

Numerik

Oliver Sander

2020-04-10

Getippt und gesetzt mit viel Hilfe von Ansgar Burchardt, Katja Hanowski, Patrick Jaap,
Lisa Nebel und Johannes R. Stojanow



Oliver Sander, 2016

Copyright 2016 by Oliver Sander. This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/>.

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Interpolation | 1 |
| 1.1 | Polynominterpolation | 2 |
| 1.1.1 | Lagrange-Interpolation | 2 |
| 1.1.2 | Newton-Form des Interpolationspolynoms | 4 |
| 1.1.3 | Interpolationsfehler | 5 |
| 1.2 | Spline-Interpolation | 10 |
| 1.2.1 | C^1 -Interpolation | 11 |
| 1.2.2 | Kubische C^2 -Splines | 12 |
| 1.2.3 | Interpolationsfehler | 14 |
| 1.2.4 | Geometrische Interpretation | 17 |
| 1.2.5 | B-Splines | 18 |
| 2 | Numerische Quadratur/Integration | 21 |
| 2.1 | Die Newton-Cotes-Formeln | 22 |
| 2.2 | Quadraturfehler der Simpson-Regel | 22 |
| 2.3 | Zusammengesetzte (summierte) Newton-Cotes-Formeln | 25 |
| 2.4 | Gauß-Quadratur | 26 |
| 2.4.1 | Gewichtete Integrale | 26 |
| 2.4.2 | Orthogonale Polynome | 27 |
| 2.4.3 | Quadratur mit Orthogonalpolynomen | 29 |
| 2.5 | Extrapolationsverfahren für Quadratur | 30 |
| 2.5.1 | Romberg-Quadratur | 31 |
| 2.5.2 | Extrapolation mit mehr als zwei Stützstellen | 32 |
| 3 | Lineare Gleichungssysteme | 35 |
| 3.1 | Dreiecksmatrizen | 36 |
| 3.2 | Das Gaußsche Eliminationsverfahren | 37 |
| 3.2.1 | Aufwand | 38 |
| 3.3 | Durchführbarkeit | 38 |
| 3.4 | Die LR-Zerlegung | 40 |
| 3.5 | Pivot-Strategien | 41 |
| 3.5.1 | Probleme mit der einfachen Gauß-Elimination | 41 |
| 3.5.2 | Pivot-Strategien | 42 |
| 3.6 | Nachiteration | 44 |

| | | |
|----------|---|-----------|
| 4 | Kondition und Stabilität | 45 |
| 4.1 | Die Kondition eines Problems | 45 |
| 4.1.1 | Kondition von Addition und Subtraktion | 47 |
| 4.1.2 | Kondition der Multiplikation | 48 |
| 4.1.3 | Kondition von linearen Gleichungssystemen | 49 |
| 4.2 | Stabilität | 51 |
| 4.2.1 | Modifikation von Algorithmen | 52 |
| 4.2.2 | Vorwärtsanalyse der Stabilität | 53 |
| 4.2.3 | Stabilität der Gauß-Elimination | 54 |
| 5 | Numerische Lösung nichtlinearer Gleichungen | 57 |
| 5.1 | Fixpunktiterationen | 57 |
| 5.1.1 | Konvergenzgeschwindigkeit | 59 |
| 5.1.2 | Das Newton-Verfahren als Fixpunktiteration | 60 |
| 5.1.3 | Alternative Interpretation des Newton-Verfahrens | 61 |
| 5.1.4 | Mehrfache Nullstellen | 62 |
| 5.2 | Das Newton-Verfahren für Systeme von Gleichungen | 62 |
| 5.3 | Affin-Invarianz | 65 |
| 5.4 | Konvergenzkriterien | 67 |
| 5.4.1 | Der Monotonietest | 67 |
| 5.4.2 | Der natürliche Monotonietest | 68 |
| 5.5 | Newton-Verfahren mit Dämpfung | 68 |
| 6 | Nichtlineare Ausgleichsprobleme | 75 |
| 6.1 | Prinzip der kleinsten Quadrate | 76 |
| 6.2 | Lineare Ausgleichsprobleme | 76 |
| 6.2.1 | Pseudoinverse | 77 |
| 6.3 | Das Gauß-Newton-Verfahren | 78 |
| 7 | Optimierung | 83 |
| 7.1 | Gradientenartige Verfahren | 84 |
| 7.1.1 | Schrittweiten | 84 |
| 7.1.2 | Suchrichtungen | 87 |
| 7.1.3 | Das Gradientenverfahren | 90 |
| 7.2 | Das Newton-Verfahren | 91 |
| 7.2.1 | Konvergenzeigenschaften des Newton-Verfahrens | 92 |
| 7.3 | Quasi-Newton-Verfahren | 93 |
| 7.4 | Trust-Region-Verfahren | 95 |
| 7.5 | Globale Konvergenz | 96 |
| 7.6 | Das Hundebein-Verfahren | 97 |
| 8 | Iterative Lösungsverfahren für große, dünnbesetzte Gleichungssysteme | 99 |
| 8.1 | Motivation: Das Poisson-Problem | 99 |
| 8.1.1 | Eigenschaften der Matrizen | 101 |

| | | |
|-----------|--|------------|
| 8.2 | Lineare iterative Verfahren | 103 |
| 8.2.1 | Konvergenz | 103 |
| 8.2.2 | Konvergenzgeschwindigkeit | 105 |
| 8.2.3 | Die Wahl von C | 106 |
| 8.2.4 | Das Jacobi-Verfahren | 107 |
| 8.2.5 | Das Gauß-Seidel-Verfahren | 110 |
| 8.2.6 | Abbruchkriterien | 112 |
| 8.3 | Das Gradientenverfahren | 113 |
| 8.3.1 | Idee des Gradientenverfahrens | 113 |
| 8.3.2 | Konvergenzanalyse | 114 |
| 8.4 | Das Verfahren der konjugierten Gradienten (CG) | 115 |
| 8.4.1 | Das Gram-Schmidt-Verfahren | 117 |
| 8.4.2 | Das Verfahren der konjugierten Gradienten | 117 |
| 8.4.3 | Das komplette Verfahren | 119 |
| 8.4.4 | Interpretation als Krylov-Verfahren | 120 |
| 8.4.5 | Konvergenz des CG-Verfahren als iterativem Verfahren | 121 |
| 8.5 | Vorkonditionierung | 124 |
| 8.5.1 | Idee der Vorkonditionierung | 124 |
| 8.5.2 | Unvollständige Cholesky-Zerlegung (ICH,ILU,...) | 126 |
| 8.5.3 | Lineare Verfahren als Vorkonditionierer | 128 |
| 9 | Direkte Lösungsverfahren für dünnbesetzte Gleichungssysteme | 131 |
| 9.1 | Die Multifrontale Methode | 132 |
| 9.1.1 | Cholesky-Zerlegung | 132 |
| 9.1.2 | Die Struktur von L | 134 |
| 9.1.3 | Ausnutzen der Dünnbesetztheit, Teil 1 | 135 |
| 9.1.4 | Graphen- und Baumdarstellung | 136 |
| 9.1.5 | Ausnutzen der Dünnbesetztheit, Teil 2 | 138 |
| 9.1.6 | Matrix-Superposition (Der extend-add Operator) | 139 |
| 9.1.7 | Der endgültige Algorithmus | 141 |
| 9.1.8 | Umsortierungen der Matrix | 141 |
| 10 | Numerik von gewöhnlichen Differentialgleichungen | 145 |
| 10.1 | Anfangswertprobleme | 146 |
| 10.2 | Existenz und Eindeutigkeit | 146 |
| 10.3 | Evolution und Phasenfluss | 150 |
| 10.4 | Explizite Einschrittverfahren für AWP | 151 |
| 10.4.1 | Das explizite Euler-Verfahren | 151 |
| 10.5 | Konsistenz | 152 |
| 10.6 | Konvergenz | 155 |
| 10.7 | Explizite Runge–Kutta-Verfahren | 159 |
| 10.7.1 | Taylor-Verfahren | 159 |
| 10.7.2 | Idee der Runge–Kutta-Verfahren | 160 |
| 10.7.3 | Autonomisierung | 162 |

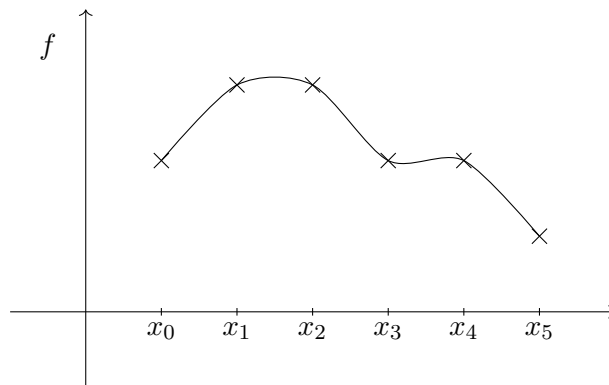
| | | |
|-----------|--|------------|
| 10.7.4 | Konstruktion von Runge-Kutta-Verfahren | 163 |
| 10.8 | Lineare Mehrschrittverfahren | 168 |
| 10.8.1 | Einführung | 168 |
| 10.8.2 | Mehrschrittverfahren für äquidistante Gitter | 169 |
| 10.8.3 | Konsistenz | 171 |
| 10.8.4 | Stabilität | 173 |
| 10.8.5 | Konvergenz | 175 |
| 11 | Steife Differentialgleichungen und implizite Verfahren | 177 |
| 11.1 | Wiederholung: Gewöhnliche Differentialgleichungen und Anfangswertpro- bleme | 177 |
| 11.1.1 | Gewöhnliche Differentialgleichungen | 177 |
| 11.1.2 | Anfangswertprobleme | 178 |
| 11.1.3 | Existenz und Eindeutigkeit | 178 |
| 11.1.4 | Evolution und Phasenfluss | 179 |
| 11.1.5 | Das explizite Euler-Verfahren | 179 |
| 11.1.6 | Konsistenz | 180 |
| 11.1.7 | Konvergenz | 181 |
| 11.1.8 | Explizite Runge-Kutta-Verfahren | 182 |
| 11.2 | Steife Differentialgleichungen | 184 |
| 11.2.1 | Steifheit und Kondition | 187 |
| 11.2.2 | Beispiel: Wieder das Modellproblem | 188 |
| 11.2.3 | Stabilität | 189 |
| 11.2.4 | Das implizite Euler-Verfahren | 190 |
| 11.3 | Stabilität von Einschrittverfahren | 191 |
| 11.3.1 | Wann sind Einschrittverfahren R stabil? | 193 |
| 11.4 | Implizite Runge-Kutta-Verfahren | 196 |
| 11.5 | Kollokationsverfahren | 201 |
| 11.5.1 | Gauß-Verfahren | 207 |
| 11.6 | Dissipative Differentialgleichungen | 208 |
| 11.7 | Linear-implizite Einschrittverfahren | 212 |
| 11.7.1 | Stabilität von Fixpunkten | 212 |
| 11.7.2 | Linear-implizite Runge-Kutta-Verfahren | 214 |
| 11.8 | Erhalt erster Integrale | 216 |
| 12 | Hamilton-Systeme | 223 |
| 12.1 | Hamilton-Systeme | 223 |
| 12.1.1 | Die Lagrange-Gleichungen | 223 |
| 12.1.2 | Die Hamiltonschen Gleichungen | 226 |
| 12.2 | Symplektizität | 228 |
| 12.3 | Symplektische Verfahren | 231 |
| 12.3.1 | Symplektische RK-Verfahren | 233 |
| 12.3.2 | Reversibilität vs. Symplektizität | 235 |
| 12.4 | Energieerhaltung | 236 |

| | | |
|--------|---|-----|
| 12.5 | Variationelle Integratoren | 237 |
| 12.5.1 | Idee der variationellen Integratoren | 239 |
| 12.5.2 | Variationelle Integratoren sind symplektisch | 241 |
| 12.5.3 | Variationelle Integratoren als klassische Einschrittverfahren | 242 |
| 12.5.4 | Variationelle Integratoren höherer Ordnung | 244 |
| 12.6 | Mechanische Systeme mit Nebenbedingungen | 250 |

1 Interpolation

Beispiel: Gegeben sind Temperaturen f_0, f_1, f_2, \dots gemessen zu bestimmten Zeitpunkten $x_0, x_1, x_2 \dots$

Frage: Wie ist die Temperatur zu einem Zeitpunkt \hat{x} mit $\hat{x} \neq x_i$?



Idee:

- Finde eine „sinnvolle“ Funktion $f : x \mapsto f(x)$ mit $f(x_i) = f_i \quad \forall i$
- Die gesuchte Temperatur ist $f(\hat{x})$
→ „intelligent geraten“

Definition (Interpolationsproblem). *Gegeben:*

- $n + 1$ Stützstellen $x_0, \dots, x_n \in \mathbb{R}$, paarweise verschieden,
- $n + 1$ Stützwerte $f_0, \dots, f_n \in \mathbb{R}$.

Finde eine „sinnvolle“ Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$ mit $f(x_i) = f_i \quad \forall i = 0, \dots, n$.

Was heißt „sinnvoll“?

- Eindeutig bestimmt in einer fixen Funktionsklasse
- Möglichst billig auszurechnen
- Kleiner Fehler
- „sinnvoll“ ist kein mathematischer Begriff – es kann Wissen aus der Anwendung dazukommen.

1.1 Polynominterpolation

- Versuche, f als Polynom zu konstruieren.
- Guter Kompromiss zwischen Flexibilität und einfacher Handhabung

Polynom n -ten Grades:

$$p(x) := a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \quad \forall x \in \mathbb{R}, a_n \neq 0$$

Sei Π_n die Menge aller Polynome vom Grad höchstens n .

Vielversprechender Ansatz:

- $n + 1$ Bedingungen $f(x_i) = f_i, \quad i = 0, \dots, n$
- $n + 1$ freie Parameter a_0, \dots, a_n

Kann man die Parameter so wählen, dass die Interpolationsbedingung erfüllt ist?

1.1.1 Lagrange-Interpolation

Konstruktion: Definiere $L_j : \mathbb{R} \rightarrow \mathbb{R}$ durch

$$L_j(x) := \prod_{\substack{i=0 \\ i \neq j}}^n \frac{x - x_i}{x_j - x_i} = \frac{(x - x_0)(x - x_1) \cdots (x - x_{j-1})(x - x_{j+1}) \cdots (x - x_n)}{(x_j - x_0)(x_j - x_1) \cdots (x_j - x_{j-1})(x_j - x_{j+1}) \cdots (x_j - x_n)}$$

→ Polynom vom Grad n

(der Nenner ist eine feste Zahl $\neq 0$, die nur von den Stützstellen abhängt)

Werte an den Stützstellen:

$$L_j(x_k) = \delta_{jk} := \begin{cases} 0 & \text{falls } j \neq k \\ 1 & \text{falls } j = k \end{cases}$$

(„Lagrange-Eigenschaft“)

Joseph-Louis Lagrange

- geb. 1736 in Turin als Guiseppe Lodovico Lagrangia
- gest. 1813 in Paris
- Mit 19 Lehrstuhl für Mathematik an der Königlichen Artillerieschule in Turin
- Ab 1766: Nachfolger von Leonard Euler (als Direktor) an der Preußischen Akademie der Wissenschaften in Berlin

- 1786 (Tod von Friedrich II) → Paris
- Napoleon machte ihn zum Grafen
- Im Pantheon begraben
- Auf dem Eiffelturm verewigt

Variationsrechnung: Lagrange-Multiplikatoren

Analysis: Lagrange-Restglied der Taylor-Formel

Definiere $p : \mathbb{R} \rightarrow \mathbb{R}$ durch

$$p(x) := \sum_{j=0}^n f_j L_j(x)$$

(Lagrange-Form des Interpolationspolynoms)

- Polynom n -ten Grades
- erfüllt die Interpolationsbedingung:

$$p(x_i) = \sum_{j=0}^n f_j L_j(x_i) = \sum_{j=0}^n f_j \delta_{ij} = f_i$$

Wir haben bewiesen:

Satz 1.1. *Zu $n + 1$ beliebigen Datenpaaren $(x_0, f_0), \dots, (x_n, f_n)$ mit paarweise verschiedenen Stützstellen existiert ein Polynom $p \in \Pi_n$, das die Interpolationsbedingung erfüllt.*

Satz 1.2. *Dieses Polynom ist eindeutig.*

Beweis. Seien $p, \tilde{p} \in \Pi_n$ beides Interpolationspolynome.

- Dann ist $p - \tilde{p} \in \Pi_n$.
- Für die Stützstellen x_k , $k = 0, \dots, n$ gilt

$$(p - \tilde{p})(x_k) = p(x_k) - \tilde{p}(x_k) = f_k - f_k = 0.$$

- $p - \tilde{p}$ hat also mindestens $n + 1$ Nullstellen.

⇒ Dann ist $p - \tilde{p}$ die Nullfunktion. □

Aufwand der Lagrange-Interpolation

- Auswerten eines einzelnen L_j an einer Stelle x : $\mathcal{O}(n)$ Operationen
- Berechnen der Summe $p(x) = \sum_{j=0}^n f_j L_j(x)$: $\mathcal{O}(n)$ Operationen
- Zusammen also $\mathcal{O}(n^2)$
- Auswertung an einem anderen Punkt: wieder $\mathcal{O}(n^2)$ Operationen.

1.1.2 Newton-Form des Interpolationspolynoms

Definition. Die Newton-Form des Interpolationspolynoms p ist

$$p(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots + c_n(x - x_0)(x - x_1) \cdots (x - x_{n-1})$$

mit passenden Koeffizienten $c_0, \dots, c_n \in \mathbb{R}$.

Wie berechnet man die Koeffizienten?

Induktiv:

Induktions-Anfang: $f_0 = p(x_0) = c_0$

Induktions-Schritt: Seien c_0, \dots, c_{j-1} bereits bekannt

$$\begin{aligned} f_j &= p(x_j) \\ &= c_0 + \sum_{k=1}^{j-1} c_k (x_j - x_0) \cdots (x_j - x_{k-1}) + c_j \underbrace{(x_j - x_0) \cdots (x_j - x_{j-1})}_{\neq 0} + \underbrace{0 + \dots + 0}_{n-j \text{ viele}} \end{aligned}$$

Also:

$$c_j = \frac{f_j - c_0 - \sum_{k=1}^{j-1} c_k (x_j - x_0) \cdots (x_j - x_{k-1})}{(x_j - x_0) \cdots (x_j - x_{j-1})}$$

Diese Prozedur entspricht gerade dem Lösen eines linearen Gleichungssystems mit unterer Dreiecksmatrix:

$$\begin{pmatrix} 1 & & & & & & \\ 1 & (x_1 - x_0) & & & & & \\ 1 & (x_2 - x_0) & (x_2 - x_0)(x_2 - x_1) & & & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ 1 & (x_n - x_0) & \cdots & \cdots & \prod_{i=0}^{n-1} (x_n - x_i) & & \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ \vdots \\ f_n \end{pmatrix}$$

Vorteil der Newton-Form: Neue Stützstellen können mit wenig Aufwand hinzugefügt werden. Bei der Lagrange-Form muss stattdessen alles komplett neu ausgerechnet werden.

Aufwand

- Die Matrix kann in $\mathcal{O}(n^2)$ (genauer: $\frac{1}{2}n^2$) Operationen aufgestellt werden.
- Ist die Matrix bekannt, so kann das System in $\mathcal{O}(n^2)$ Schritten gelöst werden.
- Man braucht also $\mathcal{O}(n^2)$ Operationen, um die Koeffizienten c_0, \dots, c_n zu bestimmen.

Angenommen, die Koeffizienten seien jetzt bekannt. Auswerten an festem x :

$$p(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots + c_n(x - x_0)(x - x_1) \cdots (x - x_{n-1})$$

Naiv: $\frac{n(n-1)}{2}$ Multiplikationen

Schlauer: Ausklammern!

$$p(x) = c_0 + (x - x_0) \left(c_1 + (x - x_1) (c_2 + (x - x_2) \dots) \right)$$

Algorithmus zum Ausrechnen von $p_0 = p(x)$:

$$\begin{aligned} p_n &= c_n \\ p_{n-1} &= c_{n-1} + (x - x_{n-1}) p_n \\ p_{n-2} &= c_{n-2} + (x - x_{n-2}) p_{n-1} \\ &\vdots \\ p_0 &= c_0 + (x - x_0) p_1 \end{aligned}$$

$\Rightarrow n - 1$ Multiplikationen \rightarrow Horner-Schema

Nach William George Horner, 1786 (Bristol) - 1837 (Bath)

- Direktor einer Schule in Bath

1.1.3 Interpolationsfehler

Welchen „Fehler“ macht man bei der Interpolation?

- Unklar: was heißt „Fehler“?
- Idee: Die gegebenen Werte sind Funktionswerte einer stetigen Funktion.

Definition (Interpolationsproblem II). Sei $f \in C[a, b]$ gegeben und

$$a \leq x_0 < x_1 < \dots < x_n \leq b$$

Stützstellen. Finde ein $p \in \Pi_n$ so dass $p(x_i) = f(x_i) \forall i = 0, \dots, n$.

Definition. Der Interpolationsfehler ist

$$\|f - p\|_\infty := \max_{x \in [a, b]} |f(x) - p(x)|.$$

\rightarrow Diesen Wert hätten wir gerne klein.

Geht das? Im Prinzip ja.

Satz 1.3 (Weierstraß). Sei $f \in C[a, b]$. Dann existiert eine Folge p_0, p_1, p_2, \dots mit $p_n \in \Pi_n$, so dass $\|f - p\|_\infty \xrightarrow{n \rightarrow \infty} 0$.

Beweis. Funktionalanalysis, 5. Semester. □

Karl Weierstraß 1815–1897

- Ab 1841 Gymnasiallehrer an verschiedenen Orten in Deutschland
- Ab 1856 → Professor in Berlin
- Solide Fundierung der Analysis („weierstraßsche Strenge“)
- Konvergenzkriterien für Reihen
- gleichmäßige Konvergenz
- Satz von Bolzano–Weierstraß

Kann man solche Polynome durch Interpolation konstruieren?

Hoffnung: Mehr Stützstellen → kleinerer Fehler

$$\lim \|f - p\|_\infty = 0 \quad \text{für immer feinere Aufteilung von } [a, b].$$

Beispiele: [Computer]

Beispielfunktion von Carl Runge (1856–1927):

$$f(x) = \frac{1}{1 + 25x^2}$$

Warum geht das schief?

Grund I: Uniform verteilte Stützstellen sind böse! (Den Grund sehen wir gleich)

Grund II: Die Runge-Funktion ist zwar C^∞ , aber die Werte der Ableitungen wachsen für höhere Ableitungsordnung:

$$\max_{[a,b]} \left| \frac{d^k}{dx^k} \left(\frac{1}{1 + 25x^2} \right) \right| \xrightarrow{k \rightarrow \infty} \infty.$$

Können wir etwas beweisen?

Definiere die Hilfsfunktion $w : \mathbb{R} \rightarrow \mathbb{R}$

$$w(x) := (x - x_0)(x - x_1) \cdots (x - x_n).$$

Satz 1.4. Sei $f \in C^{n+1}[a, b]$ und $a \leq x_0 < x_1 < \dots < x_n \leq b$. Sei $p_n \in \Pi_n$ das dazugehörige Interpolationspolynom. Dann existiert zu jedem $x \in [a, b]$ eine Zahl $\xi_x \in (a, b)$, so dass

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} w(x). \quad (1.1)$$

Insbesondere gilt also

$$\|f - p_n\|_\infty \leq \frac{1}{(n+1)!} \|f^{(n+1)}\|_\infty \|w\|_\infty.$$

Beweis. Fall 1: Sei $x = x_k$ eine Stützstelle. Dann folgt

$$f(x) = p_n(x), \quad w(x) = 0,$$

und (1.1) ist mit jedem $\xi_x \in (a, b)$ erfüllt.

Fall 2: Sei $x \neq x_k \forall k = 0, \dots, n$.

- Definiere Hilfsfunktion $g : [a, b] \rightarrow \mathbb{R}$

$$\begin{aligned} g(t) &:= f(t) - p_n(t) - [f(x) - p_n(x)] \cdot \underbrace{\frac{w(t)}{w(x)}}_{= \prod_{i=0}^n \frac{t-x_i}{x-x_i}} \\ &= \prod_{i=0}^n \frac{t-x_i}{x-x_i} \end{aligned}$$

- Da $f \in C^{n+1}$, $p \in C^\infty$ und $x \neq x_n$ folgt $g \in C^{n+1}$.
- Wir wollen den Satz von Rolle anwenden.

[Michel Rolle, 1652–1719, Pariser Akademie der Wissenschaften]

Satz 1.5 (Satz von Rolle).

Normal: Sei $g \in C^1$ mit zwei Nullstellen x_0, x_1

$$\Rightarrow \exists \xi, \text{ so dass } g'(\xi) = 0.$$

Verallgemeinert: Sei $g \in C^{n+1}$ und habe $n+2$ Nullstellen

$$\Rightarrow \exists \xi, \text{ so dass } g^{(n+1)}(\xi) = 0.$$

- Hat unser g tatsächlich $n+2$ Nullstellen?
- Wähle $t = x_k$:

$$g(x_k) = \underbrace{f(x_k) - p_n(x_k)}_{=0} - [f(x) - p_n(x)] \cdot \underbrace{\prod_{i=0}^n \frac{x_k - x_i}{x - x_i}}_{=0} = 0$$

- Wähle $t = x$:

$$g(x) = f(x) - p_n(x) - [f(x) - p_n(x)] \cdot \underbrace{\prod_{i=0}^n \frac{x - x_i}{x - x_i}}_{=1} = 0$$

$\Rightarrow n + 2$ Nullstellen

\Rightarrow Satz von Rolle: $\exists \xi \in (a, b)$ mit $g^{(n+1)}(\xi) = 0$

- Ausgeschrieben bedeutet das:

$$\begin{aligned} 0 &= g^{(n+1)}(\xi) \\ &= f^{(n+1)}(\xi) - \underbrace{p_n^{(n+1)}(\xi)}_{=0} - \underbrace{\left[f(x) - p_n(x) \right] \cdot \frac{d^{n+1}}{dt^{n+1}} \left(\prod_{i=0}^n \frac{t - x_i}{x - x_i} \right)_{t=\xi}}_{=?} \end{aligned}$$

- Zähler und Nenner des Produkts hatten wir als Hilfsfunktion eingeführt:

$$w(t) := \prod_{i=0}^n (t - x_i)$$

Dies ist ein Polynom vom Grad $n + 1$, also

$$w(t) = t^{n+1} + \dots$$

- $w(x)$ ist eine feste Zahl

$$\Rightarrow \prod_{i=0}^n \frac{t - x_i}{x - x_i} = \frac{1}{w(x)} \cdot t^{n+1} + \dots$$

$$\Rightarrow \frac{d^{n+1}}{dt^{n+1}} \prod_{i=0}^n \frac{t - x_i}{x - x_i} = \frac{(n+1)!}{w(x)}$$

- Also

$$0 = f^{(n+1)}(\xi) - \left[f(x) - p_n(x) \right] \frac{(n+1)!}{w(x)}$$

$$\Rightarrow f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} w(x) \quad \square$$

Folgerungen:

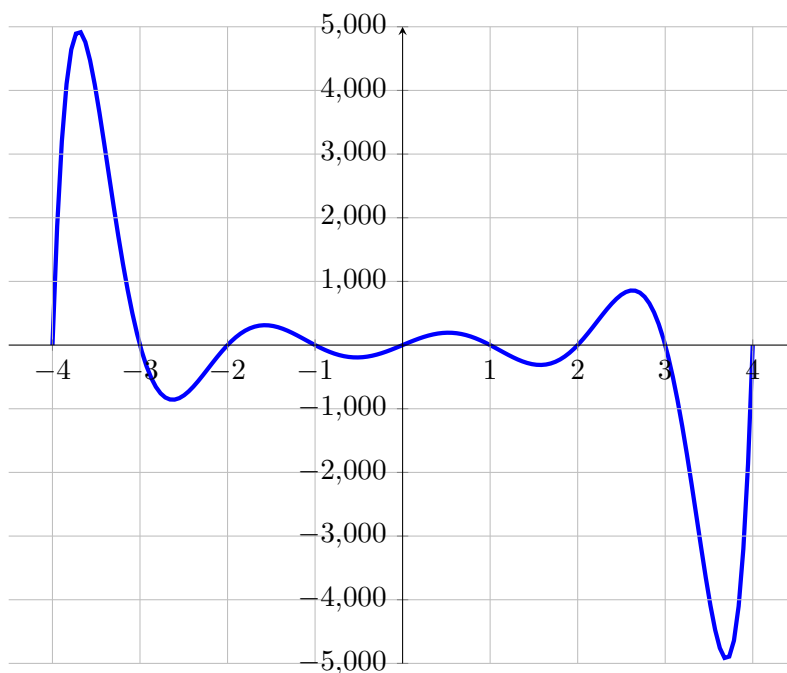
- 1) großes $f^{(n+1)}$ ist böse!
- 2) großes $w(x)$ ist auch böse!

