existiert mit $\mathbf{Ax} = \lambda\mathbf{x}$. Der Vektor \mathbf{x} heißt **Eigenvektor (EV) zum Eigenwert λ** . Die Menge aller Eigenwerte

$$\text{sp}(\mathbf{A}) := \{\lambda \in \mathbb{C} \mid \lambda \text{ ist Eigenwert von } \mathbf{A}\}$$

heißt **Spektrum** von \mathbf{A} . Der Wert

$$\rho(\mathbf{A}) := \max\{ |\lambda| \mid \lambda \in \text{sp}(\mathbf{A})\}$$

ist der **Spektralradius** von \mathbf{A} .

Falls $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_{N-1}| \geq |\lambda_N|$, so nennen wir λ_1 bzw. λ_N den betragsgrößten bzw. betragskleinsten Eigenwert.



3. Eigenwertprobleme

Wir präzisieren einen HM12-Begriff, den wir in VL 2 ad-hoc verwendet haben:

Satz 3.5 (Charakteristisches Polynom)

Sei $\mathbf{A} \in \mathbb{R}^{N \times N}$. Dann heißt

$$\chi_{\mathbf{A}}(\lambda) := \det(\mathbf{A} - \lambda \mathbf{I})$$

charakteristisches Polynom von \mathbf{A} . $\chi_{\mathbf{A}}$ ist ein Polynom vom Grad N und besitzt die N komplexen Nullstellen $\lambda_1, \dots, \lambda_N \in \mathbb{C}$. Diese Nullstellen sind gerade die Eigenwerte von \mathbf{A} . Der Fundamentalsatz der Algebra besagt, dass $\chi_{\mathbf{A}}$ nur über \mathbb{C} in Linearfaktoren zerfällt.

Wir können also alle Eigenwerte einer Matrix \mathbf{A} bestimmen, indem wir die Nullstellen von $\chi_{\mathbf{A}}$ und damit die Linearfaktoren bestimmen. Dies ist analytisch schwer und rechnerisch sehr aufwändig.



3. Eigenwertprobleme

Wir benötigen für die weitere Diskussion zwei Sätze aus der HM12:

Satz 3.6 (Eigenwerte I)

- ① Jede Matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ besitzt N komplexe Eigenwerte $\lambda_1, \dots, \lambda_N \in \mathbb{C}$, d. h.

$$\text{sp}(\mathbf{A}) = \{\lambda_1, \dots, \lambda_N\} \subset \mathbb{C}.$$

Das ist eine direkte Konsequenz aus Satz 3.5.

- ② Für jede Matrixnorm $\|\cdot\|$ auf $\mathbb{R}^{N \times N}$ gemäß Definition 2.8 gilt

$$\rho(\mathbf{A}) \leq \|\mathbf{A}\|.$$

- ③ Ist \mathbf{A} symmetrisch, so sind die Eigenwerte reell, d. h.

$$\text{sp}(\mathbf{A}) \subset \mathbb{R}.$$

- ④ Ist \mathbf{A} symm. und positiv definit (SPD), so sind alle Eigenwerte positiv, d. h.

$$\text{sp}(\mathbf{A}) \subset \mathbb{R}_{>0}.$$



3. Eigenwertprobleme

Satz 3.7 (Eigenwerte II)

- ① Regularität mittels Eigenwerten: $\det(\mathbf{A}) \neq 0 \Leftrightarrow 0 \notin \text{sp}(\mathbf{A})$
- ② Eigenwerte transponierter Matrizen: $\lambda \in \text{sp}(\mathbf{A}) \Leftrightarrow \lambda \in \text{sp}(\mathbf{A}^T)$
- ③ Eigenwerte der Inversen einer regulären Matrix:
 $\lambda \in \text{sp}(\mathbf{A}) \Leftrightarrow \lambda^{-1} \in \text{sp}(\mathbf{A}^{-1})$ mit gleichen Eigenvektoren
- ④ Sei x EV zu $\lambda \in \text{sp}(\mathbf{A})$ und $\mu \in \mathbb{C}$. Dann ist $\lambda - \mu$ EW von $\mathbf{A} - \mu\mathbf{I}$ mit EV x .

Verständnisübung: Anwendung von Satz 3.6 und 3.7 in VL 2 nachvollziehen.



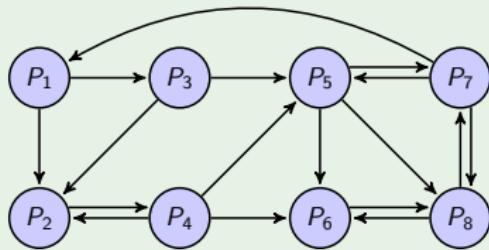
Anwendung auf das PageRank-Problem



3. Eigenwertprobleme

Beispiel 3.1 (PageRank Eigenwertproblem cont.)

Wir betrachten weiterhin die PageRank Matrix für ein Beispielnetzwerk:



$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{3} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 1 & \frac{1}{3} & 0 \end{pmatrix}$$

Wir lesen ab: Die m -te Spalte von \mathbf{A} hat ℓ_m Einträge $\neq 0$ mit den Werten $\frac{1}{\ell_m}$. Diese Spaltensummen können wir mit $\mathbf{x} = (1, \dots, 1)^T$ formalisieren als

$$(\mathbf{A}^T \mathbf{x})_m = \sum_{n=1}^N a_{nm} \underbrace{x_n}_{=1} = \sum_{n=1}^N a_{nm} = 1 = (\mathbf{x})_m.$$

Erinnerung: ℓ_m : Anzahl Links, die von P_m ausgehen, deshalb Spaltensumme 1.



3. Eigenwertprobleme

Damit können wir rechtfertigen, warum die Google-Idee mathematisch sinnvoll ist. Wir gehen in zwei Schritten vor, ausgehend von der vorherigen Folie:

$$(\mathbf{A}^T \mathbf{x})_m = \sum_{n=1}^N a_{nm} \underbrace{x_n}_{=1} = \sum_{n=1}^N a_{nm} = 1 = (\mathbf{x})_m$$

Zeilenweise können wir daran ablesen: 1 ist EW von \mathbf{A}^T , und $\mathbf{x} = (1, \dots, 1)^T$ ist der zugehörige EV:

$$\mathbf{A}^T \mathbf{x} = \mathbf{x} \quad \text{also} \quad 1 \in \text{sp}(\mathbf{A}^T)$$

Erinnerung: Satz 3.7 (Eigenwerte II)

- ② Eigenwerte transponierter Matrizen: $\lambda \in \text{sp}(\mathbf{A}) \iff \lambda \in \text{sp}(\mathbf{A}^T)$ mit gleichen Eigenvektoren.

Mit diesem Resultat erhalten wir, dass auch $1 \in \text{sp}(\mathbf{A})$ gilt: Für die PageRank-Matrix \mathbf{A} ist also in der Tat 1 EW zum EV $\mathbf{x} = (1, \dots, 1)^T$.



3. Eigenwertprobleme

Im zweiten Schritt bestätigen wir, dass die Eins hier der betragsgrößte EW ist.

Erinnerung: Satz 2.10 (Zeilen- und Spaltensummennorm)

Sei $\mathbf{A} \in \mathbb{R}^{N \times N}$. Dann sind die zu $p = 1$ und $p = \infty$ gehörigen Matrixnormen explizit berechenbar als **Spaltensummennorm** und **Zeilensummennorm**:

$$\|\mathbf{A}\|_1 = \max_{n=1,\dots,N} \left(\sum_{m=1}^N |a_{mn}| \right) \quad \|\mathbf{A}\|_\infty = \max_{m=1,\dots,N} \left(\sum_{n=1}^N |a_{mn}| \right)$$

Erinnerung: Satz 3.6 (Eigenwerte I)

- ② Für jede Matrixnorm $\|\cdot\|$ auf $\mathbb{R}^{N \times N}$ gemäß Definition 2.8 gilt

$$\rho(\mathbf{A}) \leq \|\mathbf{A}\| .$$



3. Eigenwertprobleme

Wir gehen wieder von der komponentenweisen Formulierung aus:

$$(\mathbf{A}^T \mathbf{x})_m = \sum_{n=1}^N a_{nm} = 1 = (\mathbf{x})_m$$

Mit der *Spaltensummennorm* können wir das schreiben als

$$1 = \|\mathbf{A}^T \mathbf{x}\|_1 = \max_{m=1, \dots, N} \sum_{n=1}^N |a_{nm}| = \|\mathbf{A}\|_1,$$

man beachte dabei die Vertauschung der Indizes durch die Transponierung.

Satz 3.6 (2) liefert nun $\rho(\mathbf{A}) \leq \|\mathbf{A}\|_1$, und wir haben gerade gezeigt, dass $\|\mathbf{A}\|_1 \leq 1$. Also ist 1 tatsächlich der betragsgrößte Eigenwert der PageRank Matrix, und der Google-Algorithmus ist vollständig gerechtfertigt.



Iterative Verfahren für Eigenwertprobleme



3. Eigenwertprobleme

Das folgende Beispiel zeigt, dass die HM12-Methode zur Berechnung von Eigenwerten numerisch nicht robust ist:

Die symmetrische Matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

hat die reellen Eigenwerte $\lambda_1 = 1\ 001$ und $\lambda_2 = 999$. Dies sind die Nullstellen des charakteristischen Polynoms

$$\chi_{\mathbf{A}}(\lambda) = (1\ 000 - \lambda)^2 - 1 = \lambda^2 - 2\ 000\lambda + 999\ 999.$$

Wir nehmen an, dass sich (z. B. durch Rundungsfehler) bei der Berechnung von $\chi_{\mathbf{A}}$

$$\tilde{\chi}_{\mathbf{A}}(\lambda) = \lambda^2 - 2\ 000.001\lambda + 1\ 000\ 002,$$

ergibt, d. h. ein relativer Fehler $0.001/2000 \approx 2/999\ 999 \approx 2 \cdot 10^{-6}$.



3. Eigenwertprobleme

Die exakt berechneten Nullstellen des fehlerbehafteten charakteristischen Polynoms sind

$$\tilde{\lambda}_{1,2} = 1000.0005 \pm i \cdot 0.999999875 \notin \mathbb{R},$$

woraus wir die relativen Fehler

$$|\tilde{\lambda}_{1,2} - \lambda_{1,2}| / |\lambda_{1,2}| \approx 0.0014$$

bestimmen können. Wir sehen also eine Verstärkung des relativen Fehlers mit Faktor 700.

Moral: Die Bestimmung der Nullstellen des charakteristischen Polynoms ist keine stabile numerische Methode zur Eigenwertberechnung. Die folgenden iterativen Verfahren sind also immer vorzuziehen!



3. Eigenwertprobleme

Die Bestimmung des vollständigen Spektrums mittels der Faktorisierung des charakteristischen Polynoms ist numerisch nicht robust.

Die Motivation zeigt zudem, dass es nicht *das* Eigenwertproblem gibt.

In einigen Anwendungen benötigt man Eigenwerte, in anderen Eigenvektoren, und in noch anderen beides, also *Eigenpaare*.

Oft reicht es, den betragsmäßig größten oder kleinsten Eigenwert zu bestimmen, und/oder den zugehörigen Eigenvektor. In vielen Anwendungen reicht auch ein Teil des Spektrums. Die Berechnung des vollen Spektrums ist dann eine Verschwendung von Ressourcen.



3. Eigenwertprobleme

Speziell betrachten wir in dieser VL die folgende Problemstellung:

Aufgabe (Partielles Eigenwertproblem, PEWP)

Zu $\mathbf{A} \in \mathbb{R}^{N \times N}$ finde **einen** Eigenwert $\lambda \in \mathbb{R}$ und **einen** zum Eigenwert λ gehörenden Eigenvektor $\mathbf{x} \in \mathbb{R}^N$, d. h. finde (λ, \mathbf{x}) derart, dass $\mathbf{Ax} = \lambda\mathbf{x}$.

Damit decken wir alle bisherigen Beispiele ab. Konkret entwickeln wir Verfahren zur iterativen Bestimmung des betragsgrößten Eigenpaares (d. h. des betragsgrößten EW und eines zugehörigen EV), des betragskleinsten Eigenpaares, und eines beliebigen Eigenpaares.

Insbesondere beschränken wir uns auf Probleme mit rein reellen Eigenwerten.



Die von-Mises Iteration für das betragsgrößte Eigenpaar



3. Eigenwertprobleme

Per HM123-Intuition eines Eigenpaars sollte sich bei sukzessiver Multiplikation eines beliebigen Vektors mit \mathbf{A} der „stärkste“, d. h. der zum betragsmäßig größte Eigenwert gehörende Eigenvektor „durchsetzen“. In der Mechanik nennt man das dominanter Eigenmodus.

Wir probieren aus, was exemplarisch für die Matrix $\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix}$ und den Startvektor $\mathbf{x}^{(0)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ passiert:

$$\mathbf{A} \underbrace{\begin{pmatrix} 1 \\ 1 \end{pmatrix}}_{=: \mathbf{x}^{(0)}} = \underbrace{\begin{pmatrix} 3 \\ 2 \end{pmatrix}}_{=: \mathbf{x}^{(1)}} \rightarrow \mathbf{A} \underbrace{\begin{pmatrix} 3 \\ 2 \end{pmatrix}}_{=: \mathbf{x}^{(1)}} = \underbrace{\begin{pmatrix} 7 \\ 4 \end{pmatrix}}_{=: \mathbf{x}^{(2)}} \rightarrow \mathbf{A} \underbrace{\begin{pmatrix} 7 \\ 4 \end{pmatrix}}_{=: \mathbf{x}^{(2)}} = \underbrace{\begin{pmatrix} 15 \\ 8 \end{pmatrix}}_{=: \mathbf{x}^{(3)}}$$



3. Eigenwertprobleme

Mit Papier und Bleistift errechnen wir die Eigenpaare von \mathbf{A} :

$$\lambda_1 = 2, \quad \lambda_2 = 1 \quad \text{zu den EV } \mathbf{x}_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \quad \mathbf{x}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Wir sehen im Vergleich:

$$\mathbf{x}^{(3)} = \begin{pmatrix} 15 \\ 8 \end{pmatrix} \approx 8\mathbf{x}_1 = (\lambda_1)^3 \mathbf{x}_1 = \begin{pmatrix} 16 \\ 8 \end{pmatrix}$$

Die sukzessive Multiplikation scheint gegen einen Eigenvektor zu konvergieren. Darüber hinaus scheint es einen Skalierungsfaktor zwischen den exakten und approximierten Eigenvektoren zu geben, der vom betragsgrößten EW abhängt.



3. Eigenwertprobleme

Um den Skalierungsfaktor zu quantifizieren, betrachten wir die Normen der Iterierten:

$$\left\| \underbrace{\begin{pmatrix} 1 \\ 1 \end{pmatrix}}_{x^{(0)}} \right\|_2 = \sqrt{2}, \quad \left\| \underbrace{\begin{pmatrix} 3 \\ 2 \end{pmatrix}}_{x^{(1)}} \right\|_2 = \sqrt{13}, \quad \left\| \underbrace{\begin{pmatrix} 7 \\ 4 \end{pmatrix}}_{x^{(2)}} \right\|_2 = \sqrt{65}, \quad \left\| \underbrace{\begin{pmatrix} 15 \\ 8 \end{pmatrix}}_{x^{(3)}} \right\|_2 = \sqrt{289}$$

Scharfes Hinsehen motiviert, das Verhältnis der Normen aufeinanderfolgender Iterierter zu betrachten:

$$\frac{\sqrt{13}}{\sqrt{2}} \approx 2.54, \quad \frac{\sqrt{65}}{\sqrt{13}} \approx 2.23, \quad \frac{\sqrt{289}}{\sqrt{65}} \approx 2.11 \quad \xrightarrow{?} \quad \lambda_1 = 2$$

Verdacht

Die Quotienten konvergieren gegen den Betrag $|\lambda_1|$ des betragsgrößten EW λ_1 , und die sukzessive Multiplikation mit \mathbf{A} gegen den mit dem Iterationszähler skalierten zugehörigen EV x_1 .



3. Eigenwertprobleme

Um diesen Verdacht zu formalisieren, benötigen wir eine weitere Zutat:

Satz 3.8 (Rayleigh-Quotient)

Sei $\mathbf{A} \in \mathbb{R}^{N \times N}$ und $\mathbf{x} \in \mathbb{R}^N$. Der Rayleigh-Quotient von \mathbf{A} ist

$$R_{\mathbf{A}}(\mathbf{x}) := \frac{\mathbf{x}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \frac{(\mathbf{A} \mathbf{x}, \mathbf{x})_2}{(\mathbf{x}, \mathbf{x})_2} = \frac{(\mathbf{A} \mathbf{x}, \mathbf{x})_2}{\|\mathbf{x}\|_2^2}.$$

Ist \mathbf{x} ein Eigenvektor von \mathbf{A} , so ist $R_{\mathbf{A}}(\mathbf{x})$ der zugehörige Eigenwert, d. h. es gilt $\mathbf{A} \mathbf{x} = \lambda \mathbf{x} = R_{\mathbf{A}}(\mathbf{x}) \mathbf{x}$.

Beweis: Sei λ der Eigenwert zum Eigenvektor \mathbf{x} . Zu zeigen: $R_{\mathbf{A}}(\mathbf{x}) = \lambda$.

$$R_{\mathbf{A}}(\mathbf{x}) = \frac{\mathbf{x}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \frac{\mathbf{x}^T (\lambda \mathbf{x})}{\mathbf{x}^T \mathbf{x}} = \lambda \frac{\mathbf{x}^T \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \lambda. \quad \square$$



3. Eigenwertprobleme

Die sukzessive Multiplikation mit \mathbf{A} liefert also Approximationen des Eigenvektors zum betragsgrößten Eigenwert. Mit dem Rayleigh-Quotienten können wir diesen dann berechnen.

Als Ergebnis erhalten wir die **Von-Mises Iteration**, zunächst in einer didaktischen Variante für besseres Verständnis.

Zur Verbesserung der Stabilität und Robustheit arbeiten wir danach mit normierten Vektoren, d.h. $\|\mathbf{x}\|_2 = 1$ für ein $\mathbf{x} \in \mathbb{R}^N$. Nachrechnen bestätigt sofort, dass wir jeden Vektor normieren können, indem wir komponentenweise durch seine Norm dividieren, d. h. $\mathbf{x} / \|\mathbf{x}\|_2$ berechnen. Das geht übrigens mit jeder Norm.

Die finale Version des Algorithmus erhalten wir durch Einsparung redundanter Berechnungen und Reduzierung des Speicherbedarfs auf das Minimum.

R. von Mises (1883–1953), enorm produktiver Numeriker und Mechaniker



3. Eigenwertprobleme

Von-Mises Iteration (didaktische Version)

Iteration über $k = 0, 1, \dots$ mit einem Startwert $\mathbf{x}^{(0)}$:

- ① Rayleigh-Quotient: $\lambda^{(k)} = \frac{(\mathbf{A}\mathbf{x}^{(k)}, \mathbf{x}^{(k)})_2}{\|\mathbf{x}^{(k)}\|_2^2}$
- ② Konvergenzkontrolle: Falls $\|\mathbf{A}\mathbf{x}^{(k)} - \lambda^{(k)}\mathbf{x}^{(k)}\|_2 < TOL$, dann stop
- ③ Nächste Approximation: $\mathbf{x}^{(k+1)} = \mathbf{A}\mathbf{x}^{(k)}$

Verbesserungspotential #1: $\mathbf{A}\mathbf{x}^{(k)}$ wird noch mehrfach berechnet. Das lösen wir durch die Einführung eines Hilfsvektors.

Verbesserungspotential #2: Wir müssen weder alle $\lambda^{(k)}$ noch alle $\mathbf{x}^{(k)}$ speichern, da uns nur die finale Approximation des Eigenpaares interessiert.

Verbesserungspotential #3: Einbau der Normierung.



3. Eigenwertprobleme

Algorithmus 3.9 : Von-Mises Iteration (VMI)

input : $A \in \mathbb{R}^{N \times N}$ mit $\text{sp}(A) \subset \mathbb{R}$, Startvektor $x^{(0)} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$,
TOL > 0 und $k_{\max} \in \mathbb{N}$

output : Approximation λ des betragsgrößten Eigenwerts und Approximation x des
zugehörigen Eigenvektors von A

```
1  $k = 0;$ 
2  $r = \text{TOL} + 1;$  % Trick, damit die Schleife mindestens einmal durchläuft
3  $x = x^{(0)} / \|x^{(0)}\|_2;$ 
4  $y = Ax;$ 
5 while  $k < k_{\max}$  and  $r > \text{TOL}$  do
6    $\lambda = y^T x;$  % Rayleigh-Quotient  $(Ax)^T x / \|x\|_2^2$ , da  $\|x\|_2 = 1$  und  $y = Ax$ 
7    $x = y / \|y\|_2;$ 
8    $y = Ax;$ 
9    $r = \|y - \lambda x\|_2;$ 
10   $k = k + 1;$ 
11 end
```



3. Eigenwertprobleme

Bemerkungen:

- Die Von-Mises Iteration heißt auch **Potenzmethode (power iteration)**, sie löst das PEWP für das Eigenpaar zum betragsmäßig größten Eigenwert.
- Die k -te Iterierte $\lambda^{(k)}$ (λ im Algorithmus) ist eine Approximation des betragsgrößten Eigenwerts von \mathbf{A} .
- Die k -te Iterierte $\mathbf{x}^{(k)}$ ist eine Approximation an **einen** Eigenvektor mit Norm 1 zum betragsgrößten Eigenwert λ_1 von \mathbf{A} .
- Der Aufwand beträgt eine Matrix-Vektor Multiplikation und vier Vektor-Vektor Operationen pro Iteration.



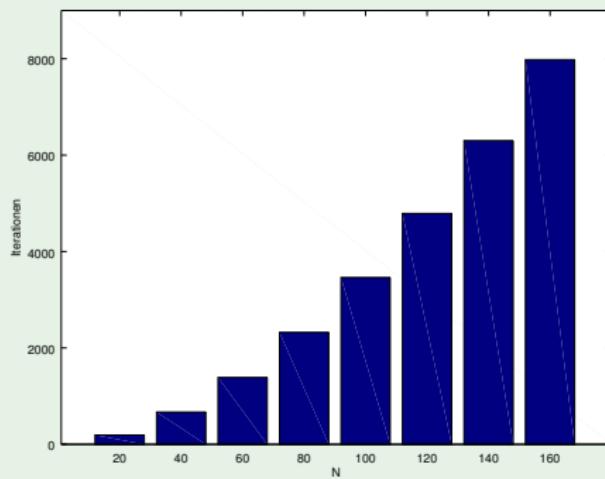
3. Eigenwertprobleme

Beispiel 3.10 (Raclette-Matrix)

Wir betrachten wieder einmal die Standardmatrix zum Raclette-Problem:

$$\mathbf{A} = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{N \times N}$$

Die Iterationszahl bis zur absoluten Toleranz $\text{TOL} = 10^{-6}$ steigt mit N :

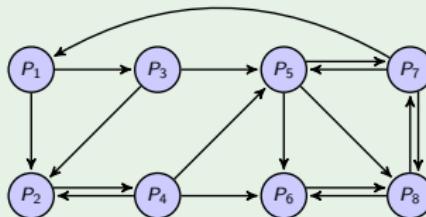




3. Eigenwertprobleme

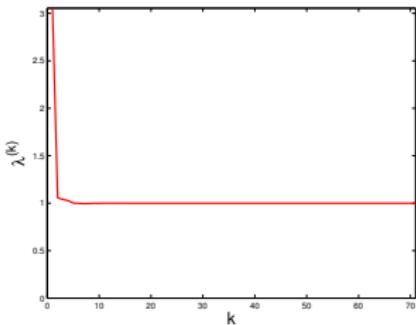
Beispiel 3.1 (PageRank Eigenwertproblem cont.)

Algorithmus 3.9 mit $x^{(0)} = (1, \dots, 1)^T$ und $\text{TOL} = 10^{-6}$:

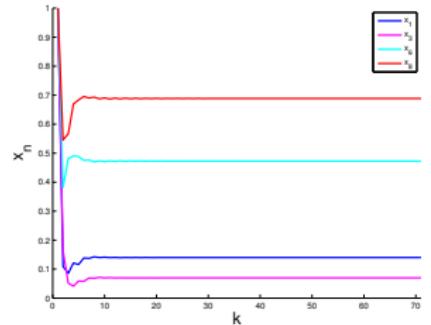


$$\text{Approx. EV: } x \approx \begin{pmatrix} 0.1400 \\ 0.1576 \\ 0.0700 \\ 0.1576 \\ 0.2276 \\ 0.4727 \\ 0.4201 \\ 0.6886 \end{pmatrix}$$

Konvergenz des größten EW



Komponentenweise Konvergenz des EV





3. Eigenwertprobleme

Man kann zeigen:

Satz 3.11 (Konvergenzgeschwindigkeit der Von-Mises Iteration)

Sei $\mathbf{A} \in \mathbb{R}^{N \times N}$ mit reellen Eigenwerten $\lambda_1, \dots, \lambda_N \in \mathbb{R}$. Die Eigenwerte seien absteigend sortiert, d. h.

$$|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_N|$$

Falls die von-Mises Iteration konvergiert, so gilt für die Konvergenzgeschwindigkeit

$$|\lambda^{(k)} - \lambda_1| = C \left| \frac{\lambda_2}{\lambda_1} \right|^k$$

mit einer Konstante C , die nicht von $\text{sp}(\mathbf{A})$ abhängt.

Wir sehen: Je kleiner $|\lambda_2|$ im Vergleich zu $|\lambda_1|$ ist, desto schneller konvergiert das Verfahren. Damit kann die schnelle Konvergenz beim PageRank-Beispiel erklärt werden, wenn zusätzlich zu $|\lambda_1| = \lambda_1 = 1$ auch $|\lambda_2| \approx 0.8702$ mit ViPLab berechnet wird: Wir erhalten $\frac{|\lambda_2|}{\lambda_1} \approx 0.8702$.



3. Eigenwertprobleme

Der Satz erklärt auch die langsame und von der Problemgröße N abhängige Konvergenz beim Raclette-Beispiel:

Beispiel 3.10 (Konvergenzgeschwindigkeit der VMI) cont.

Die Eigenwerte der Raclette-Matrix $\text{tridiag}(-1, 2, -1) \in \mathbb{R}^{N \times N}$ lauten für $n = 1, \dots, N$:

$$\lambda_n = 2 + 2 \cos\left(\frac{n\pi}{N+1}\right)$$

Wir lesen ab, dass $\lambda_2/\lambda_1 \rightarrow 1$ für $N \rightarrow \infty$, weil für beide EW der Cosinus gegen 1 geht. Die Konvergenzgeschwindigkeit wird also tatsächlich langsamer mit wachsendem N .

Mit Hilfe der Additionstheoreme kann man zusätzlich den beobachtbaren quadratischen Anstieg nachrechnen.



3. Eigenwertprobleme

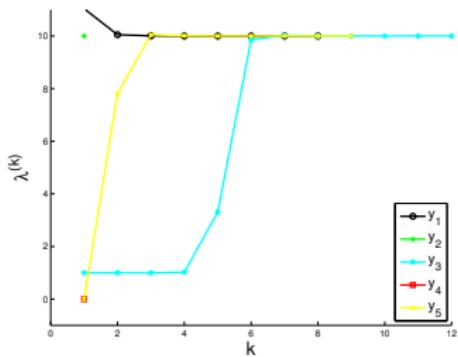
Beispiel 3.12 (Konvergenz in Abhangigkeit vom Startvektor)

Matrix \mathbf{A} mit EW 10, 1, 0

$$\mathbf{A} = \begin{pmatrix} 10 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Startvektoren $\mathbf{x}^{(0)} \in \{\mathbf{y}_1, \dots, \mathbf{y}_5\} =$

$$\left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} -0.1104 \\ 0.9939 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 10^{-10} \\ 1 \end{pmatrix} \right\}$$



Die zugehangigen EV zu $\lambda_1, \lambda_2, \lambda_3$ sind

$$\mathbf{x}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{x}_2 = \begin{pmatrix} -\frac{1}{9} \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{x}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Fur Startvektoren $\mathbf{y}_1, \mathbf{y}_3, \mathbf{y}_5$: unterschiedlich schnelle Konvergenz gegen $\lambda_1 = |\lambda_{\max}|$.

Fur Startvektor \mathbf{y}_2 : sofort $\lambda_1 = |\lambda_{\max}|$

Fur Startvektor \mathbf{y}_4 : sofort $\lambda_3 = 0$.



3. Eigenwertprobleme

Den ersten Fall (Konvergenz gegen das betragsgrößte Eigenpaar) haben wir erwartet, und den zweiten Fall sowieso. Der dritte Fall zeigt, dass wir bei den Startwerten aufpassen müssen. Das Experiment motiviert das folgende Resultat:

Satz 3.13 (Konvergenz in Abhängigkeit vom Startvektor)

Die VMI konvergiert gegen das betragsgrößte Eigenpaar, falls der Startvektor $x^{(0)}$ kein Eigenvektor eines anderen Eigenwerts oder keine Linearkombination anderer Eigenvektoren ist. Insbesondere konvergiert die VMI, wenn die geometrische und algebraische Vielfachheit des betragsmäßig größten Eigenwerts Eins ist.

Die Konvergenz ist zudem sichergestellt, wenn es einen EV x_1 zum betragsgrößten EW gibt, der im Skalarprodukt mit der Startlösung nicht verschwindet, d. h. $(x^{(0)}, x_1)_2 \neq 0$. Insbesondere darf also die Startlösung nicht orthogonal auf den gesuchten EV stehen. Damit kann der Algorithmus für mehrfache EW erweitert werden.



Die Wielandt-Iteration



3. Eigenwertprobleme

Nun überlegen wir uns die Approximation des betragskleinsten Eigenpaars.

Wir erinnern uns (Satz 3.7 (3)): Ist \mathbf{A} regulär, so gilt

$$\lambda \in \text{sp}(\mathbf{A}) \Leftrightarrow \lambda_{\text{inv}} := \lambda^{-1} \in \text{sp}(\mathbf{A}^{-1}).$$

Also folgt mit gleichem Eigenvektor

$$\min\{|\lambda| \mid \lambda \in \text{sp}(\mathbf{A})\} = (\max\{|\lambda_{\text{inv}}| \mid \lambda_{\text{inv}} \in \text{sp}(\mathbf{A}^{-1})\})^{-1}$$

Wir müssen also „eigentlich“ nur den betragsgrößten EW λ_{inv} von \mathbf{A}^{-1} mit der VMI berechnen, und setzen dann $\lambda_{\min} = (\lambda_{\text{inv}})^{-1}$.

Falls \mathbf{A} dicht besetzt ist, sind wir im Prinzip fertig: Wir bestimmen die Inverse mit einer Variante der LR-Zerlegung (VL 1), wenden die von-Mises Iteration an, und geben das Reziproke der resultierenden Approximation zurück.



3. Eigenwertprobleme

Für dünn besetzte Matrizen ist das nicht praktikabel, vgl. VL 2. Wir müssen die explizite Bestimmung der Inversen verhindern, weil sie potentiell dicht besetzt und deshalb nicht abspeicherbar ist.

Wir wissen: Eine Multiplikation mit \mathbf{A}^{-1} können wir auch schreiben als Lösung eines Hilfs-LGS.

Die **Wielandt-Iteration** (Algorithmus 3.14, nächste Folie) ersetzt die sukzessive Multiplikation mit \mathbf{A} in der von-Mises Iteration nicht durch die sukzessive Multiplikation mit \mathbf{A}^{-1} , sondern durch die Lösung eines Hilfs-LGS in jedem Schritt. Sie kommt somit ohne die explizite Berechnung von \mathbf{A}^{-1} aus.

Diese Vorgehensweise kann auch für dicht besetzte Matrizen effizienter sein.



3. Eigenwertprobleme

Algorithmus 3.14 : Wielandt-Iteration

input : $A \in \mathbb{R}^{N \times N}$ mit $\text{sp}(A) \subset \mathbb{R}$, Startvektor $x^{(0)} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$,
TOL > 0 und $k_{\max} \in \mathbb{N}$

output : Approximation λ des betragskleinsten Eigenwertes und Eigenvektors x von A

```
1  $k = 0;$ 
2  $r = \text{TOL} + 1;$ 
3  $x = x^{(0)} / \|x^{(0)}\|_2;$ 
4 löse  $Ay = x;$ 
5 while  $k < k_{\max}$  and  $r > \text{TOL}$  do
6    $\lambda_{\text{inv}} = y^T x;$ 
7    $x_{\text{old}} = x;$ 
8    $x = y / \|y\|_2;$ 
9   löse  $Ay = x;$ 
10   $r = \|x_{\text{old}} - \frac{1}{\lambda_{\text{inv}}} x\|_2;$       % analog zur VMI
11   $k = k + 1;$ 
12 end
13  $\lambda = \frac{1}{\lambda_{\text{inv}}};$ 
```



3. Eigenwertprobleme

Bemerkungen:

- Die Wielandt-Iteration heißt auch **Inverse Iteration**.
- Es gibt ähnliche Konvergenzaussagen wie bei der VMI.
- Bei einer vollbesetzten Matrix sollte die LR- oder Cholesky-Zerlegung von \mathbf{A} vor der Iteration berechnet und dann zur wiederholten Lösung von $\mathbf{Ax} = \mathbf{y}$ verwendet werden, vgl. VL 1.
- Bei einer dünnbesetzten Matrix benötigt man ein effizientes iteratives Verfahren zur Lösung von $\mathbf{Ax} = \mathbf{y}$ in jedem Schritt, vgl. VL 2.



3. Eigenwertprobleme

Auch für die Berechnung anderer Eigenwerte haben wir mit Satz 3.7 (4) schon die Grundlage gelegt: Für beliebiges $\mu \in \mathbb{R}$ gilt

$$\lambda \in \text{sp}(\mathbf{A}) \Leftrightarrow \lambda - \mu \in \text{sp}(\mathbf{A} - \mu\mathbf{I}),$$

und die zugehörigen Eigenvektoren sind identisch. Gemäß unserer Überlegungen zur Wielandt-Iteration ist zudem $\frac{1}{\lambda - \mu}$ Eigenwert von $(\mathbf{A} - \mu\mathbf{I})^{-1}$, immer noch zum selben Eigenvektor.

Das motiviert die folgende Idee: Wenn λ der Eigenwert von \mathbf{A} ist, der μ am nächsten liegt, dann ist $\frac{1}{\lambda - \mu}$ der betragsmäßig größte Eigenwert von $(\mathbf{A} - \mu\mathbf{I})^{-1}$. Wir müssen also nur auf $(\mathbf{A} - \mu\mathbf{I})^{-1}$ die von-Mises Iteration, bzw. auf $(\mathbf{A} - \mu\mathbf{I})$ die Wielandt-Iteration anwenden, die dann gegen den Eigenvektor zum Eigenwert λ von \mathbf{A} konvergieren, der μ am nächsten liegt.



3. Eigenwertprobleme

Algorithmus 3.15 : Wielandt-Iteration mit Shift

input : $\mathbf{A} \in \mathbb{R}^{N \times N}$ mit $\text{sp}(\mathbf{A}) \subset \mathbb{R}$, Startvektor $\mathbf{x}^{(0)} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$,
TOL > 0, $k_{\max} \in \mathbb{N}$ und Schätzung $\mu \in \mathbb{R}$ für EW

output : Approximation λ des Eigenwertes, welcher am nächsten an μ liegt,
und des zugehörigen Eigenvektors \mathbf{x} von \mathbf{A}

- 1 $[\nu, \mathbf{x}] = \text{WIELANDT}(\mathbf{A} - \mu \mathbf{I}, \mathbf{x}^{(0)}, k_{\max}, \text{TOL});$
 - 2 $\lambda = \nu + \mu;$
-



3. Eigenwertprobleme

Beispiel 3.16 (Wielandt-Iteration mit Shift)

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{pmatrix} \quad \text{Eigenwerte: } \lambda_1 = 0, \lambda_2 = 1, \lambda_3 = 3$$

Die Wielandt-Iteration mit Shift konvergiert für Startwerte aus den folgenden Intervallen gegen die Eigenwerte:

$$\mu \in (-\infty, 0.5) \rightarrow \lambda_1, \quad \mu \in (0.5, 2) \rightarrow \lambda_2, \quad \mu \in (2, \infty) \rightarrow \lambda_3$$

Für $\mu = 0.5$ und $\mu = 2$ ist keine Aussage möglich.

Erinnerung: Alle Beispiele können mit den ViPLab-Übungen nachvollzogen werden.

Die Wahl von μ erfordert also Vorwissen über die grobe Lage der Eigenwerte von **A**. Je näher μ an einem tatsächlichen EW liegt, desto schneller ist die Konvergenz, vgl. Satz 3.11 und die analoge Aussage für die Wielandt-Iteration.



Zusammenfassung

- Eigenwertprobleme haben vielfältige Anwendungen.
- Für partielle Eigenwertprobleme existieren i. W. drei Iterationsverfahren, bei denen der Startvektor „geeignet“ gewählt werden muss.
- Die von-Mises Iteration (Potenzmethode) liefert Approximationen des betragsgrößten EW und eines zugehörigen EV. Jeder Schritt erfordert i. W. eine Matrix-Vektor Multiplikation.
- Die Wielandt-Iteration berechnet eine Approximation des betragskleinsten EW und EV. Jeder Schritt erfordert die Lösung eines LGS.
- Das Wielandt-Verfahren mit Shift führt zu dem EW, welcher am nächsten am Eingabeparameter μ liegt.



3. Eigenwertprobleme

Hausaufgaben und Ankündigungen

- Nächste Woche findet keine VL statt wegen des Feiertags.
- Deshalb findet die 1. VÜ „versetzt“ statt, und zwar am FR 27.4. und MI 2.5.
- Die nächsten Termine für Tutorien finden Sie in ILIAS.
- Ideen zur Nachbereitung dieser VL finden Sie auf den folgenden Folien.
- Übernächste Woche beginnt Teil B, mit Techniken für nichtlineare Gleichungssysteme. Einige Vorschläge zur optimalen Vorbereitung sind:
 - ▶ Wiederholen Sie aus der HM12 die Differentialrechnung in 1D, sowie die Stetigkeit und den Zwischenwertsatz.
 - ▶ Wiederholen Sie aus der HM123 die mehrdimensionalen Analoga, d. h. insbesondere die Begriffe Gradient, Hesse-Matrix und Jacobi-Matrix.



Beispieldaufgaben



3. Eigenwertprobleme

Konvergenz der von-Mises und Wielandt-Iterationen

Gegeben seien

$$\mathbf{A} = \begin{pmatrix} -2 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1/2 \end{pmatrix} \quad \mathbf{x}^{(0)} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

Gegen welchen Eigenwert von \mathbf{A} konvergiert die von-Mises Iteration und die Wielandt-Iteration für den Startvektor $\mathbf{x}^{(0)}$?



3. Eigenwertprobleme

Lösungshinweise: Man könnte hier natürlich einige Iterationen der beiden Verfahren durchführen und dann raten. Viel Zeit spart man, wenn man begriffen hat, wie diese Verfahren funktionieren, und wenn man aus der HM123 noch weiß, bei welcher Matrixstruktur man direkt Eigenwerte ablesen kann ohne sie berechnen zu müssen.

Ergebnis: VMI liefert den Eigenwert -2 , und Wielandt den Eigenwert $\frac{1}{2}$.



3. Eigenwertprobleme

Wielandt-Iteration mit Shift

Die Wielandt-Iteration mit Shift $\mu = 4$ wird angewendet auf die Matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 3 & 15 \\ 0 & 5 & -9 \\ 0 & 0 & -2 \end{pmatrix}.$$

Geben Sie die Koeffizientenmatrix des LGS an, das in jedem Schritt gelöst werden muss.



3. Eigenwertprobleme

Lösungshinweise: Die Antwort liest man an der Vorbereitung des Verfahrens ab.

Ergebnis:

$$\mathbf{A} - \mu \mathbf{I} = \begin{pmatrix} -3 & 3 & 15 \\ 0 & 1 & -9 \\ 0 & 0 & -6 \end{pmatrix}$$

Weitere Verständnisübung: Gegen welchen Eigenwert konvergiert diese Iteration?



3. Eigenwertprobleme

Durchführung der von-Mises Iteration

Gegeben seien

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix} \quad \mathbf{x}^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Wie lautet die Approximation $\mathbf{x}^{(1)}$ des EV zum betragsgrößten EW von \mathbf{A} bei Anwendung der von-Mises Iteration mit Startvektor $\mathbf{x}^{(0)}$?



3. Eigenwertprobleme

Lösungshinweise: Eine Approximation ergibt sich nach einer Matrix-Vektor Multiplikation. Die gesuchte Approximation muss natürlich die beiden Normierungen enthalten, die im Algorithmus vorkommen!

Ergebnis:

$$\mathbf{x}^{(1)} = \begin{pmatrix} 0 \\ 1/\sqrt{5} \\ 2/\sqrt{5} \end{pmatrix}$$



3. Eigenwertprobleme

Tiefes Verständnis der VMI

In dieser Aufgabe soll die Intuition aus Beispiel 3.12 allgemein bestätigt werden. Sei \mathbf{x}_m Eigenvektor zum Eigenwert λ_m , wobei wieder $|\lambda_1| \geq \dots \geq |\lambda_N|$ sortiert sein soll. Zeigen Sie, dass die VMI $\mathbf{x}^{(1)} = \mathbf{x}_m$ liefert, sofern $\mathbf{x}^{(0)} = \mathbf{x}_m$.

Natürlich ist klar, dass die Klausur weitestgehend frei von Beweisen sein wird. Das liegt ganz einfach daran, dass die Vorlesung weitestgehend frei von Beweisen ist.

Dieser Beweis ist allerdings sehr verständnisfördernd, weil er darin besteht, Algorithmus 3.9 symbolisch durchzuführen, d. h. ohne konkrete Zahlen.



3. Eigenwertprobleme

Lösungshinweise: Ausgangspunkt ist $\mathbf{x}^{(0)} = \mathbf{x}_m$ ($\|\mathbf{x}_m\|_2 = 1$), d. h. ein normierter EV zu einem EW $\lambda_m, m \geq 2$:

$$\mathbf{x} = \frac{\mathbf{x}_m}{\|\mathbf{x}_m\|_2} = \mathbf{x}_m$$

Mit dem \mathbf{y} aus dem Algorithmus erhalten wir:

$$\mathbf{y} = \mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}_m = \lambda_m \mathbf{x}_m, \lambda = \mathbf{y}^\top \mathbf{x} = \lambda_m \mathbf{x}^\top \mathbf{x} = \lambda_m$$

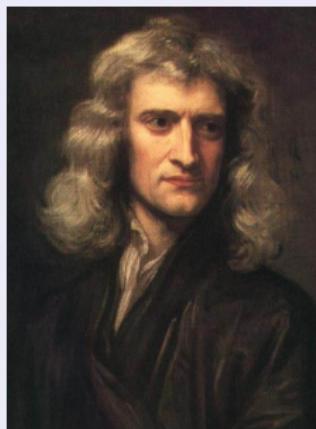
Der Algorithmus ergibt:

$$\mathbf{x} = \frac{\mathbf{y}}{\|\mathbf{y}\|_2} = \frac{\lambda_m \mathbf{x}_m}{|\lambda_m|}, \mathbf{y} = \mathbf{A}\mathbf{x} = \frac{\lambda_m^2}{|\lambda_m|} \mathbf{x}_m, r = \|\mathbf{y} - \lambda \mathbf{x}\|_2 = \left| \frac{\lambda_m^2}{|\lambda_m|} \mathbf{x}_m - \lambda_m \frac{\lambda_m \mathbf{x}_m}{|\lambda_m|} \right| = 0$$

Also liefert die VMI nach **einem** Schritt den EW $\lambda_m \neq \lambda_1$ und nicht λ_1 .



4. Nichtlineare Gleichungssysteme



Sir Isaac Newton

04.01.1643 — 31.03.1727

„Nature and Nature's Laws lay hid in Night:
God said, Let Newton be! and all was Light.“
Alexander Pope (Inschrift auf Newtons Grab)



Inhalte und Ziele dieser Vorlesungseinheit

- Erster Kontakt mit nichtlinearen Gleichungssystemen, insbesondere Realisierung, dass die Lösung eines nichtlinearen Gleichungssystems immer als nichtlineares Nullstellenproblem formuliert werden kann.
- Kennenlernen des Newtonverfahrens für nichtlineare ein- und mehrdimensionale Nullstellenprobleme, inklusive seiner Vor- und Nachteile.
- Erste Wiederholung der mehrdimensionalen Differentialrechnung, auch als Vorbereitung für VL 5.



Motivation:

Nichtlineare Gleichungssysteme



4. Nichtlineare Gleichungssysteme

Auf einem Computer müssen mit den 4 Grundoperationen $+$, $-$, $*$, $/$ kompliziertere Operationen durchgeführt werden, z. B. die Berechnung von $g(a) := \sqrt{a}$. Das können wir auch als Nullstellenproblem formulieren:

Beispiel 4.1 (Quadratwurzel)

Zu $a \in \mathbb{R}$ finde eine Nullstelle der nichtlinearen Funktion $f: \mathbb{R} \rightarrow \mathbb{R}$ mit

$$f(x) := x^2 - a.$$

Jede Nullstelle von f ist Lösung des Ausgangsproblems, $g(a)$ zu berechnen. Die Funktionen f und g sind einfache Beispiele **nichtlinearer** Funktionen. Auch wenn dieses Beispiel es nicht vermuten lässt, ist die Lösung nichtlinearer (Nullstellen-)Probleme ungleich schwieriger als die Lösung linearer Probleme, sowohl in der Theorie als auch in der Praxis.



Fortsetzung Beispiel 4.1 (Quadratwurzel)

Zu $a \in \mathbb{R}$ finde eine Nullstelle der nichtlinearen Funktion $f: \mathbb{R} \rightarrow \mathbb{R}$ mit

$$f(x) := x^2 - a.$$

An diesem Beispiel sehen wir bereits zwei wichtige Dinge:

- Jede Aufgabe „zu y finde x^* mit $g(x^*) = y$ “ lässt sich immer als Nullstellensuche „finde eine Nullstelle x^* von $f(x) := g(x) - y$ “ formulieren.
- Lösungen müssen nicht existieren (hier: $a < 0$, beachte $f: \mathbb{R} \rightarrow \mathbb{R}$), und wenn doch, müssen sie nicht eindeutig sein ($a = 0$ vs. $a > 0$).

Beide Eigenschaften sind typisch für **nichtlineare** Gleichungssysteme, im Gegensatz zu linearen Gleichungssystemen.



4. Nichtlineare Gleichungssysteme

Beispiel 4.2 (Spionagesatelliten der NSA)

Gegeben seien die Himmelskörper Erde, Mond und Sonne, beschrieben durch die Punktmassen $m_1, m_2, m_3 \in \mathbb{R}$ an den festen Positionen $\mathbf{p}_i = (x_i, y_i, z_i)^\top \in \mathbb{R}^3$ ($i = 1, 2, 3$); und ein Spionagesatellit mit bekannter Punktmasse $m \in \mathbb{R}$ an der unbekannten Position $\mathbf{p} = (x, y, z)^\top \in \mathbb{R}^3$.

Ein NSA-Problem ist nun: An welcher Position \mathbf{p} muss der Satellit platziert werden, so dass er geostationär ist? Geostationär bedeutet dabei, dass er schwerelos im Kräftegleichgewicht zwischen m_1, m_2 und m_3 liegt, d. h. dass er sich nicht bewegt relativ zu einer festen Position auf der Erdoberfläche?

Die Newton'schen Gravitationsgesetze besagen hier: Die (betragsmäßige) Kraft \mathbf{F}_i zwischen dem Satelliten und einem der drei Himmelskörper ist

$$\mathbf{F}_i = G m \frac{m_i}{\|\mathbf{p}_i - \mathbf{p}\|_2^3} (\mathbf{p}_i - \mathbf{p}) \quad i = 1, 2, 3.$$

Hierbei ist $G \approx 6.6740831 \cdot 10^{-11} \frac{\text{m}^3}{\text{kg} \cdot \text{s}^2}$ die Gravitationskonstante, und $\mathbf{p}_i - \mathbf{p}$ der Abstand.

Die Details sind nicht so wichtig, spannender ist das vektorwertige Nullstellenproblem, das sich auf der nächsten Folie ergibt.



4. Nichtlineare Gleichungssysteme

Die Gesamtkraft, die auf den Satelliten wirkt, ist einfach die Addition aller drei Einzelkräfte durch Sonne, Mond und Erde:

$$\mathbf{F} = \sum_{i=1}^3 \mathbf{F}_i = G m \sum_{i=1}^3 \frac{m_i}{\|\mathbf{p}_i - \mathbf{p}\|_2^3} (\mathbf{p}_i - \mathbf{p})$$

Geostationär bedeutet, dass sich alle Kräfte \mathbf{F}_i zu Null aufaddieren:

$$\mathbf{F} \stackrel{!}{=} \mathbf{0}$$

Zur Bestimmung der unbekannten Position \mathbf{p} des Satelliten müssen wir also eine Nullstelle $\mathbf{x}^* = (x^*, y^*, z^*)^T$ einer nichtlinearen vektorwertigen Funktion $\mathbf{F}: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ bestimmen: Finde \mathbf{x}^* mit $\mathbf{F}(\mathbf{x}^*) = \mathbf{0}$.



4. Nichtlineare Gleichungssysteme

Komponentenweise lautet das NSA-Nullstellenproblem übrigens:

$$\boldsymbol{F}(x, y, z) := \sum_{i=1}^3 \boldsymbol{F}_i(x, y, z) = \begin{pmatrix} \sum_{i=1}^3 \frac{G m m_i (x_i - x)}{((x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2)^{3/2}} \\ \sum_{i=1}^3 \frac{G m m_i (y_i - y)}{((x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2)^{3/2}} \\ \sum_{i=1}^3 \frac{G m m_i (z_i - z)}{((x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2)^{3/2}} \end{pmatrix} = \mathbf{0}$$

Wir sehen, dass Nullstellenprobleme schon in \mathbb{R}^3 schnell sehr kompliziert wirken können. Wir sehen aber auch, dass das letztendlich dasselbe ist wie das Problem $f(x) = 0$ aus dem vorherigen Beispiel.



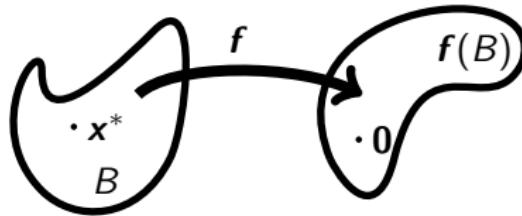
4. Nichtlineare Gleichungssysteme

Wir fassen die Beispiele zusammen:

Definition 4.3 (Allgemeines Nullstellenproblem)

Sei $f : B \subseteq \mathbb{R}^N \rightarrow \mathbb{R}^N$ gegeben, und sei B offen und beschränkt. Das **Nullstellenproblem** lautet: Finde ein $x^* \in B$ mit

$$f(x^*) = 0. \quad (4.3)$$



Die Offenheit und Beschränktheit benötigen wir später bei Verfahren, die die Differenzierbarkeit von f erfordern.



4. Nichtlineare Gleichungssysteme

Die Existenz einer Lösung hängt von der speziellen Funktion f ab, d. h. es ist keine allgemeine Existenzaussage möglich. Gleichermaßen gilt für die Eindeutigkeit der Lösung, d. h. Gleichung (4.3) kann mehrere Lösungen besitzen. Beides wird die Numerik vor Herausforderungen stellen.

Selbst für $N = 1$, also in \mathbb{R} , existieren im Allgemeinen keine geschlossenen Lösungsformeln. Wir sind also auf numerische Verfahren angewiesen.

Im Folgenden skizzieren wir zuerst anschauliche Ideen für $N = 1$, und übertragen diese dann auf den allgemeinen Fall.



Das Bisektionsverfahren in 1D



4. Nichtlineare Gleichungssysteme

Erinnerung HM12:

Satz 4.4 (Folgerung aus dem Zwischenwertsatz)

Sei $f : [a, b] \rightarrow \mathbb{R}$ stetig, und f besitze einen Vorzeichenwechsel (VZW) in $[a, b]$, d. h. $f(a) < 0 < f(b)$ oder $f(a) > 0 > f(b)$. Dann existiert eine Nullstelle $x^* \in (a, b)$ mit $f(x^*) = 0$.

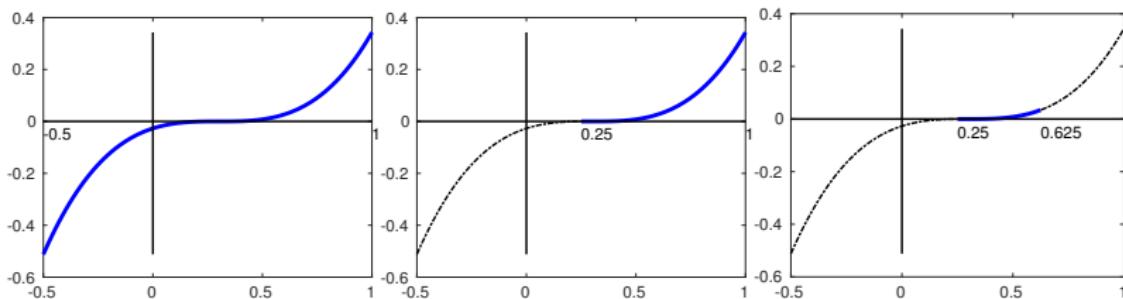
Wichtig ist: In dieser allgemeinen Form ist das keine Eindeutigkeitsaussage. Garantiert wird lediglich die Existenz einer Nullstelle.

Knobelaufgabe: Ist die Nullstelle eindeutig, wenn f streng monoton wachsend oder fallend ist?



4. Nichtlineare Gleichungssysteme

Die Idee des Bisektionsverfahrens ist nun sehr naheliegend: Bestimme das Vorzeichen von f im Intervallmittelpunkt x_M , und iteriere je nach Vorzeichen von $f(x_M)$ das Verfahren entweder im linken oder im rechten Teilintervall. Nach hinreichend vielen Iterationen ist dann $x_M \approx x^*$.





4. Nichtlineare Gleichungssysteme

Algorithmus 4.5 : Bisektionsverfahren

input : $f: [a, b] \rightarrow \mathbb{R}$ stetig mit $f(a) \cdot f(b) < 0$, $\text{TOL} > 0$, $k_{\max} \in \mathbb{N}$

output : Approximation x^* der Nullstelle von f

```
1  $k = 0;$ 
2 while  $k < k_{\max}$  and  $(b - a) > \text{TOL}$  do
3    $x = \frac{1}{2}(a + b);$            % Intervallmittelpunkt
4   if  $f(a) \cdot f(x) < 0$  then
5      $b = x;$                   % NST im linken Intervall wegen VZW
6   else
7      $a = x;$                   % NST im rechten Intervall wegen VZW
8   end
9    $k = k + 1;$ 
10 end
11  $x^* = \frac{1}{2}(a + b);$ 
```



4. Nichtlineare Gleichungssysteme

Beispiel 4.6 (Bisektionsverfahren zur Wurzelberechnung)

$$f(x) = x^2 - 0.81, \quad [a, b] = [0, 1] \Rightarrow x^* = 0.9$$

k	$[a^{(k)}, b^{(k)}]$	$x^{(k)}$	$ x^{(k)} - x^* / x^* $
$k=0$	[0.000000, 1.000000]	0.500000	0.444444
$k=1$	[0.500000, 1.000000]	0.750000	0.166667
$k=2$	[0.750000, 1.000000]	0.875000	0.027778
$k=3$	[0.875000, 1.000000]	0.937500	0.041667
$k=4$	[0.875000, 0.937500]	0.906250	0.006944
$k=5$	[0.875000, 0.906250]	0.890625	0.010417
$k=6$	[0.890625, 0.906250]	0.898438	0.001736
$k=7$	[0.898438, 0.906250]	0.902344	0.002604
$k=8$	[0.898438, 0.902344]	0.900391	0.000434
$k=9$	[0.898438, 0.900391]	0.899414	0.000651



4. Nichtlineare Gleichungssysteme

Das Verfahren benötigt ein (geratenes) Startintervall, das einen Vorzeichenwechsel beinhaltet *und* eine Nullstelle enthält.

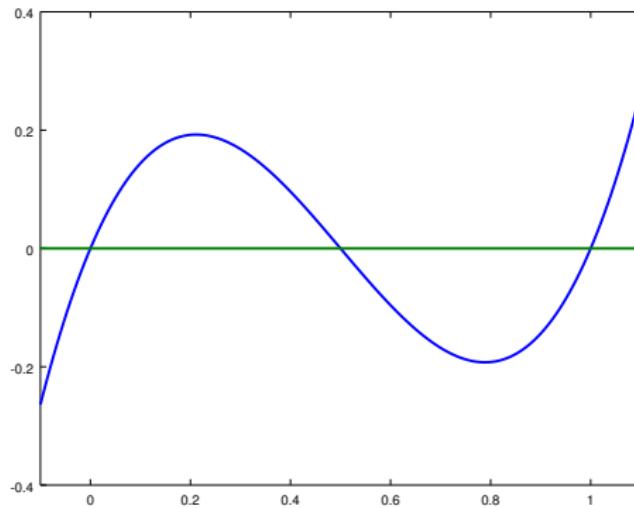
Dann ist die aktuelle Intervalllänge immer eine Fehlerschranke:

$$|x^{(k)} - x^*| \leq \frac{b-a}{2^{k+1}}.$$

Der absolute Fehler sinkt allerdings nicht unbedingt monoton, vergleiche das vorherige Beispiel.

Bisektion kann für beliebige stetige Funktionen angewendet werden, es ist keine Differenzierbarkeit erforderlich. Die Erweiterung auf höhere Dimensionen ($N > 1$) ist komplett unklar.

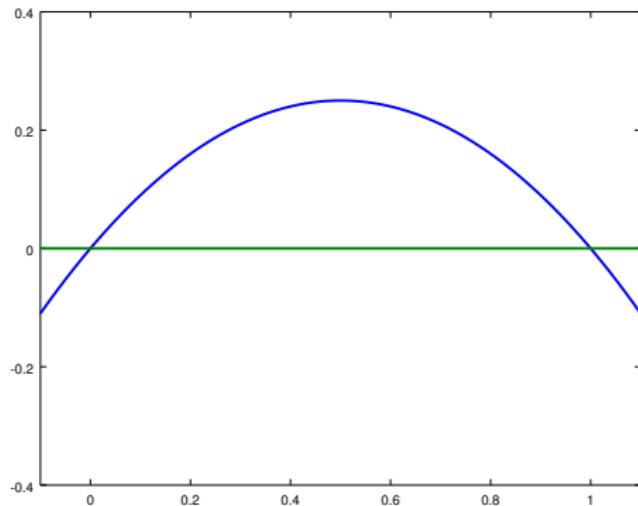
4. Nichtlineare Gleichungssysteme



Falls f mehrere Nullstellen im Intervall besitzt und die Vorzeichenbedingung gilt, so findet das Bisektionsverfahren garantiert eine der Nullstellen, bspw. die mittlere Nullstelle in diesem Beispiel für das Startintervall $[-0.1, 1.1]$. Knobelaufgabe: Für welche Startintervalle werden die anderen Nullstellen gefunden?



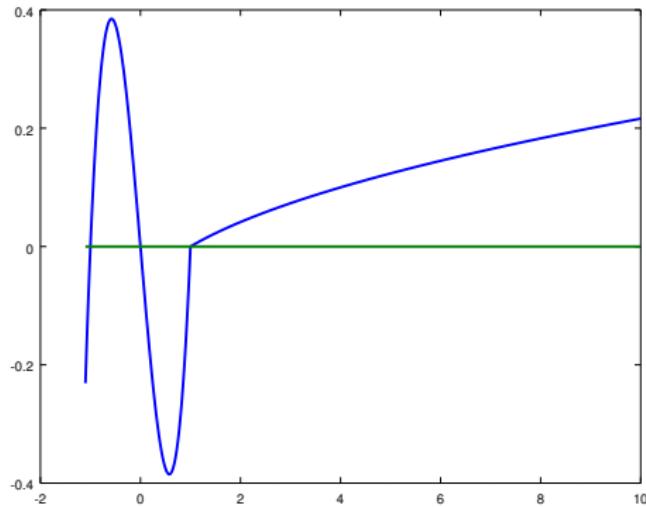
4. Nichtlineare Gleichungssysteme



Falls die Vorzeichenbedingung verletzt ist, kann das Verfahren funktionieren solange im Startintervall eine Nullstelle existiert, hier bspw. für das Startintervall $[-0.1, 1.1]$. Das liegt an der Formulierung: Falls im linken Teilintervall keine Nullstelle liegt, wird immer im rechten Teilintervall weitergesucht.



4. Nichtlineare Gleichungssysteme



Falls die Vorzeichenbedingung verletzt ist, muss das Verfahren nicht funktionieren obwohl im Startintervall eine Nullstelle existiert, hier bspw. für das Startintervall $[-0.5, 10]$. Der Algorithmus wird hier immer (erfolglos) das rechte Teilintervall weiter untersuchen.



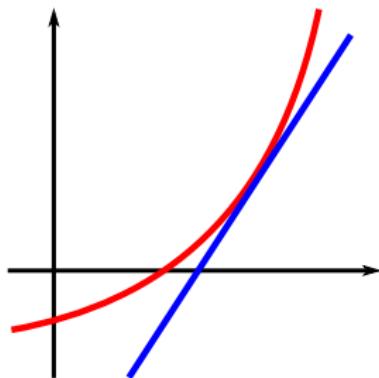
Das Newton-Verfahren in 1D



4. Nichtlineare Gleichungssysteme

Das Bisektionsverfahren nutzt nur Funktionsauswertungen und Vorzeicheninformationen. Wenn wir Ableitungsinformationen hinzuziehen, können wir die Konvergenz beschleunigen.

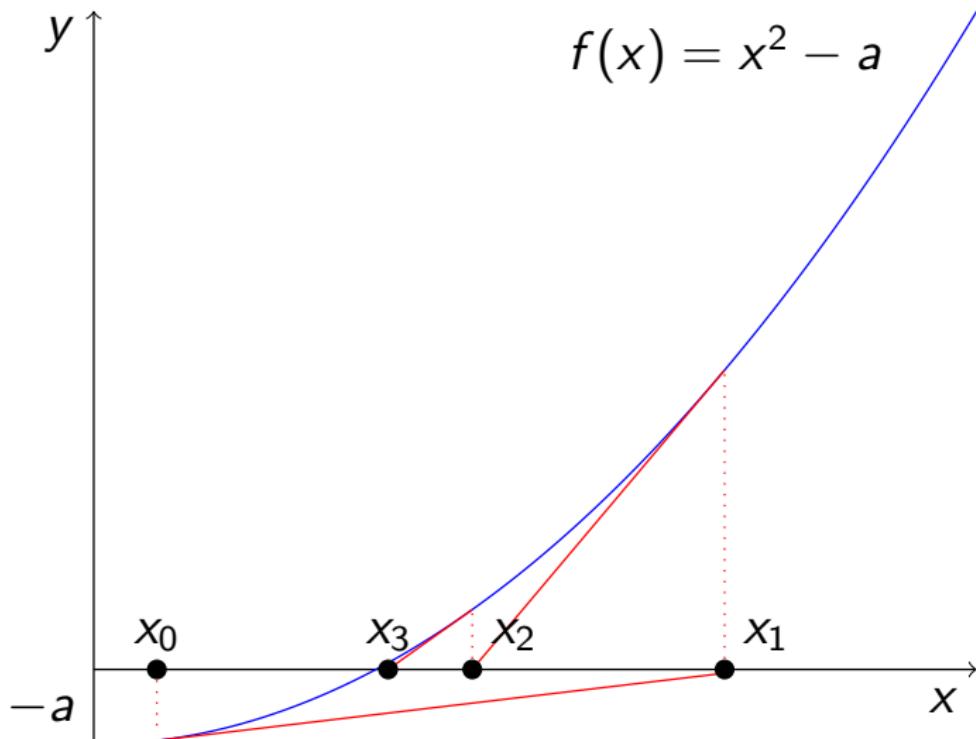
HM12: Der Wert der Ableitung $f'(\bar{x})$ ist gerade die Steigung der Tangente an die Funktion f im Punkt \bar{x} . Damit das Sinn ergibt, müssen wir von f natürlich stetige Differenzierbarkeit fordern.



Anschauliche Idee des Newton-Verfahrens

Falls \bar{x} „nahe an der gesuchten Nullstelle x^* liegt“, so ist die Nullstelle der Tangente an $f(\bar{x})$ eine gute Approximation von x^* .

4. Nichtlineare Gleichungssysteme



Die Idee lässt sich selbstverständlich iterieren.

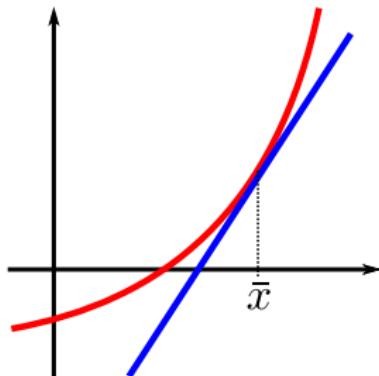


4. Nichtlineare Gleichungssysteme

Die Tangentengleichung im Auswertungspunkt $f(\bar{x})$ ist die Geradengleichung

$$\begin{aligned}T(x) &= f(\bar{x}) + f'(\bar{x})(x - \bar{x}) \\&= f(\bar{x}) + f'(\bar{x})x - f'(\bar{x})\bar{x}.\end{aligned}$$

Der erste Summand ist der Aufpunkt, und $f'(\bar{x})$ ist die Steigung im Punkt \bar{x} . Das rechtfertigt die Sprechweise einer *Linearisierung* im Auswertungspunkt.



Wir setzen diese Darstellung gleich Null:

$$0 = f(\bar{x}) + f'(\bar{x})x - f'(\bar{x})\bar{x}$$

Nullstellen von Tangenten sind immer eindeutig, weil Tangenten lineare Funktionen sind.



4. Nichtlineare Gleichungssysteme

Die Nullstelle können wir explizit ausrechnen, falls $f'(\bar{x}) \neq 0$, d. h. falls \bar{x} kein Sattelpunkt oder nicht bereits die Nullstelle ist:

$$0 = f(\bar{x}) + f'(\bar{x}) y - f'(\bar{x}) \bar{x} \quad \iff \quad x = \bar{x} - \frac{f(\bar{x})}{f'(\bar{x})}$$

Im letzten Schritt übersetzen wir diese Idee in ein Iterationsverfahren, indem wir die Nullstelle der Tangente an f im Punkt \bar{x} , also der alten Iterierten, als nächste Iterierte verwenden:

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})} \tag{4.4}$$

Das ist bereits das Newton-Verfahren in 1D.



4. Nichtlineare Gleichungssysteme

Algorithmus 4.7 : Skalares Newton-Verfahren

input : $f: [a, b] \rightarrow \mathbb{R}$ differenzierbar, Startwert $x^{(0)} \in [a, b]$, $TOL > 0$,
 $k_{\max} \in \mathbb{N}$

output : Approximation x an eine Nullstelle von f

```
1  $k = 0;$ 
2  $x = x^{(0)};$ 
3  $r = f(x);$ 
4 while  $k < k_{\max}$  and  $|r| > TOL$  do
5    $\alpha = f'(x);$       % und Überprüfung ob  $|\alpha|$  „groß genug“
6    $y = r/\alpha;$ 
7    $x = x - y;$       %  $x = x - f(x)/f'(x)$ 
8    $r = f(x);$ 
9    $k = k + 1;$ 
10 end
```



4. Nichtlineare Gleichungssysteme

Fortsetzung Beispiel 4.1 (Quadratwurzel)

Betrachte für $a > 0$ die Funktion $f(x) = x^2 - a$ mit Nullstelle $x^* = \sqrt{a}$. Das Newton-Verfahren für dieses Problem lautet, wenn wir f' analytisch berechnen:

$$\begin{aligned}x^{(k+1)} &= x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})} \\&= x^{(k)} - \frac{(x^{(k)})^2 - a}{2x^{(k)}} \\&= x^{(k)} - \frac{1}{2}x^{(k)} + \frac{a}{2x^{(k)}} \\&= \frac{1}{2} \left(x^{(k)} + \frac{a}{x^{(k)}} \right)\end{aligned}$$

Dieses sogenannte **Heron-Verfahren** wird „im Rechner“ verwendet um \sqrt{a} zu bestimmen, da nur Grundoperationen $+, *, /$ nötig sind.



4. Nichtlineare Gleichungssysteme

Beispiel 4.8 (Vergleich von Newton-Verfahren und Bisektion)

$$f(x) = x^2 - 0.81, \quad [a, b] = [0, 1] \Rightarrow x^* = 0.9$$

k	$x^{(k)}$
0	0.8100000000
1	0.9050000000
2	0.9000138122
3	0.9000000001
4	0.9000000000

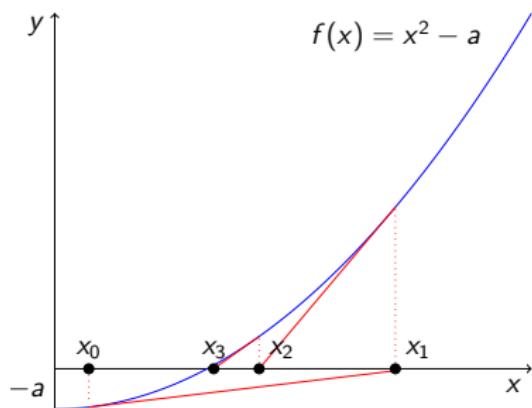
k	$x^{(k)}$	$[a^{(k)}, b^{(k)}]$
0	0.5000000000	[0.0000000000, 1.0000000000]
1	0.7500000000	[0.5000000000, 1.0000000000]
2	0.8750000000	[0.7500000000, 1.0000000000]
3	0.9375000000	[0.8750000000, 1.0000000000]
4	0.9062500000	[0.8750000000, 0.9375000000]
5	0.8906250000	[0.8750000000, 0.9062500000]
6	0.8984375000	[0.8906250000, 0.9062500000]
7	0.9023437500	[0.8984375000, 0.9062500000]
8	0.9003906250	[0.8984375000, 0.9023437500]
9	0.8994140625	[0.8984375000, 0.9003906250]
10	0.8999023438	[0.8994140625, 0.9003906250]
11	0.9001464844	[0.8999023438, 0.9003906250]
12	0.9000244141	[0.8999023438, 0.9001464844]
13	0.8999633789	[0.8999023438, 0.9000244141]
14	0.8999938965	[0.8999633789, 0.9000244141]
15	0.9000091553	[0.8999938965, 0.9000244141]
16	0.9000015259	[0.8999938965, 0.9000091553]
17	0.8999977112	[0.8999938965, 0.9000015259]
18	0.8999996185	[0.8999977112, 0.9000015259]
19	0.9000005722	[0.8999996185, 0.9000005722]
20	0.9000000954	[0.8999996185, 0.9000000954]
21	0.8999998569	[0.8999996185, 0.9000000954]
22	0.8999999762	[0.8999998569, 0.9000000954]
23	0.9000000358	[0.8999999762, 0.9000000954]
24	0.9000000060	[0.8999999762, 0.9000000358]
25	0.8999999911	[0.8999999762, 0.9000000060]

Das Newton-Verfahren konvergiert scheinbar quadratisch schneller als das Bisektionsverfahren.

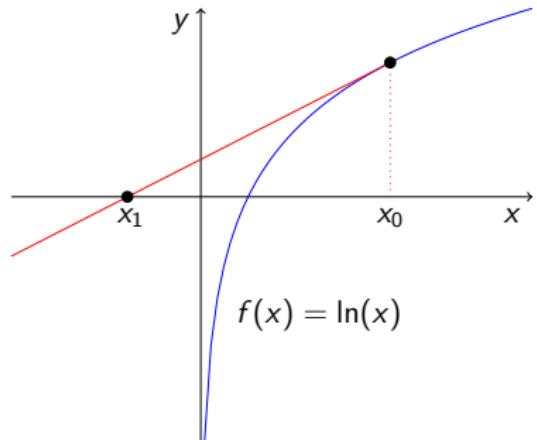


4. Nichtlineare Gleichungssysteme

Das Newton-Verfahren muss nicht für jeden Startwert konvergieren.



Globale Konvergenz für beliebiges $x^{(0)} > 0$.

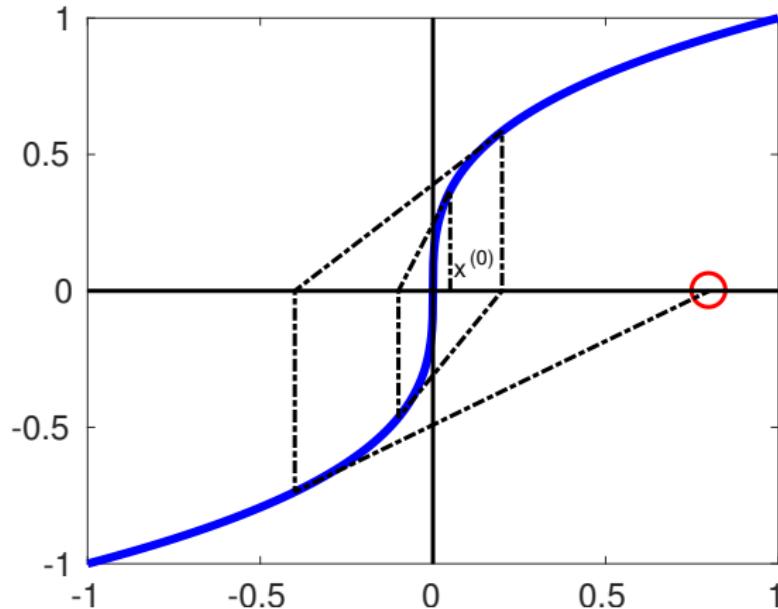


Konvergenz nur für $x^{(0)}$ „nahe“ $x^* = 1$ ($x^{(0)} < 3.5$)



4. Nichtlineare Gleichungssysteme

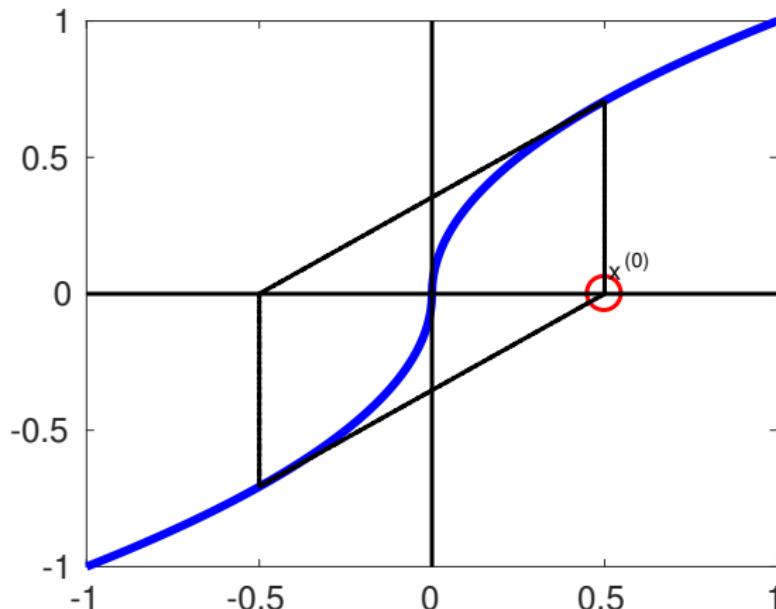
Ein boshaftes Beispiel für die Divergenz des Newton-Verfahrens:





4. Nichtlineare Gleichungssysteme

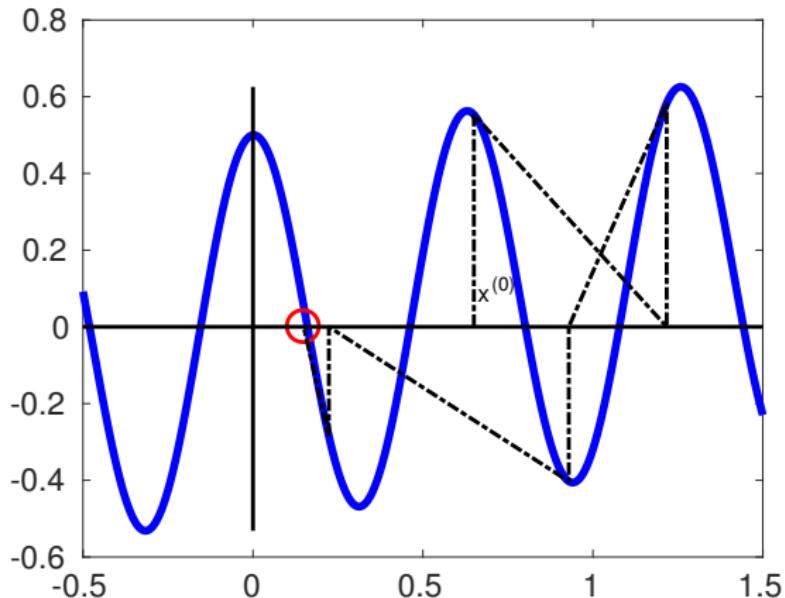
Ein anderes, aber ebenso boshafes Beispiel für die Divergenz des Newton-Verfahrens: Es können Endlosschleifen auftreten.





4. Nichtlineare Gleichungssysteme

Bei mehreren Nullstellen muss das Verfahren nicht gegen diejenige konvergieren, die dem Startwert am nächsten liegt:





4. Nichtlineare Gleichungssysteme

Die Beispiele zeigen:

Globale Konvergenz für jeden Startwert kann man i. A. nicht erwarten. Im übernächsten Abschnitt geben wir für den allgemeinen Fall $N \geq 1$ Kriterien an, wann **lokale Konvergenz** vorliegt, d. h. für $x^{(0)}$ „nah genug“ an x^* .

In der Praxis haben wir zwei Möglichkeiten:

- Neustart des Verfahrens mit einem anderen Startwert bei Divergenz, hierzu muss der Algorithmus um eine Divergenz-Detektion erweitert werden.
- Nutzung von Varianten des Newton-Verfahrens, s. übernächster Abschnitt.

Trotz der Konvergenzprobleme lohnt sich das Newton-Verfahren (fast) immer wegen der enorm schnellen Konvergenz.



Newton-Verfahren für vektorielle Nullstellenprobleme



4. Nichtlineare Gleichungssysteme

Erinnerung H123: Für eine Funktion $\mathbf{f}: \mathbb{R}^N \rightarrow \mathbb{R}^N$ ist die **Jacobi-Matrix**

$$\mathbf{J}_{\mathbf{f}}(\mathbf{x}) = \begin{pmatrix} \frac{\partial}{\partial x_1} f_1(\mathbf{x}) & \dots & \frac{\partial}{\partial x_N} f_1(\mathbf{x}) \\ \vdots & & \vdots \\ \frac{\partial}{\partial x_1} f_N(\mathbf{x}) & \dots & \frac{\partial}{\partial x_N} f_N(\mathbf{x}) \end{pmatrix}$$

die Matrix aller partiellen Ableitungen nach allen Variablen.

Im Mehrdimensionalen entspricht also die Auswertung von $f'(\bar{\mathbf{x}})$ in einem Punkt $\bar{\mathbf{x}}$ der Auswertung der Jacobi-Matrix von \mathbf{f} in einem Punkt $\bar{\mathbf{x}}$.



4. Nichtlineare Gleichungssysteme

Damit erhalten wir rein formell aus der 1D-Iterationsvorschrift

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

das **mehrdimensionale Newton-Verfahren**:

Definition 4.9 (Newton-Verfahren für $N > 1$)

Sei $\mathbf{x}^{(0)} \in B$ und $\mathbf{f} : B \rightarrow \mathbb{R}^N$ differenzierbar. Das durch die Vorschrift

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\mathbf{J}_f(\mathbf{x}^{(k)}))^{-1} \mathbf{f}(\mathbf{x}^{(k)}) \quad (k \in \mathbb{N}_0)$$

gegebene Iterationsverfahren heißt **Newton-Verfahren** (für Systeme, vektorielles Newton-Verfahren) zum Startwert $\mathbf{x}^{(0)}$.



4. Nichtlineare Gleichungssysteme

Wir betrachten die Verfahrensvorschrift etwas genauer:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \underbrace{(\mathbf{J}_f(\mathbf{x}^{(k)}))^{-1} \mathbf{f}(\mathbf{x}^{(k)})}_{(k \in \mathbb{N}_0)}$$

Wir benötigen $\det(\mathbf{J}_f(\mathbf{x}^{(k)})) \neq 0$ und $\mathbf{x}^{(k)} \in B$ für alle k , damit alle Iterierten wohldefiniert sind.

Die Auswertung der Jacobi-Matrix an einer Iterierten $\mathbf{x}^{(k)}$ ergibt eine Matrix \mathbf{A} mit reellen Einträgen. Anstatt diese Matrix zu invertieren und \mathbf{A}^{-1} mit $\mathbf{x}^{(k)}$ zu multiplizieren, lösen wir wieder ein LGS mit der Koeffizientenmatrix \mathbf{A} und der rechten Seite $\mathbf{f}(\mathbf{x}^{(k)})$, vgl. die Wielandt-Iteration aus VL 3.

Auch das mehrdimensionale Newton-Verfahren kann nicht für nicht differenzierbare Funktionen \mathbf{f} angewendet werden.



4. Nichtlineare Gleichungssysteme

Algorithmus 4.10 : Vektorielles Newton-Verfahren

input : $f: B \rightarrow \mathbb{R}^N$ differenzierbar, Startwert $x^{(0)} \in B$, $TOL > 0$, $k_{\max} \in \mathbb{N}$
output : Approximation x an eine Nullstelle von f

```
1  $x = x^{(0)}$ ;  
2  $k = 0$ ;  
3 berechne  $r = f(x)$ ;  
4 while  $k < k_{\max}$  and  $\|r\|_2 > TOL$  do  
5    $A = J_f(x)$ ;           % Auswertung Jacobi-Matrix  
6   löse  $Ay = r$ ;         %  $y = (J_f(x))^{-1}f(x)$   
7    $x = x - y$ ;          %  $x = x - (J_f(x))^{-1}f(x)$   
8    $r = f(x)$ ;  
9    $k = k + 1$ ;  
10 end
```



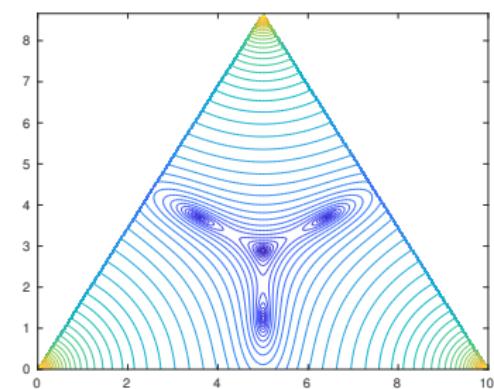
4. Nichtlineare Gleichungssysteme

Beispiel 4.11 (Lösung des NSA-Problems, Bsp. 4.2)

Wir betrachten das vereinfachte Problem mit identischen Massen, und drei Himmelskörpern an den Ecken eines gleichseitigen Dreiecks, d. h. an den Positionen $(0, 0, 0)^T$, $(10, 0, 0)^T$ und $(5, \sqrt{75}, 0)^T$. Eine exakte Lösung ist dann der Schwerpunkt des Dreiecks, $(5, \sqrt{75}/3, 0)^T$. Das Problem besitzt allerdings viele Sattelpunkte, vergleiche die Konturlinien von $\|\mathbf{F}\|_2$.

Startwert $(1, 2, 3)^T$: Konvergenz in 16 Schritten zum Sattelpunkt $(3587, 4.117, -10260)^T$.

Startwert nahe am Schwerpunkt: Konvergenz in 3 Schritten zur exakten Lösung.





Konvergenz und Varianten des Newton-Verfahrens



4. Nichtlineare Gleichungssysteme

Die bisherigen Experimente zeigen: Das Newton-Verfahren konvergiert nicht für alle Startwerte $x^{(0)}$ gegen die „gewünschte“ Lösung x^* . Der Startwert $x^{(0)}$ muss „in der Nähe“ der Lösung x^* liegen. Man spricht von **lokaler Konvergenz**.

Ein mögliches Konvergenzkriterium liefert der folgende Satz:

Satz 4.12 (Lokale Konvergenz des Newton-Verfahrens)

Sei $f \in C^2(B, \mathbb{R}^N)$ und $x^* \in B$ eine Nullstelle von f mit

$$\det(J_f(x^*)) \neq 0. \quad (4.5)$$

Dann ist das Newton-Verfahren lokal konvergent für jeden Startwert aus einer (eventuell kleinen) Umgebung von x^* .

Dieses Kriterium ist in der Praxis schwer zu überprüfen, weil wir die Nullstelle x^* nicht kennen. Der Satz liefert trotzdem die Rechtfertigung: Wenn wir nah genug an einer Nullstelle starten („gut genug raten“), dann konvergiert das Newton-Verfahren gegen die Nullstelle.



4. Nichtlineare Gleichungssysteme

Die Auswertung von J_f und die Lösung eines LGS mit unterschiedlicher Matrix in jedem Iterationsschritt ist aufwändig. Wir betrachten nun einige Vereinfachungen, die auf der Approximation $J_f(x^{(k)}) \approx A^{(k)}$ basieren:

Definition 4.13 (Quasi-Newton-Verfahren)

Sei $x^{(0)} \in B$ und $\{A^{(k)}\}_{k \in \mathbb{N}_0} \subseteq \mathbb{R}^{N \times N}$, $\det(A^{(k)}) \neq 0$. Dann heißt das durch

$$x^{(k+1)} = x^{(k)} - (A^{(k)})^{-1} f(x^{(k)}) \quad (k \in \mathbb{N}_0)$$

gegebene Iterationsverfahren **Quasi-Newton Verfahren**.

Das Quasi-Newton Verfahren hat nur Sinn, wenn $A^{(k)}$ „leicht“ invertierbar ist, und natürlich etwas mit $J_f(x^{(k)})$ zu tun hat.

Quasi-Newton Verfahren konvergieren (meist) langsamer als das volle Newton-Verfahren mit $A^{(k)} = J_f(x^{(k)})$.



4. Nichtlineare Gleichungssysteme

Beispiel 4.14 (Vereinfachtes Newton-Verfahren)

Das Quasi-Newton Verfahren mit der konstanten Wahl

$$\mathbf{A} := \mathbf{A}^{(k)} := \mathbf{J}_f(\mathbf{x}^{(0)})$$

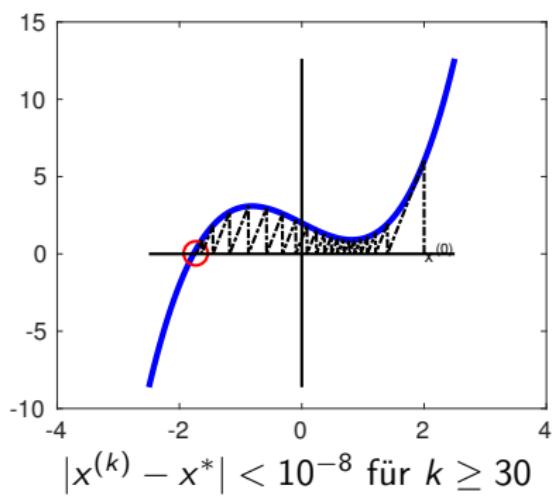
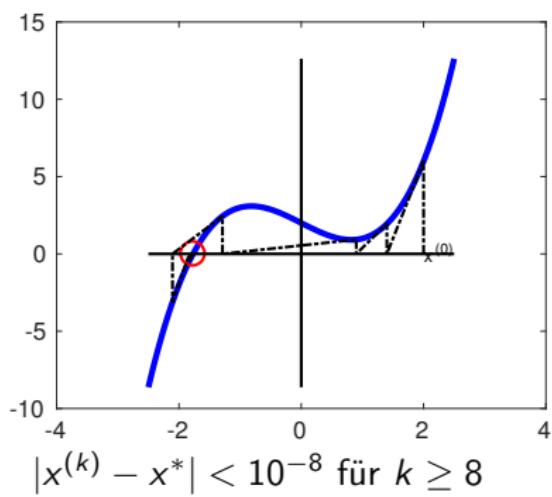
heißt **vereinfachtes Newton-Verfahren**.

Hier ist es vorteilhaft, einmal eine LR-Zerlegung von \mathbf{A} zu bestimmen, um in jeder Iteration das LGS $\mathbf{A}\mathbf{y} = \mathbf{f}(\mathbf{x}^{(k)})$ einfach durch Vorwärts- und Rückwärtseinsetzen lösen zu können.



4. Nichtlineare Gleichungssysteme

Wir betrachten als Beispiel den Fall $N = 1$ und die Funktion $f(x) = x^3 - 2x + 2$. Als Startwert verwenden wir $x^{(0)} = 2$. Gestrichelte Linien entsprechen Iterierten:





4. Nichtlineare Gleichungssysteme

Beispiel 4.15 (Sekantenverfahren)

Für $N = 1$ heißt das Quasi-Newton-Verfahren mit der Wahl

$$\mathbf{A}^{(k)} = \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}} \in \mathbb{R}^{1 \times 1}$$

Sekantenverfahren, es lautet:

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - \left(\frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}} \right)^{-1} f(x^{(k)}) \\ &= x^{(k)} - \frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})} f(x^{(k)}) \end{aligned}$$

Anschaulich ist $x^{(k+1)}$ die Nullstelle der Sekante durch $(x^{(k)}, f(x^{(k)}))$ und $(x^{(k-1)}, f(x^{(k-1)}))$. Das Verfahren erfordert zwei Startwerte $x^{(0)}, x^{(1)}$, aber keine Ableitungen.



4. Nichtlineare Gleichungssysteme

Das Newton-Verfahren ist nur lokal konvergent und benötigt einen Startwert $\mathbf{x}^{(0)}$ nahe einer Nullstelle \mathbf{x}^* . Ein Ausweg ist Dämpfung, vgl. SOR-Verfahren in VL 2:

Definition 4.16 (Gedämpftes Newton-Verfahren)

Schrittweitensteuerung mit „Standardmonotonietest“:

- ① Bestimme die „Suchrichtung“ $\mathbf{y}^{(k)} = \mathbf{J}_f(\mathbf{x}^{(k)})^{-1} \mathbf{f}(\mathbf{x}^{(k)})$ wie im normalen Newton-Verfahren.
- ② Suche iterativ das größte $s_k \in \{1, \frac{1}{2}, \frac{1}{4}, \dots\}$, so dass

$$\|\mathbf{f}(\mathbf{x}^{(k)} - s_k \mathbf{y}^{(k)})\|_2 \leq \left(1 - \frac{s_k}{2}\right) \|\mathbf{f}(\mathbf{x}^{(k)})\|_2.$$

- ③ Verwende $\mathbf{x}^{(k)} - s_k \mathbf{y}^{(k)}$ als nächste Iterierte.

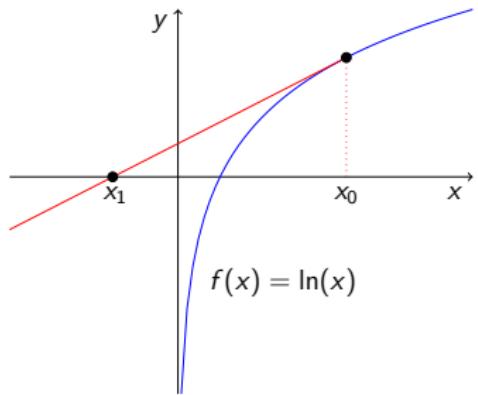
Der Konvergenzbereich wird i. A. vergrößert. Solange $s_k \ll 1$ ist, konvergiert das Verfahren relativ langsam. Ist aber $\mathbf{x}^{(k)}$ nah an \mathbf{x}^* , ergibt das Verfahren auch Werte s_k nahe 1.



4. Nichtlineare Gleichungssysteme

Einfluss der Dämpfung für $f(x) = \ln(x)$:

ungedämpft		gedämpft
0	3.50000000000	0
1	-0.8846703897	1
		2
		3
		4
		5
		6
		7



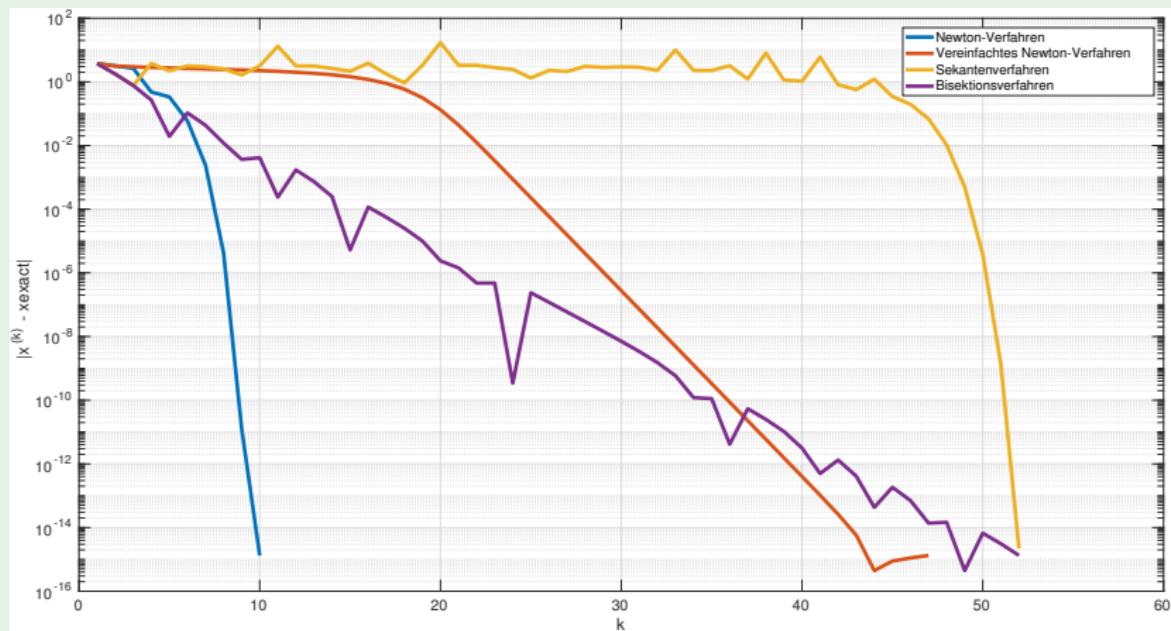
Das ungedämpfte Verfahren konvergiert nur für $x^{(0)}$ „nahe“ $x^* = 1$, konkret für $x^{(0)} < 3.5$. Das gedämpfte Verfahren konvergiert viel langsamer, aber dafür auch für sehr große $x^{(0)} \approx 100$.



4. Nichtlineare Gleichungssysteme

Beispiel 4.17 (Vergleich des Konvergenzverhaltens)

$$f(x) = x^3 - 2x + 2, \quad x^{(0)} = 2, \quad a = -2, b = 2$$

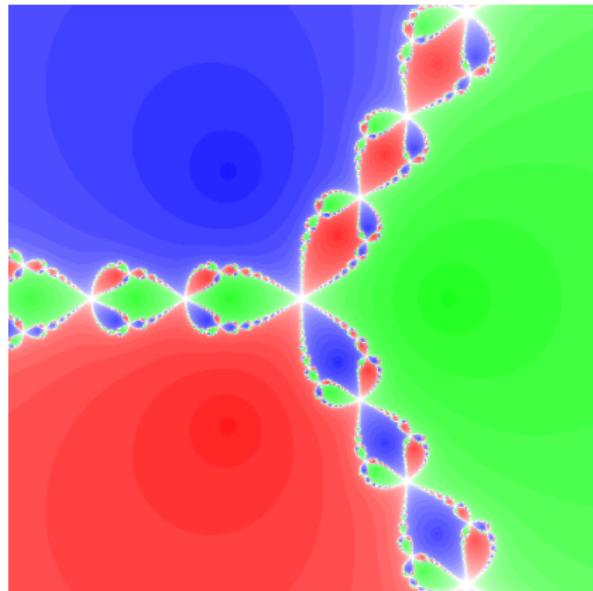




4. Nichtlineare Gleichungssysteme

Newton-Fraktale:

- Betrachte komplexwertige Funktion $f : \mathbb{C} \rightarrow \mathbb{C}$ und (skalares) Newton-Verfahren.
- Für alle $x^{(0)}$ in einem Rechteck der komplexen Ebene bestimme die Anzahl der Iterationen k und die Nullstelle x^* , welche gefunden wird.
- Färbe den Punkt $x^{(0)}$ entsprechend einer Farbe, welche zur Nullstelle korrespondiert, und einer Helligkeit, welcher der Anzahl der Iterationen k entspricht.
- Die Menge der Punkte, für die das Newton-Verfahren nicht konvergiert (weiße Menge im Bild) ist das sogenannte Newton-Fraktal.



$$f(x) = x^3 - 1 \text{ mit Nullstellen}$$

$$x^* = 1, e^{\frac{2}{3}\pi i}, e^{\frac{4}{3}\pi i}.$$



Zusammenfassung

- Die Lösung nichtlinearer Gleichungssysteme kann als Nullstellenproblem formuliert werden.
- Falls f nicht differenzierbar ist, ist das Bisektionsverfahren für $N = 1$ eine Methode zur Nullstellenbestimmung. Es konvergiert global.
- Falls f differenzierbar ist, ist das Newton-Verfahren die geeignete Methode zur Approximation von Nullstellen von skalaren oder vektorwertigen Funktionen. Das Newton-Verfahren ist nur lokal konvergent, d. h. Startwerte müssen geeignet gewählt werden
- Quasi-Newton-Verfahren wie das vereinfachte Newton-Verfahren oder das Sekantenverfahren reduzieren den Aufwand in jeder Iteration, konvergieren dafür aber langsamer.
- Dämpfung kann durchgeführt werden, um den Konvergenzbereich zu vergrößern. Dies reduziert jedoch die Konvergenzgeschwindigkeit.



4. Nichtlineare Gleichungssysteme

Zusammenfassung – Formeln

$$\text{Newton-Verfahren } (N = 1) : \quad x^{(k+1)} = x^{(k)} - (f'(x^{(k)}))^{-1} f(x^{(k)})$$

$$\text{Newton-Verfahren } (N > 1) : \quad \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\mathbf{J}_f(\mathbf{x}^{(k)}))^{-1} \mathbf{f}(\mathbf{x}^{(k)})$$

$$\text{Quasi-Newton-Verfahren :} \quad \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\mathbf{A}^{(k)})^{-1} \mathbf{f}(\mathbf{x}^{(k)})$$

$$\text{Wählen für } \mathbf{A}^{(k)} : \quad \mathbf{A}^{(k)} = \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}} \quad (N = 1, \text{ Sekantenverfahren})$$

$$\mathbf{A}^{(k)} = \mathbf{J}_f(\mathbf{x}^{(0)})$$

$(N \geq 1, \text{ vereinfachtes Newton-Verf.})$



Hausaufgaben

- Einüben des Aufstellens von Nullstellenproblemen
- Einüben der unterschiedlichen Iterationsvorschriften
- Für nächste Woche: Skalarprodukte wiederholen, Orthogonalitätsbegriff
- Die erste Hälfte des nächsten VÜ-Rechenzettels kann bereits bearbeitet werden.



Beispieldaufgaben



Verständnisfragen zum Newton-Verfahren

Sei f eine lineare Funktion mit einer Nullstelle. Konvergiert das Newton-Verfahren dann global? Wie viele Schritte werden benötigt, unabhängig vom Startwert?



4. Nichtlineare Gleichungssysteme

Lösungshinweise: Lineare Funktionen sind ihre eigenen Tangenten.

Ergebnis: Das Newton-Verfahren konvergiert global in genau einem Schritt.



4. Nichtlineare Gleichungssysteme

Newton-Verfahren in 1D

Gegeben ist die Funktion

$$f(x) = 2x^5 - 2x^3 + 6x - 2.$$

Geben Sie die Iterationsvorschrift für das Newtonverfahren an, und berechnen Sie die erste Iterierte zum Startwert $x^{(0)} = -1$.



4. Nichtlineare Gleichungssysteme

Lösungshinweise: Hier muss man einfach das Newton-Verfahren in 1D hinschreiben, und dann die Funktion f ableiten. Der Rest ist reines Einsetzen.

Ergebnis:

$$\begin{aligned}x^{(k+1)} &= x^{(k)} - \frac{2x^{(k)5} - 2x^{(k)3} + 6x^{(k)} - 2}{10x^{(k)4} - 6x^{(k)2} + 6} \\x^{(1)} &= -\frac{1}{5}\end{aligned}$$



4. Nichtlineare Gleichungssysteme

Newton-Verfahren in 2D

Gesucht ist eine Lösung $(x, y)^T \in \mathbb{R}^2$ für das nichtlineare Gleichungssystem

$$y = -e^x \sin(2y) \quad e^y = x - 2x^2.$$

Formulieren Sie die Lösung dieses Systems als Nullstellenproblem für eine Funktion $f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$. Berechnen Sie dann zwei Iterationen mit dem vereinfachten Newton-Verfahren zum Startvektor $(x^{(0)}, y^{(0)})^T = (0, 0)^T$.



4. Nichtlineare Gleichungssysteme

Lösungshinweise: Das Nullstellenproblem erhalten wir durch Nullsetzen der beiden Gleichungen. Für das (vereinfachte) Newton-Verfahren müssen wir dann die Jacobi-Matrix der Funktion \mathbf{f} bestimmen, d. h. wir müssen (partiell) ableiten. Beim vereinfachten Newton-Verfahren reicht es dann, die Matrix nur im Startvektor auszuwerten. Danach kann das Verfahren durchgeführt werden.

Ergebnis:

$$\mathbf{f}(x, y) = \begin{pmatrix} y + e^x \sin(2y) \\ e^y + 2x^2 - x \end{pmatrix}$$

$$\mathbf{J}_f(x, y) = \begin{pmatrix} \sin(2y)e^x & 1 + 2\cos(2y)e^x \\ 4x - 1 & e^y \end{pmatrix}$$

$$\begin{pmatrix} x^{(1)} \\ y^{(1)} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} x^{(2)} \\ y^{(2)} \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$$



5. Optimierung, LGS und das CG-Verfahren



Aleksey Nikolaevich Krylov

15.08.1863 — 26.10.1945



Inhalte und Ziele dieser Vorlesungseinheit

- Vorstellung ausgewählter Optimierungsprobleme und -Verfahren
- Tieferes Verständnis des Newton-Verfahrens aus VL 4 im Kontext von Optimierungsproblemen
- Erster Eindruck, wie und warum numerische Verfahren problemübergreifend funktionieren, und wie Ideen für ein Problem für ein gänzlich anderes Problem zu besseren Verfahren führen



Motivation und Anwendungsbereiche



Wir wollen den Widerstandsbeiwert c_w als Funktion der Geometrie optimieren, um so den Kraftstoffverbrauch zu reduzieren.



Eine mögliche Vorgehensweise bei der Simulation ist:

- Beschreibe die Fahrzeug-Geometrie durch Parameter $\boldsymbol{p} \in \mathbb{R}^{N \times 3}$ (z. B. Koordinaten von Punkten auf Oberfläche und Spline-Interpolation, VL 6).
- Berechne das Strömungsfeld um das Fahrzeug (z. B. mittels FEM, VL 10)
- Berechne den Widerstandsbeiwert c_w durch Integration des Strömungsfeldes über die Fahrzeug-Kontur (z. B. Quadratur, VL 7)



Beispiel 5.1 (Optimales Design)

Das Problem des optimalen Designs lautet dann: Finde $\mathbf{p}^* \in \mathbb{R}^{N \times 3}$, also eine Parametrisierung der Fahrzeug-Oberfläche, so dass

$$\mathbf{p}^* \text{ Minimierer von } c_w(\mathbf{p})$$

ist, über alle $\mathbf{p} \in \mathbb{R}^{N \times 3}$. Wir schreiben auch kurz

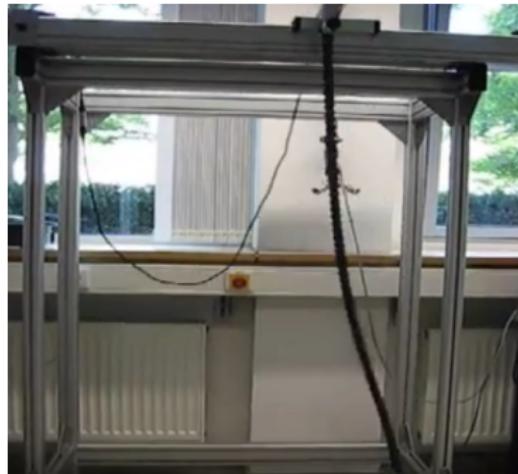
$$\mathbf{p}^* = \arg \min_{\mathbf{p} \in \mathbb{R}^{N \times 3}} c_w(\mathbf{p}).$$

Für die Auswertung der zu optimierenden Funktion müssen wir hier i. W. ein System nichtlinearer PDEs (ein Strömungsproblem) lösen und danach ein Oberflächenintegral berechnen, das ist (schwäbisch) „nicht unkompliziert“.



5. Optimierung, LGS und das CG-Verfahren

Als zweites Beispiel betrachten wir die optimale Steuerung einer Laufkatze, die prototypisch für Kräne, automatisierte Fertigungssysteme etc. ist:

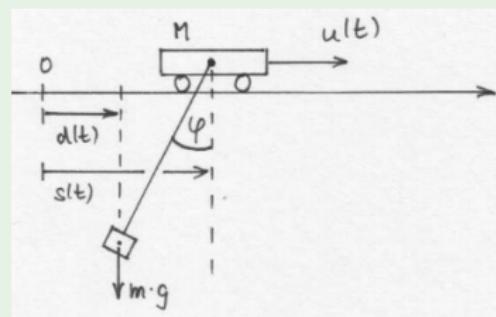


https://www.youtube.com/watch?v=BjNc_UnHN2w



Beispiel 5.2 (Optimalsteuerung einer Laufkatze)

Das Problem kann beschrieben werden durch ein System gewöhnlicher Differentialgleichungen. Insbesondere ist $u(t)$ die **Steuerung**, mit der wir das Verhalten des Systems beeinflussen können.



Gesucht ist eine Steuerung $u(t)$, so dass die Laufkatze möglichst schnell fährt, die Last die Laufkatze in Fahrtrichtung nicht überholt, die Last eine linke und rechte maximale Position nicht überschreitet, und in einem festen Punkt zum Stillstand kommt.

Solche Optimalsteuerungsprobleme treten nicht nur in der Regelungstechnik häufig auf.



Wir betrachten ein LGS $\mathbf{A}\mathbf{x} = \mathbf{b}$ mit $\mathbf{b} \in \mathbb{R}^N$ und $\mathbf{A} \in \mathbb{R}^{N \times N}$, wobei \mathbf{A} dünn besetzt sein soll und N riesig. Solche LGS treten insbesondere bei der Behandlung partieller Differentialgleichungen auf, vgl. ab VL 9.

Die Lösung mit einem direkten Verfahren aus VL 1 ist also ausgeschlossen. Wir wissen aus VL 2 und den Programmierübungen:

- Die Jacobi- und Gauß-Seidel-Verfahren konvergieren nur sehr langsam.
- Das SOR-Verfahren ist nur bei guter Wahl des Relaxationsparameters ω schnell, wir können ω aber i. A. nicht a priori optimal bestimmen.

Erinnerung: Das euklidische Skalarprodukt von $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ schreiben wir als $\mathbf{x}^T \mathbf{y} \in \mathbb{R}$.



Beispiel 5.3 (LGS als Minimierungsproblem)

Für symmetrische und positiv definite Matrizen \mathbf{A} , d. h.

$$\mathbf{A} = \mathbf{A}^T \quad \text{und} \quad \mathbf{x}^T \mathbf{A} \mathbf{x} > 0 \text{ für alle } \mathbf{x} \in \mathbb{R}^N \setminus \{\mathbf{0}\},$$

zeigen wir in dieser Vorlesung:

$$\mathbf{x}^* \text{ löst } \mathbf{A} \mathbf{x} = \mathbf{b}$$

genau dann wenn

$$\mathbf{x}^* \text{ Minimierer von } f(\mathbf{x}) := \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b}^T \mathbf{x} \text{ ist.}$$

Wir schreiben wieder äquivalent

$$\text{finde } \mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x})$$

und nennen \mathbf{x}^* **Minimierer** und f **Zielfunktion**.



$$x^* \text{ löst } Ax = b \quad \Leftrightarrow \quad x^* = \arg \min_{x \in \mathbb{R}^N} f(x)$$

Verfahren zur Lösung gewisser Optimierungsproblemen können also als LGS-Löser eingesetzt werden. Weil Löser für Optimierungsprobleme spezielle Eigenschaften von A (hier SPD, Symmetrie und positive Definitheit) ausnutzen, können sie viel effizienter sein als die bisher bekannten iterativen Verfahren, deren Konvergenz nur von spektralen Eigenschaften der Iterationsmatrix (wie $\|\mathbf{T}_{\text{ESV}}\|_2 < 1$) abhängt.

Die Betrachtung von Optimierungsverfahren führt also „nebenbei“ auf eine neue Klasse von LGS-Lösern.



Grundlagen



Definition 5.4 (Allgemeines Optimierungsproblem)

Sei $D \subseteq \mathbb{R}^N$ abgeschlossen und $f: D \rightarrow \mathbb{R}$. Gesucht ist $\mathbf{x}^* \in D$, welches f minimiert, d. h.

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in D} f(\mathbf{x})$$

Wir nennen f **Zielfunktion**, \mathbf{x}^* **Minimierer**, $f(\mathbf{x}^*)$ **Optimum** und D **Menge zulässiger Punkte**. Die Definition erfolgt analog für $\mathbf{f}: D \rightarrow \mathbb{R}^N$.

Es reicht, die Minimierung zu betrachten, da sich jedes Maximierungsproblem umschreiben lässt in ein Minimierungsproblem:

$$\max_{\mathbf{x} \in D} f(\mathbf{x}) = - \min_{\mathbf{x} \in D} (-f(\mathbf{x}))$$

Das sollte man sich bei der Nachbereitung klarmachen.

5. Optimierung, LGS und das CG-Verfahren



Falls $D \subset \mathbb{R}^N$, falls also nicht alle Punkte in \mathbb{R}^N zulässig sind, spricht man von „bedingter/restringierter Optimierung“ mit der **Nebenbedingung** $x \in D$. Nebenbedingungen können als Gleichungen oder Ungleichungen vorkommen, bspw. $\|x\| \neq 0$ oder $x > 0$.

Wir beschränken uns heute auf unrestringierte Optimierung, d. h. $D = \mathbb{R}^N$. Damit decken wir die beiden ersten Beispiele aus der Motivation nicht vollständig ab.

Restringierte Probleme lassen sich aber immer als unrestringierte Probleme formulieren durch Änderung der Zielfunktion. Deshalb sind Optimierungsverfahren für unrestringierte Probleme eine wichtige Teilkomponente bei der Optimierung unter Nebenbedingungen.



Bemerkung (Grundvoraussetzung)

Wir gehen im Folgenden immer davon aus, dass das betrachtete Optimierungsproblem eine eindeutige Lösung besitzt, die wir numerisch approximieren.

Diese Grundvoraussetzung ist recht stark in der Praxis.

Falls f ein nicht eindeutiges Optimum besitzt (oder die Existenz unklar ist), liefern die Verfahren aus der heutigen Vorlesung häufig „lokale Optima“, d. h. sie funktionieren trotzdem mit u. U. bedingt hilfreichen Ergebnissen. Weil alle Verfahren iterativ sind, kann oft durch Variation des Startwerts ein besseres Ergebnis gefunden werden.

Falls D eine diskrete zulässige Menge ist (z. B. $\mathbf{x} \in \mathbb{Z}^N$), ist die Optimierung wesentlich schwieriger. Man nennt dies „diskrete Optimierung“ oder „kombinatorische Optimierung“ im Gegensatz zur hier betrachteten **kontinuierlichen Optimierung**, d. h. $D \subseteq \mathbb{R}^N$.



Das Newton-Verfahren zur Optimierung



5. Optimierung, LGS und das CG-Verfahren

Wir erinnern uns an die HM23 und an VL 4:

Definition 5.5 (Gradient, Jacobi-Matrix, Hesse-Matrix)

Sei $D \subset \mathbb{R}^N$ offen. Für eine einmal stetig differenzierbare Funktion $f: D \rightarrow \mathbb{R}$ ist der **Gradient** der Vektor der ersten partiellen Ableitungen entlang der Koordinatenrichtungen:

$$\text{grad } f(\mathbf{x}) := \left(\frac{\partial}{\partial x_1} f(\mathbf{x}), \dots, \frac{\partial}{\partial x_N} f(\mathbf{x}) \right)^T$$

Die **Hesse-Matrix** ist die Matrix aller zweiten partiellen Ableitungen, sofern f zweimal stetig diffenziertbar ist:

$$\mathbf{H}_f(\mathbf{x}) := \left(\frac{\partial}{\partial x_i} \frac{\partial}{\partial x_j} f(\mathbf{x}) \right)_{i,j=1}^N \in \mathbb{R}^{N \times N}$$

Für eine vektorwertige stetig differenzierbare Funktion $\mathbf{f}: D \rightarrow \mathbb{R}^N$ ist die **Jacobi-Matrix** die Matrix der ersten partiellen Ableitungen aller Variablen in alle Richtungen:

$$\mathbf{J}_{\mathbf{f}}(\mathbf{x}) := \left(\frac{\partial}{\partial x_j} f_i(\mathbf{x}) \right)_{i,j=1}^N \in \mathbb{R}^{N \times N}$$

Die Hessematrix ist die Jacobi-Matrix des Gradienten einer skalarwertigen Funktion.



Ausgangspunkt unserer Überlegungen ist das allgemeine Optimierungsproblem aus Definition 5.4, d. h. wir suchen $\mathbf{x}^* = \arg \min_{\mathbf{x} \in D} f(\mathbf{x})$ für $f: D \rightarrow \mathbb{R}$ im unrestringierten Fall $D = \mathbb{R}^N$. Aus der HM23 wissen wir:

Satz 5.6 (Notwendige Bedingung für ein Optimum)

Falls f in $\mathbf{x}^* \in D$ ein Minimum oder Maximum besitzt, dann ist $\text{grad } f(\mathbf{x}^*) = \mathbf{0}$.

Mit unserer Grundannahme der eindeutigen Existenz eines Minimums, und bei Ausschluss von bspw. Sattelpunkten \mathbf{y} mit $\text{grad } f(\mathbf{y}) = \mathbf{0}$ gilt:

$$\mathbf{x}^* \text{ minimiert } f(\mathbf{x}) \quad \Leftrightarrow \quad \text{grad } f(\mathbf{x}^*) = \mathbf{0}$$

Andernfalls berechnen die folgenden Verfahren u.U. nur ein lokales Optimum.

5. Optimierung, LGS und das CG-Verfahren

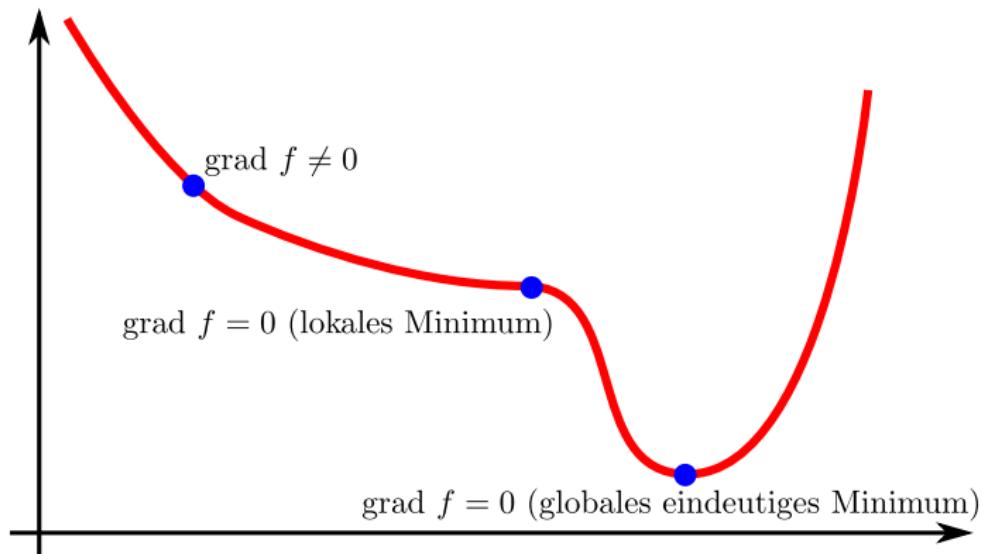


Illustration der Grundvoraussetzung und der notwendigen Bedingung



Die Äquivalenz

$$\mathbf{x}^* \text{ minimiert } f(\mathbf{x}) \quad \Leftrightarrow \quad \text{grad } f(\mathbf{x}^*) = \mathbf{0}$$

können wir nutzen, um das Newton-Verfahren aus VL 4 als Optimierungsverfahren zu interpretieren. Dazu definieren wir

$$\mathbf{g}(\mathbf{x}) := \text{grad } f(\mathbf{x})$$

und müssen „nur noch“ das (vektorielle) Nullstellenproblem $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ lösen.

Für die Anwendung des Newton-Verfahrens auf $f(\mathbf{x})$ benötigen wir die Auswertung der Jacobi-Matrix von \mathbf{g} , also der Hesse-Matrix von f in einem Punkt $\mathbf{x} \in D$, vgl. Definition 5.5 und Definition 4.9. Wir müssen also voraussetzen, dass die Zielfunktion f zweimal differenzierbar ist.



Nach kurzer Erinnerung an VL 4 erhalten wir sofort:

Definition 5.7 (Newton-Verfahren zur unrestringierten Optimierung)

Die Zielfunktion $f: D \rightarrow \mathbb{R}$ eines unrestringierten Optimierungsproblems

$$\text{finde } \mathbf{x}^* \text{ mit } \mathbf{x}^* = \arg \min_{\mathbf{x} \in D} f(\mathbf{x})$$

sei zweimal stetig differenzierbar, d. h. die Hesse-Matrix von f existiere für jeden Punkt $\mathbf{x} \in D = \mathbb{R}^N$.

Das **Newton-Verfahren zur Optimierung** lautet für einen Startwert $\mathbf{x}^{(0)} \in \mathbb{R}^N$:

$$\begin{aligned}\mathbf{x}^{(k+1)} &:= \mathbf{x}^{(k)} - \left(\mathbf{J}_g(\mathbf{x}^{(k)}) \right)^{-1} \mathbf{g}(\mathbf{x}^{(k)}) \quad k = 1, 2, \dots \\ &= \mathbf{x}^{(k)} - \left(\mathbf{H}_f(\mathbf{x}^{(k)}) \right)^{-1} (\text{grad } f(\mathbf{x}^{(k)}))\end{aligned}$$

5. Optimierung, LGS und das CG-Verfahren



$$\mathbf{x}^{(k+1)} := \mathbf{x}^{(k)} - \left(\mathbf{H}_f(\mathbf{x}^{(k)}) \right)^{-1} \left(\text{grad } f(\mathbf{x}^{(k)}) \right)$$

Mit Definition 4.9 aus VL 4 erhalten wir die Iterationsvorschrift:

- ① Werte den Gradienten $\mathbf{g} = \text{grad } f(\mathbf{x}^{(k)})$ von f im Punkt $\mathbf{x}^{(k)}$ aus
- ② Werte die Hesse-Matrix $\mathbf{A} = \mathbf{H}_f(\mathbf{x}^{(k)})$ von f im Punkt $\mathbf{x}^{(k)}$ aus
- ③ Löse das LGS $\mathbf{Ay} = \mathbf{g}$
- ④ Aktualisiere $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{y}$

Wir erwarten also dieselben Vor- und Nachteile wie in VL 4.

In der Optimierung wird der Konvergenzverlauf von Verfahren oft mit sogenannten **Niveaulinien-Plots** illustriert, die Bereiche in \mathbb{R}^2 mit gleichem Wert der Zielfunktion zeigen.

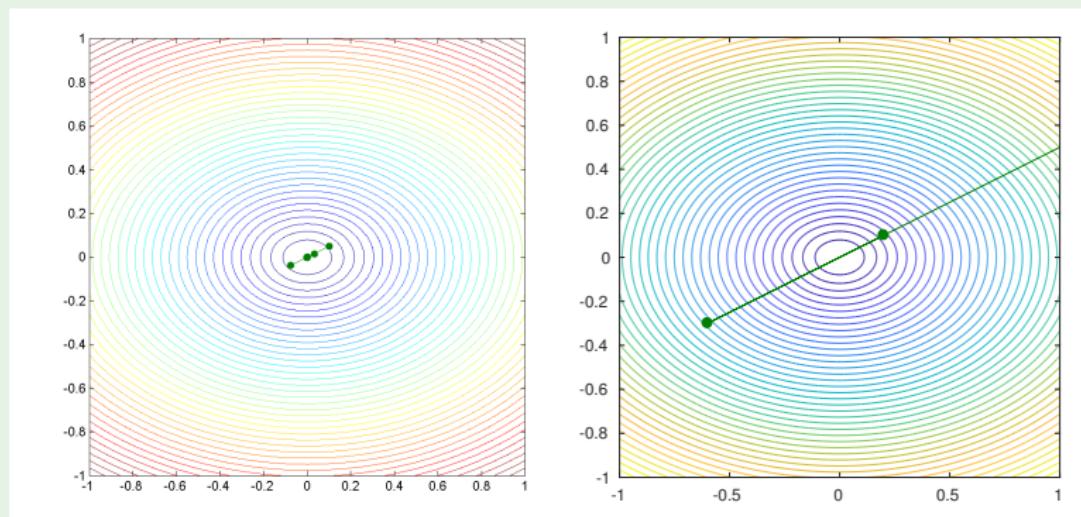
5. Optimierung, LGS und das CG-Verfahren



Beispiel 5.8 (Konvergenz des Newton-Verfahrens zur Optimierung)

Wir betrachten für zwei verschiedene Startwerte die Zielfunktion

$$f(x_1, x_2) := \frac{1}{2} \sqrt{\frac{1}{2}x_1^2 + x_2^2 + 0.01} \text{ mit Minimierer } \mathbf{x}^* = (0, 0)^T.$$



Links: Konvergenz mit $\mathbf{x}^{(0)} = (0.1, 0.05)^T$, rechts: Divergenz mit $\mathbf{x}^{(0)} = (0.2, 0.1)^T$

5. Optimierung, LGS und das CG-Verfahren



Das Verfahren konvergiert lokal wieder sehr schnell, d. h. für Startwerte, die bereits nahe am Optimum liegen, vgl. VL 4.

Der kleine Konvergenzbereich betont nochmals die Notwendigkeit für Varianten des Newton-Verfahrens, die robuster sind bzgl. der Wahl der Anfangswerte. Insbesondere sind alle Varianten des Newton-Verfahrens aus VL 4 auch für das Newton-Optimierungsverfahren anwendbar, beispielsweise das gedämpfte Newton-Verfahren, das den Konvergenzbereich deutlich vergrößern kann.

Das Newton-Verfahren zur Optimierung besitzt zwei weitere Nachteile: Die Hesse-Matrix H_f der Zielfunktion f ist i. A. dicht besetzt, somit ist das Verfahren für hochdimensionale Probleme nicht anwendbar. Außerdem muss f zweimal stetig differenzierbar sein. Alle drei Nachteile beheben wir im Folgenden mit dem **Gradientenverfahren**, auf Kosten der Konvergenzgeschwindigkeit.



Das Gradienten-Verfahren



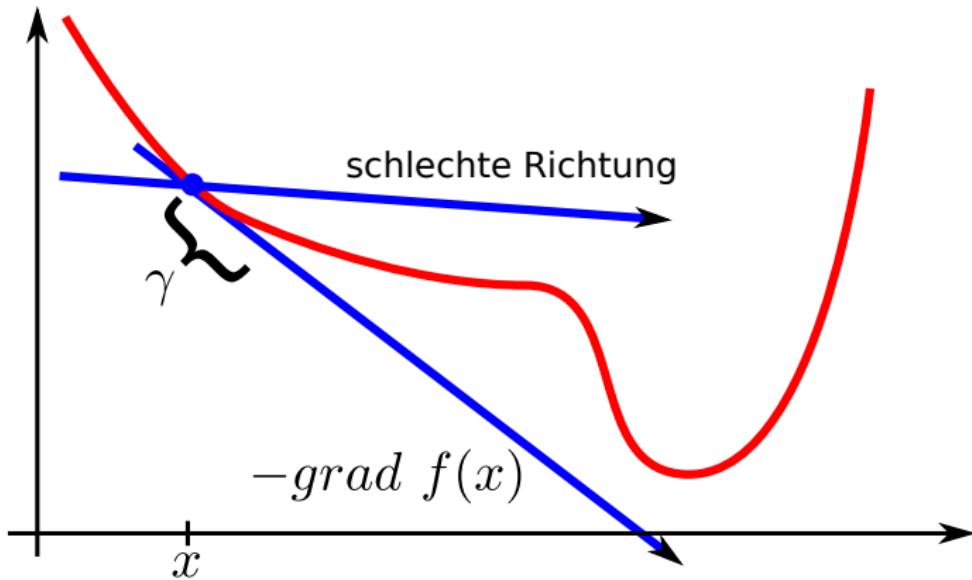
Das **Gradientenverfahren** benötigt nur noch die einmalige Differenzierbarkeit der Zielfunktion f , d. h. ihren Gradienten und nicht mehr ihre Hesse-Matrix. Somit ist das Gradientenverfahren in viel mehr Fällen anwendbar.

Erste Idee: Solange $\text{grad } f(\mathbf{x}^{(k)}) \neq \mathbf{0}$ gilt, ist $\mathbf{x}^{(k)}$ noch nicht der Minimierer, weil die notwendige Bedingung für ein Minimum noch nicht erfüllt ist. Diese Idee haben wir schon beim Newton-Verfahren zur Optimierung genutzt.

Zweite Idee: Solange wir noch nicht im Optimum sind, zeigt der negative Gradient $\mathbf{d} := -\text{grad } f(\mathbf{x})$ in die Richtung, in der **lokal** ausgehend von \mathbf{x} der größte Schritt zum Minimum erfolgt, vgl. die Analogie zwischen Ableitung und Tangentensteigung in der Schule und in VL 4.

Dritte Idee: Marschiere entlang der Richtung \mathbf{d} mit einer noch zu bestimmenden Schrittweite γ .

5. Optimierung, LGS und das CG-Verfahren





Damit erhalten wir ohne Komplikationen das Gradientenverfahren:

Definition 5.9 (Gradientenverfahren)

Die Zielfunktion $f: D \rightarrow \mathbb{R}$ eines unrestringierten Optimierungsproblems

$$\text{finde } \mathbf{x}^* \text{ mit } \mathbf{x}^* = \arg \min_{\mathbf{x} \in D} f(\mathbf{x})$$

sei einmal stetig differenzierbar, d. h. der Gradient von f existiere für jeden Punkt $\mathbf{x} \in D = \mathbb{R}^N$.

Das **Gradientenverfahren zur Optimierung** lautet für einen Startwert $\mathbf{x}^{(0)} \in \mathbb{R}^N$:

$$\begin{aligned}\mathbf{d}^{(k)} &:= -\operatorname{grad} f(\mathbf{x}^{(k)}) \\ \mathbf{x}^{(k+1)} &:= \mathbf{x}^{(k)} + \gamma^{(k)} \mathbf{d}^{(k)}.\end{aligned}$$

Hierbei sind $\gamma^{(k)}$ noch sinnvoll zu wählende **Schrittweiten**.



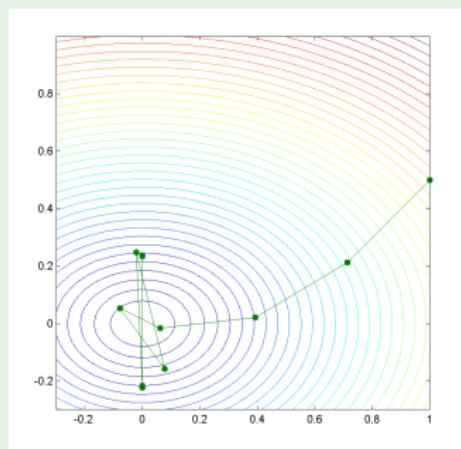
5. Optimierung, LGS und das CG-Verfahren

Eine erste Idee ist die Wahl von $\gamma^{(k)}$, ohne f zu berücksichtigen:

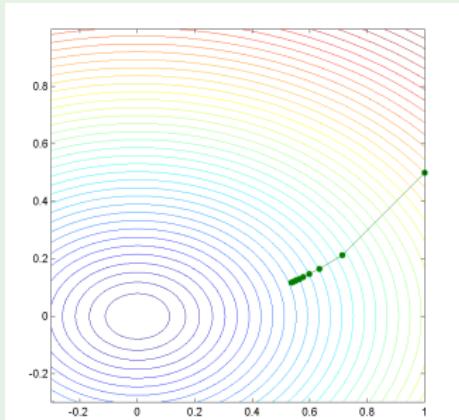
Beispiel 5.10 (Schlechte Schrittweitenwahl für Gradientenverfahren)

Wir betrachten wieder $f(x_1, x_2) := \frac{1}{2} \sqrt{\frac{1}{2}x_1^2 + x_2^2 + 0.01}$ mit $x^{(0)} = (1, 0.5)^T$

Konstante Schrittweite $\gamma^{(k)} = 1$



Abfallende Schrittweite $\gamma^{(k)} = \frac{1}{(k+1)^2}$



Optimum x^* wird „übersprungen“

Langsame Konvergenz bzw. Stagnation



5. Optimierung, LGS und das CG-Verfahren

Schlauer ist die Berücksichtigung von f bei der Wahl der Schrittweite $\gamma^{(k)}$:

Satz 5.11 (Optimale Schrittweite durch Liniensuche)

Definiere $g: \mathbb{R}_{>0} \rightarrow \mathbb{R}$ durch $g(\gamma) := f(\mathbf{x}^{(k)} + \gamma \mathbf{d}^{(k)})$ und wähle $\gamma^{(k)}$ als optimale Schrittweite, d. h.

$$\gamma^{(k)} := \arg \min_{\gamma \in \mathbb{R}_{>0}} g(\gamma).$$

Vergleiche das gedämpfte Newtonverfahren aus VL 4, das eine sehr ähnliche Liniensuche verwendet.

Mit dieser Wahl von $\gamma^{(k)}$ marschieren wir also **lokal**, d. h. im Schritt k , optimal weit entlang der optimalen Richtung $\mathbf{d}^{(k)}$. Man kann zeigen, dass dann gilt:

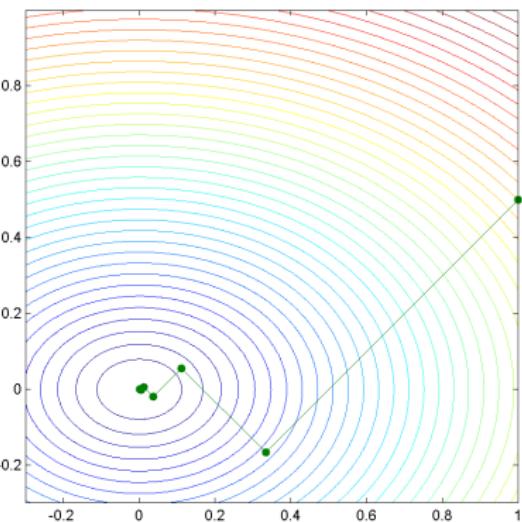
$$f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)}) \tag{5.6}$$

Die Approximation des Optimums wird also in jedem Schritt besser. I.A. erfordert dieses Verfahren in jedem Schritt die Lösung eines 1D-Hilfs-Optimierungsproblems, dies kann aufwendig sein.



Beispiel 5.12 (Gute Schrittweitenwahl für Gradientenverfahren)

Wir betrachten wieder $f(x_1, x_2) := \frac{1}{2} \sqrt{\frac{1}{2}x_1^2 + x_2^2 + 0.01}$ mit $\mathbf{x}^{(0)} = (1, 0.5)^T$:



Der Konvergenzbereich ist signifikant größer als beim Newton-Verfahren.

Beobachtung: Die Abstiegsrichtungen scheinen orthogonal zu sein, aufeinanderfolgende Vektoren $\mathbf{d}^{(k)}$ „stehen senkrecht aufeinander“. Diese Beobachtung ist essentiell für die weitere Vorlesung.



Gradientenverfahren für konvexe quadratische Funktionen



5. Optimierung, LGS und das CG-Verfahren

Wir betrachten nun eine spezielle Klasse von Optimierungsproblemen:

Definition 5.13 (Konvexes quadratisches Optimierungsproblem)

Sei $\mathbf{A} \in \mathbb{R}^{N \times N}$ symmetrisch und positiv definit, $\mathbf{b} \in \mathbb{R}^N$, $c \in \mathbb{R}$. Dann heißt

$$\text{finde } \mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x})$$

mit $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b}^T \mathbf{x} + c$ **unrestringiertes konvexes quadratisches Optimierungsproblem** („Quadratisches Programm“, QP).

vgl. Beispiel 5.3

Solche Probleme sind in der Praxis weit verbreitet. Beispiele sind Ausgleichs- und Regressionsprobleme in der Statistik, oder die Abstandsberechnung zwischen Polyedern im Raum. Quadratische Programme treten auch häufig als Teilproblem bei anderen Optimierungsverfahren auf.



5. Optimierung, LGS und das CG-Verfahren

Das Problem (QP) heißt „konvex“, weil

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b}^T \mathbf{x} + c$$

eine konvexe Funktion ist: Jede Strecke zwischen zwei Punkten auf dem Graphen von f liegt oberhalb des Graphen. Das kann man allgemein zeigen.

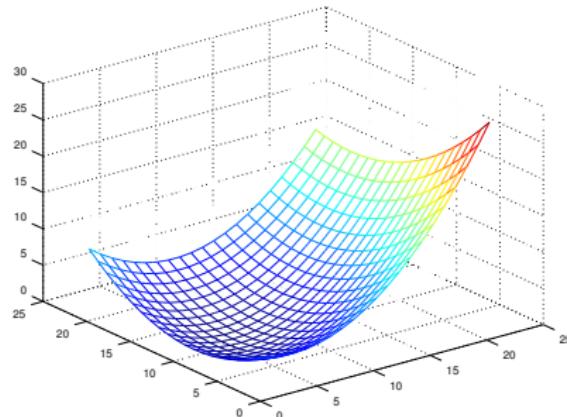


Illustration der Zielfunktion für $\mathbf{A} = \begin{pmatrix} 16 & -4 \\ -4 & 10 \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} 4 \\ 8 \end{pmatrix}$ und $c = 5$.



Für (QP) vereinfacht sich das Optimierungsproblem erheblich:

Satz 5.14 (Existenz & Eindeutigkeit, Äquivalenz zu LGS)

Das Problem (QP) besitzt einen eindeutigen Minimierer, welcher durch die Lösung des linearen Gleichungssystems $\mathbf{A}\mathbf{x} = \mathbf{b}$ gegeben ist.

Wir können das Minimum also durch die Lösung eines LGS bestimmen, das ist deutlich einfacher. Wir skizzieren nun den Beweis dieses etwas verblüffende Resultat, und überlegen uns danach Konsequenzen dieses Satzes.

5. Optimierung, LGS und das CG-Verfahren



Beweisskizze: Wir leiten die Zielfunktion

$$f(\mathbf{x}) := \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b}^T \mathbf{x} + c$$

ab, und erhalten („ $(x^2)' = 2x$ “) den Gradienten und die Hesse-Matrix von f :

$$\mathbf{g}(\mathbf{x}) := \text{grad } f(\mathbf{x}) = \mathbf{A} \mathbf{x} - \mathbf{b} \quad \text{sowie} \quad \mathbf{H}_f(\mathbf{x}) = \mathbf{A}$$

Dann sind äquivalent („erste Ableitung Null und zweite Ableitung größer Null“):

$$\mathbf{x}^* \text{ Minimierer von (QP)} \Leftrightarrow \mathbf{g}(\mathbf{x}^*) = \mathbf{0} \text{ und } \mathbf{H}_f(\mathbf{x}^*) \text{ positiv definit}$$

Nach Definition 5.13 ist $\mathbf{H}_f(\mathbf{x}) = \mathbf{A}$ für (QP) immer positiv definit, also

$$\text{grad } f(\mathbf{x}^*) = \mathbf{0} \Leftrightarrow \mathbf{A} \mathbf{x}^* - \mathbf{b} = \mathbf{0}.$$

Das wollten wir zeigen. □



Ein weiterer Vorteil von (QP) ist: Wir können die Bestimmung der optimalen Schrittweite für das Gradientenverfahren dramatisch vereinfachen: Es reicht nämlich, eine skalare quadratische Funktion zu minimieren, was durch Nullstellensuche einer linearen Gleichung per Formel möglich ist:

Satz 5.15 (Optimale Schrittweite für QP)

Sei $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{d} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$ und $g(\gamma) := f(\mathbf{x} + \gamma\mathbf{d})$ für $\gamma \in \mathbb{R}_{>0}$ und die quadratische Zielfunktion f aus Definition 5.13. Dann wird $g(\gamma)$ minimiert durch die optimale Schrittweite

$$\gamma^* := \frac{\mathbf{d}^\top \mathbf{r}}{\mathbf{d}^\top \mathbf{A} \mathbf{d}} \quad \text{mit} \quad \mathbf{r} := \mathbf{b} - \mathbf{A}\mathbf{x},$$

und es gilt $\gamma^* > 0$. Insbesondere gilt in jedem Schritt die Monotoniebedingung $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$, und somit die Konvergenz des Optimierungsverfahrens.