Wahl der Stützstellen

Was kann man gegen 2) tun?

$\rightarrow w(x)$ hängt nur von den Stützstellen ab.

w für gleichverteilte Stützstellen $-4, -3, \dots, 3, 4$:



- 2b) Gleichverteilte Stützstellen sind böse!
- Intuitiv: wir brauchen mehr Stützstellen am Rand.
- Formal: Finde eine Stützstellenverteilung, die

$$\max_{x \in [a,b]} |w(x)|$$

minimiert.

Um die Notation einfach zu halten, beschränken wir uns ab jetzt auf das Intervall $[-1, 1]$.

Definition. Das n -te Tschebyscheff-Polynom ist

$$T_n(x) = \cos(n \arccos x), \quad x \in [-1, 1]$$

Eigenschaften:

1. Ja, das ist wirklich ein Polynom! (von Grad n)
2. Die Koeffizienten sind ganzzahlig, der höchste ist 2^{n-1} .
3. $|T_n(x)| \leq 1$ für $x \in [-1, 1]$.
4. Die Nullstellen sind

$$x_k = \cos\left(\frac{2k-1}{2n}\pi\right) \quad k = 1, \dots, n \quad \text{keine doppelten!}$$

Satz 1.6. Das Tschebyscheff-Polynom T_n ist minimal bzgl.

$$\|f\|_\infty = \max_{x \in [-1,1]} |f(x)|$$

unter allen Polynomen vom Grad n mit führenden Koeffizienten 2^{n-1} .

Was nützt uns das?

- Wir suchen Stützstellen x_0, \dots, x_n so dass

$$\max_{x \in [-1,1]} |w(x)| = \max_{x \in [-1,1]} |(x - x_0)(x - x_1) \cdots (x - x_n)|$$

minimal wird.

- w ist Polynom vom Grad $n + 1$, und normiert.
 - Das kleinste normierte Polynom vom Grad $n + 1$ auf $[-1, 1]$ ist $\frac{T_{n+1}}{2^n}$.
- \Rightarrow Wähle x_0, \dots, x_n als die Nullstellen von T_{n+1} .

Bilder der Tschebyscheff-Polynome

Computer: Interpolation mit Tschebyscheff-Polynomen

1.2 Spline-Interpolation

Bisher: Interpolation mit einem Polynom. Funktioniert, *aber*:

- Wird stark oszillatorisch, wenn die Anzahl der Stützstellen (also der Polynomgrad) groß wird.
- \Rightarrow Verbesserung: Nutze Nullstellen der Tschebyscheff-Polynome als Stützstellen.

ABER:

- Häufig kann man sich die Lage der Stützstellen nicht aussuchen.
- Häufig hat man *sehr viele* Stützstellen!

Alternativer Ansatz: Wir interpolieren mit stückweisen Polynomen.

Sei Δ eine Zerlegung von $[a, b]$ durch Stützstellen:

$$a = x_0 < x_1 < \cdots < x_n = b.$$

Definition. Ein Spline vom Grad m mit Glattheit $l \in \mathbb{N}$ zur Zerlegung Δ ist eine Funktion $s : [a, b] \rightarrow \mathbb{R}$ so dass:

- $s \in C^l([a, b])$,

- für alle $k = 0, \dots, n - 1$ ist die Einschränkung $s_k = s|_{[x_k, x_{k+1}]}$ ein Polynom vom Grad höchstens m .

Die Menge aller dieser Funktionen bezeichnen wir mit $S_m^l(\Delta)$.

Wortklärung: Spline → englisch für „Straklatte“, eine Art Kurvenlineal, die im Schiffsbau verwendet wird.

Jünger als Polynome: Die erste Erwähnung des Wortes „Spline“ findet sich in 1946.

Am weitaus häufigsten kommen kubische Splines (also stückweise kubische Polynome) zur Anwendung.

Warum ?

- Einerseits: einfach, Ordnung niedrig.
- Andererseits: Man kann damit C^2 -Funktionen bauen.

Stetigkeit der zweiten Ableitung ist für viele Anwendungen sehr wichtig!

1.2.1 C^1 -Interpolation

Seien f_0, \dots, f_n Stützwerte an den Stützstellen x_0, \dots, x_n . Wir suchen eine Funktion $s \in S_3^1(\Delta)$ mit $s(x_i) = f_i$ für alle $i = 0, \dots, n$.

- Auf jedem Intervall: s ist Polynom dritten Grades.
→ 4 Parameter, aber die Interpolationsbedingungen legen nur zwei davon fest.
- Weiterhin fordern wir Stetigkeit der Ableitung.
Das gibt eine weitere Bedingung pro Intervall.
- Eine Bedingung ist immer noch frei!
Man kann den Wert der Ableitung an den Stützstellen vorgeben (Hermite-Interpolation).
- Seien m_0, \dots, m_n weitere Werte an den Stützstellen.
→ Die m_i werden die Ableitung von s an den x_i

Auf dem Intervall $[x_k, x_{k+1}]$ machen wir jetzt folgenden Ansatz:

$$s_k(x) = a_k(x - x_k)^3 + b_k(x - x_k)^2 + c_k(x - x_k) + d_k$$

Ziel: Berechne a_k, b_k, c_k, d_k aus $f_k, f_{k+1}, m_k, m_{k+1}$.

1. $s_k(x_k) = d_k = f_k$
2. $s_k'(x_k) = c_k = m_k$

Jetzt interessant: Wert und Ableitung an der Stelle x_{k+1} .

→ Setze zur Abkürzung: $h_k = x_{k+1} - x_k$. Dann ist:

$$s_k(x_{k+1}) = a_k h_k^3 + b_k h_k^2 + m_k h_k + f_k = f_{k+1}$$

Da sieht man noch nichts, also noch die Ableitung

$$s'_k(x_{k+1}) = 3a_k h_k^2 + 2b_k h_k + m_k = m_{k+1}.$$

Dies ist ein lineares Gleichungssystem:

$$\begin{pmatrix} h_k^3 & h_k^2 \\ 3h_k^2 & 2h_k \end{pmatrix} \begin{pmatrix} a_k \\ b_k \end{pmatrix} = \begin{pmatrix} f_{k+1} - m_k h_k - f_k \\ m_{k+1} - m_k \end{pmatrix}.$$

Die Matrix ist invertierbar, denn $\det(\cdot) = 2h_k^4 - 3h_k^4 = -h_k^4 \neq 0$.

Also existiert eine eindeutige Wahl der Parameter a_k, b_k .

Satz 1.7. Sei eine Zerlegung Δ des Intervalls $[a, b]$ gegeben. Dann gibt es für beliebige Zahlen f_0, \dots, f_n und m_0, \dots, m_n genau ein Spline $s \in S_3^1(\Delta)$ mit

$$s(x_k) = f_k \text{ und } s'(x_k) = m_k, \quad \forall k = 0, \dots, n.$$

- Wo kommen die m_k her?
- Möglicherweise sind die Ableitungen von f bekannt (Hermite- Interpolation)! Und wenn nicht?
- Die Interpolierende s ist C^1 , aber nicht C^2 .

Idee: Wähle die m_k so, dass s zweimal stetig differenzierbar wird.

1.2.2 Kubische C^2 -Splines

- C^2 heißt: $s''_k(x_{k+1}) = s''_{k+1}(x_{k+1}), \forall k = 0, \dots, n - 2$
- Mit den vorherigen Ansatz für s_k :

$$s''_k(x_{k+1}) = 6a_k(x - x_k) + 2b_k$$

also

$$\begin{aligned} s''_k(x_{k+1}) &= 6a_k h_k + 2b_k & s''_{k+1}(x_{k+1}) &= 2b_{k+1} \\ \Rightarrow 3a_k h_k + b_k &= b_{k+1} \end{aligned} \tag{1.2}$$

- Hier hängen also die Polynomkoeffizienten für die unterschiedlichen Teilintervalle voneinander ab.

Ausrechnen der Koeffizienten a_k, b_k

- Bisher wissen wir nur, dass die Koeffizienten das lineare Gleichungssystem

$$\begin{pmatrix} h_k^3 & h_k^2 \\ 3h_k^2 & 2h_k \end{pmatrix} \begin{pmatrix} a_k \\ b_k \end{pmatrix} = \begin{pmatrix} f_{k+1} - m_k h_k - f_k \\ m_{k+1} - m_k \end{pmatrix}$$

lösen.

Die Lösung davon ist:

$$a_k = -\frac{2}{h_k^3}(f_{k+1} - f_k) + \frac{1}{h_k^2}(m_k + m_{k+1})$$

$$b_k = \frac{3}{h_k^2}(f_{k+1} - f_k) - \frac{1}{h_k}(2m_k + m_{k+1}).$$

Einsetzen in (1.2):

$$\underbrace{-\frac{6}{h_k^2}(f_{k+1} - f_k) + \frac{3}{h_k}(m_k + m_{k+1})}_{3a_k h_k} + \underbrace{\frac{3}{h_k^2}(f_{k+1} - f_k) - \frac{1}{h_k}(2m_k + m_{k+1})}_{b_k}$$

$$= \underbrace{\frac{3}{h_{k+1}^2}(f_{k+2} - f_{k+1}) - \frac{1}{h_{k+1}}(2m_{k+1} + m_{k+2})}_{b_{k+1}}$$

Umstellen:

$$\frac{1}{h_k}(m_k + 2m_{k+1}) + \frac{1}{h_{k+1}}(2m_{k+1} + m_{k+2}) = \frac{3}{h_k^2}(f_{k+1} - f_k) + \frac{3}{h_{k+1}^2}(f_{k+2} - f_{k+1}).$$

Erweitern mit $h_k h_{k+1}$:

$$h_{k+1}m_k + 2(h_{k+1} + h_k)m_{k+1} + h_k m_{k+2} = \frac{3h_{k+1}}{h_k}(f_{k+1} - f_k) + \frac{3h_k}{h_{k+1}}(f_{k+2} - f_{k+1})$$

Lineares Gleichungssystem!

- $n + 1$ Variablen, aber nur $n - 1$ Gleichungen.
- Rang: $n - 1$. Lösbar, aber nicht eindeutig lösbar.

Natürlich nicht: Wir haben m_0 und m_n noch nicht festgelegt!

Randbedingungen:

Verschiedene Möglichkeiten:

1. Natürliche Randbedingungen

$$s''(x_0) = s''(x_n) = 0.$$

2. Vollständige Randbedingungen

$$s'(x_0) = f'(a), \quad s'(x_n) = f'(b).$$

3. Periodische Randbedingungen

$$s'(x_0) = s'(x_n), \quad s''(x_0) = s''(x_n).$$

4. etc...

Das Gleichungssystem ist *tridiagonal*! Damit ist es in $O(n)$ Schritten lösbar (Thomas-Algorithmus)!

1.2.3 Interpolationsfehler

Es existieren viele Abschätzungen für den Fehler in diversen Normen, auch Fehler der Ableitung.

Eine der besten:

Satz 1.8 (Hall und Meyer [9]). Sei Δ eine Zerlegung, und setze $h := \max |x_{k+1} - x_k|$. Sei $f \in C^4([a, b])$ und sei $s \in S_3^2(\Delta)$ die Spline-Interpolierende mit natürlichen Randbedingungen (d.h., $s''(a) = s''(b) = 0$). Dann gilt

$$\|f - s\|_\infty \leq \frac{5}{384} h^4 \|f^{(4)}\|_\infty.$$

Beachte: Resultat ist unabhängig von der Lage der Stützstellen!

- Beweis leider zu lang zum Vorrechnen.

Hier ein schwächeres Resultat als Alternative:

Definiere dafür die 2-Norm

$$\|f\|_2 := \sqrt{\int_a^b |f(x)|^2 dx}.$$

Satz 1.9. Sei Δ eine Zerlegung von $[a, b]$ und $f \in C^2([a, b])$. Sei $s \in S_3^2(\Delta)$ die Spline-Interpolierende mit natürlichen oder vollständigen Randbedingungen. Dann ist

$$\|f - s\|_\infty \leq \frac{1}{2} h^{\frac{3}{2}} \|f''\|_2.$$

Beweis. • Setze $r = f - s$.

- r hat mindestens die $n + 1$ Nullstellen x_0, \dots, x_n .
- Satz von Rolle: r' hat mindestens n Nullstellen.

- Zwei benachbarte Nullstellen von r' sind höchstens $2h$ voneinander entfernt.
- r' ist stetig, $[a, b]$ kompakt $\Rightarrow \exists z \in [a, b]$ mit $|r'(z)| = \|r'\|_\infty$.
- Sei z_0 die z am nächsten gelegene Nullstelle von r' :
 $\Rightarrow |z - z_0| \leq \frac{1}{2} \cdot 2h = h$.
- Sei O.B.d.A. $z_0 \leq z$.
- Rechnen:

$$\begin{aligned} \|r'\|_\infty^2 &= |r'(z)|^2 = |r'(z) \underbrace{-r'(z_0)}_0|^2 \\ &= \left| \int_{z_0}^z r''(x) dx \right|^2. \end{aligned}$$

Cauchy-Schwarz: $|\langle x, y \rangle|^2 \leq \langle x, x \rangle \langle y, y \rangle$

$$\begin{aligned} &= \left| \int_{z_0}^z r''(x) \cdot 1 dx \right|^2 \leq \int_{z_0}^z (r''(x))^2 dx \cdot \int_{z_0}^z 1 dx \\ &\leq h \|r''\|_2^2. \end{aligned} \tag{1.3}$$

Jetzt der gleiche Trick für r selbst:

- $\exists y \in [a, b]$ so dass $|r(y)| = \|r\|_\infty$.
- Sei y_0 die y am nächsten liegende Nullstelle von r .
- O.B.d.A. $y_0 \leq y$.
- Rechnen:

$$\begin{aligned} \|r\|_\infty &= |r(y) - r(y_0)| = \left| \int_{y_0}^y r'(x) dx \right| \\ &\leq \|r'\|_\infty \int_{y_0}^y dx = \frac{h}{2} \|r'\|_\infty. \end{aligned}$$

- Einsetzen von (1.3):

$$\|r\|_\infty \leq \frac{h}{2} \sqrt{h} \|r''\|_2.$$

- Also

$$\|f - s\|_\infty \leq \frac{1}{2} h^{\frac{3}{2}} \|f'' - s''\|_2. \quad \square$$

Verwunderung: Rechts soll $\|f''\|_2$ stehen, nicht $\|f'' - s''\|_2$!?!?

Das gilt auch, denn es ist immer $\|f'' - s''\|_2 \leq \|f''\|_2$!

Wie beweist man das?

Dreiecksungleichung? Nein, denn

$$\|f''\| = \|f'' - s'' + s''\| \leq \|f'' - s''\| + \|s''\|$$

Erstaunlich: es gilt die Dreiecks-Ungleichung mit Gleichheit!

Satz 1.10. *Es gilt*

$$\|f''\|_2^2 = \|f'' - s''\|_2^2 + \|s''\|_2^2,$$

also insbesondere

$$\|f'' - s''\| \leq \|f''\|.$$

Beachte: Mit dem Satz des Pythagoras kann man das wie folgt interpretieren:

$f'' - s''$ steht senkrecht auf s'' im Sinne des Skalarprodukts $\langle v, w \rangle = \int_a^b vw \, dx$.

Beweis. Wir zeigen $\|f''\|_2^2 - \|f'' - s''\|_2^2 = \|s''\|_2^2$.

Rechnen:

$$\begin{aligned} \|f''\|_2^2 - \|f'' - s''\|_2^2 &= \int_a^b (f''(x))^2 \, dx - \int_a^b (f''(x) - s''(x))^2 \, dx \\ &= \int_a^b \left[\underbrace{(f''(x))^2 - (f''(x))^2}_{=0} + 2f''(x)s''(x) - \underbrace{(s''(x))^2 - (s''(x))^2 + (s''(x))^2}_{=0} \right] dx \\ &= 2 \underbrace{\int_a^b (f''(x) - s''(x))s''(x) \, dx}_{=:J} + \underbrace{\int_a^b (s''(x))^2 \, dx}_{=\|s''(x)\|_2^2}. \end{aligned}$$

Wir zeigen $J = 0$ (Beachte: J ist das L_2 -Skalarprodukt von $f'' - s''$ und s'' , und $J = 0$ bedeutet gerade dass diese zwei Vektoren senkrecht aufeinander stehen!)

Partielle Integration:

$$J = \underbrace{(f' - s')s'' \Big|_a^b}_{J_1} - \underbrace{\int_a^b (f' - s')s''' \, dx}_{J_2}$$

1. $J_2 = 0$, denn s''' ist auf jedem Teilintervall konstant, und damit

$$\begin{aligned} J_2 &= \sum_{k=0}^{n-1} s''' \left(\frac{x_k + x_{k+1}}{2} \right) \int_{x_k}^{x_{k+1}} (f' - s') \, dx \\ &= \sum_{k=0}^{n-1} s''' \left(\frac{x_k + x_{k+1}}{2} \right) (f - s) \Big|_{x_k}^{x_{k+1}} \\ &= 0. \end{aligned}$$

2. Weiterhin soll $J_1 = (f'(b) - s'(b))s''(b) - (f'(a) - s'(a))s''(a)$ gleich 0 sein.

Dies gilt wenn man die Randbedingungen passend wählt, z.B.:

- Natürliche Randbedingungen: $s''(a) = s''(b) = 0$
- Vollständige Randbedingungen: $f'(a) - s'(a) = 0, f'(b) - s'(b) = 0$ □

Aus

$$\|f''\|_2^2 - \|f'' - s''\|_2^2 = \|s''\|_2^2$$

folgt

$$\|s''\|_2 \leq \|f''\|_2.$$

Korollar. s minimiert $\|\cdot\|_2$ in der Menge aller C^2 -Funktionen die die Interpolations- und Randbedingungen erfüllen.

Beweis. • Sei \tilde{s} eine C^2 -Funktion die die Rand- und Interpolationsbedingungen erfüllt, und $\|\tilde{s}''\|_s < \|s''\|_2$.

- Dann ist s auch Spline-Interpolierende von \tilde{s} .
- Mit Satz 1.10: $\|s''\|_2 \leq \|\tilde{s}''\|_s$.
- Widerspruch!

□

1.2.4 Geometrische Interpretation

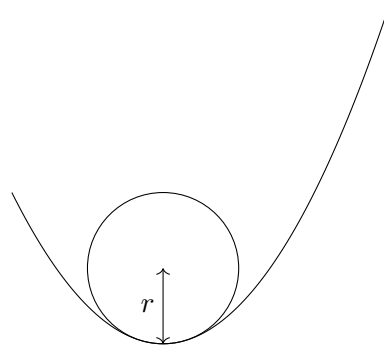
Konstrukteure wollen Kurven, die so glatt wie möglich sind.

Was heißt hier „so glatt wie möglich“?

- „So gerade wie möglich, keine unnötige Kurve“
- Die „Krümmung“ soll möglichst gering sein.

Was ist die Krümmung einer ebenen Kurve ?

- Gerade: Krümmung = 0.
- Kreis: Krümmung konstant = $\frac{1}{r}$
- Allgemeine Kurve: Inverser Radius des Schmiegekreises. (Der Kreis, der eine Kurve an einem Punkt am besten annähert.)



Sei die Kurve Graph einer Funktion $g : \mathbb{R} \rightarrow \mathbb{R}$.

- Krümmung von g ist $\frac{g''(x)}{(1+(g'(x))^2)^{\frac{3}{2}}}$.
- Falls g' klein ist ist das $\approx g''$

\Rightarrow „Kubische Splines minimieren die Krümmung“.

1.2.5 B-Splines

- Basis-Splines
- Entwickelt in den 1950ern und 1960ern in der Luftfahrt- und Automobilindustrie.

Keine neue Art von Funktionen!

Stattdessen: Einen neue Basis für den Spline-Raum $S_m^l(\Delta)$.

- Problem mit der alten Basis: Basis ist global: jeder Wert f_k , $k = 0, n$ beeinflusst den Wert der Splinefunktion auf ganz $[a, b]$.
- Kein Problem, wenn die Stützpunkte f_i fest gegeben sind.
- Aber: Problem, wenn Splines zum Modellieren von Kurven und Flächen verwendet werden.
- Deshalb: konstruiere lokale Basis von $S_m^l(\Delta)$
(D.h. Basisfunktionen haben lokalen Träger)
- Beschreibe Funktionen/ Kurven/ Flächen durch Koeffizienten bzgl dieser Basis!
Solche Splines sind nicht interpolierend. In der Modellierung ist das aber egal.

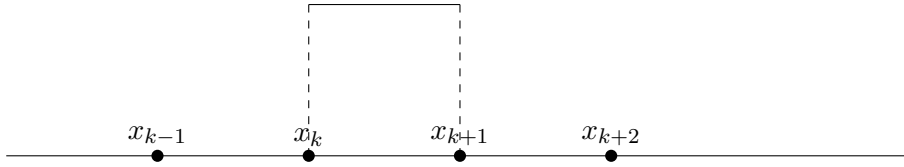
Definition. Sei $x_1 \leq \dots \leq x_n$ eine Folge von Knoten (d.h.: Stützstellen). Dann sind die B-Splines $N_p(x)$ der Ordnung p für $p = 1, \dots, n$ und $i = 1, \dots, n - p$ erklärt durch:

$$N_{k,1}(x) := \begin{cases} 1 & \text{falls } x_k \leq x < x_{k+1}, \\ 0 & \text{sonst.} \end{cases}$$

$$N_{k,p}(x) := \frac{x - x_k}{x_{k+p-1} - x_k} N_{k,p-1}(x) + \frac{x_{k+p} - x}{x_{k+p} - x_{k+1}} N_{k+1,p-1}(x)$$

Beispiel. • $p = 1$:

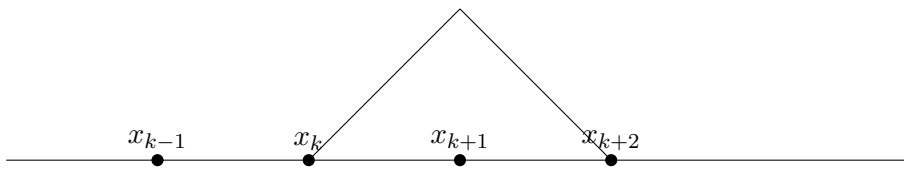
$$N_{k,1}(x) := \begin{cases} 1 & \text{falls } x_k \leq x < x_{k+1} \\ 0 & \text{sonst.} \end{cases}$$



• $p = 2$:

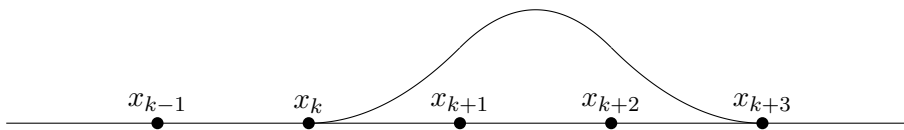
$$N_{k,2}(x) = \frac{x - x_k}{x_{k+1} - x_k} \chi_{[x_k, x_{k+1}]}(x) + \frac{x_{k+2} - x}{x_{k+2} - x_{k+1}} \chi_{[x_{k+1}, x_{k+2}]}(x)$$

$$= \begin{cases} \frac{x - x_k}{x_{k+1} - x_k} & \text{falls } x_k \leq x \leq x_{k+1} \\ \frac{x_{k+2} - x}{x_{k+2} - x_{k+1}} & \text{falls } x_{k+1} \leq x \leq x_{k+2} \\ 0 & \text{sonst.} \end{cases}$$



- $p \rightarrow p + 1$ Stückweise Multiplikation mit Linearfaktor
- $\Rightarrow N_{k,p}$ ist stückweises Polynom der Ordnung $p - 1$.
- Information aus p Intervallen \Rightarrow lokaler Träger

• $p = 3$



Eigenschaft:

$$N_{k,p}(x) \geq 0, \quad \forall k, p, x$$

Darstellung von Spline-Funktionen:

$$s(x) = \sum_{k=1}^n f_k N_{k,p}(x)$$

Achtung: es gilt NICHT $s(x_k) = f_k$!

Ableitung:

$$N'_{k,p}(x) = (p-1) \left(\frac{-N_{k+1,p-1}(x)}{x_{k+p} - x_{k+1}} + \frac{N_{k,p-1}(x)}{x_{k+p-1} - x_k} \right)$$

Großer Vorteil: Man kann die Glattheit der Funktionen an den Stützstellen kontrollieren, indem man mehrfache Stützstellen zulässt.

Satz 1.11. Sei x ein m -facher Knoten, d.h

$$x_{j-1} < x_j = x_{j+1} \cdots = x_{j+m-1} < x_{j+m}$$

Dann ist $N_{k,p}$ an der Stelle x_j mindestens $p-1-m$ -mal stetig differenzierbar.

2 Numerische Quadratur/Integration

Gegeben: integrierbare Funktion $f : [a, b] \rightarrow \mathbb{R}$

Ziel: Finde $I(f) := \int_a^b f(x) dx$

Satz 2.1 (Fundamentalsatz). Sei F Stammfunktion von f , also $F' = f$. Dann ist:

$$I(f) = F(b) - F(a).$$

Aber: Die Stammfunktion F ist häufig nicht bekannt!

Oder: F ist schwierig zu handhaben!

Beispiel.

$$f(x) = e^{x^2} \quad F(x) = \sum_{k=0}^{\infty} \frac{x^{2k+1}}{(2k+1)k!}$$

Besser: $I(f)$ numerisch (d.h. approximativ) berechnen.

Idee: Wir wissen, wie man Polynome integriert.

- Sei $\Delta = a \leq x_0 < x_1 \dots x_n \leq b$ eine Zerlegung des Integrationsgebiets.
- Sei p_n das dazugehörige Interpolationspolynom von f .
- Approximiere

$$I(f) = \int_a^b f(x) dx \quad \text{durch} \quad Q_n(f) := \int_a^b p_n(x) dx.$$

Lagrange-Form von p_n :

$$p_n(x) = \sum_{k=0}^n f(x_k) L_k(x)$$

Einsetzen ergibt

$$\begin{aligned} Q_n(f) &= \int_a^b p_n(x) dx = \int_a^b \sum_{k=0}^n f(x_k) L_k(x) dx = \sum_{k=0}^n f(x_k) \int_a^b L_k(x) dx \\ &= \sum_{k=0}^n f(x_k) a_k, \end{aligned}$$

mit reellen Koeffizienten a_k , die nicht von f abhängen.

Wie genau ist diese Approximation?

- Wir ahnen schon: es hängt von der Verteilung der Stützstellen ab.

2.1 Die Newton-Cotes-Formeln

(Roger Cotes 1682–1716, engl. Mathematiker)

Sei $h = \frac{b-a}{n}$ und $x_k = a + hk$.

Das dazugehörige $Q_n(\cdot)$ heißt (abgeschlossene) Newton-Cotes-Formel.