5. Optimierung, LGS und das CG-Verfahren

Beweisskizze: Aus der Zielfunktion

$$f(\mathbf{x}) := \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b}^T \mathbf{x} + c$$

erhalten wir nach lästiger Rechnung explizite Darstellungen von $g(\gamma)$ und $g'(\gamma)$:

$$\begin{aligned} g(\gamma) &= \frac{1}{2}(\mathbf{x} + \gamma \mathbf{d})^T \mathbf{A}(\mathbf{x} + \gamma \mathbf{d}) - \mathbf{b}^T(\mathbf{x} + \gamma \mathbf{d}) + c \\ &= (\frac{1}{2} \mathbf{d}^T \mathbf{A} \mathbf{d})\gamma^2 + (\mathbf{x}^T \mathbf{A} \mathbf{d} - \mathbf{b}^T \mathbf{d})\gamma + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b}^T \mathbf{x} + c \\ g'(\gamma) &= (\mathbf{d}^T \mathbf{A} \mathbf{d})\gamma + \mathbf{x}^T \mathbf{A} \mathbf{d} - \mathbf{b}^T \mathbf{d} \end{aligned}$$

Weil f konvex ist, ist auch g konvex, damit ist die hinreichende Bedingung für ein globales Minimum erfüllt. Die notwendige Bedingung lautet:

$$g'(\gamma^*) = 0 \Leftrightarrow \gamma^* = \frac{\mathbf{b}^T \mathbf{d} - \mathbf{x}^T \mathbf{A} \mathbf{d}}{\mathbf{d}^T \mathbf{A} \mathbf{d}} = \frac{\overbrace{\mathbf{d}^T (\mathbf{b} - \mathbf{A} \mathbf{x})}^{=r}}{\mathbf{d}^T \mathbf{A} \mathbf{d}}$$

Das wollten wir zeigen. □



5. Optimierung, LGS und das CG-Verfahren

Damit können wir für (QP) die optimale Schrittweite berechnen, weil alle benötigten Größen zur Verfügung stehen. Weil auch die nötige Abstiegsrichtung des Gradientenverfahrens

$$\mathbf{d}^{(k)} = -\text{grad } f(\mathbf{x}^{(k)}) = -\mathbf{A}\mathbf{x}^{(k)} + \mathbf{b}$$

explizit verfügbar ist, erhalten wir einen stark vereinfachten Algorithmus:

Definition 5.16 (Gradientenverfahren für QP)

Ausgehend von einem Startwert $\mathbf{x}^{(0)}$ lauten die Iterierten des Gradientenverfahrens für das QP-Problem:

$$\mathbf{x}^{(k+1)} := \mathbf{x}^{(k)} + \gamma^{(k)} \mathbf{d}^{(k)}$$

Hierbei sind

$$\mathbf{d}^{(k)} := -\mathbf{A}\mathbf{x}^{(k)} + \mathbf{b} = \mathbf{r}^{(k)} \quad \text{und} \quad \gamma^{(k)} := \frac{(\mathbf{d}^{(k)})^\top \mathbf{d}^{(k)}}{(\mathbf{d}^{(k)})^\top \mathbf{A} \mathbf{d}^{(k)}}.$$



In Beispiel 5.12 haben wir geometrisch-exemplarisch die Orthogonalität der gewählten Abstiegsrichtungen gesehen. Diese Orthogonalität ist ein wichtiger Zwischenschritt zur weiteren Verbesserung des Gradientenverfahrens für (QP): Dazu erinnern wir uns an HM123 und VL 3:

Definition 5.17 (Orthogonalität bzgl. Skalarprodukt)

Wir nennen zwei Vektoren $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$ orthogonal, falls

$$\mathbf{x}^T \mathbf{y} = 0.$$



Satz 5.18 (Orthogonalität der Abstiegsrichtungen)

Für die Abstiegsrichtungen $\mathbf{d}^{(k)}$ aus dem Gradientenverfahren für (QP) gilt:

$\mathbf{d}^{(k)}$ ist orthogonal zu $\mathbf{d}^{(k+1)}$

Beweis: Wir setzen ein,

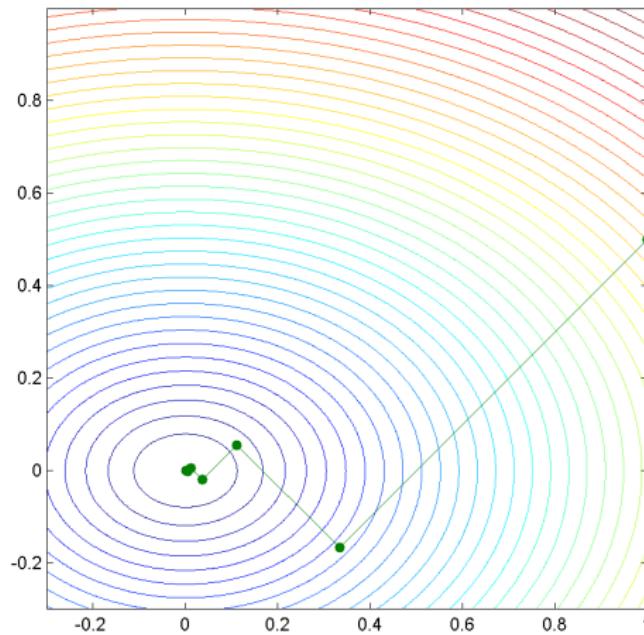
$$\begin{aligned}\mathbf{d}^{(k+1)} &= -\mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{b} \\ &= -\mathbf{A}(\mathbf{x}^{(k)} + \gamma^{(k)}\mathbf{d}^{(k)}) + \mathbf{b} \\ &= \mathbf{d}^{(k)} - \gamma^{(k)}\mathbf{A}\mathbf{d}^{(k)},\end{aligned}$$

und skalarmultiplizieren alles mit $\mathbf{d}^{(k)}$. So erhalten wir die Orthogonalität:

$$(\mathbf{d}^{(k)})^\top \mathbf{d}^{(k+1)} = (\mathbf{d}^{(k)})^\top \mathbf{d}^{(k)} - \underbrace{\frac{(\mathbf{d}^{(k)})^\top \mathbf{d}^{(k)}}{(\mathbf{d}^{(k)})^\top (\mathbf{A}\mathbf{d}^{(k)})}}_{=\gamma^{(k)}} (\mathbf{d}^{(k)})^\top (\mathbf{A}\mathbf{d}^{(k)}) = 0.$$



5. Optimierung, LGS und das CG-Verfahren



Die Orthogonalität manifestiert sich anschaulich als „Zick-Zack-Verhalten“.



Durch geometrische Überlegungen haben wir für konvexe quadratische Optimierungsprobleme eine sehr effiziente Formulierung des Gradientenverfahrens gefunden: In jedem Schritt werden zwei Matrix-Vektor Multiplikationen und drei Vektor-Vektor Operationen durchgeführt.

Wegen Satz 5.14 (Äquivalenz QP und LGS) können wir das Gradientenverfahren auch als iterativen LGS-Löser verwenden, falls \mathbf{A} symmetrisch und positiv definit ist. Man kann zeigen, dass das Gradientenverfahren etwa die Konvergenzgeschwindigkeit des Einzelschrittverfahrens aus VL 2 aufweist.



CG-Verfahren für (QP)

und LGS mit A SPD



Anschaulich ist der Grund für die relativ langsame Konvergenz des Gradientenverfahrens für (QP) das Zick-Zack-Verhalten der Abstiegsrichtungen. Die Abstiegsrichtung als negativer Gradient ist nur lokal optimal, durch die Orthogonalität geht Information verloren. Etwas präziser formuliert ist das Problem, dass die Orthogonalität nicht transitiv ist: Aus der Orthogonalität von $\mathbf{d}^{(k-2)}$ zu $\mathbf{d}^{(k-1)}$ und der Orthogonalität von $\mathbf{d}^{(k-1)}$ zu $\mathbf{d}^{(k)}$ folgt gerade nicht die Orthogonalität von $\mathbf{d}^{(k-2)}$ zu $\mathbf{d}^{(k)}$.

Das beheben wir nun mit dem **Verfahren der konjugierten Gradienten (Conjugate Gradient, CG)**. Die Grundidee dabei ist, den Orthogonalitätsbegriff so zu ändern, dass durch die Ausnutzung von \mathbf{A} die Transitivität der Abstiegsrichtungen gewährleistet wird. Wir investieren also zusätzliches problemspezifisches Wissen, um das Verfahren weiter zu verbessern.



Definition 5.19 (\mathbf{A} -konjugierte Vektoren)

Sei $\mathbf{A} \in \mathbb{R}^{N \times N}$ symmetrisch und positiv definit. Dann heißen die Vektoren $\{\mathbf{d}^{(k)}\}_{k=1}^m$ mit $m \in \mathbb{N}_{\geq 2}$ **paarweise \mathbf{A} -konjugiert (paarweise \mathbf{A} -orthogonal)**, falls

$$(\mathbf{d}^{(k)})^T \mathbf{A} \mathbf{d}^{(\ell)} = 0, \quad \text{für alle } \ell \neq k, \quad k, \ell = 1, \dots, m.$$

Die \mathbf{A} -Konjugiertheit impliziert sofort die Basiseigenschaft von $\{\mathbf{d}^{(k)}\}_{k=1}^m$ in \mathbb{R}^m . Deshalb folgt: Ein Gradientenverfahren mit solchen Abstiegsrichtungen und optimaler Schrittweite gemäß Satz 5.15 konvergiert (abgesehen von Rundungsfehlern) in (höchstens) N Iterationen, d. h. nach Konstruktion einer Basis des \mathbb{R}^N .

Die Herleitung des Verfahrens ist nicht prüfungsrelevant und findet sich im Anhang.



Algorithmus 5.20 : CG-Verfahren

input : $\mathbf{A} \in \mathbb{R}^{N \times N}$ SPD, $\mathbf{b} \in \mathbb{R}^N$, $\mathbf{x}^{(0)} \in \mathbb{R}^N$, $\text{TOL} > 0$, k_{\max}
output : Approximation des Minimierers $\mathbf{x}^* \in \mathbb{R}^N$ von $f(\mathbf{x}) := \frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x} - \mathbf{b}^\top \mathbf{x}$

1 Setze $\mathbf{d}^{(0)} := \mathbf{r}^{(0)} := \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$ und $\alpha^{(0)} := (\mathbf{r}^{(0)})^\top (\mathbf{r}^{(0)})$;

2 $k := 0$;

3 **while** $k < k_{\max}$ **and** $\sqrt{\alpha^{(k)}} > \text{TOL}$ **do**

4 $\mathbf{v}^{(k)} := \mathbf{A}\mathbf{d}^{(k)}$;

5 $\gamma^{(k)} := \alpha^{(k)} / (\mathbf{v}^{(k)})^\top (\mathbf{d}^{(k)})$;

6 $\mathbf{x}^{(k+1)} := \mathbf{x}^{(k)} + \gamma^{(k)} \mathbf{d}^{(k)}$;

7 $\mathbf{r}^{(k+1)} := \mathbf{r}^{(k)} - \gamma^{(k)} \mathbf{v}^{(k)}$;

8 $\alpha^{(k+1)} := (\mathbf{r}^{(k+1)})^\top (\mathbf{r}^{(k+1)})$;

9 $\mathbf{d}^{(k+1)} := \mathbf{r}^{(k+1)} + (\alpha^{(k+1)} / \alpha^{(k)}) \mathbf{d}^{(k)}$;

10 **end**

11 **return** $\mathbf{x}^{(k)}$;

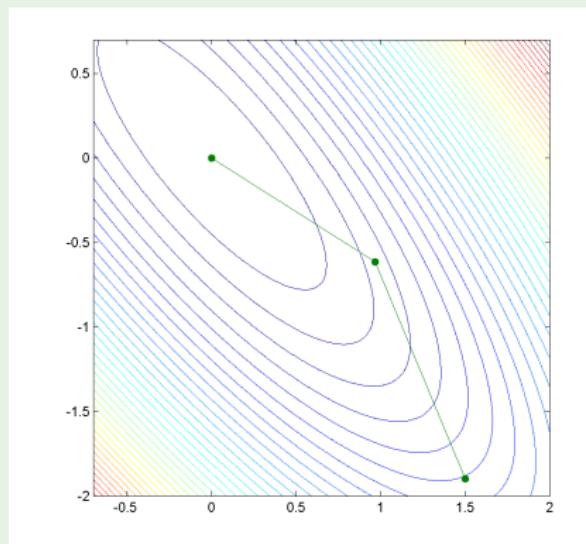


Beispiel 5.21 (Konvergenz des CG-Verfahrens)

Wir betrachten $f(\mathbf{x})$ mit $\mathbf{A} = \mathbf{U} \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{U}^T$, mit \mathbf{U} Rotationsmatrix, $\mathbf{b} = (0, 0)^T$. Das exakte Minimum ist also $\mathbf{x}^* = (0, 0)^T$.

Wir sehen: Das Zick-Zack-Verhalten ist verschwunden, und wegen $N = 2$ ist das Verfahren nach nur 2 Schritten konvergiert.

Man kann zeigen: Die Konvergenzgeschwindigkeit ist quadratisch schneller als beim Gradientenverfahren für (QP).



5. Optimierung, LGS und das CG-Verfahren

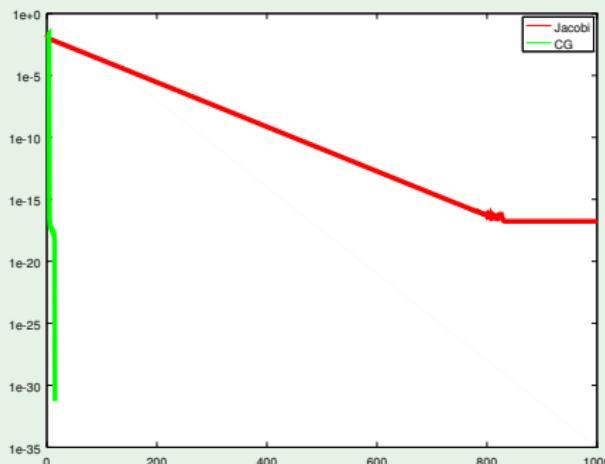


Beispiel 5.22 (CG-Verfahren als LGS-Löser)

Wir betrachten wie üblich das Raclette-Problem aus VL 0–2, d. h.

$\mathbf{A} = \text{tridiag}(-1, 2, -1)$. \mathbf{A} ist symmetrisch und positiv definit.

Konvergenzverlauf für $N = 10$:



N	Jacobi	CG
10	337	5
20	1151	10
40	4053	20
80	14468	40
160	51756	80
320	184133	160
640	647870	320



Zusammenfassung

- Optimierungsprobleme sind in der Praxis allgegenwärtig (Formoptimierung, Kostenminimierung, Materialminimierung, Energieminimierung,...).
- Für differenzierbare Funktionen kann das Gradientenverfahren oder das Newton-Verfahren verwendet werden. Eine Schrittweitensteuerung ist dabei unerlässlich.
- Für konvexe quadratische Optimierungsprobleme (QP) kann die optimale Schrittweite explizit berechnet werden. Zusammen mit dem Gradienten als Suchrichtung ergibt sich das Gradientenverfahren für (QP).
- Durch die Änderung des Orthogonalitätsbegriffs ergibt sich das deutlich effizientere CG-Verfahren.
- Jedes LGS mit SPD-Matrix kann als unrestringiertes konvexes quadratisches Minimierungsproblem geschrieben werden. Deshalb können die CG- und Gradientenverfahren als LGS-Löser verwendet werden.



Hausaufgaben

- Wiederholung der Begriffe Basis, Linearkombination etc.
- Wiederholung Polynome und Polynomraum, Monombasis
- Genießen Sie die Pfingstpause



Beispieldaufgaben



Das \mathbf{A} -Skalarprodukt

Sei $\mathbf{A} \in \mathbb{R}^{N \times N}$ symmetrisch und positiv definit, und seien $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$. Wir definieren das \mathbf{A} -Skalarprodukt als

$$(\mathbf{x}, \mathbf{y})_{\mathbf{A}} := \mathbf{x}^T \mathbf{A} \mathbf{y}.$$

Zeigen Sie, dass dies tatsächlich ein Skalarprodukt ist.

5. Optimierung, LGS und das CG-Verfahren



Lösungshinweise: Nachzuweisen sind die Linearität in beiden Argumenten,

$$(x, y + \lambda z)_A = (x, y)_A + \lambda(x, z)_A \quad \text{und} \quad (x + \lambda y, z)_A = (x, z)_A + \lambda(y, z)_A,$$

die Symmetrie

$$(x, y)_A = (y, x)_A,$$

und die positive Definitheit

$$(x, x)_A \geq 0 \quad \forall x \quad \text{und} \quad (x, x)_A = 0 \text{ für } x = 0.$$

Ergebnis: Der Nachweis basiert auf der Rückführung auf diese Eigenschaften beim euklidischen Skalarprodukt.



Orthogonalität

Zeigen Sie für die drei Vektoren

$$\mathbf{a} = (1, 0, 1, 0)^T, \quad \mathbf{b} = (1, 1, -1, 0)^T, \quad \mathbf{c} = (0, 1, 0, 1)^T,$$

dass die Orthogonalität tatsächlich nicht transitiv ist.



Lösungshinweise: Das ist eine reine Rechenaufgabe: Für die paarweise Orthogonalität müssen drei Skalarprodukte berechnet werden.

Ergebnis: \mathbf{a} und \mathbf{b} sind orthogonal, \mathbf{a} und \mathbf{c} auch, allerdings sind \mathbf{b} und \mathbf{c} nicht orthogonal.



Ergänzungen



Wir skizzieren für interessierte Teilnehmerinnen und Teilnehmer kurz die Knackpunkte der Herleitung des CG-Verfahrens.

Mit den Residuen $\mathbf{r}^{(0)}, \dots, \mathbf{r}^{(k)}$ stellen wir zur Ermittlung der \mathbf{A} -konjugierten Abstiegsrichtungen den folgenden Ansatz auf:

$$\begin{aligned}\mathbf{d}^{(0)} &= \mathbf{r}^{(0)} \\ \mathbf{d}^{(k)} &= \mathbf{r}^{(k)} + \sum_{i=0}^{k-1} \mu_i \mathbf{d}^{(j)}\end{aligned}$$

Wir bestimmen also die nächste Richtung als LiKo des aktuellen Residuums und den bisher bekannten Richtungen. Falls alle $\mu_j = 0$ wären, erhalten wir das Gradientenverfahren für (QP). Wir wollen nun die μ_j so bestimmen, dass alle Richtungen \mathbf{A} -orthogonal sind.



5. Optimierung, LGS und das CG-Verfahren

Die entsprechenden \mathbf{A} -Konjugiertheitsbedingungen lauten:

$$(\mathbf{A}\mathbf{d}^{(k)})^\top \mathbf{d}^{(i)} = 0 \quad \forall i = 0, \dots, k-1$$

Einsetzen des Ansatzes von der vorherigen Folie für $\mathbf{d}^{(k)}$ und Ausnutzen der Bilinearität des Skalarprodukts ergibt:

$$0 = (\mathbf{A}\mathbf{d}^{(k)})^\top \mathbf{d}^{(i)} = (\mathbf{A}\mathbf{r}^{(k)})^\top \mathbf{d}^{(i)} + \sum_{j=0}^{k-1} \mu_j (\mathbf{A}\mathbf{d}^{(j)})^\top \mathbf{d}^{(i)} \quad \forall i = 0, \dots, k-1$$

Das ist formal ein LGS mit k Gleichungen und k Unbekannten. In der Summe verschwinden jedoch in Gleichung i alle Summanden mit $j \neq i$, weil $\mathbf{d}^{(0)}, \dots, \mathbf{d}^{(k-1)}$ bereits in vorherigen Iterationen \mathbf{A} -konjugiert konstruiert wurden:

$$0 = (\mathbf{A}\mathbf{d}^{(k)})^\top \mathbf{d}^{(i)} = (\mathbf{A}\mathbf{r}^{(k)})^\top \mathbf{d}^{(i)} + \mu_i (\mathbf{A}\mathbf{d}^{(i)})^\top \mathbf{d}^{(i)} \quad \forall i = 0, \dots, k-1$$



5. Optimierung, LGS und das CG-Verfahren

Auflösen nach μ_i liefert die Werte für den Ansatz, die \mathbf{A} -Konjugiertheit sicherstellen:

$$\mu_i = -\frac{(\mathbf{A}\mathbf{r}^{(k)})^\top \mathbf{d}^{(i)}}{(\mathbf{A}\mathbf{d}^{(i)})^\top \mathbf{d}^{(i)}} \quad \forall i = 0, \dots, k-1$$

Diese Richtungen setzen wir nun in das Gradientenverfahren für (QP) ein:

$$① \quad \gamma^{(k)} = \frac{(\mathbf{r}^{(k)})^\top \mathbf{d}^{(k)}}{(\mathbf{A}\mathbf{d}^{(k)})^\top \mathbf{d}^{(k)}}$$

$$② \quad \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \gamma^{(k)} \mathbf{d}^{(k)}$$

$$③ \quad \mathbf{r}^{(k+1)} = -\mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{b}$$

$$④ \quad \mathbf{d}^{(k+1)} = \mathbf{r}^{(k+1)} - \sum_{j=0}^k \frac{(\mathbf{A}\mathbf{r}^{(k+1)})^\top \mathbf{d}^{(j)}}{(\mathbf{A}\mathbf{d}^{(j)})^\top \mathbf{d}^{(j)}} \mathbf{d}^{(j)}$$

Das ist nicht praxisrelevant, weil alle bisherigen Richtungen gespeichert werden müssen.



5. Optimierung, LGS und das CG-Verfahren

Als nächstes muss man beweisen, dass $r^{(k)}$ \mathbf{A} -konjugiert zu allen vorherigen Suchrichtungen $d^{(j)}$ ist. Das vereinfacht nun die Berechnung gewaltig, weil von der Summe nur der letzte Term übrig bleibt:

$$d^{(k+1)} = r^{(k+1)} - \frac{(\mathbf{A}r^{(k+1)})^\top d^{(k)}}{(\mathbf{A}d^{(k)})^\top d^{(k)}} d^{(k)}$$

Wir müssen also nicht mehr alle Richtungen abspeichern, sondern lediglich die beiden letzten Residuen.

Man kann nun noch einige weitere Verbesserungen durchführen: Die $r^{(k)}$ lassen sich inkrementell berechnen, und es lassen sich durch Umformungen und Abspeichern von Hilfs Ergebnissen fast alle Matrix-Vektor Multiplikationen einsparen.



6. Interpolation mit Polynomen und Splines



Joseph Louis Lagrange, 1736–1813

Si j'avais été riche, je ne m'aurais probablement pas attaché aux mathématiques.

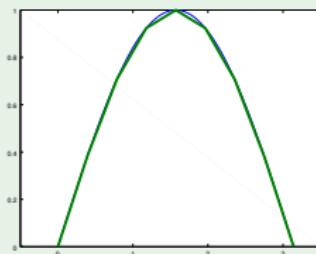
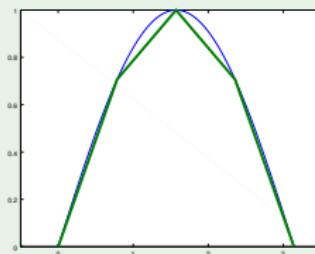
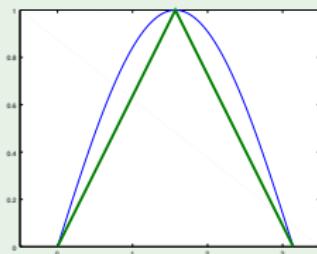


Motivation und Anwendungsbeispiele

6. Interpolation mit Polynomen und Splines



Beispiel 6.1 (Funktionsplots)



Ergebnis des `plot`-Befehls zur Darstellung der Sinus-Funktion auf dem Intervall $[0, \pi]$, für 3, 5 und 9 Auswertungspunkte. Eingezeichnet ist jeweils auch die „Referenzlösung“ mit mehr als 100 Auswertungspunkten.

Interpolation hat also etwas mit der einfach(er)en Repräsentation komplizierter Funktionen zu tun.

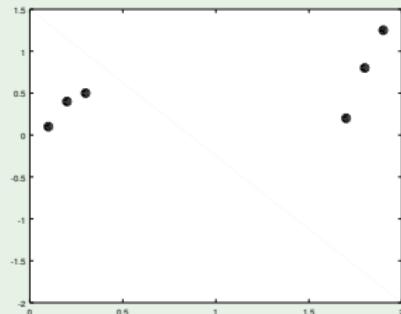


6. Interpolation mit Polynomen und Splines

Beispiel 6.2 (Messreihen)

Gegeben: endlich viele Messwerte für irgendeinen physikalischen, biologischen, chemischen, ... Vorgang

Nicht gegeben: Idee für einen funktionalen Zusammenhang



Wie kann man eine stetige Funktion bestimmen, die überall definiert ist und deren Graph durch **alle** Messwerte läuft? Das ist wichtig, um bspw. Auswertungen an Zwischenpunkten ohne erneute Experimente vornehmen zu können.

Interpolation hat also etwas mit der Konstruktion eines funktionalen, mindestens stetigen Zusammenhangs zwischen diskreten Werten zu tun.

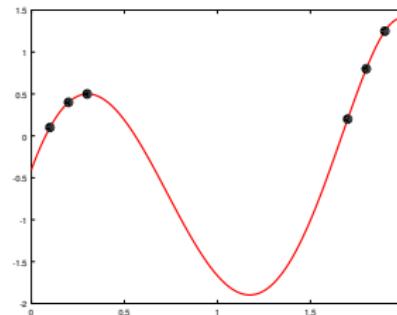
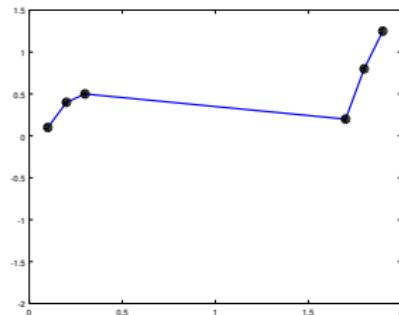


6. Interpolation mit Polynomen und Splines

Zwei Ideen zur Interpolation sind naheliegend:

Polygonzug (linearer Spline, stückweise Interpolation): Wir verbinden alle Messwerte sukzessive durch affine Funktionen, d. h. mit der Verbindungsgerade durch zwei benachbarte Messpunkte.

Einfache globale Funktionen: Wir bestimmen bspw. ein Polynom oder eine trigonometrische Funktion, das durch alle Messpunkte verläuft.



6. Interpolation mit Polynomen und Splines



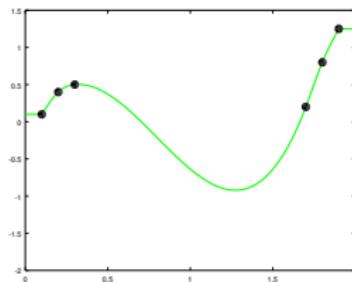
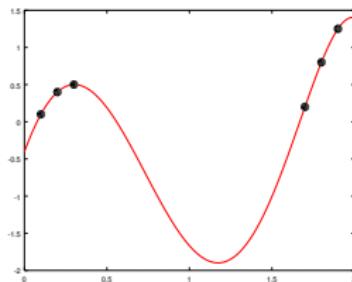
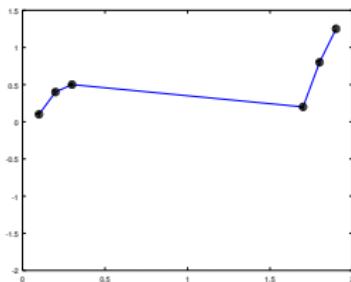
In vielen Anwendungen existieren zusätzliche Anforderungen:

Im CAD, bspw. im Karosseriedesign, sind runde Fahrzeugformen optisch ansprechender und haben Vorteile beim Luftwiderstand.

Numerische Methoden der Kontinuumsmechanik (VL 0 und 9ff) und der Optimierung (VL 5) benötigen häufig nicht nur Funktionsauswertungen, sondern auch Auswertungen der ersten (Gradient/Jacobi-Matrix) und/oder zweiten (Hesse-Matrix) partiellen Ableitungen.

Die interpolierende Funktion muss also hinreichend glatt sein, d.h. anschaulich keine Knicke enthalten. Etwas mathematischer fordern wir C^k mit einem vorgegebenen $k \in \{1, 2, \dots, \infty\}$.

6. Interpolation mit Polynomen und Splines



stückweise linear, Polynom vom Grad 5, kubischer natürlicher Spline

ViPLab-Demo in ILIAS

Die stückweise lineare Interpolation ist in den Messpunkten offenbar nur stetig und nicht glatt. Polynominterpolation liefert eine unendlich oft differenzierbare Approximation. Splines, d. h. stückweise polynomiale Funktionen mit (hier) C^2 -Übergängen sind meist ein guter Kompromiss: Sie weisen geringere Schwankungen als C^∞ -Polynome auf.

6. Interpolation mit Polynomen und Splines

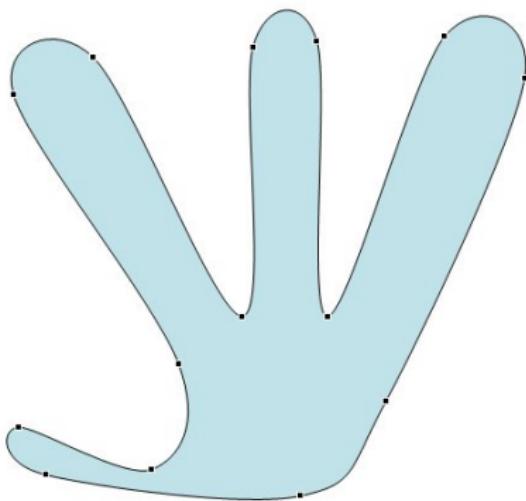


Beispiel 6.3 (Freihand-Kurven in geometrischer Modellierung / CAD)

Gegeben Punktsequenz $(x_0, y_0), \dots, (x_N, y_N)$ in der Ebene.

Wie kann das durch eine Kurve approximiert werden?

Lösungsmöglichkeit:



Separate Interpolation beider Koordinatensequenzen: Finde $p_x, p_y : [0, N] \rightarrow \mathbb{R}$ mit

$$p_x(n) = x_n, \quad p_y(n) = y_n, \quad n = 0, \dots, N.$$

Dann ist

$$\mathbf{p}(t) := (p_x(t), p_y(t))^T, \quad t \in [0, N]$$

interpolierende Kurve mit

$$\mathbf{p}(n) = (x_n, y_n), \quad n = 0, \dots, N.$$



Bevor wir basierend auf den Beispielen die Interpolationsaufgabe präzise formulieren können, benötigen wir noch eine Vokabel:

Definition 6.4 (Stützstellenmenge)

Zu einem Intervall $[a, b] \subset \mathbb{R}$ heißt die Menge $Z_N := \{x_0, \dots, x_N\}$ **Stützstellenmenge** von $[a, b]$, falls gilt:

$$a \leq x_0 < x_1 < \dots < x_N \leq b$$

Insbesondere sind alle Stützstellen paarweise verschieden, und es gilt $Z_N \subseteq [a, b]$. Die Intervallgrenzen a und b können Stützstellen sein, müssen aber nicht.



6. Interpolation mit Polynomen und Splines

Damit können wir die Interpolationsaufgabe präzise stellen:

Definition 6.5 (Allgemeine Interpolationsaufgabe)

Sei $Z_N = \{x_0, \dots, x_N\}$ eine Stützstellenmenge von $[a, b]$ und sei $\{y_0, y_1, \dots, y_N\}$ die Menge zugehöriger **Stützwerte** (z. B. Messdaten). Bestimme eine geeignete Funktion $p : [a, b] \rightarrow \mathbb{R}$, genannt **Interpolierende**, mit

$$p(x_n) = y_n \quad n = 0, \dots, N.$$

Folgende Fragen beantworten wir (unter anderem) in dieser Vorlesung:

- ① Wahl des Funktionenraums für die Interpolierende p
- ② Existenz und Eindeutigkeit der Interpolierenden p
- ③ Stetigkeit und Glattheit von p , d. h. $p \in C^k([x_0, x_N])$ mit $k \geq 0$
- ④ Konvergenz für eine gegebene (glatte) Funktion f mit $f(x_n) = y_n$ ($n = 0, \dots, N$): Gilt $p_N \rightarrow f$ für $N \rightarrow \infty$, d. h. für eine immer feiner werdende Stützstellenmenge Z_N ?



Polynominterpolation



Definition 6.6 (Polynome)

Sei $M \in \mathbb{N}$ und $\{a_0, \dots, a_M\} \subset \mathbb{R}$. Eine Funktion

$$p(x) = \sum_{m=0}^M a_m x^m$$

heißt **Polynom** vom Grad $\leq M$. Gilt $a_M \neq 0$ so ist $\text{Grad}(p) = M$.

Der Raum aller Polynome ist

$$\mathcal{P} := \{p : \mathbb{R} \rightarrow \mathbb{R} \mid p \text{ ist ein Polynom}\}$$

und

$$\mathcal{P}_M := \{p \in \mathcal{P} \mid \text{Grad}(p) \leq M\}$$

ist der Raum aller Polynome vom Grad $\leq M$.



Definition 6.7 (Polynominterpolation)

Sei $Z_N = \{x_0, \dots, x_N\}$ eine Stützstellenmenge von $[a, b] \subset \mathbb{R}$ und seien $\{y_0, \dots, y_N\} \subset \mathbb{R}$ gegebene Stützwerte. Das Polynom $p \in \mathcal{P}$ mit

$$p(x_n) = y_n \quad n = 0, \dots, N.$$

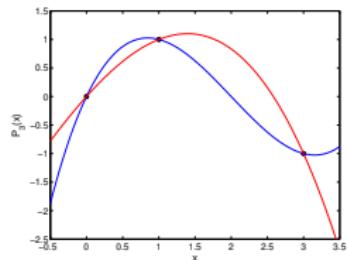
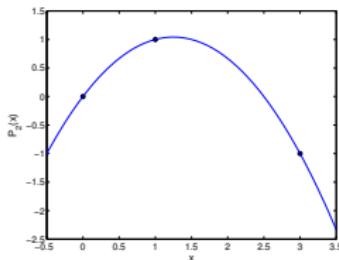
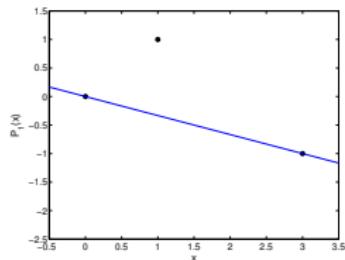
heißt **(Polynom-) Interpolierende** der Stützwerte in den Stützstellen.

Man beachte, dass der Grad des Interpolationspolynoms noch nicht festgelegt ist. Wir überlegen uns nun, dass die Existenz und Eindeutigkeit der Interpolierenden abhängt vom Zusammenhang zwischen dem Grad und der Anzahl der Stützstellen.



6. Interpolation mit Polynomen und Splines

Als Beispiel betrachten wir die Stützstellenmenge $\{x_0 = 0, x_1 = 1, x_2 = 3\}$ mit den Stützwerten $\{y_0 = 0, y_1 = 1, y_2 = -1\}$, und die Polynomräume $\mathcal{P}_1([0, 3])$, $\mathcal{P}_2([0, 3])$, $\mathcal{P}_3([0, 3])$:



Diese Interpolationsaufgabe zu 3 Stützstellen ist in $\mathcal{P}_1([0, 3])$ nicht lösbar, in $\mathcal{P}_2([0, 3])$ (eindeutig?) lösbar, und in $\mathcal{P}_3([0, 3])$ mehrdeutig lösbar.

Es drängt sich der Verdacht auf, dass die Aufgabe bei $N + 1$ Datenpunkten eindeutig lösbar in \mathcal{P}_N ist. Das beweisen wir nun in zwei Schritten, wobei der 2. Schritt (Existenz) konstruktiv ist, d. h. eine erste Berechnungsvorschrift für die Polynominterpolierende liefert.



Satz 6.8 (Eindeutigkeit in \mathcal{P}_N)

Seien eine Stützstellenmenge Z_N mit $N + 1$ Stützstellen und eine zugehörige Menge von Stützwerten $\{y_0, \dots, y_N\} \subset \mathbb{R}$ gegeben. Dann existiert höchstens eine Interpolierende in \mathcal{P}_N .

Paraphrasiert lautet die Aussage: Wenn eine Interpolierende in \mathcal{P}_N existiert, ist sie automatisch eindeutig.

Beweis: Seien p_1 und p_2 zwei Interpolierende in \mathcal{P}_N . Wir definieren das Polynom $p := p_1 - p_2$. Klar ist: $p \in \mathcal{P}_N$, und wegen $p_1(x_n) = p_2(x_n) \forall n = 0, \dots, N$ folgt

$$p(x_n) = 0 \quad \forall n = 0, \dots, N.$$

p besitzt also (mindestens) die $N + 1$ reellen Nullstellen $\{x_0, \dots, x_N\}$. Deshalb muss $p \equiv 0$ gelten, also ist $p_1 \equiv p_2$, und die Eindeutigkeit ist gezeigt. □



6. Interpolation mit Polynomen und Splines

Der letzte Satz legt nahe, für eine Menge von $N + 1$ Stützstellen (und zugehörige Stützwerte) die Lösung in \mathcal{P}_N zu suchen. Das passt zu unseren Beispielen.

Für die Existenz der Interpolierenden erinnern wir uns an die HM123:

Satz 6.9 (Basisdarstellung von \mathcal{P}_M)

Sei die Menge $\{\phi_0, \dots, \phi_M\}$ von Funktionen eine Basis von \mathcal{P}_M . Dann existieren für jedes $p \in \mathcal{P}_M$ Zahlen (Koeffizienten) $a_0, \dots, a_M \in \mathbb{R}$, so dass sich p schreiben lässt in der **Basisdarstellung** (als Linearkombination)

$$p(x) = \sum_{m=0}^M a_m \phi_m(x). \quad (6.7)$$

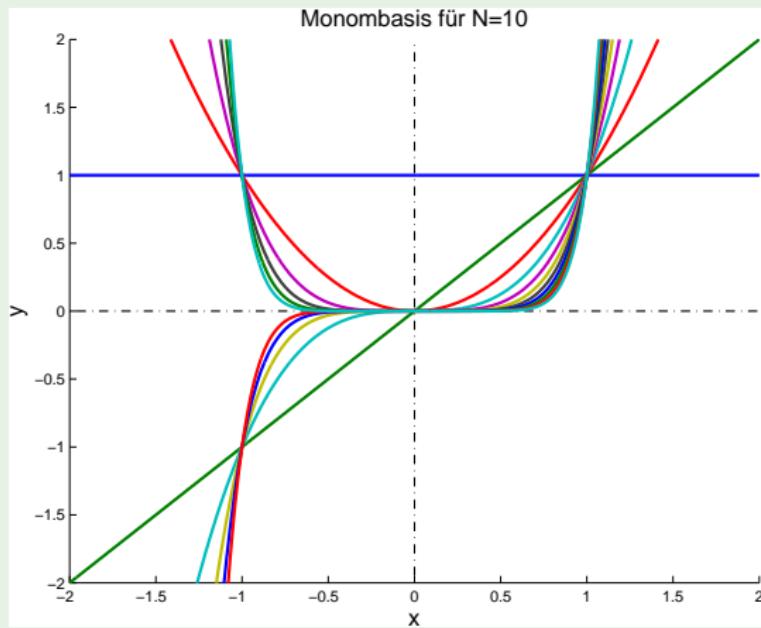
Wir verwenden M und nicht N , weil wir später den allgemeinen Fall $M < N$ benötigen.

6. Interpolation mit Polynomen und Splines



Beispiel 6.10 (Monombasis)

Die Funktionen (**Monome**) $\phi_m(x) := x^m$ für $m = 0, \dots, M$ bilden die **Monombasis** $\{x^0 = 1, x^1 = x, x^2, \dots, x^M\}$ von \mathcal{P}_M .





6. Interpolation mit Polynomen und Splines

Wir nutzen die Monombasis, um die Interpolierende zu berechnen, und damit ihre Existenz konstruktiv nachzuweisen. Dazu setzen wir die Interpolationsbedingungen $p(x_n) = y_n$ in die Monombasis-LiKo ein für $M = N$:

$$p(\textcolor{red}{x_n}) = \sum_{m=0}^N a_m \underbrace{\textcolor{red}{x_n}^m}_{=\phi_m(\textcolor{red}{x_n})} \stackrel{!}{=} y_n \quad \forall n = 0, \dots, N$$

Das ist nun aber nichts anderes als ein LGS für die unbekannten Koeffizienten a_0, \dots, a_N :

$$\underbrace{\begin{pmatrix} x_0^0 & x_0^1 & \dots & x_0^N \\ \vdots & & & \vdots \\ x_N^0 & x_N^1 & \dots & x_N^N \end{pmatrix}}_{=: \mathbf{A}} \underbrace{\begin{pmatrix} a_0 \\ \vdots \\ a_N \end{pmatrix}}_{=: \mathbf{a}} = \underbrace{\begin{pmatrix} y_0 \\ \vdots \\ y_N \end{pmatrix}}_{=: \mathbf{y}} \quad (6.8)$$

Im Eintrag a_{nm} der **Interpolationsmatrix \mathbf{A}** steht die Auswertung des m -ten Basismonoms an der n -ten Stützstelle. Wegen Z_N und \mathcal{P}_N ist die **Vandermonde-Matrix $\mathbf{A} \in \mathbb{R}^{N+1 \times N+1}$** quadratisch.



6. Interpolation mit Polynomen und Splines

Satz 6.11 (Vandermonde-Matrix)

Zu einer Zerlegung $Z_N = \{x_n\}_{n=0}^N$ zerfällt die Determinante der **Vandermonde-Matrix** $\mathbf{A} = (x_n^m)_{n,m=0}^N \in \mathbb{R}^{N+1 \times N+1}$ in ein Produkt aus Linearfaktoren:

$$\det(\mathbf{A}) = \prod_{0 \leq n' < n \leq N} (x_n - x_{n'})$$

Mit den Regularitätskriterien aus HM123 und VL 1 gilt also:

$$\det(\mathbf{A}) \neq 0 \Leftrightarrow \text{alle Stützstellen paarweise verschieden}$$

Wir schließen, dass das LGS (6.8) dann eindeutig lösbar ist:

Satz 6.12 (Existenz und Eindeutigkeit der Polynominterpolierenden)

Falls die Stützstellen paarweise verschieden sind, d. h. $x_n \neq x_{n'}$ für $n \neq n'$ ($n, n' = 0, \dots, N$), so ist die Vandermonde-Matrix regulär, und für beliebige Wahl der Stützwerte y_0, \dots, y_N existiert eine eindeutige Interpolierende $p \in \mathcal{P}_N$.



6. Interpolation mit Polynomen und Splines

Beispiel 6.13 (Berechnung des Interpolationspolynoms)

Wir betrachten die folgenden Stützstellen und -werte:

n	0	1	2
x_n	-1	0	1
y_n	-1	0	3

Die Interpolationsmatrix hat die folgende Gestalt:

$$\begin{pmatrix} \phi_0(x_0) & \phi_1(x_0) & \phi_2(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & \phi_2(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \phi_2(x_2) \end{pmatrix} = \begin{pmatrix} (-1)^0 & (-1)^1 & (-1)^2 \\ 0^0 & 0^1 & 0^2 \\ 1^0 & 1^1 & 1^2 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Die Lösung des linearen Gleichungssystems

$$\begin{pmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 3 \end{pmatrix} \quad \text{lautet} \quad \mathbf{a} = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}$$

und somit die Interpolierende $p(x) = 0 \cdot x^0 + 2 \cdot x^1 + 1 \cdot x^2 = x^2 + 2x$.



Mit Papier und Bleistift ist die Polynominterpolation also ein gelöstes Problem, wir müssen nur ein (dicht besetztes) LGS für die Koeffizienten lösen. Numerisch sind wir noch nicht fertig:

Beispiel 6.14 (Kondition der Vandermonde-Matrix)

Die Vandermonde-Matrix \mathbf{A} für die Polynominterpolation in der Monombasis ist schon für moderate N sehr schlecht konditioniert. Das Lösen des LGS (6.8) für die unbekannten Koeffizienten ist somit sehr anfällig für Rundungsfehler.

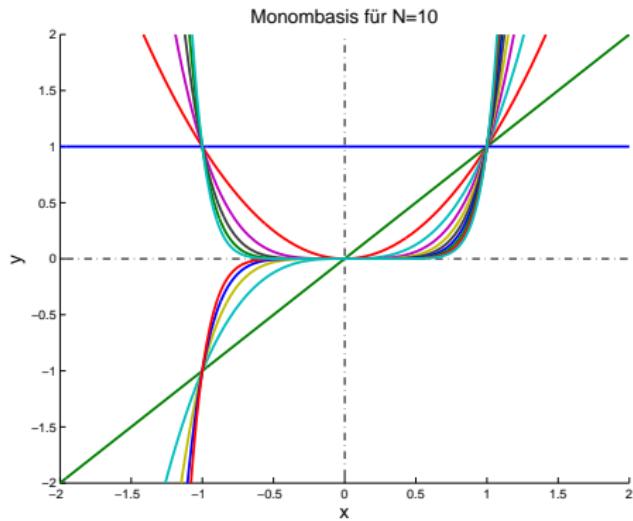
N	$\kappa_2(\mathbf{A})$
5	$4.924 \cdot 10^3$
10	$1.156 \cdot 10^8$
15	$3.122 \cdot 10^{12}$
20	$8.213 \cdot 10^{16}$

Spektralkondition κ_2 der Vandermonde-Matrix für äquidistante Zerlegung von $[0, 1]$, vgl. VIPLab-Demo in ILIAS.



6. Interpolation mit Polynomen und Splines

Die anschauliche Erklärung für dieses Phänomen ist, dass die Monome für große N im Vergleich zur Intervalllänge sehr ähnlich sind, und somit die Spalten der Vandermonde-Matrix ebenfalls sehr ähnlich, also (fast) linear abhängig sind.



Naheliegende Idee: Wir stellen das Interpolationspolynom bzgl. einer anderen Basis mit besseren Eigenschaften dar.

6. Interpolation mit Polynomen und Splines



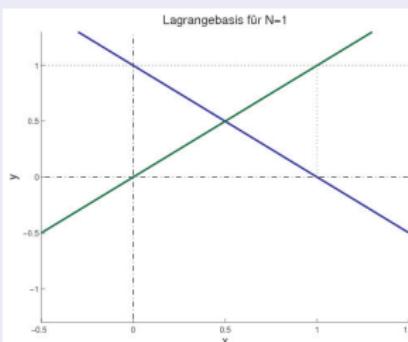
Definition 6.15 (Lagrange-Basis)

Für $m = 0, \dots, N$ und eine Zerlegung Z_N heißt $L_m \in \mathcal{P}_N$ gegeben durch

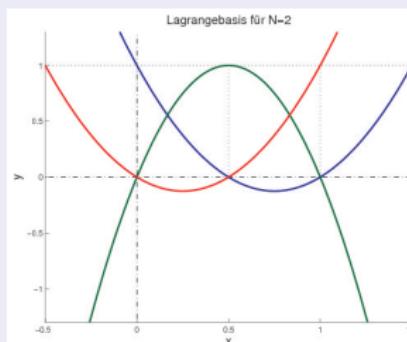
$$L_m(x) = \prod_{n=0, n \neq m}^N \frac{x - x_n}{x_m - x_n}$$

m -te Lagrange-Basisfunktion. $\{L_0(\cdot), \dots, L_N(\cdot)\}$ ist eine Basis von \mathcal{P}_N .

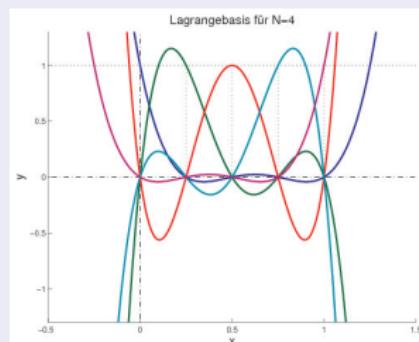
$N = 1:$



$N = 2:$



$N = 4:$



Verständnisfrage: Ist L_m für alle $m = 0, \dots, N$ wohldefiniert? Beispiel in 2 Folien



6. Interpolation mit Polynomen und Splines

Der Ansatz für die Interpolierende lautet in der Lagrange-Basis:

$$p(x) = \sum_{m=0}^N a_m L_m(x) \quad (6.9)$$

Wir erhalten also analog zur Monombasis das LGS

$$\mathbf{L}\mathbf{a} = \mathbf{y} \quad \text{mit Interpolationsmatrix} \quad \mathbf{L} = (L_m(x_n))_{n,m=0}^N. \quad (6.10)$$

Satz 6.16 (Interpolation mit Lagrange-Basis)

- (i) Es gilt $L_m(x_n) = \delta_{mn}$ für $m, n = 0, \dots, n$ mit dem Kronecker-Delta.
- (ii) Die Interpolationsmatrix ist also die Einheitsmatrix, d.h. $\mathbf{L} = \mathbf{I}$, das LGS (6.10) wird gelöst durch $\mathbf{a} = \mathbf{y}$, und die Funktion $p \in \mathcal{P}_N$ aus (6.9) löst die Interpolationsaufgabe.

Beweis: (i) Einsetzen und nachrechnen mit Fallunterscheidung $n \neq m$.
(ii) klar aus (i)





6. Interpolation mit Polynomen und Splines

Bei der Verwendung der Lagrange-Basis müssen wir also kein LGS lösen.

Insbesondere haben wir (Einheitsmatrix!) $\kappa_2(L) = 1$ für die Spektralkonditionszahl der Interpolationsmatrix, das ist optimal (vgl. VL 2).

Die Interpolierende lautet in allgemeiner Form

$$p(x) = \sum_{m=0}^N y_m \prod_{n=0, n \neq m}^N \frac{x - x_n}{x_m - x_n}$$

beispielsweise für die ersten 3 Interpolationspolynome:

$$p_0(x) = y_0$$

$$p_1(x) = y_0 \frac{(x-x_1)}{(x_0-x_1)} + y_1 \frac{(x-x_0)}{(x_1-x_0)}$$

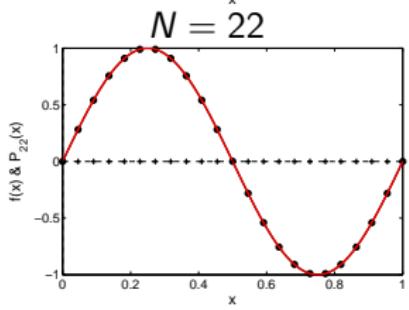
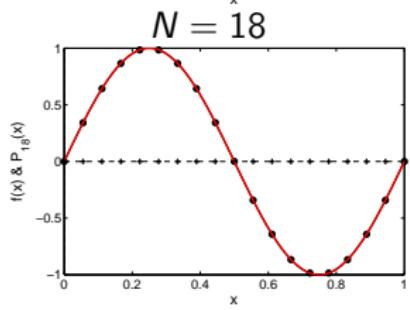
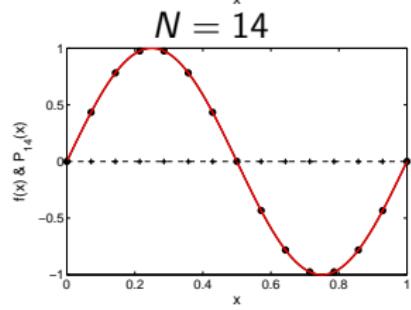
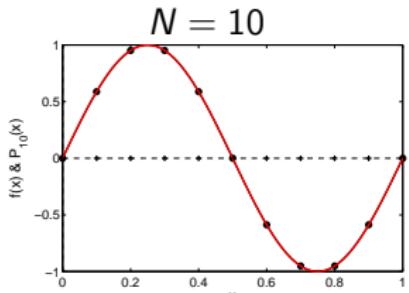
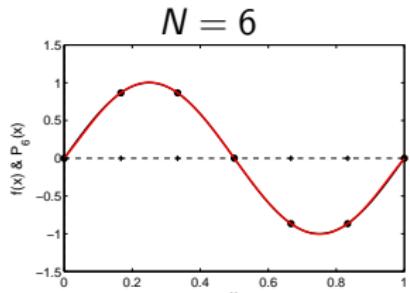
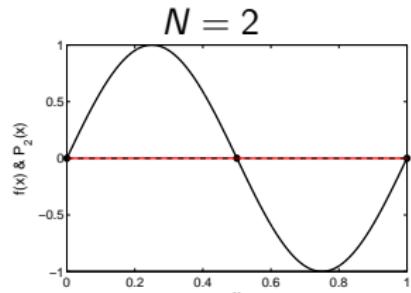
$$p_2(x) = y_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + y_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

Im Hinblick auf den Aufwand haben wir allerdings nichts gewonnen: Die Punktauswertung der Interpolierenden erfordert $\mathcal{O}(N^2)$ arithmetische Operationen, wie wir an den drei Beispielen schnell nachzählen.



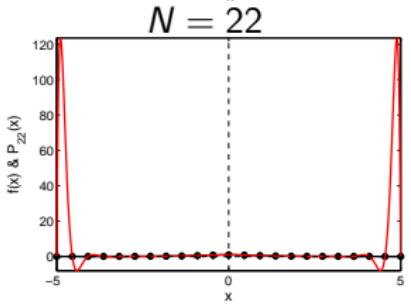
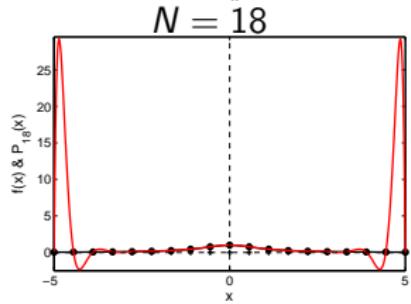
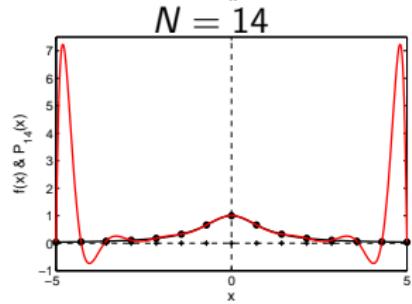
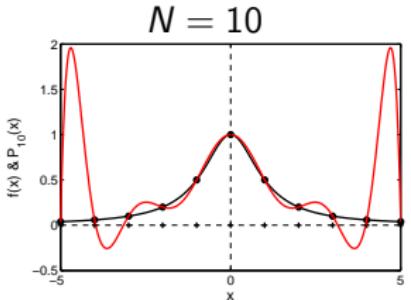
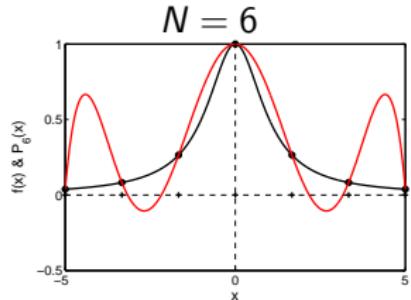
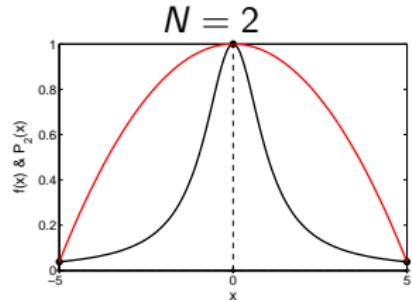
6. Interpolation mit Polynomen und Splines

Uns interessiert nun die Fehleranalyse zu einer gegebenen Funktion $f : [a, b] \rightarrow \mathbb{R}$, Stützstellen $\{x_0, \dots, x_N\}$ und zu interpolierende Stützwerte $\{y_n := f(x_n)\}_{n=0}^N$.



Interpolation von $f(x) = \sin(2\pi x)$ auf $[0, 1]$ mit äquidistanten Stützstellen

6. Interpolation mit Polynomen und Splines



Beispiel von Runge: Interpolation von $f(x) = (1 + x^2)^{-1}$ auf $[-5, 5]$ mit äquidistanten Stützstellen. Wir sehen extreme Oszillationen am Rand.



6. Interpolation mit Polynomen und Splines

Die Wahl der Stützstellen ist ein freier Parameter, um idealerweise die Oszillationen der Interpolierenden zu unterbinden oder wenigstens zu reduzieren.

Definition 6.17 (Äquidistante und Tschebyschow-Stützstellen)

Für ein Intervall $[a, b]$ und $N \in \mathbb{N}_0$ sind die **äquidistanten Stützstellen**

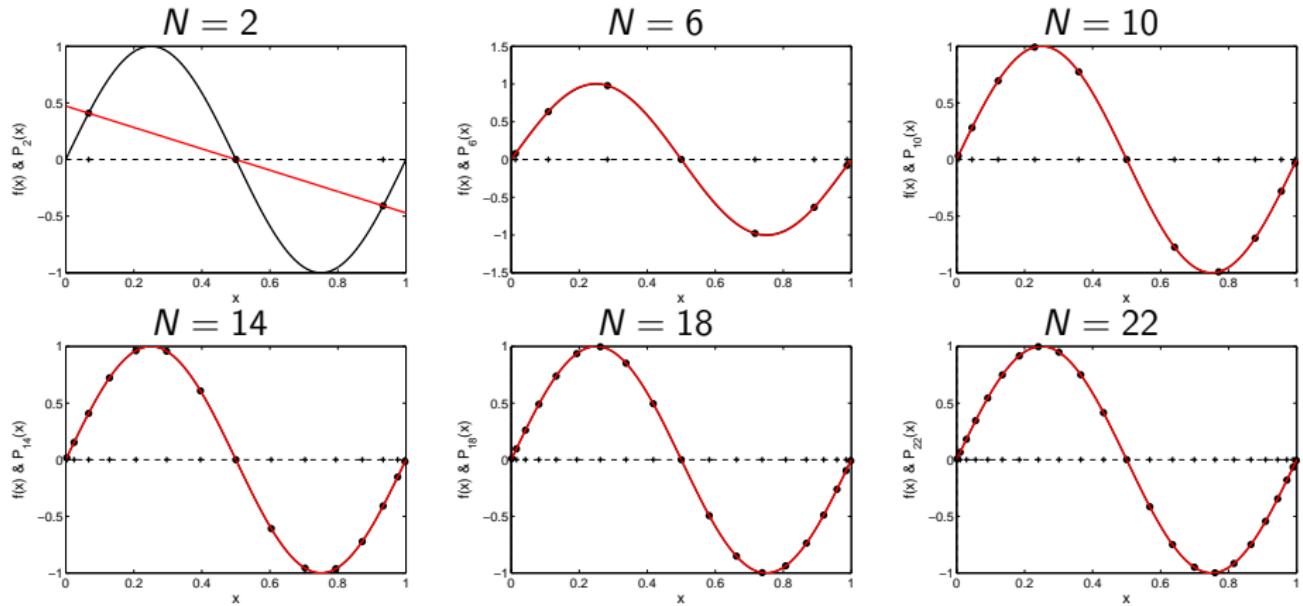
$$x_n = a + n \frac{b - a}{N} \quad n = 0, \dots, N$$

und die **Tschebyschow-Stützstellen**

$$x_n = \frac{1}{2} \left((b + a) - (b - a) \cos \left(\frac{2n+1}{2N+2} \pi \right) \right) \quad n = 0, \dots, N.$$

Man kann beweisen, dass die Tschebyschow-Stützstellen den Interpolationsfehler verglichen mit allen anderen Polynomen gleichen Grades minimieren, in der Maximumsnorm.

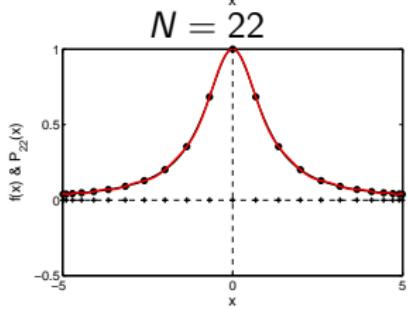
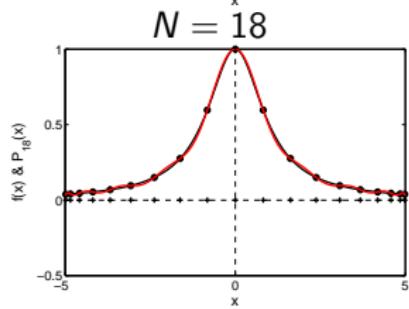
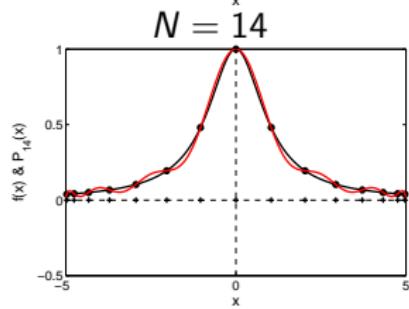
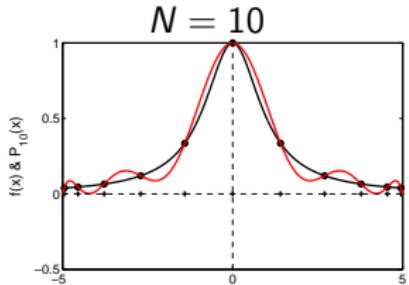
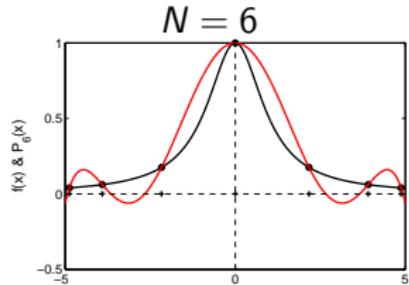
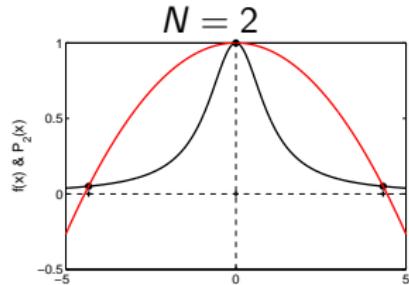
6. Interpolation mit Polynomen und Splines



Interpolation von $f(x) = \sin(2\pi x)$ auf $[0, 1]$ mit Tschebyschow-Stützstellen.



6. Interpolation mit Polynomen und Splines



Interpolation von $f(x) = (1 + x^2)^{-1}$ auf $[-5, 5]$ mit Tschebyschow-Stützstellen.
Die Oszillationen sind deutlich reduziert.



6. Interpolation mit Polynomen und Splines

Man kann sogar eine obere Schranke für den Interpolationsfehler zeigen, wenn die zu interpolierende Funktion hinreichend oft differenzierbar ist:

Satz 6.18 (Interpolationsfehler)

Sei $f \in C^{N+1}([a, b])$, $Z_N = \{x_0, \dots, x_N\}$ eine Zerlegung von $[a, b]$, $\{y_0 = f(x_0), \dots, y_N = f(x_N)\}$ eine Menge von Stützwerten und $p \in \mathcal{P}_N([x_0, x_N])$ das zugehörige Interpolationspolynom. Dann gilt:

$$\max_{x \in [x_0, x_N]} |p(x) - f(x)| \leq \frac{(b-a)^{N+1}}{(N+1)!} \max_{x \in [x_0, x_N]} |f^{(N+1)}(x)|$$

Wenn $b-a < 1$, liegt also Konvergenz der Interpolierenden gegen die Funktion vor, wenn die Ableitungen von f langsamer als faktoriell wachsen. Die Bedingung $b-a < 1$ kann oft durch stückweise Interpolation erreicht werden, das behandeln wir im nächsten Abschnitt.

6. Interpolation mit Polynomen und Splines



Wir fassen zusammen: Die Lagrange-Darstellung des Interpolationspolynoms $p \in \mathcal{P}_N$ ist einfach zu bestimmen im Sinne einer expliziten „Formel“. Als global definiertes Polynom ist die Interpolierende beliebig oft stetig differenzierbar.

Der Polynomgrad steigt jedoch mit jedem neu hinzukommenden Paar aus Stützstelle und Stützwert. Die Interpolierende muss dann jedes Mal neu berechnet werden, weil jede Stützstelle in jedes Lagrange-Basispolynom eingeht. Unsere Experimente zeigen, dass Polynome mit hohem Grad zu Oszillationen neigen. Dies kann durch Tschebyschow-Stützstellen reduziert werden.

Mit der stückweisen Interpolation werden einige dieser Probleme behoben.



Stückweise Polynominterpolation

6. Interpolation mit Polynomen und Splines

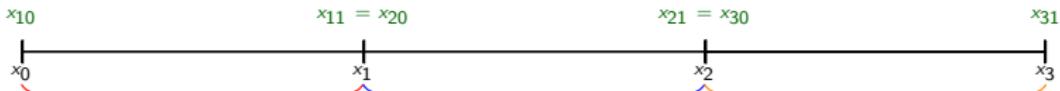


Idee: Stückweise Polynominterpolation

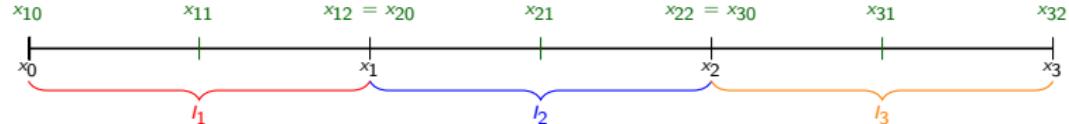
Sei $\{x_0, \dots, x_N\}$ eine Stützstellenmenge von $[a, b]$. Wir konstruieren auf den Teilintervallen $I_n = [x_{n-1}, x_n]$ ($n = 1, \dots, N$) jeweils separate Polynominterpolierende $p|_{I_n} \in \mathcal{P}_k$ mit festem Grad $k \ll N$. Dazu unterteilen wir jedes Teilintervall in äquidistante Hilfs-Stützstellen, vgl. Definition 6.4:

$$x_{nm} = x_{n-1} + m \frac{x_n - x_{n-1}}{k} \quad m = 0, \dots, k$$

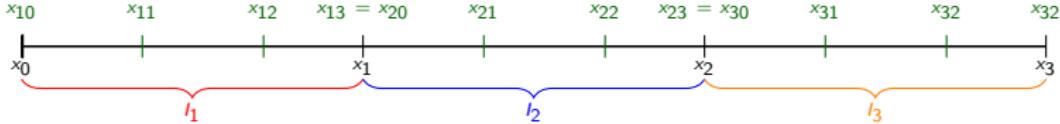
$k = 1$:



$k = 2$:

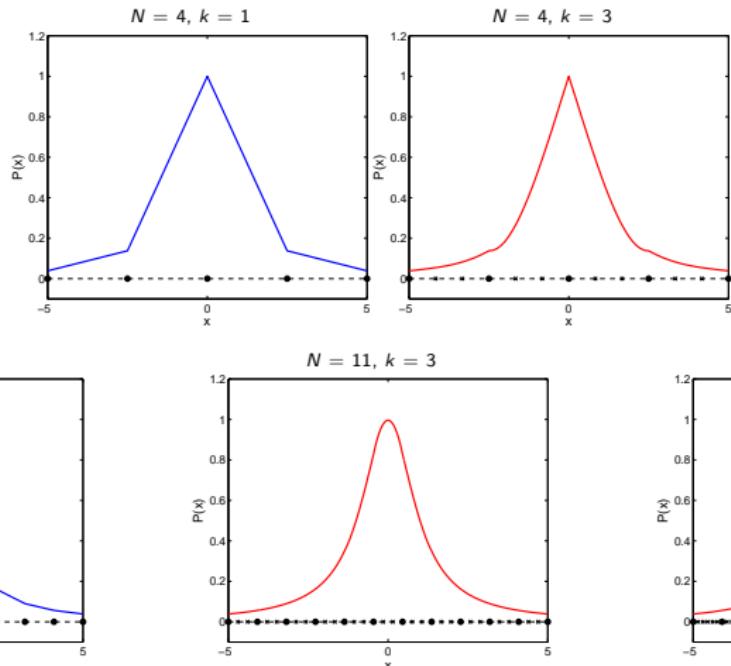


$k = 3$:





6. Interpolation mit Polynomen und Splines



Beispiel von Runge: Stückweise Interpolation von $f(x) = (1 + x^2)^{-1}$ auf $I = [-5, 5]$.



6. Interpolation mit Polynomen und Splines

Die Vorteile der Polynominterpolation bleiben erhalten, und die einzelnen Interpolierenden können auf den Teilintervallen so wie bisher konstruiert werden. Durch den nun moderaten Polynomgrad k im Vergleich zur Anzahl $(k+1)N$ der Stützstellen werden Oszillationen deutlich reduziert. Es gilt mit $h_{\max} = \max\{x_n - x_{n-1} \mid n = 1, \dots, N\}$ die Fehlerabschätzung

$$\max_{x \in [a, b]} |p(x) - f(x)| \leq \frac{h_{\max}^{k+1}}{(k+1)!} \max_{x \in [a, b]} |f^{(k+1)}(x)|$$

Neben dem Term $(k+1)!$ haben wir nun h_{\max} als Stellschraube, um explodierende Ableitungen einzufangen.

Die Interpolierende ist allerdings nach Konstruktion global nur stetig, selbst für $k > 1$. Insbesondere ist sie in den Stützstellen $\{x_1, \dots, x_{N-1}\}$ nicht differenzierbar. Im letzten Beispiel sollte man also seinen Augen lieber nicht trauen.



6. Interpolation mit Polynomen und Splines

Die k -fache Differenzierbarkeit auf $[a, b]$ kann durch die Vorgabe von Ableitungswerten statt Funktionswerten in den Stützstellen erreicht werden:

$$f(x_n), f'(x_n), \dots, f^{(k)}(x_n) \quad \forall n = 1, \dots, N - 1$$

Dies führt auf die (stückweise) **Hermite-Interpolation**.

In der Praxis sind höhere Ableitungswerte mit Messungen nur schwer oder gar nicht zu gewinnen. Daher kann i. A. die (stückweise) Hermite-Interpolation für Daten aus Messungen nicht verwendet werden.

Dieses Problem wird durch die Polynominterpolation mit Splines behoben, die allein aus Messwerten eine hinreichend glatte stückweise Polynominterpolation liefert.



Stückweise Interpolation mit Splines



Definition 6.19 (Spline k -ten Grades)

Sei $Z_N = \{x_0 = a, \dots, x_N = b\} \subseteq [a, b]$ eine Stützstellenmenge und $k \in \mathbb{N}$. Wir nennen eine Funktion $s : [a, b] \rightarrow \mathbb{R}$ einen **Spline k -ten Grades** (zu Z_N), wenn

- (i) $s|_{[x_{n-1}, x_n]} \in \mathcal{P}_k$ für alle $n \in \{1, \dots, N\}$,
- (ii) $s \in C^{k-1}(]a, b[)$.

Analog zum Raum $\mathcal{P}_N([a, b])$ heißt die Menge von Funktionen

$$S_{k,N} = \{s : [a, b] \rightarrow \mathbb{R} \mid s \text{ Spline } k\text{-ten Grades zu } Z_N\}$$

Splineraum. Man kann zeigen, dass $\dim S_{k,N} = N + k$.

Ein Spline k -ten Grades ist also auf jedem Teilintervall ein Polynom von Grad k , aber im Gegensatz zu einem Polynom aus \mathcal{P} nicht unendlich oft differenzierbar: In den Punkten x_n existieren die k -te und alle höheren Ableitungen nicht.



Die Definition einer Spline-Interpolierenden ist nun generisch möglich:

Definition 6.20 (Spline-Interpolierende)

Sei $Z_N = \{x_0 = a, \dots, x_N = b\} \subseteq [a, b]$ eine Stützstellenmenge und $k \in \mathbb{N}$ und Werte $\{y_0, \dots, y_N\} \subset \mathbb{R}$ gegeben. Wir nennen eine Funktion $s \in S_{k,N}$ **interpolierenden Spline**, falls

$$s(x_n) = y_n, \quad \forall n = 0, \dots, N.$$

Wir überlegen uns nun, wie wir für beliebiges N und die beiden gebräuchlichsten Fälle $k = 1$ und $k = 3$ den interpolierenden Spline bestimmen können.

6. Interpolation mit Polynomen und Splines



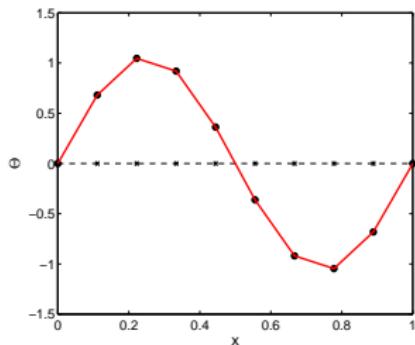
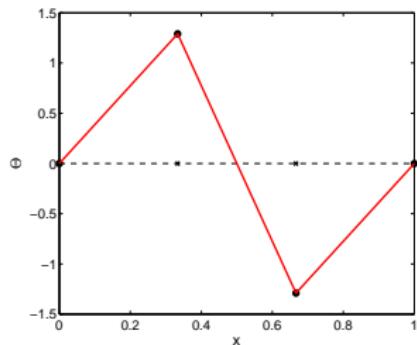
Beispiel 6.21 (Spline-Interpolierende ersten Grades)

Wir suchen eine Funktion s , so dass die folgenden drei Bedingungen erfüllt sind:

- ① Interpolation: $s(x_n) = y_n$ für $n = 0, \dots, N$
- ② Stückweises lineares Polynom: $s_n := s|_{[x_{n-1}, x_n]} \in \mathcal{P}_1$ für $n = 1, \dots, N$
- ③ Stetigkeit auf $[a, b]$: $s_n(x_n) = s_{n+1}(x_n)$ für $n = 1, \dots, N - 1$

Dies ist die stückweise lineare Interpolierende mit der expliziten Darstellung

$$s(x) = y_n + \frac{y_{n+1} - y_n}{x_{n+1} - x_n}(x - x_n) \text{ für alle } x \in [x_n, x_{n+1}] \text{ alle } n = 0, \dots, N - 1.$$





Beispiel 6.22 (Kubischer Spline)

Wir suchen eine Funktion s , so dass die folgenden drei Bedingungen erfüllt sind:

- ① Interpolation: $s(x_n) = y_n$ für $n = 0, \dots, N$
- ② Stückweises kubisches Polynom: $s_n := s|_{[x_{n-1}, x_n]} \in \mathcal{P}_3$ für $n = 1, \dots, N$
- ③ $s \in C^2([a, b])$: $s_n(x_n) = s_{n+1}(x_n)$ und $s'_n(x_n) = s'_{n+1}(x_n)$ und $s''_n(x_n) = s''_{n+1}(x_n)$ für $n = 1, \dots, N$

Für $x \in I_n := [x_{n-1}, x_n]$ machen wir den Ansatz

$$s_n(x) = s|_{I_n}(x) = \alpha_n + \beta_n(x - x_{n-1}) + \gamma_n(x - x_{n-1})^2 + \delta_n(x - x_{n-1})^3 \in \mathcal{P}_3(I_n).$$

Wir müssen also pro Teilintervall I_n die 4 Koeffizienten $\alpha_n, \beta_n, \gamma_n, \delta_n$ bestimmen, also insgesamt die $4N$ Koeffizienten $\{\alpha_n, \beta_n, \gamma_n, \delta_n\}_{n=1, \dots, N}$.



6. Interpolation mit Polynomen und Splines

Dazu nutzen wir die Definition des kubischen Splines:

- ① Interpolation: $s(x_n) = y_n$ für $0 \leq n \leq N$,
- ② Global C^2 : $s_n(x_n) = s_n(x_{n+1})$, $s'_n(x_n) = s'_{n+1}(x_n)$ und $s''_n(x_n) = s''_{n+1}(x_n)$ für $1 \leq n \leq N - 1$.

Das sind $(N + 1) + 3(N - 1) = 4N - 2$ linear unabhängige Bestimmungsgleichungen für die $4N$ unbekannten Koeffizienten. Es fehlen nur noch zwei weitere.

Definition 6.23 (Spline-Typ)

Ein kubischer Spline heißt

- ① **natürlicher Spline**, falls $s''(a) = s''(b) = 0$, d. h. keine Krümmung am Rand,
- ② **vollständiger Spline**, falls $s'(a) = A$ und $s'(b) = B$, d. h. explizite Vorgabe der ersten Ableitung am Rand,
- ③ **periodischer Spline**, falls $s'(a) = s'(b)$ und $s''(a) = s''(b)$



6. Interpolation mit Polynomen und Splines

Durch Einsetzen der Interpolationsbedingung $s_n(x_{n-1}) = y_{n-1}$ für $n = 1, \dots, N$ in den Ansatz

$$s_n(x) = \alpha_n + \beta_n(x - x_{n-1}) + \gamma_n(x - x_{n-1})^2 + \delta_n(x - x_{n-1})^3$$

erhalten wir sofort $\alpha_n = y_{n-1}$ für $n = 1, \dots, N$. Für die anderen Koeffizienten muss man etwas mehr rechnen, und die Bestimmungsgleichungen geschickt verknüpfen. Wir geben direkt das Ergebnis an, in zwei Schritten:

Satz 6.24 (Momentenansatz I)

Wir definieren die **Momente** $M_n = s''(x_n)$ und die Intervall-Längen $h_n = x_n - x_{n-1}$ für $n = 1, \dots, N$. Dann gilt für $n = 1, \dots, N$:

$$\begin{aligned}\alpha_n &= y_{n-1} & \beta_n &= \frac{y_n - y_{n-1}}{h_n} - \frac{2M_{n-1} + M_n}{6} h_n \\ \gamma_n &= \frac{M_{n-1}}{2} & \delta_n &= \frac{M_n - M_{n-1}}{6h_n}\end{aligned}$$

Verständnisübung: Verifizieren Sie exemplarisch $s(x_n) = y_n$ für $n = 0, \dots, N$, und $s''_n(x_n) = s''_{n+1}(x_n)$ für $n = 1, \dots, N-1$.



Satz 6.25 (Momentenansatz II)

Die Momente $M_n = s''(x_n)$ erfüllen für $n = 1, \dots, N - 1$ das LGS

$$h_n M_{n-1} + 2(h_n + h_{n+1})M_n + h_{n+1}M_{n+1} = 6 \left(\frac{y_{n+1} - y_n}{h_{n+1}} - \frac{y_n - y_{n-1}}{h_n} \right).$$

Das ist ein tridiagonales LGS für die $N - 2$ Hilfs-Unbekannten M_1, \dots, M_{N-1} . Beim natürlichen Spline haben wir per Definition zusätzlich $M_0 = M_N = 0$. Auch für den vollständigen und den periodischen Spline lassen sich zwei weitere Bestimmungsgleichungen konstruieren, in denen nur y_n -Werte vorkommen.

Zur Bestimmung der $4N$ Koeffizienten entscheiden wir uns also für einen Spline-Typ, lösen dann das tridiagonale Hilfssystem aus diesem Satz, und berechnen die Koeffizienten mit dem vorherigen Satz.



6. Interpolation mit Polynomen und Splines

Zur Verdeutlichung schreiben wir das LGS noch einmal ausführlich:

$$\begin{pmatrix} 2(h_1 + h_2) & h_2 \\ \ddots & \ddots \\ h_{N-1} & 2(h_{N-1} + h_N) \end{pmatrix} \begin{pmatrix} M_1 \\ \vdots \\ M_{N-1} \end{pmatrix} = \begin{pmatrix} 6\frac{y_2 - y_1}{h_2} - 6\frac{y_1 - y_0}{h_1} \\ \vdots \\ 6\frac{y_N - y_{N-1}}{h_N} - 6\frac{y_{N-1} - y_{N-2}}{h_{N-1}} \end{pmatrix}$$

Wir sehen wegen $h_n = x_n - x_{n-1}$ für $n = 1, \dots, N$: Die Matrix ist strikt diagonaldominant, also regulär, falls die Stützstellen paarweise verschieden sind. Deshalb existiert die kubische Spline-Interpolierende bei solchen Stützstellen immer, und ist eindeutig bestimmt durch die Vorgabe des Spline-Typs.

6. Interpolation mit Polynomen und Splines



Für die Stützstellen $\{x_0, \dots, x_N\}$ und Stützwerte $(y_0, \dots, y_N)^T \in \mathbb{R}^{N+1}$ können wir zusammengefasst einen kubischen Spline wie folgt bestimmen:

- ① Löse das LGS zur Bestimmung der Momente M_1, \dots, M_{N-1} , und bestimme je nach Spline-Typ M_0 und M_N .
- ② Bestimme die lokalen Koeffizienten $\{\alpha_n, \beta_n, \gamma_n, \delta_n\}$ von s_n für $n = 1, \dots, N$ aus y_0, \dots, y_N und M_0, \dots, M_N .

Hinweis: Die Gauß-Elimination für tridiagonale Systeme erfordert lediglich $\mathcal{O}(N)$ arithmetische Operationen. Wegen der strikten Diagonaldominanz ist keine Pivotisierung erforderlich, vgl. VL 1.

6. Interpolation mit Polynomen und Splines



Nach der einmaligen Berechnung der Koeffizienten erfolgt die Auswertung des Splines an einer beliebigen Stelle $x \in [a, b]$ in zwei Schritten:

- ① Bestimme das Teilintervall mit $x \in [x_{n-1}, x_n]$.
- ② Werte den Spline aus:

$$s_n(x) = \alpha_n + \beta_n(x - x_{n-1}) + \gamma_n(x - x_{n-1})^2 + \delta_n(x - x_{n-1})^3$$

Eine weniger rechenaufwendige Formulierung erhalten wir durch partielles Ausmultiplizieren:

$$s_n(x) = \alpha_n + (x - x_{n-1}) \left(\beta_n + (x - x_{n-1}) (\gamma_n + (x - x_{n-1}) \delta_n) \right)$$

Wir berechnen für ein gegebenes x zuerst $x - x_{n-1}$, multiplizieren mit δ_n , addieren γ_n , multiplizieren nochmals mit $x - x_{n-1}$, usw. Diese Formel zur Polynomauswertung heißt auch *Horner-Schema*.



Splines haben eine ausgesprochen nette praxisrelevante Eigenschaft:

Satz 6.26 (Minimalkrümmung)

Sei $\{x_0, \dots, x_N\}$ eine Stützstellenmenge von $[a, b]$ und seien $\{y_0, \dots, y_N\}$ die zugehörigen Stützwerte. Für den natürlichen kubischen Spline s und irgendeine beliebige zweimal stetig differenzierbare Funktion f mit $f(x_n) = y_n$ für $n = 0, \dots, N$ gilt:

$$\|s''\|_{2;[a,b]}^2 := \int_a^b |s''(x)|^2 dx \leq \int_a^b |f''(x)|^2 dx = \|f''\|_{2;[a,b]}^2$$

Der Ausdruck $\|s''\|_{2;[a,b]}$ ist ein Maß für die Gesamtkrümmung von s . Unter allen interpolierenden Funktionen hat der interpolierende natürliche kubische Spline also minimale Gesamtkrümmung. Ähnliches gilt für den vollständigen und periodischen Spline.

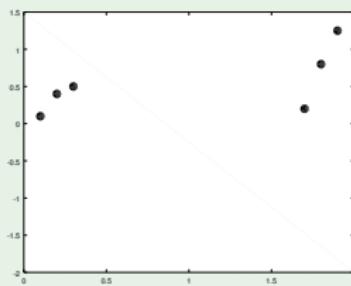


6. Interpolation mit Polynomen und Splines

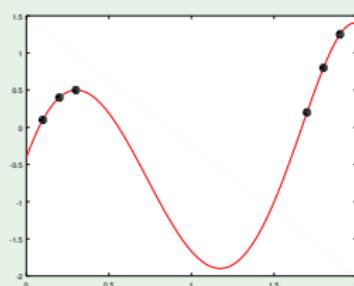
Beispiel 6.27 (Erklärung des Beispiels vom Beginn der VL)

Vergleich Polynominterpolation in \mathcal{P}_5 und Spline-Interpolation.

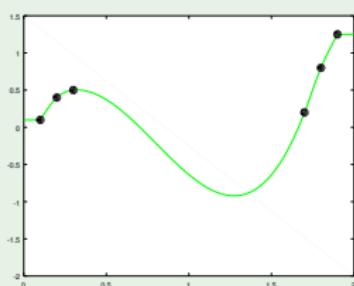
Messdaten



Polynominterpolation



Nat. kub. Spline



Wir sehen in der Tat geringere Krümmung.

Der Begriff „Spline“ stammt übrigens aus dem Schiffbau: Mit einem sogenannten Strakwerkzeug (englisch spline) können Oberflächen mit Minimalkrümmung konstruiert werden, d. h. Holzplanken optimal verformt werden. Es beruhigt, dass die Mathematik und die Simulation diese Vorgehensweise nachbildet.



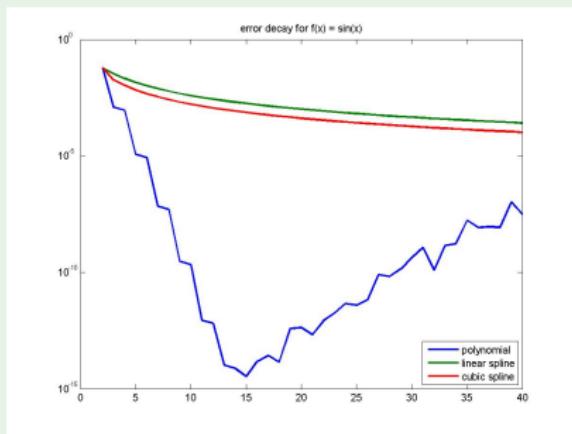
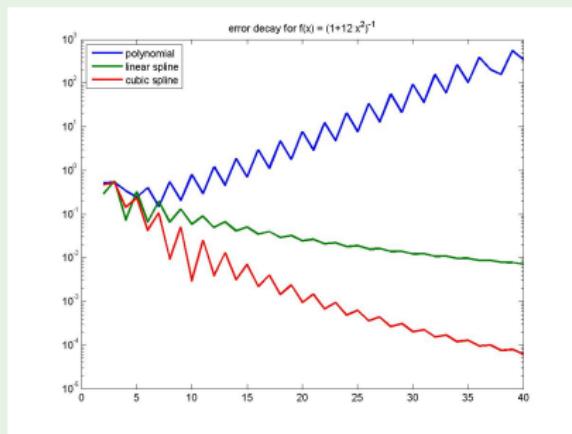
6. Interpolation mit Polynomen und Splines

Beispiel 6.28 (Fehlerkonvergenz-Verhalten)

Wir messen die Fehler $\max_{x \in [a,b]} |f(x) - p(x)|$ und $\max_{x \in [a,b]} |f(x) - s(x)|$ der Lagrange-Interpolierenden und der linearen und kubischen Splines in Abhängigkeit von der Anzahl der globalen Stützstellen N :

$$f(x) = (1 + 12x^2)^{-1}$$

$$f(x) = \sin(x)$$



Der kubische natürliche Spline zeigt insgesamt das robusteste Verhalten.



Zusammenfassung I

- Aus einer Stützstellenmenge aus $N + 1$ Punkten und Funktionswerten können interpolierende Funktionen berechnet werden.
- Die Polynominterpolation liefert ein eindeutiges Interpolationspolynom von Grad N . Dieses ist beliebig oft differenzierbar.
- Die Erhöhung der Anzahl von Interpolationspunkten führt zu einer Erhöhung des Polynomgrads, was u.U. zu Oszillationen führen kann.
- Die Verwendung der Tschebyschow-Stützstellen kann das Verhalten verbessern, wenn die Messungen passend generiert werden können.
- Bei der Polynominterpolation ist aus Konditionsgründen die Lagrange-Darstellung der Monom-Darstellung vorzuziehen.



Zusammenfassung II

- Um den hohen Polynomgrad zu vermeiden, kann die stückweise Interpolation angewendet werden. Die Spline-Interpolation ist eine spezielle stückweise Interpolation.
- Die Spline-Interpolation ersten Grades entspricht einem Polygonzug. Dieser ist trivial zu berechnen, aber nicht differenzierbar in den Stützstellen.
- Splines sind nicht beliebig glatt im Gegensatz zu globalen Polynomen. Allerdings ist die Glattheit in den Stützstellen steuerbar.
- Kubische Splines sind global zweimal stetig differenzierbar.
- Die Interpolation mit kubischen Splines ist durch Lösen eines tridiagonalen LGS erreichbar. Der interpolierende kubische Spline oszilliert deutlich weniger, tatsächlich ist die Minimalkrümmungseigenschaft beweisbar.



Hausaufgaben

- Einüben der Berechnung der Polynominterpolierenden, des linearen und des kubischen Splines zu gegebenen Mengen von Stützstellen und Stützwerten.
- Praktische Umsetzung der Polynom- und Spline-Interpolation in ViPLab.
- Für nächste Woche: Lagrange-Darstellung der Polynominterpolierenden verstehen, dito Eigenschaften der Lagrange-Basispolynome und optimale Stützstellen.
- Für nächste Woche: Integralbegriff wiederholen, insb. Linearität des Integrals, Hauptsatz der Differential- und Integralrechnung, Stammfunktionen



Beispielaufgaben



Interpolation in der Monombasis

Eine Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$ sei nur an den folgenden Auswertungspunkten bekannt:

i	0	1	2
x_i	1	2	3
$f(x_i)$	2	-1	2

Welchen Grad hat das eindeutig bestimmte Interpolationspolynom p mit $p(x_i) = f(x_i)$ für $i = 0, 1, 2$? Bestimmen Sie es in der Monombasis.

6. Interpolation mit Polynomen und Splines



Lösungshinweise: Das ist eine reine Rechenaufgabe, deshalb fallen Lösungshinweise unter Spoiler-Alert.

Ergebnis: Wir benötigen einen Ansatz für ein quadratisches Polynom:
 $p(x) = c_0 + c_1x + c_2x^2$. Das LGS zur Bestimmung der Koeffizienten in der Monombasis lautet:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \\ 2 \end{pmatrix}$$

So erhalten wir $p(x) = 3x^2 - 12x + 11$.



Interpolation in der Lagrangebasis

Bestimmen Sie für die Daten aus der vorherigen Aufgabe die Interpolierende in der Lagrange-Basis.

6. Interpolation mit Polynomen und Splines



Lösungshinweise: Das ist wieder eine reine Rechenaufgabe. Wir müssen die Basispolynome aufstellen und die gegebenen Stützstellen einsetzen. Man beachte, dass keine Vereinfachung des Polynoms gefordert ist

Ergebnis: Die Basispolynome lauten:

$$L_0(x) = \frac{1}{2}(x - 2)(x - 3)$$

$$L_1(x) = -(x - 1)(x - 3)$$

$$L_2(x) = \frac{1}{2}(x - 1)(x - 2)$$

Damit erhalten wir:

$$p(x) = 2L_0(x) - L_1(x) + 2L_2(x) = (x - 2)(x - 3) + (x - 1)(x - 3) + (x - 1)(x - 2)$$



Interpolation in beiden Basen

Weisen Sie nach, dass die Interpolierenden aus den vorherigen Aufgaben identisch sind.

6. Interpolation mit Polynomen und Splines

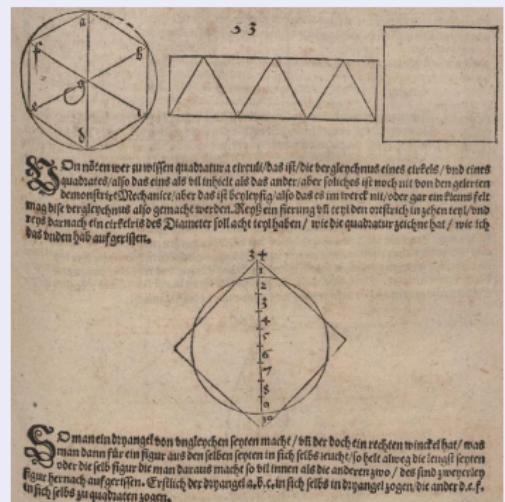


Lösungshinweise: Diese Aufgabe kann man schlau und weniger schlau lösen. Für den weniger schlauen Teil formen wir die Lagrange-Darstellung solange um, bis die Monomdarstellung da steht. Für die schlaue Antwort bemerken wir, dass die Interpolierende eindeutig bestimmt ist unabhängig von der Wahl der Basis. So sparen wir 30 Minuten Rechnerei.



7. Numerische Integration (Quadratur)

„Vonnoeten wäre zu wissen Quadratura circuli, das ist die Gleichheit eines Zirkels und eines Quadrates, also daß eines ebenso viel Inhalt hätte als das andere. Aber solches ist noch nicht von den Gelehrten demonstrirt. . . . Mechanice, das ist beiläufig, also daß es im Werk nicht oder nur um ein kleines fehlt, mag diese Gleichheit also gemacht werden. Reiß eine Vierung und teile den Ortsstrich in zehn Teile und reiße danach einen Zirkelriß, dessen Durchmesser acht Teile haben soll, wie die Quadratur deren 10; wie ich das unten aufgerissen habe.“



Albrecht Dürer zur Quadratur des Kreises (1525)



Motivation und Anwendungsbeispiele



7. Numerische Integration (Quadratur)

Wir beginnen mit offenen Problemen aus der HM23: Wie kann man ein Integral berechnen, wenn es keine Formel für die Stammfunktion gibt?

$$\int_a^b e^{-x^2} dx = ?$$

Die Funktion $f(x) = e^{-x^2}$ besitzt zwar beweisbar eine Stammfunktion, für diese existiert aber keine explizite Formel.

$$\int_0^{\frac{\pi}{2}} 4a \sqrt{1 - \cos^2(x)} \left(1 - \frac{b^2}{a^2}\right) dx = ?$$

Dieses Integral beschreibt für $a > b > 0$ den Umfang einer Ellipse, deren Halbachsen die Längen a und b haben. Auch hier besitzt der Integrand keine explizite Stammfunktion.



7. Numerische Integration (Quadratur)

In dieser VL lernen wir Methoden kennen, um solche Integrale numerisch zu approximieren. Die Anwendungsszenarien der numerischen Integration gehen allerdings noch viel weiter: In allen folgenden Vorlesungen zur numerischen Behandlung von Differentialgleichungen werden wir Integrale als Teilprobleme in einem „größeren“ Verfahren berechnen müssen.

Wenn wir algorithmisch denken, ergibt sich die zentrale Motivation dieser Vorlesung: Eine Implementierung, die eine Näherungslösung für eine Differentialgleichung berechnet, sollte nicht auf Teilproblemen basieren, für die wir Papier und Bleistift benötigen.



7. Numerische Integration (Quadratur)

Wir definieren die Problemstellung präzise:

Definition 7.1 (Approximation von Integralen, Quadratur)

Sei $D \subset \mathbb{R}^d$ ein Gebiet, $N \in \mathbb{N}$, und f eine integrierbare und punktweise auswertbare Funktion. Als **Numerische Integration (Quadratur)** bezeichnen wir die Berechnung einer Approximation $Q_D^N[f]$ an das Integral $I_D[f]$ mittels

$$I_D[f] := \int_D f(x) dx \quad \approx \quad Q_D^N[f] := \sum_{n=0}^N w_n f(x_n).$$

Dabei ist N ein Parameter zur Genauigkeitssteuerung, $Z_N := \{x_0, \dots, x_N\} \subset D$ sind paarweise verschiedene Stützstellen in D , und $\{w_0, \dots, w_N\} \subset \mathbb{R} \setminus \{0\}$ sind noch festzulegende Gewichte.

Der Wert $I_D[f]$ des Integrals wird also approximiert durch das gewichtete Aufsummieren von Funktionsauswertungen an geeigneten Stützstellen.



Newton-Cotes Quadratur



7. Numerische Integration (Quadratur)

Beispiel 7.2 (Integration von Polynomen)

Ist $f: [a, b] \rightarrow \mathbb{R}$ ein Polynom vom Grad N gegeben in der Monombasis, d. h.

$$f(x) = \alpha_0 + \alpha_1 x + \cdots + \alpha_N x^N \in \mathcal{P}_N,$$

so hat f die explizite Stammfunktion

$$F(x) = \alpha_0 x + \frac{\alpha_1}{2} x^2 + \cdots + \frac{\alpha_N}{N+1} x^{N+1},$$

wie wir durch Ableiten schnell verifizieren können. Deshalb ist das Integral

$$\int_a^b f(x) dx = F(b) - F(a)$$

einfach und insbesondere exakt aus den Koeffizienten $\{\alpha_0, \dots, \alpha_N\}$ und den Parametern $\{a, b, N\}$ berechenbar.

Das sollte man in der Nachbereitung explizit nachvollziehen.



7. Numerische Integration (Quadratur)

Naheliegende Idee

Approximiere $f: [a, b] \rightarrow \mathbb{R}$ durch ein geeignetes $p_N \in \mathcal{P}_N$ und verwende

$$Q_{[a,b]}^N[f] := I_{[a,b]}[p_N] = \int_a^b p_N(x) dx.$$

Nach VL 6 und dem vorherigen Beispiel ist klar: Ein geeignetes p_N erhalten wir durch Interpolation.

Die Approximation des Integrals über f ist also die **exakte** Integration eines geeigneten Polynoms $p_N \approx f$. Dabei müssen wir natürlich die Resultate aus VL 6 berücksichtigen. Wir diskutieren zunächst, ob wir $Q_{[a,b]}^N[f]$ immer in die gewünschte Form

$$Q_{[a,b]}^N[f] = \sum_{n=0}^N w_n f(x_n)$$

bringen können. Zur Notationsvereinfachung fixieren wir $[a, b]$ und schreiben $I[f] := I_{[a,b]}[f]$ und $Q^N[f] := Q_{[a,b]}^N[f]$.

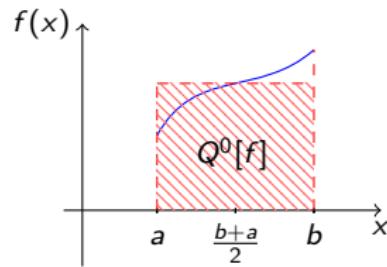
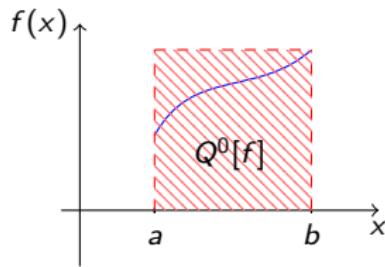
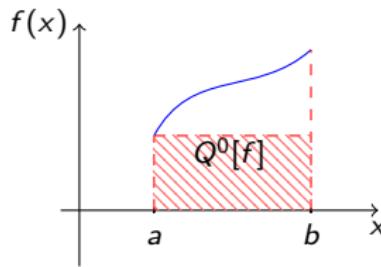
7. Numerische Integration (Quadratur)



Beispiel 7.3 (Konstante Approximation)

Für $N = 0$ gibt es drei natürliche Möglichkeiten für die einzige Stützstelle x_0 :

- ① Vorwärts-Rechtecksregel mit $\{x_0 = a\}$: $Q^0[f] = (b - a) f(a)$
- ② Rückwärts-Rechtecksregel mit $\{x_0 = b\}$: $Q^0[f] = (b - a) f(b)$
- ③ Mittelpunktsregel mit $\{x_0 = \frac{1}{2}(b + a)\}$: $Q^0[f] = (b - a) f\left(\frac{1}{2}(b + a)\right)$



Alle drei Quadraturformeln haben die gewünschte Gestalt.



7. Numerische Integration (Quadratur)

Wir vergleichen den absoluten **Quadraturfehler** $|Q^0[f] - I[f]|$ auf $[-1, 1]$:

	$\{x_0 = a\}$	$\{x_0 = b\}$	$\{x_0 = \frac{1}{2}(b + a)\}$
$f(x) = 3$	0	0	0
$f(x) = x$	2	2	0
$f(x) = x^2$	1.33	1.33	0.67
$f(x) = e^x$	1.61	3.09	0.35

Alle 3 Quadraturen integrieren konstante Funktionen, d. h. $p \in \mathcal{P}_0$ korrekt. Die letzte Quadratur integriert sogar lineare Funktionen, d. h. $p \in \mathcal{P}_1$ korrekt, und scheint insgesamt genauer zu sein. Beide Effekte untersuchen wir nun genauer, und konstruieren so systematisch Quadraturen mit mehr Stützstellen.



7. Numerische Integration (Quadratur)

Rückblick VL 6: Interpolationspolynom via Lagrange-Basis

Aufgrund der grauenhaften Kondition der Monombasis sollten zu einer Stützstellenmenge $Z_N := \{x_0, \dots, x_N\}$ von $[a, b]$ immer die Lagrange-Polynome $\{L_0, \dots, L_N\} \in \mathcal{P}_N$ mit

$$L_m(x) = \prod_{n=0, n \neq m}^N \frac{x - x_n}{x_m - x_n}$$

zur globalen Interpolation verwendet werden. Sie sind eine Basis von \mathcal{P}_N mit

$$L_m(x_n) = \delta_{mn} \quad \text{für alle } m, n = 0, \dots, N.$$

Das eindeutige Interpolationspolynom $p_N \in \mathcal{P}_N$ zu den Stützwerten $\{f(x_0), \dots, f(x_N)\}$ ist

$$p_N(x) = \sum_{n=0}^N f(x_n) L_n(x).$$



7. Numerische Integration (Quadratur)

Lagrange-Interpolierende: $p_N(x) = \sum_{n=0}^N f(x_n)L_n(x)$

Definition 7.4 (Interpolationsquadratur, Newton-Cotes Quadratur)

Zu einer Stützstellenmenge $\{x_0, \dots, x_N\}$ von $[a, b]$ definieren wir die Gewichte

$$w_n := \int_a^b L_n(x) dx \in \mathbb{R} \quad n = 0, \dots, N \quad (7.11)$$

und die Quadratur

$$Q^N[f] := \sum_{n=0}^N w_n f(x_n).$$

Wir sprechen von „Interpolationsquadratur“, weil die Gewichte gerade die exakte Integration der Lagrange-Basisfunktionen der Polynominterpolierenden von f sind. Diese Quadratur hat also die gewünschte Darstellung.



7. Numerische Integration (Quadratur)

Satz 7.5 (Exaktheit auf \mathcal{P}_N)

Die Quadraturformel aus Definition 7.4,

$$Q^N[f] := \sum_{n=0}^N w_n f(x_n), \quad w_n := \int_a^b L_n(x) dx \in \mathbb{R} \quad (n = 0, \dots, N)$$

ist **exakt** auf \mathcal{P}_N , d. h. es gilt

$$Q^N[p] = I[p] \quad \forall p \in \mathcal{P}_N.$$

Die Exaktheit auf \mathcal{P}_N ist der Schlüssel für $Q^N[f] \approx I[f]$ bei der interpolatorischen Quadratur. Es kann sogar Exaktheit auf \mathcal{P}_K für $K > N$ vorliegen, vgl. das Beispiel oben. Wir bezeichnen mit K den **maximalen Grad der Exaktheit** der Quadratur.



7. Numerische Integration (Quadratur)

Beweis von Satz 7.5: Jedes $p \in \mathcal{P}_N$ lässt sich in der Lagrange-Basis schreiben als

$$p(x) = \sum_{n=0}^N p(x_n) L_n(x),$$

wegen der δ_{mn} -Eigenschaft und weil jedes Polynom sich selbst interpoliert. Daraus folgt mit (7.11) und der Linearität des Integrals:

$$\begin{aligned} Q^N[p] &= \sum_{n=0}^N w_n p(x_n) = \sum_{n=0}^N p(x_n) \int_a^b L_n(x) dx \\ &= \int_a^b \sum_{n=0}^N p(x_n) L_n(x) dx = \int_a^b p(x) dx = I[p] \quad \square \end{aligned}$$

Je nach Anzahl und Wahl der Stützstellen $\{x_0, \dots, x_N\}$ erhalten wir unterschiedliche Quadraturformeln. Wir betrachten nun einige etablierte Beispiele, bei denen sich die allgemeine Formel meist vereinfacht darstellen lässt.

7. Numerische Integration (Quadratur)

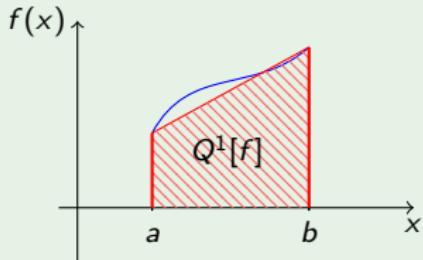


Beispiel 7.6 (Trapezregel)

Für $N = 1$ und $\{x_0 = a, x_1 = b\}$ erhalten wir die **Trapezregel**:

$$Q^1[f] = \frac{b-a}{2} (f(a) + f(b))$$

Die Trapezregel ist exakt für $p \in \mathcal{P}_1$.



Begründung: Nachrechnen liefert mit der Definition der Basispolynome:

$$w_0 = \int_a^b L_0(x) dx = \int_a^b \frac{x-b}{a-b} dx = \frac{1}{a-b} \int_a^b x-b dx = \frac{1}{a-b} \left. \frac{(x-b)^2}{2} \right|_a^b = \frac{b-a}{2}$$

$$w_1 = \int_a^b L_1(x) dx = \int_a^b \frac{x-a}{b-a} dx = \frac{1}{b-a} \left. \frac{(x-a)^2}{2} \right|_a^b = \frac{b-a}{2}$$

Somit ist obiges $Q^1[f] = w_0 f(x_0) + w_1 f(x_1)$ exakt auf \mathcal{P}_1 nach Satz 7.5. Dies ist sogar der maximale Grad der Exaktheit, d.h. $K = 1$.



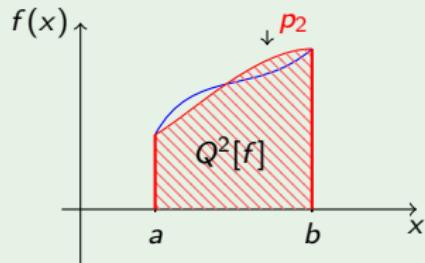
7. Numerische Integration (Quadratur)

Beispiel 7.7 (Simpsonregel)

Für $N = 2$ und $\{x_0 = a, x_1 = \frac{1}{2}(b + a), x_2 = b\}$ erhalten wir die **Simpsonregel**:

$$Q^2[f] = \frac{b - a}{6} (f(a) + 4f(\frac{1}{2}(b + a)) + f(b))$$

Die Simpsonregel ist exakt für $p \in \mathcal{P}_3$, also exakter als erwartet.



Begründung: Wir rechnen wieder einfach nach:

$$w_0 = \int_a^b L_0(x) dx = \frac{b - a}{6} = \int_a^b L_2(x) dx = w_2$$

$$w_1 = \int_a^b L_1(x) dx = \frac{2(b - a)}{3} = \frac{4(b - a)}{6}$$

Die Aussage zur Exaktheit diskutieren wir später.



7. Numerische Integration (Quadratur)

In der Praxis wäre es hilfreich, die Gewichte (die exakte Integration der Basispolynome) für ein Referenzintervall einmalig vorberechnen zu können, da die Auswertung eines Lagrange-Basispolynoms $\mathcal{O}(N)$ Operationen erfordert:

Satz 7.8 (Transformationssatz)

Sei \hat{Q}^N eine Quadratur auf $[0, 1]$ mit Stützstellen $\{\hat{x}_0, \dots, \hat{x}_N\}$ und Gewichten $\hat{w}_0, \dots, \hat{w}_N$, welche exakt auf \mathcal{P}_K ist. Dann ist die auf $[a, b]$ transformierte Quadratur $Q_{[a,b]}^N$ mit

$$x_n = a + \hat{x}_n(b - a), \quad w_n = (b - a)\hat{w}_n \quad n = 0, \dots, N \quad (7.12)$$

eine Quadratur auf $[a, b]$ mit Stützstellenmenge $Z_N = \{x_0, \dots, x_N\}$ und Gewichten $\{w_0, \dots, w_N\}$, welche ebenfalls exakt auf \mathcal{P}_K ist.

Wir müssen also tatsächlich die Stützstellen $\{\hat{x}_0, \dots, \hat{x}_N\}$ und die Gewichte $\{\hat{w}_0, \dots, \hat{w}_N\}$ **nur** auf $[0, 1]$ bestimmen, und (7.12) erlaubt die Umrechnung auf beliebige Intervalle, vgl. die Übungen.



7. Numerische Integration (Quadratur)

Beweis: Zu $f: [a, b] \rightarrow \mathbb{R}$ definieren wir $\hat{f}: [0, 1] \rightarrow \mathbb{R}$ durch

$$\hat{f}(\hat{x}) = f(a + \hat{x}(b - a)) \quad \hat{x} \in [0, 1].$$

Dann gilt mit der Substitution $x = a + \hat{x}(b - a)$ bzw. $dx = (b - a)d\hat{x}$

$$\int_a^b f(x) dx = (b - a) \int_0^1 \hat{f}(\hat{x}) d\hat{x}.$$

Damit folgt für $p \in \mathcal{P}_K$:

$$\begin{aligned} Q_{[a, b]}^N[p] &= \sum_{n=0}^N w_n p(x_n) = \sum_{n=0}^N (b - a) \hat{w}_n p(a + \hat{x}_n(b - a)) \\ &= (b - a) \sum_{n=0}^N \hat{w}_n \hat{p}(\hat{x}_n) = (b - a) \int_0^1 p(\hat{x}) d\hat{x} = \int_a^b p(x) dx \quad \square \end{aligned}$$



7. Numerische Integration (Quadratur)

Definition 7.9 (Newton-Cotes Quadratur auf $[0, 1]$)

Sei $N \in \mathbb{N}$ und $D = [0, 1]$. Die Quadratur $\hat{Q}^N[f] := \sum_{n=0}^N \hat{w}_n f(\hat{x}_n)$ mit den $N + 1$ äquidistanten Stützstellen und Gewichten

$$\hat{x}_n = \frac{n}{N} \quad n = 0, \dots, N \quad \text{und} \quad \hat{w}_n = \int_0^1 L_n(x) dx$$

gemäß Satz 7.5 heißt geschlossene **Newton-Cotes Quadratur**.

Mit dem Transformationssatz 7.8 können diese Formeln einfach auf beliebige Intervalle übersetzt werden, vgl. die Übungen. Die Quadraturen heißen „geschlossen“, da $a = 0$ und $b = 1$ Stützstellen sind.

Zu $N = 0$ ist die Mittelpunktsregel eine sogenannte **offene Newton-Cotes Quadratur**, da die Randpunkte nicht Stützstellen sind. Offene Newton-Cotes Formeln können analog allgemein definiert werden.



7. Numerische Integration (Quadratur)

Die Gewichte für die ersten geschlossenen Newton-Cotes Quadraturen lauten:

N	$\hat{w}_0, \dots, \hat{w}_N$								Name
1	$\frac{1}{2} \quad \frac{1}{2}$								Trapez-Regel
2	$\frac{1}{6} \quad \frac{4}{6} \quad \frac{1}{6}$								Simpson-Regel
3	$\frac{1}{8} \quad \frac{3}{8} \quad \frac{3}{8} \quad \frac{1}{8}$								Newton-3/8-Regel
4	$\frac{7}{90} \quad \frac{32}{90} \quad \frac{12}{90} \quad \frac{32}{90} \quad \frac{7}{90}$								Milne-Regel
5	$\frac{19}{288} \quad \frac{75}{288} \quad \frac{50}{288} \quad \frac{50}{288} \quad \frac{75}{288} \quad \frac{19}{288}$								6-Punkt-Regel
6	$\frac{41}{840} \quad \frac{216}{840} \quad \frac{27}{840} \quad \frac{272}{840} \quad \frac{27}{840} \quad \frac{216}{840} \quad \frac{41}{840}$								Weddle-Regel
7	$\frac{751}{17280} \quad \frac{3577}{17280} \quad \frac{49}{640} \quad \frac{2989}{17280} \quad \frac{2989}{17280} \quad \frac{49}{640} \quad \frac{3577}{17280} \quad \frac{751}{17280}$								
8	$\frac{989}{28350} \quad \frac{2944}{14175} \quad \frac{-464}{14175} \quad \frac{5248}{14175} \quad \frac{-454}{2835} \quad \frac{5248}{14175} \quad \frac{-464}{14175} \quad \frac{2944}{14175} \quad \frac{989}{28350}$								

Schöne Übersicht: <https://de.wikipedia.org/wiki/Newton-Cotes-Formeln>

Scharfes Hinsehen: Die Gewichte summieren sich für festes N zu Eins auf. Ab $N = 8$ treten negative Gewichte auf. Beides wird noch spannend. In der Praxis werden die Gewichte selbstverständlich einmalig vorberechnet (oder aus einer verlässlichen Quelle abgetippt) und in einer Tabelle abgespeichert.



7. Numerische Integration (Quadratur)

Beispiel 7.10 (Monome mit Newton-Cotes Quadratur)

Wir betrachten den Quadraturfehler $|\hat{Q}_{[0,1]}^N[p_k] - I[p_k]|$ mit $p_k(x) = (k+1)x^k$:

p_k	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = 6$
x^0	0	0	0	0	0	0
$2x^1$	0	0	0	0	0	0
$3x^2$	0.500000	0	0	0	0	0
$4x^3$	1.000000	0	0	0	0	0
$5x^4$	1.500000	0.041667	0.018519	0	0	0
$6x^5$	2.000000	0.125000	0.055556	0	0	0
$7x^6$	2.500000	0.239583	0.109053	0.002604	0.001467	0
$8x^7$	3.000000	0.375000	0.176955	0.010417	0.005867	0
$9x^8$	3.500000	0.523438	0.257202	0.025098	0.014240	0.000231
$10x^9$	4.000000	0.679688	0.347737	0.047363	0.027200	0.001157

Trick: Die Skalierung der Monome ist so gewählt, dass $I[p_k] = 1$, also sind die Fehler gleichzeitig absolut und relativ.
 Demo in ILIAS

Vermutung: Falls N ungerade ist, ist \hat{Q}^N exakt auf \mathcal{P}_N ; ansonsten ist \hat{Q}^N exakt auf \mathcal{P}_{N+1} . Hier kommt also K als maximal erreichbare Exaktheit ins Spiel.



Tatsächlich können wir die gewonnenen Vermutungen auch präzise beweisen:

Satz 7.11 (Newton-Cotes Quadraturen)

- ① Falls N ungerade ist, ist \hat{Q}^N exakt auf \mathcal{P}_N , ansonsten ist \hat{Q}^N exakt auf \mathcal{P}_{N+1} .
- ② Die Gewichte sind symmetrisch: $\hat{w}_n = \hat{w}_{N-n}$ für alle $n = 0, \dots, N$
- ③ Es gilt $\sum_{n=0}^N \hat{w}_n = 1$ für $n < 8$.

Das sieht man durch Einsetzen und Nachrechnen wie bei den Beispielen ein. Die Symmetrie und Eins-Summe folgt aus der Polynominterpolation, vgl. VL 6.

7. Numerische Integration (Quadratur)



Wir fassen die bisherigen Erkenntnisse zur Newton-Cotes Quadratur zusammen:

Die erreichbare Exaktheit scheint zwischen N und $N + 1$ bei Verwendung von \mathcal{P}_N zu schwanken. Für $N \geq 8$ sind einige Gewichte negativ. Negative Gewichte führen zu unerwünschten Effekten: Insbesondere kann für eine positive Funktion f (mit deshalb positivem Integral $I[f]$) nicht mehr garantiert werden, dass auch $Q^N[f]$ positiv ist. Die Numerik erfindet also unter Umständen „unphysikalische“ Werte.

Daraus ergeben sich zwei offene Fragen: Welche Exaktheit K können wir für festes N überhaupt maximal erreichen? Können wir negative Gewichte dabei vermeiden?



Gauß-Legendre Quadratur



7. Numerische Integration (Quadratur)

Wir stellen zunächst einige prinzipielle Überlegungen zur Exaktheit an. Wegen der Basiseigenschaft (vgl. VL 6) reicht es, nur die Monombasis des \mathcal{P}_K zu betrachten; das Referenzintervall $[0, 1]$ vereinfacht die Überlegungen:

- ① Die Exaktheit von \hat{Q}^N auf \mathcal{P}_K liefert $K + 1$ Bedingungen:

$$\hat{Q}^N[x^k] = \sum_{n=0}^N w_n x_n^k \stackrel{!}{=} \underbrace{\int_0^1 x^k \, dx}_{=I[x^k]} = \frac{1}{k+1} \quad k = 0, \dots, K \quad (7.13)$$

- ② Jede interpolatorische Quadratur \hat{Q}^N wird durch $2N + 2$ Parameter (\hat{x}_n, \hat{w}_n) , $n = 0, \dots, N$ beschrieben.
- ③ Gleichung (7.13) ist also ein **nichtlineares Gleichungssystem** mit $K + 1$ Gleichungen für die $2N + 2$ Parameter von \hat{Q}^N .

Ein kurzes Zählargument liefert: \hat{Q}^N ist bestenfalls exakt auf \mathcal{P}_K für $K + 1 = 2N + 2$, also $K = 2N + 1$.



7. Numerische Integration (Quadratur)

Das können wir ohne großen Aufwand beweisen:

Satz 7.12 (Maximale Exaktheit)

Es gibt keine Quadratur \hat{Q}^N , die exakt auf \mathcal{P}_{2N+2} ist.

Beweis: Wir nehmen an, dass \hat{Q}^N exakt auf \mathcal{P}_{2N+2} ist, und betrachten das spezielle Polynom $p(x) := (x - x_0) \cdots (x - x_N) \in \mathcal{P}_{N+1}$. Dann ist $p^2 \in \mathcal{P}_{2N+2}$ (klar) und (ausmultiplizieren) $p^2 \not\equiv 0$, weil wir paarweise verschiedene Stützstellen voraussetzen. Daher folgt

$$0 \quad \neq \quad I[p^2] = \int_0^1 p^2(x) dx = \hat{Q}^N[p^2] = \sum_{n=0}^N \hat{w}_n \underbrace{p^2(x_n)}_{=0} = 0,$$

also ein Widerspruch. □

Das Argument funktioniert genauso für beliebige Integrationsintervalle.



7. Numerische Integration (Quadratur)

Wie bei der optimalen Polynominterpolation ist auch bei der optimalen Quadratur die Wahl der Stützstellen die schlaue Stellschraube. Zur einfachen Darstellung ändern wir das Referenzintervall:

Definition 7.13 (Legendre-Polynome)

Auf $[-1, 1]$ heißt

$$\ell_m(x) := \frac{1}{2^m m!} \frac{\partial^m}{\partial x^m} \left((x^2 - 1)^m \right)$$

das m -te **Legendre-Polynom**.

Es gilt $\ell_m \in \mathcal{P}_m([-1, 1])$, das sehen wir durch Ableiten ein. Insbesondere hat ℓ_m auf $[-1, 1]$ m verschiedene Nullstellen.



7. Numerische Integration (Quadratur)

Es zeigt sich, dass die Nullstellen der Legendre-Polynome bei der Quadratur die gleiche Rolle spielen wie die Nullstellen der Tschebyschow-Polynome bei der Interpolation (vgl. VL 6):

Definition 7.14 (Gauß-Legendre Quadratur)

Auf $[-1, 1]$ seien $\{\bar{x}_0, \dots, \bar{x}_N\}$ die $N + 1$ verschiedenen Nullstellen des Legendre-Polynoms ℓ_{N+1} . Zu diesen Stützstellen seien $\bar{L}_n \in \mathcal{P}_N$ ($n = 0, \dots, N$) die Lagrange-Polynome auf $[-1, 1]$. Dann heißt die Quadratur

$$\bar{Q}^N[f] := \sum_{n=0}^N \underbrace{\int_{-1}^1 \bar{L}_n(x) \, dx}_{=: \bar{w}_n} f(\bar{x}_n)$$

Gauß-Legendre Quadratur.

Die Wahl der Gewichte impliziert mit Satz 7.5, dass die Quadratur mindestens exakt ist auf \mathcal{P}_N .



7. Numerische Integration (Quadratur)

Die ersten Gauß-Legendre Quadraturen auf $[-1, 1]$ lauten:

$N = 0 :$	$\bar{w}_n :$	2
	$\bar{x}_n :$	0
$N = 1 :$	$\bar{w}_n :$	1 1
	$\bar{x}_n :$	$-\sqrt{\frac{1}{3}}$ $\sqrt{\frac{1}{3}}$
$N = 2 :$	$\bar{w}_n :$	$\frac{5}{9}$ $\frac{8}{9}$ $\frac{5}{9}$
	$\bar{x}_n :$	$-\sqrt{\frac{3}{5}}$ 0 $\sqrt{\frac{3}{5}}$
$N = 3 :$	$\bar{w}_n :$	$\frac{18-\sqrt{30}}{36}$ $\frac{18+\sqrt{30}}{36}$ $\frac{18+\sqrt{30}}{36}$ $\frac{18-\sqrt{30}}{36}$
	$\bar{x}_n :$	$-\sqrt{\frac{3}{7} + \frac{2}{7}\sqrt{\frac{6}{5}}}$ $-\sqrt{\frac{3}{7} - \frac{2}{7}\sqrt{\frac{6}{5}}}$ $\sqrt{\frac{3}{7} - \frac{2}{7}\sqrt{\frac{6}{5}}}$ $-\sqrt{\frac{3}{7} + \frac{2}{7}\sqrt{\frac{6}{5}}}$
$N = 4 :$	$\bar{w}_n :$	$\frac{322-13\sqrt{70}}{900}$ $\frac{322+13\sqrt{70}}{900}$ $\frac{128}{225}$ $\frac{322+13\sqrt{70}}{900}$ $\frac{322-13\sqrt{70}}{900}$
	$\bar{x}_n :$	$-\frac{1}{3}\sqrt{5+2\sqrt{\frac{10}{7}}}$ $-\frac{1}{3}\sqrt{5-2\sqrt{\frac{10}{7}}}$ 0 $\frac{1}{3}\sqrt{5-2\sqrt{\frac{10}{7}}}$ $\frac{1}{3}\sqrt{5+2\sqrt{\frac{10}{7}}}$

Alle Stützstellen und Gewichte werden in der Praxis wieder einmalig vorberechnet (oder abgetippt) und tabelliert.



7. Numerische Integration (Quadratur)

Der nächste Satz zeigt, dass die Gauß-Legendre Quadratur unsere Ziele erfüllt:

Satz 7.15 (Gauß-Legendre Quadratur)

Sei \bar{Q}^N eine Gauß-Legendre Quadratur auf $[-1, 1]$ mit Stützstellenmenge $\{\bar{x}_0, \dots, \bar{x}_N\}$ und Gewichten $\{\bar{w}_0, \dots, \bar{w}_N\}$.

Dann gilt:

- ① Alle Gewichte sind positiv, d. h. $\bar{w}_n > 0$ für alle $n = 0, \dots, N$.
- ② \bar{Q}^N ist exakt auf \mathcal{P}_{2N+1} , also optimal.
- ③ Die Stützstellen und Gewichte sind symmetrisch, d. h. $\bar{x}_n = -\bar{x}_{N-n}$ und $\bar{w}_n = \bar{w}_{N-n}$ für $n = 0, \dots, N$.
- ④ $\sum_{n=1}^N \bar{w}_n = 2$ im Gegensatz zu Newton-Cotes Formeln

Beweis: Episches nachrechnen.





7. Numerische Integration (Quadratur)

Auch hier können wir beliebige Integrationsintervalle mit einem Transformationssatz in den Griff bekommen:

Satz 7.16 (Transformation der Gauß-Legendre Quadratur)

Sei \bar{Q}^N eine Gauß-Legendre Quadratur auf $[-1, 1]$ mit Stützstellenmenge $\{\bar{x}_0, \dots, \bar{x}_N\}$ und Gewichten $\{\bar{w}_0, \dots, \bar{w}_N\}$. Auf einem Intervall $[a, b]$ erhalten wir durch

$$x_n = \frac{1}{2}((b-a)\bar{x}_n + (b+a)) \quad \text{und} \quad w_n = \frac{1}{2}(b-a)\bar{w}_n \quad n = 0, \dots, N$$

die Stützstellen und Gewichte einer Quadratur $Q_{[a,b]}^N$, die ebenfalls exakt auf \mathcal{P}_{2N+1} ist.

Beweis: Das funktioniert analog zum Beweis von Satz 7.8, wenn wir die Transformation $[-1, 1] \rightarrow [a, b]$ mit

$$x = \frac{1}{2}((b-a)\bar{x} + (b+a))$$

verwenden.



7. Numerische Integration (Quadratur)

Beispiel 7.17 (Monome mit Gauß-Legendre Quadratur)

Quadraturfehler $|Q^N[p_k] - I[p_k]|$ für $p_k(x) = (k+1)x^k$ auf $[0, 1]$:

	$N = 0$	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = 6$
x^0	0	0	0	0	0	0	0
$2x^1$	0	0	0	0	0	0	0
$3x^2$	0.250000	0	0	0	0	0	0
$4x^3$	0.500000	0	0	0	0	0	0
$5x^4$	0.687500	0.027778	0	0	0	0	0
$6x^5$	0.812500	0.083333	0	0	0	0	0
$7x^6$	0.890625	0.157407	0.002500	0	0	0	0
$8x^7$	0.937500	0.240741	0.010000	0	0	0	0
$9x^8$	0.964844	0.326389	0.023875	0.000204	0	0	0
$10x^9$	0.980469	0.409722	0.044375	0.001020	0	0	0
$11x^{10}$	0.989258	0.487912	0.070981	0.002945	0.000016	0	0

Das Beispiel bestätigt: Die Gauß-Legendre Quadratur ist maximal exakt auf \mathcal{P}_{2N+1} . Die Fehler sind meist kleiner als bei den Newton-Cotes Quadraturen.



Zusammengesetzte Formeln



7. Numerische Integration (Quadratur)

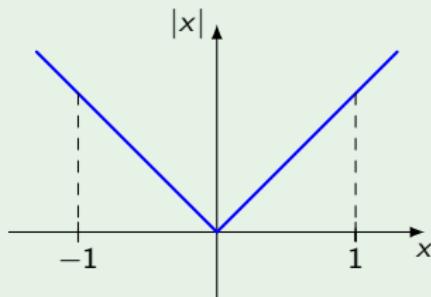
Wie bei der Polynominterpolation führt i. A. die Erhöhung des Polynomgrads N nicht unbedingt zu einer Verbesserung des Quadraturfehlers.

Beispiel 7.18 (Betragsfunktion)

Newton-Cotes und Gauß-Legendre Quadratur für $|x| : [-1, 1] \rightarrow \mathbb{R}$:

Newton-Cotes	
N	$ Q^N[f] - I[f] $
1	1.000000
2	0.333333
3	0.000000
4	0.022222
5	0.027778
6	0.076190

Gauß-Legendre	
N	$ Q^N[f] - I[f] $
0	1.000000
1	0.154701
2	0.139337
3	0.042535
4	0.055150
5	0.019894
6	0.029461
7	0.011528
8	0.018310





7. Numerische Integration (Quadratur)

Idee der zusammengesetzten Quadraturen

Das Integral $I_{[a,b]}[f]$ ist additiv. Das bedeutet: Wenn $a = y_0 < y_1 < \dots < y_M = b$ (die y haben nichts mit den Interpolations-Stützstellen aus VL 6 zu tun) eine Zerlegung von $[a, b]$ ist, dann gilt:

$$I_{[a,b]}[f] = \int_a^b f(x) dx = \sum_{m=1}^M \int_{y_{m-1}}^{y_m} f(x) dx = \sum_{m=1}^M (I_{[y_{m-1}, y_m]}[f])$$

Wir können also eine separate Quadratur auf jedem Teilintervall verwenden:

$$Q_{[y_{m-1}, y_m]}^N[f] \approx I_{[y_{m-1}, y_m]}[f]$$

Auch wenn die Quadraturen $Q_{[y_{m-1}, y_m]}^N$ auf den einzelnen Teilintervallen nach Konstruktion vollständig unabhängig sind, wird in der Regel eine feste Referenzquadratur \hat{Q}^N oder \bar{Q}^N auf das Teilintervall $[y_{m-1}, y_m]$ transformiert.



7. Numerische Integration (Quadratur)

Wir beschränken uns exemplarisch auf die Newton-Cotes Quadratur:

Definition 7.19 (Zusammengesetzte Quadratur)

Sei $N \in \mathbb{N}$ und \hat{Q}^N eine Newton-Cotes-Quadratur auf $[0, 1]$. Zu einer Zerlegung $a = y_0 < y_1 < \dots < y_M = b$ seien $Q_{[y_{m-1}, y_m]}^N$ für $m = 1, \dots, M$ die entsprechenden auf $[y_{m-1}, y_m]$ transformierten Quadraturen.

Die **zusammengesetzte Quadratur** $ZQ_{[a,b]}^N$ ist dann

$$I_{[a,b]}[f] \approx ZQ_{[a,b]}^N[f] = \sum_{m=1}^M Q_{[y_{m-1}, y_m]}^N[f].$$

Bemerkung: Für die geschlossenen Newton-Cotes Quadraturen gilt $\hat{x}_0 = 0$ und $\hat{x}_N = 1$. Deshalb wird der identische Wert $f(y_m)$ auf $[y_{m-1}, y_m]$ und $[y_m, y_{m+1}]$ verwendet.



7. Numerische Integration (Quadratur)

Algorithmus 7.20 : Zusammengesetzte Newton-Cotes Quadratur

input : $f: [a, b] \rightarrow \mathbb{R}$ stetig, Newton-Cotes Quadratur \hat{Q}^N auf $[0, 1]$ und eine Zerlegung $a = y_0 < \dots < y_M = b$
output : $ZQ_{[a,b]}^N[f]$

```
1  $ZQf = 0;$ 
2 for  $m = 1$  to  $M$  do
3    $Qf = 0;$ 
4   for  $n = 0$  to  $N$  do
5     transformiere  $x_n = y_{m-1} + (y_m - y_{m-1}) \hat{x}_n$ ;      % vgl. Satz 7.8
6      $Qf = Qf + \hat{w}_n f(x_n);$                                 % Aufsummieren
7   end
8    $ZQf = ZQf + (y_m - y_{m-1}) Qf;$ 
9 end
10 return  $ZQf;$ 
```

Die innere Schleife entspricht gerade der normalen Quadratur auf einem Teilintervall.



7. Numerische Integration (Quadratur)

Beispiel 7.21 (Zusammengesetzte Simpsonregel)

Es ist $N = 2$ und

$$\{\hat{x}_0 = 0, \hat{x}_1 = \frac{1}{2}, \hat{x}_2 = 1\} \quad \text{und} \quad \{\hat{w}_0 = \frac{1}{6}, \hat{w}_1 = \frac{4}{6}, \hat{w}_2 = \frac{1}{6}\}.$$

Sei $a = y_0 < \dots < y_M = b$ eine äquidistante Zerlegung von $[a, b]$ mit Gitterweite $h = \frac{(b-a)}{M}$. Dann lautet die zusammengesetzte Simpsonregel:

$$\begin{aligned} ZQ_{[a,b]}^2[f] &= \frac{h}{6} \sum_{m=1}^M \left(f(y_{m-1}) + 4f\left(\frac{1}{2}(y_m + y_{m-1})\right) + f(y_m) \right) \\ &= \frac{h}{6} \left(f(y_0) + 2 \sum_{m=1}^{M-1} f(y_m) + f(y_M) + 4 \sum_{m=1}^M f\left(\frac{1}{2}(y_m + y_{m-1})\right) \right) \end{aligned}$$

Die letzte Umformung reduziert die Anzahl der Auswertungen von f , und lohnt sich für den Fall, dass die Berechnung von f sehr teuer ist.



7. Numerische Integration (Quadratur)

Beispiel 7.22 (Betragsfunktion)

Zusammengesetzte Simpsonregel für $|x| : [-1, 1] \rightarrow \mathbb{R}$:

M	$ ZQ^2[f] - I[f] $
1	0.3333
3	0.0370
5	0.0133
7	0.0068

M	$ ZQ^2[f] - I[f] $
9	0.0041
11	0.0028
13	0.0020
15	0.0015

Für gerades M ist $y_{\frac{M}{2}} = 0$ und daher wird $|x|$ exakt integriert, da $|x|$ linear auf $[-1, 0]$ und $[0, 1]$ ist. Für ungerades, wachsendes M wird der Fehler kleiner, wenn auch langsam. Dies liegt an der fehlenden Regularität von $f: f \notin C^k([-1, 1])$ für alle $k \geq 1$.



Beispiel 7.23 (Zusammengesetzte Quadraturen für Polynome)

Der Fehler für die Funktion $f(x) = x^4$ auf $[0, 1]$ beträgt für die zusammengesetzte Simpsonregel:

M	Fehler $ I[f] - ZQ^2[f] $
1	0.00833333333333
2	0.00052083333333
3	0.000102880658436
4	0.000032552083333
5	0.000013333333333
6	0.000006430041152
7	0.000003470776065
8	0.000002034505208
9	0.000001270131586
10	0.000000833333333

Bei zusammengesetzten Formeln werden Polynome nicht mehr unbedingt exakt integriert. Das entspricht unserem Bauchgefühl zur zusammengesetzten Interpolation aus VL 6.



Mehrdimensionale Integration



7. Numerische Integration (Quadratur)

Beispiel 7.24 (Einheitsquadrat in \mathbb{R}^2)

Seien Q^N , $\tilde{Q}^{\tilde{N}}$ Quadraturen auf $[0, 1]$:

$$Q^N[g] = \sum_{n=0}^N w_n g(x_n) \approx I[g] \quad \text{und} \quad \tilde{Q}^{\tilde{N}}[g] = \sum_{n=0}^{\tilde{N}} \tilde{w}_n g(\tilde{x}_n) \approx I[g].$$

Für $f: D = [0, 1] \times [0, 1] \rightarrow \mathbb{R}$ können wir eine Tensorprodukt-Konstruktion verwenden:

$$\begin{aligned} I_D[f] &= \int_D f(x, y) dx dy = \int_0^1 \int_0^1 f(\textcolor{red}{x}, y) \textcolor{red}{dx} dy \\ &\approx \int_0^1 \sum_{n=0}^N \textcolor{red}{w}_n f(\textcolor{red}{x}_n, y) dy = \sum_{n=0}^N w_n \int_0^1 f(x_n, \textcolor{red}{y}) \textcolor{red}{dy} \\ &\approx \sum_{n=0}^N \sum_{\tilde{n}=0}^{\tilde{N}} w_n \tilde{w}_{\tilde{n}} f(x_n, y_{\tilde{n}}) =: Q_D^{N\tilde{N}}[f] \end{aligned}$$



7. Numerische Integration (Quadratur)

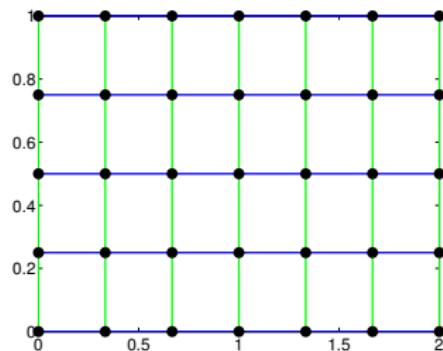
Die Quadratur $Q_D^{N\tilde{N}}$ hat $(N+1)(\tilde{N}+1)$ Stützstellen und Gewichte

$$\mathbf{x}_{n\tilde{n}} = [x_n, y_{\tilde{n}}]^T \quad \text{und} \quad w_{n\tilde{n}} = w_n \tilde{w}_{\tilde{n}} \quad n = 0, \dots, N, \quad \tilde{n} = 0, \dots, \tilde{N}.$$

Die Übertragung auf Rechtecke $D = [a_1, b_1] \times [a_2, b_2]$ erfolgt durch die eindimensionalen Transformationsformeln, ebenso die Verallgemeinerung auf Hyperquader $D = \prod_{i=1}^d [a_i, b_i] \subset \mathbb{R}^d$.

Beispiel: Stützstellen für

- $D = [0, 2] \times [0, 1]$
- $N = 6$
- $\tilde{N} = 4$





7. Numerische Integration (Quadratur)

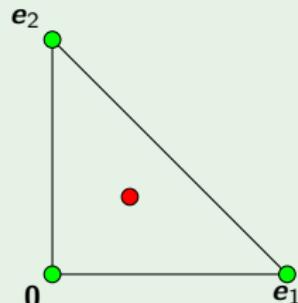
Beispiel 7.25 (Einheitsdreieck in \mathbb{R}^2)

Sei D das Einheitsdreieck mit Ecken $\mathbf{0}, \mathbf{e}_1, \mathbf{e}_2$. Quadraturen werden hier meist über baryzentrische Koordinaten anstelle von Tensorprodukten konstruiert:

Die Quadratur $Q_D^0[f] = \frac{1}{2}f\left(\frac{1}{3}, \frac{1}{3}\right)$ ist exakt für konstante Funktionen und heißt **Mittelpunktsregel für Dreiecke**.

Die Quadratur $Q_D^2[f] = \frac{1}{6}(f(\mathbf{0}) + f(\mathbf{e}_1) + f(\mathbf{e}_2))$ ist exakt für lineare Funktionen.

Die Übertragung auf Simplizes in 3D und Formeln höherer Ordnung ist einfach möglich.





Zusammenfassung

- Quadraturen erlauben die numerische Berechnung von Integralen.
- Interpolatorische Quadraturen sind durch die Stützstellen vollständig beschrieben. Die Gewichte sind die Integrale der Lagrange-Polynome.
- Newton-Cotes Formeln basieren auf äquidistanten Stützstellen, bei der Gauß-Legendre Quadratur werden die Nullstellen der Legendre-Polynome verwendet.
- Newton-Cotes Formeln sind exakt auf \mathcal{P}_N (N ungerade) bzw. \mathcal{P}_{N+1} (N gerade). Gauß-Legendre Formeln sind exakt auf \mathcal{P}_{2N+1} , dies ist optimal.
- Zusammengesetzte Quadraturen verbessern die Genauigkeit und Robustheit.
- Quadratur in höheren Raumdimensionen ist für spezielle Gebiete wie Quader und Simplizes möglich.



Hausaufgaben

- Üben Sie die Anwendung von Quadraturformeln ein, dies wird in VL 8 und bei Finiten Elementen (VL 11+12) eine große Rolle spielen.
- Trainieren Sie den Umgang mit den Transformationsformeln, insbesondere für zusammengesetzte Quadraturen.
- Wiederholen Sie gewöhnliche Differentialgleichungen und Anfangswertaufgaben, sowie Methoden zu ihrer Lösung wie die Trennung der Variablen.
- In ILIAS finden Sie eine aktualisierte Übersicht früherer Klausuraufgaben, die alle bisherigen Vorlesungen abdeckt.



Beispieldaufgaben



Auswertung von Quadraturformeln

Sei

$$f(x) := x^3 - 2x + 1.$$

Berechnen Sie mit der Mittelpunktsregel eine Näherung $Q_{[-1,1]}^0[f]$, und mit der Trapezregel eine Näherung $Q_{[-1,1]}^1[f]$, jeweils von

$$\int_{-1}^1 f(x) \, dx.$$



7. Numerische Integration (Quadratur)

Lösungshinweise: Weil das Polynom Grad 3 besitzt, können wir nicht von der Exaktheit profitieren und das Integral nicht mit den Methoden aus der HM123 ausrechnen. Stattdessen müssen wir die beiden Quadraturformeln tatsächlich auswerten.

Ergebnis: $Q_{[-1,1]}^0[f] = 2$ und $Q_{[-1,1]}^1[f] = 2$



7. Numerische Integration (Quadratur)

Transformationsformeln

Gegeben sei eine geschlossene Quadraturformel mit den Stützstellen $\hat{x}_0 = 0$, $\hat{x}_1 = \frac{1}{2}$, $\hat{x}_2 = 1$ und den Gewichten $\hat{w}_0 = \frac{1}{6}$, $\hat{w}_1 = \frac{2}{3}$, $\hat{w}_2 = \frac{1}{6}$.

Transformieren Sie diese Quadraturformel auf die Intervalle $I_1 = [-1, 1]$ und $I_2 = [1, 3]$.

Wie lautet die Quadraturformel für ein beliebiges Intervall $I = [a, b]$?



7. Numerische Integration (Quadratur)

Lösungshinweise: Hier kommt einfach Satz 7.8 zur Anwendung, einmal mit konkreten Zahlen und einmal „symbolisch“.

Ergebnis: Für das Intervall I_1 erhalten wir

$$\frac{1}{3} (f(-1) + 4f(0) + f(1)),$$

für das Intervall I_2

$$\frac{1}{3} (f(1) + 4f(2) + f(3)),$$

und allgemein

$$\frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right).$$



7. Numerische Integration (Quadratur)

Zusammengesetzte Quadraturen

Gegeben sei eine geschlossene Quadraturformel mit den Stützstellen $\hat{x}_0 = 0$, $\hat{x}_1 = \frac{1}{2}$, $\hat{x}_2 = 1$ und den Gewichten $\hat{w}_0 = \frac{1}{6}$, $\hat{w}_1 = \frac{2}{3}$, $\hat{w}_2 = \frac{1}{6}$.