Trapezregel: $n = 1$

- Stützstellen $x_0 = a, x_1 = b$
- Lagrange-Polynome:

$$L_0(x) = \frac{x - x_1}{x_0 - x_1} = \frac{b - x}{h} \quad L_1(x) = \frac{x - x_0}{x_1 - x_0} = \frac{x - a}{h}$$

- Integrale der Lagrange-Polynome

$$\begin{aligned} \int_a^b L_0(x) dx &= \frac{1}{h} \int_a^b (b - x) dx = \frac{1}{h} \left(bx - \frac{x^2}{2} \right) \Big|_a^b \\ &= \frac{1}{h} \left[b^2 - \frac{b^2}{2} - ab + \frac{a^2}{2} \right] = \frac{1}{2h} (b - a)^2 = \frac{h}{2} \end{aligned}$$

und

$$\int_a^b L_1(x) dx = \frac{h}{2}.$$

- Also:

$$Q_1(f) = f(x_0) \int_a^b L_0(x) dx + f(x_1) \int_a^b L_1(x) dx = \frac{h}{2} f(x_0) + \frac{h}{2} f(x_1).$$

Simpson-Regel: $n = 2$

$$Q_2(f) = \frac{h}{6} f(x_0) + \frac{4h}{6} f(x_1) + \frac{h}{6} f(x_2)$$

2.2 Quadraturfehler der Simpson-Regel

Satz 2.2. Sei $f \in C^4[a, b]$. Dann ist

$$|Q_2(f) - I(f)| \leq \frac{1}{12} h^5 \|f^{(4)}\|_\infty.$$

[Wenig überraschend: Polynome vom Grad < 3 werden exakt integriert.]

Prüfen: Bei Dd steht sogar 1/90.

Beweis. • Wir benutzen die Hilfsfunktion aus Satz 1.4

$$w(x) := (x - a)(x - x_1)(x - b) = (y + h) y (y - h)$$

mit $y := x - x_1$.

- Das Integral davon ist

$$W(h) := \int_{-h}^h (y + h) y (y - h) dy = 0$$

da der Integrand ungerade ist.

Jetzt der Satz 1.4 zum Interpolationsfehler von p_2 :

- Es existiert ein $\xi \in (a, b)$ so dass:

$$f(x) - p_2(x) = \frac{f'''(\xi)}{3!} w(x)$$

- Integriere über $[a, b]$, und addiere eine Null:

$$\begin{aligned} \left| \int_a^b (f(x) - p_2(x)) dx \right| &= \frac{1}{6} f'''(x_1) w(x) + \frac{1}{6} f'''(\xi) w(x) - \frac{1}{6} f'''(x_1) w(x) \\ &= \frac{1}{6} \left| f'''(x_1) \underbrace{\int_a^b w(x) dx}_{=0} + \int_a^b (f'''(\xi) - f'''(x_1)) w(x) dx \right| \\ &\leq \frac{1}{6} \max_{x \in [a, b]} |f'''(x) - f'''(x_1)| \int_a^b |w(x)| dx. \end{aligned}$$

- Mittelwertsatz (f ist C^4):

$\exists \zeta \in (a, b)$ so dass

$$|f'''(x) - f'''(x_1)| = |f^{(4)}(\zeta)| \underbrace{|x - x_1|}_{\leq h} \leq h \|f^{(4)}\|_\infty.$$

- Integrals des Betrags der Hilfsfunktion:

$$\int_a^b |w(x)| dx = \int_{-h}^h |(y + h) y (y - h)| dy = -2 \int_0^h (y^2 - h^2) y dy = \frac{1}{2} h^4$$

- Deshalb

$$\begin{aligned} |Q_2(f) - I(f)| &= \left| \int_a^b [f(x) - p_2(x)] dx \right| \\ &\leq \frac{1}{6} h \|f^{(4)}\|_\infty \underbrace{\int_a^b |w(x)| dx}_{=\frac{1}{2} h^4} \\ &= \frac{1}{12} h^5 \|f^{(4)}\|_\infty. \end{aligned}$$

□

Ähnlich zeigt man:

Lemma 2.1. Für die Trapezregel Q_1 gilt

$$|Q_1(f) - I(f)| \leq \frac{1}{12} h^3 \|f''\|_\infty.$$

Wir haben also

$$\begin{array}{ll} Q_1 \rightarrow h^3 & \text{Trapez-Regel} \\ Q_2 \rightarrow h^5 & \text{Simpson} \\ Q_3 \rightarrow h^7 & p = 3 \end{array}$$

Vermutlich liefert Q_3 die Konvergenzordnung h^7 ?

Satz 2.3. Sei Q_n die n -te Newton-Cotes-Formel

a) Falls n gerade und $f \in C^{n+2}$, so existiert ein $\xi \in (a, b)$ so dass

$$I(f) - Q_n(f) = \frac{h^{n+3} f^{(n+2)}(\xi)}{(n+2)!} \int_0^n t^2(t-1)(t-2)\cdots(t-n) dt$$

b) Falls n ungerade und $f \in C^{n+1}$, so existiert ein $\xi \in (a, b)$ so dass

$$I(f) - Q_n(f) = \frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!} \int_0^n t(t-1)(t-2)\cdots(t-n) dt$$

Moral: Nur beim Übergang von ungerader zu gerader Ordnung gewinnt man tatsächlich etwas.

Im Prinzip könnte man Formeln immer höherer Ordnung nehmen, und würde immer genauere Approximationen bekommen.

Aber: Ab $n \geq 7$ bekommt man teilweise negative Gewichte, d.h.

$$Q_n(f) = \sum_{k=0}^n f(x_k) a_k \quad \text{mit} \quad a_k = \int_a^b L_k(x) dx < 0.$$

Solche Quadraturformeln verletzen die folgende Monotonie des Integrals:

$$\text{aus } f(x) \geq 0 \quad \forall x \in (a, b) \quad \text{folgt} \quad \int_a^b f dx \geq 0.$$

Deshalb verwendet man i.A. nur Newton-Cotes-Formeln bis $n = 6$.

2.3 Zusammengesetzte (summierte) Newton-Cotes-Formeln

- Zerlege Intervall in Teilintervalle
- Quadraturformel niedriger Ordnung auf jedem Teilintervall
- Zerlege $[a, b]$ in r gleichgroße Stücke der Breite $h := \frac{b-a}{r}$.
- Zerlege jedes dieser Stücke wiederum in n gleichgroße Stücke.
- Zusammengesetzte Trapezregel:

$$T_h(f) := \frac{h}{2} (f(x_0) + 2f(x_1) + 2f(x_2) + \cdots + 2f(x_{l-1}) + f(x_{rn}))$$

- Zusammengesetzte Simpsonregel:

$$S_h(f) := \frac{h}{6} (f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2 \cdots + 2f(x_{l-2}) + 4f(x_{l-1}) + f(x_{rn}))$$

Satz 2.4.

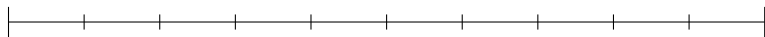
- Für alle $f \in C^2$ gilt

$$|T_h(f) - I(f)| \leq \frac{b-a}{12} h^2 \|f''\|_\infty.$$

- Für alle $f \in C^4$ gilt

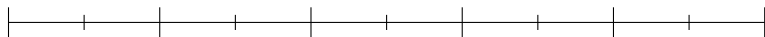
$$|S_h(f) - I(f)| \leq \frac{b-a}{180} h^4 \|f^{(4)}\|_\infty$$

Aufwand: Trapezregel



$$n = 1, \quad h = \frac{b-a}{r}$$

Simpson



$$n = 2, \quad h = 2h_{\text{Trapez}}$$

Eine Funktionsauswertung pro Stützstelle:

- Summierte Trapez- & Simpson-Regel sind in etwa gleich teuer.
- Aber die summierte Simpson-Regel ist deutlich genauer.

2.4 Gauß-Quadratur

Newton-Cotes: Approximiere

$$\int_a^b f(x) dx \quad \text{durch} \quad \int_a^b p_n(x) dx,$$

wobei p_n das Interpolationspolynom n -ten Grades für gleichverteilte Stützstellen ist.

Bewiesen: Diese Approximation ist exakt für alle Polynome vom Grad höchstens n .

Bekannt:

- Gleichverteilte Stützstellen sind nicht optimal.
- Können wir mit anderen Stützstellen genauer integrieren?
- Idealerweise:
 - a) Wir haben $2n + 2$ Freiheitsgrade ($n + 1$ Werte + $n + 1$ Pos.)
 - b) Ein Polynom vom Grad $2n + 1$ hat $2n + 2$ Koeffizienten
 ⇒ Wir wollen Polynome vom Grad $2n + 1$ exakt integrieren können.

Aber: Die Positionen der Stützstellen gehen nichtlinear in den Fehler der Quadraturformel ein.

Deshalb reicht ein simples Abzählen der Freiheitsgrade nicht aus.

2.4.1 Gewichtete Integrale

Die Gauß-Quadratur wird meistens in einem etwas allgemeineren Kontext eingeführt.

- Der Preis für diese zusätzliche Allgemeinheit ist gering.

Folgendes Problem wird gelöst: Wie integrieren wir z.B. \sqrt{x} ?

- Diese Funktion hat unbeschränkte Ableitungen!
- Die Fehlerabschätzungen sagen deshalb große Quadraturfehler voraus!

Idee: Gewichtete Integrale

Sei $w : [a, b] \rightarrow \mathbb{R}$ eine Gewichtsfunktion, d.h. $w(x) \geq 0$ auf $[a, b]$ und $w(x) > 0$ auf (a, b) .

Approximiere

$$I(f, w) := \int_a^b f(x)w(x) dx \quad \text{durch} \quad \sum_{k=0}^n a_k f(x_k).$$

Hoffnung: Nur die Glattheit von f soll in den Fehlerabschätzungen auftauchen. Die Gewichtsfunktion w muss irgendwie in die Quadraturgewichte a_k einfließen.

„Klassische“ Gewichte:

| w | $[a, b]$ | Name |
|---|---------------------|---------------------|
| 1 | $[-1, 1]$ | Gauss-Legendre |
| $\frac{a}{\sqrt{1-x^2}}$ | $[-1, 1]$ | Gauss-Tschebyscheff |
| e^{-x^2} | $(-\infty, \infty)$ | Gauss-Hermite |
| $(1-x)^\alpha(1+x)^\beta, \quad \alpha, \beta > -1$ | $[-1, 1]$ | Gauss-Jacobi |

Können wir jetzt die Stützstellen so wählen dass

$$Q_{n,w}(f) = \int_a^b w(x)f(x) dx$$

gilt für alle Polynome vom Grad höchstens $2n + 1$?

2.4.2 Orthogonale Polynome

Die Geheimwaffe: orthogonale Polynome

- Das sind Familien von Polynomen

$$p_0, p_1, p_2, \dots \quad p_i \in \Pi_i$$

die paarweise orthogonal sind (bzgl. eines noch zu wählenden Skalarprodukts).

Lemma 2.2. Für alle $n \in \mathbb{N}$ sei $Q_{n,w}(\cdot)$ eine Quadraturformel mit Stützstellen

$$x_{0n} < x_{1n} < \dots < x_{nn}.$$

Für jedes $n \in \mathbb{N}$ definiere das Polynom

$$p_{n+1}(x) := (x - x_{0n})(x - x_{1n}) \dots (x - x_{nn}) \in \Pi_{n+1}.$$

Falls $Q_{n,w}$ für alle Polynome aus Π_{2n+1} exakt ist, dann steht p_{n+1} w -senkrecht auf allen Polynomen niedrigerer Ordnung. Daraus wiederum folgt dass die p_0, \dots, p_n paarweise w -orthogonal sind, d.h.

$$(p_i, p_j)_w := \int_a^b w(x)p_i(x)p_j(x) dx = 0 \quad i \neq j, \quad i, j = 0, \dots, n$$

Beweis. Es reicht zu zeigen, dass $(p_j, p_{n+1})_w = 0$ für alle $j < n + 1$.

- Sei also $j < n + 1$, und somit $p_j p_{n+1} \in \Pi_{2n+1}$.
- Rechnen:

$$\begin{aligned} (p_j, p_{n+1})_w &= \int_a^b w p_j p_{n+1} dx \\ &= Q_{n,w}(p_j p_{n+1}) \quad \text{da } Q_{n,w} \text{ exakt auf } \Pi_{2n+1} \\ &= \sum_{k=0}^n a_{kn} p_j(x_{kn}) \underbrace{p_{n+1}(x_{kn})}_{=0} dx \\ &= 0. \end{aligned}$$

□

Die gesuchten Knoten x_{kn} müssen also die Nullstellen von Orthogonalpolynomen sein. Gibt es so etwas immer?

Satz 2.5 (Deuffhard und Hohmann, Satz 6.2). *Sei $w : (a, b) \rightarrow \mathbb{R}$ eine Gewichtsfunktion, so dass*

$$\|p\|_w := \sqrt{(p, p)_w} = \sqrt{\int_a^b wp^2 dx} < \infty$$

für alle Polynome p . Dann gibt es eine eindeutig bestimmte Familie von Orthogonalpolynomen $p_k \in \Pi_k$ mit führendem Koeffizienten 1.

Die Polynome genügen der Dreitermrekursion

$$p_k(x) = (x + \alpha_k)p_{k-1} + \beta_k p_{k-2}(x) \quad k = 1, 2, \dots$$

mit

$$p_{-1} := 0, \quad p_0 := 1$$

und Koeffizienten

$$\alpha_k = -\frac{(xp_{k-1}, p_{k-1})_w}{(p_{k-1}, p_{k-1})_w}, \quad \beta_k = -\frac{(p_{k-1}, p_{k-1})_w}{(p_{k-2}, p_{k-2})_w}.$$

Okay, es gibt solche Polynome. Aber kann man die Nullstellen auch wirklich als Quadraturpunkte nutzen?

Satz 2.6 (Deuffhard und Hohmann, Satz 6.5). *Das Orthogonalpolynom $p_k \in \Pi_k$ hat genau k einfache Nullstellen in (a, b) .*

Beweis.

- Seien x_1, \dots, x_m die m verschiedenen Punkte $x_i \in (a, b)$, an denen p_k sein Vorzeichen wechselt.
- Zu zeigen: $m \geq k$.
- Definiere $q(x) := (x - x_1) \cdots (x - x_m)$
- $q(x)$ wechselt an den gleichen Stellen sein Vorzeichen.
- wqp_k wechselt sein Vorzeichen gar nicht!
- Daraus folgt

$$(q, p_k)_w = \int_a^b wqp_k dx \neq 0.$$

- p_k steht aber senkrecht auf Π_{k-1} (Denn p_0, \dots, p_{k-1} ist Basis von Π_{k-1})
- $q \notin \Pi_{k-1}$
- $\Rightarrow m \geq k$ □

2.4.3 Quadratur mit Orthogonalpolynomen

Wir wissen jetzt

- $Q_{n,w}$ exakt für $\Pi_{2n+1} \Rightarrow$ Stützstellen sind Nullstellen eines Orthogonalpolynoms.

Aber gilt auch die andere Richtung?

- Die Gewichte wählen wir auf jeden Fall wie gehabt:

$$a_{kn} = \int_a^b L_{kn}(x) dx.$$

Damit ist die Formel automatisch exakt bis Ordnung n .

Das reicht schon!

Lemma 2.3 (Deuffhard und Hohmann, Lemma 9.10). *Seien x_0, \dots, x_n die Nullstellen des $(n+1)$ -ten Orthogonalpolynoms p_{n+1} . Sei $Q_{n,w}(f) := \sum_{i=0}^n a_i f(x_i)$ eine beliebige Quadraturformel. Dann gilt*

$$Q_{n,w} \text{ exakt auf } \Pi_n \quad \Longrightarrow \quad Q_{n,w} \text{ exakt auf } \Pi_{2n+1}.$$

Beweis.

- Sei $Q_{n,w}$ exakt auf Π_n und $p \in \Pi_{2n+1}$.
- Polynomdivision: $\exists q, r \in \Pi_n$ so dass

$$p = qp_{n+1} + r.$$

- p_{n+1} ist w -senkrecht zu Π_n

$$\begin{aligned} \Rightarrow \int_a^b wp dx &= \underbrace{\int_a^b wqp_{n+1} dx}_{=0} + \int_a^b wr dx = Q_{n,w}(r) \\ &= \sum_{k=0}^n a_k r(x_k) \\ &= \sum_{k=0}^n a_k \left(q(x_k) \underbrace{p_{n+1}(x_k)}_{=0} + r(x_k) \right) \\ &= Q_{n,w}(p). \quad \square \end{aligned}$$

Wie genau ist die Gauss-Quadratur, falls der Integrand kein Polynom in Π_{2n+1} ist?

Satz 2.7 (Deuffhard und Hohmann, Satz 9.12). *Sei $f \in C^{2n+2}$. Dann existiert ein $\xi \in [a, b]$ so dass*

$$\int_a^b wf dx - Q_{n,w}(f) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \|p_{n+1}\|_w^2.$$

Die rechte Seite hängt also tatsächlich nicht von den Ableitungen der Gewichtsfunktion ab!

Die Gauß-Quadratur hat noch einen weiteren wichtigen Vorteil:

Satz 2.8. *Alle Gewichte einer Gauss-Formel sind positiv!*

Beweis.

- Sei $q \in \Pi_{2n+1}$ ein Polynom, das nur an einem Knoten x_k nicht verschwindet, also

$$q(x_k) \neq 0 \quad \text{und} \quad q(x_i) = 0 \quad \forall i \neq k.$$

- Wir integrieren dieses Polynom:

$$\int_a^b wq \, dx = a_k q(x_k) \quad \Rightarrow \quad a_k = \frac{1}{q(x_k)} \int_a^b wq \, dx.$$

- a_k wird positiv wenn q zum Beispiel ein Quadrat ist.

Wähle ein bestimmtes q :

$$\begin{aligned} q(x) &:= \left(\frac{p_{n+1}(x)}{x - x_k} \right)^2 \\ &= (x - x_0)^2 \cdots (x - x_{k-1})^2 (x - x_{k+1})^2 \cdots (x - x_n)^2. \end{aligned}$$

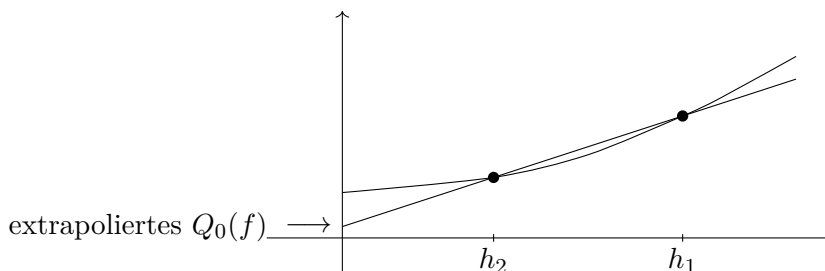
- Dieses Polynom ist punktweise nicht-negativ, und da die Nullstellen paarweise verschieden sind gilt $q(x_k) > 0$. Deshalb folgt

$$a_k = \frac{1}{q(x_k)} \int_a^b wq \, dx > 0. \quad \square$$

2.5 Extrapolationsverfahren für Quadratur

Idee: Sei $Q_h(f)$ eine Approximation von $I(f)$, die von einem Parameter h abhängt.

- Annahme: $\lim_{h \rightarrow 0} Q_h(f) = I(f)$
- Berechne zwei (oder mehr) Werte $Q_{h_1}(f)$, $Q_{h_2}(f)$
- Versuche daraus auf $Q_0(f)$ zu schließen



Gibt es Quadraturregeln für die diese Extrapolation besonders gut funktioniert?

2.5.1 Romberg-Quadratur

[Nach: Werner Romberg, 1909 Berlin - 2003 Heidelberg, emigrierte in die Sowjetunion und Norwegen, Professor in Trondheim und Heidelberg]

Romberg-Quadratur wendet Extrapolation auf die summierte Trapezregel an:

$$T(h) := \frac{h}{2}f(a) + h \sum_{i=1}^{n-1} f(a+ih) + \frac{h}{2}f(b).$$

Um das Fehlerverhalten von $T(h)$ zu verstehen brauchen wir eine asymptotische Entwicklung:

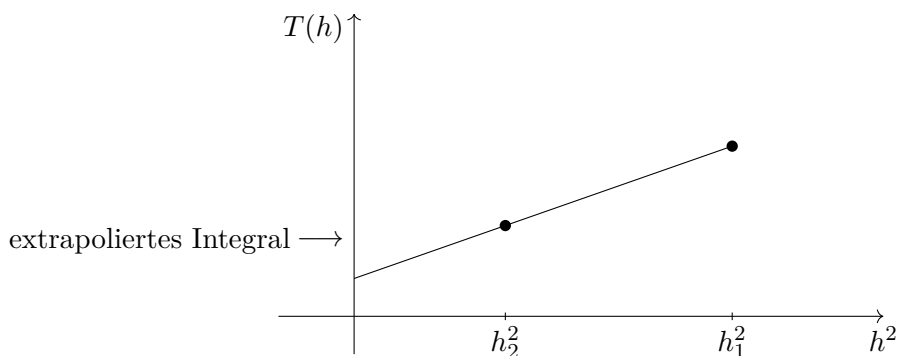
Satz 2.9 (Deuffhard und Hohmann, Satz 9.6). *Sei $f \in C^{2m+1}[a, b]$ und $h = \frac{b-a}{n}$ für ein $n \in \mathbb{N}$. Dann gilt für die summierte Trapezsumme*

$$\int_a^b f(x) dx = T(h) + c_2 h^2 + c_4 h^4 + \dots + c_{2m} h^{2m} + R_{2m+2}(h) h^{2m+2}$$

mit Koeffizienten c_2, c_4, \dots . Der Restterm $R_{2m+2}(h) h^{2m+2}$ ist beschränkt in h .

(Interessant: für $m \rightarrow \infty$ gilt $\tau_{2m} \rightarrow \infty$. Deshalb konvergiert die Reihe selbst für $f \in C^\infty$ nicht!)

Da die Reihe von T_h nur gerade Potenzen enthält, empfiehlt es sich, die Extrapolation in der Variablen h^2 statt h zu machen:



- Lineares Interpolationspolynom durch die Punkte $(h_1^2, T(h_1))$ und $(h_2^2, T(h_2))$:

$$p(h^2) = T(h_1) + \frac{T(h_2) - T(h_1)}{h_2^2 - h_1^2} (h^2 - h_1^2)$$

Extrapolierter Wert:

$$p(0) = T(h_1) - \frac{T(h_2) - T(h_1)}{h_2^2 - h_1^2} h_1^2. \quad (2.1)$$

Beispiel: $h_1 = 2h_2$

$$\begin{aligned}
 p(0) &= T(2h_2) - \frac{T(h_2) - T(2h_2)}{h_2^2 - (2h_2)^2} (2h_2)^2 \\
 &= T(2h_2) + \frac{(T(h_2) - T(2h_2)) \cdot 4}{3} \\
 &= \frac{4T(h_2) - T(2h_2)}{3} \\
 &= \frac{1}{3} \left[h_2(2f(a + 0 \cdot h_2) + 4f(a + 1 \cdot h_2) + \dots + 2f(b)) \right. \\
 &\quad \left. + 2h_2 \left(-\frac{1}{2}f(a + 0 \cdot h_2) - \frac{1}{2}f(a + 2 \cdot h_2) - \dots - \frac{1}{2}f(b) \right) \right] \\
 &= S(h), \text{ die summierte Simpson-Formel!}
 \end{aligned}$$

→ Wiederentdeckung einer bekannten Formel, aber mit dem Extrapolationszugang sind wir flexibler.

Alternative Begründung warum die Formel (2.1) eine gute Idee ist:

- Asymptotische Entwicklung der Trapezregel:

I)

$$\int_a^b f(x) dx = T(h) + c_2 h^2 + c_4 h^4 + c_6 h^6 + \dots$$

II)

$$\int_a^b f(x) dx = T(2h) + 4c_2 h^2 + 16c_4 h^4 + 64c_6 h^6 + \dots$$

- Multipliziere (I) mit 4, ziehe dann (II) ab:

$$\begin{aligned}
 3 \int_a^b f(x) dx &= 4T(h) - T(2h) - 12c_4 h^4 - 60c_6 h^6 - \dots \\
 \Rightarrow \int_a^b f(x) dx &= \frac{4T(h) - T(2h)}{3} - 4c_4 h^4 - 20c_6 h^6 - \dots
 \end{aligned}$$

Der Fehler ist jetzt also in $O(h^4)$ statt in $O(h^2)$.

2.5.2 Extrapolation mit mehr als zwei Stützstellen

- Angenommen wir haben Werte von T für k unterschiedliche Schrittweiten h_1, \dots, h_k .
- Wähle p das Interpolationspolynom mit Stützstellen $h_1^2, h_2^2, \dots, h_k^2$ zu den Werten $T(h_1), T(h_2), \dots, T(h_k)$.
- Werte p an der Stelle 0 aus.

Das kann man mit dem Wissen machen, das wir schon von der Polynominterpolation haben.

Andererseits bietet sich hier ein besonderer Trick an: Wir wollen ja nicht das ganze Polynom p , sondern nur dessen Wert an einer einzigen Stelle.

Der Algorithmus von Aitken und Neville Der Algorithmus von Aitken und Neville ist ein schnelles Verfahren um ein Interpolationspolynom an einem einzigen Punkt auszuwerten. Er funktioniert allgemein, d.h., nicht nur im Kontext von numerischer Quadratur.

Lemma 2.4. Für das Interpolationspolynom $P = P(f|x_0, \dots, x_n)$ gilt

$$P(f|x_0, \dots, x_n)(x) = \frac{(x_0 - x)P(f|x_1, \dots, x_n)(x) - (x_n - x)P(f|x_0, \dots, x_{n-1})(x)}{x_0 x_n}.$$

Sei x jetzt fest (Für die Romberg-Quadratur z.B: $x = 0$).

- Vereinfachte Notation: $P_{ik} := P(f|x_{i-k}, \dots, x_i)(x)$ für $i \geq k$
- Gesucht: P_{nn}
- Schema von Neville:

$$\begin{array}{ccccccc}
 & P_{00} & & & & & \\
 & P_{10} & \nearrow & & P_{11} & & \\
 & \vdots & \nearrow & & \ddots & & \\
 P_{n-1,0} & \rightarrow & & \cdots & & P_{n-1,n-1} & \\
 P_{n,0} & \nearrow & & \cdots & & P_{n,n-1} & \nearrow P_{nn}
 \end{array}$$

3 Lineare Gleichungssysteme

Problem: Finde $x \in \mathbb{R}^n$ so dass

$$Ax = b$$

für gegebenes $A \in \mathbb{R}^{n \times n}$ und $b \in \mathbb{R}^n$.

Dies ist ein sehr wichtiges Problem!

- 1) Viele Anwendungsprobleme lassen sich in dieser Form schreiben
- 2) Die einzigen Gleichungssysteme, die sich tatsächlich direkt (d.h. nicht iterativ) lösen lassen
- 3) Nichtlineare Gleichungen werden häufig gelöst, indem Folgen von linearen Gleichungssystemen gelöst werden.

Erinnerung:

Satz 3.1. *Es existiert ein eindeutiges $x \in \mathbb{R}^n$ mit $Ax = b$ genau dann wenn $\det A \neq 0$.*

Cramer'sche Regel: (1750, vorher schon Leibniz bekannt)

$$x_i = \frac{\det A_i}{\det A}, \quad i = 1, \dots, n \quad \text{wobei } A_i = A \text{ mit der } i\text{-ten Spalte durch } b \text{ ersetzt}$$

Funktioniert, ist aber viel zu teuer!

- $n + 1$ Determinanten
- 1 Determinante: $\det A = \sum_{\sigma \in S_n} \text{sgn}(\sigma) a_{1,\sigma(1)} \cdots a_{n,\sigma(n)}$
- $n \cdot n!$ Multiplikationen (Es gibt auch schnellere Algorithmen)

| n | $(n+1) \cdot n \cdot n!$ | Zeit (bei 1 GFLOP) |
|----------|--------------------------|--------------------|
| 1 | 2 | |
| 2 | 12 | |
| 3 | 72 | |
| 10 | $400 \cdot 10^6$ | 0,4 s |
| 11 | $5,26 \cdot 10^9$ | 5,26 s |
| 12 | | 74 s |
| 13 | | 1133s (ca. 19 min) |
| 14 | | 5 h |
| 15 | | 87 h |
| 16 | | 65 Tage |
| \vdots | \vdots | \vdots |

Man könnte versucht sein, A^{-1} zu berechnen und dann einfach

$$x = A^{-1}b$$

zu setzen.

Davon ist abzuraten:

- Man löst quasi $Ax = b$ für alle b
- Manchmal ist das schwierig, selbst wenn es für einzelne b leicht ist.
- Wenn man nur endliche Rechengenauigkeit hat wird x möglicherweise sehr unpräzise.
- Eine goldene Regel der Numerik: Invertieren von Matrizen ist nie notwendig, und sollte immer vermieden werden.
- Genaueres später

3.1 Dreiecksmatrizen

Das Lösen ist einfacher, wenn die Matrizen eine besondere Form haben.