Wie lautet die zusammengesetzte Quadraturformel für das Intervall $I = [-1, 3]$ bei äquidistanter Zerlegung und Gitterweite $h = 2$?

Stellen Sie die zusammengesetzte Quadraturformel für eine äquidistante Zerlegung $a = y_0 < \dots < y_M = b$ eines Intervalls $[a, b]$ mit der Gitterweite $h = \frac{b-a}{M}$ auf.



7. Numerische Integration (Quadratur)

Lösungshinweise: Wer aufgepasst hat, erkennt, dass wir lediglich die Ergebnisse aus der vorherigen Beispielaufgabe zusammenstöpseln müssen.

Ergebnis: Die Quadraturformel für das konkrete Beispiel lautet:

$$\frac{1}{3} (f(-1) + 4f(0) + 2f(1) + 4f(2) + f(3)),$$

und die allgemeine Formel lautet:

$$\frac{h}{6} \left(f(y_0) + 2 \sum_{m=1}^{M-1} f(y_m) + 4 \sum_{m=1}^M f\left(y_m + \frac{h}{2}\right) + f(y_M) \right).$$



8. Anfangswertprobleme



Martin Wilhelm Kutta

03.10.1867 — 25.12.1944

Professor für Mathematik an der Technischen Hochschule Stuttgart,
1912—1935



Motivation und Wiederholung HM3



8. Anfangswertprobleme

Beispiel 8.1 (Einfachstmögliche Beispiel)

Das denkbar einfachste Beispiel einer gewöhnlichen Differentialgleichung lautet:

$$\dot{u}(t) = 0 \quad \text{mit } u: \mathbb{R} \rightarrow \mathbb{R} \text{ stetig differenzierbar}$$

Mit dem Hauptsatz der Differential- und Integralrechnung sehen wir sofort, dass

$$u(t) = c, \quad c \in \mathbb{R}$$

für jedes c eine Lösung ist. Wir benötigen also noch eine Zusatzbedingung, um die Lösung eindeutig zu machen. Dazu geben wir einen sogenannten Anfangswert $u(t_0) = u_0$ vor, und erhalten das **Anfangswertproblem**: Finde $u \in C^1([0, T])$ mit

$$\dot{u}(t) = 0 \quad \text{für } t \in (t_0, T) \quad \text{und } u(t_0) = u_0.$$

Die Lösung ist die konstante Funktion $u(t) = u_0$ für alle $t \in [t_0, T[$.

Zur Verifizierung einer gegebenen Lösung müssen wir nur differenzieren.



8. Anfangswertprobleme

Beispiel 8.2 (Etwas nichttrivialer)

Wir betrachten die gewöhnliche Differentialgleichung

$$\dot{u}(t) = u(t)$$

mit den Lösungen $u(t) = c e^t$ für $c \in \mathbb{R}$. Die spezielle Lösung mit $c = 1$ erhalten wir wieder durch die Vorgabe eines Anfangswerts, diesmal $u(0) = 1$, so dass das vollständige Anfangswertproblem lautet: Finde $u \in \mathcal{C}^1(\mathbb{R})$ mit

$$\dot{u}(t) = u(t) \quad \text{für } t \in \mathbb{R} \quad \text{und} \quad u(0) = 1.$$

Die Lösung ist dann gerade die natürliche Exponentialfunktion $u(t) = e^t$.

Wir schreiben auch kompakt $\dot{u} = u$, $u(0) = 1$, wenn Definitions- und Wertebereich klar sind.



8. Anfangswertprobleme

Beispiel 8.3 (Einfaches Wachstumsmodell)

Wir bezeichnen mit $u(t)$ die Größe einer Population zum Zeitpunkt t . Uns interessiert die **relative Wachstumsrate**

$$r(t, u(t)) = \frac{\dot{u}(t)}{u(t)},$$

die die zeitliche Änderung der Population bezogen auf die Gesamtpopulation beschreibt. Ein besonders einfaches Modell erhalten wir, wenn wir $r = \alpha = \text{const}$ annehmen, d. h. die Population vermehrt bzw. reduziert sich proportional zu ihrer Größe: $\dot{u} = \alpha u$. Das zugehörige Anfangswertproblem lautet für den (aus Bequemlichkeit gewählten) Startzeitpunkt $t = 0$

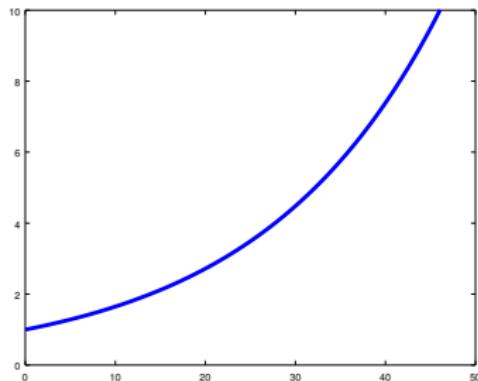
$$\dot{u}(t) = \alpha u(t) \quad \text{für } t > 0 \quad \text{und} \quad u(0) = u_0 > 0$$

mit der Lösung $u(t) = u_0 e^{\alpha t}$ und u_0 als Populationsgröße zum Startzeitpunkt.

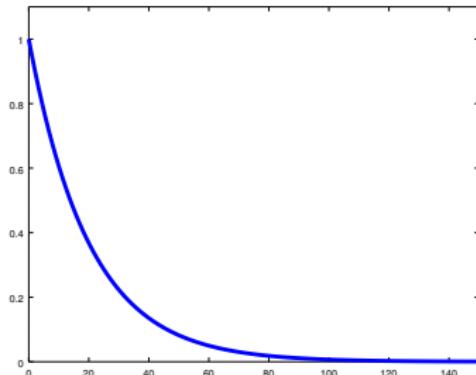


8. Anfangswertprobleme

$$\dot{u}(t) = \alpha u(t), \quad t > 0, \quad u(0) = u_0 = 1$$



$$\alpha = 0.05 : \lim_{t \rightarrow \infty} u(t) = \infty$$



$$\alpha = -0.05 : \lim_{t \rightarrow \infty} u(t) = 0$$

Links: Die Population wächst unbeschränkt. **Rechts:** Die Population stirbt aus.



8. Anfangswertprobleme

Beispiel 8.4 (Beschränktes Wachstum)

Wir bleiben bei der konstanten Wachstumsrate, und führen eine kritische Populationsgröße β ein:

$$r(t, u(t)) = \alpha(\beta - u(t))$$

r wird kleiner bei Annäherung von u an β , und bei Überschreiten der kritischen Größe wird r sogar negativ. Einsetzen in $r = \frac{\dot{u}}{u}$ liefert:

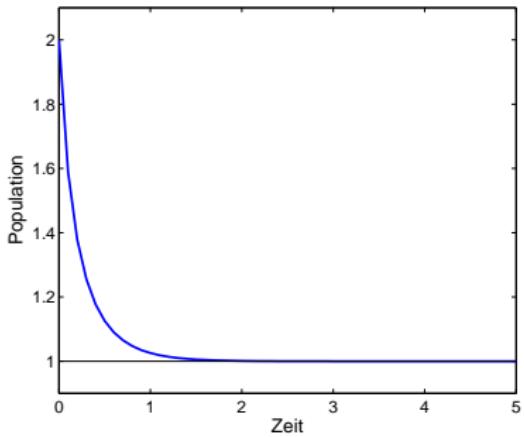
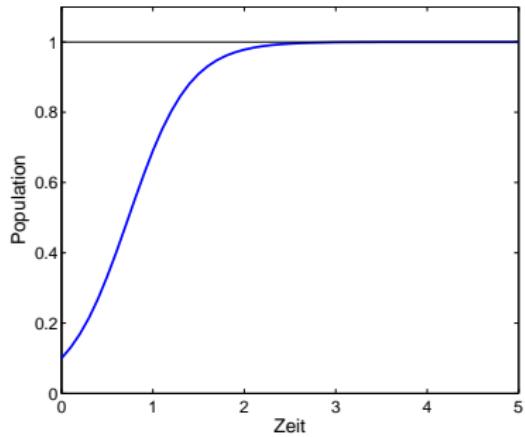
$$\dot{u}(t) = \alpha u(t)(\beta - u(t)) = \alpha \beta u(t) - \alpha u^2(t), \quad \text{für } t > 0 \quad \text{und} \quad u(0) = u_0 > 0.$$

Mit der Trennung der Veränderlichen (Separation der Variablen, vgl. HM3) lässt sich die explizite Lösung bestimmen:

$$u(t) = \beta \frac{u_0}{u_0 + (\beta - u_0) e^{-\alpha t}}$$



8. Anfangswertprobleme



$\lim_{t \rightarrow \infty} u(t) = \beta$ für $\alpha = 3.0$ und $\beta = 1.0$ je nach Anfangswert u_0



8. Anfangswertprobleme

Beispiel 8.5 (Lotka-Volterra)

Ein semi-realistisches Modell für zwei interagierende, gekoppelte Populationen mit einer Räuber- und einer Beute-Spezies basiert auf vier Annahmen:

- ① Ohne Räuber vermehrt sich die Beute unbeschränkt mit Wachstumsrate α .
- ② Ohne Beute verhungern die Räuber mit konstanter Rate $-\gamma$.
- ③ Die Wachstumsrate α wird durch Räuber reduziert, proportional zur Räuber-Populationsgröße (Faktor β).
- ④ Die Sterberate $-\gamma$ wird durch Beute verringert, proportional zur Beute-Populationsgröße (Faktor δ).

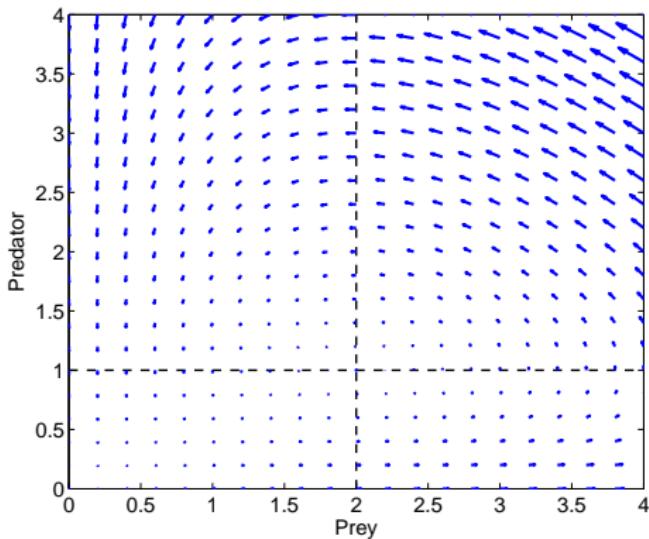
Dies führt auf das **Anfangswertsystem**

$$\dot{\mathbf{u}} := \begin{pmatrix} \dot{B} \\ \dot{R} \end{pmatrix} = \begin{pmatrix} \alpha B - \beta BR \\ -\gamma R + \delta BR \end{pmatrix} \mathbf{u} =: \mathbf{f}(\mathbf{u}), \quad t > 0 \quad \text{und} \quad \mathbf{u}(0) = \mathbf{u}_0 = \begin{pmatrix} B_0 \\ R_0 \end{pmatrix}$$

Im Allgemeinen ist keine explizite Lösungsformel bekannt.



8. Anfangswertprobleme



Numerisch bestimmtes Richtungsfeld für $\alpha = \beta = \delta = 1.0$ und $\gamma = 2.0$

Man kann zeigen:

Satz 8.6

Die Lösungen des Räuber-Beute Modells sind periodisch.



Theorie für Anfangswertprobleme



8. Anfangswertprobleme

Wir definieren zunächst ein gewöhnliches AWP erster Ordnung vernünftig:

Problem 8.7 (Anfangswertproblem, AWP)

Sei $\mathbf{f}: [0, T) \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ eine stetige Funktion und $\mathbf{u}_0 \in \mathbb{R}^d$ ein Anfangswert.
Gesucht ist eine Funktion $\mathbf{u}: [0, T) \rightarrow \mathbb{R}^d$, die

$$\dot{\mathbf{u}}(t) = \mathbf{f}(t, \mathbf{u}(t)), \quad \text{für } t \in (0, T) \quad \text{und} \quad \mathbf{u}(0) = \mathbf{u}_0$$

erfüllt. Diese Aufgabenstellung heißt **Anfangswertproblem**. Ist \mathbf{f} linear in \mathbf{u} , sprechen wir von einem **linearen** AWP, sonst von einem **nichtlinearen** AWP. Wenn die rechte Seite \mathbf{f} nicht von der Zeitvariablen t abhängt, heißt das Anfangswertproblem **autonom**.

Die ausführliche Formulierung des AWP lautet:

$$\begin{pmatrix} \dot{u}_1(t) \\ \vdots \\ \dot{u}_d(t) \end{pmatrix} = \begin{pmatrix} f_1(t, u_1(t), \dots, u_d(t)) \\ \vdots \\ f_d(t, u_1(t), \dots, u_d(t)) \end{pmatrix} \quad t \in (0, T), \quad \mathbf{u}(0) = \begin{pmatrix} u_{0,1} \\ \vdots \\ u_{0,d} \end{pmatrix}$$



8. Anfangswertprobleme

Zur Vereinfachung betrachten wir immer den Startzeitpunkt $t_0 = 0$. Für den Endzeitpunkt ist auch $T = \infty$ zulässig, $[0, T]$ diskutieren wir gleich.

Die bisherigen Beispiele lassen sich wie folgt einordnen: Das einfache Wachstumsmodell mit konstanter Rate ist linear und autonom, weil die rechte Seite $f: [0, T] \times \mathbb{R} \rightarrow \mathbb{R}$ mit $f(t, u) = \alpha u$ nur indirekt von t abhängt. Das Modell beschränkten Wachstums ist nichtlinear und autonom: $f(t, u)$ beinhaltet u und u^2 . Das Lotka-Volterra Modell ist ein nichtlineares autonomes System. Ein Beispiel für ein nichtautonomes (nichtlineares) Problem ist Beispiel 8.10 weiter unten.

Anfangswertprobleme, in denen höhere als erste Ableitungen vorkommen, können immer auf Systeme von Anfangswertproblemen erster Ordnung (d. h. nur mit ersten Ableitungen) zurückgeführt werden. → Übung und HM3



8. Anfangswertprobleme

Die Existenz und Eindeutigkeit von Lösungen ist allgemein hochgradig nichttrivial:

Beispiel 8.8 (Nichteindeutigkeit)

$f(u) = \sqrt{u}$, $u(0) = 0$: Das AWP besitzt die beiden Lösungen

$$u(t) \equiv 0 \quad \text{und} \quad u(t) = \frac{1}{4}t^2.$$

Beispiel 8.9 (Lokale Existenz)

$f(u) = u^2$, $u(0) = u_0 > 0$: Die Lösung des AWP existiert nur lokal:

$$u(t) = \frac{u_0}{1 - tu_0} \quad \text{für} \quad t < \frac{1}{u_0}$$

Beispiel 8.10 (Keine geschlossene Formel)

$f(t, u) = t^2 + u^2$, $u(0) = u_0 > 0$: Die Lösung des AWP existiert, es ist aber keine explizite Formel für die Lösung bekannt.



8. Anfangswertprobleme

Die Existenz- und Eindeutigkeitstheorie lässt sich in drei Resultate gießen:

Satz 8.11 (Satz von Peano)

Wenn f stetig auf $(0, T) \times \mathbb{R}^d$ ist, dann existiert ein $\delta > 0$, so dass Problem 8.7 eine lokale Lösung auf $[-\delta, \delta]$ besitzt.

Der Satz von Peano benötigt die Minimalvoraussetzung der Stetigkeit, um den Hauptsatz der Differential- und Integralrechnung (HDI) anwenden zu können. Er liefert die Existenz einer lokalen Lösung in einer Umgebung um den Anfangswert, aber nicht die Eindeutigkeit.



8. Anfangswertprobleme

Satz 8.12 (Satz von Picard-Lindelöf, einfache Version)

Wenn f Lipschitz-stetig und beschränkt auf $(0, T) \times \mathbb{R}^d$ ist, dann existiert ein $\delta > 0$, so dass Problem 8.7 eine eindeutige lokale Lösung auf $[-\delta, \delta]$ besitzt.

Erinnerung: Eine Funktion heißt Lipschitz-stetig, wenn sie sich nur beschränkt ändern kann: Es muss eine Konstante L existieren mit $|f(x_1) - f(x_2)| \leq L |x_1 - x_2|$ für alle x_1, x_2 . Wenn wir mehr Voraussetzungen investieren, können wir also die lokal eindeutige Existenz der Lösung in einer Umgebung um den Anfangswert garantieren. Das stimmt optimistisch.

In Beispiel 8.8 ist die Wurzelfunktion nicht Lipschitz-stetig, vgl. HM123. Deshalb dürfen wir keine Eindeutigkeit erwarten. In der Tat existieren zwei Lösungen.



8. Anfangswertprobleme

Satz 8.13 (Fortsetzungssatz, einfache Version)

Unter den Voraussetzungen des Satzes von Picard-Lindelöf lässt sich die lokal eindeutig existierende Lösung immer auf das maximale Existenzintervall fortsetzen, auf dem die Lösung sinnvoll definiert ist.

In Beispiel 8.9 garantiert der Fortsetzungssatz die Eindeutigkeit der Lösung für das komplette Existenzintervall $[0, \frac{1}{u_0})$, aber eben nicht weiter.

Allgemein bekommen wir, falls die Lösung zum Zeitpunkt $t = T < \infty$ definiert ist, eine Darstellung bis zu einem gegebenen Endzeitpunkt. Das ist oft wichtig in der Praxis bei Evolutionen wie den Populationsmodellen.



8. Anfangswertprobleme

Wegen der letzten Bemerkung können wir ruhigen mathematischen Gewissens fordern:

Generalvoraussetzung

Wir nehmen im Folgenden an, dass das allgemeine Anfangswertproblem (Problem 8.7) eine eindeutige Lösung $\mathbf{u}: [0, T] \rightarrow \mathbb{R}^d$ besitzt für den Startzeitpunkt $t = 0$ und einen gegebenen Endzeitpunkt $T < \infty$.

Wir setzen die Gültigkeit dieser Annahme ab jetzt immer voraus, ohne dies explizit dazuschreiben.

Wird ein technischer/naturwissenschaftlicher Prozess **richtig** durch ein AWP modelliert, so ist die Generalvoraussetzung meist erfüllt.

Das nächste Resultat ist nun die Grundlage für die numerische Approximation von Lösungen.



8. Anfangswertprobleme

Satz 8.14 (Volterra-Integralgleichung)

Wenn \mathbf{u} eine Lösung des Anfangswertproblems 8.7 auf dem Intervall $[0, T]$ ist, dann ist \mathbf{u} auch eine Lösung der **Volterra-Integralgleichung**

$$\mathbf{u}(t) = \mathbf{u}_0 + \int_0^t \mathbf{f}(s, \mathbf{u}(s)) ds \quad (8.14)$$

Wenn umgekehrt \mathbf{u} eine stetige Lösung der Volterra-Integralgleichung ist, dann ist \mathbf{u} auch Lösung des AWP aus Definition 8.7.

Die „Hinrichtung“ ist gerade die Anwendung des HDI, die Rückrichtung erfordert mehr Mathematik. Wichtig für das Verständnis ist: Mit der Volterra-Integralgleichung haben wir die Lösung eines AWP zurückgeführt auf die Bestimmung einer Stammfunktion von \mathbf{f} . Für die numerische Bestimmung einer Lösung müssen wir also „nur noch“ ein Integral numerisch approximieren.



Einschrittverfahren



8. Anfangswertprobleme

Ausgangspunkt unserer Überlegungen ist die Volterra-Integralgleichung, und die naheliegende Idee, diese Gleichung numerisch zu integrieren (vgl. VL 7). Wir beginnen mit dem einfachsten Fall:

Beispiel 8.15 (Autonome skalarwertige Systeme)

Wir betrachten $\dot{u}(t) = f(u(t))$ für $t \in [0, T]$, $f: \mathbb{R} \rightarrow \mathbb{R}$ und den Anfangswert $u(0) = u_0$. Die Lösung dieses autonomen AWP lässt sich schreiben als:

$$u(t) = u_0 + \int_0^t f(s) \, ds \quad \text{für } t \in [0, T]$$

Zur Bestimmung einer numerischen Approximation der Lösung verwenden wir numerische Integration, vgl. VL 7:

$$u(t) \approx u_0 + ZQ[f]$$

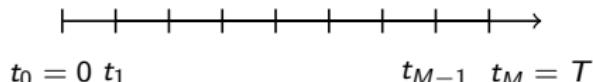
Erinnerung: Quadratur $Q[f]$, zusammengesetzte Quadratur $ZQ[f]$ auf entsprechenden Referenzintervallen.



8. Anfangswertprobleme

$$\mathbf{u}(t) = \mathbf{u}_0 + \int_0^t \mathbf{f}(s) \, ds \quad \approx \quad \mathbf{u}_0 + ZQ[\mathbf{f}]$$

Für die praktische Umsetzung verwenden wir eine zusammengesetzte Newton-Cotes Quadratur auf einem (groben) Zeitgitter: Wir wählen $h_M = \frac{T}{M}$ und setzen $t_m = mh_M$, $m = 0, \dots, M$:



Ausgehend von $\mathbf{U}_0 := \mathbf{u}_0$ bestimmen wir iterativ Approximationen $\mathbf{U}_m \approx \mathbf{u}(t_m)$ durch die numerische Quadratur des Integrals

$$\mathbf{U}_m := \mathbf{U}_{m-1} + Q_{[t_{m-1}, t_m]}^N[\mathbf{f}] = \mathbf{U}_{m-1} + h_M \sum_{n=0}^N \hat{w}_n \mathbf{f}(s_n) \quad m = 1, \dots, M$$

Der Faktor h stammt aus Transformationssatz 7.8, s. nächste Folie.

mit einer auf $[t_{m-1}, t_m]$ transformierten $[0, 1]$ -Quadratur \hat{Q}^N mit $N + 1$ **zusätzlichen** transformierten Stützstellen s_n im Intervall $[t_{m-1}, t_m]$.

Die Punktmenge $\{\mathbf{U}_0, \dots, \mathbf{U}_M\}$ ist die Approximation der gesuchten Lösung.



8. Anfangswertprobleme

Erinnerung: Eine Newton-Cotes Quadratur \hat{Q}^N auf $[0, 1]$,

$$\hat{Q}^N[f] = \sum_{n=0}^N \hat{w}_n f(\hat{s}_n)$$

mit den Gewichten \hat{w}_n und den äquidistanten Stützstellen $\hat{s}_n = \frac{n}{N}$ auf $[0, 1]$, wird mittels

$$w_n = h_M \hat{w}_n, \quad s_n = t_{m-1} + h_M \hat{s}_n \quad n = 0, \dots, N$$

auf das Intervall $[t_{m-1}, t_m]$ transformiert, und wir erhalten so:

$$U_m := U_{m-1} + Q_{[t_{m-1}, t_m]}^N[f] = U_{m-1} + h_M \sum_{n=0}^N \hat{w}_n f(s_n)$$

Die zusammengesetzte Quadratur und damit die Approximation des AWP steckt also in der Iteration über m , und die Quadratur auf dem Teilintervall nutzt die $N + 1$ tabellierten Gewichte und Stützstellen der $[0, 1]$ -Quadratur, transformiert auf $[t_{m-1}, t_m]$. Die Vorgehensweise funktioniert vollkommen analog für eine transformierte Gauß-Legendre Quadratur \bar{Q}_N , vgl. Satz 7.16.



8. Anfangswertprobleme

Allgemeiner Fall

Für den nichtautonomen Fall $\mathbf{f} = \mathbf{f}(t, \mathbf{u}(t))$ ergibt sich auf ähnliche Weise:

$$\mathbf{U}_m \approx \mathbf{U}_{m-1} + h_M \sum_{n=0}^N \hat{w}_n \mathbf{f}(s_n, \mathbf{u}(s_n))$$

Hier können wir dummerweise nicht direkt eine Quadratur ansetzen, weil die Werte $\mathbf{f}(s_n, \mathbf{u}(s_n))$ für $s_n \in [t_{m-1}, t_m]$ potentiell unbekannt sind.

Wir erhalten einen ganzen Zoo verschiedener Verfahren, je nachdem welche lokalen Quadraturen wir wählen und welche diskreten Werte $\mathbf{F}_n \approx \mathbf{f}(s_n, \mathbf{u}(s_n))$ wir zur Vervollständigung heranziehen.

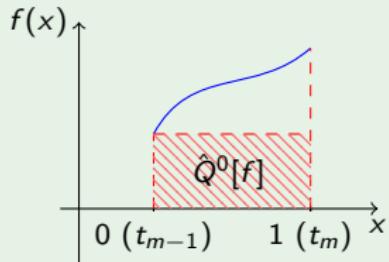
Wir betrachten nun ausgewählte Beispiele, die diese Ideen umsetzen. Für die ersten beiden Beispiele mit $N = 0$ verwenden wir offene Newton-Cotes Quadraturen.



8. Anfangswertprobleme

Beispiel 8.16 (Explizites Euler-Verfahren)

Wir nutzen die offene Vorwärts-Rechtecksregel \hat{Q}^0 , siehe VL 7. Hier ist $\hat{s}_0 = 0$ und deshalb $s_0 = t_{m-1}$, sowie $\hat{w}_0 = 1$ also $w_0 = h_M$, vgl. die Abbildung. Wir erhalten



$$\mathbf{u}(t_m) = \mathbf{u}(t_{m-1}) + \int_{t_{m-1}}^{t_m} \mathbf{f}(s, \mathbf{u}(s)) ds \approx \mathbf{u}(t_{m-1}) + h_M \mathbf{f}(t_{m-1}, \mathbf{u}(t_{m-1}))$$

und somit die Iterationsvorschrift des **expliziten Euler-Verfahrens**:

$$\mathbf{U}_m = \mathbf{U}_{m-1} + h_M \mathbf{f}(t_{m-1}, \mathbf{U}_{m-1})$$

Diese Formel funktioniert im autonomen und nichtautonomen Fall, weil $\mathbf{u}(t_{m-1})$ in der vorherigen Iteration bereits berechnet wurde. Wir können den Wert $\mathbf{f}(t_{m-1}, \mathbf{u}(t_{m-1}))$ direkt bestimmen, durch eine einzige **Auswertung** von \mathbf{f} .



8. Anfangswertprobleme

Algorithmus 8.17 : Explizites Euler-Verfahren

input : $f: [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ stetig, $\mathbf{u}_0 \in \mathbb{R}^d$, $M \in \mathbb{N}$

output : $\{\mathbf{U}_0, \dots, \mathbf{U}_M\}$ als Approximation von
 $\{\mathbf{u}(0) = \mathbf{u}(t_0), \dots, \mathbf{u}(T) = \mathbf{u}(t_M)\}$

- 1 Berechne $h_M = \frac{T}{M}$;
 - 2 Setze $\mathbf{U}_0 = \mathbf{u}_0$;
 - 3 **for** $m = 1$ **to** M **do**
 - 4 $t_{m-1} = (m - 1) \cdot h_M$;
 - 5 $\mathbf{U}_m = \mathbf{U}_{m-1} + h_M \cdot f(t_{m-1}, \mathbf{U}_{m-1})$;
 - 6 **end**
-

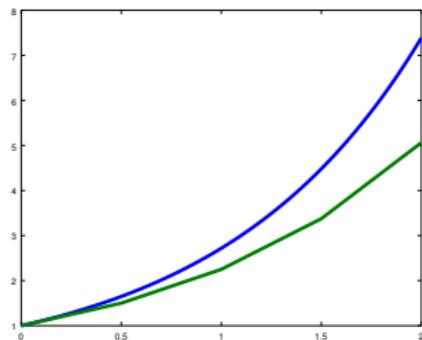
Der Algorithmus funktioniert analog, wenn statt der Anzahl der Schritte M die Schrittweite h_M vorgegeben wird, dazu ist Zeile 1 trivial zu ändern. In der letzten Iteration gilt $m = M$, also $m \cdot h_M = T$. Die Menge $\{\mathbf{U}_0, \dots, \mathbf{U}_M\}$ der Approximationen können wir einfach plotten.



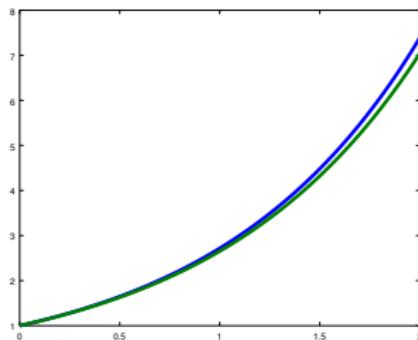
8. Anfangswertprobleme

Beispiel: Ergebnisse des expliziten Euler-Verfahrens für Bsp. 8.3 (unbeschränktes Wachstumsmodell)

$$h_M = 0.5$$



$$h_M = 0.05$$



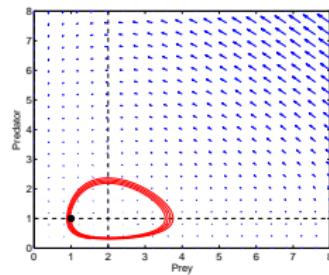
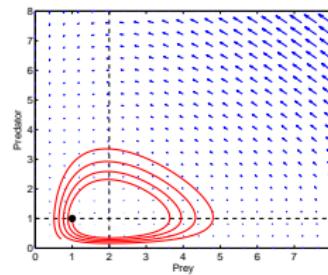
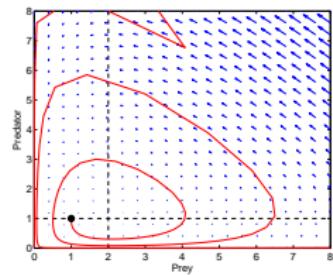
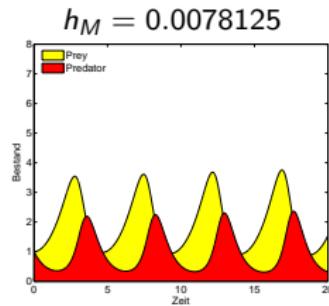
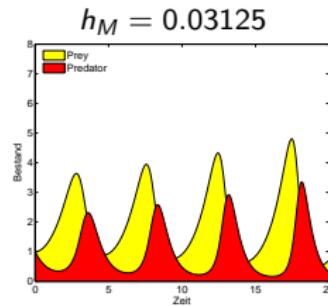
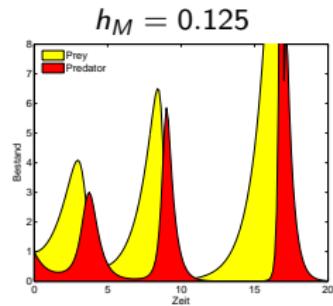
VIPLab-Demo in Ilias

Für feine Schrittweiten h_M sehen wir eine gute Approximation (grün) der exakten Lösung (blau).



8. Anfangswertprobleme

Beispiel: Ergebnisse des expliziten Euler-Verfahrens für Bsp. 8.5 (Lotka-Volterra)



Für kleine h_M sind die Ergebnisse gut, die Explosion für große h_M diskutieren wir später.

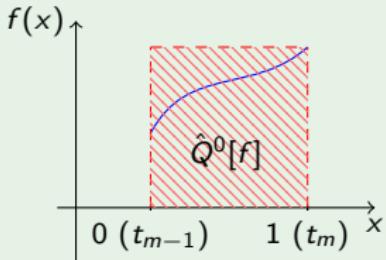


8. Anfangswertprobleme

Beispiel 8.18 (Implizites Euler-Verfahren)

Wir nutzen die offene Rückwärts-Rechtecksregel, vgl. VL 7. Hier ist $\hat{s}_0 = 1$ und deshalb $s_0 = t_m$, sowie $\hat{w}_0 = 1$ also $w_0 = h_M$.

Wir erhalten



$$\mathbf{u}(t_m) = \mathbf{u}(t_{m-1}) + \int_{t_{m-1}}^{t_m} \mathbf{f}(s, \mathbf{u}(s)) ds \approx \mathbf{u}(t_{m-1}) + h_M \mathbf{f}(t_m, \mathbf{u}(t_m)).$$

Wir approximieren wieder $\mathbf{u}(t_m)$ durch \mathbf{U}_m und erhalten das **implizite Euler-Verfahren**

$$\mathbf{U}_m = \mathbf{U}_{m-1} + h_M \mathbf{f}(t_m, \mathbf{U}_m).$$



8. Anfangswertprobleme

Die Iterationsvorschrift des impliziten Euler-Verfahrens

$$\mathbf{U}_m = \mathbf{U}_{m-1} + h_M \mathbf{f}(t_m, \mathbf{U}_m) \quad (8.15)$$

ist problematisch: Auf der linken und der rechten Seite kommt \mathbf{U}_m vor, das nennt man **implizit**. Die unbekannte Größe \mathbf{U}_m können wir also nicht per Umstellung einer Formel berechnen.

Durch Einsetzen sehen wir sofort ein, dass der unbekannte Wert \mathbf{U}_m Nullstelle der Funktion $\mathbf{G}_m: \mathbb{R}^d \rightarrow \mathbb{R}^d$ mit

$$\mathbf{G}_m(\mathbf{U}) = \mathbf{U} - h_M \mathbf{f}(t_m, \mathbf{U}) - \mathbf{U}_{m-1}$$

ist. Diese Gleichung ist linear, falls \mathbf{f} linear ist, und sonst nichtlinear. Wir müssen also in jeder Iteration der zusammengesetzten Quadratur ein Hilfsproblem lösen.



8. Anfangswertprobleme

$$\mathbf{G}_m(\mathbf{U}) = \mathbf{U} - h_M \mathbf{f}(t_m, \mathbf{U}) - \mathbf{U}_{m-1} \stackrel{!}{=} \mathbf{0}$$

Für das unbeschränkte Wachstumsmodell ist \mathbf{f} linear und autonom, und wir müssen eine Nullstelle \mathbf{U}^* des skalaren linearen Gleichungssystems

$$\mathbf{U} - h_M \alpha \mathbf{U} - \mathbf{U}_{m-1} = 0 \quad \Leftrightarrow \quad \mathbf{U}^* = \frac{\mathbf{U}_{m-1}}{1 - h_M \alpha}$$

bestimmen. Die nächste Iterierte ist dann $\mathbf{U}_m := \mathbf{U}^*$.

Im linearen nichtskalaren Fall muss analog ein höherdimensionales lineares Gleichungssystem gelöst werden, und im nichtlinearen Fall bietet sich gemäß VL 4 eine Variante des Newton-Verfahrens an. In jedem Fall erfüllt $\mathbf{U}_m = \mathbf{U}^*$ bzw. die mit dem Newton-Verfahren berechnete Approximation Gleichung (8.15).



8. Anfangswertprobleme

Übertragung aus VL 4:

Algorithmus 8.19 : Vektorielles Newton-Verfahren

input : $\mathbf{G}_m(\mathbf{U}) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ differenzierbar, Startwert $\mathbf{U}_{m-1} \in \mathbb{R}^d$, $\text{TOL} > 0$,
 $k_{\max} \in \mathbb{N}$

output : Approximation x an eine Nullstelle von \mathbf{G}

```
1  $\mathbf{x} = \mathbf{U}_{m-1};$ 
2  $k = 0;$ 
3 berechne  $\mathbf{r} = \mathbf{G}_m(\mathbf{x});$ 
4 while  $k < k_{\max}$  and  $\|\mathbf{r}\|_2 > \text{TOL}$  do
5    $\mathbf{A} = \mathbf{J}_{G_m}(\mathbf{x});$            % Auswertung Jacobi-Matrix
6   löse  $\mathbf{Ay} = \mathbf{r};$           %  $\mathbf{y} = (\mathbf{J}_{G_m})^{-1} \mathbf{G}_m(\mathbf{x})$ 
7    $\mathbf{x} = \mathbf{x} - \mathbf{y};$         %  $\mathbf{x} = \mathbf{x} - (\mathbf{J}_{G_m})^{-1} \mathbf{G}_m(\mathbf{x})$ 
8    $\mathbf{r} = \mathbf{G}_m(\mathbf{x});$ 
9    $k = k + 1;$ 
10 end
```



8. Anfangswertprobleme

Algorithmus 8.20 : Implizites Euler-Verfahren

input : $f: [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ stetig, $\mathbf{u}_0 \in \mathbb{R}^d$, $M \in \mathbb{N}$

output : $\{\mathbf{U}_0, \dots, \mathbf{U}_M\}$ als Approximation von $\{\mathbf{u}(0), \dots, \mathbf{u}(T)\}$

- 1 Berechne $h_M = \frac{T}{M}$;
 - 2 Setze $\mathbf{U}_0 = \mathbf{u}_0$;
 - 3 **for** $m = 1$ **to** M **do**
 - 4 $t_m = m \cdot h_M$;
 - 5 Berechne eine Nullstelle \mathbf{U}^* von $\mathbf{G}_m(\mathbf{U}) := \mathbf{U} - h_M f(t_m, \mathbf{U}) - \mathbf{U}_{m-1}$;
 - 6 $\mathbf{U}_m = \mathbf{U}^*$;
 - 7 **end**
-

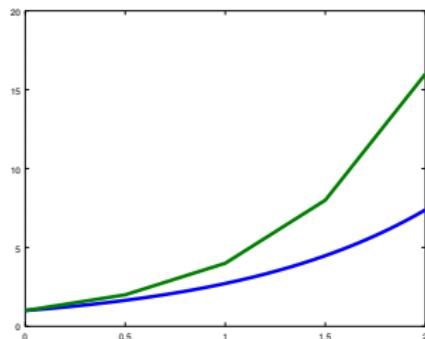
Die Berechnung von \mathbf{U}_m benötigt die Lösung eines (nicht) linearen Gleichungssystems in \mathbb{R}^d . Dies ist aufwändiger als einzelne f -Auswertungen in einem expliziten Verfahren. Ein guter Startwert für das Newton-Verfahren ist \mathbf{U}_{m-1} . Wegen der i. A. fehlenden globalen Konvergenz des Newton-Verfahrens sollte h_M nicht zu groß gewählt werden.



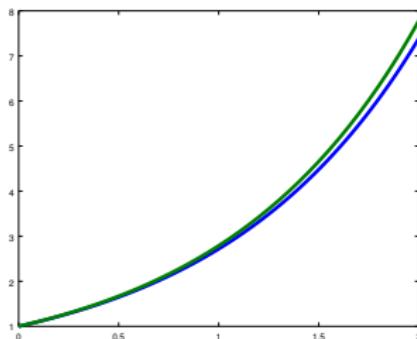
8. Anfangswertprobleme

Beispiel: Ergebnisse des impliziten Euler-Verfahrens für Bsp. 8.3 (unbeschränktes Wachstumsmodell)

$$h_M = 0.5$$



$$h_M = 0.05$$



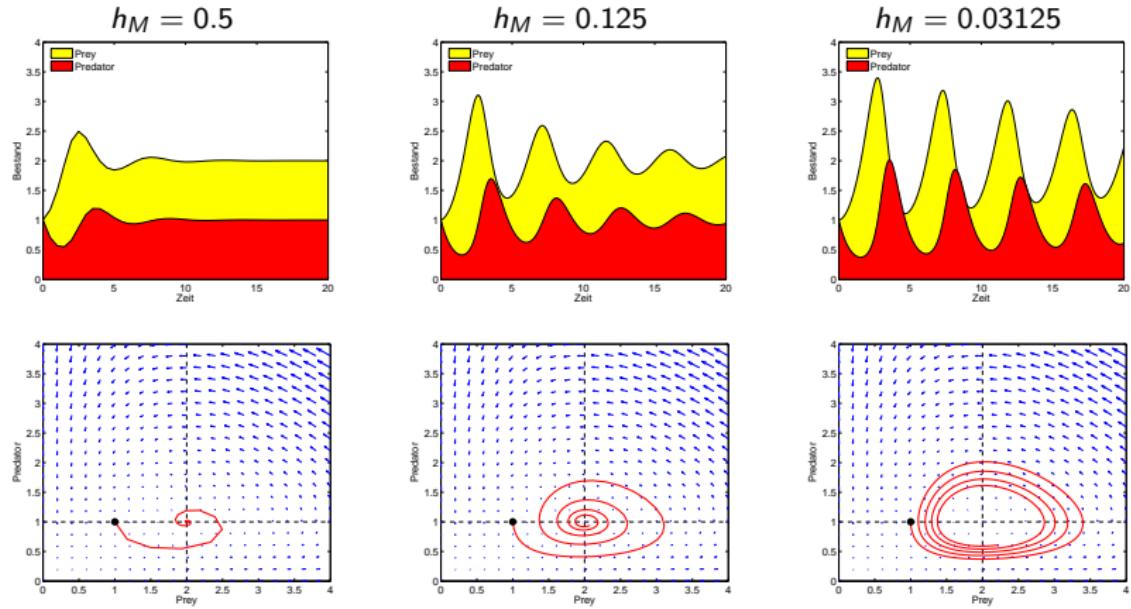
VIPLab-Demo in ILIAS

Für feine Schrittweiten h_M sehen wir eine gute Approximation (grün) der exakten Lösung (blau). Im Vergleich zum expliziten Euler-Verfahren ist eine gröbere Schrittweite für eine gute Approximation möglich.



8. Anfangswertprobleme

Beispiel: Ergebnisse des impliziten Euler-Verfahrens für Bsp. 8.5 (Lotka-Volterra)



Das implizite Verfahren erlaubt deutliche größere Schrittweiten h_M , \rightarrow später.

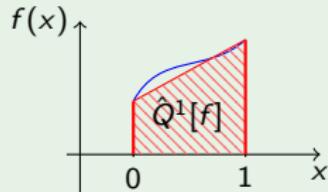


8. Anfangswertprobleme

Beispiel 8.21 (Trapez-Regel)

Die Parameter der $[0, 1]$ -Trapez-Regel \hat{Q}^1 lauten $\hat{s}_0 = 0$, $\hat{s}_1 = 1$ und $\hat{w}_0 = \hat{w}_1 = \frac{1}{2}$, vgl. VL 7.

Damit erhalten wir



$$\begin{aligned}\mathbf{u}(t_m) &= \mathbf{u}(t_{m-1}) + \int_{t_{m-1}}^{t_m} \mathbf{f}(s, \mathbf{u}(s)) ds \\ &\approx \mathbf{u}(t_{m-1}) + \frac{h_M}{2} \left(\underbrace{\mathbf{f}(t_{m-1}, \mathbf{u}(t_{m-1}))}_{F_0} + \underbrace{\mathbf{f}(t_m, \mathbf{u}(t_m))}_{F_1} \right).\end{aligned}$$

Die Berechnung von F_0 erfordert nur eine Funktionsauswertung mit bereits berechneten Werten. Der Wert $F_1 = \mathbf{f}(t_m, \mathbf{u}(t_m))$ ist nicht so einfach. Die Idee ist nun, den fehlenden Wert mit einer *anderen* Quadratur zu approximieren.



8. Anfangswertprobleme

Um insgesamt ein explizites Verfahren zu erhalten, führen wir einen expliziten Euler-Schritt durch, und ersetzen

$$\mathbf{f}(t_m, \mathbf{u}(t_m)) \approx \mathbf{f}(t_m, \mathbf{U}_{m-1} + h_M \mathbf{f}(t_{m-1}, \mathbf{U}_{m-1})).$$

Dies ergibt das explizite Verfahren von Heun.

Algorithmus 8.22 : Verfahrensschritt von Heun

input : $\mathbf{f}: [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ stetig, t_{m-1} , h_M , $\mathbf{U}_{m-1} \in \mathbb{R}^d$

output : \mathbf{U}_m

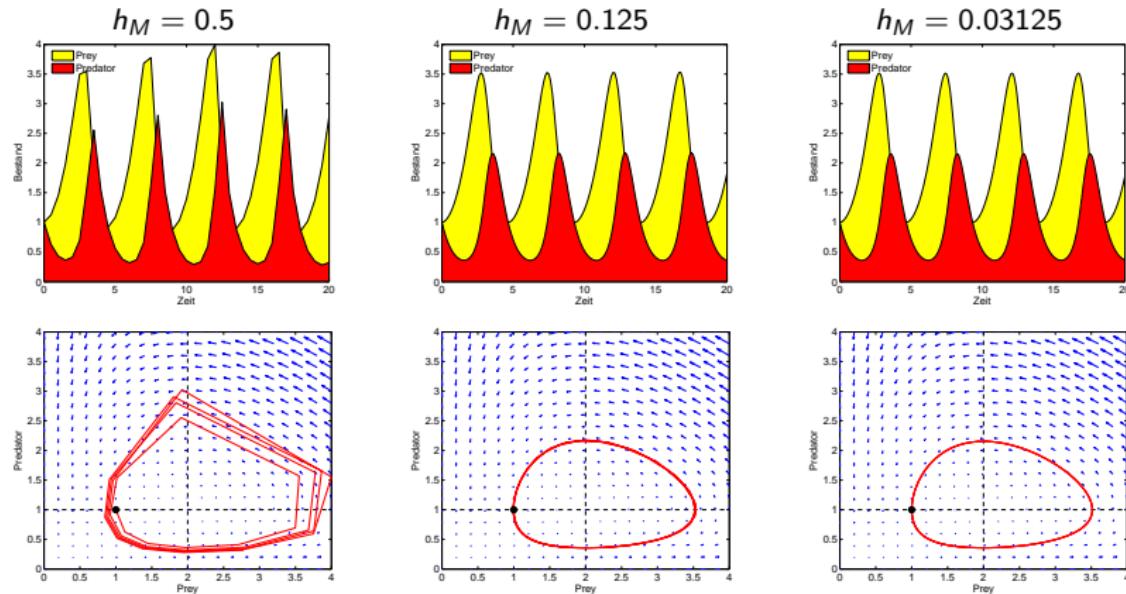
- 1 Setze $\mathbf{F}_0 = \mathbf{f}(t_{m-1}, \mathbf{U}_{m-1})$;
 - 2 Setze $\mathbf{F}_1 = \mathbf{f}(t_m, \mathbf{U}_{m-1} + h_M \mathbf{F}_0)$; % Trick zur Aufwandsreduktion
 - 3 Setze $\mathbf{U}_m = \mathbf{U}_{m-1} + \frac{h_M}{2} (\mathbf{F}_0 + \mathbf{F}_1)$;
-

Für das vollständige Verfahren von Heun ergänzen wir nun noch die Schleife (stückweise Quadratur) über m wie im expliziten Eulerverfahren.



8. Anfangswertprobleme

Beispiel: Ergebnisse des Verfahrens von Heun für Bsp 8.5 (Lotka-Volterra)



Das Verfahren von Heun liefert für deutlich größere h_M als beim expliziten Euler-Verfahren gute Approximationen, sogar besser als das implizite Euler-Verfahren.



8. Anfangswertprobleme

Um insgesamt ein implizites Verfahren zu erhalten, führen wir einen impliziten Euler-Schritt durch, und ersetzen

$$\mathbf{f}(t_m, \mathbf{u}(t_m)) \approx \mathbf{f}(t_m, \mathbf{U}_m).$$

Dies ergibt das implizite Verfahren von Crank-Nicolson, dessen Implementierung i. A. wieder eine Anwendung des Newton-Verfahrens erfordert.

Algorithmus 8.23 : Verfahrensschritt von Crank-Nicolson

input : $\mathbf{f}: [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ stetig, t_{m-1} , h_M , $\mathbf{U}_{m-1} \in \mathbb{R}^d$

output : \mathbf{U}_m

1 Berechne $t_m = m \cdot h_M$;

2 Definiere die Funktion

$$\mathbf{G}_m(\mathbf{U}) := \mathbf{U} - \frac{h_M}{2} (\mathbf{f}(t_{m-1}, \mathbf{U}_{m-1}) + \mathbf{f}(t_m, \mathbf{U})) - \mathbf{U}_{m-1};$$

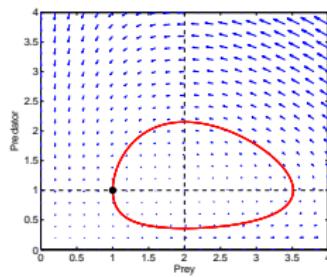
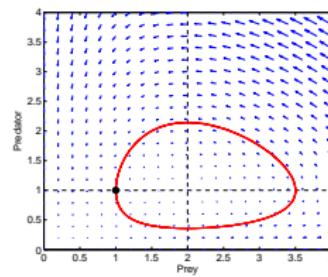
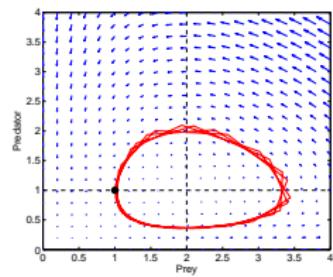
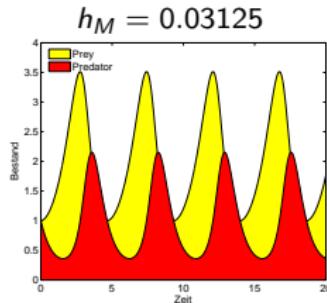
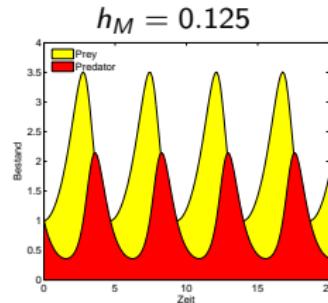
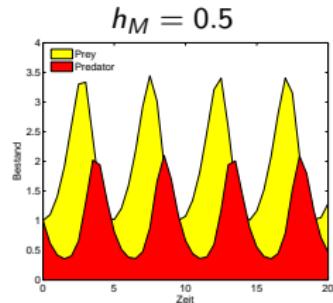
3 Berechne eine Nullstelle \mathbf{U}^* von \mathbf{G}_m ;

4 Setze $\mathbf{U}_m := \mathbf{U}^*$;



8. Anfangswertprobleme

Beispiel: Ergebnisse des Verfahrens von Crank-Nicolson für Bsp. 8.5



Bereits für die „riesige“ Schrittweite $h_M = 0.5$ sind die Ergebnisse (fast) akzeptabel!



8. Anfangswertprobleme

Um die experimentell gewonnenen Zusammenhänge zwischen Genauigkeit und globaler Schrittweite h_M zu quantifizieren, benötigen wir einen neuen Begriff:

Definition 8.24 (Konvergenzordnung)

Sei \mathbf{u} eine Lösung von Problem 8.7 und seien $\mathbf{U}_0, \dots, \mathbf{U}_M$ die diskreten Lösungsapproximationen zu den Zeitpunkten $t_0 = 0, \dots, t_M = T$. Dann heißt ein Verfahren **konvergent mit Ordnung p** , falls mit einer von h_M unabhängigen Konstanten C gilt

$$e_h := \max_{m=0, \dots, M} \|\mathbf{U}_m - \mathbf{u}(t_m)\| \leq Ch_M^p.$$

Dabei ist $\|\cdot\|$ eine beliebige Norm auf \mathbb{R}^d .

Bei einem Verfahren erster Ordnung ($p = 1$) reduziert sich der Fehler bei Halbierung von h_M also um einen Faktor 2, bei einem Verfahren zweiter Ordnung um den Faktor 4, usw. Außerdem sehen wir, dass wir für klein genug h_M den Fehler auch beliebig klein bekommen.



8. Anfangswertprobleme

Der folgende Satz bestätigt die Experimente:

Satz 8.25 (Konvergenzordnung)

Die expliziten und impliziten Euler-Verfahren besitzen die Konvergenzordnung $p = 1$. Das explizite Verfahren von Heun und das implizite Verfahren von Crank-Nicolson besitzen die Konvergenzordnung $p = 2$.

Es stellen sich nun zwei Fragen:

- ① Welchen Vorteil haben implizite Verfahren gegenüber expliziten Verfahren eigentlich?
- ② Wie können wir auf der Basis des bisherigen Vorgehens (approximative Ersetzung fehlender Werte durch Quadraturen niedrigerer Ordnung) systematisch Verfahren noch höherer Konvergenzordnung konstruieren?



8. Anfangswertprobleme

Wir beantworten zuerst die Frage nach der Bedeutung impliziter Verfahren.

Satz 8.26

Für die Konstante C in Definition 8.24 gilt:

$$C \leq \frac{e^{LT} - 1}{L} \max_{t \in [0, T]} \|\mathbf{u}^{(p+1)}\|$$

Hierbei hängt die Konstante L vom Verfahren und der rechten Seite \mathbf{f} ab.

Der Ableitungsterm $\|\mathbf{u}^{(p+1)}\|$ ist keine Überraschung, vgl. Satz 6.18 zum Interpolationsfehler in VL 6. Wir sehen, dass die Konstante C exponentiell von T und L abhängt. Das ist schlecht. T kann schnell groß sein, wenn uns das Langzeitverhalten der Lösung interessiert.



8. Anfangswertprobleme

$$e_h := \max_{m=0, \dots, M} \|\mathbf{U}_m - \mathbf{u}(t_m)\| \leq Ch_M^p \quad \text{und} \quad C \leq \frac{e^{LT} - 1}{L} \max_{t \in [0, T]} \|\mathbf{u}^{(p+1)}\|$$

Wir sehen: Falls T groß, oder L groß, oder die Ableitungen explodieren, dann benötigen wir eine sehr kleine Zeitschrittweite, um den Fehler klein zu halten. Ein Verfahren heißt **(numerisch) stabil**, wenn es Lösungen beschränkt hält.

Aber: Für viele Probleme kann man eine andere Schranke für die Konstante zeigen, so dass C nicht von T abhängt. **Das ist aber nur für implizite Verfahren möglich!**

Der Mehraufwand impliziter Verfahren kann also gerechtfertigt sein, da viel größere Zeitschrittweiten als bei expliziten Verfahren möglich sind.



8. Anfangswertprobleme

Abschließend skizzieren wir noch, wie Verfahren höherer Ordnung systematisch konstruiert werden können.

Wir wollen im Intervall $[t_{m-1}, t_m]$ eine Quadratur mit $N + 1$ Zwischenstellen ansetzen. Bisher haben wir für Verfahren der globalen Konvergenzordnung p Quadraturen der Konvergenzordnung $p - 1$ zur Approximation der sogenannten **Stufen**

$$\mathbf{F}_n \approx \mathbf{f}(s_n, \mathbf{u}(s_n))$$

verwendet. Das Prinzip lässt sich genau so fortsetzen, und führt auf die Familie der **Runge-Kutta Verfahren**. Wir beschränken uns auf ein Beispiel eines expliziten Runge-Kutta Verfahrens.



8. Anfangswertprobleme

Die Verwendung aller aufsteigend sortierten lokalen Quadraturen führt auf das explizite Runge-Kutta Verfahren dritter Ordnung (RK3):

Algorithmus 8.27 : Verfahrensschritt des RK3-Verfahrens

input : $f: [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ stetig, $t_{m-1}, h_M, \mathbf{U}_{m-1} \in \mathbb{R}^d$

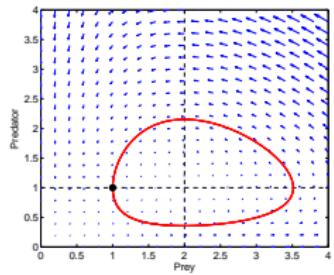
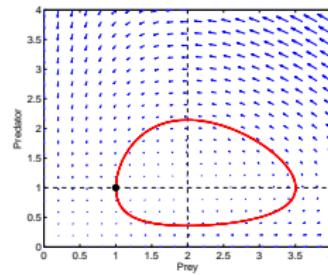
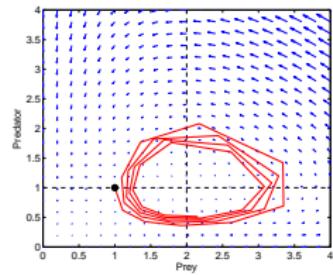
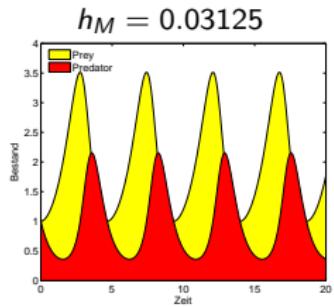
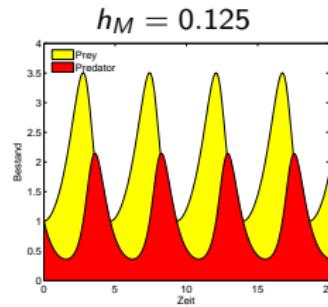
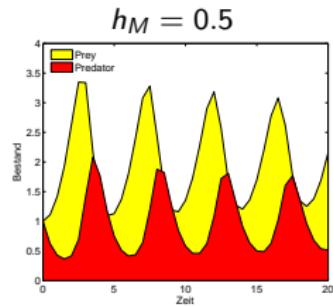
output : $\mathbf{U}_m \approx \mathbf{u}(t_m)$

- 1 setze $\mathbf{F}_0 = f(t_{m-1}, \mathbf{U}_{m-1});$ % expl. Euler
 - 2 setze $\mathbf{F}_1 = f(t_{m-1} + \frac{h_M}{2}, \mathbf{U}_{m-1} + \frac{h_M}{2} \mathbf{F}_0);$ % expl. Heun
 - 3 setze $\mathbf{F}_2 = f(t_{m-1} + h_M, \mathbf{U}_{m-1} - h_M (\mathbf{F}_0 - 2\mathbf{F}_1));$ % expl. Simpson
 - 4 setze $\mathbf{U}_m = \mathbf{U}_{m-1} + \frac{h_M}{6} (\mathbf{F}_0 + 4\mathbf{F}_1 + \mathbf{F}_2);$ % summierte Regel
-



8. Anfangswertprobleme

Beispiel: Ergebnisse des RK3-Verfahrens für Beispiel 8.5





Zusammenfassender Vergleich der Verfahren



8. Anfangswertprobleme

Wir vergleichen sechs ausgewählte Verfahren anhand Beispiel 8.4 (beschränktes Wachstum) mit den Parametern $\beta = 1.0$, $\alpha = 3.0$ und $u_0 = 0.2$.

Wir kennen die exakte Lösung:

$$u(t) = \beta \frac{u_0}{u_0 + (\beta - u_0) e^{-\alpha t}}$$

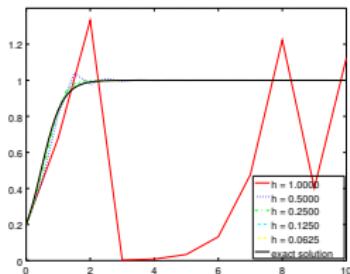
und es gilt

$$\lim_{t \rightarrow \infty} u(t) = \beta = 1.$$

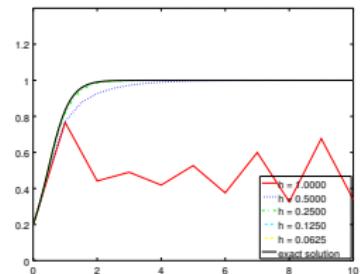


8. Anfangswertprobleme

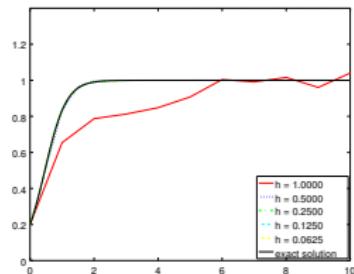
Expliziter Euler



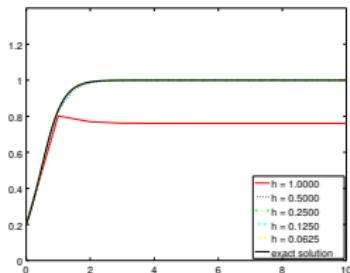
(expl.) Verfahren von Heun



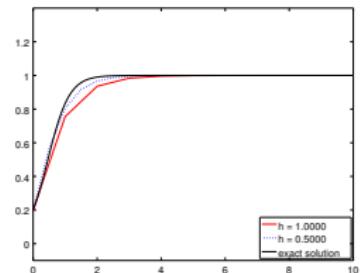
Simpson-Regel



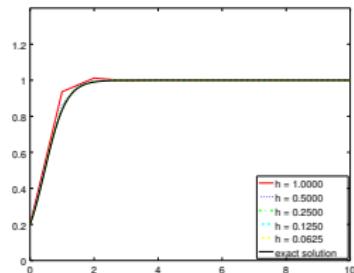
Klassisches Runge-Kutta



Impliziter Euler



Crank-Nicolson



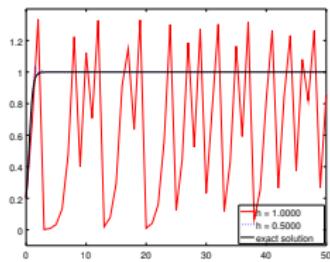
Für hinreichend kleine h_M sehen wir immer Konvergenz. Implizite Verfahren (untere Reihe) sind stabiler als explizite Verfahren.



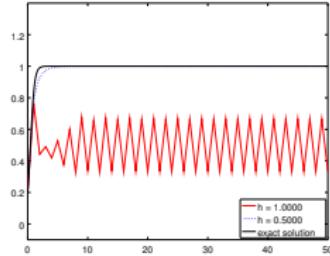
8. Anfangswertprobleme

Wichtig bei expliziten Verfahren: Sorgfalt ist geboten bei der Wahl von h_M .

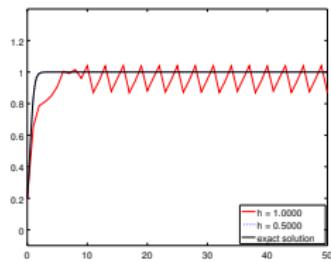
Expliziter Euler



Verfahren von Heun



Simpson Regel

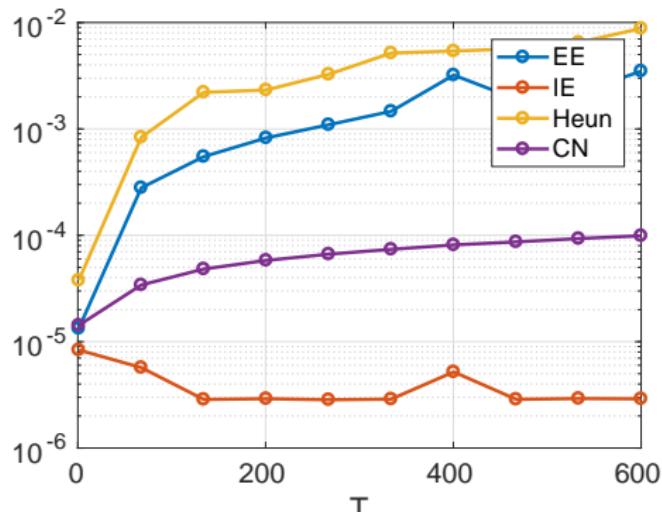


Zu großen Schrittweiten bei expliziten Verfahren führen sehr oft zu Oszillationen bis hin zur kompletten Instabilität. Das werden wir im Kontext partieller Differentialgleichungen nochmals beleuchten.



8. Anfangswertprobleme

Der theoretische Vorteil von Verfahren höherer Ordnung, und oft auch der von impliziten Verfahren, kann sich in die Praxis übertragen:



Modellproblem $\dot{u} = -20u + b$, gemittelte Laufzeiten (100 Läufe) in Abhängigkeit von T , jeweils für vorgegebene und deshalb vergleichbare Genauigkeit der Approximation für $t = T$.

Bei gleicher Ordnung sind die impliziten Verfahren hier jeweils laufzeit-effizienter, wenn die Linearität des Problems ausgenutzt wird. Für dieses Beispiel zahlt sich höhere Ordnung nicht aus.



8. Anfangswertprobleme

Zusammenfassung

- Quadraturen sind die Basis zur Herleitung von Lösungsverfahren für AWP.
- Einschrittverfahren benötigen nur die Lösungsapproximation des vorigen Zeitschrittes. Wichtige Einschrittverfahren sind das explizite/implizite Euler, Heun oder Runge-Kutta-Verfahren.
- Die Reduzierung der Schrittweite verringert den Fehler, insbesondere bei Verfahren höherer Ordnung.
- Mit steigender Ordnung eines Verfahrens steigt der Aufwand (was sich aber in der Regel lohnt). Explizite Verfahren benötigen nur die Auswertung der rechten Seite. Bei impliziten Verfahren muss ein lineares oder nichtlineares Gleichungssystem gelöst werden.
- Zu große Schrittweiten können zu Problemen führen:
 - ▶ Stabilitätsprobleme bei expliziten Verfahren,
 - ▶ Lösbarkeit der nichtlinearen Probleme bei impliziten Verfahren.



8. Anfangswertprobleme

Hausaufgaben

- Wiederholung der Notation für Gradienten etc.
- Verinnerlichung von Zeitschrittverfahren aus dieser Vorlesung
- Wiederholung der Definition einer Ableitung über Differenzenquotienten aus der HM1
- Wiederholung des Satz von Taylor aus der HM12



Beispieldaufgaben



8. Anfangswertprobleme

Euler-Verfahren

Wir betrachten das Anfangswertproblem

$$u(0) = 1, \quad \dot{u}(t) = -2u(t), \quad t > 0.$$

Berechnen Sie die ersten beiden Schritte des expliziten und impliziten Euler-Verfahrens zur Schrittweite $h_M = 1$ mit dem Startwert $\mathbf{U}_0 = u(0)$.



8. Anfangswertprobleme

Lösungshinweise: Das ist eine reine Rechenaufgabe, um ein Gefühl für die beiden Verfahren aufzubauen.

Ergebnis: Für das explizite Euler-Verfahren erhalten wir $U_1 = -1$ und $U_2 = 1$.
Für das implizite Euler-Verfahren erhalten wir $U_1 = \frac{1}{3}$ und $U_2 = \frac{1}{9}$.



8. Anfangswertprobleme

Stabilität

Wir betrachten nochmal das Anfangswertproblem

$$u(0) = 1, \quad \dot{u}(t) = -2u(t), \quad t > 0.$$

Für welche Schrittweiten h_M bleiben die Werte U_m , die die beiden Euler-Verfahren berechnen, immer positiv?



8. Anfangswertprobleme

Lösungshinweise: Diese Aufgabe soll einerseits nochmal das Verständnis verstärken, und andererseits auf ein weiteres Problem expliziter Verfahren hinweisen: Zu grobe Schrittweiten können nichtphysikalische Lösungsapproximationen ergeben. Hier lautet die exakte Lösung e^{-2t} .

Konkreter Lösungsansatz: Verfahren für genau dieses Problem hinschreiben und Umstellen nach h_M .

Ergebnis: Das explizite Euler-Verfahren erzeugt nur für $h_M \in (0, 0.5)$ positive Lösungen, das implizite Euler-Verfahren für $h_M > 0$.



8. Anfangswertprobleme

Implizite Verfahren

Wir betrachten das Anfangswertproblem

$$\dot{u}_1(t) = u_2(t), \quad \dot{u}_2(t) = 3u_1(t) + u_2(t) \quad \text{für } t \in (0, T),$$

mit den Anfangswerten $u_1(0) = u_2(0) = 1$. Formulieren Sie das implizite Euler-Verfahren für dieses Problem. Ist hier das Newton-Verfahren erforderlich? Was passiert, wenn die zweite Gleichung $\dot{u}_2(t) = 3u_1(t)u_2(t)$ lautet?



8. Anfangswertprobleme

Lösungshinweise: Nachdem man das System aufgeschrieben hat, sollte man bekannte Strukturen identifizieren können.

Ergebnis: Das implizite Euler-Verfahren lautet für dieses Problem:

$$\mathbf{U}_m = \mathbf{U}_{m-1} + h_M \begin{pmatrix} 0 & 1 \\ 3 & 2 \end{pmatrix} \mathbf{U}_m \quad \text{mit} \quad \mathbf{U}_m \approx \begin{pmatrix} u_1(t_m) \\ u_2(t_m) \end{pmatrix}$$

Das können wir umstellen:

$$\left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - h_M \begin{pmatrix} 0 & 1 \\ 3 & 2 \end{pmatrix} \right) \mathbf{U}_m = \mathbf{U}_{m-1}$$

In jedem Schritt ist also ein LGS zu lösen. Für die modifizierte Problemstellung funktioniert dies nicht mehr, weil die „Matrix“ die Funktion u_1 enthält.



9. PDEs: Finite Differenzen I



John von Neumann 1903–1957

The advance of analysis is, at this moment,
stagnant along the entire front of non-linear
problems.

John von Neumann 1946
(als Begründung zum Bau eines
virtuellen Windkanals basierend auf
numerischen Approximationen)



Motivation: Modellierung in der Kontinuumsmechanik



Beispiel 9.1 (Modellierung von Transportprozessen)

Wir modellieren den Transport eines Stoffes in einem dünnen, unendlich langen Rohr mit longitudinalem Fluss $F = F(x, t)$. Die gesuchte Funktion $u(x, t)$ beschreibt die Konzentration des Stoffes im Ortspunkt x zum Zeitpunkt t .



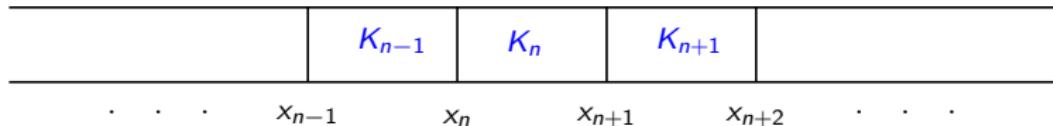
Wir leiten im Folgenden eine partielle Differentialgleichung (PDE) für u her, die dieses Phänomen beschreibt. PDEs beinhalten Ableitungen nach mehreren Variablen, im Gegensatz zu gewöhnlichen Differentialgleichungen (ODEs) in VL 8. Bei uns sind dies Orts- und Zeitvariablen.