Ein wichtiger Fall: Dreiecksmatrizen

$$A = R = \left(\begin{array}{c|c} \triangle & \\ \hline 0 & \end{array} \right)$$

$$Rx = z, \quad r_{ij} = 0, \quad i > j$$

$$r_{11}x_1 + r_{12}x_2 + \cdots + r_{1n}x_n = z_1$$

$$r_{22}x_2 + \cdots + r_{2n}x_n = z_2$$

$$\vdots$$

$$r_{nn}x_n = z_n$$

Lösen durch *Rückwärtssubstitution*:

$$\begin{aligned}x_n &= z_n / r_{nn} \\x_{n-1} &= (z_n - r_{n-1,n}x_n) / r_{n-1,n-1} \\&\vdots \\x_1 &= (z_1 - r_{12}x_2 - \dots - r_{1n}x_n) / r_{11}.\end{aligned}$$

Durchführbar, falls alle $r_{ii} \neq 0$, $i = 1, \dots, n$.

- Es gilt: $\det R = r_{11} \cdot r_{22} \cdots r_{nn}$

\Rightarrow Durchführbar, falls $\det R \neq 0$

Aufwand: $\approx \frac{n^2}{2}$ Multiplikationen/Divisionen

3.2 Das Gaußsche Eliminationsverfahren

- Von Gauß 1809 beschrieben (als bekannt erwähnt!)
- Lagrange schon 1759 bekannt
- In China schon kurz vor Christi Geburt bekannt

Allgemeines Gleichungssystem:

$$Ax = b$$

Ausgeschrieben:

$$\begin{array}{cccccc}a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\& & \vdots & & & & \vdots & & \vdots \\a_{n1}x_1 & + & a_{n2}x_2 & + & \dots & + & a_{nn}x_n & = & b_n\end{array}$$

Idee: Forme Gleichungssystem in ein oberes Dreieckssystem um.

Dafür ausreichend: eliminiere alle Einträge unterhalb von a_{11} (der Rest geht rekursiv).

Wie macht man das?

- Voraussetzung: $a_{11} \neq 0$
- Um $a_{i1}x_1$ in Zeile i , ($i = 2, \dots, n$) zu eliminieren:
 \rightarrow Subtrahiere von Zeile i ein Vielfaches von Zeile 1

Neue Zeile i :

$$\underbrace{(a_{i1} - l_{i1}a_{11})}_{\substack{= 0 \\ \text{falls } l_{i1} = \frac{a_{i1}}{a_{11}}}} x_1 + \underbrace{(a_{i2} - l_{i1}a_{12})}_{=: a'_{i2}} x_2 + \dots + \underbrace{(a_{in} - l_{i1}a_{1n})}_{=: a'_{in}} x_n = \underbrace{b_i - l_{i1}b_1}_{=: b'_i}$$

- Durchführbar falls $a_{11} \neq 0$.
- In den Zeilen $2, \dots, n$ steht eine $(n-1) \times (n-1)$ -Matrix
- Man wende das Verfahren darauf an:

\Rightarrow Folge von Matrizen:

$$A = A^{(1)} \rightarrow A^{(2)} \rightarrow A^{(n)} =: R$$

Wir haben das allgemeine Gleichungssystem $Ax = b$ in ein Dreieckssystem $Rx = z$ umgeformt.

3.2.1 Aufwand

- Rechenaufwand für die Umformung: $O(n^3)$ Multiplikationen
- Lösen des Dreieckssystems: $O(n^2)$ Schritte

\rightarrow insgesamt also $O(n^3)$

Immer noch viel, aber deutlich besser als $(n+1) \cdot n \cdot n!$

Laufzeit:

| n | n^3 | Zeit bei 1 GFLOP |
|------|----------------|------------------|
| 1 | 1 | |
| 2 | 8 | |
| 3 | 27 | |
| 10 | 1000 | |
| 100 | $1 \cdot 10^6$ | |
| 1000 | $1 \cdot 10^9$ | 1s |

3.3 Durchführbarkeit

Vorwärtselimination ist durchführbar, falls $a_{kk}^{(k)} \neq 0$, $k = 1, \dots, n$.

Problem: Diese Zahlen entstehen erst während des Verfahrens.

Entscheide a priori, ob das Gauß-Verfahren für ein A durchführbar ist.

Definition. Eine Matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ heißt streng diagonaldominant, falls

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \text{ für alle } i = 1, \dots, n$$

Lemma 3.1. Falls $A \in \mathbb{R}^{n \times n}$ streng diagonaldominant ist, so ist die Vorwärtselimination durchführbar.

Beweis. Die Matrix $A^{(1)}$ ist streng diagonaldominant.

→ Also insbesondere $|a_{11}| > 0$

→ Der erste Schritt ist durchführbar.

Induktion: Aus $A^{(k)}$ streng diagonaldominant folgt $A^{(k+1)}$ streng diagonaldominant.

- Sei i eine Matrixzeile.
- Falls $i \leq k$, so ist die i -te Zeile von $A^{(k+1)}$ identisch mit der i -ten Zeile von $A^{(k)}$
→ nichts zu zeigen.
- Sei also $i > k$.

$$\begin{aligned}
 \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}^{(k+1)}| &= \sum_{\substack{j=k+1 \\ j \neq i}}^n |a_{ij}^{(k+1)}| \quad (\text{für kleinere } j \text{ ist } a_{ij}^{(k+1)} = 0) \\
 &= \sum_{\substack{j=k+1 \\ j \neq i}}^n \left| a_{ij}^{(k)} - \frac{a_{kj}^{(k)} a_{ik}^{(k)}}{a_{kk}^{(k)}} \right| \quad (k+1\text{-ter Eliminationsschritt}) \\
 &\leq \sum_{\substack{j=k+1 \\ j \neq i}}^n |a_{ij}^{(k)}| + \left| \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \right| \sum_{\substack{j=k+1 \\ j \neq i}}^n |a_{kj}^{(k)}| \\
 &= \underbrace{\sum_{\substack{j=k \\ j \neq i}}^n |a_{ij}^{(k)}| - |a_{ik}^{(k)}|}_{< |a_{ii}^{(k)}|} + \left| \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \right| \underbrace{\left(\sum_{j=k+1}^n |a_{kj}^{(k)}| - |a_{ki}^{(k)}| \right)}_{< |a_{kk}^{(k)}|} \\
 &< |a_{ii}^{(k)}| - \cancel{|a_{ik}^{(k)}|} + \left| \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \right| \left(\cancel{|a_{kk}^{(k)}|} - |a_{ki}^{(k)}| \right) \\
 &= |a_{ii}^{(k)}| - \left| \frac{a_{ik}^{(k)} a_{ki}^{(k)}}{a_{kk}^{(k)}} \right| \\
 &\leq \left| a_{ii}^{(k)} - \frac{a_{ik}^{(k)} a_{ki}^{(k)}}{a_{kk}^{(k)}} \right| \\
 &= |a_{ii}^{(k+1)}|. \quad \square
 \end{aligned}$$

Korollar. Sei A streng diagonaldominant. Dann ist auch die Rückwärtssubstitution durchführbar.

Beweis. $R = A^{(n)}$ ist streng diagonaldominant, also sind alle Diagonalelemente $\neq 0$. \square

3.4 Die LR-Zerlegung

Folge von Matrizen:

$$A = A^{(1)} \rightarrow A^{(2)} \rightarrow A^{(n)} := R$$

Der Übergang von $A^{(k)}, b^{(k)}$ zu $A^{(k+1)}, b^{(k+1)}$ ist linear

\Rightarrow Es gibt eine Matrix $L_k \in \mathbb{R}^{n \times n}$, so dass

$$A^{(k+1)} = L_k A^{(k)} \quad \text{und} \quad b^{(k+1)} = L_k b^{(k)}.$$

Die Matrix L_k heißt *Frobenius-Matrix*.

- Explizite Form:

$$L_k = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -l_{n,k} & & & 1 \end{pmatrix}$$

- Interessanterweise:

$$L_k^{-1} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & l_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & l_{n,k} & & & 1 \end{pmatrix}$$

- Außerdem:

$$L^{-1} := L_{n-1} \cdots L_{n-2} \cdots L_1 = \begin{pmatrix} 1 & & & & & \\ -l_{21} & 1 & & & & \\ -l_{31} & -l_{32} & 1 & & & \\ \vdots & & & \ddots & & \\ -l_{n1} & & & & -l_{n,n-1} & 1 \end{pmatrix}$$

Damit ist

$$R = A^{(n)} = L_{n-1} A^{(n-1)} = L_{n-1} L_{n-2} A^{(n-2)} = L^{-1} A^{(1)} = L^{-1} A,$$

also

$$A = LR \quad \text{und} \quad z = L^{-1} b.$$

Diese Zerlegung von A in zwei Dreiecksmatrizen heißt *LR-Zerlegung*.

(Häufig auch *LU-Zerlegung*, wegen englisch: "lower-upper")

Die Gauß-Elimination nimmt damit folgende Form an:

- 1) Berechne Zerlegung $A = LR$
- 2) Bestimme $z \in \mathbb{R}^n$ so dass $Lz = b$ (Vorwärtssubstitution)
- 3) Bestimme $x \in \mathbb{R}^n$ so dass $Rx = z$ (Rückwärtssubstitution)

Vorteil dieser Sichtweise: nur 1) ist teuer ($O(n^3)$), Schritte 2) und 3) sind in $O(n^2)$.

- Falls man mehrere Gleichungssysteme $Ax_i = b_i$, $i = 1, \dots, m$ zu lösen hat, muss man den teuren Schritt 1) nur einmal machen.

Bemerkung. Die LR -Zerlegung bietet auch eine klassische Möglichkeit, um die Determinante von A auszurechnen. Denn

$$\det A = \det LR = \det L \cdot \det R = \prod_{i=1}^n \underbrace{l_{ii}}_{=1} \cdot \prod_{i=1}^n r_{ii} = \prod_{i=1}^n r_{ii}.$$

3.5 Pivot-Strategien

Die Elemente der Matrizen $A^{(k)}$ durch die dividiert wird nennt man *Pivot-Elemente* (Pivot: frz.: Drehpunkt)

3.5.1 Probleme mit der einfachen Gauß-Elimination

Es ist einfach, Fälle zu konstruieren, wo Gauß-Elimination versagt, z. B.

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \det A = -1 \neq 0.$$

→ Lösung: vertausche einfach die beiden Zeilen!

Bevor wir diese Idee formalisieren:

→ Das Problem ist noch schlimmer:

Auch Pivot-Elemente, die $\neq 0$, aber sehr klein sind machen Probleme.

Beispiel.

$$\begin{aligned} 10^{-4}x_1 + x_2 &= 1 \\ x_1 + x_2 &= 2 \end{aligned}$$

Zum Lösen eliminieren wir a_{21} :

Die zweite Zeile wird:

$$0 - 9999x_2 = -9998$$

Rückwärtssubstitution:

$$\begin{aligned} x_2 &= \frac{9998}{9999} = 0,\overline{9998} \\ x_1 &= 10000(1 - x_2) = 10000\left(\frac{1}{9999}\right) = 1,\overline{0001}. \end{aligned}$$

Realität: Rechner können nur Zahlen mit einer Maximalzahl an Dezimalstellen darstellen.

Beispiel: 3 Dezimalstellen

Runden der exakten Lösung auf 3 Stellen: $x_1 = 1, x_2 = 1$.

Dieses Ergebnis würden wir erwarten. Aber...

Jetzt machen wir Gauß-Elimination und haben zu jedem Zeitpunkt nur 3 gültige Stellen:

$$\begin{aligned} 1,00 \cdot 10^{-4} x_1 + 1,00 x_2 &= 1,00 \\ 1,00 x_1 + 1,00 x_2 &= 2,00 \end{aligned}$$

Eliminiere die zweite Zeile. Dafür brauchen wir den Faktor

$$l_{21} = \frac{1,00}{1,00 \cdot 10^{-4}} = 1,00 \cdot 10^4$$

Aus der zweiten Zeilen wird:

$$(1,00 - 1,00 \cdot 10^4 \cdot 1,00 \cdot 10^{-4}) x_1 + (1,00 - 1,00 \cdot 10^4 \cdot 1,00) x_2 = 2,00 - 1,00 \cdot 10^4 \cdot 1,00$$

bzw.

$$0 \cdot x_1 + \underbrace{(1,00 - 10000)}_{-9990} x_2 = 2,00 - 10000$$

$$x_2 = -9990$$

Rückwärtssubstitution:

$$\begin{aligned} x_2 &= 1,00 && \checkmark \\ x_1 &= 0 && \ominus \end{aligned}$$

Der Algorithmus hat anscheinend ein völlig falsches Ergebnis produziert!

Fazit: Das Gauß-Verfahren versagt

- falls eins der Pivotelemente $a_{kk}^{(k)}$ gleich Null ist.
- Anscheinend versagt es zumindest manchmal auch, wenn $a_{kk}^{(k)} \neq 0$ aber klein ist, und wir nur eine endliche Rechengenauigkeit haben (so richtig verstanden haben wir das noch nicht).

3.5.2 Pivot-Strategien

Als $a_{11} = 0$ war hat Vertauschen der Zeilen geholfen.

Das probieren wir jetzt noch mal:

$$\begin{aligned} 1,00x_1 + 1,00x_2 &= 2,00 \\ 1,00 \cdot 10^{-4}x_1 + 1,00x_2 &= 1,00 \end{aligned}$$

Faktor zum Eliminieren von \tilde{a}_{21} :

$$\tilde{l}_{21} = 1,00 \cdot 10^{-4}.$$

Dreieckssystem:

$$\begin{aligned} 1,00x_1 + 1,00x_2 &= 2,00 \\ 1,00x_2 &= 1,00 \end{aligned}$$

Rückwärtssubstitution:

$$x_2 = 1,00, \quad x_1 = 1,00$$

Warum hat das besser funktioniert?

$$|\tilde{l}_{21}| < 1 \quad \text{bzw.} \quad |\tilde{a}_{11}| \geq |\tilde{a}_{21}|$$

Idee der Spaltenpivotisierung:

- Vertausche Zeilen(!) der Matrix, um ein möglichst geeignetes Pivot-Element zu finden.

Algorithmus:

a) Wähle im Schritt $A^{(k)} \rightarrow A^{(k+1)}$ ein $p \in \{k, \dots, n\}$ so dass

$$\left| a_{pk}^{(k)} \right| \geq \left| a_{jk}^{(k)} \right| \quad \forall j = k, \dots, n.$$

b) Vertausche die Zeilen p und k

$$A^{(k)} \rightarrow \tilde{A}^{(k)} \quad \text{mit} \quad \tilde{a}_{ij}^{(k)} = \begin{cases} a_{kj}^{(k)} & \text{falls } i = p \\ a_{pj}^{(k)} & \text{falls } i = k \\ a_{ij}^{(k)} & \text{sonst.} \end{cases}$$

$$\text{Es gilt immer } |\tilde{l}_{ik}| = \left| \frac{\tilde{a}_{ik}^{(k)}}{\tilde{a}_{kk}^{(k)}} \right| = \left| \frac{a_{ik}^{(k)}}{a_{pk}^{(k)}} \right| \leq 1$$

c) Führe den nächsten Eliminationsschritt mit $\tilde{A}^{(k)}$ statt $A^{(k)}$ aus:

$$\tilde{A}^{(k)} \rightarrow A^{(k+1)}.$$

Satz 3.2 (Deuffhard und Hohmann, Satz 1.8). *Falls A nicht singulär ist, so ist die Gauß-Elimination mit Spaltenpivotisierung durchführbar.*

Zusätzlicher Aufwand: $O(n^2)$

Es gibt auch:

- Zeilenpivotisierung \rightarrow vertausche Spalten
- vollständige Pivotisierung: vertausche Zeilen und Spalten
Aufwand: $O(n^3)$, wird nur sehr selten verwendet.

3.6 Nachiteration

(engl. *iterative refinement*)

Selbst mit Pivotisierung kann die Lösung noch ziemlich ungenau sein.

Idee der Nachiteration:

- Sei x die exakte Lösung.
- Sei \tilde{x} die numerische Lösung.
- Der Fehler $\delta := x - \tilde{x}$ löst die *Defektgleichung*

$$A\delta = r(\tilde{x}) := b - A\tilde{x} \tag{3.1}$$

- Löse dieses System numerisch. Nenne die Lösung $\tilde{\delta}$.
- Dann ist $\tilde{x} + \tilde{\delta}$ eine bessere Lösung des Ausgangsproblems als \tilde{x} .
- Falls nötig: Wiederholen.
- Lösen von (3.1) kostet nur $O(n^2)$ Operationen, da die Zerlegung der Matrix A wiederverwendet werden kann.

4 Kondition und Stabilität

Warum hatten wir Probleme mit der Matrix

$$\begin{pmatrix} 10^{-4} & 1 \\ 1 & 1 \end{pmatrix} \quad ?$$

Ein Teil der Antwort:

- Manche linearen Gleichungssysteme sind schwieriger als andere.
- manche Algorithmen sind anfälliger für Rundungsfehler als andere.

4.1 Die Kondition eines Problems

Abstrakt: Das Lösen von linearen Gleichungssystemen ist eine Abbildung

$$f : \mathbb{R}^{n \times n} \times \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad f(A, b) = A^{-1}b \quad (\text{Lösungsoperator})$$

Allgemein: Lösungsoperatoren sind Abbildungen

$$f : X \rightarrow Y.$$

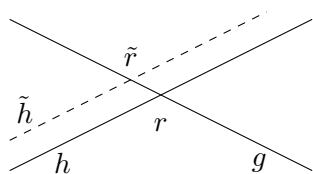
Algorithmus: Approximation $\tilde{f} : X \rightarrow Y$ eines Lösungsoperators:

$$\boxed{\text{Eingabedaten}} \rightarrow \boxed{\text{Algorithmus}} \rightarrow \boxed{\text{Ergebnis}}$$

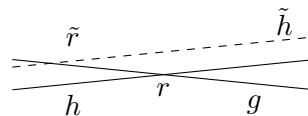
Zu einem fehlerhaften Ergebnis können verschiedene Effekte beitragen:

- 1) Fehler in den Eingabedaten
 - a) Messfehler
 - b) Fehler durch endliche Zahlendarstellung
- 2) Stabilität des Algorithmus
 - Verstärkt ein Algorithmus existierende Fehler oder dämpft er sie?
- 3) Kondition des Problems
 - Verstärkt das Problem selbst die Fehler?

Beispiel. Gegeben zwei Geraden g und h . Bestimme den Schnittpunkt r



gut konditioniert



schlecht konditioniert

Die Schwierigkeit (Kondition) des Problems hängt vom Winkel ab, in dem sich die Geraden schneiden:

- Stumpfer Winkel: Wird eine Gerade leicht gestört, so erfährt auch der Schnittpunkt eine leichte Störung.
- Spitzer Winkel: Wird eine Gerade leicht gestört, so ist der Schnittpunkt *stark* gestört.

Formal: Gegeben ein Eingabewert $x \in X$.

- x ist nur Repräsentant einer ganzen Menge E von möglichen Eingaben
- Absolute Fehler:

$$E = \{\tilde{x} \in X : \|\tilde{x} - x\| \leq \delta\}$$

z.B. Messfehler

- Relative Fehler:

$$E = \{\tilde{x} \in X : \|\tilde{x} - x\| \leq \epsilon \|x\|\}$$

Der Lösungsoperator f bildet E auf eine Menge $f(E)$ ab.

- Die Kondition eines Problems ist das „Verhältnis von $f(E)$ zu E “.

Quantitativ kann man (fast) nur rechnen, wenn die Fehler klein sind. Dann kann man eine linearisierte Theorie betreiben.

Asymptotische Lipschitz-Bedingung:

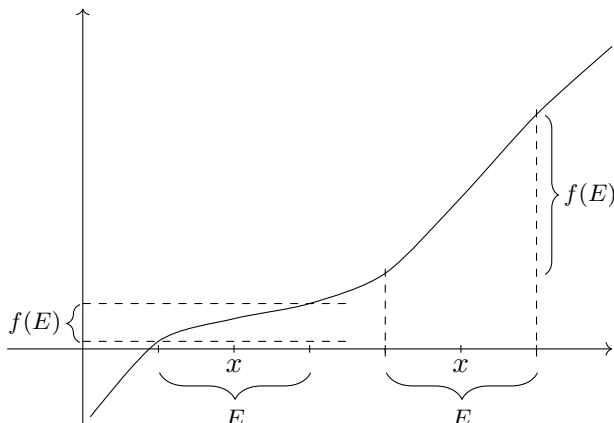
Definition. Die absolute Kondition des Problems f an der Stelle x ist die kleinste Zahl $\kappa_{abs} \geq 0$, so dass

$$\|f(\tilde{x}) - f(x)\| \leq \kappa_{abs} \|\tilde{x} - x\| + \varphi(\tilde{x})$$

mit $\frac{\varphi(\tilde{x})}{\|\tilde{x} - x\|} \rightarrow 0$ für $\tilde{x} \rightarrow x$.

- Ist f differenzierbar in x , so ist $\kappa_{abs} = \|f'(x)\|$. Dabei ist $f' \in \mathbb{R}^{n \times n}$ die Jacobi-Matrix, $\|f'\| = \sup_{x \neq 0} \frac{\|f'(x)\|}{\|x\|}$ die dazugehörige Matrix-Norm.
- f heißt *gut konditioniert*, falls κ_{abs} „klein“ ist.

- f heißt *schlecht konditioniert*, falls κ_{abs} „groß“ ist.



Definition. Die relative Kondition des Problems f an der Stelle x ist die kleinste Zahl $\kappa_{\text{rel}} \geq 0$, so dass

$$\frac{\|f(\tilde{x}) - f(x)\|}{\|f(x)\|} \leq \kappa_{\text{rel}} \frac{\|\tilde{x} - x\|}{\|x\|} + \varphi(\tilde{x})$$

mit einem φ wie oben.

Falls f differenzierbar ist, so gilt $\kappa_{\text{rel}} = \frac{\|x\|}{\|f(x)\|} \|f'(x)\|$.

4.1.1 Kondition von Addition und Subtraktion

Die Addition zweier Zahlen ist eine Abbildung

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad f : \begin{pmatrix} a \\ b \end{pmatrix} \mapsto a + b.$$

Als Norm auf \mathbb{R}^2 wählen wir die 1-Norm:

$$\|(a, b)\|_1 := |a| + |b|.$$

Damit ist

$$\begin{aligned} \|f'(a, b)\| &= \sup_{\substack{x \in \mathbb{R}^2 \\ x \neq 0}} \frac{\|f'(a, b) \cdot x\|_1}{\|x\|_1} = \sup \frac{\left| (1 \quad 1) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \right|}{|x_1| + |x_2|} \\ &= \sup \frac{|x_1 + x_2|}{|x_1| + |x_2|} = 1. \end{aligned}$$

Deshalb:

$$\kappa_{\text{abs}} = 1.$$

Relative Kondition:

$$\kappa_{\text{rel}} = \frac{\|(a, b)\|_1}{\|f(a, b)\|} \|f'\| = \frac{|a| + |b|}{|a + b|}$$

- $\kappa_{\text{rel}} = 1$ falls a, b gleiche Vorzeichen haben
- $\kappa_{\text{rel}} \gg 1$ falls a, b unterschiedliche Vorzeichen haben!

Dieses Phänomen heißt Auslöschung

Beispiel. $\epsilon = 10^{-7}$

$$\begin{aligned} a &= 0,123467\overbrace{xxxx}^{\text{unbekannt}} \\ b &= 0,123456xxxx \\ a - b &= 0,000011xxxx \end{aligned}$$

Eingabe: Fehler ab der 7. Stelle

Ausgabe: Fehler in der 3. Stelle

Merke: Subtraktion fast gleicher Zahlen ist zu vermeiden!

Auslöschung führt ab und zu zu praktisch relevanten Problemen.

Es gibt diverse Tricks, um Auslöschung zu vermeiden:

- Bei der Addition von Zahlenfolgen:
→ Sortiere die Zahlenfolge vor der Addition!
- Benutze Reihenentwicklung:
Beispiel: Für kleine x ersetze

$$\frac{1 - \cos(x)}{x} \quad \text{durch} \quad \frac{1}{x} \left(1 - \left(1 - \frac{x^2}{2} + \frac{x^4}{24} - \dots \right) \right) = \frac{x}{2} \left(1 - \frac{x^2}{12} + \dots \right)$$

4.1.2 Kondition der Multiplikation

Die Multiplikation als Abbildung ist

$$f : \mathbb{R}^2 \rightarrow \mathbb{R} \quad f(a, b) = a \cdot b.$$

Ableitung davon:

$$f'(a, b) = (b \ a)$$

Geeignete Norm: Die 2-Norm

$$\|(b \ a)\|_2 := \sqrt{a^2 + b^2}.$$

Eine dazu passende Matrixnorm: Frobeniusnorm (Ferdinand Georg Frobenius):

$$\|f'(a, b)\|_2 = \|(b \ a)\|_2 := \sqrt{a^2 + b^2}$$

Deshalb:

$$\begin{aligned}\kappa_{\text{abs}} &= \|f'(a, b)\|_2 = \sqrt{a^2 + b^2} \\ \kappa_{\text{rel}} &= \frac{\|(b \ a)\|_2}{\|f(a, b)\|} \|f'(a, b)\|_2 = \frac{\sqrt{a^2 + b^2}}{|a \cdot b|} \sqrt{b^2 + a^2} = \frac{a^2 + b^2}{|a \cdot b|}.\end{aligned}$$

Falls $a \approx b$, dann ist

$$\kappa_{\text{rel}} \approx \frac{2a^2}{|a^2|} = 2.$$

Die Multiplikation ist dann gut konditioniert.

Sind die Zahlen sehr unterschiedlich, zum Beispiel $a \approx 1$, b sehr klein (oder $a \approx 1$, b sehr groß)

$$\kappa_{\text{rel}} = \frac{1 + \text{klein}}{\text{klein}} \gg 1.$$

4.1.3 Kondition von linearen Gleichungssystemen

Betrachte das lineare Gleichungssystem

$$Ax = b.$$

Der einfache Fall: Sei b gestört, aber A ohne Fehler bekannt.

- Lösungsoperator: $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $f(b) = A^{-1}b$
- Linear
- Ableitung: $f'(b) = A^{-1}$
- Kondition:

$$\begin{aligned}\kappa_{\text{abs}} &= \|f'(b)\| = \|A^{-1}\| \\ \kappa_{\text{rel}} &= \frac{\|b\|}{\|f(b)\|} \|f'(b)\| = \frac{\|b\|}{\|A^{-1}b\|} \|A^{-1}\| = \frac{\|Ax\|}{\|x\|} \|A^{-1}\|.\end{aligned}$$

Weniger einfach: Sei b ohne Fehler bekannt, aber die Matrix gestört.

Lösungsoperator:

$$f : \text{GL}(n) \rightarrow \mathbb{R}^n, \quad f(A) = A^{-1}b$$

- nicht linear!
- Aber differenzierbar. Das folgt z.B. aus der Cramerschen Regel.

Wir rechnen die Ableitung aus:

Lemma 4.1 (Deuffhard und Hohmann, Lemma 2.8). *Die Abbildung*

$$g : GL(n) \rightarrow GL(n) \quad g(A) := A^{-1}$$

ist differenzierbar. Die Richtungsableitung in Richtung $C \in \mathbb{R}^{n \times n}$ ist

$$\left. \frac{dg(A + tC)}{dt} \right|_{t=0} = -A^{-1}CA^{-1}.$$

Beweis. • Es gilt

$$(A + tC)(A + tC)^{-1} = I, \quad \forall t \in (-\epsilon, \epsilon).$$

- Differenziere nach t :

$$C(A + tC)^{-1} + (A + tC) \frac{d}{dt}(A + tC)^{-1} = 0.$$

- An der Stelle $t = 0$:

$$CA^{-1} + A \frac{d}{dt}(A + tC)^{-1} \Big|_{t=0} = 0.$$

- Also

$$\frac{d}{dt}(A + tC)^{-1} \Big|_{t=0} = -A^{-1}CA^{-1}. \quad \square$$

Die Richtungsableitung ist linear in der Richtung C . Wir schreiben deshalb

$$g'(A)C := -A^{-1}CA^{-1}.$$

Für den Lösungsoperator $f : A \mapsto A^{-1}b$ gilt also

$$f'(A)C = -A^{-1}CA^{-1}b = -A^{-1}Cx, \quad \forall C \in \mathbb{R}^{n \times n},$$

und ($f'(A)$ ist eine Abbildung $\mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$)

$$\begin{aligned} \|f'(A)\| &= \sup_{C \neq 0} \frac{\|f'(A)C\|}{\|C\|} = \sup_{\|C\|=1} \|f'(A)C\| \\ &= \sup_{\|C\|=1} \|A^{-1}Cx\| \leq \sup_{\|C\|=1} \|A^{-1}\| \|C\| \|x\| \\ &= \|A^{-1}\| \|x\|. \end{aligned}$$

Also:

$$\kappa_{\text{abs}} = \|A^{-1}\| \|x\|$$

und

$$\kappa_{\text{rel}} = \frac{\|A\|}{\|x\|} \|f'(A)\| \leq \|A\| \|A^{-1}\|.$$

Definition. Die Zahl $\kappa(A) := \|A\| \|A^{-1}\|$ (auch: $\text{cond}(A)$) heißt *Kondition der Matrix A* .

- Bestimmt die Verstärkung von relativen Fehlern in A .
- Bestimmt auch die relative Verstärkung von Fehlern in b !

Erinnerung: mit $f : b \mapsto A^{-1}b$ war

$$\kappa_{\text{rel}} = \frac{\|Ax\|}{\|x\|} \|A^{-1}\| \leq \frac{\|A\| \|x\|}{\|x\|} \|A^{-1}\| = \kappa(A).$$

- Beeinflusst die Konvergenzgeschwindigkeit von iterativen Verfahren.

Alternative Darstellung:

$$\kappa(A) := \frac{\max_{\|x\|=1} \|Ax\|}{\min_{\|x\|=1} \|Ax\|} \quad (\text{Übung!})$$

Eigenschaften:

- $\kappa(A) \geq 1$.
- $\kappa(\alpha A) = \kappa(A), \quad \forall \alpha \in \mathbb{R}, \alpha \neq 0$.
- A ist singulär genau dann wenn $\kappa(A) = \infty$.
- Falls A symmetrisch ist und $\|\cdot\| = \|\cdot\|_2$:

$$\kappa(A) = \frac{|\text{betragsmäßig größter Eigenwert}|}{|\text{betragsmäßig kleinster Eigenwert}|}$$

4.2 Stabilität

Ein Algorithmus heißt numerisch instabil, wenn es Eingabedaten gibt, bei denen sich die Rundungsfehler während der Rechnung so akkumulieren, dass ein völlig verfälschtes Ergebnis entsteht.

Beispiel. Werte die Funktion $f(x) = \ln(x - \sqrt{x^2 - 1})$ an der Stelle $x = 30$ aus.

Tatsächliches Ergebnis: $f(30) = -4,094066 \dots$

- Kondition des Problems:

$$\begin{aligned} \kappa_{\text{abs}} = |f'(x)| &= \left| \frac{1}{x - \sqrt{x^2 - 1}} \left(1 - \frac{x}{\sqrt{x^2 - 1}} \right) \right| \\ &= \left| \frac{1}{x - \sqrt{x^2 - 1}} \cdot \frac{\sqrt{x^2 - 1} - x}{\sqrt{x^2 - 1}} \right| = \left| \frac{-1}{\sqrt{x^2 - 1}} \right|. \end{aligned}$$

- An der Stelle $x = 30$:

$$\kappa_{\text{abs}} = |f'(30)| \approx 0,033$$

Problem ist gut konditioniert!

- Sei der absolute Eingabefehler z.B. $\delta = |\hat{x} - x| = 0.05$.
 \Rightarrow Absoluter Ausgabefehler:

$$f(\hat{x}) - f(30) = |f(30 + \delta) - f(30)| \approx |f'(30)|\delta = 0,00165.$$

Das ist der absolute Ausgabefehler den wir erwarten würden.

Wir betrachten, wie ein Algorithmus die Formel auswertet.

- Zunächst ist $\sqrt{x^2 - 1}|_{x=30} = \sqrt{899} = 29,9833287\dots \approx 30$
 \Rightarrow Bei der Berechnung von $x - \sqrt{x^2 - 1}$ kommt es zu Auslöschung.
- Angenommen wir rechnen mit 4 Dezimalstellen:

$$(x - \sqrt{x^2 - 1})|_{x=30} = 30 - 29,98 = 0,02.$$

- Der tatsächliche Wert ist: $30 - 29,9833287\dots = 0,0166713\dots$
- Absoluter Fehler: $0,02 - 0,0166713\dots = 0,0033287\dots$
- Der Wert 0,02 wird jetzt in den Logarithmus gesteckt.
- Der absolute Fehler wird durch die Kondition des Logarithmus verstärkt:

$$\kappa_{\text{abs,log}} = |\ln(x)'| = \left| \frac{1}{x} \right|.$$

Bei $x = 0,02$ ist $\kappa_{\text{abs,log}} = 50$.

- In der Tat ist der Fehler des Gesamtergebnisses:

$$|f(\hat{x}) - f(30)| = |\ln(0,02) - \ln(0,0166713\dots)| = 0,1820436\dots$$

- Bei einer tatsächlichen Lösung von $-4,094066$ ist also schon die erste Nachkommastelle falsch!

4.2.1 Modifikation von Algorithmen

Manchmal können einfache Modifikationen die Stabilität verbessern.

Beispiel. Löse die quadratische Gleichung

$$x^2 - 2px + q = 0.$$

Lösungsoperator:

$$f(p, q) = p \pm \sqrt{p^2 - q}.$$

Interpretiere f als Algorithmus:

- Problematisch, falls eine Nullstelle nahe bei Null liegt.
- Dies ist genau dann der Fall, wenn q sehr klein ist.
 $\Rightarrow \sqrt{p^2 - q} \approx p$, es gibt Auslöschung bei der Subtraktion $p - \sqrt{\dots}$.

Alternative:

Satz 4.1 (Satz von Vieta). Seien x_1, x_2 die Nullstellen von $x^2 - 2px + q$. Dann ist $x_1 x_2 = q$.

Berechne deshalb x_1, x_2 durch

$$x_1 = p + \operatorname{sgn}(p)\sqrt{p^2 - q}, \quad x_2 = \frac{q}{x_1}.$$

4.2.2 Vorwärtsanalyse der Stabilität

Zur detaillierten Untersuchung der Stabilität eines Algorithmus fasst man diesen als Kette von elementaren Operationen auf.

- Für das Beispiel oben:
 1. $g_1 : \mathbb{R} \rightarrow \mathbb{R}^2, \quad x \mapsto (x, x^2)$
 2. $g_2 : \mathbb{R}^2 \rightarrow \mathbb{R}^2, \quad (y, z) \mapsto (y, z - 1)$
 3. $g_3 : \mathbb{R}^2 \rightarrow \mathbb{R}^2, \quad (s, t) \mapsto (s, \sqrt{t})$
 4. $g_4 : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad (u, v) \mapsto u - v$
 5. $g_5 : \mathbb{R} \rightarrow \mathbb{R}, \quad w \mapsto \ln w$

Insgesamt also $f(x) = (g_5 \circ g_4 \circ g_3 \circ g_2 \circ g_1)(x)$.

- So eine Darstellung wird natürlich schnell sehr umfangreich.
- Deshalb kann man auch größere Blöcke nehmen, z.B.
 1. $\hat{g}_1 : \mathbb{R} \rightarrow \mathbb{R}^2, \quad x \mapsto (x, \sqrt{x^2 - 1})$
 2. $\hat{g}_2 : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad (u, v) \mapsto u - v$
 3. $\hat{g}_3 : \mathbb{R} \rightarrow \mathbb{R}, \quad w \mapsto \ln w$

Sei $f = g_k \circ g_{k-1} \circ \dots \circ g_1$.

- Wir untersuchen Stabilität bzgl. des absoluten Fehlers.
- Seien $x = x^{(0)}$ die Eingabedaten, und

$$x^{(i)} = g_i(x^{(i-1)}) = g_i \circ g_{i-1} \circ \dots \circ g_1(x^{(0)})$$

die Zwischenergebnisse.

- Die tatsächliche Eingabedaten $\hat{x}^{(0)}$ sind mit einem absoluten Fehler behaftet:

$$\hat{x}^{(0)} = x^{(0)} + \delta^{(0)}$$

- Statt der Zwischenergebnisse $\hat{x}^{(i)}$ tauchen durch Rundungsfehler verfälschte Ergebnisse

$$\hat{x}^{(i)} = g_i(\hat{x}^{(i-1)}) + \delta^{(i)}$$

auf.

Fehlerverstärkung eines einzelnen Schritts $g_i : \mathbb{R}^{n_{i-1}} \rightarrow \mathbb{R}^{n_i}$

→ Kondition von g_i also $\kappa_i = \|g'_i\|$.

- Jeder Fehler $\delta^{(i)}$ wird durch die Ableitung der folgenden Schritte $h_i := g_k \circ g_{k-1} \circ \dots \circ g_{i+1}$ verstärkt.
- Wegen der Kettenregel

$$h'_i = (g_k \circ g_{k-1} \circ \dots \circ g_{i+1})' = g'_k \cdot g'_{k-1} \cdot \dots \cdot g'_{i+1}$$

- Also Einfluss des i-ten Fehlers auf das Resultat $h_i \delta^{(i)}$.

Fehler insgesamt:

$$\sum_{i=0}^k h'_i \delta^{(i)}.$$

Algorithmus ist instabil falls zumindest einige der $h_i \gg f'$ sind.

4.2.3 Stabilität der Gauß-Elimination

- Kompliziert; wir geben nur ein paar Ergebnisse.
- Gauß-Elimination ohne Pivot-Suche ist NICHT stabil.

Rückwärtsanalyse der Gauß-Elimination mit Pivot-Suche:

Rückwärtsanalyse: Sei der *Ausgabefehler* gegeben.

Das gestörte Resultat ist die *exakte* Lösung eines gestörten Problem:

$$(A + E)\hat{x} = b$$

(Wir betrachten nur Störung in A)

Ein Verfahren heißt stabil im Sinne der Rückwärtsanalyse, falls $\|E\|$ klein ist.

Für die Fehlermatrix E gilt

$$\|E\|_{\infty} \leq 3\rho_n n^3 \epsilon \|A\|_{\infty}.$$

Dabei ist:

- ϵ die Maschinengenauigkeit.
- ρ_n der sog. Wachstumsfaktor

$$\rho_n := \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|}.$$

Der Wachstumsfaktor Der Wachstumsfaktor ist die für die Stabilität relevante Größe. Wie verhält sich also ρ_n ?

- Falls A strikt diagonal dominant: $\rho_n \leq 2$.
- Falls A sym. pos. def. $\rho_n \leq 1$.

Allgemein: Für Gauß-Elimination mit Spalten-Pivotsuche: $\rho_n \leq 2^{n-1}$.

Diese Abschätzung ist scharf, denn:

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

$$A^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & -1 & 1 & 0 & 2 \\ 0 & -1 & -1 & 1 & 2 \\ 0 & -1 & -1 & -1 & 2 \end{pmatrix}$$

$$A^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & -1 & 1 & 4 \\ 0 & 0 & -1 & -1 & 4 \end{pmatrix}$$

$$A^{(4)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 & 8 \\ 0 & 0 & 0 & -1 & 8 \end{pmatrix}$$

etc., also

$$\rho_n = \frac{\max |a_{ij}^{(5)}|}{\max |a_{ij}|} = 16.$$

Statistisch sieht man $\rho_n \approx n^{\frac{2}{3}}$.

5 Numerische Lösung nichtlinearer Gleichungen

5.1 Fixpunktiterationen

Sei $f: \mathbb{R} \rightarrow \mathbb{R}$ eine Funktion. Wir interessieren uns für Lösungen x der Gleichung

$$f(x) = 0.$$

Idee der Fixpunktiteration:

- Forme diese Gleichung äquivalent in eine *Fixpunktgleichung*

$$g(x) = x$$

um.

- Konstruiere mit Hilfe der Iterationsvorschrift

$$x_{k+1} = g(x_k) \quad k = 0, 1, 2, \dots$$

für einen gegebenen Startwert x_0 eine Folge x_0, x_1, \dots

Wunsch: Die Folge (x_k) konvergiert gegen einen *Fixpunkt*, d.h. gegen einen Punkt x^* mit $g(x^*) = x^*$. Dieser ist dann auch Lösung der nichtlinearen Gleichung $f(x^*) = 0$.

Die Existenz von Fixpunkten folgt z.B. aus dem Fixpunktsatz von Banach.

Satz 5.1. Sei $I = [a, b] \subset \mathbb{R}$ ein Intervall und $g: I \rightarrow I$ eine kontrahierende Abbildung, d.h. g ist Lipschitz-stetig mit Lipschitz-Konstante $L < 1$. Dann folgt:

a) Es existiert genau ein Fixpunkt x^* von g .

b) Für jeden Startwert $x_0 \in I$ konvergiert die Fixpunktiteration $x_{k+1} = g(x_k)$ gegen x^* mit

$$|x_{k+1} - x_k| \leq L|x_k - x_{k-1}| \quad \text{und} \quad |x^* - x_k| \leq \frac{L^k}{1-L}|x_1 - x_0|.$$

Beweis. • Für alle $x_0 \in I$ gilt

$$|x_{k+1} - x_k| = |g(x_k) - g(x_{k-1})| \leq L|x_k - x_{k-1}|.$$

- Induktiv erhält man

$$|x_{k+1} - x_k| \leq L^k |x_1 - x_0|.$$

- Wir wollen zeigen, dass (x_k) eine Cauchy-Folge ist und betrachten

$$\begin{aligned} |x_{k+m} - x_k| &\leq |x_{k+m} - x_{k+m-1}| + \dots + |x_{k+1} - x_k| \\ &\leq \underbrace{\left(L^{k+m-1} + L^{k+m-2} + \dots + L^k \right)}_{=L^k(1+L+\dots+L^{m-1})} |x_1 - x_0| \\ &\leq \frac{L^k}{1-L} |x_1 - x_0|. \end{aligned}$$

Hierbei wurde verwendet, dass

$$\forall L \in (0, 1) \quad : \quad \sum_{k=0}^{\infty} L^k = \frac{1}{1-L}.$$

Damit ist gezeigt, dass (x_k) eine Cauchy-Folge ist.

- Diese konvergiert in \mathbb{R} gegen den Grenzwert

$$x^* := \lim_{k \rightarrow \infty} x_k.$$

- Der Punkt x^* ist aber auch Fixpunkt von g , da

$$\begin{aligned} |x^* - g(x^*)| &= |x^* - x_{k+1} + x_{k+1} - g(x^*)| \\ &= |x^* - x_{k+1} + g(x_k) - g(x^*)| \\ &\leq |x^* - x_{k+1}| + |g(x_k) - g(x^*)| \\ &\leq |x^* - x_{k+1}| + L|x_k - x^*| \\ &\rightarrow 0 \quad k \rightarrow \infty. \end{aligned}$$

Damit haben wir ??(b) und die Existenz des Fixpunktes gezeigt.

Für die Eindeutigkeit seien x^*, y^* zwei Fixpunkte. Dann gilt

$$0 \leq |x^* - y^*| = |g(x^*) - g(y^*)| \leq L|x^* - y^*| < |x^* - y^*|.$$

Da $L < 1$ ist dies nur für $|x^* - y^*| = 0$ möglich.

Daher ist der Fixpunkt von g eindeutig bestimmt. □

Beispiel. Betrachte die nichtlineare Gleichung

$$f(x) = x^2 - \ln(x) - 2 = 0 \quad \text{bzw.} \quad x^2 - 2 = \ln(x).$$

Die Gleichung hat in \mathbb{R} zwei Lösungen x_1^* und x_2^* .

Diese sind anscheinend in den Intervallen $I_1 = [0, 0,2]$ bzw. $I_2 = [1, 2]$ enthalten.

BILD!

Wie sieht eine geeignete Fixpunktiteration aus?

a)

$$x = x^2 + x - \ln(x) - 2 =: g_1(x).$$

Hinreichend für Lipschitz-Stetigkeit:

$$\max_{x \in \mathbb{R}} |g_1'(x)| = L < 1.$$

Differenziere g_1 und erhalte

$$g_1'(x) = 2x + 1 - \frac{1}{x} \implies L \geq 1$$

für alle $x \in I_1$ und $x \in I_2$.

Es gibt also keine Garantie für die Konvergenz gegen einen Fixpunkt.

b)

$$\ln(x) = x^2 - 2 \iff x = e^{x^2-2} =: g_2(x)$$

Ableitung: $g_2'(x) = 2xe^{x^2-2}$.

Es ist $|g_2'(x)| < 1$ in einer Umgebung von 0 \implies Konvergenz gegen x_1^* .

c)

$$x^2 = \ln(x) + 2 \iff x = \sqrt{\ln(x) + 2} =: g_3(x)$$

Ableitung:

$$g_3'(x) = \frac{1}{2x\sqrt{\ln(x) + 2}}$$

$|g_3'(x)| < 1$ in $[1, 2]$ \implies Konvergenz gegen x_2^* .

5.1.1 Konvergenzgeschwindigkeit

Definition. Eine gegen x^* konvergente Folge $(x_k)_{k \in \mathbb{N}}$ heißt linear konvergent, falls es eine Konstante $0 \leq C < 1$ gibt, so dass

$$\|x_{k+1} - x^*\| \leq C \|x_k - x^*\|.$$

Die Folge heißt quadratisch konvergent, falls es eine Konstante $C \geq 0$ gibt, so dass

$$\|x_{k+1} - x^*\| \leq C \|x_k - x^*\|^2.$$

Die Fixpunktiteration ist im Allgemeinen nur linear konvergent. Erstrebenswert wäre eine Iteration, die quadratisch konvergiert.

Kann in speziellen Fällen die Fixpunktiteration quadratisch konvergieren?

Satz 5.2. Die Funktion $g: \mathbb{R} \rightarrow \mathbb{R}$ besitze in $x^* \in I$ einen Fixpunkt. Auf der Menge $U = [x^* - r, x^* + r] \subset I$ sei $g \in C^2(U)$ mit $g'(x^*) = 0$. Dann konvergiert die Fixpunktiteration für jeden Startwert $x_0 \in U$ mit $|x_0 - x^*| \leq \varrho = \frac{1}{\max_{x \in U} |g''(x)|}$ quadratisch.

Beweis. Wir zeigen mittels vollständiger Induktion

$$|x_k - x^*| \leq \frac{1}{2^k} |x_0 - x^*|. \quad (5.1)$$

Induktionsanfang: $k = 0$ klar

Induktionsschritt:

$$\begin{aligned} |x_{k+1} - x^*| &= |g(x_k) - g(x^*)| \\ &\leq |g(x^*) + \underbrace{g'(x^*)}_{=0}(x_k - x^*) + \frac{g''(\xi)}{2}(x_k - x^*)^2 - g(x^*)| \\ &\quad \text{(Taylorentwicklung mit Lagrange-Restglied)} \\ &= \left| \frac{g''(\xi)}{2}(x_k - x^*)^2 \right| \\ &\leq \frac{M}{2} |x_k - x^*|^2 \quad \text{mit } M := \max_{x \in U} |g''(x)| = \frac{1}{\varrho}. \end{aligned} \quad (5.2)$$

Nun gilt: $|x_k - x^*| \leq |x_0 - x^*| \leq \varrho$, also

$$|x_{k+1} - x^*| \leq \frac{M}{2} \varrho \frac{1}{2^k} |x_0 - x^*| = \frac{1}{2^{k+1}} |x_0 - x^*|.$$

Also gilt (5.1) und es folgt Konvergenz. Aus (5.2) folgt Ordnung 2. \square

5.1.2 Das Newton-Verfahren als Fixpunktiteration

Wir benutzen den obigen Satz um das Newton-Verfahren zu konstruieren. Schreibe die folgende Fixpunktform:

$$f(x) = 0 \iff x = x - h(x)f(x) =: g(x)$$

mit noch zu bestimmender Funktion h .

Die Ableitung von g ist

$$g'(x) = 1 - h'(x)f(x) - h(x)f'(x).$$

Um Satz 5.2 anwenden zu können, muss gelten:

$$g'(x^*) = 0 = 1 - h(x^*)f'(x^*).$$

Idee: Wir wählen

$$h(x) = \frac{1}{f'(x)}.$$

Wir erhalten die Fixpunktiteration

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Dies ist das Newton-Verfahren.

5.1.3 Alternative Interpretation des Newton-Verfahrens

Sei $f: \mathbb{R} \rightarrow \mathbb{R}$ stetig differenzierbar. *Gesucht:* $x^* \in \mathbb{R}: f(x^*) = 0$.

Bild einer Funktion mit Tangente und nächster Iterierten

- Wähle einen Startpunkt x_0 .
- Für $k = 0, 1, 2, \dots$ approximiere f durch eine Tangente p_k in x_k .
- Anstelle der Nullstelle x^* von f berechne die Nullstelle der Tangente p_k . Wähle diese als x_{k+1} .

In Formeln:

- Die Tangente p_k von f in x_k hat die Darstellung $p_k(x) = f(x_k) + f'(x_k)(x - x_k)$.
- Die Nullstelle davon ist

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

(sofern $f'(x_k) \neq 0$).

Folgerung: Sei f hinreichend glatt und besitze eine Nullstelle x^* mit $f'(x^*) \neq 0$. Für hinreichend gute Startwerte x_0 konvergiert das Newton-Verfahren quadratisch gegen x^* .

Beispiel. Sei $f(x) = x^2 - 3$. Die Nullstellen sind $x^* = \pm\sqrt{3}$.

$f'(x) = 2x \implies$ Das Newton-Verfahren ist in der Nähe der Lösungen durchführbar.

Mit $x_0 = 1$ ergeben sich die folgenden 5 ersten Iterationen:

| | |
|---|------------|
| 0 | 1,00000000 |
| 1 | 2,00000000 |
| 2 | 1,75000000 |
| 3 | 1,73214286 |
| 4 | 1,73205081 |
| 5 | 1,73205081 |

Beispiel (Lokale Konvergenz). Sei

$$f(x) = \frac{x}{\sqrt{1 + c^3 x^2}} \quad \text{mit} \quad f(x) = 0 \Leftrightarrow x = 0.$$

Wie gut muss der Startwert sein, damit das Newton-Verfahren konvergiert?

$$f'(x) = \frac{1}{(1 + c^3 x^2)^{\frac{3}{2}}}$$

$$\frac{f(x)}{f'(x)} = x(1 + c^3 x^2) \quad \implies \quad g(x) = -c^3 x^3.$$

Das Newton-Verfahren dazu ist

$$x_{k+1} = -c^3(x_k)^3 \quad \text{bzw.} \quad x_k = (-1)^k c^{\frac{-3}{2}} (c^{\frac{3}{2}} x_0)^{3^k}.$$

(Herleitung: Vollständige Induktion!)

Konvergenz ist garantiert, falls $c^{\frac{3}{2}} x_0 < 1$ gilt.

Bild für $c = 2$ und das Verhalten der Iterierten

5.1.4 Mehrfache Nullstellen

Betrachte wieder eine Funktion $f: \mathbb{R} \rightarrow \mathbb{R}$ und $f(x^*) = 0$.

Sei $m \in \mathbb{N}_{\geq 2}$, x^* eine m -fache Nullstelle von f . Dann gilt

$$f^{(\nu)}(x^*) = 0 \quad \forall \nu = 0, \dots, m-1 \quad \text{and} \quad f^{(m)}(x^*) \neq 0.$$

Dann besitzen f und f' eine Darstellung der Form

$$\begin{aligned} f(x) &= (x - x^*)^m h(x) \\ f'(x) &= m(x - x^*)^{m-1} h(x) + (x - x^*)^m h'(x) \end{aligned}$$

mit einer differenzierbaren Funktion h , welche $h(x^*) \neq 0$ erfüllt.

Es folgt nun

$$\begin{aligned} g(x) &= x - \frac{f(x)}{f'(x)} = x - \frac{(x - x^*)^m h(x)}{m(x - x^*)^{m-1} h(x) + (x - x^*)^m h'(x)} \\ &= x - \frac{(x - x^*) h(x)}{m h(x) + (x - x^*) h'(x)} \\ &\implies g'(x^*) = 1 - \frac{1}{m}. \end{aligned}$$

Folgerung: Für mehrfache Nullstellen ist das Verfahren zwar noch konvergent, aber nicht mehr quadratisch konvergent, denn nach Satz 5.2 erhält man quadratische Konvergenz nur wenn $g'(x^*) = 0$.

5.2 Das Newton-Verfahren für Systeme von Gleichungen

Die Verallgemeinerung des Newton-Verfahrens auf Systeme von Gleichungen ist relativ einfach.

Sei $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ stetig differenzierbar.

Gesucht: ein $x^* \in \mathbb{R}^n$ mit $F(x^*) = 0$.

Formel für x_{k+1} (Nullstelle der Tangente in x_k)

$$x_{k+1} := x_k - F'(x_k)^{-1} F(x_k).$$

Da das Invertieren der Matrix $F(x_k)$ sehr teuer ist, schreibt man den Iterationsschritt stattdessen als lineares Gleichungssystem für die Newton-Korrektur $\Delta x_k \in \mathbb{R}^n$:

$$\begin{aligned} F'(x^k)\Delta x^k &= -F(x^k) \\ x^{k+1} &= x^k + \Delta x^k \end{aligned}$$

Wir haben gesehen:

- konvergiert manchmal, manchmal auch nicht
- Wenn es konvergiert, dann konvergiert es *quadratisch* (d.h. schnell!),

Wir zeigen jetzt einen alternativen Beweis der quadratischen Konvergenz des Newton-Verfahrens.

Wann konvergiert das Verfahren gut?

- Falls F affin-linear ist: Dann findet es die Lösung in einem Schritt.
- Vermutlich: Verfahren konvergiert, falls F „fast affin-linear“ ist.

Was bedeutet nun „fast affin-linear“?

- F'' klein, bzw.
- $F' : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ Lipschitz-stetig, d.h., es existiert ein $L > 0$ mit

$$\|F'(x) - F'(y)\| \leq L\|x - y\| \quad \forall x, y.$$

Satz 5.3 (Fischer-Skript 5.2). *Sei $D \subseteq \mathbb{R}^n$ offen und konvex, und $F : D \rightarrow \mathbb{R}^n$ stetig differenzierbar mit invertierbarer Jacobi-Matrix $F'(x)$ für alle $x \in D$. Sei F' Lipschitz-stetig in D mit Lipschitz-Konstante L . Sei $x^* \in D$ Nullstelle von F .*

- Dann existiert eine offene Kugel $B_\varrho(x^*) := \{x \in \mathbb{R}^n : \|x - x^*\| < \varrho\} \subset D$, sodass das Newton-Verfahren für jede Startiterierte $x^0 \in B_\varrho(x^*)$ wohldefiniert ist.*
- Falls $x_0 \in B_\varrho(x^*)$ und ϱ hinreichend klein ist konvergiert die Folge (x^k) quadratisch gegen x^* .*

Für den Beweis brauchen wir folgende Variante der Taylor-Formel:

Lemma 5.1. *Sei $F \in C^1$. Dann gilt $\forall x, y$:*

$$F(y) = F(x) + F'(x)(y - x) + \int_0^1 (F'(x + s(y - x)) - F'(x))(y - x) ds$$

Beweis. Hauptsatz der Integralrechnung, sowie die Substitutionsregel liefern:

$$f(y) = f(x) + \int_x^y f'(t) dt = f(x) + \int_0^1 f'(x + s(y-x))(y-x) ds.$$

Für $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ gilt

$$F(y) = F(x) + \int_0^1 F'(x + s(y-x))(y-x) ds.$$

Addiere zur rechten Seite dieser Gleichung $F'(x)(y-x) - F'(x)(y-x) = 0$.

Der Übergang zum vektorwertigen F ist nicht sehr hübsch. Sauberer wäre es, f als Restriktion von F auf dem Streckensegment von x nach y einzuführen.

□

Beweis von Satz 5.3.