Das Raclette-Problem (VL 0) ist eine konkrete Instanz dieses Beispiels.



9. PDEs: Finite Differenzen I

Wir modellieren das Rohr als eindimensionales unendliches Intervall, d. h. wir vernachlässigen Effekte, die orthogonal zur Flussrichtung auftreten. Die erste Idee ist nun, das Rohr in Teilstücke (Teilintervalle) $K_n = [x_n, x_{n+1}]$ der Länge $h_x := x_{n+1} - x_n > 0$ zu zerlegen:



Wir nehmen an, dass der Fluss $F(x, t)$ in positive x -Richtung orientiert ist. Innerhalb eines beliebigen Teilstücks K_n gilt dann zwischen zwei Zeitpunkten t und $t + h_t$ mit $h_t > 0$ die **Massenbilanz**:

$$\int_{K_n} u(x, t + h_t) dx = \int_{K_n} u(x, t) dx + \int_t^{t+h_t} F(x_n, s) - F(x_{n+1}, s) ds.$$

Strenggenommen ist u die Massendichte der Konzentration (analog für F), und erst das Integral über die Dichte ist die Konzentration.

Die Konzentration in K_n zum Zeitpunkt $t + h_t$ ist also die Konzentration, die vorher schon in K_n herrschte, plus der Zu- und Abfluss des Stoffes am linken und rechten Rand von K_n durch den Fluss F im Zeitraum $[t, t + h_t]$.



9. PDEs: Finite Differenzen I

Für genügend kleines h_t gilt (Integral \approx Intervalllänge mal Integrand)

$$\int_t^{t+h_t} F(x_n, s) - F(x_{n+1}, s) \, ds \approx h_t (F(x_n, t) - F(x_{n+1}, t)),$$

und mit dem Hauptsatz der Differential- und Integralrechnung folgt:

$$h_t (F(x_n, t) - F(x_{n+1}, t)) = h_t \int_{x_{n+1}}^{x_n} \partial_x F(x, t) \, dx = -h_t \int_{K_n} \partial_x F(x, t) \, dx$$

Das setzen wir in die Massenbilanz von der vorherigen Folie,

$$\int_{K_n} u(x, t + h_t) \, dx = \int_{K_n} u(x, t) \, dx + \int_t^{t+h_t} F(x_n, s) - F(x_{n+1}, s) \, ds,$$

ein und erhalten, weil h_t unabhängig von K_n ins Integral gezogen werden darf:

$$\int_{K_n} \frac{u(x, t + h_t) - u(x, t)}{h_t} \, dx \approx - \int_{K_n} \partial_x F(x, t) \, dx$$



9. PDEs: Finite Differenzen I

$$\int_{K_n} \frac{u(x, t + h_t) - u(x, t)}{h_t} dx \approx - \int_{K_n} \partial_x F(x, t) dx$$

Wir betrachten den Integrand auf der linken Seite. Aus der Schule und der HM12 kennen wir die Definition der Ableitung über einen Differenzenquotient, zunächst allgemein für eine Funktion $f: \mathbb{R} \rightarrow \mathbb{R}$ und eine Stelle x_0 :

$$f'(x_0) = \lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0} \quad \Leftrightarrow \quad f'(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

Hierbei haben wir $h = x - x_0$, also $x = x_0 + h$, gesetzt. Die Ausdrücke innerhalb des Limes heißen **Differenzenquotienten** und stehen im Mittelpunkt dieser und der nächsten Vorlesung.

Verständnisübung: In den DQ die anschauliche Definition der Ableitung über die Tangentensteigung wiederfinden, vgl. VL 4.



9. PDEs: Finite Differenzen I

$$\int_{K_n} \frac{u(x, t + h_t) - u(x, t)}{h_t} dx \approx - \int_{K_n} \partial_x F(x, t) dx$$

Angewendet auf unsere Situation erhalten wir also für den Integranden im Grenzwert $h_t \rightarrow 0$:

$$\int_{K_n} \partial_t u(x, t) dx \approx - \int_{K_n} \partial_x F(x, t) dx \quad \forall n \in \mathbb{Z}, t > 0$$

In VL 8 haben wir Zeitableitungen als \dot{u} bezeichnet, hier schreiben wir $\partial_t u$ zur besseren Unterscheidung von der Ortsableitung.

Jetzt werden wir noch die konzeptionelle Aufteilung in Teilrohre los, d. h. wir gehen zur Grenze $h_x \rightarrow 0$ über. Weil die Gleichung für infinitesimale K_n gelten muss, erhalten wir:

$$\partial_t u(x, t) \approx -\partial_x F(x, t) \quad \forall (x, t) \in \mathbb{R} \times \mathbb{R}_{>0}$$



9. PDEs: Finite Differenzen I

Allgemeine Transportgleichung

$$\partial_t u(x, t) + \partial_x F(x, t) = 0 \quad \forall (x, t) \in \mathbb{R} \times \mathbb{R}_{>0}$$

Für konkrete Transportvorgänge wird der Fluss F durch sogenannte **konstitutive Gesetze** spezifiziert, bspw. aus Experimenten. Sind advektive Effekte dominant, können wir den Fluss schreiben als

$$F(x, t) = v(x, t)u(x, t)$$

mit einer Advektionsgeschwindigkeit (Transportgeschwindigkeit) v :

Homogene lineare Advektionsgleichung

$$\partial_t u + \partial_x(vu) = 0$$



9. PDEs: Finite Differenzen I

Allgemeine Transportgleichung

$$\partial_t u(x, t) + \partial_x F(x, t) = 0 \quad \forall (x, t) \in \mathbb{R} \times \mathbb{R}_{>0}$$

Sind diffusive Effekte dominant, liefert das sogenannte **Ficksche Gesetz** für zweikomponentige Stoffmischungen, oder das **Fouriersche Gesetz** für den Wärmefluss in isotropen Materialien, die Proportionalität zur negativen Flussrichtung, mit Proportionalitätskonstante $\alpha \in \mathbb{R} \setminus \{0\}$:

$$F(x, t) = -\alpha \partial_x u(x, t)$$

Homogene lineare Diffusionsgleichung

$$\partial_t u - \alpha \partial_{xx} u = 0$$

Das ist ein Spezialfall des zeitabhängigen Raclette-Problems aus VL 0.



Interessant ist auch der stationäre Fall, in dem sich keine zeitlichen Änderungen mehr ergeben. Das Problem lautet dann wegen $\partial_t u = 0$:

Stationäre homogene lineare Diffusionsgleichung

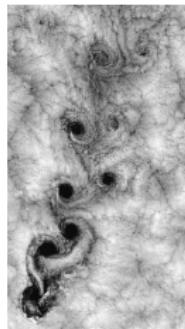
$$-\alpha \partial_{xx} u = 0$$

Das ist ein weiterer Spezialfall des stationären Raclette-Problems aus VL 0.



9. PDEs: Finite Differenzen I

Die drei Gleichungen lassen sich auf ähnliche Weise für den allgemeinen Fall herleiten, d. h. in mehr als einer Ortsdimension $\mathbf{x} \in \mathbb{R}^d$ mit $d > 1$; und für inhomogene rechte Seiten (Quellterme, externe Kräfte) $\mathbf{f} \neq \mathbf{0}$. Sie ergeben sich auch oft als Teilprobleme in komplizierteren Fragestellungen Ihres Studiums, bspw. in der Strömungs- und Festkörpermechanik.



Umströmung einer (hohen) Insel, Wirbelschleppen hinter Flugzeugen

NASA public domain



9. PDEs: Finite Differenzen I

Deshalb ist es legitim, dass wir uns in den verbleibenden Vorlesungen auf solche **Modellprobleme** einschränken. So arbeiten wir den mathematischen und methodischen Kern heraus, ohne die breite Anwendbarkeit unserer Methoden zu verlieren.

PDEs, die stationäre Diffusionsprozesse modellieren, werden oft als **elliptisch** bezeichnet. Instationäre Diffusionsprozesse werden mit dem Begriff **parabolisch** assoziiert, und instationäre Transportprozesse oft mit dem Begriff **hyperbolisch**.

Dafür gibt es eine mathematische Rechtfertigung, die letztendlich auf Kegelschnitten basiert.

Wir werden sehen: Diese drei Modellprobleme weisen gänzlich unterschiedliches Verhalten auf, die die Numerik vor unterschiedliche Herausforderungen stellt. Dies rechtfertigt nochmals die Sprechweise „Modellprobleme“.



Eine erste Finite Differenzen Methode



9. PDEs: Finite Differenzen I

Wir betrachten den Fall $\Omega =]a, b[$ und die stationäre inhomogene lineare Diffusionsgleichung mit $\alpha = 1$:

$$-\partial_{xx} u(x) = f(x) \quad \text{für } x \in]a, b[$$

An den Intervallgrenzen fixieren wir die Temperatur auf Null,

$$u(a) = u(b) = 0$$

im Inneren des Intervalls wird die Temperaturverteilung dadurch und die rechte Seite f bestimmt.

Anhand dieses Modellproblems wollen wir die Grundideen der Finite Differenzen Approximation erarbeiten. Danach erweitern wir das zugrundeliegende Prinzip um eine systematische Konstruktion besserer (d. h. genauerer) Approximationen.



9. PDEs: Finite Differenzen I

Aus der HM1, der Schule, und der Motivation ist bekannt, dass die Ableitung von u im Ort an einer Stelle $x \in]a, b[$ für $h_x > 0$ definiert ist als:

$$\partial_x u(x) = \lim_{h_x \rightarrow 0} \frac{u(x + h_x) - u(x)}{h_x}$$

Wir diskretisieren später auch in der Zeit, deshalb unterscheiden wir h_x und h_t .

Deshalb ist für kleines h_x der sogenannte **Vorwärts-Differenzenquotient**

$$(D_{h_x}^+ u)(x) := \frac{u(x + h_x) - u(x)}{h_x}$$

eine geeignete Approximation der Ableitung an der Stelle x , d. h.

$$(D_{h_x}^+ u)(x) \approx \partial_x u(x)$$

in einer (kleinen) Umgebung von x .



Vollkommen analog ergibt sich aus

$$\partial_x u(x) = \lim_{h_x \rightarrow 0} \frac{u(x) - u(x - h_x)}{h_x}$$

eine Approximation der Ableitung durch den **Rückwärts-Differenzenquotient**:

$$(D_{h_x}^- u)(x) := \frac{u(x) - u(x - h_x)}{h_x} \approx \partial_x u(x)$$

Wir können also die erste Ableitung $\partial_x u(x)$ durch die *Auswertung* der Funktion u an zwei Punkten $\{x, x + h_x\}$ beziehungsweise alternativ $\{x - h_x, x\}$ approximieren.



9. PDEs: Finite Differenzen I

Die Erweiterung dieser Idee auf höhere Ableitungen wie $\partial_{xx} u(x)$, erfolgt kanonisch durch Hintereinanderanwenden. Eine geschickte Möglichkeit zur Approximation von $\partial_{xx} u$ ist, zunächst D_h^- durch Auswertungen von $\partial_x u$, und dann D_h^+ zur Approximation von $\partial_x u$ durch Auswertungen von u anzuwenden:

$$\begin{aligned}\partial_{xx} u(x) \approx D_{h_x}^- \partial_x u(x) \approx D_{h_x}^- D_{h_x}^+ u(x) &= \frac{D_{h_x}^- u(x + h_x) - D_{h_x}^- u(x)}{h_x} \\ &= \frac{1}{h_x} \left(\frac{u(x + h_x) - u(x)}{h_x} - \frac{u(x) - u(x - h_x)}{h_x} \right) \\ &= \frac{u(x + h_x) - 2u(x) + u(x - h_x)}{h_x^2} \\ &= \frac{u(x - h_x) - 2u(x) + u(x + h_x)}{h_x^2}\end{aligned}$$

Diese Approximation der zweiten Ableitung wird aus naheliegenden Gründen auch als **zentraler Differenzenquotient** bezeichnet.

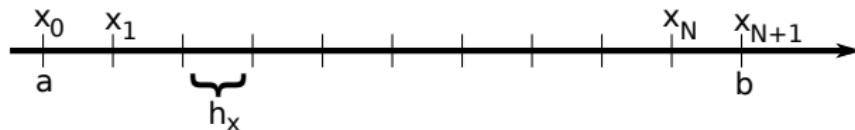


9. PDEs: Finite Differenzen I

Als letzte Zutat benötigen wir noch ein **Gitter**. Zu einem $N \in \mathbb{N}$ setzen wir:

$$h_x := \frac{b - a}{N + 1} \quad \text{und} \quad x_n := a + nh_x \quad \forall n = 0, \dots, N + 1$$

Wegen $x_0 = a + 0h = a$ und $x_{N+1} = a + (N + 1) \frac{b-a}{N+1} = a + b - a = b$ ist das ganze Intervall mitsamt seinen Grenzen überdeckt. Wir nennen h_x **Gitterweite**, und die Menge aller $N + 2$ **Gitterpunkte** zusammen Gitter.



Die N Gitterpunkte x_1, \dots, x_N bezeichnen wir als **innere Gitterpunkte**, und die Punkte x_0 und x_{N+1} als **Randpunkte**.



9. PDEs: Finite Differenzen I

Nun ersetzen wir die zweite Ableitung $\partial_{xx} u$ in der Gleichung

$$-\partial_{xx} u(x) = f(x)$$

in jedem inneren Punkt durch den zentralen Differenzenquotienten, und erhalten

$$-\frac{1}{h_x^2} \left(u(x_n - h_x) - 2u(x_n) + u(x_n + h_x) \right) \approx f(x_n) \quad \text{für alle } n = 1, \dots, N$$

bzw. nach Einsetzen der Definition der Gitterpunkte und Vorzeichen-Kosmetik:

$$\frac{1}{h_x^2} \left(-u(x_{n-1}) + 2u(x_n) - u(x_{n+1}) \right) \approx f(x_n) \quad \text{für alle } n = 1, \dots, N$$

Die Funktion f werten wir also auch nur an den Gitterpunkten aus.

Verständnisübung: Warum setzen wir die Approximation nicht in den Randpunkten an?



9. PDEs: Finite Differenzen I

Wir bestimmen nun die Lösung u nicht mehr überall, sondern nur noch in den endlich vielen Gitterpunkten, d. h. wir suchen Approximationen

$$u_n \approx u(x_n) \quad \text{für } n = 1, \dots, N.$$

Die Randwerte, d. h. die Funktionswerte $u_0 := u(a) = 0$ und $u_{N+1} := u(b) = 0$, liegen bereits vor und sind somit keine Unbekannten.

Zur Vereinheitlichung der Notation schreiben wir $f_n := f(x_n)$ für alle $n = 1, \dots, N$. Insgesamt suchen wir also N Approximationen u_1, \dots, u_N , so dass gilt:

$$\frac{1}{h_x^2} \left(-u_{n-1} + 2u_n - u_{n+1} \right) = f_n \quad \text{für alle } n = 1, \dots, N$$

und $u_0 = u_{N+1} = 0$

Unsere Intuition besagt, dass der Fehler, den wir uns bei dieser Approximation einhandeln, mit h_x kleiner werden sollte. Das überlegen wir uns später im Detail.

Verständnisübung: Warum macht das nur Sinn für innere Gitterpunkte?



9. PDEs: Finite Differenzen I

Wir schaufeln den h_x^{-2} -Term auf die rechte Seite:

$$\left(-u_{n-1} + 2u_n - u_{n+1} \right) = h_x^2 f_n \quad \text{für } n = 1, \dots, N \quad \text{und} \quad u_0 = u_{n+1} = 0$$

Komponentenweise lesen wir ab, wenn wir die bekannten Werte $u_0 = 0$ und $u_{n+1} = 0$ substituieren:

$$\begin{aligned} 2u_1 - u_2 &= h_x^2 f_1 \\ -u_1 + 2u_2 - u_3 &= h_x^2 f_2 \\ -u_2 + 2u_3 - u_4 &= h_x^2 f_3 \\ &\vdots \\ -u_{N-2} + 2u_{N-1} - u_N &= h_x^2 f_{N-1} \\ -u_{N-1} + 2u_N &= h_x^2 f_N \end{aligned}$$



9. PDEs: Finite Differenzen I

Das ist gerade ein lineares Gleichungssystem für die N unbekannten Werte der gesuchten Approximation $u_n \approx u(x_n)$ in den inneren Gitterpunkten. Mit den Vektoren

$$\mathbf{f} := (h_x^2 f_1, \dots, h_x^2 f_n)^T \in \mathbb{R}^N \quad \text{und} \quad \mathbf{u} := (u_1, \dots, u_n)^T \in \mathbb{R}^N$$

sowie der Matrix

$$\mathbf{A} := \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{N \times N}$$

können wir dieses Gleichungssystem kompakt schreiben:

$$\mathbf{A}\mathbf{u} = \mathbf{f}$$

Das ist genau das (stationäre) Raclette-Problem aus VL 0!



Modellgleichungen der Diffusion



9. PDEs: Finite Differenzen I

Wir beginnen wie üblich mit der Definition dessen, worüber wir reden wollen:

Definition 9.2 (Differentialgleichung für Diffusionsprozesse)

Sei $\Omega \subset \mathbb{R}^d$ zusammenhängend, offen und nichtleer, und sei $[0, T]$ ein Zeitintervall. Dann beschreibt die partielle Differentialgleichung

$$\partial_t u(x, t) - \Delta u(x, t) = f(x, t) \quad \text{für } (x, t) \in \Omega \times]0, T[$$

einen Diffusionsprozess für die unbekannte Größe $u: \Omega \times]0, T[\rightarrow \mathbb{R}$ und die rechte Seite (Quellterme, externe Kräfte) $f: \Omega \times]0, T[$.

Für $x = (x_1, \dots, x_d)^T$ ist dabei der **Laplace-Operator** die Summe aller zweiten partiellen Ableitungen im Ort Ω , d. h.

$$\Delta u(x) = \nabla \cdot \nabla u(x) = \operatorname{div} \nabla u(x) = \sum_{i=1}^d \frac{\partial^2}{\partial x_i^2} u(x).$$

Verständnisübung: 1D-Beispiel $\partial_t u - \partial_{xx} u = 0$ wiederfinden.



9. PDEs: Finite Differenzen I

Sofort klar ist auch die folgende praxisrelevante Erweiterung:

Definition 9.3 (Anisotrope Diffusionsprozesse)

Mit den Bezeichnern aus der vorherigen Definition, und $x \mapsto A(x) \in \mathbb{R}^{d \times d}$ beschreibt

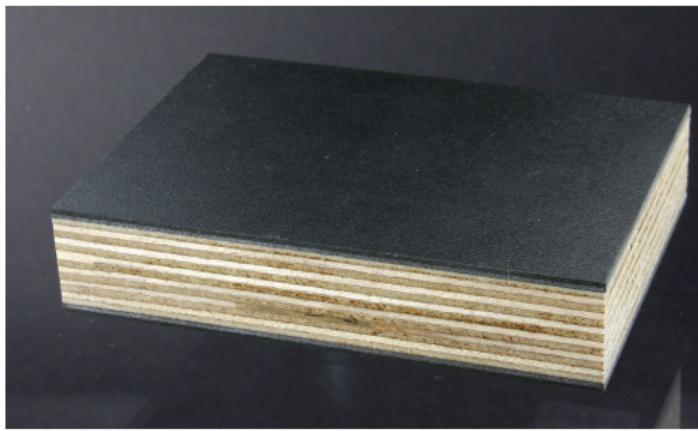
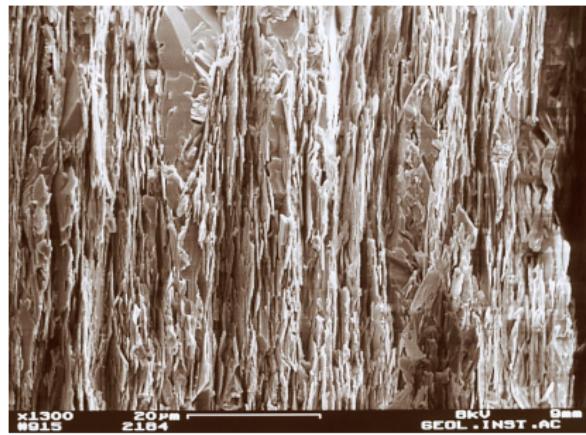
$$\partial_t u(x, t) - \nabla \cdot (A(x) \nabla u(x, t)) = f(x, t) \quad \text{für } (x, t) \in \Omega \times]0, T[$$

einen anisotropen Diffusionsprozess.

Verständnisübung: 1D-Beispiel $\partial_t u - \alpha \partial_{xx} u = 0$ wiederfinden.

Um dies einzusehen, sollte man das Skalarprodukt $\nabla \cdot A \nabla u$ einmal nachvollziehen. Der Gradient bezieht sich wie der Laplace-Operator nur auf die Ortsvariablen. Wir beschränken uns in dieser Vorlesung auf zeitunabhängige Diffusionskoeffizienten $A = A(x) \neq A(x, t)$.

9. PDEs: Finite Differenzen I



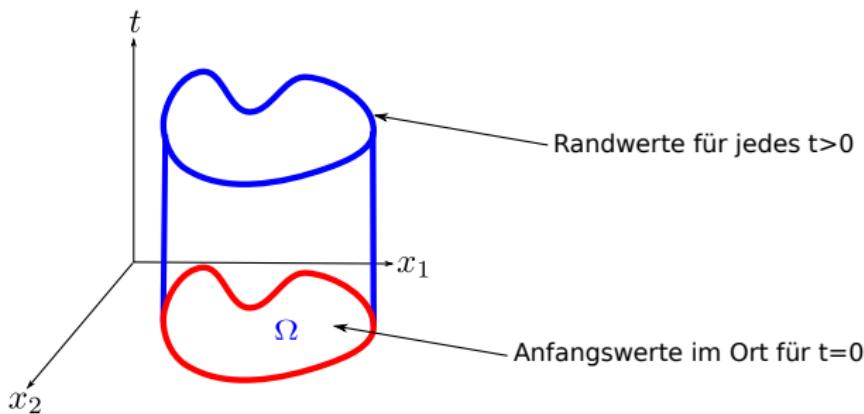
Faserstruktur von Schiefer und Sperrholz

Wikipedia, CC BY-SA 3.0

9. PDEs: Finite Differenzen I



Die Differentialgleichung alleine garantiert kein wohlgestelltes Problem. Analog zu gewöhnlichen Differentialgleichungen in VL 8 benötigen wir **Anfangswerte** für den (willkürlichen) Startzeitpunkt $t = 0$, und zusätzlich **Randwerte**:



Als Gedankenmodell dient der **Orts-Zeit-Zylinder**, in dem die Lösung lebt: Auf dem Boden des Zylinders ($t = 0$) werden Anfangswerte, und auf dem Mantel für jede „Zeitschicht“ $t > 0$ Randwerte vorgegeben.



9. PDEs: Finite Differenzen I

Anfangswerte zum Zeitpunkt $t = 0$ werden über eine nur vom Ort abhängige Funktion $u_0: \Omega \rightarrow \mathbb{R}$ vorgegeben:

$$\begin{aligned}\partial_t u(x, t) - \Delta u(x, t) &= f(x, t) & (x, t) \in \Omega \times \mathbb{R}_{>0} \\ u(x, 0) &= u_0(x) & x \in \Omega\end{aligned}$$

Bei der Wärmeleitungsgleichung kann so beispielsweise die initiale Temperatur vorgegeben werden. Weil wir an der Differentialgleichung nichts geändert haben, funktioniert dies auch für den anisotropen Fall.

Analog benötigen wir **Randwerte**. Wir betrachten hierzu zwei praxisrelevante Möglichkeiten, und bezeichnen den Rand von Ω mit $\partial\Omega := \bar{\Omega} \setminus \Omega$. Beispielsweise ist der Rand von $\Omega =]a, b[$ die Punktmenge $\{a, b\}$.



9. PDEs: Finite Differenzen I

Definition 9.4 (Dirichlet- und Neumann-Randbedingungen)

Es gelten die Bezeichner aus der vorherigen Definition, und $g: \partial\Omega \times [0, T] \rightarrow \mathbb{R}$ sei eine vorgegebene Funktion.

Bei **Dirichlet-Randbedingungen** werden Funktionswerte für die Lösung u vorgegeben:

$$u(x, t) \stackrel{!}{=} g(x, t) \quad \text{für } (x, t) \in \partial\Omega \times [0, T]$$

Bei **Neumann-Randbedingungen** werden Ableitungswerte für die Lösung u vorgegeben:

$$\nabla u(x, t) \cdot n(x) \stackrel{!}{=} g(x, t) \quad \text{für } (x, t) \in \partial\Omega \times [0, T]$$

Hierbei ist $n(x)$ die *äußere Normale* von $\partial\Omega$ im Punkt $x = (x_1, \dots, x_d)^T$.

Falls $g(x, t) \equiv 0$, sprechen wir von **homogenen Randbedingungen**, sonst von **inhomogenen Randbedingungen**.



9. PDEs: Finite Differenzen I

Bei der Wärmeleitungsgleichung (Raclette-Problem) entsprechen Dirichlet-Randbedingungen der Vorgabe von **Temperaturwerten** auf dem Rand. Falls $g(x, t) = g(x)$, also im zeitunabhängigen Fall, wird so bspw. eine permanente Aufheizung modelliert. Neumann-Randbedingungen modellieren einen vorgegebenen **Wärmefluss** über den Rand, d. h. eine Temperaturänderung in Normalenrichtung.

Für den Fall $\Omega =]a, b[$ und Dirichlet-Randbedingungen fordern wir bspw.

$$u(a) \stackrel{!}{=} u(b) \stackrel{!}{=} 0 \quad \text{bzw.} \quad u(a) \stackrel{!}{=} 42 \text{ und } u(b) \stackrel{!}{=} 23$$

im homogenen bzw. inhomogenen Fall. Neumann-Randbedingungen lauten wegen den äußeren Normalen $n(a) = -1$ und $n(b) = 1$ bspw.

$$\partial_x u(a) \stackrel{!}{=} \partial_x u(b) \stackrel{!}{=} 0 \quad \text{bzw.} \quad \partial_x u(a) \stackrel{!}{=} -42 \text{ und } \partial_x u(b) \stackrel{!}{=} 23.$$



9. PDEs: Finite Differenzen I

Wenn wir alles zusammenstöpseln, erhalten wir ein **Anfangs-Randwert-Problem**, exemplarisch für den Fall von Dirichlet-Randbedingungen:

Definition 9.5 (Anfangs-Randwertproblem für Diffusionsprozesse)

Sei $\Omega \subset \mathbb{R}^d$ offen und zusammenhängend. Seien weiter $f: \Omega \times]0, T[\rightarrow \mathbb{R}$, $u_0: \Omega \rightarrow \mathbb{R}$, $A: \Omega \rightarrow \mathbb{R}^d$ und $g: \partial\Omega \times]0, T[\rightarrow \mathbb{R}$. Das

Anfangs-Randwertproblem (ARWP) für Diffusionsprozesse besteht in der Aufgabe, eine Funktion $u: \Omega \times]0, T[\rightarrow \mathbb{R}$ zu finden, die die folgenden Bedingungen erfüllt:

$$\begin{aligned}\partial_t u(x, t) - \nabla \cdot (A(x) \nabla u(x, t)) &= f(x, t) && \text{für } (x, t) \in \Omega \times]0, T[\\ u(x, t) &= g(x, t) && \text{für } (x, t) \in \partial\Omega \times]0, T[\quad (\text{Randwerte}) \\ u(x, 0) &= u_0(x) && \text{für } x \in \Omega \quad (\text{Anfangswert})\end{aligned}$$

Für Neumann-Randbedingungen erfolgt die Definition analog.



9. PDEs: Finite Differenzen I

Wir betrachten als Beispiel das Raclette-Problem in 1D ($\Omega =]a, b[$) mit der Vorgabe homogener Dirichlet-Randbedingungen und einem konstanten Diffusionskoeffizienten $\alpha > 0$. Das ARWP lautet dann:

$$\begin{aligned}\partial_t u(x, t) - \alpha \partial_{xx} u(x, t) &= f(x, t) && \text{für } (x, t) \in]a, b[\times]0, T[\\ u(a, t) = u(b, t) &= 0 && \text{für } t \in [0, T[\\ u(x, 0) &= u_0(x) && \text{für } x \in]a, b[\end{aligned}$$

Als Verständnisübung sollte man dieses Problem einmal für den Fall von Neumann-Randbedingungen formulieren, oder komponentenweise für konkrete Diffusionskoeffizienten $A(x)$.



9. PDEs: Finite Differenzen I

Das Problem weist bereits im Fall isotroper Diffusion Modellcharakter auf.

Definition 9.6 (Parabolisches Modellproblem, Wärmeleitungs-ARWP)

Es gelten die Bezeichner aus der vorherigen Definition. Das **parabolische Modellproblem** besteht in der Aufgabe, eine Funktion $u: \Omega \times]0, T[\rightarrow \mathbb{R}$ zu finden, die die folgenden Bedingungen erfüllt:

$$\begin{aligned}\partial_t u(x, t) - \Delta u(x, t) &= f(x, t) && \text{für } (x, t) \in \Omega \times]0, T[\\ u(x, t) &= g(x, t) && \text{für } (x, t) \in \partial\Omega \times [0, T[\\ u(x, 0) &= u_0(x) && \text{für } x \in \Omega\end{aligned}$$

Das Raclette-Problem in \mathbb{R}^d ist also eine Instanz dieses Modellproblems.



9. PDEs: Finite Differenzen I

Wie in der Motivation interessiert uns auch der stationäre Fall mit $\partial_t u(x, t) \equiv 0$. In diesem Fall müssen wir keine Anfangswerte vorgeben, und es ergibt sich ein reines Randwertproblem, das wieder Modellcharakter aufweist:

Definition 9.7 (Elliptisches Modellproblem, Poisson-RWP)

Es gelten die Bezeichner aus der vorherigen Definition. Das **elliptische Modellproblem** besteht in der Aufgabe, eine Funktion $u: \Omega \rightarrow \mathbb{R}$ zu finden, die die folgenden Bedingungen erfüllt:

$$\begin{aligned} -\Delta u(x) &= f(x) && \text{für } x \in \Omega \\ u(x) &= g(x) && \text{für } x \in \partial\Omega \end{aligned}$$

Man kann beweisen: Unter gewissen Bedingungen konvergieren für $t \rightarrow \infty$ Lösungen $u(x, t)$ des parabolischen Modellproblems gegen Lösungen $u(x)$ des elliptischen Modellproblems. Die Sprechweise des **stationären Falls** ist also gerechtfertigt.



Fahrplan für VL 9–12:

- VL 9: Numerische Approximation von Lösungen im zeitunabhängigen Fall (d. h. für das elliptische Modellproblem) mit **Finite Differenzen Verfahren**, also der Idee aus der Motivation, Ableitungen durch Differenzenquotienten anzunähern.
- VL 10: Numerische Approximation von Lösungen im zeitabhängigen Fall mit Finite Differenzen Verfahren, für das parabolische Modellproblem und die advektive Transportgleichung. Hierzu integrieren wir die Zeitschrittverfahren aus VL 8.
- VL 11+12: **Finite Elemente Methoden** als viel breiter anwendbare Alternative.



Systematische Konstruktion von Finite Differenzen Verfahren



9. PDEs: Finite Differenzen I

Wir überlegen uns nun eine systematische Konstruktion. Dabei werden wir feststellen, dass dies direkt eine Information über den Fehler der Approximation liefert. Wir schlagen also zwei Fliegen mit einer Klappe.

Der Ausgangspunkt unserer Überlegungen ist die Taylorreihe aus der HM12:

Satz 9.8 (Taylorreihe)

Für ein beliebiges offenes Intervall $]a, b[$, eine beliebige Entwicklungsstelle $x_0 \in]a, b[$, und eine Funktion $f :]a, b[\rightarrow \mathbb{R}$ heißt

$$\begin{aligned} T(x; x_0) &= \sum_{k=0}^{\infty} \frac{\partial_x^k f(x_0)}{k!} (x - x_0)^k \\ &= f(x_0) + \partial_x f(x_0) (x - x_0) + \frac{\partial_x^2 f(x_0)}{2} (x - x_0)^2 + \frac{\partial_x^3 f(x_0)}{6} (x - x_0)^3 + \dots \end{aligned}$$

Taylorreihe (Taylorentwicklung) von f um x_0 . Für ein festes $k < \infty$ sprechen wir von einem **Taylorpolynom** $T_k(x; x_0)$. Für x nahe x_0 ist bereits ein Taylorpolynom niedrigen Grades eine gute Approximation von f .



9. PDEs: Finite Differenzen I

Wir entwickeln nun $u(x_{n+1})$ und $u(x_{n-1})$ jeweils um h_x nach rechts und links, d. h. um die Stelle x_n . Hierbei ist x_n mit $n \in \{1, \dots, N\}$ ein beliebiger innerer Gitterpunkt. So erhalten wir:

$$u(x_{n+1}) = \sum_{k=0}^{\infty} \frac{h_x^k}{k!} \partial^k u(x_n) = u(x_n) + h_x \partial_x u(x_n) + \frac{h_x^2}{2} \partial_{xx} u(x_n) + \frac{h_x^3}{6} \partial_{xxx} u(x_n) + \dots$$

$$u(x_{n-1}) = \sum_{k=0}^{\infty} \frac{(-h_x)^k}{k!} \partial^k u(x_n) = u(x_n) - h_x \partial_x u(x_n) + \frac{h_x^2}{2} \partial_{xx} u(x_n) - \frac{h_x^3}{6} \partial_{xxx} u(x_n) \pm \dots$$

Die Idee ist nun, diese Taylorreihen „abzuschneiden“, d. h. zu Taylorpolynomen überzugehen. Weil wir die Gitterweite h_x später gegen Null gehen lassen, ist bei der Approximation der Reihe durch das Taylorpolynom die niedrigste h_x -Potenz die, die den Fehler dominiert.



9. PDEs: Finite Differenzen I

Beispielsweise gilt für $h_x \rightarrow 0$, wenn wir nach dem linearen Term abschneiden:

$$\begin{aligned} u(x_{n+1}) &= u(x_n) + h_x \partial_x u(x_n) + \mathcal{O}(h_x^2) \\ u(x_{n-1}) &= u(x_n) - h_x \partial_x u(x_n) + \mathcal{O}(h_x^2) \end{aligned}$$

Wenn wir die erste Entwicklung nach $\partial_x u(x_n)$ auflösen, erhalten wir:

$$\partial_x u(x_n) = \frac{u(x_{n+1}) - u(x_n)}{h_x} + \mathcal{O}(h_x)$$

Wenn wir die zweite Entwicklung nach $\partial_x u(x_n)$ auflösen, erhalten wir:

$$\partial_x u(x_n) = \frac{u(x_n) - u(x_{n-1})}{h_x} + \mathcal{O}(h_x)$$

Das sind gerade die **Vorwärts- und Rückwärts-Differenzenquotienten**, die wir uns eben ad-hoc überlegt haben, nun allerdings versehen mit einer Fehlerinformation der Größenordnung $\mathcal{O}(h_x)$.

Erinnerung: Vorzeichen sind in der \mathcal{O} -Notation egal.



9. PDEs: Finite Differenzen I

Approximationen mit besserem Fehler erhalten wir, wenn wir die Taylorentwicklungen später abschneiden, und dann geschickt linear kombinieren:

$$\begin{aligned} u(x_{n+1}) &= u(x_n) + h_x \partial_x u(x_n) + \frac{h_x^2}{2} \partial_{xx} u(x_n) + \mathcal{O}(h_x^3) \\ u(x_{n-1}) &= u(x_n) - h_x \partial_x u(x_n) + \frac{h_x^2}{2} \partial_{xx} u(x_n) + \mathcal{O}(h_x^3) \end{aligned}$$

Wenn wir nun die zweite Taylor-Entwicklung von der ersten subtrahieren und das Ergebnis nach $\partial_x u(x_n)$ auflösen, erhalten wir:

$$\partial_x u(x_n) = \frac{u(x_{n+1}) - u(x_{n-1})}{2h_x} + \mathcal{O}(h_x^2)$$

Das ist ein **zentraler Differenzenquotient** zur Approximation der ersten Ableitung mit einem Fehler der Ordnung $\mathcal{O}(h_x^2)$.



9. PDEs: Finite Differenzen I

Das Konstruktionsprinzip lässt sich für höhere Ableitungen genau so durchführen.
Wenn wir beispielsweise die beiden Taylor-Entwicklungen

$$\begin{aligned} u(x_{n+1}) &= u(x_n) + h_x \partial_x u(x_n) + \frac{h_x^2}{2} \partial_{xx} u(x_n) + \frac{h_x^3}{6} \partial_{xxx} u(x_n) + \mathcal{O}(h_x^4) \\ u(x_{n-1}) &= u(x_n) - h_x \partial_x u(x_n) + \frac{h_x^2}{2} \partial_{xx} u(x_n) - \frac{h_x^3}{6} \partial_{xxx} u(x_n) + \mathcal{O}(h_x^4) \end{aligned}$$

addieren, ergibt sich:

$$u(x_{n+1}) + u(x_{n-1}) = 2u(x_n) + h_x^2 \partial_{xx} u(x_n) + \mathcal{O}(h_x^4)$$

Auflösen nach $\partial_{xx} u(x_n)$ liefert:

$$\partial_{xx} u(x_n) = \frac{u(x_{n+1}) - 2u(x_n) + u(x_{n-1})}{h_x^2} + \mathcal{O}(h_x^2)$$

Das entspricht gerade dem zentralen Differenzenquotient zur Approximation der zweiten Ableitung aus unserer ad-hoc Herleitung, und wir lesen ab, dass der Fehler quadratisch in h_x ist.

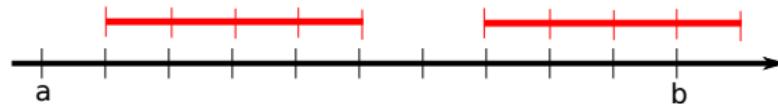


9. PDEs: Finite Differenzen I

Auf diese Weise lassen sich Finite Differenzen für höhere Ableitungsstufen und/oder höhere Genauigkeit konstruieren. Dazu sind i.A. weitere Taylorentwicklungen nötig, d. h. weitere Auswertungspunkte der Funktion. Wenn wir beispielsweise zusätzlich $u(x_{n+2})$ und $u(x_{n-2})$ um die Stelle x_n entwickeln, und die somit vier Entwicklungen geschickt kombinieren, erhalten wir:

$$\partial_x^3 u(x_n) = \frac{-0.5u(x_{n-2}) + u(x_{n-1}) - u(x_{n+1}) + 0.5u(x_{n+2})}{h_x^3} + \mathcal{O}(h_x^2)$$

Das ist ein zentraler Differenzenquotient für die dritte Ableitung mit quadratischem Fehler. Für die nicht definierten Auswertungspunkte muss ein einseitiger Differenzenquotient (mit gleichem Fehlerterm) verwendet werden.



Das Aufstellen eines LGS zur Bestimmung der Finite Differenzen Approximation $u_n \approx u(x_n)$ der Lösung einer Differentialgleichung erfolgt immer genau so wie in unserem ersten Beispiel, vgl. die Übungen.

https://en.wikipedia.org/wiki/Finite_difference_coefficient



9. PDEs: Finite Differenzen I

Den Fall inhomogener Dirichlet-Randwerte diskutieren wir exemplarisch für die linke Intervallgrenze. Die entsprechende Gleichung lautet:

$$\frac{1}{h_x^2} \left(-u_0 + 2u_1 - u_2 \right) = f_1$$

Wir substituieren den Randwert, also $u_0 = g(a)$:

$$\frac{1}{h_x^2} \left(-g(a) + 2u_1 - u_2 \right) = f_1$$

Das können wir umstellen:

$$(2u_1 - u_2) = h_x^2 f_1 + g(a)$$

Wir sehen: Inhomogene Dirichlet-Randwerte können problemlos in die rechte Seite des LGS integriert werden. Dabei muss die Skalierung mit h_x^{-2} beachtet werden.



9. PDEs: Finite Differenzen I

Den Fall inhomogener Neumann-Randwerte diskutieren wir exemplarisch für die rechte Intervallgrenze. Dazu nehmen wir temporär einen virtuellen Gitterpunkt x_{N+2} her, um die Randbedingung

$$\partial_x u(x_{N+1}) = g(b)$$

mit dem zentralen Differenzenquotienten der Ordnung 2 zu diskretisieren:

$$\partial_x u(x_{N+1}) \approx \frac{u_{N+2} - u_N}{2h_x} = g(b)$$

Diesen Ausdruck lösen wir nach u_{N+2} auf, und setzen ihn in den zentralen Differenzenquotienten um u_{N+1} ein:

$$\frac{-u_{N+2} + 2u_{N+1} - u_N}{h_x^2} = f_{N+1}$$

So erweitern wir das LGS um eine Unbekannte.



Konsistenz, Stabilität und Konvergenz



9. PDEs: Finite Differenzen I

Wir benötigen einige Bezeichner: $\{x_0 = a, x_1, \dots, x_N, x_{N+1} = b\}$ sei ein Gitter für das Intervall $[a, b]$ der Schrittweite h_x , und $u: [a, b] \rightarrow \mathbb{R}$ die exakte Lösung eines Randwertproblems. Mit

$$U_{\text{ex}} := \begin{pmatrix} u(x_1) \\ \vdots \\ u(x_N) \end{pmatrix} \quad U := \begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix} \quad F_{\text{ex}} := \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{pmatrix}$$

bezeichnen wir die Vektoren aus dem \mathbb{R}^N der Auswertungen der exakten Lösung bzw. den Vektor ihrer Approximation, sowie den Vektor der Auswertungen der rechten Seite, jeweils in den Gitterpunkten.

Den Vektor $U_{\text{ex}} - U$ nennen wir auch **Diskretisierungsfehler**, und quantifizieren ihn naheliegenderweise in der Maximumsnorm, d. h.

$$\|U_{\text{ex}} - U\|_{\infty} = \max_{n=1, \dots, N} |u(x_n) - u_n|.$$



9. PDEs: Finite Differenzen I

Um auch Variationen der Modellprobleme abzudecken, betrachten wir eine beliebige lineare Differentialgleichung, die durch einen **Differentialoperator** \mathcal{L} definiert wird. Für das Poisson-Problem und das anisotrope stationäre Diffusionsproblem

$$-\Delta u = f \quad \text{bzw.} \quad -\nabla \cdot A \nabla u = f$$

lauten die Differentialoperatoren beispielsweise

$$\mathcal{L} := -\Delta \quad \text{bzw.} \quad \mathcal{L} := -\nabla \cdot A \nabla.$$

Analog bezeichnen wir mit L_{h_x} die Finite Differenzen Approximation des Differentialoperators, für das Poisson-Problem haben wir uns bspw. überlegt:

$$L_{h_x} = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{N \times N}$$



9. PDEs: Finite Differenzen I

Später und in den Übungen werden wir sehen: Auch für kompliziertere (lineare) Differentialoperatoren \mathcal{L} (d. h. für kompliziertere RWP als das Modellproblem) können wir immer ein LGS aufstellen, dessen Koeffizientenmatrix in diesem abstrakten Setting L_{h_x} entspricht.

Die zentrale Idee für die Analyse ist nun, den Diskretisierungsfehler $U_{\text{ex}} - U$ in das Finite Differenzen Verfahren einzusetzen. Weil der Diskretisierungsfehler ein Vektor ist, weil das Verfahren eine Repräsentation als Matrix besitzt, und weil die Dimensionen zusammenpassen, funktioniert das weitestgehend schmerzfrei.



9. PDEs: Finite Differenzen I

Das machen wir also, und nutzen direkt die Linearität von L_{h_x} :

$$L_{h_x}(U_{\text{ex}} - U) = L_{h_x}U_{\text{ex}} - L_{h_x}U$$

Per Definition des LGS zur Bestimmung von $U = (u_1, \dots, u_N)^T$ erhalten wir einerseits

$$L_{h_x}U = F_{\text{ex}},$$

und andererseits mit dem exakten Differentialoperator,

$$F_{\text{ex}} = \mathcal{L}U_{\text{ex}},$$

wobei die Auswertungen punktweise zu verstehen sind. Insgesamt haben wir also:

$$L_{h_x}(U_{\text{ex}} - U) = L_{h_x}U_{\text{ex}} - \mathcal{L}U_{\text{ex}}$$



9. PDEs: Finite Differenzen I

$$L_{h_x} (U_{\text{ex}} - U) = L_{h_x} U_{\text{ex}} - \mathcal{L} U_{\text{ex}}$$

Das ist natürlich grandios, wenn wir es zurückübersetzen: Links haben wir die Anwendung des Verfahrens auf den Diskretisierungsfehler, was sehr stark nach einem Stabilitätsargument riecht: Beschränkte exakte Lösungen sollen beschränkte approximative Lösungen implizieren.

Rechts haben wir in gewisser Weise ein Residuum, d. h. den Unterschied zwischen dem Einsetzen der exakten Lösung in das Finite Differenzen Verfahren und in das kontinuierliche Problem. Dieses Residuum verwenden wir in der Konsistenzanalyse.



9. PDEs: Finite Differenzen I

Damit ist die folgende Definition naheliegend:

Definition 9.9 (Konsistenz)

Ein Finite Differenzen Verfahren heißt **konsistent**, falls

$$\lim_{h_x \rightarrow 0} \|L_{h_x} U_{\text{ex}} - \mathcal{L} U_{\text{ex}}\|_{\infty} = 0,$$

und es heißt konsistent mit Konsistenzordnung p , falls

$$\|L_{h_x} U_{\text{ex}} - \mathcal{L} U_{\text{ex}}\|_{\infty} \leq C_K h_x^p$$

mit einer von h_x unabhängigen Konstante C_K gilt.

Die Konsistenzordnung beschreibt also, wie gut der exakte Differentialoperator \mathcal{L} durch Finite Differenzen approximiert wird.



9. PDEs: Finite Differenzen I

Aufgrund der Konstruktion aus abgeschnittenen Taylor-Reihen ist klar:

Satz 9.10 (Konsistenz)

Die Konsistenzordnung der Vorwärts- und Rückwärtsdifferenzen-Quotienten für die erste Ableitung ist $p = 1$, die Konsistenzordnung für den zentralen Differenzenquotienten der ersten Ableitung ist $p = 2$, und die Konsistenzordnung für den zentralen Differenzenquotienten für die zweite Ableitung ist $p = 2$.

Ähnliche Resultate gelten für die anderen Approximationen, die wir unterwegs konstruiert haben.



9. PDEs: Finite Differenzen I

Der Stabilitätsbegriff ist ebenfalls klar:

Definition 9.11 (Stabilität)

Ein durch die Matrix L_{h_x} repräsentiertes Finite Differenzen Verfahren heißt **stabil**, falls die Lösung stetig von den Daten abhängt, d. h.

$$\|U\|_\infty \leq C_S \|L_{h_x} U\|_\infty = \|F_{\text{ex}}\|_\infty$$

mit einer Konstante C_S .

Die Stabilität erfordert per Definition eine problemabhängige Untersuchung. Man kann algebraisch (M-Matrix Theorie) oder über Eigenwert-Abschätzungen (Samarski-Theorie) Stabilität zeigen für das Modellproblem und viele weitere Probleme.



9. PDEs: Finite Differenzen I

Der Konvergenzbegriff basiert nun auf dem Diskretisierungsfehler:

Definition 9.12 (Konvergenz)

Ein durch die Matrix L_{h_x} repräsentiertes Finite Differenzen Verfahren heißt **konvergent**, falls der Diskretisierungsfehler eine Nullfolge ist, d. h.

$$\lim_{h_x \rightarrow 0} \|U_{\text{ex}} - U\|_{\infty} = 0$$

Das Verfahren besitzt Konvergenzordnung p , falls

$$\|U_{\text{ex}} - U\|_{\infty} \leq M h_x^p$$

Das ist ein punktweiser Konvergenzbegriff.



Wenn wir die Konsistenz und die Stabilität zusammenstöpseln, erhalten wir ein zentrales Resultat:

Satz 9.13 (Konsistenz+Stabilität=Konvergenz)

Wenn ein Finite Differenzen Verfahren konsistent mit Ordnung p und stabil ist, dann ist es auch konvergent, und zwar mit derselben Ordnung.

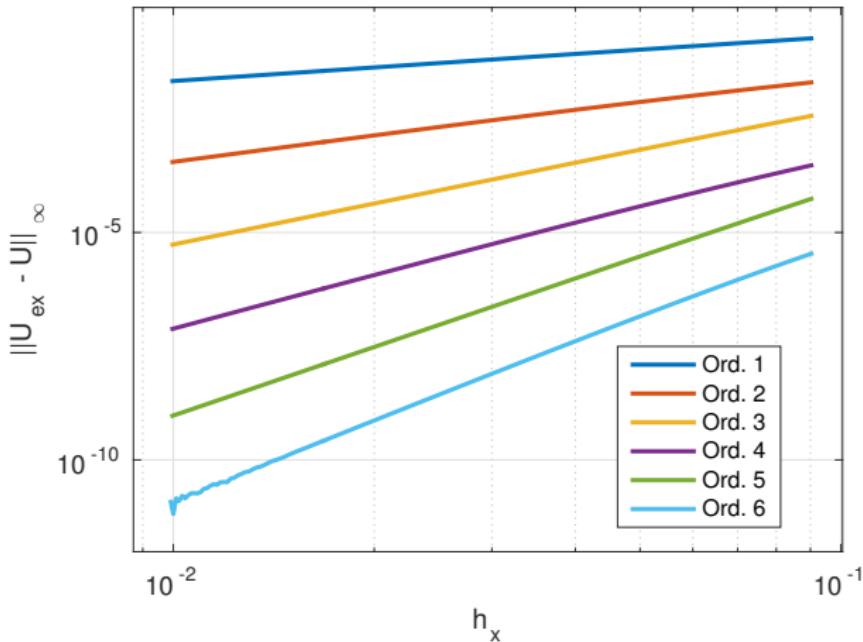
Insbesondere konvergiert die Approximation der Lösung des Poisson-Problems mit dem Differenzenquotient $D_{h_x}^- D_{h_x}^+$ quadratisch.



Numerische Beispiele



9. PDEs: Finite Differenzen I



Fehler gegen die exakte Lösung für
 $\partial_{xx} u(x) = (x(x - 1) - 2) \cos(x) + 2(2x - 1) \sin(x)$ auf $]0, 1[$ mit homogenen
Dirichlet-Randwerten.



Zusammenfassung

- Das Poisson-Problem, die Wärmeleitungsgleichung und die Advektionsgleichung weisen jeweils Modellcharakter auf. Deshalb konzentrieren wir uns in den verbleibenden VLen auf diese Modellprobleme.
- Finite Differenzen Verfahren basieren darauf, mittels Differenzenquotienten die Ableitung(en) einer Funktion durch Funktionsauswertungen zu ersetzen. Sie können systematisch aus abgeschnittenen Taylorreihen konstruiert werden, dies liefert direkt den Konsistenzfehler.
- Wir haben einseitige und zentrale Differenzenquotienten unterschiedlicher Ordnung für die erste und zweite Ableitung kennengelernt.
- Zur Anwendung der Finite Differenzen Methode werden die Differenzenquotienten auf einem Gitter betrachtet, und die Lösung wird in den Gitterpunkten approximiert. Dies führt auf ein dünn besetztes LGS, das mit den Verfahren aus VL 2 und VL 5 gelöst werden kann.
- Zentrale Differenzenquotienten sind für Diffusionsprobleme gut geeignet.



Hausaufgaben

- Wiederholung und Vorbereitung für nächste Woche:
 - ▶ Wärmeleitungs- und Advektionsgleichung
 - ▶ Einseitige und zentrale Differenzenquotienten von heute
 - ▶ Explizites und implizites Euler-Verfahren aus VL 8
 - ▶ Stabilitätsbegriffe von VL 8 und heute



Beispieldaufgaben



9. PDEs: Finite Differenzen I

Differenzenquotienten

Konstruieren Sie den zentralen Differenzenquotient

$$\partial_x^3 u(x_n) = \frac{-0.5u(x_{n-2}) + u(x_{n-1}) - u(x_{n+1}) + 0.5u(x_{n+2})}{h_x^3} + \mathcal{O}(h_x^2)$$

zweiter Ordnung für die dritte Ableitung. Für welche Gitterpunkte ist er nicht definiert? Wie können Sie das Problem lösen?



9. PDEs: Finite Differenzen I

Lösungshinweise: Entwickeln Sie zusätzlich $u(x_{n+2})$ und $u(x_{n-2})$ um die Stelle x_n , und kombinieren Sie die somit vier Entwicklungen geschickt.

Ergebnis: Die zusätzlichen Taylorentwicklungen lauten:

$$u(x_{n+2}) = \sum_{k=0}^{\infty} \frac{(2h_x)^k}{k!} u(x_n)^{(k)} = u(x_n) + 2h_x \partial_x u(x_n) + \frac{(2h_x)^2}{2} \partial_{xx} u(x_n) + \frac{(2h_x)^3}{6} \partial_{xxx} u(x_n) + \dots$$

$$u(x_{n-2}) = \sum_{k=0}^{\infty} \frac{(-2h_x)^k}{k!} u(x_n)^{(k)} = u(x_n) - 2h_x \partial_x u(x_n) + \frac{(2h_x)^2}{2} \partial_{xx} u(x_n) - \frac{(2h_x)^3}{6} \partial_{xxx} u(x_n) \pm \dots$$

Der Differenzenquotient ist nur für die Punkte x_2, \dots, x_{N-1} definiert. Ein Ausweg ist die zusätzliche Konstruktion zweier einseitiger Differenzenquotienten gleicher Ordnung.



9. PDEs: Finite Differenzen I

Diffusions-Konvektions-Reaktionsgleichung

Wir betrachten das RWP

$$\begin{aligned}-\partial_{xx} u(x) + q(x)\partial_x u(x) + r(x)u(x) &= f(x) \quad x \in]0, 1[\\ u(0) = u(1) &= 0\end{aligned}$$

Diskretisieren Sie das Problem mit zentralen Differenzenquotienten zweiter Ordnung auf einem äquidistanten Gitter der Schrittweite h_x mit N inneren Gitterpunkten. Stellen Sie das LGS zur Bestimmung der Lösungsapproximation in den Gitterpunkten auf.



9. PDEs: Finite Differenzen I

Lösungshinweise: Es empfiehlt sich, zunächst die Ableitungen separat zu diskretisieren. Alle anderen Funktionen, sowie die „nullte Ableitung“ von u , werden einfach ausgewertet in den Gitterpunkten. Umformungen liefern dann eine (tridiagonale) Matrix, weil beide Differenzenquotienten nur jeweils einen Gitterpunkt nach links und rechts benötigen

Ergebnis: Einsetzen der Differenzenquotienten liefert

$$-\frac{u(x_{n+1}) - 2u(x_n) + u(x_{n-1})}{h_x^2} + q(x_n) \frac{u(x_{n+1}) - u(x_{n-1})}{2h_x} + r(x_n)u(x_n) \approx f(x_n)$$

Hauptnennerbildung liefert mit den üblichen Abkürzungen das LGS:

$$-\left(1 + \frac{q_n h_x}{2}\right) u_{n-1} + \left(2 + r_n h_x^2\right) u_n - \left(1 - \frac{q_n h_x}{2}\right) u_{n+1} = h_x^2 f_n \quad n = 1, \dots, N$$



10. PDEs: Finite Differenzen II



John von Neumann 1903–1957

The advance of analysis is, at this moment,
stagnant along the entire front of non-linear
problems.

John von Neumann 1946
(als Begründung zum Bau eines
virtuellen Windkanals basierend auf
numerischen Approximationen)



Ankündigungen



10. PDEs: Finite Differenzen II

- Evaluierungsergebnisse: Vielen Dank!
- Konkrete Kritikpunkte und Verbesserungsvorschläge
 - ▶ Einsicht in ViPLab-Punkte: schon von Anfang an, einfach alte Bearbeitung wieder öffnen
 - ▶ Berechtigte Korrekturen der Bewertung: Tutorinnen
 - ▶ Folien schon zu Semesterbeginn: nicht machbar beim Reboot einer Veranstaltung
- Massensprechstunden zur Klausurvorbereitung:
 - ▶ Dienstag 17.7. PWR57, Raum 8.135, 11:30 Uhr bis 15:30 Uhr
 - ▶ In der Woche vor der Klausur, Ankündigung über ILIAS



Wiederholung der letzten Vorlesung



10. PDEs: Finite Differenzen II

Aus kontinuumsmechanischen Überlegungen heraus haben wir die Wärmeleitungsgleichung hergeleitet, und mit normiertem Diffusionsquotient $\alpha = 1$ als Modellproblem identifiziert:

Definition 10.1 (Parabolisches Modellproblem, Wärmeleitungs-ARWP)

Es gelten die Bezeichner aus der vorherigen Vorlesung. Das **parabolische Modellproblem** besteht in der Aufgabe, eine Funktion $u: \Omega \times]0, T[\rightarrow \mathbb{R}$ zu finden, die die folgenden Bedingungen erfüllt:

$$\begin{aligned}\partial_t u(x, t) - \Delta u(x, t) &= f(x, t) && \text{für } (x, t) \in \Omega \times]0, T[\\ u(x, t) &= g(x, t) && \text{für } (x, t) \in \partial\Omega \times [0, T] \\ u(x, 0) &= u_0(x) && \text{für } x \in \Omega\end{aligned}$$

Dieses zeitabhängige Modellproblem steht im Fokus der heutigen Vorlesung.

Das Raclette-Problem in \mathbb{R}^d ist also eine Instanz dieses Modellproblems.



10. PDEs: Finite Differenzen II

Das zugehörige stationäre Modellproblem war das Poisson-Problem, das wir in VL 9 untersucht haben:

Definition 10.2 (Elliptisches Modellproblem, Poisson-RWP)

Es gelten die Bezeichner aus der vorherigen Vorlesung. Das **elliptische Modellproblem** besteht in der Aufgabe, eine Funktion $u: \Omega \rightarrow \mathbb{R}$ zu finden, die die folgenden Bedingungen erfüllt:

$$\begin{aligned}-\Delta u(x) &= f(x) && \text{für } x \in \Omega \\ u(x) &= g(x) && \text{für } x \in \partial\Omega\end{aligned}$$

Man kann beweisen: Unter gewissen Bedingungen konvergieren für $t \rightarrow \infty$ Lösungen $u(x, t)$ des parabolischen Modellproblems gegen Lösungen $u(x)$ des elliptischen Modellproblems. Die Sprechweise des **stationären Falls** ist also gerechtfertigt.



10. PDEs: Finite Differenzen II

Mit anderen konstitutiven Gesetzen haben wir aus der Kontinuumsmechanik eine weitere Differentialgleichung hergeleitet:

Homogene lineare Advektionsgleichung

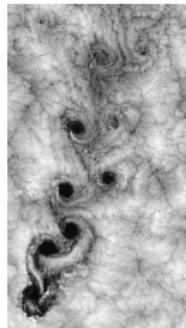
$$\partial_t u + \nabla(vu) = 0$$

Dieses Problem werden wir heute zu einem ARWP ausbauen, und ebenfalls im Detail untersuchen.



10. PDEs: Finite Differenzen II

Die drei Gleichungen ergeben sich oft als Teilprobleme in komplizierteren Fragestellungen Ihres Studiums, bspw. in der Strömungs- und Festkörpermechanik.



Umströmung einer (hohen) Insel, Wirbelschleppen hinter Flugzeugen

NASA public domain



Ergänzung der letzten VL: Erweiterung auf 2D



10. PDEs: Finite Differenzen II

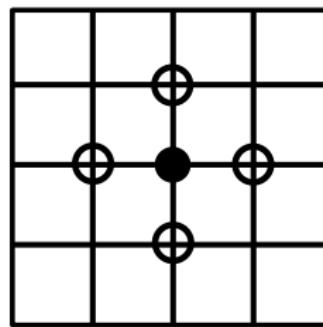
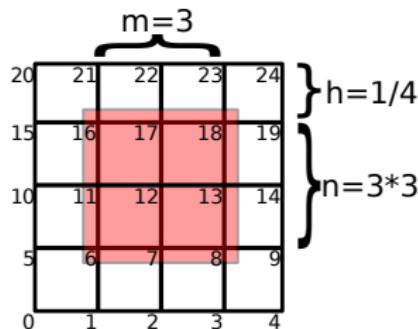
Die Vorgehensweise ist im Mehrdimensionalen sehr ähnlich. Als Beispiel betrachten wir $\Omega =]0, 1[^2$ und das Poisson-Problem

$$-\Delta u = -\partial_{x_1 x_1} u - \partial_{x_2 x_2} u = f.$$

Die Indizierung geschieht zur besseren Übersicht zunächst zweidimensional. Zu einem $M \in \mathbb{N}$ wählen wir die äquidistante Schrittweite $h_x = \frac{1}{M+1}$, und erhalten die Gitterpunkte

$$x_{mn} = (mh_x, nh_x) \quad \text{für } m, n = 0, \dots, M + 1.$$

Wir haben also M^2 innere Gitterpunkte und $(M + 2)^2$ insgesamt.





10. PDEs: Finite Differenzen II

Die Ableitungen diskretisieren wir auf diesem Gitter mit zentralen Differenzenquotienten zweiter Ordnung:

$$\begin{aligned}\partial_{x_1 x_1} u(x_{m,n}) &= \frac{u(x_{m-1,n}) - 2u(x_{m,n}) + u(x_{m+1,n})}{h_x^2} + \mathcal{O}(h_x^2) \\ \partial_{x_2 x_2} u(x_{m,n}) &= \frac{u(x_{m,n-1}) - 2u(x_{m,n}) + u(x_{m,n+1})}{h_x^2} + \mathcal{O}(h_x^2)\end{aligned}$$

Damit erhalten wir:

$$\Delta u(x_{m,n}) \approx \frac{u(x_{m+1,n}) + u(x_{m-1,n}) - 4u(x_{m,n}) + u(x_{m,n-1}) + u(x_{m,n+1})}{h_x^2}$$

Man spricht anschaulich auch von einem Differenzenstern, vergleiche die Abbildung auf der vorherigen Folie. Um die Approximation in den inneren Gitterpunkten zu berechnen, müssen wir also das LGS

$$-u_{m-1,n} - u_{m+1,n} + 4u_{m,n} - u_{m,n-1} - u_{m,n+1} = h_x^2 f_{m,n}, \quad m, n = 1, \dots, M$$

lösen. In der Praxis werden dazu die Gitterpunkte noch linear durchnummieriert, vgl. die Abbildung auf der vorherigen Folie.



Zeitabhängige lineare Diffusionsprozesse



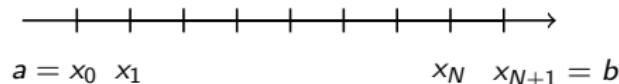
10. PDEs: Finite Differenzen II

Wir betrachten zuerst die Wärmeleitungsgleichung in 1D:

$$\begin{aligned}\partial_t u(x, t) - \alpha \partial_{xx} u(x, t) &= f(x, t) & (x, t) \in (a, b) \times [0, T] \\ u(a, t) = u(b, t) &= 0 & t \in [0, T] \quad (\text{Randwerte}) \\ u(x, 0) &= u_0(x) & x \in [a, b] \quad (\text{Anfangswert})\end{aligned}$$

Zur Diskretisierung verwenden wir die sogenannte **Linienmethode**, d. h. wir diskretisieren zuerst im Ort und danach in der Zeit. Wie in VL 9 wählen wir also ein $N \in \mathbb{N}$ und setzen

$$h_x := \frac{b - a}{N + 1}, \quad \text{und} \quad x_n := a + n h_x \quad \text{für } n = 0, \dots, N + 1.$$





10. PDEs: Finite Differenzen II

Auf diesem Gitter approximieren wir die zweite Ortsableitung von u an der Stelle x_n durch den zentralen Differenzenquotienten zweiter Ordnung:

$$\partial_{xx} u(x_n, t) \approx \frac{u(x_{n-1}, t) - 2u(x_n, t) + u(x_{n+1}, t)}{h_x^2}$$

Eingesetzt in die Wärmeleitungsgleichung erhalten wir:

$$\partial_t u(x_n, t) - \alpha \frac{u(x_{n-1}, t) - 2u(x_n, t) + u(x_{n+1}, t)}{h_x^2} \approx f(x_n, t)$$

Die Gleichung hängt nun nur noch von t ab, man spricht von einem **semidiskreten Problem**. Weil das für alle $n = 1, \dots, N$ gelten muss, besteht das semidiskrete Problem tatsächlich aus N gewöhnlichen Differentialgleichungen (in der Zeit), einer pro innerem Gitterpunkt.



10. PDEs: Finite Differenzen II

Wir suchen nun Approximationen $U_n(t) \approx u(x_n, t)$ bzw. $\dot{U}_n(t) \approx \partial_t u(x_n, t)$:

Semidiskrete Formulierung

Für $n = 1, \dots, N$ finde Funktionen $U_n : [0, T] \rightarrow \mathbb{R}$, so dass mit den Anfangswerten $U_n(0) = u_0(x_n)$ und den Randwerten $U_0(t) = U_{N+1}(t) = 0$ gilt:

$$\dot{U}_n(t) - \alpha \frac{U_{n-1}(t) - 2U_n(t) + U_{n+1}(t)}{h_x^2} = f(x_n, t)$$

In jedem diskreten inneren Ortsplatz x_1, \dots, x_N müssen wir eine kontinuierliche, nur noch von der Zeit abhängige Funktion $U_n : [0, T] \rightarrow \mathbb{R}$ bestimmen. Jede dieser Funktionen U_n ist eine Approximation an die gesuchte Lösung, allerdings nur in „ihrem“ Ortsplatz. Dabei machen wir einen Fehler, der für jede Funktion $\mathcal{O}(h_x^2)$ beträgt wegen der im Ort verwendeten zentralen Finiten Differenz zweiter Ordnung.



10. PDEs: Finite Differenzen II

$$\dot{U}_n(t) - \alpha \frac{U_{n-1}(t) - 2U_n(t) + U_{n+1}(t)}{h_x^2} = f(x_n, t)$$

Wir betrachten nur die inneren Gitterpunkte $n = 1, \dots, N$, weil an x_0 und x_{N+1} die Lösung durch die Vorgabe der Randwerte bereits bekannt ist, und zwar für alle Zeitpunkte. Wichtig ist zudem, dass die Ortsdiskretisierung eben durch die Vorgabe der Randpunkte wohldefiniert ist: Für $n = 1$ wird $U_0 = 0$, also die Approximation von u an x_0 benötigt, die wegen des linken Randwerts exakt existiert, analog für den rechten Rand mit $n = N$.



10. PDEs: Finite Differenzen II

$$\dot{U}_n(t) - \alpha \frac{U_{n-1}(t) - 2U_n(t) + U_{n+1}(t)}{h_x^2} = f(x_n, t)$$

Wir schreiben das Problem ausführlich hin:

$$\underbrace{\begin{pmatrix} \dot{U}_1(t) \\ \dot{U}_2(t) \\ \vdots \\ \dot{U}_{N-1}(t) \\ \dot{U}_N(t) \end{pmatrix}}_{=: \dot{\mathbf{U}}(t)} - \underbrace{\frac{\alpha}{h_x^2} \begin{pmatrix} \color{red}{U_0(t)} - 2U_1(t) + U_2(t) \\ U_1(t) - 2U_2(t) + U_3(t) \\ \vdots \\ U_{N-2}(t) - 2U_{N-1}(t) + U_N(t) \\ U_{N-1}(t) - 2U_N(t) + \color{red}{U_{N+1}(t)} \end{pmatrix}}_{=: \mathbf{A}\mathbf{U}(t)} = \underbrace{\begin{pmatrix} f(x_1, t) \\ f(x_2, t) \\ \vdots \\ f(x_{N-1}, t) \\ f(x_N, t) \end{pmatrix}}_{=: \mathbf{b}(t)}$$

Die rot markierten Ausdrücke sind Null wegen der Randwerte.