- F' ist stetig in D , d.h. $\exists \varrho_1 > 0$ und $M \geq 0$ mit $B_{\varrho_1}(x^*) \subset D$ sowie

$$\|F'(x)^{-1}\| \leq M \quad \forall x \in B_{\varrho_1}(x^*).$$

- Also gilt für beliebiges $x^k \in B_{\varrho_1}(x^*)$

$$\begin{aligned} \|x^{k+1} - x^*\| &= \|x^k - F'(x^k)^{-1}F(x^k) - x^*\| \\ &= \left\| -F'(x^k)^{-1} \left[F(x^k) + F'(x^k)(x^* - x^k) \right] \right\| \\ &\leq M \left\| F(x^k) + F'(x^k)(x^* - x^k) \right\| \\ &= M \left\| F(x^*) - F(x^k) - F'(x^k)(x^* - x^k) \right\| \\ &\leq \frac{1}{2}ML \|x^* - x^k\|^2. \end{aligned}$$

- Taylor-Formel aus Lemma 5.1: $\forall x, y \in D$ gilt

$$F(x) = F(y) + F'(y)(x-y) + \int_0^1 \left[F'(y + s(x-y)) - F'(y) \right] (x-y) ds.$$

- Daraus folgt für alle $x, y \in D$:

$$\begin{aligned}
 M \left\| F(x^*) - F(x^k) - F'(x^k)(x^* - x^k) \right\| &= M \left\| \int_0^1 \left[F'(x^k + s(x^* - x^k)) - F'(x^k) \right] (x^* - x^k) ds \right\| \\
 &\leq M \int_0^1 \| \dots \| ds \\
 &\leq M \int_0^1 \| F'(x^k + s(x^* - x^k)) - F'(x^k) \| ds \cdot \| x^* - x^k \| \\
 &\leq M \int_0^1 L \| (x^k + s(x^* - x^k)) - x^k \| ds \cdot \| x^* - x^k \| \\
 &= ML \int_0^1 s ds \cdot \| x^* - x^k \|^2 \\
 &= \frac{1}{2} ML \| x^* - x^k \|^2.
 \end{aligned}$$

- Sei $\varrho \in (0, \varrho_1]$: $ML\varrho \leq 1$. Dann gilt für alle $x^k \in B_\varrho(x^*)$

$$\| x^{k+1} - x^* \| \leq \frac{1}{2} ML \| x^k - x^* \|^2 \leq \frac{1}{2} ML \underbrace{\| x^k - x^* \|^2}_{\leq \varrho} \cdot \| x^k - x^* \| \leq \frac{1}{2} \| x^k - x^* \|.$$

Somit konvergiert die Folge (x_k) gegen x^* , und zwar quadratisch. \square

5.3 Affin-Invarianz

- Es gibt viele Varianten des Satzes, dass das Newton-Verfahren lokal quadratisch konvergiert.
- Peter Deuffhard hat folgendes Kriterium vorgeschlagen, das solche Sätze erfüllen sollten.

Sei $A \in \mathbb{R}^{n \times n}$ invertierbar. Dann ist

$$F(x) = 0$$

äquivalent zu

$$G(x) := AF(x) = 0.$$

(Dann heißt das Problem $F(x) = 0$ affin-invariant.)

Auch das Newton-Verfahren ist affin-invariant, denn

$$\begin{aligned}
 -\Delta x^k &= F'(x^k)^{-1} F(x^k) = F'(x^k)^{-1} A^{-1} AF(x^k) \\
 &= (AF'(x^k))^{-1} \cdot AF(x^k) = G'(x^k)^{-1} \cdot G(x^k).
 \end{aligned}$$

\implies Die vom Newton-Verfahren erzeugte Folge $(x^k)_{k \in \mathbb{N}}$ ist unabhängig von der Matrix A .

Wir verlangen, dass auch die Konvergenzresultate unabhängig von A sein sollten.

Satz 5.4 (Deuffhard und Hohmann [4], Satz 4.10). Sei $D \subset \mathbb{R}^n$ offen und konvex.

- Sei $F \in C^1(D, \mathbb{R}^n)$, und so dass $F'(x)^{-1}$ für alle $x \in D$ existiert.
- Für ein $\omega > 0$ gelte die Lipschitz-Bedingung

$$\|F'(x)^{-1}(F'(x + s(y - x)) - F'(x))(y - x)\| \leq s\omega \|y - x\|^2$$

für alle $s \in [0, 1]$ und $x, y \in D$.

- Es existiere eine Lösung $x^* \in D$ (also $F(x^*) = 0$) und ein Startwert $x^0 \in D$ derart, dass

$$\varrho := \|x^* - x^0\| < \frac{2}{\omega} \quad \text{und} \quad B_\varrho(x^*) \subseteq D$$

Dann gilt:

- Die durch die Newton-Iteration definierte Folge (x^k) bleibt in der offenen Kugel $B_\varrho(x^*)$, und konvergiert gegen x^* .
- Konvergenz ist quadratisch; genauer

$$\forall k \in \mathbb{N}: \quad \|x^{k+1} - x^*\| \leq \frac{\omega}{2} \|x^k - x^*\|^2$$

Beweis des Satzes.

Beweis von (ii)

$$\begin{aligned} x^{k+1} - x^* &= x^k - F'(x^k)^{-1}F(x^k) - x^* \\ &= x^k - x^* - F'(x^k)^{-1}\left(F(x^k) - \underbrace{F(x^*)}_{=0}\right) \\ &= F'(x^k)^{-1}F'(x^k)(x^k - x^*) - F'(x^k)^{-1}(F(x^k) - F(x^*)) \\ &= F'(x^k)^{-1}\left(F(x^*) - F(x^k) - F'(x^k)(x^* - x^k)\right) \end{aligned}$$

Aus Lemma 5.1 wissen wir

$$F(y) - F(x) - F'(x)(y - x) = \int_0^1 (F'(x + s(y - x)) - F'(x))(y - x) ds.$$

Damit folgt

$$\begin{aligned} \|F'(x)^{-1}(F(y) - F(x) - F'(x)(y - x))\| &= \left\| \int_0^1 F'(x)^{-1}F'(x + s(y - x)) - F'(x)(y - x) ds \right\| \\ &\leq \int_0^1 s\omega \|y - x\|^2 ds \quad \text{nach Voraussetzung} \\ &= \frac{\omega}{2} \|y - x\|^2. \end{aligned}$$

Das wenden wir an und erhalten

$$\|x^{k+1} - x^*\| \leq \frac{\omega}{2} \|x^k - x^*\|^2.$$

Das heißt, falls (x^k) konvergiert, so konvergiert es quadratisch.

Beweis von (i) Falls $\|x^k - x^*\| \leq \varrho$, so folgt

$$\|x^{k+1} - x^*\| \leq \underbrace{\frac{\omega}{2}}_{\leq \varrho \frac{\omega}{2} < 1} \|x^k - x^*\| \cdot \|x^k - x^*\|.$$

Da $\|x^0 - x^*\| = \varrho$ gilt $\|x^k - x^*\| < \varrho$ für alle $k > 0$, und (x^k) konvergiert gegen x^* . \square

5.4 Konvergenzkriterien

- Die Voraussetzungen des vorherigen Satzes können nicht algorithmisch geprüft werden.
- Trotzdem möchte man gern während der Durchführung der Newton-Methode wissen, ob das Verfahren konvergiert.

5.4.1 Der Monotonietest

Betrachte das Residuum $F(x^k)$.

\implies Das Lösen von $F(x) = 0$ ist äquivalent zum Minimieren von $\|F(x)\|$.

Wir vermuten/hoffen: Falls (x^k) konvergiert, dann ist $\|F(x^k)\|$ eine monoton fallende Folge.

Monotonietest: Für ein $\theta < 1$ prüfe nach jedem Schritt, ob

$$\|F(x^{k+1})\| \leq \theta \|F(x^k)\|.$$

Breche das Verfahren ab, wenn die Bedingung nicht erfüllt ist.

Problem: Dieses Verfahren ist nicht affin-invariant.

Beispiel. Betrachte zwei Iterierte $x^k, x^{k+1} \in \mathbb{R}^2$ so dass

$$F(x^{k+1}) = \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix}, \quad F(x^k) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Es gilt

$$\left\| \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} \right\|_2 \leq \left\| \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\|_2,$$

der Monotonietest ist also erfüllt.

Für ein $\varepsilon > 0$ wähle nun $A \in \mathbb{R}^{2 \times 2}$ als

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

Dann ist

$$\begin{aligned} \|AF(x^{k+1})\|_2 &\leq \|AF(x^k)\|_2 \\ \iff \left\| \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon \end{pmatrix} \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} \right\|_2 &\leq \left\| \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\|_2 \\ \iff \left\| \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} \right\|_2 &\leq \left\| \begin{pmatrix} 0 \\ \varepsilon \end{pmatrix} \right\|_2. \end{aligned}$$

Der Monotonietest ist also nicht erfüllt, falls $\varepsilon > \frac{1}{2}$. Und das obwohl sich das Verfahren nicht geändert hat!

5.4.2 Der natürliche Monotonietest

Deuffhard hat stattdessen den folgenden affin-invarianten Test vorgeschlagen:

$$\|F'(x^k)^{-1}F(x^{k+1})\| \leq \theta \|F'(x^k)^{-1}F(x^k)\|.$$

Wie berechnet man diese Terme?

1. Rechte Seite

$$F'(x^k)^{-1}F(x^k) = \Delta x^k.$$

Die ist die Newton-Korrektur, die wir ohnehin berechnen müssen.

2. Linke Seite

$$F'(x^k)^{-1}F(x^{k+1}) = \overline{\Delta x^{k+1}}$$

ist die Lösung eines weiteren linearen Gleichungssystems, was teuer werden kann.

- ABER: Die Matrix $F'(x^k)$ ist die gleiche wie bei 1).
- Deshalb ist die LR-Zerlegung bereits bekannt.
- Es sind nur Vorwärts- und Rückwärtssubstitution nötig, der Aufwand ist damit $\mathcal{O}(n^2)$.

5.5 Newton-Verfahren mit Dämpfung

- Kann man das Verfahren so erweitern, dass es für alle (oder zumindest mehr) Startwerte konvergiert?
- Solch eine Erweiterung nennt man *Globalisierung*.
- Gleichzeitig möchte man die schnelle quadratische Konvergenz behalten.

Idee. Wir messen die „Güte“ einer Approximation x^k von x^* durch eine skalare Funktion, hier

$$\phi(x) := \frac{1}{2} \|F(x)\|_2^2.$$

Wie wirkt das Newton-Verfahrens auf ϕ ?

Lemma 5.2. Die Newton-Korrektur $\Delta x := -(F'(x))^{-1}F(x)$ ist eine Abstiegsrichtung für ϕ , d.h.

$$\phi(x + t\Delta x) < \phi(x)$$

für $t > 0$ klein genug.

Beweis. Zunächst ist

$$\phi'(x) = F(x)^T F'(x).$$

Daraus folgt

$$\begin{aligned} \phi'(x)\Delta x &= F(x)^T F'(x)\Delta x \\ &= -F(x)^T F'(x)F'(x)^{-1}F(x) \\ &= -F(x)^T F(x) = -2\phi(x) < 0, \end{aligned}$$

falls x nicht Lösung von $F(x) = 0$ ist.

Damit berechnen wir

$$\begin{aligned} \phi(x + t\Delta x) &= \phi(x) + \phi'(x) \cdot t\Delta x + o(t) && \text{(Taylor)} \\ &= \phi(x) - 2t\phi(x) + o(t) \\ &= (1 - 2t)\phi(x) + o(t). \end{aligned}$$

Präzisieren, wie daraus die Behauptung folgt. Begründung: $o(t)$ steht für eine Funktion $\theta(t)$, mit $\lim_{t \rightarrow 0} \theta/t = 0$. Das heißt aber dass für jedes $\epsilon > 0$ ein T existiert so dass $|\theta(t)| < \epsilon t$ für alle $t < T$.

□

Idee. Wähle anstelle der Korrektur Δx^k die Korrektur $t_k \Delta x^k$. Dabei sei $t_k \in (0, 1)$ so gewählt, dass „hinreichender Abstieg“ von ϕ erzeugt wird.

Was heißt „hinreichender Abstieg“?

- Angenommen, $\phi(x^k)$ sei streng monoton fallend.
- Dann konvergiert diese Folge (da ϕ von unten beschränkt ist).
- Aber sie konvergiert nicht notwendigerweise gegen 0.
- Sie konvergiert nur dann gegen Null, wenn der Abstieg in jedem Schritt groß genug ist.

Die Armijo-Schrittweitenregel (nach Larry Armijo):

Sei $q \in (0, 1)$ ein Parameter. Wähle t_k als das größte Element aus

$$\left\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots, \frac{1}{2^n}, \dots\right\},$$

für welches

$$\phi(x^k + t_k \Delta x^k) \leq (1 - qt) \phi(x^k).$$

So ein t_k existiert wegen Lemma 5.2.

Achtung, das folgt nicht einfach aus der Aussage! Man muss schon in den Beweis schauen. Am besten man baut das noch in den Beweis für den folgenden Satz ein.

Satz 5.5 (Fischer-Skript, Satz 5.3). *Es sei $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ stetig differenzierbar. Sei F' lokal Lipschitz-stetig, und $F'(x)$ regulär für alle x aus*

$$W(x^0) := \{x \in \mathbb{R}^n : \phi(x) \leq \phi(x^0)\}.$$

Dann ist das Newton-Verfahren mit der Armijo-Dämpfung wohldefiniert, und es gilt $x^k \in W(x^0)$ für alle $k \in \mathbb{N}$.

Beweis. Sei $x^k \in W(x^0)$.

- Nach Voraussetzung ist $F'(x^k)$ regulär, daher ist Δx^k wohldefiniert.
- Nach Konstruktion ist auch t_k wohldefiniert und

$$\phi(x^{k+1}) < \phi(x^k) \implies x^{k+1} \in W(x^0). \quad \square$$

Satz 5.6 (Fischer-Skript, Satz 5.3). *Voraussetzungen wie eben. Falls die Folge (x^k) eine Teilfolge besitzt, die gegen ein \tilde{x} konvergiert, so gilt $F(\tilde{x}) = 0$.*

So eine Teilfolge gibt es insbesondere dann, wenn $W(x^0)$ beschränkt ist.

Beweis. (1) Wir brauchen eine obere Schranke für $\|\Delta x^k\|$.

- F und F' sind stetig in \mathbb{R}^n .
 - Da $\tilde{x} \in W(x_0)$ ist $F'(\tilde{x})$ regulär.
 - Wähle $\varrho > 0$ so klein, dass $B_\varrho(\tilde{x}) \subset W(x_0)$.
 - Dann ist $x \mapsto \|\Delta x\| = \|F'(x)^{-1}F(x)\|$ stetig in $B_\varrho(\tilde{x})$.
- $\implies \exists c > 0$ mit $\|\Delta x\| \leq c$ für alle $x \in B_\varrho(\tilde{x})$.

(2) Da F' Lipschitz-stetig ist, ist auch $\phi' = F^T F'$ Lipschitz-stetig.

D.h. Es gibt ein $L > 0$ so dass

$$\|\phi'(x) - \phi'(y)\| \leq L\|x - y\| \quad \forall x, y \in B_\varrho(\tilde{x}).$$

(3) Nach Voraussetzung existiert eine Teilfolge (x^{k_i}) , die gegen \tilde{x} konvergiert. Bezeichne diese Teilfolge wieder als (x^k) .

Erklären wofür das das hier braucht:

- Konvergenz gegen \tilde{x} heißt insbesondere: $\exists k_0 \in \mathbb{N}$ so dass $x^k \in B_\varrho(\tilde{x})$ für alle $k > k_0$.
- Da x^k gegen den Kugelmittelpunkt konvergiert, bildet sich ein endlicher Abstand zum Kugelrand, also

$$\exists \bar{t} > 0 \text{ und } k_0 \in \mathbb{N} \text{ so dass } x^k + t\Delta x^k \in B_\varrho(\tilde{x}) \quad \forall k \geq k_0 \text{ und } \forall t \in [0, \bar{t}].$$

Jetzt benutzen wir wieder die spezielle Taylorformel aus Lemma 5.1, diesmal für die Funktion ϕ :

$$\phi(y) = \phi(x) + \phi'(x)(y-x) + \int_0^1 (\phi'(x+s(y-x)) - \phi'(x))(y-x) ds$$

Für $y = x + t\Delta x$ ist dann

$$\phi(x + t\Delta x) = \phi(x) + \phi'(x)(t\Delta x) + \int_0^1 (\phi'(x + st\Delta x) - \phi'(x))t\Delta x ds.$$

Wegen $\phi'(x)\Delta x = -2\phi(x)$ folgt

$$\begin{aligned} \phi(x + t\Delta x) &= (1 - 2t)\phi(x) + \int_0^1 \dots ds \\ &\leq (1 - 2t)\phi(x) + \left| \int_0^1 \dots ds \right| \\ &\leq (1 - 2t)\phi(x) + \int_0^1 \|\phi'(x + st\Delta x) - \phi'(x)\| ds \cdot t \underbrace{\|\Delta x\|}_{\leq c} \\ &\leq (1 - 2t)\phi(x) + \max_s \|\phi'(x + st\Delta x) - \phi'(x)\| \cdot t \cdot c \\ &\leq (1 - 2t)\phi(x) + \max_s L \|x + st\Delta x - x\| \cdot t \cdot c \\ &\leq (1 - 2t)\phi(x) + L \cdot t \|\Delta x\| \cdot t \cdot c \\ &\leq (1 - 2t)\phi(x) + Lt^2 c^2. \end{aligned} \tag{5.3}$$

Wir wollen zeigen, dass die t_k von 0 weg beschränkt sind.

Armijo: Wähle t_k als größtes t der Form $k_k = 2^{-n}$, $n \in \mathbb{N}_0$, für dass

$$\phi(x^k + t_k \Delta x^k) \leq (1 - qt_k)\phi(x^k).$$

Da t_k die größte Zweierpotenz ist, für die diese Bedingung gilt, gilt sie für die nächstgrößere Potenz $2t_k$ nicht mehr:

$$(1 - 2qt_k)\phi(x^k) < \phi(x^k + 2t_k \Delta x^k).$$

Mit (5.3) folgt

$$(1 - 2qt_k)\phi(x^k) < (1 - 4t_k)\phi(x) + 4Lt_k^2c^2.$$

Auflösen nach t_k :

$$\begin{aligned} -2qt_k\phi(x^k) &\leq -4t_k\phi(x^k) + 4Lt_k^2c^2 \\ -q\phi(x^k) &\leq -2\phi(x^k) + 2Lt_kc^2 \\ \frac{(2-q)\phi(x^k)}{2Lc^2} &\leq t_k. \end{aligned}$$

Das heißt noch nicht, dass t_k von 0 weg beschränkt ist, da $\phi(x^k) \rightarrow 0$ sein könnte. (Genau das wollen wir ja sogar beweisen, aber noch ist es nur eine Möglichkeit.)

Stattdessen: Angenommen $\phi(\tilde{x}) > 0$. Wg. $x^k \rightarrow \tilde{x}$ und Stetigkeit von ϕ gilt

$$\phi(x^k) \geq \frac{1}{2}\phi(\tilde{x})$$

für unendlich viele k .

Für die dazugehörigen t_k gilt

$$t_k \geq \hat{t} := \frac{(2-q)\phi(\tilde{x})}{4Lc^2} > 0.$$

Aber $(\phi(x^k))$ ist monoton fallend

$$\phi(x^{k+1}) = \phi(x^k + t_k\Delta x^k) < (1 - qt_k)\phi(x^k) \leq (1 - q\hat{t})\phi(x^k),$$

also $\phi(x^{k+1}) < \alpha\phi(x^k)$ für ein $\alpha < 1$. Diese Ungleichung gilt für unendlich viele Indizes k . Daraus folgt $\lim_{k \rightarrow \infty} \phi(x^k) = 0$.

Stetigkeit von ϕ liefert

$$0 = \lim_{k \rightarrow \infty} \phi(x^k) = \phi(\lim_{k \rightarrow \infty} x^k) = \phi(\tilde{x}). \quad \square$$

Satz 5.7 (Fischer-Skript, Satz 5.3). *Sei $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ stetig differenzierbar. Weiter sei F' lokal Lipschitz-stetig mit Konstante L_0 und für alle $x \in W(x_0) := \{x \in \mathbb{R}^n : \phi(x) < \phi(x_0)\}$ sei $F'(x_0)$ invertierbar. Dann gibt es ein $k_0 \in \mathbb{N}$, sodass $t_k = 1$ für alle $k \geq k_0$.*

Die Dämpfung schaltet sich also irgendwann automatisch ab.

Korollar. Das gedämpfte Verfahren konvergiert lokal quadratisch.

Beweis. • Für $k \geq k_0$ degeneriert das Verfahren zum normalen Newton-Verfahren.

- Da zumindest eine Teilfolge konvergiert, erhält man früher oder später ein x^k , das so nah an x^* liegt, dass der lokale Konvergenzsatz greift. \square

Beweis von Satz 5.7. Man verwendet wieder die Taylor-Formel.

- Sei ϱ so klein, dass $B_\varrho(x^*) \subset W(x_0)$.
- Dann ist F' regulär für alle $x \in B_\varrho(x^*)$ und es existiert ein $M > 0$, sodass $\|F'(x)^{-1}\| \leq M$ für alle $x \in B_\varrho(x^*)$.
- Da (x^k) gegen x^* konvergiert gibt es ein $N \in \mathbb{N}$, sodass $x^k \in B_\varrho(x^*)$ für alle $k \geq N$.
- Für solche k gilt für jedes $s \in [0, 1]$

$$\begin{aligned} \|F'(x^k + s\Delta x^k) - F'(x^k)\| &\leq L_0 \|s\Delta x^k\| \leq L_0 \|\Delta x^k\| \\ &= L_0 \|F'(x^k)^{-1} F(x^k)\| \leq L_0 M \|F(x^k)\|. \end{aligned}$$

- Wieder die Taylor-Formel mit Integralrestglied:

$$\begin{aligned} \|F(x^k + \Delta x^k)\| &= \left\| \underbrace{F(x^k) + F'(x^k)\Delta x^k}_{= 0, \text{ nach Def. von } \Delta x^k} + \int_0^1 [F'(x^k + s\Delta x^k) - F'(x^k)] \Delta x^k ds \right\| \\ &\leq \max_{s \in [0,1]} \underbrace{\|F'(x^k + s\Delta x^k) - F'(x^k)\|}_{\leq L_0 M \|F(x^k)\|} \cdot \underbrace{\|\Delta x^k\|}_{\leq M \|F(x^k)\|} \\ &\leq L_0 M^2 \|F(x^k)\|^2. \end{aligned}$$

- Wähle $\varrho > 0$ so klein, dass

$$\|F(x)\| \leq L_0^{-1} M^{-2} \sqrt{1 - q} \quad \forall x \in B_\varrho(x^*).$$

- Dann ist für alle k groß genug

$$\|F(x^k + \Delta x^k)\| \leq L_0 M^2 \|F(x^k)\|^2 \leq \sqrt{1 - q} \|F(x^k)\|,$$

also

$$\begin{aligned} \phi(x^k + \Delta x^k) &= \frac{1}{2} \|F(x^k + \Delta x^k)\|^2 \leq \frac{1}{2} (1 - q) \|F(x^k)\|^2 \\ &= (1 - q) \phi(x^k). \end{aligned}$$

- Das Armijo-Kriterium ist also mit $t_k = 1$ erfüllt. □

6 Nichtlineare Ausgleichsprobleme

Bisher haben wir Vektoren $x^* \in \mathbb{R}^n$ gesucht, sodass

$$F(x^*) = 0 \quad \text{bzw.} \quad \|F(x^*)\| = 0.$$

Wir verallgemeinern nun diese Situation.

Gegeben: m Messwerte $b_1, \dots, b_m \in \mathbb{R}$ zu Zeitpunkten $t_1, \dots, t_m \in \mathbb{R}$.

Gesucht: Funktion $\varphi: \mathbb{R} \rightarrow \mathbb{R}$, die die Daten „möglichst gut approximiert“.

Wir kennen schon:

1. Polynominterpolation: Es existiert ein Polynom vom Grad höchstens $m - 1$, welches die Daten interpoliert.
Dies funktioniert aber schlecht, wenn m groß ist.
Das ist aber gerade der interessante Fall.
2. Spline-Interpolation: Ja, *aber*: häufig vermuten wir schon eine Gesetzmäßigkeit und wollen eigentlich nur ein paar Parameter bestimmen.

Beispiel (Normalverteilung).

$$\varphi(t; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t - \mu)^2}{2\sigma^2}\right)$$

mit nur zwei Parametern: dem Erwartungswert μ und der Standardabweichung σ .

Das Problem ist hier: Finde $\mu, \sigma \in \mathbb{R}$, so dass $\varphi(t_i; \mu, \sigma) = b_i$ für alle $i = 1, \dots, m$.

Abstrakt: Gegeben eine Modellfunktion mit n reellen Parametern x_1, \dots, x_n

$$\varphi(t; x_1, x_2, \dots, x_n)$$

Falls die Messwerte die Gesetzmäßigkeit exakt erfüllen, dann gibt es Parameter x_1, \dots, x_n so dass

$$b_i = \varphi(t_i; x_1, \dots, x_n) \quad \forall i = 1, \dots, m$$

Normalerweise ist aber $m \gg n$.

Deswegen kann man nur

$$b_i \approx \varphi(t_i; x_1, \dots, x_n) \quad \forall i = 1, \dots, m$$

erwarten.

Betrachte die Differenzen

$$\Delta_i := b_i - \varphi(t_i; x_1, \dots, x_n) \quad \forall i = 1, \dots, m.$$

Diese sollen „irgendwie klein“ werden.

6.1 Prinzip der kleinsten Quadrate

Es ist sinnvoll, die $x_1, \dots, x_n \in \mathbb{R}$ so zu wählen, dass das Fehlerfunktional

$$\Delta^2 := \sum_{i=1}^m \Delta_i^2 = \sum_{i=1}^m (b_i - \varphi(t_i; x_1, \dots, x_n))^2$$

minimal wird.

Es gibt Alternativen, z.B.

- Minimiere $\Delta_1 := \sum_{i=1}^m |\Delta_i|$.
- Minimiere $\Delta_\infty := \max_{i=1}^m |\Delta_i|$.

Diese sind aber zum Beispiel nicht differenzierbar.

6.2 Lineare Ausgleichsprobleme

Wir nehmen zunächst an, dass φ linear sei in x_1, \dots, x_n , also

$$\varphi(t; x_1, \dots, x_n) = a_1(t)x_1 + a_2(t)x_2 + a_3(t)x_3 + \dots + a_n(t)x_n$$

mit Funktionen $a_1, \dots, a_n : \mathbb{R} \rightarrow \mathbb{R}$.

Dann ist

$$\Delta^2 = \sum_{i=1}^m [b_i - (a_1(t_i)x_1 + \dots + a_n(t_i)x_n)]^2 = \|b - Ax\|_2^2$$

mit

$$A \in \mathbb{R}^{m \times n}, \quad A_{ij} = a_j(t_i) \quad x = (x_1, \dots, x_n)^T \in \mathbb{R}^n.$$

Geometrisch heißt das: Wir suchen einen Punkt $z = Ax$ aus dem Bildraum $R(A)$ von A , der den kleinsten Abstand zu b hat.

Bild

- $R(A)$ ist ein linearer Raum.
- $b - Ax$ steht senkrecht auf $R(A)$.

Satz 6.1 (Deuffhard und Hohmann [4], Satz 3.7). *i) Der Vektor $x \in \mathbb{R}^n$ ist genau dann Lösung von $\|b - Ax\| \rightarrow \min$, falls er die sogenannte Normalengleichung*

$$A^T Ax = Ab$$

erfüllt.

ii) Das Problem ist genau dann eindeutig lösbar, wenn $\text{Rang } A = n$, also maximal ist.

Beweis. i) Es gilt

$$\begin{aligned} \|b - Ax\| \rightarrow \min &\iff \langle b - Ax, Ax' \rangle = 0 && \text{für alle } x' \in \mathbb{R}^n \\ &\iff \langle A^T(b - Ax), x' \rangle = 0 && \text{für alle } x' \in \mathbb{R}^n \\ &\iff A^T(b - Ax) = 0 \\ &\iff A^T Ax = A^T b. \end{aligned}$$

ii)

- Es gilt $\text{Rang } A^T A = \text{Rang } A$.
- Also ist $A^T A$ genau dann invertierbar, wenn A den Rang n hat. □

6.2.1 Pseudoinverse

Die Lösungen der Normalengleichung lassen sich elegant mittels sogenannter Pseudoinversen darstellen.

Betrachte:

$$\|b - Ax\| \longrightarrow \min$$

mit $A \in \mathbb{R}^{m \times n}$, $m \geq n$.

- Falls A invertierbar ist, dann gilt $x = A^{-1}b$
- Falls A nicht invertierbar ist, dann löst x die Normalengleichung

$$A^T Ax = A^T b$$

- Die Matrix A habe Rang n . Dann ist $A^T A$ invertierbar und es folgt

$$x = (A^T A)^{-1} A^T b.$$

Definition (Pseudoinverse). Sei $\text{rank } A = n$. Dann ist die Pseudoinverse von A definiert als $A^+ := (A^T A)^{-1} A^T$.

Allgemeiner:

Definition. A^+ ist die Matrix aus $\mathbb{R}^{n \times m}$, so dass für alle $b \in \mathbb{R}^n$ der Vektor $x = A^+ b$ die kleinste Lösung von $\|b - Ax\| \rightarrow \min$ ist.

Satz 6.2. Die Moore-Penrose-Pseudoinverse $A^+ \in \mathbb{R}^{n \times m}$ einer Matrix $A \in \mathbb{R}^{m \times n}$ besitzt folgende Eigenschaften

1. $AA^+A = A$
2. $A^+AA^+ = A^+$
3. A^+A und AA^+ sind symmetrisch.

4. $(A^+)^+ = A$

5. $(A^T)^+ = (A^+)^T$

Falls $A \in \mathbb{R}^{m \times n}$ vollen Rang hat, dann gilt

$$A^+ A = I_n \in \mathbb{R}^{n \times n}$$

und

$$A A^+ = I_m \in \mathbb{R}^{m \times m}.$$

6.3 Das Gauß–Newton-Verfahren

Nun zu *nichtlinearen* Ausgleichsproblemen.

- Seien wieder $b_1, \dots, b_m \in \mathbb{R}$ Messdaten zu Zeitpunkten t_1, \dots, t_m .
- Sei $\varphi(\cdot, x_1, \dots, x_n)$ jetzt eine nichtlineare Modellfunktion, die von n reellen Parametern abhängt.
- Gesucht werden Parameter $x = (x_1, \dots, x_n) \in \mathbb{R}^n$, so dass das Residuum

$$\|F(x)\|^2 := \sum_{i=1}^m (b_i - \varphi(t_i, x_1, \dots, x_n))^2$$

(lokal) minimal wird.

Der Bequemlichkeit halber definieren wir

$$g(x) := \frac{1}{2} \|F(x)\|_2^2.$$

Hinreichende Kriterien für einen lokalen Minimierer sind

$$g'(x^*) = 0, \quad g''(x^*) \text{ positiv definit.}$$

Idee. Benutze ein Newton-Verfahren für die Gleichung

$$0 = g'(x) = F'(x)^T F(x) =: G(x), \quad G: \mathbb{R}^n \rightarrow \mathbb{R}^n.$$

Eine Newton-Korrektur Δx^k für diese Gleichung löst:

$$G'(x^k) \Delta x^k = -G(x^k) \quad \forall k = 0, 1, 2, \dots$$

Ausrechnen

$$G'(x) = F'(x)^T F'(x) + F''(x)^T F(x).$$

G' ist unter den gemachten Annahmen in der Nähe von x^* positiv definit, also invertierbar. (In x^* selbst ist $G'(x^*) = g''(x^*)$ nach Annahme pos. def. Weil g nach Annahme C^2 ist gilt die pos. Def.heit auch in der Nähe von x^* .)

Wir haben ein Problem: F'' wird benötigt, dies ist ein Tensor dritter Ordnung.

- Ausrechnen davon kann beschwerlich sein.
- Ausrechnen davon kann teuer sein, denn F'' hat n^3 Einträge.
- Können wir den 2. Summanden in G' einfach weglassen?
 - Wir erwarten/glauben, dass die Modellfunktion φ „gut“ ist, d.h., dass es Punkte/offene Mengen gibt, bei denen $\|F(x)\|^2$ zumindest „klein“ wird.
 - Diese Probleme nennt man dann „fast kompatible Probleme“.

Wir lassen also den zweiten Summanden in G' weg und hoffen auf das Beste.

Definition. Das Newton-Verfahren für die Gleichung $G(x) =$ ohne den zweiten Term in G' nennt man Gauß-Newton-Verfahren.

Es ist kein echtes Newton-Verfahren.

Deswegen bekommt man nicht alle schönen Eigenschaften eines Newton-Verfahrens.

Iterationsvorschrift mit modifizierten G' lautet:

$$F'(x^k)^T F'(x^k) \Delta x^k = -F'(x^k)^T F(x^k) \quad (6.1)$$

Das ist gerade die Normalengleichung des *linearen* Ausgleichsproblems

$$\|F(x^k) + F'(x^k) \Delta x^k\| \rightarrow \min.$$

Wir können ein nichtlineares Ausgleichsproblem lösen, indem wir eine Folge von linearen Problemen lösen.

Mit der Definition der Pseudoinversen erhalten wir

$$\|F(x^k) + F'(x^k) \Delta x^k\| \rightarrow \min \implies \Delta x^k = -F'(x^k)^+ F(x^k).$$

Satz 6.3 (Deuffhard und Hohmann [4], Satz 4.15). Sei $D \subset \mathbb{R}^m$ offen und konvex, $F: D \rightarrow \mathbb{R}^m$, $m \geq n$, stetig differenzierbar und $F'(x)$ habe vollen Rang $\forall x \in D$. Es existiere eine Lösung x^* des dazugehörigen Ausgleichsproblems. F' sei affin-invariant Lipschitz-stetig, d.h. es gibt ein $\omega > 0$ so dass für alle $s \in [0, 1]$

$$\|F'(x)^+(F'(x + s(y - x)) - F'(x))\| \leq s\omega \|y - x\|^2 \quad \forall x, y \in D.$$

Es gebe eine Konstante $\kappa_* \in [0, 1)$ so, dass

$$\forall x \in D: \|F'(x)^+ F(x^*)\| \leq \kappa_* \|x - x^*\|.$$

Diese Bedingung fordert, dass das Problem „fast kompatibel“, also $\|F(x^*)\|$ klein ist. Sei weiterhin der Startwert $x^0 \in D$ so, dass

$$\|x^0 - x^*\| < \frac{2}{\omega} (1 - \kappa_*).$$

1. Dann konvergiert das Gauß-Newton-Verfahren gegen x^*
2. Die Konvergenzgeschwindigkeit ist

$$\|x^{k+1} - x^*\| \leq \frac{\omega}{2} \|x^k - x^*\|^2 + \kappa_* \|x^k - x^*\|. \quad (6.2)$$

Beweis. Der Beweis ist dem Beweis der Konvergenz Newton-Verfahren sehr ähnlich.

Für alle $x, y \in D$ gilt:

$$\begin{aligned} \left\| F'(x)^+ [F(y) - F(x) - F'(y-x)] \right\| &\leq \left\| F'(x)^+ \int_0^1 [F'(x + s(y-x)) - F'(x)] (y-x) ds \right\| \\ &\leq \int_0^1 \left\| F'(x)^+ [F'(x + s(y-x)) - F'(x)] (y-x) \right\| ds \\ &\leq \int_0^1 s\omega \|y-x\|_2^2 ds \\ &= \frac{\omega}{2} \|y-x\|_2^2. \end{aligned}$$

Beachte: $F'(x)^+ F'(x) = I_n \forall x \in D$, da F' vollen Range habe.

Damit erhält man

$$\begin{aligned} x^{k+1} - x^* &= (x^k - x^*) - F'(x^k)^+ F(x^k) \\ &= \underbrace{F'(x^k)^+ F'(x^k)}_{=I_n} (x^k - x^*) - F'(x^k)^+ F(x^k) + \underbrace{F'(x^k)^+ F(x^*) - F'(x^k)^+ F(x^*)}_{=0} \\ &= \underbrace{F'(x^k)^+ [F(x^*) - F(x^k) - F'(x^k)(x^* - x^k)]}_{\leq \frac{\omega}{2} \|x^k - x^*\|^2} - \underbrace{F'(x^k)^+ F(x^*)}_{\leq \kappa_* \|x^k - x^*\|} \\ \implies \|x^{k+1} - x^*\| &\leq \left(\frac{\omega}{2} \|x^k - x^*\| + \kappa_* \right) \|x^k - x^*\|. \end{aligned}$$

Das ist gerade Behauptung 2) zur Konvergenzgeschwindigkeit.

Man erhält damit die Konvergenz des Verfahrens, falls

$$\frac{\omega}{2} \|x^k - x^*\| + \kappa_* < c < 1 \quad \forall k$$

Nach Voraussetzung:

$$\|x^0 - x^*\| < \frac{2}{\omega}(1 - \kappa_*) \iff \underbrace{\frac{\omega}{2} \|x^0 - x^*\| + \kappa_*}_{:=c} < 1$$

Nach Induktion ist damit

$$\begin{aligned} \forall k \in \mathbb{N}: \|x^{k+1} - x^*\| &< \left(\frac{\omega}{2} \|x^k - x^*\| + \kappa_* \right) \|x^k - x^*\| < \|x^k - x^*\| \\ \implies \forall k \in \mathbb{N}: \frac{\omega}{2} \|x^k - x^*\| + \kappa_* &< \frac{\omega}{2} \|x^0 - x^*\| + \kappa_* = c. \end{aligned}$$

Das Verfahren konvergiert. □

Das Verfahren konvergiert nur dann lokal quadratisch, falls $\kappa_* = 0$, falls also das Problem kompatibel ist.

Das ist der Preis dafür, dass wir F'' weggelassen haben.

7 Optimierung

Wir verallgemeinern unser Problem weiter.

- Sei $f: \mathbb{R}^n \rightarrow \mathbb{R}$ gegeben.
- Aufgabe: Finde einen (lokalen) Minimierer von f .

Beispiel. • x_1, \dots, x_n : Designparameter eines Rennautos
(z.B. Hubraum, Reifengröße, Gewicht, etc.)

- $f: \mathbb{R}^n \rightarrow \mathbb{R}$: Höchstgeschwindigkeit des Autos.
- Finde Minimierer von $-f$.

Beispiel (Festkörpermechanik). • Elastisches Objekt $\Omega \subset \mathbb{R}^3$

- Deformation: $\Phi: \Omega \rightarrow \mathbb{R}^3$
- Hyperelastizität: stabile Zustände Φ minimieren eine Energie

$$\mathcal{J}: C^1(\Omega, \mathbb{R}^3) \rightarrow \mathbb{R} \quad \mathcal{J}(\Phi) = \int_{\Omega} W(\nabla\Phi(x)) dx.$$

- Diskretisierung: Fülle Ω mit Dreiecken bzw. Tetraedern.
- Betrachte nur noch Position der Eckpunkte der Dreiecke (Knoten) (n viele).
- Aus $\mathcal{J}: C^1(\Omega, \mathbb{R}^3) \rightarrow \mathbb{R}$ wird $f: \mathbb{R}^{3n} \rightarrow \mathbb{R}$
- Stichwort: Finite Elemente

Sei zunächst f quadratisch mit symmetrischer Matrix $A \in \mathbb{R}^{n \times n}$:

$$f(x) = \frac{1}{2}x^T Ax - b^T x + c.$$

- Falls A positiv definit ist, dann existiert genau ein Minimierer x^* .
- Dieser löst

$$\nabla f(x^*) = 0 = Ax - b \iff Ax = b$$

- Problem zurückgeführt auf lineares Gleichungssystem \rightarrow schon bekannt.

7.1 Gradientenartige Verfahren

Sei ab jetzt f *nicht* quadratisch.

Alle bekannten Verfahren sind iterativ.

Allgemeiner Ansatz: Sei $x^0 \in \mathbb{R}^n$ gegeben.

Für $k = 1, 2, \dots$

- Wähle eine Richtung $p_k \in \mathbb{R}^n$
- Wähle eine Schrittweite $t_k \in \mathbb{R}$
- Setze $x^{k+1} = x^k + t_k p_k$.

Hoffnung:

1. (x_k) konvergiert gegen einen Minimierer von f für möglichst viele Startwerte x^0 .
2. Die Konvergenz ist schnell.

Hängt ab von:

- a) geschickter Wahl der Suchrichtungen p_k
- b) geschickter Wahl der Schrittweiten t_k .

In den allermeisten Fällen will man *Abstiegsverfahren*, d.h. es soll gelten

$$f(x^{k+1}) \leq f(x^k)$$

für alle $k \in \mathbb{N}$, mit Gleichheit nur wenn x^k Minimierer ist.

7.1.1 Schrittweiten

Sei $x^k \in \mathbb{R}^n$ und eine Abstiegsrichtung p_k gegeben.

Wie sollte man ein „gutes“ t_k wählen?

Dilemma:

- t_k muss sorgfältig gewählt werden, um möglichst viel Energiereduktion zu erhalten.
- Die Wahl von t_k selbst darf nicht zu aufwändig sein.

Idealerweise: Wähle t_k als globalen Minimierer von

$$\Theta: \mathbb{R} \rightarrow \mathbb{R}, t \mapsto f(x^k + t p_k), \quad (\text{exakte Liniensuche})$$

aber das ist i.A. viel zu teuer.

Stattdessen: inexacte Liniensuche

- Gegeben eine Folge von möglichen Schrittweiten.

- Wähle die erste Schrittweite, die einer gewissen Bedingung genügt.

Hoffnung: mit deutlich weniger Aufwand eine Schrittweite zu finden, die fast genauso gut ist.

Diese Methode ist der Dämpfungsstrategie im gedämpften Newton-Verfahren sehr ähnlich.


Aber:

- Dort wusste man, dass $t_k \in (0, 1]$ sein muss und dass $t_k = 1$ gewisse Vorteile bietet.
- Dieses Wissen hat man hier nicht.

Die einfachste Bedingung an die Schrittweite ist

- Wähle t_k so, dass $f(x^k + t_k p_k) < f(x^k)$ für alle $k \in \mathbb{N}$.

Das reicht nicht: betrachte beispielsweise die Funktion $f(x) = x^2 - 1$.



Absteigende Folge, die nicht gegen den Minimierer konvergiert.

Wir brauchen *hinreichenden Abstieg*.

Die Wolfe-Bedingungen

a) Armijo-Regel: Fordere Reduktion, die linear ist in der Schrittweite t_k und der Richtungsableitung

$$\frac{df}{dp} = \frac{d}{dt} f(x^k + t p_k) \Big|_{t=0} = \nabla f(x)^T p_k.$$

Das bedeutet

$$f(x^k + t p_k) \leq f(x^k) + c_1 t \nabla f(x^k)^T p_k, \quad c_1 \in (0, 1) \tag{7.1}$$

Auch diese Bedingung wird in der Literatur Armijo-Regel genannt

z.B. bei Nocedal & Wright?



Bild für die Armijo-Regel

Achtung: Die Armijo-Regel alleine reicht nicht aus. Die Schrittweiten können unnötig klein werden.

Um das zu vermeiden fordern wir:

b) Krümmungsbedingung:

$$\Theta'(t) \geq \underbrace{c_2 \Theta'(0)}_{<0} \quad c_2 \in (c_1, 1)$$

oder auch

$$\nabla f(x^k + tp_k)^T p_k \geq c_2 \nabla f(x^k)^T p_k, \quad (7.2)$$

Idee:

- Falls $\Theta'(t)$ stark negativ ist, bekomme ich mehr Abstieg, wenn ich t vergrößere.
- Falls $\Theta'(t)$ positiv oder nur wenig negativ ist, dann lohnt es sich nicht/kaum, t zu vergrößern.

(7.1) & (7.2) heißen zusammen *Wolfe-Bedingungen*.

Satz 7.1 (Nocedal und Wright [14], Lemma 3.1). *Sei $f: \mathbb{R}^n \rightarrow \mathbb{R}$ stetig differenzierbar und von unten beschränkt. Sei p_k Abstiegsrichtung in x^k . Für alle $c_1, c_2 \in \mathbb{R}$ mit $0 < c_1 < c_2 < 1$ existieren Intervalle von Schrittweiten t für die die Wolfe-Bedingungen gelten.*

Beweis. • $\phi(t) = f(x^k + tp_k)$ ist von unten beschränkt.

- Da p_k Abstiegsrichtung $\implies \nabla f(x^k)^T p_k < 0$.
- Deshalb ist $\ell(t) = f(x^k) + tc_1 \nabla f(x^k)^T p_k$ für $t > 0$ nicht von unten beschränkt.
- f ist stetig: \exists ein kleinstes $t' > 0$ mit

$$f(x^k + t'p_k) = f(x^k) + t'c_1 \nabla f(x^k)^T p_k.$$

- Also gilt die Armijo-Bedingung (7.1) für alle $t < t'$.
- Mittelwertsatz: $\exists t'' \in (0, t')$ so dass

$$\underbrace{f(x^k + t'p_k) - f(x^k)}_{=t'c_1 \nabla f(x^k)^T p_k} = t' \nabla f(x^k + t''p_k)^T p_k.$$

- Teile durch t' :

$$\nabla f(x^k + t''p_k)^T p_k = \underbrace{c_1}_{<c_2} \underbrace{\nabla f(x^k)^T p_k}_{<0} \geq c_2 \nabla f(x^k)^T p_k.$$

- t'' erfüllt auch Bedingung (7.2). □

7.1.2 Suchrichtungen

Wie wählt man die Suchrichtungen p_k ?

Eine scheinbar vernünftige Idee:

Wähle p_k als Richtung des steilsten Abstiegs von f in x^k (engl.: steepest descent).

Richtungsableitung:

$$\frac{df}{dp} := \left. \frac{d}{d\alpha} f(x + \alpha p) \right|_{\alpha=0}$$

Definition. Die Richtung des steilsten Abstiegs von f in x ist der Minimierer von $\frac{df}{dp}$ bezüglich p unter der Nebenbedingung $\|p\| = 1$.

Lemma 7.1. Der Minimierer ist

$$p^* = \frac{-\nabla f(x)}{\|\nabla f(x)\|}.$$

Beweis. Sei θ der Winkel zwischen p und $\nabla f(x)$.

- Dann ist

$$\frac{df}{dp} = p^T \nabla f(x) = \|p\| \|\nabla f(x)\| \cos \theta = \|\nabla f(x)\| \cos \theta$$

- Minimal, wenn $\cos \theta = -1 \implies \theta = \pi$

$$p = \frac{-\nabla f}{\|\nabla f\|}.$$

□

Das Gradientenverfahren wirkt vernünftig, kann aber sehr langsam sein.

Beispiel. Sei

$$f: \mathbb{R}^2 \rightarrow \mathbb{R}, \quad x \mapsto x^T \underbrace{\begin{pmatrix} \varepsilon & 0 \\ 0 & 1 \end{pmatrix}}_{=:A} x$$

mit $\varepsilon > 0$ klein.

Bemerkung. Das Problem ist schlecht konditioniert.

Man kann die optimale Schrittweite mit vertretbarem Aufwand berechnen:

$$t_k^* = \frac{\nabla f(x^k)^T \nabla f(x^k)}{\nabla f(x^k)^T A \nabla f(x^k)}$$

Und dennoch:



Bild vom schlecht konvergierenden Gradientenverfahren

Viele Algorithmen verwenden deshalb andere Suchrichtungen.

Man will aber häufig, dass die Richtungen p_k wenigstens ähnlich dem steilsten Abstieg sind (engl.: gradient-related), die also zumindest ungefähr in Richtung $-\nabla f(x^k)$ zeigen. Denn dann kann man Konvergenz beweisen.

Definiere θ_k als Winkel zwischen p_k und $-\nabla f(x^k)$

$$\cos \theta_k = \frac{-\nabla f(x^k)^T p_k}{\|\nabla f(x^k)\| \cdot \|p_k\|}.$$

Satz 7.2 (Nocedal und Wright [14], Satz 3.2). *Gegeben sei ein Verfahren*

$$x^{k+1} = x^k + t_k p_k,$$

wobei p_k immer Absiegsrichtung ist und t_k immer die Wolfe-Bedingung erfüllt. Sei $f: \mathbb{R}^n \rightarrow \mathbb{R}$ von unten beschränkt und stetig differenzierbar. Der Gradient ∇f sei Lipschitzstetig mit Konstante L . Dann folgt

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \|\nabla f(x^k)\|^2 < \infty.$$

Konsequenzen:

- Es gilt

$$\lim_{k \rightarrow \infty} \cos^2 \theta_k \|\nabla f(x^k)\|^2 = 0.$$

- Falls p_k so gewählt ist, dass θ_k von 90° weg beschränkt ist, dann

$$\exists \delta > 0: \cos \theta_k \geq \delta > 0$$

für alle $k \in \mathbb{N}$. Also gilt

$$\lim \|\nabla f(x^k)\| = 0.$$

- Das Verfahren konvergiert also gegen einen stationären Punkt, wenn die Suchrichtungen nicht „zu senkrecht“ auf $-\nabla f(x^k)$ stehen.
- Insbesondere „konvergiert“ das Gradientenverfahren gegen einen stationären Punkt, wenn die Schrittweite immer die Wolfe-Bedingung erfüllt.
- Konvergenz gegen stationäre Punkte, *nicht* gegen Minimierer!
- Mehr ist mit den oben genannten Annahmen nicht zu erreichen.
- Konvergenz gegen Minimierer nur mit zusätzlichen Annahmen an die p_k (Krümmung, d.h. Information über $\nabla^2 f$)
- Das ist natürlich teurer.

Beachte: Stationäre Punkte (außer Minimierer) sind instabil!

- Sei x^* ein Sattelpunkt und (x^k) mit $x^k \rightarrow x^*$ durch das Liniensuchverfahren erzeugt.
- Dann ist $f(x^k) \geq f(x^*)$ für alle $k \in \mathbb{N}$.
- Tatsächlich aber treten Rundungsfehler auf: Wenn x^k schon sehr nah an x^* ist, kann eventuell gelten

$$f(x^{k+1}) \geq f(x^*)$$

aber

$$f(\text{gerundet}(x^{k+1})) < f(x^*).$$

- Danach kann die Folge nicht mehr gegen x^* konvergieren.

Beweis von Satz 7.2. Die Wolfe-Bedingungen sind:

a) $f(x^{k+1}) \leq f(x^k) + c_1 t_k \nabla f(x^k)^T p_k$ (Armijo)

b) $\nabla f(x^{k+1})^T p_k \geq c_2 \nabla f(x^k)^T p_k$ (Krümmung)

Subtrahiere $\nabla f(x^k)^T p_k$ von b)

$$\left(\nabla f(x^{k+1}) - \nabla f(x^k) \right)^T p_k \geq (c_2 - 1) \nabla f(x^k)^T p_k.$$

∇f ist Lipschitz-stetig, deshalb gilt

$$\begin{aligned} \left| \left(\nabla f(x^{k+1}) - \nabla f(x^k) \right)^T p_k \right| &\leq \left\| \nabla f(x^{k+1}) - \nabla f(x^k) \right\| \cdot \|p_k\| \\ &\leq L \overbrace{\|x^{k+1} - x^k\|}^{t_k p_k} \cdot \|p_k\| = L t_k \|p_k\|^2 \end{aligned}$$

Zusammen

$$\begin{aligned} t_k L \|p_k\|^2 &\geq \left(\nabla f(x^{k+1}) - \nabla f(x^k) \right)^T p_k \geq (c_2 - 1) \nabla f(x^k)^T p_k \\ \implies t_k &\geq \frac{c_2 - 1}{L} \cdot \frac{\nabla f(x^k)^T p_k}{\|p_k\|^2}. \end{aligned}$$

Einsetzen in Armijo-Bedingung:

$$\begin{aligned} f(x^{k+1}) &\leq f(x^k) + c_1 t_k \underbrace{f(x^k)^T p_k}_{\leq 0} \\ &\leq f(x^k) + c_1 \frac{c_2 - 1}{L} \frac{(\nabla f(x^k)^T p_k)^2}{\|p_k\|^2} \\ &= f(x^k) - \underbrace{c_1 \frac{1 - c_2}{L}}_{=:c} \cdot \underbrace{\frac{(\nabla f(x^k)^T p_k)^2}{\|p_k\|^2 \cdot \|\nabla f(x^k)\|^2}}_{=\cos^2 \theta_k} \cdot \|\nabla f(x^k)\|^2 \\ &= f(x^k) - c \cdot \cos^2 \theta_k \cdot \|\nabla f(x^k)\|^2. \end{aligned}$$

Rekursives Einsetzen

$$f(x^{k+1}) \leq f(x^0) - c \sum_{j=0}^k \cos^2 \theta_j \|\nabla f(x^j)\|^2$$

beziehungsweise

$$\sum_{j=0}^k \cos^2 \theta_j \|\nabla f(x^j)\|^2 \leq \frac{1}{c} (f(x^0) - f(x^{k+1})).$$

- Rechte Seite ist nach oben beschränkt, da f nach unten beschränkt ist.
- Partialsummen sind also beschränkt, außerdem monoton steigend:

$$\implies \lim_{k \rightarrow \infty} \sum_{j=0}^k \cos^2 \theta_j \|\nabla f(x^j)\|^2 < \infty. \quad \square$$

7.1.3 Das Gradientenverfahren

Gegeben $x^0 \in \mathbb{R}^n$.

Für $k = 0, 1, 2, \dots$

$$x^{k+1} = x^k - t_k \nabla f(x^k).$$

Verfahren konvergiert global, falls die t_k die Wolfe-Bedingungen erfüllen.

Aber: Konvergenz kann sehr langsam sein.

Das zeigen wir jetzt ordentlicher:

Satz 7.3 (Nocedal und Wright [14], Satz 3.3). *Sei f quadratisch, also*

$$f(x) = \frac{1}{2} x^T A x - b^T x$$

mit symmetrischer, positiv definiter Matrix A .

- *Exakte Liniensuche*

$$t_k = \frac{\nabla f(x^k)^T \nabla f(x^k)}{\nabla f(x^k)^T A \nabla f(x^k)}$$

- *Energie-Norm $\|x\|_A^2 := x^T A x$.*

Dann gilt für den $k + 1$ -ten Fehler

$$\|x^{k+1} - x^*\|_A \leq \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \|x^k - x^*\|_A$$

mit $0 < \lambda_1 \leq \dots \leq \lambda_n$ den Eigenwerten von A .

Beachte: Die Konvergenz ist umso besser, je näher die Eigenwerte beieinander liegen

Beachte ($\kappa(A)$ ist die Kondition von A):

$$\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} = \frac{\frac{\lambda_n}{\lambda_1} - 1}{\frac{\lambda_n}{\lambda_1} + 1} = \frac{\kappa(A) - 1}{\kappa(A) + 1} \rightarrow \begin{cases} 1 & \text{für } \kappa(A) \rightarrow \infty, \\ 0 & \text{für } \kappa(A) \rightarrow 1. \end{cases}$$

Für nichtquadratische f gilt folgendes Resultat.

Satz 7.4 (Nocedal und Wright [14], Satz 3.4). *Sei $f: \mathbb{R}^n \rightarrow \mathbb{R}$ zweimal stetig differenzierbar. Angenommen, die Iterierten x^k des Gradientenverfahrens konvergieren zu einem x^* , wo $\nabla^2 f(x^*)$ positiv definit ist. Sei außerdem*

$$r \in \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}, 1 \right)$$

wobei $0 < \lambda_1 \leq \dots \leq \lambda_n$ die Eigenwerte von $\nabla^2 f(x^*)$ sind. Dann gilt

$$f(x^{k+1}) - f(x^*) \leq r^2(f(x^k) - f(x^*))$$

für alle k groß genug.