10. PDEs: Finite Differenzen II

$$\underbrace{\begin{pmatrix} \dot{U}_1(t) \\ \dot{U}_2(t) \\ \vdots \\ \dot{U}_{N-1}(t) \\ \dot{U}_N(t) \end{pmatrix}}_{=: \dot{\mathbf{U}}(t)} - \frac{\alpha}{h_x^2} \underbrace{\begin{pmatrix} 0 & -2U_1(t) & +U_2(t) \\ +U_1(t) & -2U_2(t) & +U_3(t) \\ & \vdots & \\ +U_{N-2}(t) & -2U_{N-1}(t) & +U_N(t) \\ +U_{N-1}(t) & -2U_N(t) & +0 \end{pmatrix}}_{=: \mathbf{A}\mathbf{U}(t)} = \underbrace{\begin{pmatrix} f(x_1, t) \\ f(x_2, t) \\ \vdots \\ f(x_{N-1}, t) \\ f(x_N, t) \end{pmatrix}}_{=: \mathbf{b}(t)}$$

Mit **einem kleinen Vorzeichentausch** und

- $\mathbf{A} = \frac{\alpha}{h_x^2} \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{N \times N}$,
- $\mathbf{U}(t) = (U_1(t), \dots, U_N(t))^T : \mathbb{R}^N \rightarrow \mathbb{R}^N$,
- $\dot{\mathbf{U}}(t) = (\dot{U}_1(t), \dots, \dot{U}_N(t))^T : \mathbb{R}^N \rightarrow \mathbb{R}^N$
- $\mathbf{b}(t) = (f(x_1, t), \dots, f(x_N, t))^T \in \mathbb{R}^N$,
- $\mathbf{u}_0 = (u_0(x_1), \dots, u_0(x_N))^T \in \mathbb{R}^N$

ist dies äquivalent zum semidiskreten kompakt geschriebenen System

$$\dot{\mathbf{U}}(t) + \mathbf{A}\mathbf{U}(t) = \mathbf{b}(t) \quad \mathbf{U}(0) = \mathbf{u}_0.$$



10. PDEs: Finite Differenzen II

$$\dot{\mathbf{U}}(t) + \mathbf{A}\mathbf{U}(t) = \mathbf{b}(t) \quad \mathbf{U}(0) = \mathbf{u}_0$$

Die ausführliche Formulierung lautet:

$$\underbrace{\begin{pmatrix} \dot{U}_1(t) \\ \dot{U}_2(t) \\ \vdots \\ \dot{U}_{N-1}(t) \\ \dot{U}_N(t) \end{pmatrix}}_{=\dot{\mathbf{U}}(t)} + \frac{\alpha}{h_x^2} \underbrace{\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}}_{=\mathbf{A}} \underbrace{\begin{pmatrix} U_1(t) \\ U_2(t) \\ \vdots \\ U_{N-1}(t) \\ U_N(t) \end{pmatrix}}_{=\mathbf{U}(t)} = \underbrace{\begin{pmatrix} f(x_1, t) \\ f(x_2, t) \\ \vdots \\ f(x_{N-1}, t) \\ f(x_N, t) \end{pmatrix}}_{=\mathbf{b}(t)}$$

Weil wir hier im Gegensatz zu VL 0 homogene Dirichlet- Randbedingungen vorschreiben, muss die rechte Seite $\mathbf{b}(t) = (f(x_1, t), \dots, f(x_N, t))^T$ nicht modifiziert werden wegen der unterwegs herausgekegelter Nullbeiträge.



10. PDEs: Finite Differenzen II

Ergebnis: Semidiskrete Ortsdiskretisierung der
Wärmeleitungsgleichung

$$\dot{\mathbf{U}}(t) + \mathbf{A}\mathbf{U}(t) = \mathbf{b}(t) \quad t \in [0, T], \quad \mathbf{U}(0) = \mathbf{u}_0.$$

Dieses semidiskrete Problem ist ein System von (gewöhnlichen) AWP wie in VL 8!

$$\dot{\mathbf{U}}(t) = \mathbf{F}(t, \mathbf{U}) := \mathbf{b}(t) - \mathbf{A}\mathbf{U}(t)$$

Wir können also die aus VL 8 bekannten Verfahren zur Zeitdiskretisierung dieses semidiskreten AWPs verwenden.

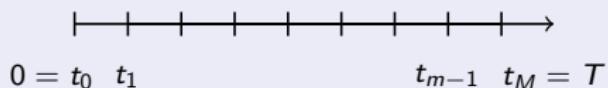
Bemerkung zur Anschauung: Wir approximieren die Lösung des vollständigen Problems durch diese Vorgehensweise entlang von (Zeit-) Linien (x_n, t) für $t \in [0, T]$. Daher heißt dieser Ansatz auch „Linienmethode“ (engl. ‘Method of Lines’), weil wir im Orts-Zeit-Zylinder ausgehend von den Anfangswerten pro diskretem Ortspunkt eine ODE in der Zeit lösen.



10. PDEs: Finite Differenzen II

Erinnerung: Zeitdiskretisierung

Wähle $M \in \mathbb{N}$, $h_t = \frac{T}{M}$ und setze $t_m = mh_t$, $m = 0, \dots, M$:



Im Folgenden übertragen wir die Methoden zur Diskretisierung gewöhnlicher Anfangswertprobleme aus VL 8 auf das semidiskrete Problem.

Wir starten mit dem expliziten Euler-Verfahren zur Approximation von $\mathbf{U}(t_m) \approx \mathbf{U}^{(m)}$ mit Anfangswert $\mathbf{U}^{(0)} = (u_0(x_1), \dots, u_0(x_N))^T$:

$$\mathbf{U}^{(m)} = \mathbf{U}^{(m-1)} + h_t \mathbf{F} \left(t_{m-1}, \mathbf{U}^{(m-1)} \right)$$

Für die Wärmeleitung haben wir $\mathbf{F} = \mathbf{b} - \mathbf{A}\mathbf{U}^{(m-1)}$ unabhängig von t_{m-1} . $\mathbf{U}^{(m)} \in \mathbb{R}^N$ ist ein Vektor, der komponentenweise für die Ortspunkte die (volldiskrete) Approximation der Lösung im m -ten Zeitpunkt enthält.



Satz 10.3 (Explizites Euler-Verfahren für Wärmeleitung mit FD)

Das **explizite Euler-Verfahren** für die Wärmeleitungsgleichung mit FD-Ortsdiskretisierung (semidiskretes Problem, vgl. Folie 593) lautet: Setze $\mathbf{U}^{(0)} = (u_0(x_1), \dots, u_0(x_N))^T$ und bestimme für $m = 1, \dots, M$ iterativ:

$$\mathbf{U}^{(m)} = \mathbf{U}^{(m-1)} + h_t \mathbf{b} - h_t \mathbf{A} \mathbf{U}^{(m-1)} = (\mathbf{I} - h_t \mathbf{A}) \mathbf{U}^{(m-1)} + h_t \mathbf{b},$$

\mathbf{A} ist eine Tridiagonalmatrix, also auch $\mathbf{I} - h_t \mathbf{A}$. Die Berechnung von $\mathbf{U}^{(m)}$ benötigt die Multiplikation dieser Tridiagonalmatrix mit dem Vektor $\mathbf{U}^{(m-1)}$, d. h. wie in VL 8 eine reine (Matrix-) Formelauswertung. Zum besseren Verständnis schreiben wir das auch komponentenweise:

$$U_n^{(m)} = U_n^{(m-1)} + \frac{h_t \alpha}{h_x^2} \left(U_{n-1}^{(m-1)} - 2U_n^{(m-1)} + U_{n+1}^{(m-1)} \right) + h_t f(x_n, t_{m-1})$$

für $n = 1, \dots, N$ und mit den Randwerten $U_0^{(m-1)} = U_{N+1}^{(m-1)} = 0$.



10. PDEs: Finite Differenzen II

Analog ist das implizite Euler-Verfahren aus VL 8 anwendbar, wenn wir rechts $\mathbf{U}^{(m)}$ statt $\mathbf{U}^{(m-1)}$ einsetzen:

Satz 10.4 (Implizites Euler-Verfahren für Wärmeleitung mit FD)

Das **implizite Euler-Verfahren** für die Wärmeleitungsgleichung mit FD-Ortsdiskretisierung lautet: Setze $\mathbf{U}(0) = (u_0(x_1), \dots, u_0(x_N))^T$ und bestimme für $m = 1, \dots, M$ iterativ

$$\mathbf{U}^{(m)} = \mathbf{U}^{(m-1)} + h_t \left(\mathbf{b} - \mathbf{A} \mathbf{U}^{(m)} \right)$$

Man beachte, dass diese Verfahren prinzipiell unabhängig von der Ortsdiskretisierung formulierbar sind. Wenn wir im Ort andere Differenzenquotienten verwenden (oder sogar andere Differentialoperatoren), so ändert sich nur die Matrix \mathbf{A} .



10. PDEs: Finite Differenzen II

Weil \mathbf{A} linear ist, können wir das implizite Euler-Verfahren etwas umformulieren:

$$\begin{aligned}\mathbf{U}^{(m)} &= \mathbf{U}^{(m-1)} + h_t \left(\mathbf{b} - \mathbf{A} \mathbf{U}^{(m)} \right) = \mathbf{U}^{(m-1)} + h_t \mathbf{b} - h_t \mathbf{A} \mathbf{U}^{(m)} \\ \Leftrightarrow \quad \mathbf{U}^{(m)} + h_t \mathbf{A} \mathbf{U}^{(m)} &= \mathbf{U}^{(m-1)} + h_t \mathbf{b} \\ \Leftrightarrow \quad \mathbf{U}^{(m)} &= (\mathbf{I} + h_t \mathbf{A})^{-1} \left(\mathbf{U}^{(m-1)} + h_t \mathbf{b} \right)\end{aligned}$$

Die Berechnung von $\mathbf{U}^{(m)}$ benötigt also die Lösung eines LGS mit der Tridiagonalmatrix $\mathbf{I} + h_t \mathbf{A}$. Das Newton-Verfahren ist hier nicht nötig.



Stabilität der Diskretisierung

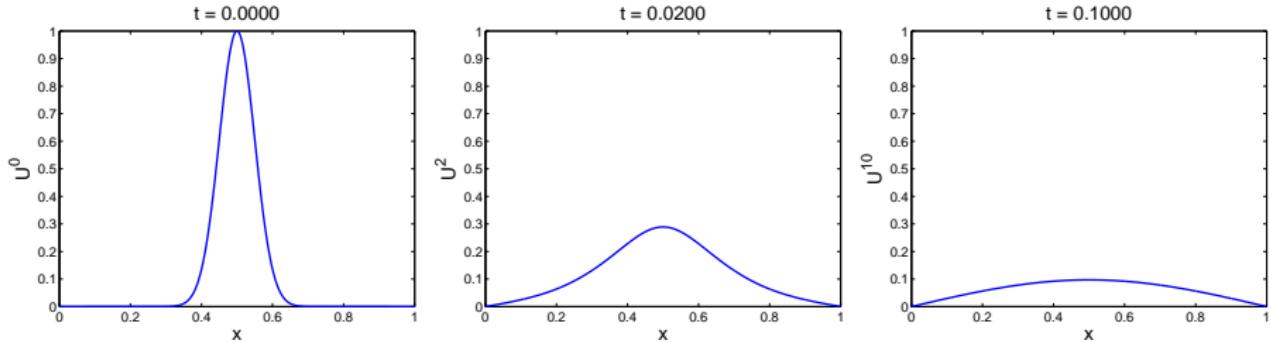


10. PDEs: Finite Differenzen II

Beispiel 10.5

Wir betrachten die homogene Wärmeleitungsgleichung mit $f \equiv 0$, das Ortsintervall $[0, 1]$ sowie homogene Dirichlet-Randbedingungen und den Anfangswert $u_0(x) = e^{-50(x-\frac{1}{2})^2}$.

Die exakte Lösung zu verschiedenen Zeitpunkten sieht über dem Ortsintervall wie folgt aus (vgl. ViPLab-Demo):

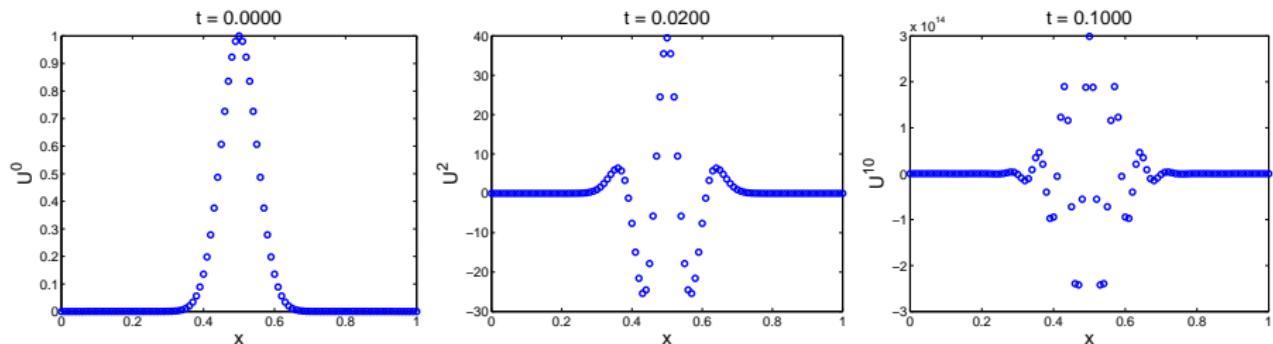


Das entspricht unserer Erwartung: Ohne externe Kräfte „diffundiert“ der Anfangswert mit der Zeit weg.



10. PDEs: Finite Differenzen II

Das explizite Euler-Verfahren mit $h_x = 0.01$ und $h_t = 0.01$ ergibt:

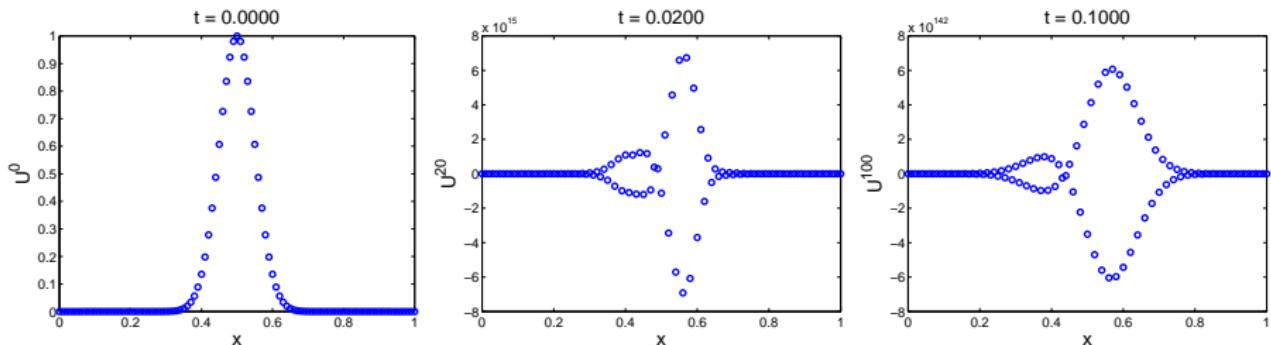


Mit diesen Parametern liefert das Verfahren große Oszillationen, und keine vernünftige numerische Lösung: Man beachte die unterschiedliche Skalierung der Plots.



10. PDEs: Finite Differenzen II

Das explizite Euler-Verfahren mit $h_x = 0.01$ und $h_t = 0.001$ ergibt:

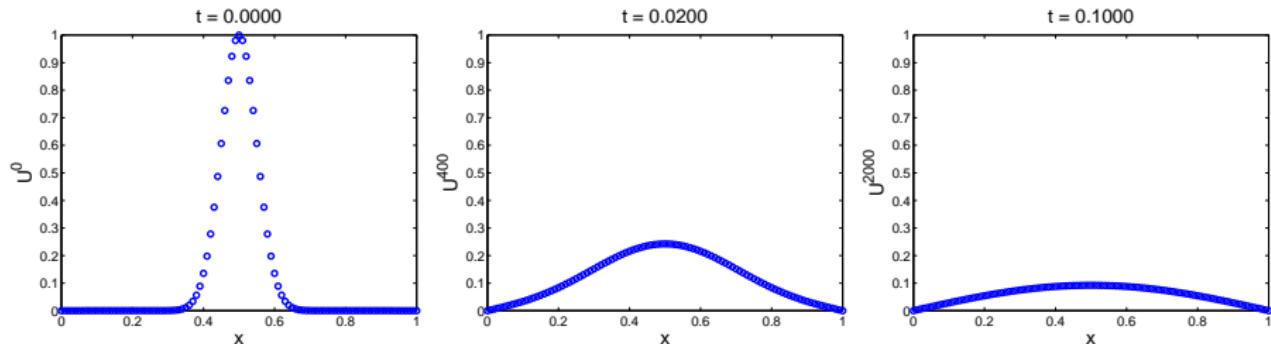


Mit diesen Parametern liefert das Verfahren ebenfalls große Oszillationen, und keine vernünftige numerische Lösung: Man beachte die unterschiedliche Skalierung der Plots. Scheint also die Reduzierung der Zeitschrittweite nicht zu helfen?



10. PDEs: Finite Differenzen II

Wir gönnen uns noch einen letzten Versuch. Das explizite Euler-Verfahren mit $h_x = 0.01$ und $h_t = 0.0001$ ergibt:

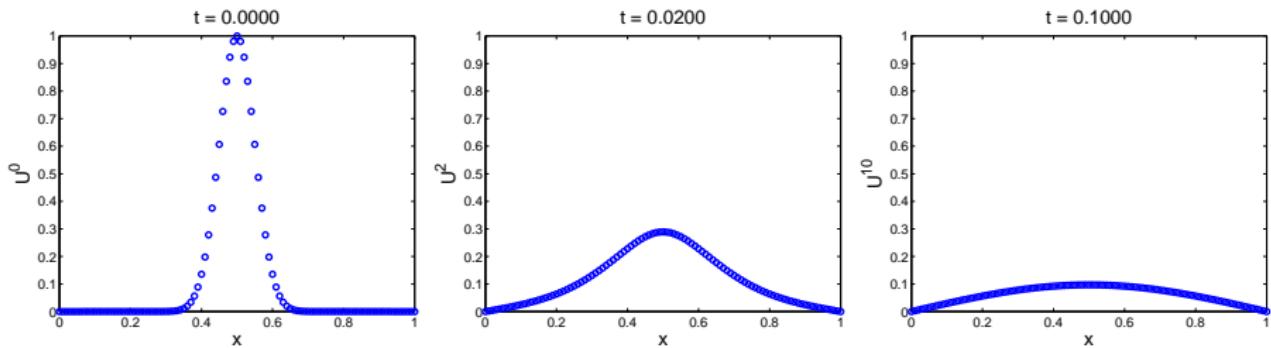


Mit diesen Parametern wird die exakte Lösung gut approximiert. Scheinbar muss also die Zeitschrittweite h_t im Vergleich zur Ortsschrittweite h_x winzig gewählt werden. Der Verdacht $h_t = h_x^2$ drängt sich auf.



10. PDEs: Finite Differenzen II

Das implizite Euler-Verfahren mit $h_x = 0.01$ und $h_t = 0.01$ ergibt:



Schon mit dieser groben Zeitschrittweite h_t wird die exakte Lösung gut approximiert, es scheint keine (starke) Verschränkung der Orts- und Zeitschrittweiten zu geben. Beides wollen wir nun quantitativ untersuchen.



10. PDEs: Finite Differenzen II

Der homogene Fall $f \equiv 0$ entspricht anschaulich für die Wärmeleitung dem Szenario ohne externe Wärmequellen. Intuitiv ist klar, dass dann die Temperatur nie höher werden kann als zu Beginn. Für die kontinuierliche Lösung kann man dies tatsächlich beweisen. Deshalb macht es Sinn, genau das auch für die diskrete Approximation der Lösung zu fordern:

Definition 10.6 (Stabilität von Zeitschrittverfahren)

Ein diskretes Verfahren zur Lösung des homogenen Wärmeleitungs-ARWP (d. h. mit $f \equiv 0$) heißt genau dann (Zeit-) stabil, falls für beliebige $\mathbf{U}^{(0)} \in \mathbb{R}^N$ gilt:

$$\|\mathbf{U}^{(m)}\|_2 \leq \|\mathbf{U}^{(0)}\|_2 \quad \forall m = 1, \dots, M$$



10. PDEs: Finite Differenzen II

Aus $f \equiv 0$ erhalten wir nach der Diskretisierung $\mathbf{b} \equiv \mathbf{0}$, so dass die beiden Euler-Verfahren lauten:

$$\begin{aligned}\mathbf{U}^{(m)} &= (\mathbf{I} - h_t \mathbf{A}) \mathbf{U}^{(m-1)} \quad + h_t \mathbf{b} \\ \mathbf{U}^{(m)} &= (\mathbf{I} + h_t \mathbf{A})^{-1} \mathbf{U}^{(m-1)} \quad + \underline{h_t (\mathbf{I} + h_t \mathbf{A})^{-1}} \mathbf{b}\end{aligned}$$

In Operatorschreibweise erhalten wir also:

$$\mathbf{U}^{(m)} = \mathbf{B} \mathbf{U}^{(m-1)} = \mathbf{B}^2 \mathbf{U}^{(m-2)} = \dots \mathbf{B}^m \mathbf{U}^{(0)} \quad (10.16)$$

mit $\mathbf{B} = (\mathbf{I} - h_t \mathbf{A})$ bzw. $\mathbf{B} = (\mathbf{I} + h_t \mathbf{A})^{-1}$ für das explizite bzw. implizite Euler-Verfahren. Nach Konstruktion ist die Matrix \mathbf{B} symmetrisch, vgl. Folie 592.



10. PDEs: Finite Differenzen II

Nach dieser Umformung können wir einen wichtigen Satz formulieren und beweisen:

Satz 10.7 (Stabilität der Euler-Verfahren)

Die Euler-Verfahren sind genau dann stabil für die Ortsdiskretisierung mit Finiten Differenzen, falls $\|\mathbf{B}\|_2 \leq 1$.

Da \mathbf{B} symmetrisch ist, ist der Wert der Spektralnorm $\|\mathbf{B}\|_2$ gerade der betragsgrößte Eigenwert. Nach Konstruktion hängt \mathbf{B} und damit $\|\mathbf{B}\|_2$ von α und h_x (aus \mathbf{A}) sowie h_t (aus den Euler-Verfahren) ab.

Die Matrix \mathbf{B} ist unterschiedlich für das explizite und implizite Euler-Verfahren, und daher werden unterschiedliche Bedingungen an α , h_t und h_x gelten, um $\|\mathbf{B}\|_2 \leq 1$ sicherzustellen.



10. PDEs: Finite Differenzen II

Beweis: „ \Leftarrow “: Wir gehen in $\mathbf{U}^{(m)} = \mathbf{B}\mathbf{U}^{(m-1)} = \mathbf{B}^2\mathbf{U}^{(m-2)} = \dots \mathbf{B}^m\mathbf{U}^{(0)}$ zur Spektralnorm über und nutzen die Submultiplikativität (vgl. VL 2):

$$\|\mathbf{U}^{(m)}\|_2 \leq \|\mathbf{B}\|_2^m \|\mathbf{U}^{(0)}\|_2$$

Falls nun $\|\mathbf{B}\|_2 \leq 1$, können wir direkt die Stabilität $\|\mathbf{U}^{(m)}\|_2 \leq \|\mathbf{U}^{(0)}\|_2$ ablesen.

„ \Rightarrow “ durch Widerspruch: Sei λ der betragsgrößte EW von \mathbf{B} , d. h. $\|\mathbf{B}\|_2 = |\lambda|$, und sei $\mathbf{U}^{(0)} \neq \mathbf{0}$ ein zugehöriger EV. Wir nehmen an dass $\|\mathbf{B}\|_2 = |\lambda| > 1$ ist, und erhalten mit der Homogenität der Norm (vgl. VL 2):

$$\|\mathbf{U}^{(m)}\|_2 \leq \|\mathbf{B}\|_2^m \|\mathbf{U}^{(0)}\|_2 = |\lambda|^m \|\mathbf{U}^{(0)}\|_2 \rightarrow \infty \quad m \rightarrow \infty.$$

Widerspruch, also muss $\|\mathbf{B}\|_2 \leq 1$ sein. □



10. PDEs: Finite Differenzen II

Für die Stabilität der Verfahren muss also $\text{sp}(\mathbf{B}) \leq 1$, d. h. $\text{sp}(\mathbf{B}) \subset [-1, 1]$ gelten.
Für die Tridiagonalmatrix in unserem Beispiel ist das Spektrum bekannt:

Satz 10.8 (Eigenwerte der Wärmeleitungs-Ortsdiskretisierung)

Die Eigenwerte der Tridiagonalmatrix \mathbf{M}

$$\mathbf{M} = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{N \times N}$$

sind

$$\lambda_n = 2 + 2 \cos\left(\frac{n\pi}{N+1}\right) \quad n = 1, \dots, N.$$

Insbesondere gilt $\text{sp}(\mathbf{M}) \subset]0, 4[$.

Für unsere Überlegungen folgt: $\mathbf{A} = \frac{\alpha}{h_x^2} \mathbf{M}$ und damit $\text{sp}(\mathbf{A}) \subset]0, 4 \frac{\alpha}{h_x^2}[$.



10. PDEs: Finite Differenzen II

Satz 10.9 (Bedingte Stabilität des expliziten Euler-Verfahrens)

Das explizite Euler-Verfahren für das Wärmeleitungs-ARWP mit einer zentralen Finite Differenzen Ortsdiskretisierung zweiter Ordnung ist stabil für

$$h_t \leq \frac{h_x^2}{2\alpha}.$$

Beweis: Mit $\gamma = h_t \frac{\alpha}{h_x^2}$ ist $\mathbf{B} = \mathbf{I} - h_t \mathbf{A} = \mathbf{I} - \gamma \mathbf{M}$. μ ist EW von \mathbf{M} genau dann, wenn $\lambda = 1 - \gamma\mu$ EW von \mathbf{B} ist, vgl. VL 4.

Ist $0 < \mu < 4$ ein EW von \mathbf{M} , so muss für die Stabilität gelten:

$$-1 \leq \lambda = 1 - \gamma\mu \leq 1 \Leftrightarrow \gamma\mu \leq 2 \Leftrightarrow \gamma = h_t \frac{\alpha}{h_x^2} \leq \frac{2}{\mu} \quad (10.17)$$

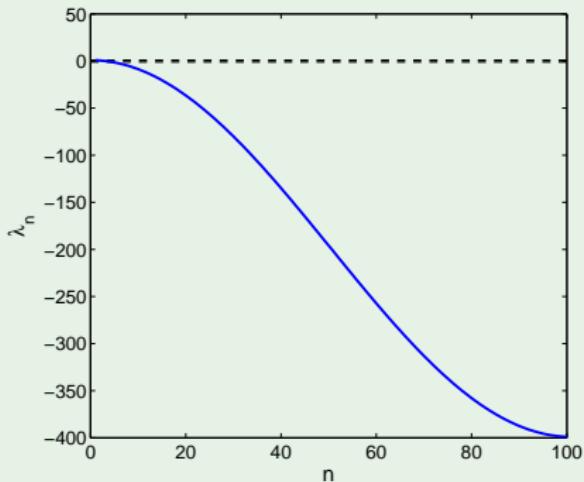
Wegen $\mu \in]0, 4[$ und $h_t \leq \frac{h_x^2}{2\alpha}$ ist die rechte Ungleichung in (10.17) erfüllt. □



10. PDEs: Finite Differenzen II

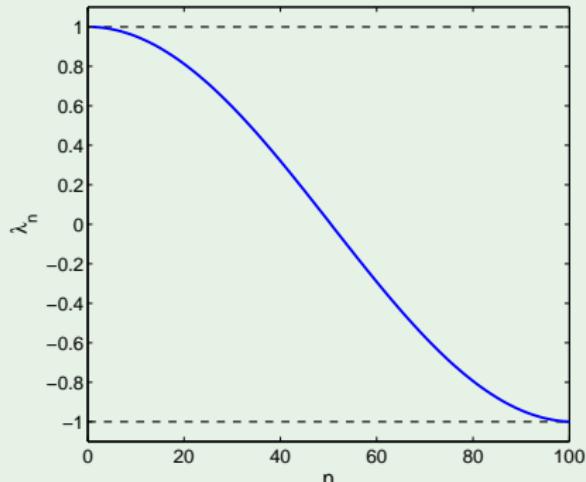
Beispiel 10.10 (Eigenwerte des expliziten Euler-Verfahrens)

$$\alpha = 1, h_x = 0.01 \text{ und } h_t = h_x = 0.01$$



$$\text{sp}(\mathbf{I} - h_t \mathbf{A}) \subset [-398.9033, 0.9032565]$$

$$\alpha = 1, h_x = 0.01 \text{ und } h_t = h_x^2 = 0.0001$$



$$\text{sp}(\mathbf{I} - h_t \mathbf{A}) \subset [0.0024944, 0.9117902]$$

Dies erklärt die beobachtete Instabilität bzw. Stabilität der numerischen Experimente von oben.



10. PDEs: Finite Differenzen II

Satz 10.11 (Unbedingte Stabilität des impliziten Euler-Verfahrens)

Das implizite Euler-Verfahren das Wärmeleitungs-ARWP mit einer zentralen Finite Differenzen Ortsdiskretisierung zweiter Ordnung ist für alle $h_t > 0$ unabhängig von h_x und α stabil.

Beweis: Mit $\gamma = h_t \frac{\alpha}{h_x^2}$ ist

$$\mathbf{B} = (\mathbf{I} + h_t \frac{\alpha}{h_x^2} \mathbf{M})^{-1} = (\mathbf{I} + \gamma \mathbf{M})^{-1}.$$

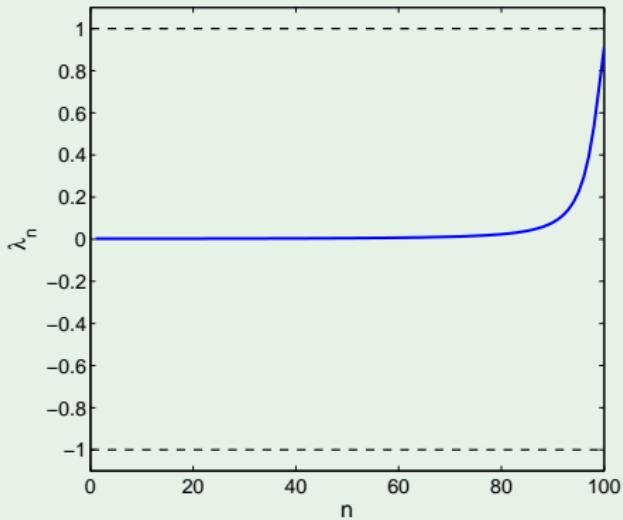
$\mu \in]0, 4[$ ist EW von \mathbf{M} genau dann, wenn $\lambda = \frac{1}{1+\gamma\mu}$ EW von \mathbf{B} ist.

Da $\mu \in]0, 4[$ und $1 + \gamma\mu = 1 + h_t \frac{\alpha}{h_x^2} \mu > 1$ ist, folgt $\lambda < 1$ also $\text{sp}(\mathbf{B}) \subset [0, 1]$. □



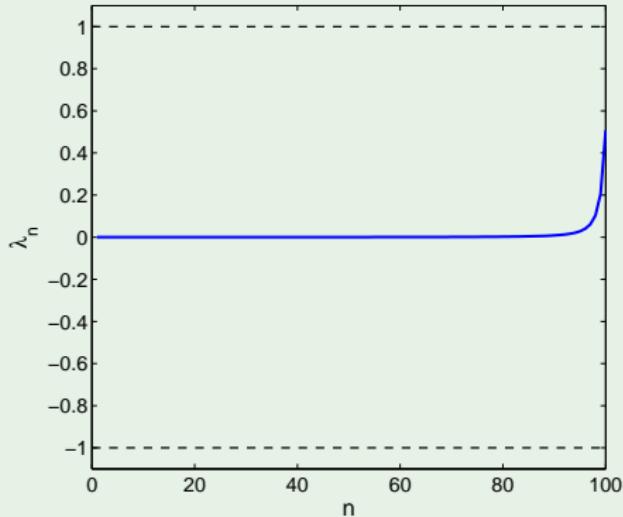
Beispiel 10.12 (Eigenwerte des impliziten Euler-Verfahrens)

$$\alpha = 1, h_x = 0.01 \text{ und } h_t = h_x = 0.01$$



$$\text{sp}((\mathbf{I} + h_t \mathbf{A})^{-1}) \subset [0.00249437, 0.9117902]$$

$$\alpha = 1, h_x = 0.01 \text{ und } h_t = \sqrt{h_x} = 0.1$$



$$\text{sp}((\mathbf{I} + h_t \mathbf{A})^{-1}) \subset [0.000249, 0.5082759]$$



10. PDEs: Finite Differenzen II

Die Resultate in diesem Abschnitt untermauern nochmals die Bedeutung von impliziten Verfahren: Die bewiesene Zeitschrittweitenbedingung macht das explizite Euler-Verfahren für das Wärmeleitungsgleichung-Modellproblem faktisch unbrauchbar in der Praxis.

Wir haben die Konvergenzordnung $\mathcal{O}(h_x^2)$ für die verwendete Ortsdiskretisierung nachgewiesen. In VL 8 haben wir zudem gezeigt, dass für die beiden Euler-Verfahren $\mathcal{O}(h_t)$ gilt. Die betrachteten Verfahren besitzen die Gesamtordnung $\mathcal{O}(h_x^2 + h_t)$. Das muss man natürlich noch beweisen.



Lineare Advektionsprobleme



10. PDEs: Finite Differenzen II

Wir betrachten abschließend kurz die letzte Klasse von Modellproblemen, nämlich das **hyperbolische** Advektionsproblem.

Problem 10.13 (Advektionsgleichung in 1D)

Zu gegebenem $v: \mathbb{R} \times \mathbb{R}_{>0} \rightarrow \mathbb{R}$ und $u_0: \mathbb{R} \rightarrow \mathbb{R}$ bestimme $u: \mathbb{R} \times \mathbb{R}_{>0} \rightarrow \mathbb{R}$ mit

$$\begin{aligned}\partial_t u(x, t) + \partial_x (v(x, t) u(x, t)) &= 0 & (x, t) \in \mathbb{R} \times \mathbb{R}_{>0}, \\ u(x, 0) &= u_0 & x \in \mathbb{R}.\end{aligned}$$

u ist z. B. die Dichte eines Fluids und v dessen Geschwindigkeit. Dann beschreibt dieses Problem physikalisch die Massenerhaltung, vgl. VL 9.

Bei beschränkten Ortsgebieten ist zusätzlich die Vorgabe von Randbedingungen auf dem Einflussrand erforderlich. Dies hängt vom Verhältnis von Ω und v ab. Die Vorgabe von Randbedingungen ist bei hyperbolischen Problemen in unbeschränkten Gebieten nicht erforderlich.



10. PDEs: Finite Differenzen II

Wir führen wieder zwei theoretische Resultate an:

Satz 10.14 (Existenz und Eindeutigkeit)

Ist v stetig differenzierbar, so besitzt Problem 10.13 eine eindeutige Lösung. Diese Lösung lautet $u(x, t) = u_0(x - v(x, t)t)$

Satz 10.15 (Stabilität)

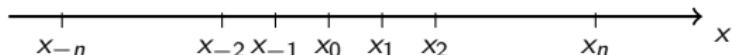
Sei v konstant und u_0 differenzierbar. Dann gilt für beliebiges $t > 0$:

$$\sup_{x \in \mathbb{R}} |u(x, t)| = \sup_{x \in \mathbb{R}} |u_0(x)|$$



10. PDEs: Finite Differenzen II

Die getrennte Orts- und Zeitdiskretisierung funktioniert (fast) genauso wie eben:
Wir wählen h_x und definieren Gitterpunkte $x_n = n h_x$ für $n \in \mathbb{Z}$, da das
Ortsintervall ganz \mathbb{R} ist.



Auf diesem Gitter approximieren wir die Ableitung mit den Vorwärts- oder Rückwärtsdifferenzen

$$\partial_x u(x_n, t) \approx \frac{u(x_{n+1}, t) - u(x_n, t)}{h_x} \quad \partial_x u(x_n, t) \approx \frac{u(x_n, t) - u(x_{n-1}, t)}{h_x}$$

mit Fehler $\mathcal{O}(h_x)$, oder mit der zentralen Differenz

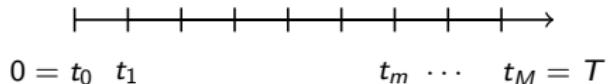
$$\partial_x u(x_n, t) \approx \frac{u(x_{n+1}, t) - u(x_{n-1}, t)}{2h_x}$$

mit Fehler $\mathcal{O}(h_x^2)$



10. PDEs: Finite Differenzen II

Zur Zeitdiskretisierung des Intervalls $[0, T]$ setzen wir $h_t = \frac{T}{M}$, d. h. $t_m = m h_t$, $m \in \mathbb{N}_0$:



Analog zur Wärmeleitungsgleichung erhalten wir exemplarisch:

Satz 10.16 (Expliziter Euler für Advektion mit zentralen FD)

Setze $U_n^{(0)} = u_0(x_n)$ für $n \in \mathbb{Z}$. Für $m \geq 1$ berechne iterativ

$$U_n^{(m)} := U_n^{(m-1)} - v(x_n, t_{m-1}) \frac{h_t}{2h_x} \left(U_{n+1}^{(m-1)} - U_{n-1}^{(m-1)} \right) \quad n \in \mathbb{Z}.$$



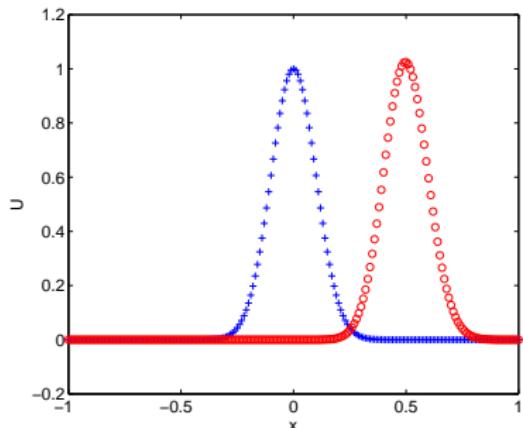
Stabilität des Verfahrens



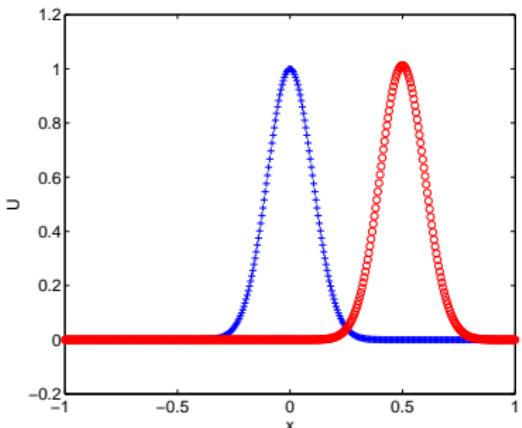
Beispiel 10.17 (Glatte Anfangsdaten)

Wir verwenden als Anfangswerte $u_0(x) = e^{-50x^2}$, Geschwindigkeit $v = 1$ und als Endzeitpunkt $T = 0.5$.

$$h_x = 0.01 \text{ und } h_t = 0.1 h_x$$



$$h_x = 0.005 \text{ und } h_t = 0.1 h_x$$



Anfangs- (blau) und Endzustand (rot) bei Verwendung des expliziten Euler-Verfahrens mit zentralen Finiten Differenzen im Ort

Das sieht schon sehr gut aus.

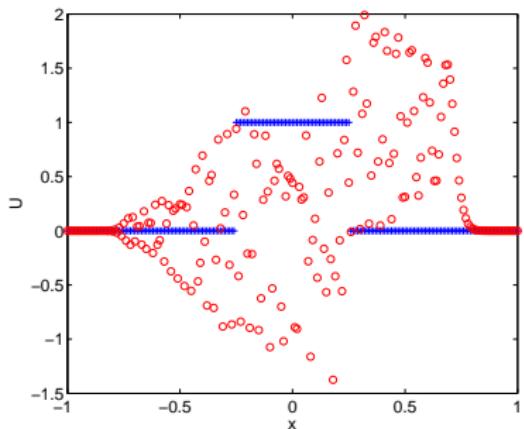


10. PDEs: Finite Differenzen II

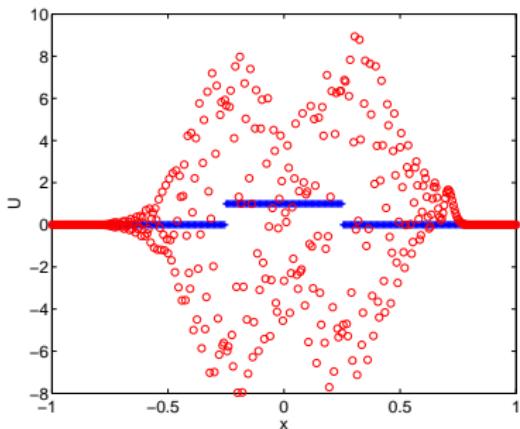
Beispiel 10.18 (Nichtglatte Anfangsdaten)

Wir verwenden als Anfangswert die charakteristische Funktion des Intervalls $[-0.25, 0.25]$, d. h. $u_0(x) = \mathbf{1}_{[-0.25, 0.25]}(x)$ und als Endzeitpunkt $T = 0.5$.

$$h_x = 0.01 \text{ und } h_t = 0.1 h_x$$



$$h_x = 0.005 \text{ und } h_t = 0.1 h_x$$



Das sieht ausgesprochen schlecht aus.



10. PDEs: Finite Differenzen II

Wir lesen an der exakten Lösung

$$u(x, t) = u_0(x - v(x, t)t)$$

ab, dass für $v > 0$ die Information von links nach rechts und für $v < 0$ von rechts nach links „fließt“. Diese Eigenschaft können wir im Verfahren nutzen.

Satz 10.19 (Upwind-Diskretisierung)

Sei $v > 0$. Das explizite Euler-Verfahren für die Advektionsgleichung mit Rückwärts-Differenzen lautet:

Setze $U_n^{(0)} = u_0(x_n)$ für $n \in \mathbb{Z}$, und berechne für $m \geq 1$ iterativ

$$U_n^{(m)} := U_n^{(m-1)} - v(x_n, t_{m-1}) \frac{h_t}{h_x} (U_{\textcolor{red}{n}}^{(m-1)} - U_{n-1}^{(m-1)}) \quad n \in \mathbb{Z}.$$



10. PDEs: Finite Differenzen II

$$U_n^{(m)} := U_n^{(m-1)} - v(x_n, t_{m-1}) \frac{h_t}{h_x} \left(U_n^{(m-1)} - U_{n-1}^{(m-1)} \right) \quad n \in \mathbb{Z}$$

Die einzige Änderung ist also, dass wir von einem zentralen Differenzenquotienten für die erste Ableitung zu einem einseitigen Differenzenquotienten übergehen, konkret zur Rückwärtsdifferenz. Dabei verlieren wir die quadratische Ordnung.

Upwind („Aufwind“) bedeutet, dass Information von $U_{n-1}^{(m-1)}$ nach $U_n^{(m)}$ transportiert wird.

Der Fall $v < 0$ wird in den Miniübungen betrachtet.

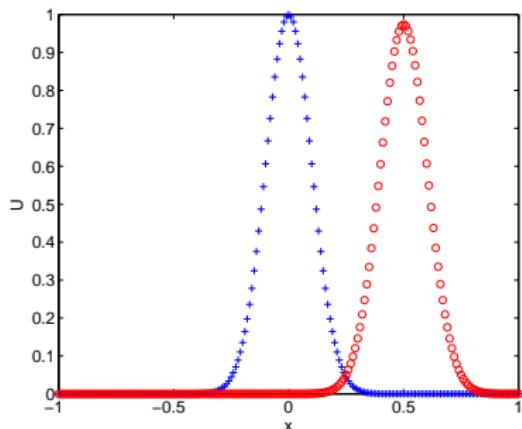
Falls die Aufwind-Analogie nicht hilft, macht das nichts. Dann helfen die Formeln.

10. PDEs: Finite Differenzen II

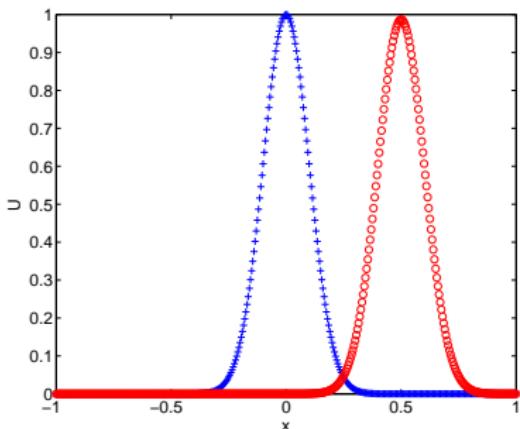


Beispiel: Upwind Diskretisierung für Beispiel 10.17 (glatte Daten)

$$h_x = 0.01 \text{ und } h_t = 0.9 h_x$$



$$h_x = 0.005 \text{ und } h_t = 0.9 h_x$$



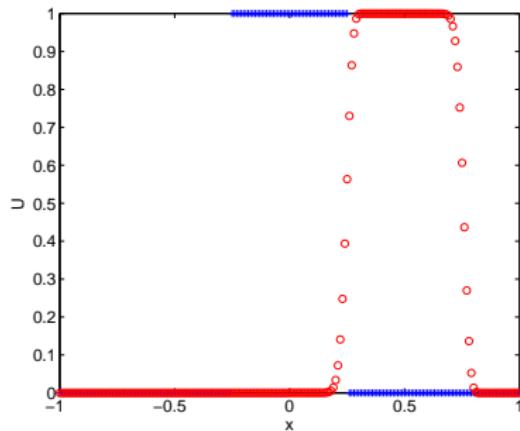
Die Upwind-Ortsdiskretisierung liefert bereits für leicht kleinere Zeit- als Ortsschrittweite gute Ergebnisse.



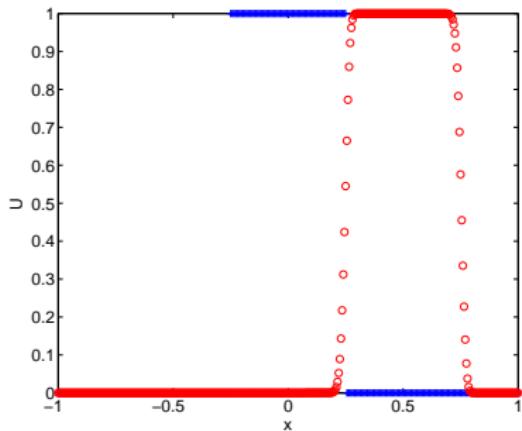
10. PDEs: Finite Differenzen II

Beispiel: Upwind-Diskretisierung für Beispiel 10.18 (nichtglatte Daten)

$$h_x = 0.01 \text{ und } h_t = 0.9 h_x$$



$$h_x = 0.005 \text{ und } h_t = 0.9 h_x$$



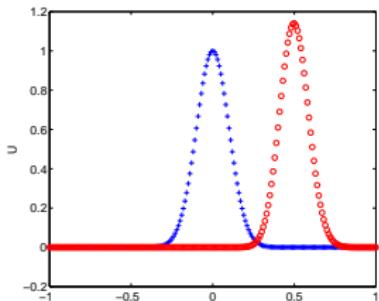
Die Explosionen sind verschwunden, stattdessen sehen wir bei zu grober
Ortsschrittweite eine „Verschmierung“ des unstetigen Übergangs.



10. PDEs: Finite Differenzen II

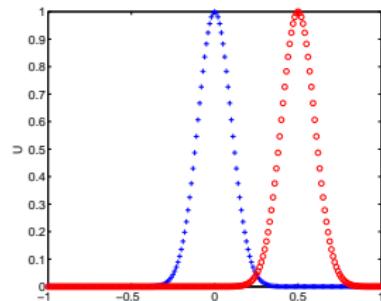
Beispiel: Einfluss der Zeitschrittweite, Upwind-Diskretisierung für Bsp. 10.17

$$h_x = 0.01 \text{ und } h_t = 1.5 h_x$$

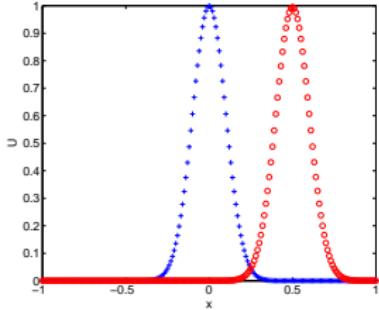


$$h_x = 0.01 \text{ und } h_t = 1.0 h_x$$

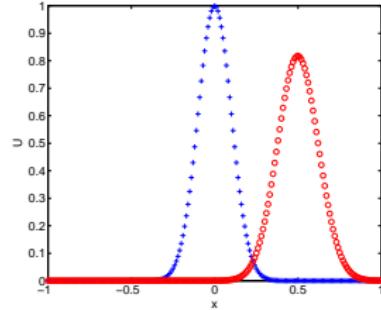
$$h_x = 0.01 \text{ und } h_t = 1.0 h_x$$



$$h_x = 0.01 \text{ und } h_t = 0.99 h_x$$



$$h_x = 0.01 \text{ und } h_t = 0.5 h_x$$

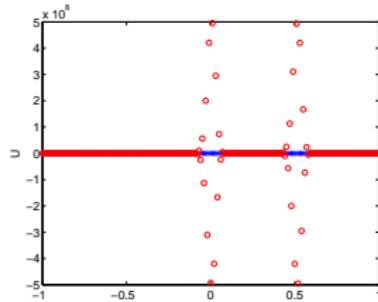


Für zu kleine oder große Zeitschrittweite sind die Ergebnisse unphysikalisch.

10. PDEs: Finite Differenzen II

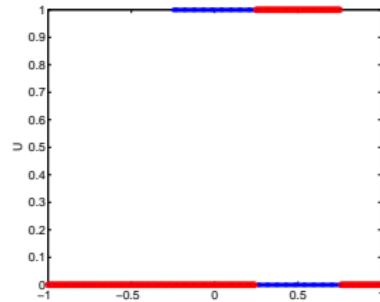
Beispiel: Einfluss der Zeitschrittweite, Upwind-Diskr. 10.19 für Bsp. 10.18

$$h_x = 0.01 \text{ und } h_t = 1.5 h_x$$

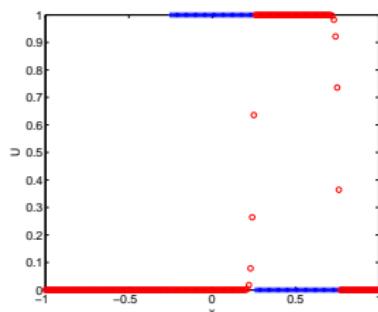


$$h_x = 0.01 \text{ und } h_t = 1.0 h_x$$

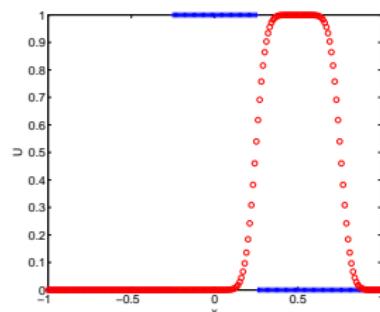
$$h_x = 0.01 \text{ und } h_t = 1.0 h_x$$



$$h_x = 0.01 \text{ und } h_t = 0.99 h_x$$



$$h_x = 0.01 \text{ und } h_t = 0.5 h_x$$



Bei zu großer Zeitschrittweite oszilliert die Lösung („Überspringen“ des kurzen u_0 -Intervalls), bei zu kleiner Zeitschrittweite ist die Lösung verschmiert.



10. PDEs: Finite Differenzen II

Definition 10.20 (Stabilität)

Ein numerisches Verfahren zur Lösung von Problem 10.13 heißt stabil, falls für alle $m \geq 0$ gilt

$$\sup_{n \in \mathbb{Z}} |U_n^{(m)}| \leq \sup_{n \in \mathbb{Z}} |U_n^{(0)}| \leq \sup_{x \in \mathbb{R}} |u_0(x)|.$$

Diese Definition ist das diskrete Analogon zu Satz 10.15.

Satz 10.21 (CFL Bedingung)

Gilt die sogenannte **Courant-Friedrichs-Levy (CFL) Bedingung**

$$h_t \leq \frac{h_x}{\|v\|},$$

so ist die Upwind-Diskretisierung aus Verfahren 10.19 stabil.



Zusammenfassung

- Bei der Linienmethode wird zuerst im Ort und dann in der Zeit diskretisiert. Das nur noch von der Zeit abhängige semidiscrete Problem ist ein System gewöhnlicher AWP, das mit den Methoden aus VL 8 gelöst werden kann.
- Zentrale Differenzen sind für Diffusionsprobleme (2. Ortsableitung) geeignet, jedoch nicht für Advektionsprobleme (erste Ortsableitung), hier ist „Upwinding“ ein sinnvollerer Ansatz.
- Explizite Zeitschrittverfahren für parabolische Probleme erfordern eine quadratische Abhängigkeit der Zeitschrittweite von der Ortsschrittweite, beziehungsweise die CFL-Bedingung im hyperbolischen Fall. Das macht sie in der Praxis faktisch unverwendbar. Implizite Verfahren sind stabil ohne Restriktionen an die Zeitschrittweite.
- Auch wenn wir nur Modellprobleme betrachtet haben, sind die Resultate aus VL 9+10 prototypisch für parabolische und hyperbolische PDEs, bspw. in höheren Ortsdimensionen oder bei gemischten Problemen wie der kombinierten Diffusions-Advektionsgleichung.



Hausaufgaben

- Programmierübung bearbeiten und versuchen, die Plots aus der VL zu reproduzieren und mit den Stabilitätsbedingungen zu erklären.
- Wiederholung und Vorbereitung für die nächsten beiden Wochen:
 - ▶ Partielle Integration
 - ▶ Satz über parameterabhängige Integrale (Vertauschung Differentiation und Integration)
 - ▶ Poisson-Problem von letzter Woche
 - ▶ Lagrange-Quadratur aus VL 6
 - ▶ Quadraturformeln aus VL 7
- Diese Hinweise zur Wiederholung sind natürlich auch für die Klausurvorbereitung nicht grundverkehrt.



Beispieldaufgaben



10. PDEs: Finite Differenzen II

Zeitabhängige Diffusion-Advektion

Wir betrachten das ARWP, vgl. die letzten Beispielaufgaben:

$$\partial_t u(x, t) = \partial_{xx} u(x, t) - q(x) \partial_x u(x, t) - r(x) u(x, t) \quad x \in]0, 1[$$

$$u(0, t) = u(1, t) = 0$$

$$u(x, 0) = u_0$$

Formulieren Sie, ausgehend von der Ortsdiskretisierung aus der letzten Woche, für dieses Problem das explizite und das implizite Euler-Verfahren.



10. PDEs: Finite Differenzen II

Lösungshinweise: Die Anwendung der Linienmethode bedeutet, dass die Ortsdiskretisierung fast 1:1 übernommen werden kann. Die Diskretisierung in der Zeit erfolgt dann mit den Mitteln dieser Vorlesung. Beachten Sie die Vorzeichen.

Ergebnis: Das semidiskrete Problem lautet mit $U_n(0) = u_0(x_n)$ und $U_0(t) = U_{N+1}(t) = 0$:

$$\dot{U}(t) = \left(1 + \frac{q_n h_x}{2}\right) U_{n-1}(t) - (2 + r_n h_x^2) U_n(t) + \left(1 - \frac{q_n h_x}{2}\right) U_{n+1}$$

In Matrixform erhalten wir: $\dot{\mathbf{U}}(t) = \mathbf{A}\mathbf{U}$ mit
 $\mathbf{A} = \text{tridiag}\left(1 + \frac{q_n h_x}{2}, -(2 + r_n h_x^2), 1 - \frac{q_n h_x}{2}\right)$, und somit die Euler-Verfahren:

$$\begin{aligned}\mathbf{U}^{(m)} &= \mathbf{U}^{(m-1)} + h_t \mathbf{A} \mathbf{U}^{(m-1)} \\ \mathbf{U}^{(m)} &= \mathbf{U}^{(m-1)} + h_t \mathbf{A} \mathbf{U}^{(m)} \quad \Leftrightarrow \quad \mathbf{U}^{(m)} = (\mathbf{I} - h_t \mathbf{A})^{-1} \mathbf{U}^{(m-1)}\end{aligned}$$

Achtung: Bei zu großem q dominiert die Advektion, und dieser Term sollte mit Upwind diskretisiert werden, vgl. die ViPLab-Übung.



Upwind-Diskretisierung

Formulieren Sie die Upwind-Diskretisierung für den Fall $v < 0$.



10. PDEs: Finite Differenzen II

Lösungshinweise: Verwenden Sie die Vorwärtsdifferenz $\frac{1}{h_x}(U_{n+1}^{(m-1)} - U_n^{(m-1)})$, um die Information im Aufwind von $U_{n+1}^{(m-1)}$ nach $U_n^{(m)}$ zu transportieren.

Ergebnis: Setze $U_n^{(0)} = u_0(x_n)$ für $n \in \mathbb{Z}$, und berechne für $m \geq 1$ iterativ

$$U_n^{(m)} := U_n^{(m-1)} - v(x_n, t_{m-1}) \frac{h_t}{h_x} (U_{n+1}^{(m-1)} - U_n^{(m-1)}) \quad n \in \mathbb{Z}.$$



11. PDEs: Finite Elemente I



John H. Argyris

19.08.1913 — 02.04.2004

Professor am Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, 1959—1993

Technische Hochschule/Universität Stuttgart



Einleitung



11. PDEs: Finite Elemente I

In VL 9 und 10 haben wir die Finite Differenzen Methode zur Lösung von Anfangs-Randwertproblemen (ARWP) und Randwertproblemen (RWP) für partielle Differentialgleichungen kennengelernt.

Es stellt sich nun die Frage, warum wir in VL 11 und 12 noch eine andere Lösungsmethode behandeln. Prinzipiell sind ja alle (A)RWP mit Finiten Differenzen diskretisierbar.

Die kurze (provokante) Antwort lautet: Finite Differenzen sind zu einfach! Finite Elemente Methoden können viel leistungsfähiger sein.

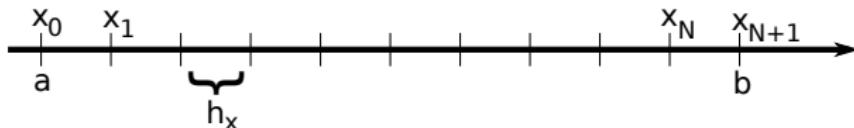
Wir versuchen dies nun mit fünf einfachen Argumenten zu begründen.



11. PDEs: Finite Elemente I

Argument #1

Finite Differenzen liefern eine **punktweise** Approximation der unbekannten Lösung.



Die Approximation der Lösung u eines (A)RWP erfolgt nur in den Gitterpunkten: Wir erhalten nach der Ersetzung aller vorkommenden Ableitungen durch geeignete Differenzenquotienten und nach der Lösung eines (dünn besetzten) linearen Gleichungssystems eine Approximation als **Punktwolke** $\{u_0, \dots, u_{N+1}\}$:

$$u(x_n) \approx u_n \quad \text{für} \quad n = 0, \dots, N + 1$$

Zwischenpunkte erhalten wir nur nach weiterer Verfeinerung.



11. PDEs: Finite Elemente I

Finite Elemente (FE) Methoden approximieren hingegen die unbekannte Lösung u durch eine **(mindestens) stetige Funktion u_h** :

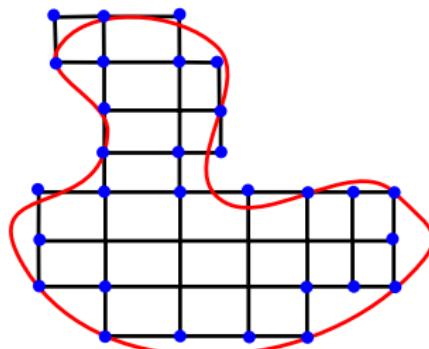
$$u_h \in : [a, b] \rightarrow \mathbb{R} \quad \text{bzw.} \quad u_h \in C^0([a, b], \mathbb{R})$$

Die Funktion u_h ist dabei gegeben in einer geschickt gewählten Basis, und ihre Koeffizienten werden ebenfalls durch die Lösung eines (dünn besetzten) LGS bestimmt. Finite Elemente Methoden sind somit nicht per se teurer.



Argument #2

Finite Differenzen Verfahren leiden außer in Spezialfällen immer unter einem **Randapproximationsfehler**.





11. PDEs: Finite Elemente I

Bei komplizierten Geometrien verlieren Finite Differenzen Verfahren ihren Vorteil, weil variierende Gitterweiten nötig sind. Außerdem muss die Oberfläche kompliziert restauriert werden, beispielsweise mit mehrdimensionaler Interpolation. Das scheint unnötig kompliziert, wenn der Ausgangspunkt beispielsweise ein CAD-Modell ist.

Die Basen in der Finite Elemente Methode können so gewählt werden, dass beliebige krummlinige Geometrien möglich sind. Finite Elemente Methoden sind zudem nach Konstruktion für beliebige unstrukturierte Gitter geeignet.



Argument #3

Die mathematische Rechtfertigung Finiter Differenzen erfordert starke Differenzierbarkeitseigenschaften (hohe Regularität) der unbekannten Lösung.

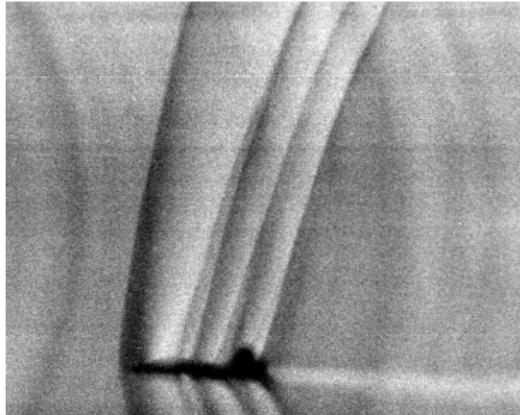
Für das Poisson-Problem $-\Delta u(x) = f(x)$ müssen Lösung $u \in C^2(\Omega)$ sein, damit die Differentialgleichung Sinn ergibt. Für die Rechtfertigung des zentralen DQ haben wir sogar $u \in C^3(\Omega)$ benötigt, um den Konsistenzfehler durch die abgeschnittene Taylorentwicklung quantifizieren zu können.

Beides ist oft unrealistisch in der Praxis: Nur wenige Probleme sind so oft differenzierbar.



11. PDEs: Finite Elemente I

Wir werden sehen, dass für Finite Elemente eine Regularitätsstufe (Differenzierbarkeitsstufe) weniger erforderlich ist für die Rechtfertigung der Methode.



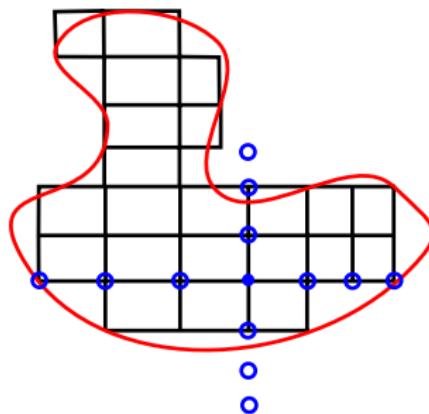
Erste Schlieren-Fotografie eines Überschallflugzeugs, 1993

Tatsächlich müssen wir diese Regularität nur „fast überall“ fordern, Unstetigkeitsstellen sind (mathematisch) erlaubt.



Argument #4

Finite Differenzen Methoden hoher Ordnung sind sehr aufwendig.





11. PDEs: Finite Elemente I

Um die hohe Ordnung nicht zu verlieren, sind viele einseitige Differenzenquotienten nötig. spätestens in 3D wird es beliebig kompliziert, diese Sonderfälle korrekt zu berücksichtigen.

Finite Elemente Diskretisierungen weisen dieses Problem **nicht** auf, und Methoden hoher Ordnung sind viel einfacher zu konstruieren.



Argument #5

Finite Differenzen Verfahren ignorieren Wissen aus der Modellierung.

Oft ist ein Modellierungsergebnis in Form eines Minimierungsproblems oder einer Integralgleichung natürlicher. Finite Elemente Methoden erlauben eine viel stärkere (mathematische) Berücksichtigung von Modellierungstechniken und Modellierungsergebnissen.

Dazu betrachten wir ein Beispiel.



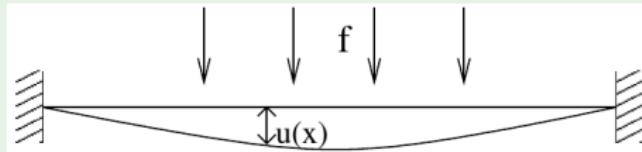
Modellierung mit Energieprinzipien



Beispiel 11.1 (Elastischer Draht in 1D)

Wir betrachten einen am Rand fest eingespannten Draht, der mit der Kraft f von oben belastet wird. Gesucht ist die Auslenkung u des Drahtes.

Wir modellieren den Draht durch ein Gebiet $\Omega =]0, 1[$, und die feste Einspannung durch die Vorgabe der Dirichlet-Randbedingungen $u(0) = u(1) = 0$. Die gesuchte Auslenkung ist eine Funktion $u: \Omega \rightarrow \mathbb{R}$.



Die Modellierung funktioniert für eine Membran $\Omega \subset \mathbb{R}^2$ genauso, oder irgendein Kraftfeld im \mathbb{R}^d wie in Ihren Studienfächern.



11. PDEs: Finite Elemente I

Um ein Modell zu erhalten, betrachten wir die Gesamtenergie des Systems aus Kraft und Draht.

Die elastische Energie E_{el} ist proportional zur Längenänderung $\delta\ell$ des Drahts, die durch die Krafteinwirkung verursacht wird:

$$E_{\text{el}} = \alpha \delta\ell$$

Die Proportionalitätskonstante α (Elastizität des Materials) setzen wir vereinfachend auf $\alpha = 1$. Dann gilt:

$$\delta\ell = \int_0^1 \sqrt{1 + (\partial_x u(x))^2} dx - \int_0^1 1 dx$$

Hierbei ist der erste Summand die Länge des Drahtes nach der Verformung (vgl. Kurvenintegrale aus HM3), und der zweite die Länge vor der Verformung.



11. PDEs: Finite Elemente I

$$\delta\ell = \int_0^1 \sqrt{1 + (\partial_x u(x))^2} dx - \int_0^1 1 dx$$

Mit der Annahme kleiner Auslenkungen (bspw. das Fell einer Trommel nach einem Paukenschlag) können wir das vereinfachen, indem wir die Wurzel weglassen:

$$\delta\ell \approx \int_0^1 1 dx + \int_0^1 (\partial_x u(x))^2 dx - \int_0^1 1 dx$$

Die Vereinfachung müssen wir kompensieren, dazu mitteln wir:

$$E_{\text{el}} = \delta\ell \approx \frac{1}{2} \int_0^1 (\partial_x u(x))^2 dx$$



11. PDEs: Finite Elemente I

Durch die Einwirkung der Kraft (in negative y -Richtung, „nach unten“) besitzt der Draht eine potentielle Energie E_{pot} :

$$E_{\text{pot}} = - \int_0^1 f(x) u(x) dx$$

Diese Gleichung entspricht gerade „Arbeit = Kraft · Weg“.

Weitere (kinetische, gravitationelle, ...) Energien betrachten wir zur Vereinfachung nicht.



11. PDEs: Finite Elemente I

Das Prinzip des Energieminimums besagt nun, dass ein System (wie unser Draht) immer den Zustand niedrigster Energie anstrebt.

Wir können also die stabile Gleichgewichtslage (den stationären Zustand) auch dadurch charakterisieren, dass die Gesamtenergie

$$E(u) := E_{\text{el}} + E_{\text{pot}} = \frac{1}{2} \int_0^1 (\partial_x u(x))^2 \, dx - \int_0^1 f(x)u(x) \, dx$$

minimal wird unter den Nebenbedingungen $u(0) = u(1) = 0$:

Finde $u: \Omega \rightarrow \mathbb{R}$ mit $E(u) \rightarrow \min!$

Damit das Sinn ergibt, muss $u \in C^1(\Omega)$ sein.



Das ist nun (auf den ersten Blick) etwas völlig anderes als das, was wir bisher an Differentialgleichungsmodellen kennengelernt haben.

Wir werden sehen: Dieses **Minimierungsproblem** ist unter gewissen Voraussetzungen **äquivalent** zum Poisson-Problem, in dem Sinne, dass eine \mathcal{C}^2 -Lösung des Minimierungsproblems auch das Poisson-Problem löst, und umgekehrt.

Ohne die \mathcal{C}^2 -Bedingung kann offenbar nur die Rückrichtung gelten. Wir verallgemeinern also den Lösungsbegriff. So kann erreicht werden, dass auch Lösungen mit (wenigen) Unstetigkeitsstellen erlaubt sind.



Lösung des Minimierungsproblems und neue Lösungsbegriffe



11. PDEs: Finite Elemente I

Wir präzisieren nun das Minimierungsproblem, und konstruieren dann zwei äquivalente Beschreibungen, nämlich das Variationsproblem und die schwache Formulierung. Dann erarbeiten wir den Zusammenhang zur bisherigen Formulierung des Poisson-Problems.

Die Äquivalenz (unter gewissen Annahmen) der insgesamt vier Formulierungen ist aus zwei Gründen sehr wichtig. Einerseits dient die schwache Formulierung als Grundlage für die Finite Elemente Methode: Ohne die Äquivalenz dürfen wir die Finite Elemente Methode nicht auf Probleme anwenden, die nicht in schwacher Formulierung vorliegen.



11. PDEs: Finite Elemente I

Andererseits, und deutlich wichtiger aus Anwendungssicht, liefern die Äquivalenzbeweise einen Werkzeugkasten, um die Formulierungen ineinander zu übersetzen. Weil das Ergebnis der Modellierung je nach Problemstellung und Modellierungstechnik in jeder der vier Formulierungen vorliegen kann, ist so ein Werkzeugkasten essentiell für die Praxis Ihrer Studiengänge.

Exemplarisch beschränken wir uns auf das obige konkrete Minimierungsproblem. Die Aussagen und Techniken sind ohne Probleme übertragbar auf andere Fragestellungen.

Wir beginnen mit der Präzisierung des Minimierungsproblems:



11. PDEs: Finite Elemente I

Definition 11.2 (Minimierungsproblem)

Zur Lösung des Minimierungsproblems

$$E(u) := \frac{1}{2} \int_0^1 (\partial_x u(x))^2 \, dx - \int_0^1 f(x)u(x) \, dx \quad \rightarrow \min!$$

suchen wir in einem gegebenen Raum V aller zulässigen Auslenkungen eine Funktion $u \in V$ mit

$$E(u) \leq E(v) \quad \text{für alle } v \in V.$$

Dass diese Formulierung dem Minimierungsproblem entspricht ist klar: Unter allen zulässigen Auslenkungen suchen wie die mit der geringsten Energie.



11. PDEs: Finite Elemente I

Definition 11.2 (Minimierungsproblem)

Zur Lösung des Minimierungsproblems

$$E(u) := \frac{1}{2} \int_0^1 (\partial_x u(x))^2 \, dx - \int_0^1 f(x)u(x) \, dx \quad \rightarrow \min!$$

suchen wir in einem gegebenen Raum V aller zulässigen Auslenkungen eine Funktion $u \in V$ mit

$$E(u) \leq E(v) \quad \text{für alle } v \in V.$$

V heißt auch **Raum der Vergleichsfunktionen (Testfunktionen)**, weil der Minimierer u von E verglichen wird mit allen zulässigen Auslenkungen v . Das ist natürlich eine theoretische und noch keine praxisrelevante Konstruktion.



11. PDEs: Finite Elemente I

Definition 11.2 (Minimierungsproblem)

Zur Lösung des Minimierungsproblems

$$E(u) := \frac{1}{2} \int_0^1 (\partial_x u(x))^2 \, dx - \int_0^1 f(x)u(x) \, dx \quad \rightarrow \min!$$

suchen wir in einem gegebenen Raum V aller zulässigen Auslenkungen eine Funktion $u \in V$ mit

$$E(u) \leq E(v) \quad \text{für alle } v \in V.$$

Damit das Problem sinnvoll gestellt ist, muss V mindestens die Funktionen enthalten, die auf $]0, 1[$ stetige beschränkte Ableitungen besitzen (sonst ist das Integral über $\partial_x u$ nicht definiert), und die Nullrandwerte erfüllen (sonst haben wir keine Lösung des Randwertproblems).



11. PDEs: Finite Elemente I

Ein Beispiel für V ist also der Funktionenraum $C^1(]0, 1[)$ mit der Zusatzforderung der Beschränktheit von Funktion und Ableitung, sowie der Randwerte:

$$V := \left\{ v \in C^1(]0, 1[) \mid v(0) = v(1) = 0, \int_0^1 (\partial_x v)^2 \, dx < \infty, \int_0^1 v \, dx < \infty \right\}$$

Diese Forderung sorgt nebenbei dafür, dass für jede stetige Funktion f das zweite Integral auch Sinn macht.

Bei allgemeineren Problemstellungen sind hier auch andere Funktionenräume erlaubt, bspw. nur stückweise stetig differenzierbare Funktionen, oder integrierbare Funktionen mit höheren Ableitungen, etc.



11. PDEs: Finite Elemente I

Wir schreiben nun das Minimierungsproblem um:

Definition 11.3 (Variationsproblem)

Wir stellen die Vergleichsfunktion $v \in V$ aus dem Minimierungsproblem dar als $v = u + \varepsilon\varphi$ für passende $\varepsilon \in \mathbb{R}$ und $\varphi \in V$. Aufgrund der Minimaleigenschaft gilt dann für die gesuchte Minimallösung (den Minimierer) u :

$$E(u) \leq E(v) = E(u + \varepsilon\varphi) \quad \forall \varepsilon \in \mathbb{R}$$

Der Begriff **Variationsproblem** stammt daher, dass zulässige Auslenkungen v mit dem Parameter ε variiert werden, um so die optimale Minimallösung u zu erhalten. Wir haben das Minimierungsproblem über irgendeinen Funktionenraum also zurückgeführt auf eine Minimierung über \mathbb{R} .

Eine Lösung u des Minimierungsproblems ist offenbar eine Lösung des Variationsproblems, und umgekehrt, wir haben ja nur substituiert.



11. PDEs: Finite Elemente I

Für den nächsten Schritt benötigen wir einige Hilfsmittel aus der HM3, die vermutlich auch aus den Mechanik-Vorlesungen bekannt sein sollten. Zur Motivation schreiben wir das Variationsproblem ausführlich:

$$\begin{aligned} & \frac{1}{2} \int_0^1 (\partial_x u(x))^2 dx - \int_0^1 f(x)u(x) dx \\ \leq & \frac{1}{2} \int_0^1 (\partial_x(u(x) + \varepsilon\varphi(x)))^2 dx - \int_0^1 f(x)(u(x) + \varepsilon\varphi(x)) dx \end{aligned}$$

Wir wollen im Variationsproblem die Lösung explizit bestimmen. Gemäß der notwendigen Bedingung für ein Minimum müssen wir dazu nach ε ableiten (und dann das Ergebnis Null setzen). Wir sehen, dass wir dazu eine verallgemeinerte Kettenregel benötigen.



Satz 11.4 (Materielle Ableitung)

Für die **materielle (substantielle) Ableitung** gilt:

$$\frac{d}{d\varepsilon} \int_0^1 (\partial_x u + \varepsilon \partial_x \varphi)^2 dx = \int_0^1 2(\partial_x u + \varepsilon \partial_x \varphi) \partial_x \varphi dx$$

Wenn wir Differentiation und Integration vertauschen, müssen wir also mit der Kettenregel zur partiellen Ableitung übergehen.

Die materielle Ableitung wird ausführlich in Mechanik-Vorlesungen behandelt.



11. PDEs: Finite Elemente I

Wir wollen nun das Minimum des Variationsproblems

$$E(u) \leq E(v) = E(u + \varepsilon\varphi) \quad \forall \varepsilon \in \mathbb{R}$$

explizit ausrechnen. Die notwendige Bedingung dafür lautet:

$$\frac{d}{d\varepsilon} E(u + \varepsilon\varphi) \Big|_{\varepsilon=0} = 0$$

Wir setzen die Definition von $E(\cdot)$ ein und nutzen die Linearität der Ableitung:

$$\left[\frac{d}{d\varepsilon} \left(\frac{1}{2} \int_0^1 (\partial_x u + \varepsilon \partial_x \varphi)^2 - \int_0^1 f(u + \varepsilon\varphi) dx \right) \right]_{\varepsilon=0} = 0$$

Ausnutzen des letzten Satzes über die materielle Ableitung ergibt:

$$\left[\frac{1}{2} \int_0^1 2(\partial_x u + \varepsilon \partial_x \varphi) \partial_x \varphi dx - \int_0^1 f \varphi dx \right]_{\varepsilon=0} = 0$$

Für das zweite Integral benötigen wir die materielle Ableitung nicht.



11. PDEs: Finite Elemente I

Nun setzen wir im Ausdruck

$$\left[\frac{1}{2} \int_0^1 2(\partial_x u + \varepsilon \partial_x \varphi) \partial_x \varphi \, dx - \int_0^1 f \varphi \, dx \right]_{\varepsilon=0} = 0$$

den Wert $\varepsilon = 0$ ein, damit vereinfacht sich vieles:

$$\int_0^1 \partial_x u \partial_x \varphi \, dx - \int_0^1 f \varphi \, dx = 0$$

Das Ergebnis bringen wir in die übliche Form:

$$\int_0^1 \partial_x u \partial_x \varphi \, dx = \int_0^1 f \varphi \, dx$$

Man beachte, dass alle Schritte Äquivalenzumformungen waren.



11. PDEs: Finite Elemente I

Damit haben wir den nächsten Satz schon zur Hälfte gezeigt, weil φ in der ganzen Rechnung beliebig war.

Satz 11.5 (Schwache Formulierung)

Jede Lösung des Variationsproblems 11.3 ist auch Lösung der schwachen Formulierung

$$\text{finde } u \in V \text{ mit } \int_0^1 \partial_x u \, \partial_x \varphi \, dx = \int_0^1 f \varphi \, dx \quad \text{für alle } \varphi \in V.$$

Beweis: Die notwendige Bedingung haben wir bereits überprüft. Weil das Energiefunktional E konvex ist, d. h.

$$E(\lambda w_1 + (1 - \lambda) w_2) \leq \lambda E(w_1) + (1 - \lambda) E(w_2) \quad \forall \lambda \in]0, 1[; \quad w_1, w_2 \in V,$$

ist die notwendige Bedingung auch hinreichend für ein Minimum. □



11. PDEs: Finite Elemente I

Wir lassen „finde u mit“ meist weg, und verwenden wegen $\forall \varphi$ bzw. $\forall v$ synonym $v \in V$, vgl. Funktionen- und Vektorräume in HM123.

Es gilt auch die Umkehrung:

Satz 11.6 (Schwache Formulierung)

Jede Funktion u , die die schwache Formulierung erfüllt, ist auch eine Lösung des Variationsproblems (der variationellen Formulierung) 11.3 und damit ein Energieminimum von Problem 11.2.

Beweis: Sei u eine Lösung der schwachen Formulierung. Wir wählen ein $v \in V$ und setzen $w = v - u$. Wegen $v = u + w \in V$ (man beachte den Variationszusammenhang) ist dann $w \in V$, also auch zulässig. Nun können wir abschätzen:



11. PDEs: Finite Elemente I

Wir setzen $v = u + w$ in die Definition der Energie ein:

$$E(v) = E(u + w) = \frac{1}{2} \int_0^1 (\partial_x u + \partial_x w) \cdot (\partial_x u + \partial_x w) dx - \int_0^1 f (u + w) dx$$

Ausmultiplizieren unter fleißiger Ausnutzung der Linearität des Integrals ergibt:

$$\begin{aligned} E(v) &= \frac{1}{2} \int_0^1 \partial_x u \partial_x u dx - \int_0^1 f u dx \\ &\quad + \int_0^1 \partial_x u \partial_x w dx - \int_0^1 f w dx + \frac{1}{2} \int_0^1 \partial_x w \partial_x w dx \end{aligned}$$



11. PDEs: Finite Elemente I

$$\begin{aligned} E(v) &= \frac{1}{2} \int_0^1 \partial_x u \, \partial_x u \, dx - \int_0^1 f u \, dx \\ &+ \underbrace{\int_0^1 \partial_x u \, \partial_x w \, dx}_{\text{}} - \underbrace{\int_0^1 f w \, dx}_{\text{}} + \underbrace{\frac{1}{2} \int_0^1 \partial_x w \, \partial_x w \, dx}_{\text{}} \end{aligned}$$

Die erste Klammer verschwindet, weil u nach Voraussetzung die schwache Formulierung erfüllt. Der Integrand der zweiten Klammer ist als quadratische Funktion positiv. Wenn wir den Term weglassen, werden wir insgesamt kleiner:

$$E(v) \geq \frac{1}{2} \int_0^1 \partial_x u \, \partial_x u \, dx - \int_0^1 f u \, dx = E(u)$$

Rücksubstitution der Energie und Umstellen ergibt

$$E(u) \leq E(v) \quad \forall v \in V,$$

d. h. u ist Minimum. Das wollten wir zeigen.





11. PDEs: Finite Elemente I

Insgesamt haben wir gezeigt: Lösungen des Minimierungsproblems sind auch Lösungen des Variationsproblems und umgekehrt. Weiter sind Lösungen des Variationsproblems auch Lösungen der schwachen Formulierung und umgekehrt. Alle drei Formulierungen sind also äquivalent. Das ist fundamental wichtig.

Für den nächsten Satz benötigen wir etwas HM123:

Satz 11.7 (Partielle Integration)

Sei $[a, b]$ ein Intervall, und seien $g, h: [a, b] \rightarrow \mathbb{R}$ stetig differenzierbar. Dann gilt:

$$\begin{aligned}\int_a^b g'(x) \cdot h(x) dx &= [g(x) \cdot h(x)]_a^b - \int_a^b g(x) \cdot h'(x) dx \\ &= g(b) \cdot h(b) - g(a) \cdot h(a) - \int_a^b g(x) \cdot h'(x) dx\end{aligned}$$



11. PDEs: Finite Elemente I

Jetzt können wir zaubern:

Satz 11.8 (Starke Formulierung I)

Sei u Lösung der schwachen Formulierung (des Variationsproblems), und sei **zusätzlich** u zweimal stetig differenzierbar. Dann gilt

$$-\partial_{xx} u(x) = f(x) \quad \text{und} \quad u(0) = u(1) = 0,$$

d. h. u erfüllt die Poisson-Randwertaufgabe.

Das ist jetzt verblüffend, und man sollte einen Moment nachdenken und den Beweis auf der nächsten Folie nachvollziehen. Enorm wichtig ist, dass dies keine Äquivalenzaussage darstellt: Allgemeine „schwache Lösungen“ ohne die C^2 -Zusatzbedingung erfüllen die Voraussetzungen für den Satz nicht!



11. PDEs: Finite Elemente I

Beweis: u ist Lösung des Variationsproblems, d. h.:

$$\int_0^1 \partial_x u \, \partial_x v \, dx - \int_0^1 f v \, dx = 0 \quad \forall v \in V.$$

Wir integrieren den ersten Ausdruck partiell ($g = \partial_x v$, $h = \partial_x u$):

$$\underbrace{[\partial_x u \, v]_0^1}_{\text{Boundary term}} - \int_0^1 \partial_{xx} u \, v \, dx - \int_0^1 f v \, dx = 0$$

Hierfür ist die Zusatzforderung notwendig.



11. PDEs: Finite Elemente I

Ausführlich lautet der unterklammerte Ausdruck:

$$\partial_x u(b)v(b) - \partial_x u(a)v(a)$$

Weil $v \in V$, erfüllt v die Randbedingungen $v(a) = v(b) = 0$. Übrig bleibt:

$$\int_0^1 (-\partial_{xx} u - f) v \, dx = 0 \quad \forall v \in V$$

Das gilt für alle v , insbesondere für alle $v \neq 0$, deshalb muss bereits

$$-\partial_{xx} u - f = 0 \quad \text{auf } (0, 1)$$

erfüllt sein.





11. PDEs: Finite Elemente I

Der Beweis zeigt auch, wie wir eine starke Formulierung in eine schwache Formulierung übersetzen können. Dazu ein Beispiel:

$$-\partial_{xx} u = f \quad \text{in } \Omega =]0, 1[\quad \text{und } u(0) = u(1) = 0$$

Wir wählen zuerst einen geeigneten Testraum V , bspw. den Raum aller quadratintegrierbaren \mathcal{C}^1 -Funktionen auf $]0, 1[$ mit Nullrandbedingungen:

$$V := \left\{ v \in \mathcal{C}^1 (]0, 1[\rightarrow \mathbb{R}) \mid \int_0^1 v^2(x) dx < \infty, v(0) = v(1) = 0 \right\}$$

Wichtig ist dabei, dass wir die Randbedingungen in den Raum V einbauen, anstatt sie explizit zu fordern.



11. PDEs: Finite Elemente I

Im zweiten Schritt multiplizieren wir die PDE mit einer beliebigen **Testfunktion** $v \in V$, und integrieren über das Intervall:

$$-\int_0^1 (\partial_{xx} u(x)) v(x) dx = \int_0^1 f(x)v(x) dx$$

Im dritten Schritt integrieren wir die linke Seite partiell:

$$-\left(\underbrace{[\partial_x u(x) v(x)]_0^1}_{=0} - \int_0^1 \partial_x u(x) \partial_x v(x) dx \right) = \int_0^1 f(x)v(x) dx$$

Der unterklammerte Term verschwindet, weil $v \in V$ die Nullrandbedingungen erfüllt. Übrig bleibt die schwache Formulierung:

$$\int_0^1 \partial_x u(x) \partial_x v(x) dx = \int_0^1 f(x)v(x) dx \quad \forall v \in V$$



11. PDEs: Finite Elemente I

Diese Vorgehensweise ist „algorithmisch“ und für eine gegebene starke Formulierung immer formell durchführbar, wenn der Raum V passt. Erweiterungen für andere Probleme und Gebiete $\Omega \subset \mathbb{R}^d$ diskutieren wir später.

Für andere Randbedingungen (inhomogen Dirichlet, Neumann, ...) fällt bei der partiellen Integration der Randterm nicht heraus. Für die Theorie ist dies eine technische Komplikation, für die Praxis verweisen wir auf die Übungen. In beiden Fällen beinhaltet die schwache Formulierung nicht nur Gebietsintegrale \int_{Ω} , sondern auch Randintegrale $\int_{\partial\Omega}$.

In der schwachen Formulierung wird die linke Seite oft als **Bilinearform** bezeichnet, und die rechte Seite als **Linearform**, da die Abbildung $u, v \mapsto (u, v)_{L^2} := \int_{\Omega} uv \, dx$ gerade ein Skalarprodukt (und damit eine Bilinearform) auf dem Funktionenraum V ist, analog für $v \mapsto \int_{\Omega} fv \, dx$.



11. PDEs: Finite Elemente I

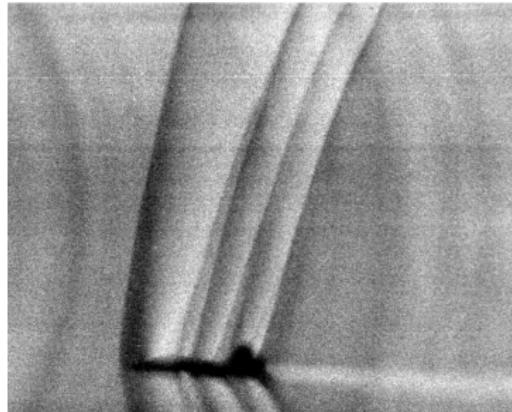
Es ist klar, dass wir für eine starke Lösung (klassische Lösung) $u \in C^2$ fordern müssen, sonst sind die zweiten Ableitungen nicht definiert.

Für die schwache Formulierung

$$\int_0^1 \partial_x u(x) \partial_x v(x) dx = \int_0^1 f(x)v(x) dx \quad \forall v \in V$$

benötigen wir formell nur noch $u, v \in C^1$ (damit der Integrand definiert ist), hinzu kommt die Integrierbarkeit der Produktfunktionen in den Integranden. Dies sind deutlich **schwächere** Voraussetzungen an u als für klassische Lösungen.

Zum besseren Verständnis sollte man sich nochmals klarmachen, dass durch die partielle Integration eine Ableitungsstufe von u auf die **Testfunktion** v verschoben wird.



Erste Schlieren-Fotographie (Dichteverteilung) eines Überschnall-Knalls (Quelle: NASA)

Das neue Konzept schwacher Lösungen hat nicht nur mathematische Vorteile. Ein Überschnall-Knall ist ein reales Phänomen, der durch eine unstetige Dichte verursacht wird. Klassische Lösungen können nicht unstetig sein. Man kann durch sogenannte **Sobolev-Räume** in der schwachen Formulierung Unstetigkeiten zulassen.



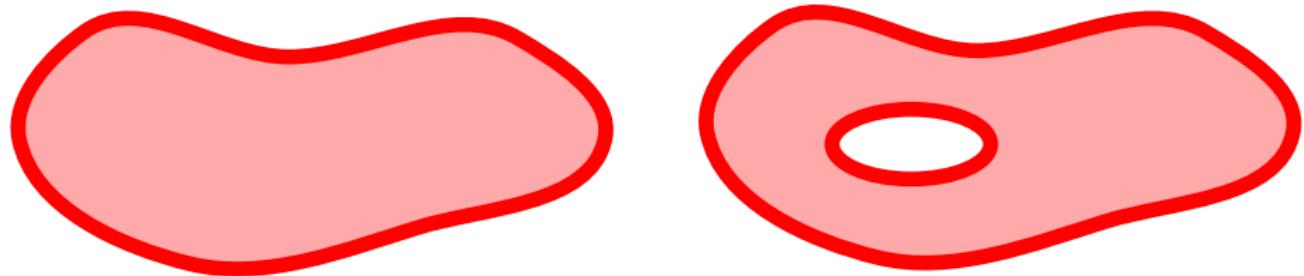
Der allgemeine Fall

11. PDEs: Finite Elemente I



Wir geben der Vollständigkeit halber die allgemeinen Problemformulierungen und Resultate an. Die oben bewiesenen Zusammenhänge gelten entsprechend.

Anstelle von Intervallen werden in \mathbb{R}^d **Gebiete** $\Omega \subset \mathbb{R}^d$ im Ort betrachtet, d. h. Ω ist offen, nichtleer, zusammenhängend und beschränkt, darf aber „Löcher“ enthalten wenn dadurch keine Bedingung verletzt wird.



Die Abbildung zeigt zwei Gebiete in \mathbb{R}^2 .

Quizfrage: Warum ist ein endliches Intervall $(a, b) \subset \mathbb{R}$ das einzige Gebiet in \mathbb{R} ?



11. PDEs: Finite Elemente I

Das anisotrope Diffusionsproblem aus VL 9 lautet in den Formulierungen aus der heutigen Vorlesung:

Starke Formulierung:

$$-\operatorname{div}(A(x)\nabla u(x)) = f(x) \quad (x \in \Omega)$$

Minimierungsformulierung:

$$E(u) = \frac{1}{2} \int_{\Omega} (\nabla u(x))^T A(x) \nabla u(x) - f(x) u(x) dx \quad \rightarrow \quad \min!$$

Schwache Formulierung:

$$\int_{\Omega} (\nabla v(x))^T A(x) \nabla u(x) dx = \int_{\Omega} f(x) v(x) dx \quad \forall v: \Omega \rightarrow \mathbb{R}$$

Für die Äquivalenzen gilt dasselbe wie im Fall von 1D-Poisson.



Zusammenfassung

- Wir haben neben der klassischen Formulierung drei weitere Formulierungen kennengelernt: das Minimierungsproblem, die Variationsformulierung und die schwache Formulierung.
- In der starken Formulierung sind Finite Differenzen direkt anwendbar. In der nächsten Vorlesung sehen wir Finite Elemente Methoden, die auf der schwachen Formulierung basieren.
- Alle Formulierungen lassen sich ineinander umrechnen. Deshalb ist es egal, mit welcher Technik (bspw. Energieminimierung) wir modellieren.
- Die drei neuen Formulierungen sind äquivalent in dem Sinne, dass eine Lösung einer Formulierung auch eine Lösung der anderen Formulierung ist und umgekehrt.
- Die schwache Form stellt eine Abschwächung des Lösungsbegriffs dar: Wir benötigen nur noch \mathcal{C}^1 statt \mathcal{C}^2 wie bei der klassischen Formulierung.



Ausblick

- Man kann dies noch weiter verallgemeinern auf fast überall stetige Differenzierbarkeit (mit einem neuen Differenzierbarkeitsbegriff), d. h. es sind Unstetigkeiten erlaubt. Dies können wir in der Vorlesung nicht behandeln.
- Die Finite Elemente Methode behebt viele der Schwierigkeiten, die wir für die Finite Differenzen Methode aufgezeigt haben. Sie ist zudem viel flexibler und deshalb in den Ingenieurwissenschaften, insb. der Struktur- und Festkörpermechanik, viel weiter verbreitet als die Finite Differenzen Methode.



Hausaufgaben

- Übertragung der neuen Lösungsbegriffe auf andere Probleme
- Wiederholung Quadratur und Interpolation



Beispieldaufgaben



11. PDEs: Finite Elemente I

Neumann-Randbedingungen

Wir betrachten das Randwertproblem

$$-\partial_{xx} u(x) = f(x) \quad \text{in } \Omega =]0, 1[$$

mit den Randbedingungen ($\ell \in \mathbb{R}$)

$$u(0) = 0, \quad \partial_x u(1) = \ell.$$

Geben Sie einen geeigneten Testraum V für die schwache Formulierung an.



11. PDEs: Finite Elemente I

Lösungshinweise: Die Differentialgleichung ist das Poisson-Problem, also wird die einmalige stetige Differenzierbarkeit ausreichen. Es fehlen dann noch die Randbedingungen und die Integrierbarkeit.

Ergebnis:

$$V = \left\{ v \in C^1(]0, 1[) \mid v(0) = 0, \partial_x v(1) = \ell, \int_0^1 v^2(x) dx < \infty \right\}$$



Neumann-Randbedingungen

Wir betrachten nochmals das Randwertproblem

$$-\partial_{xx} u(x) = f(x) \quad \text{in } \Omega =]0, 1[$$

mit den Randbedingungen ($\ell \in \mathbb{R}$)

$$u(0) = 0, \quad \partial_x u(1) = \ell.$$

Wie lautet die schwache Formulierung mit dem Testraum aus der vorherigen Aufgabe?



Lösungshinweise: Multiplikation des Problems mit einer Testfunktion, partielle Integration:

Ergebnis: Finde $u \in V$ mit

$$\int_0^1 \partial_x u(x) \partial_x v(x) dx = \partial_x u(1) v(1) + \int_0^1 f(x) v(x) dx \quad \text{für alle } v \in V.$$



12. PDEs: Finite Elemente II



John H. Argyris

19.08.1913 — 02.04.2004

Professor am Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, 1959—1993

Technische Hochschule/Universität Stuttgart



Einleitung



12. PDEs: Finite Elemente II

In VL 11 haben wir neben der klassischen drei weitere Formulierungen kennengelernt: Unterschiedliche Modellierungstechniken ergeben Minimierungsprobleme, Variationsformulierungen oder schwache Formulierungen.

Alle Formulierungen lassen sich ineinander umrechnen, und sind äquivalent in dem Sinne, dass eine Lösung einer Formulierung auch eine Lösung der anderen Formulierung ist und umgekehrt. Nur für den Schritt von der schwachen zur starken Formulierung benötigen wir dabei eine Zusatzbedingung.

Die schwache Form stellt eine Abschwächung des Lösungsbegriffs dar: Wir benötigen bspw. für das Poisson-Problem nur noch C^1 statt C^2 wie bei der klassischen Formulierung.



12. PDEs: Finite Elemente II

Das Ziel dieser Vorlesung ist, die schwache Formulierung aus der letzten Vorlesung zur Finite Elemente Methode (FEM) auszubauen. Dabei werden wir sehen, dass die FEM viele Vorteile gegenüber Finiten Differenzen aufweist.

Wir beschränken uns dabei zunächst auf das Poisson-Problem in 1D, um die zentralen Ideen zu verdeutlichen.

Neben der Konstruktion der Finite Elemente Methode nutzen wir diese letzte Vorlesung des Semesters, um einigen Stoff früherer Vorlesungen zu wiederholen.



12. PDEs: Finite Elemente II

In der letzten Woche haben wir uns bspw. überlegt, wie die klassische Formulierung

$$-\partial_{xx} u = f \quad \text{in } \Omega =]0, 1[\quad \text{und } u(0) = u(1) = 0$$

des Poisson-Problems in eine schwache Formulierung übersetzt wird.

Wir wählen dazu zuerst einen geeigneten Raum V , bspw. den Raum aller quadratintegrierbaren \mathcal{C}^1 -Funktionen auf $]0, 1[$ mit Nullrandbedingungen:

$$V := \left\{ v \in \mathcal{C}^1 (]0, 1[\rightarrow \mathbb{R}) \mid \int_0^1 v^2(x) dx < \infty, v(0) = v(1) = 0 \right\}$$

Wichtig ist dabei, dass wir die Randbedingungen in den Raum V einbauen, anstatt sie explizit zu fordern.



12. PDEs: Finite Elemente II

Im zweiten Schritt multiplizieren wir die PDE mit einer beliebigen **Testfunktion** $v \in V$, und integrieren über das Intervall:

$$-\int_0^1 (\partial_{xx} u(x)) v(x) dx = \int_0^1 f(x)v(x) dx$$

Im dritten Schritt integrieren wir die linke Seite partiell:

$$-\left(\underbrace{[\partial_x u(x) v(x)]_0^1}_{\text{unterklammerte Term}} - \int_0^1 \partial_x u(x) \partial_x v(x) dx \right) = \int_0^1 f(x)v(x) dx$$

Der unterklammerte Term verschwindet, weil $v \in V$ die Nullrandbedingungen erfüllt:

$$\int_0^1 \partial_x u(x) \partial_x v(x) dx = \int_0^1 f(x)v(x) dx \quad \text{für alle } v \in V$$



12. PDEs: Finite Elemente II

Wenn wir die Lösung nun auch im Raum V suchen, erhalten wir die schwache Formulierung:

$$\text{finde } u \in V \text{ so dass } \int_0^1 \partial_x u(x) \partial_x v(x) dx = \int_0^1 f(x)v(x) dx \text{ für alle } v \in V$$

Das führt auf Galerkin-FEM, weil u und v aus demselben Raum stammen

In der letzten Vorlesung haben wir bewiesen, dass jede klassische Lösung $u \in \mathcal{C}^2$ auch eine Lösung der schwachen Formulierung ist. Eine Lösung der schwachen Formulierung muss wegen der Definition des Raums V nur \mathcal{C}^1 sein, ist sie sogar \mathcal{C}^2 , löst sie auch die klassische Formulierung.

Wir haben also durch die Zusatzforderung der Quadratintegrierbarkeit eine Differenzierbarkeitsstufe eingespart.



12. PDEs: Finite Elemente II

In den Übungen haben wir gesehen, dass (in)homogene Neumann-Randbedingungen nicht in den Raum V eingebaut werden, sondern zu zusätzlichen (Integral-) Termen in der schwachen Formulierung führen.

Für inhomogene Dirichlet-Randbedingungen, also bspw.

$$-\partial_{xx} u = f \text{ für } x \in]0, 1[\quad \text{und} \quad u = g := \begin{cases} 42 & x = 0 \\ 23 & x = 1 \end{cases}$$

konstruieren wir eine \mathcal{C}^2 -Funktion \bar{g} mit $\bar{g} = g$ für $x = 0$ und $x = 1$. Wir befördern die punktweise Funktion g also zu einer \mathcal{C}^2 -Funktion auf dem ganzen Intervall. Hierzu können wir beispielsweise die Polynominterpolation aus VL 6 verwenden.



12. PDEs: Finite Elemente II

Wenn u eine Lösung des Problems ist, gilt für $\bar{u} := u - \bar{g}$:

$$\begin{aligned}-\partial_{xx}\bar{u} &= -\partial_{xx}(u - \bar{g}) = \underbrace{-\partial_{xx}u}_{=f} + \partial_{xx}\bar{g} && \text{für } x \in]0, 1[\\ \bar{u} &= u - \bar{g} = g - \bar{g} && \text{für } x = 0 \text{ und } x = 1\end{aligned}$$

Also ist u genau dann Lösung des Ausgangsproblems, wenn \bar{u} das Problem

$$-\partial_{xx}\bar{u} = f + \partial_{xx}\bar{g}$$

mit **homogenen** Dirichlet-Randwerten löst.



12. PDEs: Finite Elemente II

Wir müssen also das Problem

$$-\partial_{xx}\bar{u} = f + \partial_{xx}\bar{g} \text{ für } x \in]0, 1[\quad \text{und} \quad \bar{u}(0) = \bar{u}(1) = 0$$

so wie bisher in eine schwache Formulierung überführen, und erhalten mit

$$u = \bar{u} + \bar{g}$$

eine schwache Lösung des Problems mit inhomogenen Randwerten.

Die Überführung in eine schwache Formulierung ist eine tolle Wiederholung.

Inhomogene Dirichlet-Randwerte benötigen also denselben Raum V wie homogene Randwerte, und ein wenig Nachbearbeitung der Lösung eines homogenen Hilfsproblems.

Gemischte Randbedingungen wurden in den Übungen betrachtet.



Provisorische Konstruktion der Finite Elemente Methode



12. PDEs: Finite Elemente II

Wir gehen davon aus, dass das zu lösende Problem in einer schwachen Formulierung vorliegt, und das insbesondere der **Ansatz- und Testraum V** homogene Randbedingungen enthält. Eventuelle zusätzliche Neumann-Terme betrachten wir der besseren Übersicht halber nicht. Sie können genauso behandelt werden wie die beiden anderen Integrale in der schwachen Formulierung.

Als durchgehendes Beispiel betrachten wir das schwache Poisson-Problem

$$\text{finde } u \in V \text{ so dass } \int_0^1 \partial_x u(x) \partial_x v(x) dx = \int_0^1 f(x)v(x) dx \text{ für alle } v \in V$$

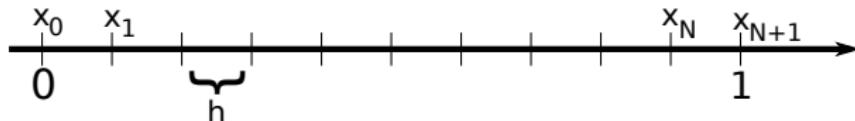
mit dem Raum

$$V := \left\{ v \in C^1 (]0, 1[\rightarrow \mathbb{R}) \mid \int_0^1 v^2(x) dx < \infty, v(0) = v(1) = 0 \right\}.$$

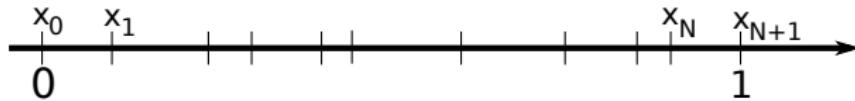


12. PDEs: Finite Elemente II

Wie bei der Finite Differenzen Methode benötigen wir ein **Gitter**:



Im Gegensatz zur Finite Differenzen Methode muss das Gitter nicht äquidistant sein:

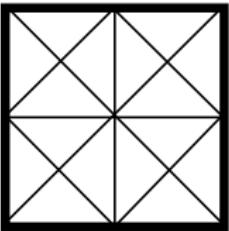
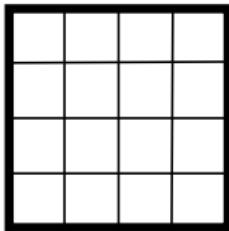
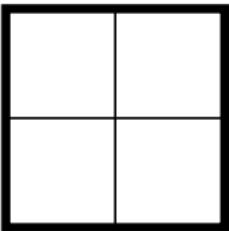


In der FEM-Literatur wird auch oft das Synonym **Triangulierung** verwendet.



12. PDEs: Finite Elemente II

In 2D und 3D besteht das Gitter grundsätzlich aus geometrisch einfachen Objekten wie Dreiecken, Vierecken, Tetraedern, Hexaedern, Prismen, Pyramiden etc.

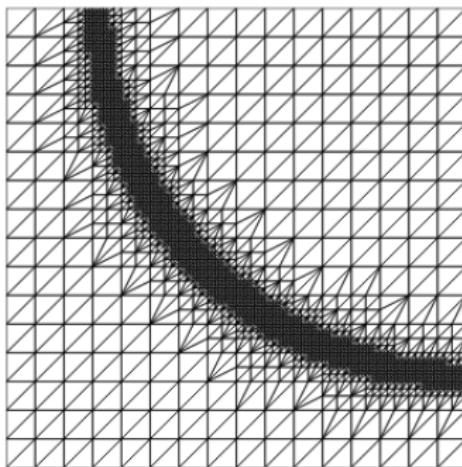


Äquidistante Triangulierungen des Einheitsquadrats mit Dreiecken und Vierecken

12. PDEs: Finite Elemente II



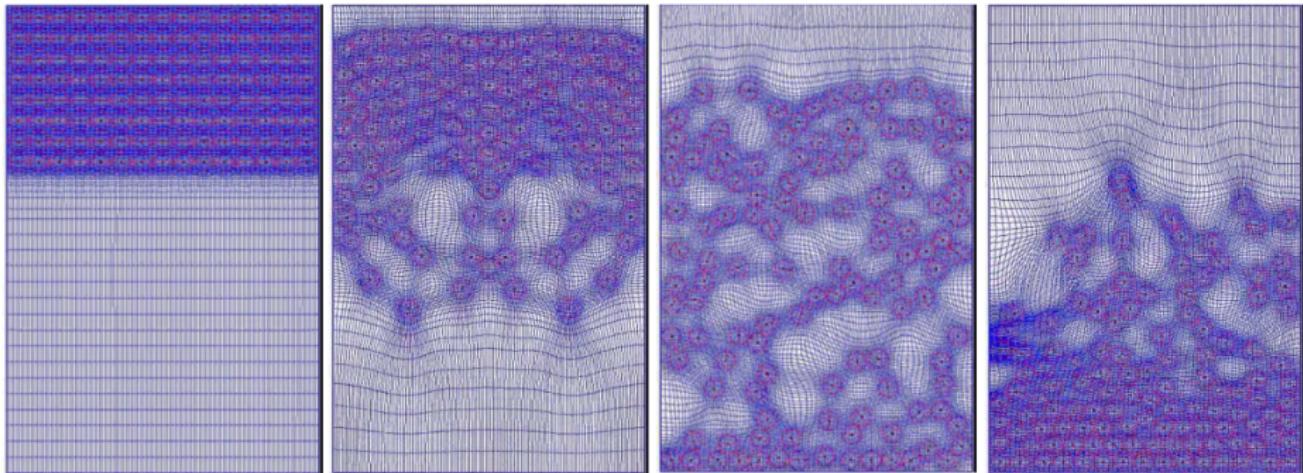
Die Gitter können beliebig an das zugrundeliegende Problem angepasst werden, hier zur Auflösung einer Schockfront:



12. PDEs: Finite Elemente II



Bei zeitabhängigen Problemen können sich Gitter ändern, hier bei einem Sedimentierungsprozess:





12. PDEs: Finite Elemente II

Wir beschränken uns ab jetzt auf den 1D-Fall, und ein nicht notwendigerweise äquidistantes Gitter bestehend aus den Punkten $\{x_0, \dots, x_{N+1}\}$.

Bei Finiten Differenzen haben wir die Lösung u **punktweise** in den Gitterpunkten approximiert:

$$u_n \approx u(x_n) \quad \text{für alle Gitterpunkte}$$

Bei Finiten Elementen wollen wir keine Approximation als Punktmenge, sondern als eine (mindestens stetige) Funktion u^h :

$$u^h:]0, 1[\rightarrow \mathbb{R} \quad \text{mit } u^h \in \mathcal{C}^0(]0, 1[)$$



12. PDEs: Finite Elemente II

Das klingt zunächst wie ein Interpolationsproblem, vgl. VL 6. Wenn wir Lösungsapproximationen u_n in den Gitterpunkten x_n haben, können wir die Funktion u^h darstellen als

$$u^h(x) = \sum_{n=0}^{N+1} u_n \phi_n(x)$$

mit Basisfunktionen ϕ_n , bei der Polynominterpolation bspw. mit der Monombasis von \mathcal{P}_{N+1} .

So können wir jede Finite Differenzen Approximation zu einer stetigen Lösung des Ausgangsproblems befördern. Diese Idee berücksichtigt aber nicht die schwache Formulierung, in der wir die Lösungsapproximationen u_n nicht kennen.



12. PDEs: Finite Elemente II

Wir überlegen uns, ob wenigstens der Ansatz $u^h = \sum \gamma_n \phi_n$ als Linearkombination von Basisvektoren bei der schwachen Formulierung

$$\text{finde } u \in V \text{ so dass } \int_0^1 \partial_x u(x) \partial_x v(x) dx = \int_0^1 f(x) v(x) dx \text{ für alle } v \in V$$

zu retten ist. Immerhin sind ja alle Operatoren (Ableitungen und Integrale) linear.

Das funktioniert aber nicht, weil der Raum

$$V := \left\{ v \in C^1 (]0, 1[\rightarrow \mathbb{R}) \mid \int_0^1 v^2(x) dx < \infty, v(0) = v(1) = 0 \right\},$$

in dem wir Lösungen suchen, unendlich-dimensional ist. Der Ansatz wäre also eine unendliche Summe von Linearfaktoren.



12. PDEs: Finite Elemente II

Als letzte Schwierigkeit kommt in der schwachen Formulierung

$$\text{finde } u \in V \text{ so dass } \int_0^1 \partial_x u(x) \partial_x v(x) dx = \int_0^1 f(x)v(x) dx \text{ für alle } v \in V$$

hinzu: Sie muss **für alle** Testfunktionen gelten. Das ist genau so impraktikabel wie die unendliche LiKo für die Lösungsapproximation u^h .

Alle Probleme bekommen wir aber auf einen Schlag in den Griff:

Diskretisiere nicht nur das Ortsgebiet (bspw. $\Omega =]0, 1[$), sondern **dazu passend auch den Raum** V , in dem Lösungen gesucht werden und mit dem getestet wird.



Konstruktion der Finite Elemente Methode



12. PDEs: Finite Elemente II

Um alle sorgfältig in V untergebrachte Eigenschaften (\mathcal{C}^1 , Quadratintegrierbarkeit, Randbedingungen) zu erhalten, diskretisieren wir V , indem wir zu einem **endlichdimensionalen Teilraum V^h** übergehen:

$$V^h \subset V \quad \text{mit } \dim(V^h) \leq N < \infty$$

Die Dimension des Raums ist also mindestens gleich der Anzahl der inneren Gitterpunkte. Wir beschränken uns zunächst auf $\dim(V^h) = N$.

Für den Raum V^h existiert eine Basis aus linear unabhängigen Basisfunktionen $\{v_1, \dots, v_N\}$, und wir können die gesuchte **diskrete Lösung u^h** als LiKo in der Basis darstellen:

$$u^h = \sum_{n=1}^N \gamma_n v_n \quad \text{mit eindeutigen Koeffizienten } \gamma_n \in \mathbb{R}$$

Wenn wir die Koeffizienten γ_n bestimmen können, haben wir die Lösung.



12. PDEs: Finite Elemente II

Wenn wir die unbekannte Lösung u^h in die schwache Formulierung einsetzen, erhalten wir die **diskrete schwache Formulierung**: *Finde $u^h \in V^h$ mit*

$$\int_a^b (\partial_x u^h)(x) (\partial_x v^h)(x) dx = \int_a^b f(x) v^h(x) dx \quad \text{für alle } v^h \in V^h.$$

Hier sehen wir, warum die Teilraumbeziehung $V^h \subset V$ so wichtig ist: Weil V_h immer noch die C^1 -Forderung enthält, existieren $\partial_x u^h$ und $\partial_x v^h$, wegen der Quadratintegrabilität existieren die Integrale, und weil V^h die Randbedingungen enthält, können wir noch von einer Lösung des RWP sprechen.

Wirklich diskret sieht das aber noch nicht aus.



12. PDEs: Finite Elemente II

Wir erinnern uns an eine wichtige Eigenschaft von Basisfunktionen aus der HM12:
Weil in der diskreten schwachen Formulierung

$$\int_a^b (\partial_x u^h)(x) (\partial_x v^h)(x) dx = \int_a^b f(x) v^h(x) dx \quad \text{für alle } v^h \in V^h$$

alle Operatoren (Ableitungen und Integrale) linear sind, ist diese Bedingung für alle $v^h \in V^h$ erfüllt, wenn sie für alle Basisfunktionen v_1, \dots, v_N erfüllt ist. Um dies einzusehen, schreiben wir v^h in der Basis:

$$v^h = \sum_{m=1}^N \alpha_m v_m \quad \text{mit eindeutigen Koeffizienten } \alpha_m \in \mathbb{R}$$



12. PDEs: Finite Elemente II

Wir setzen diese Darstellung

$$v^h = \sum_{m=1}^N \alpha_m v_m \quad \text{mit eindeutigen Koeffizienten } \alpha_m \in \mathbb{R}$$

in die schwache Formulierung ein, und erhalten mit der Linearität:

$$\begin{aligned} & \int_a^b (\partial_x u^h) (\partial_x v^h) dx = \int_a^b f v^h dx \text{ für alle } v^h \in V^h \\ \Leftrightarrow & \int_a^b (\partial_x u^h) \left(\partial_x \left(\sum_{m=1}^N \alpha_m v_m \right) \right) dx = \int_a^b f \left(\sum_{m=1}^N \alpha_m v_m \right) dx \quad \forall v_m \in \{v_1, \dots, v_N\} \\ \Leftrightarrow & \sum_{m=1}^N \alpha_m \left(\int_a^b (\partial_x u^h) (\partial_x v_m) dx \right) = \sum_{m=1}^N \alpha_m \left(\int_a^b f v_m dx \right) \quad \forall v_m \in \{v_1, \dots, v_N\} \end{aligned}$$

Es reicht also, die Gültigkeit der diskreten schwachen Formulierung für alle Basisfunktionen von V^h zu fordern.



12. PDEs: Finite Elemente II

Weil das so gut funktioniert hat, setzen wir nun die LiKo der unbekannten Lösung

$$u^h = \sum_{n=1}^N \gamma_n v_n \quad \text{mit eindeutigen Koeffizienten } \gamma_n \in \mathbb{R}$$

in die schwache Formulierung ein, für eine beliebige Basisfunktion v_m als Testfunktion:

$$\begin{aligned} & \int_a^b (\partial_x u^h) (\partial_x v_m) dx = \int_a^b f(x) v_m dx \\ \Leftrightarrow & \int_a^b \left(\partial_x \left(\sum_{n=1}^N \gamma_n v_n \right) \right) (\partial_x v_m) dx = \int_a^b f v_m dx \\ \Leftrightarrow & \sum_{n=1}^N \gamma_n \left(\int_a^b (\partial_x v_n) (\partial_x v_m) dx \right) = \int_a^b f v_m dx \end{aligned}$$



12. PDEs: Finite Elemente II

Wir haben also, wenn wir beide Ideen zusammenstöpseln:

$$\sum_{n=1}^N \gamma_n \left(\int_a^b (\partial_x v_n) (\partial_x v_m) dx \right) = \int_a^b f v_m dx \quad \text{für alle } m = 1, \dots, N$$

Lineare Probleme dieser Bauart (links eine Summe über alle Basisfunktionen, rechts ein „für alle“ Basisvektoren) haben wir dieses Semester schon häufiger gesehen, bspw. in VL 6 für die Berechnung der Koeffizienten bei der Interpolation mit Polynomen oder Splines.

Eine solche Darstellung ist insbesondere ein LGS, hier für die γ_n .



12. PDEs: Finite Elemente II

$$\sum_{n=1}^N \gamma_n \left(\int_a^b (\partial_x v_n) (\partial_x v_m) dx \right) = \int_a^b f v_m dx \quad \text{für alle } m = 1, \dots, N$$

Das LGS können wir mit den Setzungen

$$\mathbf{A} = (a_{mn})_{m,n=1,\dots,N}, \quad a_{mn} := \int_a^b \partial_x v_n(x) \partial_x v_m(x) dx$$

$$\mathbf{b} = (b_1, \dots, b_N)^T \in \mathbb{R}^N, \quad b_m := \int_a^b f(x) v_m(x) dx$$

$$\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_N)^T \in \mathbb{R}^N$$

schreiben als $\mathbf{A}\boldsymbol{\gamma} = \mathbf{b}$. So erhalten wir die gesuchten Koeffizienten der Darstellung der Finite Elemente Approximation.



12. PDEs: Finite Elemente II

Wir fassen in Form eines Zwischenfazits zusammen:

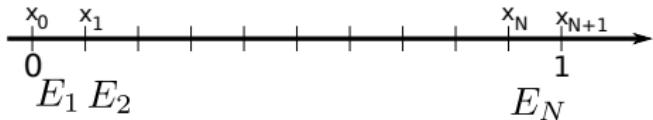
Satz 12.1 (Konstruktionsprinzip einer Finite Elemente Methode)

- ① Wähle einen geeigneten Ansatz- und Testraum V für das gegebene Problem. **Das haben wir für das Poisson-Problem erledigt.**
- ② Wähle einen geeigneten endlichdimensionalen Unterraum $V^h \subset V$ und eine Basis von V^h . **Das fehlt uns noch.**
- ③ Stelle die Matrix \mathbf{A} aus paarweisen Integralen über die (Ableitungen der) Basisfunktionen auf. **Das können wir mit den Methoden aus diesem Semester prinzipiell. (VL 7 → später).**
- ④ Stelle den Vektor \mathbf{b} durch die Integration der Produkte aus f und den Basisfunktionen auf. **Das können wir mit den Methoden aus diesem Semester prinzipiell. (VL 7 → später).**
- ⑤ Löse das LGS $\mathbf{A}\gamma = \mathbf{b}$, um die Koeffizienten der Lösung u^h in der LiKo $u^h = \sum_{n=1}^N \gamma_n v_n$ zu erhalten. **Das können wir mit den Methoden aus diesem Semester sicher (VL 1,2,5)**



12. PDEs: Finite Elemente II

Effiziente Finite Elemente Methoden zeichnen sich dadurch aus, dass die noch offenen Fragen gemeinsam adressiert werden. Hierbei kommt das Gitter wieder ins Spiel:



Im Gegensatz zu Finiten Differenzen sind hier neben den Gitterpunkten auch die Teilintervalle $E_k := [x_{k-1}, x_k]$ von Bedeutung. Sie heißen auch **Elemente**.

Wenn wir es nun schaffen, Basisfunktionen so zu wählen, dass sie nur auf wenigen Elementen von Null verschieden sind, verschwinden wegen der Linearität ziemlich viele Teilintegrale, bspw. für die rechte Seite $\mathbf{b} = (b_1, \dots, b_N)^\top \in \mathbb{R}^N$:

$$b_m = \int_a^b f(x)v_m(x) dx = \int_{E_1} f(x)v_m(x) dx + \dots + \int_{E_N} f(x)v_m(x) dx$$



12. PDEs: Finite Elemente II

Analog wird die Matrix \mathbf{A} automatisch dünn besetzt sein. Dies reduziert den Rechenaufwand in den Schritten 3–5 aus dem letzten Satz, und wir können effiziente iterative LGS-Löser einsetzen anstelle der Gauß-Elimination. Man spricht von **Basisfunktionen mit lokalem Träger**.

Hier haben wir nun ein kleines Problem: Der endlichdimensionale Teilraum V^h soll C^1 -Funktionen beinhalten, damit die schwache Formulierung möglich ist. Wenn wir die Lösung rein lokal zusammensetzen, erhalten wir aber nur Stetigkeit, also C^0 . Das lässt sich mit der **Sobolev-Theorie** reparieren, die (leicht vereinfacht) besagt, dass es reicht, global C^1 nur überall außer in den Gitterpunkten zu fordern.

Das müssen wir glauben, im reinen Mathematikstudium kommt sie frühestens im 4. Semester vor.

Insgesamt haben wir einen allgemeinen Konstruktionsrahmen für Finite Elemente erarbeitet, und können (endlich) Beispiele betrachten.



Beispiel:

Lineare Lagrange-FEM



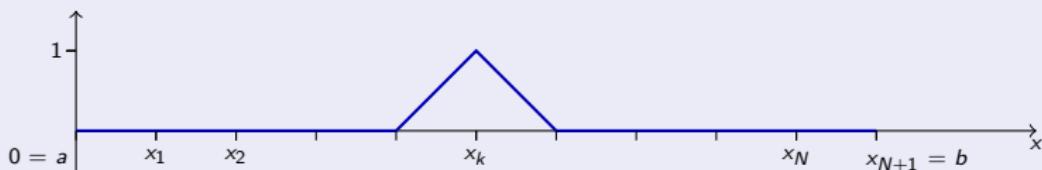
12. PDEs: Finite Elemente II

Bei der **linearen Lagrange-FEM** werden als Basisfunktionen sogenannte Hütchenfunktionen verwendet:

Definition 12.2 (Hütchenfunktionen)

Auf einem Gitter mit den Gitterpunkten x_0, \dots, x_{N+1} ist für $k = 1, \dots, N$ die k -te Hütchenfunktion definiert durch

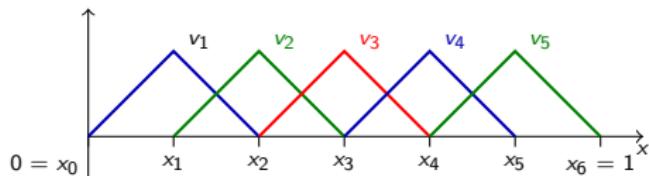
$$v_k(x) = \begin{cases} 0 & x \leq x_{k-1} \\ \frac{x-x_{k-1}}{x_k-x_{k-1}} & x_{k-1} \leq x < x_k \\ \frac{x_{k+1}-x}{x_{k+1}-x_k} & x_k \leq x < x_{k+1} \\ 0 & x \geq x_{k+1} \end{cases}$$





12. PDEs: Finite Elemente II

Wir betrachten das Intervall $\Omega =]0, 1[$ und zur Verdeutlichung ein äquidistantes Gitter aus 7 Punkten:



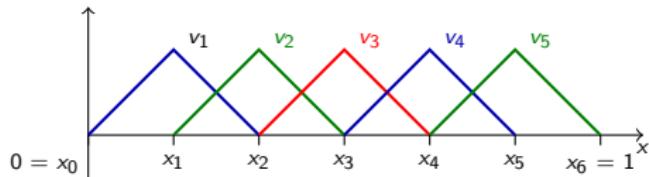
Die Gitterweite ist gerade $h = x_{k+1} - x_k$ für alle $k = 0, \dots, 5$.

Wir sehen: Die Hütchenfunktionen sind stückweise linear auf jedem Teilintervall, und deshalb quadratintegrabel. Sie verschwinden außerhalb der beiden Intervalle $[x_{k-1}, x_k] \cup [x_k, x_{k+1}[$, deshalb müsste die Aufstellung des LGS wenig Aufwand erfordern.

Auch die Randwerte werden eingehalten, wegen $v_1(x_0) = 0$ und $v_N(x_{N+1}) = 0$.



12. PDEs: Finite Elemente II



Scharfes Hinsehen (bzw. eigentlich Nachrechnen) liefert mit dem Kronecker-Delta

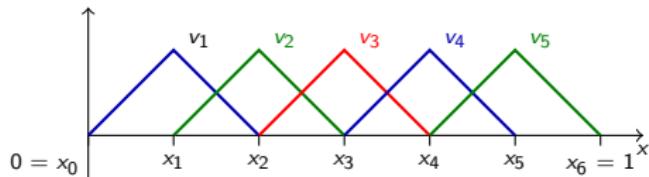
$$v_k(x_n) = \delta_{kn} \quad k, n = 1, \dots, N.$$

LiKos aus den Hütchenfunktionen sind deshalb in den Gitterpunkten eindeutig bestimmt und damit auf dem ganzen Intervall stetig.

Insgesamt haben wir die Basiseigenschaft für den Raum V^h .



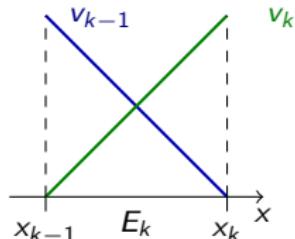
12. PDEs: Finite Elemente II



Übersetzt in die Sprache der FEM sehen wir, dass auf jedem Element (Teilintervall) $E_k = [x_{k-1}, x_k]$ maximal zwei Basisfunktionen von Null verschieden sind, nämlich:

$$v_{k-1}(x) \Big|_{E_k} = \frac{x_k - x}{x_k - x_{k-1}}$$

$$v_k(x) \Big|_{E_k} = \frac{x - x_{k-1}}{x_k - x_{k-1}}$$



Das ist eine Konsequenz des lokalen Trägers von eben.



12. PDEs: Finite Elemente II

Zur Berechnung der Koeffizienten der Lösung

$$u^h = \sum_{n=1}^N \gamma_n v_n$$

müssen wir das LGS $\mathbf{A}\gamma = \mathbf{b}$ lösen. Für $\mathbf{b} = (b_m)_{m=1}^N$ hatten wir

$$b_m = \int_0^1 f(x) v_m(x) dx = \sum_{k=1, \dots, N} \int_{E_k} f(x) v_m(x) dx$$

Dank des lokalen Trägers der Basisfunktionen vereinfacht sich die Summe erheblich:

$$b_m = \int_0^1 f(x) v_m(x) dx = \int_{E_m} f(x) v_m(x) dx + \int_{E_{m+1}} f(x) v_m(x) dx$$



12. PDEs: Finite Elemente II

Genauso vereinfacht sich die Berechnung der Matrixeinträge für $n \neq m$:

$$\begin{aligned} a_{mn} &= \int_0^1 \partial_x v_n(x) \partial_x v_m(x) dx = \sum_{k=1,\dots,N} \int_{E_k} \partial_x v_n(x) \partial_x v_m(x) dx \\ &= \int_{E_m} \partial_x v_n(x) \partial_x v_m(x) dx + \int_{E_{m+1}} \partial_x v_n(x) \partial_x v_m(x) dx \\ &\quad + \int_{E_n} \partial_x v_n(x) \partial_x v_m(x) dx + \int_{E_{n+1}} \partial_x v_n(x) \partial_x v_m(x) dx \end{aligned}$$

Für $n = m$ entfällt die letzte Zeile, und wenn in einem Eintrag mehrfach über ein Intervall integriert wird, wird es nur einmal berechnet.

In der Sprache der Finiten Elemente spricht man auch von der **Assemblierung** der Matrix \mathbf{A} und der rechten Seite \mathbf{b} .



12. PDEs: Finite Elemente II

Wegen des lokalen Trägers sind fast alle Matrixeinträge Null. Für unser 1D-Beispiel reicht es, für $m = 1, \dots, M$ die folgenden Einträge zu berechnen:

$$\begin{aligned} a_{mm} &= \int_{E_m} \partial_x v_m(x) \partial_x v_m(x) dx + \int_{E_{m+1}} \partial_x v_m(x) \partial_x v_m(x) dx \\ a_{m,m+1} &= \int_{E_{m+1}} \partial_x v_{m+1}(x) \partial_x v_m(x) dx \quad \text{für } m < N \\ a_{m,m-1} &= \int_{E_m} \partial_x v_{m-1}(x) \partial_x v_m(x) dx \quad \text{für } m > 1 \end{aligned}$$

In der Praxis wird über alle Elemente iteriert, so dass keine Integrale mehrfach berechnet werden. In diesem Beispiel reichen dann zwei Integrale pro Matrixzeile.



12. PDEs: Finite Elemente II

Um die Element-Integrale zu berechnen, benötigen wir i.A. numerische Quadraturformeln, bspw. für die rechte Seite:

$$\int_{E_k} f(x) v_k(x) dx \approx Q_{[x_{k-1}, x_k]}^M [f v_k],$$
$$\int_{E_k} f(x) v_{k+1}(x) dx \approx Q_{[x_{k-1}, x_k]}^M [f v_{k+1}]$$

Hierbei ist Q eine auf $[x_{k-1}, x_k]$ transformierte Quadratur \hat{Q}^M , vgl. VL 7.

Damit wir uns keinen zusätzlichen Quadraturfehler einhandeln, sollte M so gewählt sein, dass \hat{Q}^M exakt ist für Funktionen aus V^h , also in unserem Beispiel exakt für lineare Funktionen. Das ist bspw. für die Mittelpunktsregel erfüllt, nicht aber für die Vorwärts- und Rückwärts-Rechtecksregeln.



12. PDEs: Finite Elemente II

Für die Einträge der Matrix \mathbf{A} wird analog vorgegangen, indem Produkte aus Ableitungen von Basisfunktionen numerisch integriert werden.

Für ein äquidistantes Gitter der Schrittweite h erhalten wir:

$$\mathbf{A} = \frac{1}{h} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} = \frac{1}{h} \text{tridiag}(-1, 2, -1)$$

Bis auf eine Skalierung mit h^{-1} ist \mathbf{A} die gleiche Matrix wie im Fall Finiter Differenzen. **Dies gilt nur für das verwendete uniforme Gitter!**

Vergleiche eine der Mini-Übungen am Ende dieses Foliensatzes.

Wegen der Symmetrie und positiven Definitheit ist bspw. das CG-Verfahren (VL 5) zu seiner Lösung anwendbar.



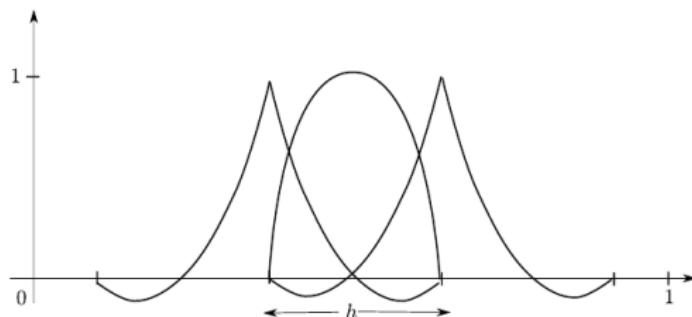
Verallgemeinerungen

und

Erweiterungen



Eine Lagrange Finite Elemente Methode höherer Ordnung als $p = 1$ erhalten wir, wenn wir keine stückweise linearen, sondern stückweise quadratische Basisfunktionen verwenden:

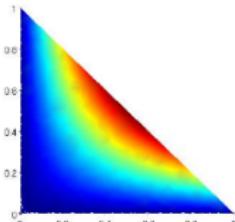
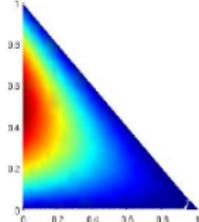
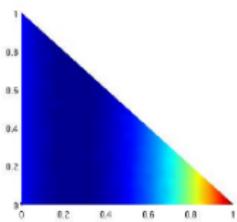
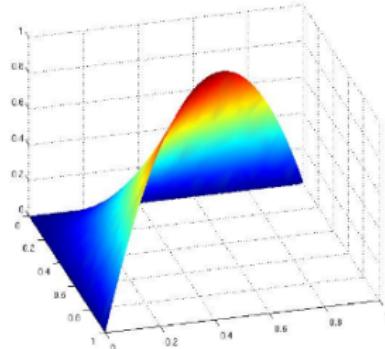
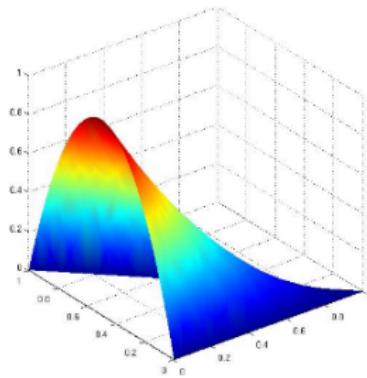
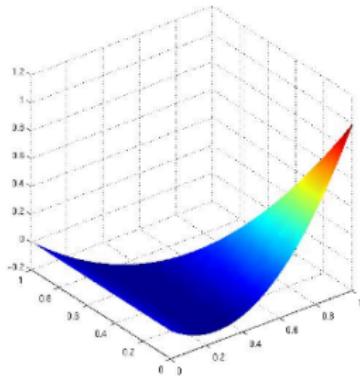


Hier sind nun auf jedem Element drei Basisfunktionen von Null verschieden. Jede Basisfunktion „lebt“ jedoch weiterhin nur auf zwei benachbarten Elementen. Die Konstruktion lässt sich für $p > 2$ fortsetzen.



12. PDEs: Finite Elemente II

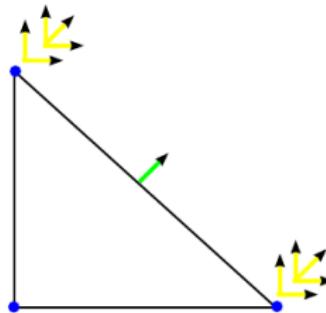
Die Konstruktion funktioniert auch in 2D und 3D. Die Abbildung zeigt die Basisfunktionen für lokal quadratische, global stetige Basisfunktionen in 2D auf Dreiecken.





12. PDEs: Finite Elemente II

Auch C^1 -Übergänge sind möglich, das führt im einfachsten Fall auf das Argyris-Element:



Die Basisfunktionen stellen sicher, dass in den Ecken eines Dreiecks Funktionswerte (blau), sowie alle Ableitungen und zweiten Ableitungen (gelb) übereinstimmen, sowie die Normalenableitung entlang jeder Kante (grün). Das geht, ist aber furchtbar kompliziert. Deshalb hat es auch ein Ingenieur erfunden.



12. PDEs: Finite Elemente II

Beispiel: Fehler gegen exakte Lösung $u(x, y) = \sin(\pi x) \sin(\pi y)$ für das Poisson-Problem bei einer regulären Verfeinerung eines Dreiecksgitters in 2D (jedes Dreieck wird in 4 kongruente Dreiecke zerlegt):

Verf.	$p = 1$		$p = 2$		$p = 3$	
	$\ u^h - u\ _{L^2}$		$\ u^h - u\ _{L^2}$		$\ u^h - u\ _{L^2}$	
2	2.73e-2		1.29e-3		6.53e-5	
3	7.19e-3	3.81	1.64e-4	7.87	4.11e-6	15.90
4	1.83e-3	3.93	2.07e-5	7.93	2.57e-7	15.97
5	4.61e-4	3.97	2.60e-6	7.96	1.61e-8	15.99
6	1.15e-4	3.99	3.26e-7	7.98	1.00e-9	16.00
7	2.88e-5	4.00	4.08e-8	7.99	6.29e-11	15.98

Die L^2 -Norm ist das Analogon der Euklid-Norm für Funktionen

Wir sehen in den jeweils zweiten Spalten die gleichen Fehlerreduktionsraten wie bei Finiten Differenzen. Das kann man beweisen.



Zusammenfassung

- Die Finite Elemente Methode ist zugeschnitten auf schwache Formulierungen. Wegen der theoretischen Überlegungen aus VL 11 ist sie auch anwendbar, wenn die Modellierung bspw. auf ein Minimierungsproblem führt.
- Ihre Anwendung lässt sich auf folgende Schritte herunterbrechen:
 - ➊ Wahl eines Ansatz- und Testraums V und Übergang zur schwachen Formulierung (\rightarrow VL 11+12)
 - ➋ Wahl eines geeigneten endlichdimensionalen Unterraums $V^h \subset V$ (\rightarrow VL 12): Zerlegung des Gebiets in Elemente und passende Basis, bspw. Hütchenfunktionen
 - ➌ Assemblierung der Matrix und der rechten Seite, die der diskreten schwachen Formulierung entsprechen (\rightarrow VL 7+12)
 - ➍ Lösen des LGS liefert Koeffizienten
 - ➎ Zusammensetzen der der FE-Approximation als LiKo in der Basis mit den Koeffizienten (\rightarrow VL 1,2,5)



Zusammenfassung

- Lagrange-FEM liefert Approximationen, die lokal polynomiell und global stetig sind.
- Die einfachste Lagrange-FEM basiert auf stückweise linearen Basisfunktionen.
- Alle zu Beginn von VL 11 postulierten Vorteile der FEM wurden bestätigt.



Hausaufgaben

- Verinnerlichung des Stoffs von VL 11+12
- Wiederholen der heute benötigten früheren VLen, und der anderen Vorlesungen
- Vorbereitung auf die Massensprechstunden (s. ILIAS) auf der Basis der bereitgestellten älteren Klausuraufgaben und der Übungen zu dieser Vorlesung.
- Achtung: Weil die VL umgebaut wurde (vgl. VL 0), decken die alten Klausuraufgaben nur eine Teilmenge des relevanten Stoffs ab.



Beispieldaufgaben



Hütchenfunktionen

Wir betrachten das Randwertproblem

$$-\partial_{xx} u(x) = f(x) \quad \text{in } \Omega =]0, 1[\quad \text{mit } u(0) = u(1) = 0$$

auf dem „Gitter“ aus den drei Punkten $x_0 = 0$ und $x_1 = 0.5$ und $x_2 = 1$. Wie lautet der LiKo-Ansatz für die Finite Elemente Lösung bei Verwendung der Hütchenfunktionen als Basis?



Lösungshinweise: Überlegen Sie sich, welche Basisfunktionen auf diesem Gitter überhaupt aktiv sind.

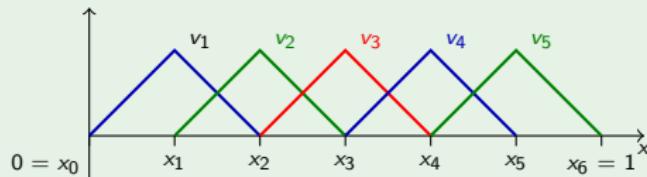
Ergebnis:

$$u^h = \gamma_1 \begin{cases} \frac{x-0}{0.5-0} & 0 \leq x < 0.5 \\ \frac{1-x}{1-0.5} & 0.5 \leq x \leq 1 \end{cases}$$



Assemblierung

Wir betrachten das Poisson-Problem aus der vorherigen Aufgabe, jedoch auf dem Gitter und mit den Basisfunktionen gemäß der folgenden Abbildung:



Assemblieren Sie die Matrix \mathbf{A} des LGS $\mathbf{A}\gamma = \mathbf{b}$ analytisch, indem Sie alle notwendigen Integrale mit Papier und Bleistift statt mit numerischer Quadratur ausrechnen.



12. PDEs: Finite Elemente II

Lösungshinweise: Überlegen Sie sich für die Matrix, wie die Ableitungen der Basisfunktionen aussehen. Nutzen Sie dann den kompakten Träger.

Ergebnis: Das Gitter ist äquidistant, deshalb lauten die nicht verschwindenden Ableitungen der Basisfunktionen auf E_k :

$$\partial_x v_k \equiv -\frac{1}{h} \quad \text{und} \quad \partial_x v_{k+1} \equiv \frac{1}{h}$$

Die Elementintegrale sind in diesem Beispiel:

$$\int_{E_k} \partial_x v_k \partial_x v_k \, dx = h \left(-\frac{1}{h} \right)^2 = \frac{1}{h}$$

$$\int_{E_k} \partial_x v_{k+1} \partial_x v_{k+1} \, dx = h \left(\frac{1}{h} \right)^2 = \frac{1}{h}$$

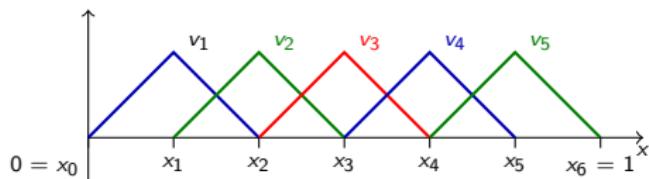
$$\int_{E_k} \partial_x v_k \partial_x v_{k+1} \, dx = h \left(-\frac{1}{h} \right) \frac{1}{h} = -\frac{1}{h}$$

$$\int_{E_k} \partial_x v_{k+1} \partial_x v_k \, dx = h \frac{1}{h} \left(-\frac{1}{h} \right) = -\frac{1}{h}$$



12. PDEs: Finite Elemente II

Insgesamt ergibt sich die Matrix aus der VL.



$$\mathbf{A} = \frac{1}{h} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} = \frac{1}{h} \text{tridiag}(-1, 2, -1)$$