7.2 Das Newton-Verfahren

- Sei $x^k \in \mathbb{R}^n$.
- Approximiere f um x^k durch ein quadratisches Modell

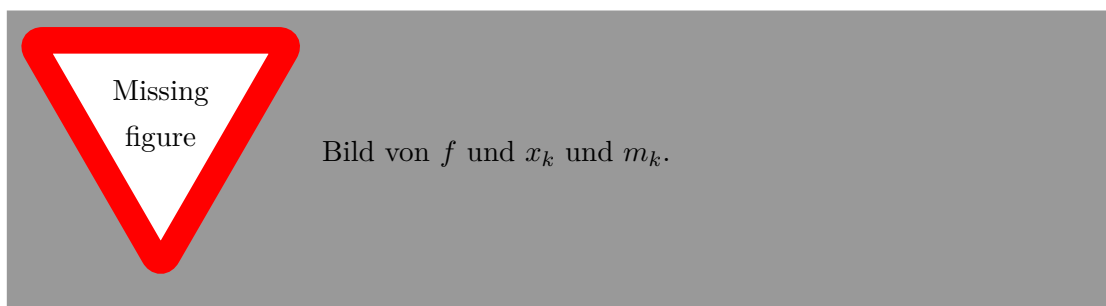
$$m_k(x^k + p) := f(x^k) + \nabla f(x^k)^T p + \frac{1}{2} p^T \nabla^2 f(x^k) p.$$

- Falls $\nabla^2 f(x^k)$ positiv definit ist, hat m_k einen eindeutigen Minimierer

$$p_k = -\nabla^2 f(x^k)^{-1} \nabla f(x^k)$$

Im Prinzip liegt das bekannte Newton-Verfahren für die Optimalitätsbedingung $F(x) := \nabla f(x) = 0$ vor.

Aber Abstiegsrichtungen erhält man nur, falls $\nabla F(x) = \nabla^2 f(x)$ positiv definit ist.



Wir wissen schon:

- Das ungedämpfte Newton-Verfahren konvergiert lokal quadratisch, also viel schneller als das Gradientenverfahren.
- Insbesondere gibt es eine besondere Schrittweite: zumindest in der Nähe einer Lösung ist $t_k = 1$ eine gute Wahl.
- Weiterer Vorteil bei Optimierungsproblemen: Anders als ∇F für beliebige Vektorfunktionen ist $\nabla^2 F$ auf jeden Fall symmetrisch.

Falls $\nabla^2 f(x^k)$ nicht positiv definit ist, dann ... müssen wir tricksen:

- Ersetze $\nabla^2 f(x^k)$ durch eine ähnliche Matrix, die symmetrisch und positiv definit ist.
- Addiere Vielfaches der Identität (Einheitsmatrix)
- Modifizierte Cholesky-Zerlegung
- ...

7.2.1 Konvergenzeigenschaften des Newton-Verfahrens

Wir hatten gesehen, dass ein allgemeines Liniensuchverfahren konvergiert, falls ein $\sigma > 0$ existiert, so dass

$$\cos \theta_k := \frac{-\nabla f(x^k)^T p_k}{\|\nabla f(x^k)\| \cdot \|p_k\|} \geq \sigma \quad \forall k \in \mathbb{N}.$$

Lemma 7.2 (Übung). Sei $M \in \mathbb{R}$ eine obere Schranke der Kondition von $\nabla^2 f$, also

$$\kappa(\nabla^2 f) = \|\nabla^2 f\| \cdot \|\nabla^2 f^{-1}\| \leq M \quad \forall k \in \mathbb{N}.$$

Dann gilt $\cos \theta_k \geq \frac{1}{M}$.

Es folgt: Das Newton-Verfahren konvergiert, falls die Folge der $\nabla^2 f(x^k)$ beschränkte Kondition hat.

Satz 7.5 (Nocedal und Wright [14], Satz 3.5). Sei $f: \mathbb{R}^n \rightarrow \mathbb{R}$ zweimal stetig differenzierbar und $x^* \in \mathbb{R}^n$ mit $\nabla f(x^*) = 0$ und $\nabla^2 f(x^*)$ positiv definit. Sei $\nabla^2 f$ Lipschitz-stetig in einer Umgebung von x^* , (x^k) die Newton-Folge $x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k)$. Falls x^0 hinreichend nah an x^* liegt

- a) konvergiert die Folge gegen x^* .
- b) Die Konvergenz ist lokal quadratisch.
- c) Die Folge $(\|\nabla f(x^k)\|)_{k \in \mathbb{N}}$ konvergiert quadratisch gegen 0.

Beachte: Für alle x^k hinreichend nah an x^* erfüllt die Schrittweite $t_k = 1$ die Wolfe-Bedingung.

7.3 Quasi-Newton-Verfahren

Nachteil des Newton-Verfahrens:

Die Auswertung von $\nabla^2 f$ kann schwierig/teuer sein.

Quasi-Newton-Verfahren:

- Ersetze $\nabla^2 f(x^k)$ durch eine Approximation $B_k \in \mathbb{R}^{n \times n}$.
- Suchrichtung $p_k = -B_k^{-1} \nabla f(x^k)$.
- Konstruktion der B_k :

Idee: Die Folge der Gradienten $(\nabla f(x^k))_k$ enthält Information über die zweiten Ableitungen von f .

$$f'(x^{k+1}) > f'(x^k) \implies "f''(x^{k+1}) > 0.$$

Dazu ein Bild?

Formaler: Taylor-Entwicklung

$$\nabla f(x+p) = \nabla f(x) + \nabla^2 f(x)p + \underbrace{\int_0^1 [\nabla^2 f(x+sp) - \nabla^2 f(x)] p ds}_{=:R(p)}$$

∇f ist stetig, deshalb gilt für das Restglied R

$$\|R(p)\| \in o(\|p\|) \iff \lim_{\|p\| \rightarrow 0} \frac{\|R(p)\|}{\|p\|} = 0.$$

Also folgt

$$\nabla f(x^{k+1}) = \nabla f(x^k) + \nabla^2 f(x^k) \cdot (x^{k+1} - x^k) + o(\|x^{k+1} - x^k\|).$$

Seien x^{k+1}, x^k in einer Umgebung von x^* , in der $\nabla^2 f$ „hinreichend“ positiv definit ist. Dann ist

$$\nabla^2 f(x^k) \underbrace{(x^{k+1} - x^k)}_{=:s^k} \approx \underbrace{\nabla f(x^{k+1}) - \nabla f(x^k)}_{=:y^k}$$

Idee des Quasi-Newton-Verfahrens:

Konstruiere B_{k+1} so, dass diese Bedingung erfüllt ist:

$$B_{k+1}s^k = y^k \quad (\text{Sekantengleichung}).$$

Die Sekantengleichung bestimmt B_{k+1} allerdings nur falls $n = 1$.

Wir benötigen also zusätzliche Bedingungen.

Weitere Wünsche:

- Symmetrie
- $B_{k+1} - B_k$ habe niedrigen Rang (spart viel Speicher, falls $k \leq n$)

Es existieren viele Varianten.

Die wohl wichtigste Variante ist die *BFGS-Formel* (nach Broyden, Fletcher, Goldfarb, Shanno)

$$B_{k+1} = B_k - \frac{B_k s_k \cdot s_k^T B_k^T}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

Nützlich:

- $B_{k+1} - B_k$ hat Rang 2
- Alle B_k sind symmetrisch.
- Alle B_k erfüllen die Sekantengleichung.
- Alle B_k sind positiv definit, falls auch B_0 positiv definit ist.

Für Quasi-Newton-Methoden brauchen wir eine neue Art von Konvergenzgeschwindigkeit.

Definition. Eine Folge (x^k) konvergiert superlinear gegen x^* , falls

$$\lim_{k \rightarrow \infty} \frac{\|x^{k+1} - x^k\|}{\|x^k - x^*\|} = 0.$$

Satz 7.6 (Nocedal und Wright [14], Satz 3.6). Sei $f \in C^2(\mathbb{R}^n, \mathbb{R})$. Betrachte die Iteration

$$x^{k+1} = x^k + t_k p_k$$

wobei:

- p_k ist Abstiegsrichtung
- t_k erfülle die Wolfe-Bedingungen mit $c \leq \frac{1}{2}$.

Die Folge (x^k) konvergiere gegen ein x^* mit $\nabla f(x^*) = 0$ und $\nabla^2 f(x^*)$ positiv definit. Falls

$$\lim_{k \rightarrow \infty} \frac{\|\nabla f(x^k) + \nabla^2 f(x^k) p_k\|}{\|p_k\|} = 0,$$

dann gilt:

- Die Schrittweite $t_k = 1$ erfüllt die Wolfe-Bedingungen für alle k groß genug.
- Falls $t_k = 1$ gewählt wird für alle k groß genug, dann konvergiert (x^k) superlinear.

Was heißt das für Quasi-Newton-Verfahren?

Sei $p_k = -B_k^{-1}\nabla f(x^k)$.

Dann ist die zentrale Bedingung aus Satz 7.6

$$\lim_{k \rightarrow \infty} \frac{\|\nabla f(x^k) + \nabla^2 f(x^k)p_k\|}{\|p_k\|} = \lim_{k \rightarrow \infty} \frac{\|(B_k - \nabla^2 f(x^k))p_k\|}{\|p_k\|} = 0.$$

Dabei haben wir benutzt, dass

$$\nabla f(x^k) = B_k B_k^{-1} \nabla f(x^k) = -B_k p_k.$$

Das sind gute Nachrichten!

→ Es heißt nämlich NICHT, dass die B_k immer bessere Approximationen von $\nabla^2 f(x^k)$ werden müssen.

→ Sie müssen $\nabla^2 f(x^k)$ nur *entlang der Suchrichtungen* immer besser approximieren.

7.4 Trust-Region-Verfahren

Bisher haben wir Liniensuchmethoden behandelt

- Suche *erst* eine Richtung $p_k \in \mathbb{R}^n$
- Suche dann eine Schrittweite $t_k \in \mathbb{R}$
- Setze $x^{k+1} = x^k + t_k p_k$

Trust-Region-Verfahren wählen p_k und t_k zusammen.

- Sei x^k die aktuelle Iterierte
- Approximiere f um x^k durch ein quadratisches Modell

$$m_k(p) = f(x^k) + g^T p + \frac{1}{2} p^T B_k p$$

mit $g_k = \nabla f(x^k)$, $B_k \in \mathbb{R}^{n \times n}$ ist Approximation von $\nabla^2 f(x^k)$, zum Beispiel $\nabla^2 f(x^k)$ selbst.

Idee: Vertraue m_k nur in einer Kugel um x^k (der Trust-Region) mit Radius Δ_k .

Wähle als Korrektur-Schritt

$$p_k = \operatorname{argmin}_{\|p\| \leq \Delta_k} m_k(p)$$

Wir erhalten den Vorteil, dass p_k immer definiert ist, selbst wenn B_k nicht positiv definit ist, dafür aber den Nachteil, dass in jedem Schritt ein Minimierungsproblem *mit Nebenbedingung* zu lösen ist. *Wie wählt man Δ_k ?* Grundlage: Wie gut hat m_k den Energieverlust $x^k \rightarrow x^k + p_k$ prognostiziert? Definiere:

$$\rho_k = \frac{f(x^k) - f(x^k + p_k)}{m_k(0) - m_k(p_k)}$$

Wähle zwei Konstanten $0 < \eta_1 < \eta_2 < 1$, z.B. $\eta_1 = 0,1, \eta_2 = 0,9$. Für $k = 0, 1, 2, \dots$

- Setze $p_k = \operatorname{argmin}_{\|p\| < \Delta_k} m_k(p)$
- Berechne ϱ_k
- Fall 1: $\varrho_k < \eta_1$
 - 1) $x^{k+1} = x^k$
 - 2) $\Delta_{k+1} = \frac{1}{2}\Delta_k$
- Fall 2: $\eta_1 \leq \varrho_k \leq \eta_2$
 - 1) $x^{k+1} = x^k + p_k$
- Fall 3: $\varrho_k > \eta_2$
 - 1) $x^{k+1} = x^k + p_k$
 - 2) $\Delta_{k+1} = 2\Delta_k$

7.5 Globale Konvergenz

Definition (Cauchy-Punkt). Der Cauchy-Punkt p_k^c ist der Minimierer von m_k innerhalb der Trust-Region in Richtung des negativen Gradienten.

$$p_k^c = -\frac{g_k}{\|g_k\|} \cdot \tau_k$$

wobei

$$\tau_k = \begin{cases} \Delta_k, & \text{falls } g_k^T B_k g_k < 0 \\ \min \left\{ \Delta_k, \underbrace{\frac{\|g_k\|^3}{g_k^T B_k g_k}}_{\text{billig}} \right\}, & \text{sonst} \end{cases}$$

Der Cauchy-Punkt erzeugt Energieabstieg *im Modell* ähnlich wie der von der Armijo-Bedingung gefordert.

Lemma 7.3 (nocedal_wright, Satz 4.3). Für den Cauchy-Punkt p_k^c gilt

$$m(0) - m(p_k^c) \geq \frac{1}{2} \|g_k\| \cdot \min \left\{ \Delta_k, \frac{\|g_k\|}{\|B_k\|} \right\} \quad (*)$$

Satz 7.7. Falls [technische Bedingungen], und p_k so gewählt wird, dass (*) für alle $k \in \mathbb{N}$ erfüllt ist, dann folgt

$$\lim_{k \rightarrow \infty} \|g_k\| = 0$$

7.6 Das Hundebein-Verfahren

[dogleg method] Sei B_k positiv definit. Der Minimierer von m_k ohne Nebenbedingung ist

$$p^B = -B_k^{-1}g_k$$

Falls p^B zulässig ist, also $\|p^B\| \leq \Delta$, dann ist p^B auch Lösung des quadratischen Minimierungsproblems mit Nebenbedingungen. Sei $p^*(\Delta)$ der Minimierer von m_k in der Trust-Region als Funktion des Radius. Sei Δ klein im Verhältnis zu $\|p^B\|$. Dann ist der quadratische Term in $m(p) = f(x^k) + g_k^T p + \frac{1}{2}p^T B_k p$ eher irrelevant. Dann ist der Minimierer von m_k ungefähr der Cauchy-Punkt

$$p^*(\Delta) \approx -\Delta \frac{g}{\|g\|}$$

dogleg-Methode: Wähle p_k als Minimierer von m_k auf dem gelben Pfad unter der Nebenbedingung $\|p_k\| \leq \Delta_k$.

Satz 7.8. Der Vektor p^* ist Minimierer von

$$\min_{\|p\| \leq \Delta_k} m(p) = f(x^k) + g^T p + \frac{1}{2}p^T B_k p$$

$\iff \|p^*\| \leq \Delta_k$, und es eine Zahl $\lambda \geq 0$ gibt so dass

$$\begin{aligned} (B + \lambda I)p^* &= -g \\ \lambda(\Delta - \|p^*\|) &= 0 \end{aligned}$$

und $(B + \lambda I)$ ist positiv semidefinit.

Berechnungsmethode von Minimierern

Algorithmus. Für λ groß genug definiere

$$p(\lambda) = -(B + \lambda I)^{-1}g$$

Falls $\|p^*\|$ auf dem Rand der Trust-Region liegt, dann verwende das Newton-Verfahren zu Lösen von

$$\|p(\lambda)\| - \Delta_k = 0$$

8 Iterative Lösungsverfahren für große, dünnbesetzte Gleichungssysteme

In manchen Anwendungen stößt man auf Matrizen, die sehr groß sind, aber fast ausschließlich Nullen enthalten.

Solche Matrizen nennt man *dünnbesetzt* oder *dünn* (engl. *sparse*).

8.1 Motivation: Das Poisson-Problem

Sei Ω eine offene, beschränkte Menge in \mathbb{R}^2 , und $f : \Omega \rightarrow \mathbb{R}$ eine gegebene Funktion. Gesucht wird eine Funktion $u : \Omega \rightarrow \mathbb{R}$ für die

$$\begin{aligned} -\Delta u &:= -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f && \text{auf } \Omega, \\ u &= 0 && \text{auf dem Rand von } \Omega. \end{aligned}$$

Solch eine Funktion u beschreibt z. B.

- Temperaturverteilung bei gegebener Wärmezufuhr f ,
- Elektrostatisches Potential bei gegebener Ladungsdichte f ,
- Flüssigkeitsdruck in einem porösen Medium.

Wie findet man so ein u ?

Eine Möglichkeit: Finite Differenzen

- Sei $g : \mathbb{R} \rightarrow \mathbb{R}$ hinreichend oft stetig differenzierbar.
- Taylorentwicklung um ein $x \in \mathbb{R}$:

$$g(x+h) = g(x) + g'(x)h + \text{Rest},$$

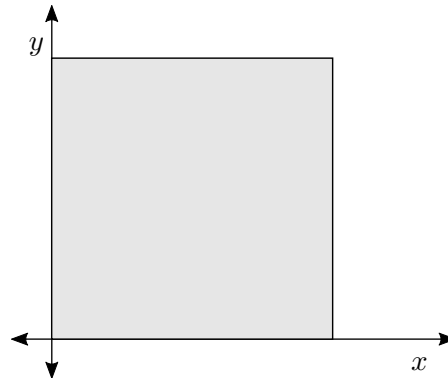
also

$$g'(x) \approx \frac{g(x+h) - g(x)}{h}$$

- Ähnlich erhält man

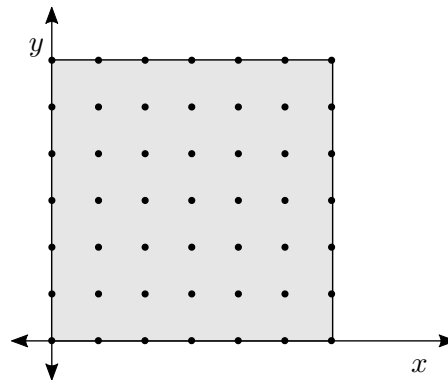
$$g''(x) \approx \frac{g(x+h) - 2g(x) + g(x-h)}{h^2}.$$

Das machen wir jetzt für die zweiten partiellen Ableitungen.
Sei Ω der Einfachheit halber das Einheitsquadrat.



Wähle ein $N \in \mathbb{N}$, definiere die Gitterweite $h := \frac{1}{N}$, und das Gitter

$$(x_i, y_j) := (ih, jh), \quad 0 \leq i, j \leq N.$$



Betrachte den Laplace-Operator an einem inneren Gitterpunkt (x_i, y_j) :

$$\begin{aligned} \Delta u(x_i, y_j) &= \frac{\partial^2 u(x_i, y_j)}{\partial x^2} + \frac{\partial^2 u(x_i, y_j)}{\partial y^2} \\ &\approx \frac{u(x_i + h, y_j) - 2u(x_i, y_j) + u(x_i - h, y_j)}{h^2} + \frac{u(x_i, y_j + h) - 2u(x_i, y_j) + u(x_i, y_j - h)}{h^2} \\ &= \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2} + \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{h^2}. \end{aligned}$$

Nummeriere die Gitterknoten von links unten nach rechts oben durch.
Seien u_i, f_i die Werte der Funktionen u bzw. f am i -ten Gitterknoten.
Man erhält das lineare Gleichungssystem

$$A\bar{u} = b$$

mit

$$A = \frac{1}{h^2} \begin{pmatrix} T & -I & & & \\ -I & T & -I & 0 & \\ & \ddots & \ddots & \ddots & \\ & & -I & T & -I \\ & & & -I & T \end{pmatrix}, \quad T = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & 0 & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix} \in \mathbb{R}^{(N-1) \times (N-1)},$$

und I der $(N-1) \times (N-1)$ Einheitsmatrix.

Der Vektor \bar{u} enthält die Einträge $(u_1 \ u_2 \ \dots \ u_n)$ der numerischen Lösung an den Gitterpunkten, und $b = (f_1 \ f_2 \ \dots \ f_n)$ enthält die Werte der Funktion f dort.

Das Gleichungssystem hat die Größe $n = (N-1)^2$. Zwar sind es insgesamt $(N+1)^2$ Gitterpunkte, aber an allen Gitterpunkten auf dem Rand ist die Lösung u durch die Randbedingung $u = 0$ festgelegt.

8.1.1 Eigenschaften der Matrizen

Größe

Die Matrizen aus dem obigen Beispiel können sehr groß werden.

- Für jeden Gitterknoten eine Gleichung
- Für d -dimensionale Gebiete hat man etwa $n \approx \text{Vol}(\Omega) \cdot h^{-d}$ Knoten
- Je feiner das Gitter, desto präziser die Lösung, desto größer aber auch die Matrix.
- Mein Laptop: ca. 8 GB RAM; eine Zahl in doppelter Genauigkeit braucht 8 Byte. Also hat man Platz für 1 Milliarde Zahlen.
- Aktuelle Hochleistungsrechner: $n \approx 10^{11}$.

Wie löst man diese Gleichungssysteme? Direkte Verfahren wie Gauß-Elimination oder Cholesky-Zerlegung sind i.A. zu teuer.

Erinnerung: Gauß-Elimination braucht $O(n^3)$ Rechenoperationen!

Dünnbesetztheit

- Sei i ein innerer Knoten im Gitter für das Poisson-Problem.
- Die Gleichung für u_i ist

$$4u_i - u_{i-1} - u_{i+1} - u_{i-(N+1)} - u_{i+N+1} = f_i.$$

Die i -te Zeile von A enthält also nur 5 Einträge, der Rest ist Null.

- Allgemein: Die Anzahl der Nicht-Null-Einträge pro Zeile ist durch eine kleine Konstante beschränkt.

- Die Matrix enthält also nur $O(n)$ Einträge
- Wendet man das Gauß-Verfahren auf solch eine Matrix an, so entstehen bei den Zwischenschritten in der Matrix eine beträchtliche Anzahl von zusätzlichen Einträgen („fill-in“).
- Das Gauß-Verfahren ist deshalb nicht nur zu langsam, es braucht auch zu viel Speicher.

Konsequenz: Wir brauchen Algorithmen und Datenstrukturen, die die Dünnbesetztheit ausnutzen.

Beispiel: Matrix–Vektor-Multiplikation $v = Aw$.

1. Naiv:

```

1 for alle Zeilen  $i$  do
2    $v_i = 0$ 
3   for alle Spalten  $j$  do
4      $v_i = v_i + A_{ij}w_j$ 
5   end
6 end

```

Das braucht $O(n^2)$ Operationen.

2. Pseudo-schlau:

```

1 for alle Zeilen  $i$  do
2    $v_i = 0$ 
3   for alle Spalten  $j$  do
4     if  $A_{ij} \neq 0$  then
5        $v_i = v_i + A_{ij}w_j$ 
6     end
7   end
8 end

```

Das braucht ebenfalls $O(n^2)$ Operationen!

3. Wirklich schlau:

```

1 for alle Zeilen  $i$  do
2    $v_i = 0$ 
3   for alle Spalten  $j$  in denen  $A_{ij} \neq 0$  do
4      $v_i = v_i + A_{ij}w_j$ 
5   end
6 end

```

Das braucht nur $O(\#\text{Nichtnulleinträge})$ Operationen.

Besondere Forschungsrichtung: direkte Sparse-Verfahren (Das machen wir in Kapitel 9.)

8.2 Lineare iterative Verfahren

Dieses Kapitel ist weitestgehend dem Buch von Dahmen und Reusken [2] entnommen.

Sei $A \in \mathbb{R}^{n \times n}$ nichtsingulär, groß und dünnbesetzt und $b \in \mathbb{R}^n$. Finde $x \in \mathbb{R}^n$ so, dass

$$Ax = b$$

(x^* sei von nun an die Lösung).

Idee der iterativen Verfahren:

1. Wähle eine Startiterierte $x^0 \in \mathbb{R}^n$.
2. Berechne daraus eine Iterierte $x^1 \in \mathbb{R}^n$. x^1 ist zwar nicht die Lösung, aber hoffentlich „näher dran“ als x^0 .
3. Wiederhole 2) so lange, bis ausreichend Genauigkeit erreicht ist.

Man erhält eine Folge x^0, x^1, x^2, \dots , die (hoffentlich) gegen x^* konvergiert.

Es gibt sehr viele Ansätze für 2). Ein paar werden wir jetzt betrachten.

Folgende Idee führt auf eine ganze Klasse von Verfahren: Das *Residuum* $b - Ax$ ist eine Art Fehler. Dann ist

$$x^{k+1} := x^k + b - Ax^k$$

vielleicht näher an x^* als x^k .

Formaler: Schreibe $Ax = b$ als Fixpunktgleichung

$$x = x + C(b - Ax)$$

mit $C \in \mathbb{R}^{n \times n}$ nichtsingulär. Setze

$$\Phi: \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad x \mapsto x + C(b - Ax).$$

Die Lösung x^* ist Fixpunkt von Φ .

Dafür machen wir jetzt eine Fixpunktiteration:

$$x^{k+1} := \Phi(x^k) = x^k + C(b - Ax^k) = (I - CA)x^k + Cb, \quad k = 0, 1, 2, \dots$$

8.2.1 Konvergenz

Unter welchen Umständen konvergiert dieses Verfahren?

Der Fehler im k -ten Iterationsschritt ist $e^k = x^k - x^*$. Es gilt

$$e^{k+1} = x^{k+1} - x^* = \Phi(x^k) - \Phi(x^*) = \underbrace{(I - CA)}_{\text{Iterationsmatrix}} e^k.$$

Also gilt

$$e^k = (I - CA)^k e^0 \quad \forall k = 0, 1, 2, \dots \quad (8.1)$$

Definition. Die Matrix $I - CA$ heißt Iterationsmatrix der Methode.

Die Fehlerfortpflanzung (8.1) ist linear, deshalb werden solche Verfahren lineare Verfahren genannt.

Für die Konvergenz gilt der folgende wichtige Satz.

Satz 8.1. Sei $\rho(I - CA)$ der Spektralradius von $I - CA$. Das Verfahren konvergiert für jeden Startwert $x^0 \in \mathbb{R}$ gegen die Lösung von $Ax = b$ genau dann, wenn $\rho(I - CA) < 1$.

Beweis. Wir beweisen nur den einfachen, aber wichtigen Fall, dass $I - CA$ symmetrisch positiv-definit (s.p.d.) ist.

Erstens: Aus $\rho < 1$ folgt Konvergenz.

- $I - CA$ ist symmetrisch und positiv definit, also diagonalisierbar. Das heißt es existiert eine nichtsinguläre Matrix T so dass

$$T^{-1}(I - CA)T = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{pmatrix} =: D,$$

wobei $\lambda_1, \lambda_2, \dots, \lambda_n$ die Eigenwerte von $I - CA$ sind.

- $e^k = (I - CA)^k e^0 = (TDT^{-1})^k e^0 = TD^k T^{-1} e^0$.
- Schätze e^k in einer Norm ab, z.B. der $\|\cdot\|_2$ -Norm

$$\begin{aligned} \|e^k\|_2 &= \|TD^k T^{-1} e^0\|_2 \\ &\leq \|T\|_2 \cdot \|D^k\|_2 \cdot \|T^{-1} e^0\|_2 \\ &\leq \|T\|_2 \cdot \|T^{-1} e^0\|_2 \cdot \underbrace{\max_{i=1, \dots, n} |\lambda_i|^k}_{=\rho(I-CA)}. \end{aligned}$$

- Dieser Term geht gegen 0, wenn $\rho(I - CA) = \max_i |\lambda_i| < 1$ ist.

Zeige jetzt: Aus Konvergenz folgt $\rho < 1$

- Angenommen $|\lambda_j| \geq 1$ für ein j , und $|\lambda_j| = \rho(I - CA)$.
- Sei v ein zu λ_j gehörender Eigenvektor.
- Wähle als Startwert $x^0 = x^* + v$, also $e^0 = v$.
- Dann folgt

$$\|e^k\|_2 = \|(I - CA)^k e_0\|_2 = \|(I - CA)^k v\|_2 = |\lambda_j|^k \|v\|_2 \geq \|e^0\|_2$$

für alle k .

⇒ Das Verfahren konvergiert nicht.

In allen endlich-dimensionalen Vektorräumen sind alle Normen äquivalent. Deshalb ändert sich das Resultat auch nicht, wenn man eine andere Norm betrachtet. □

Der Spektralradius einer Matrix ist nur mit Mühe auszurechnen. Allerdings gilt $\rho(B) \leq \|B\|$ für jede submultiplikative Matrixnorm.

[Denn: Sei v ein Eigenvektor von B zum Eigenwert λ . Dann ist

$$|\lambda|\|v\| = \|\lambda v\| = \|Bv\| \leq \|B\|\|v\|.$$

Deshalb gilt $|\lambda| \leq \|B\|$ für alle Eigenwerte λ von B .]

Deshalb:

Korollar. Für jede Vektornorm $\|\cdot\|$ mit dazugehöriger Operatornorm gilt

$$\forall k = 0, 1, 2, \dots : \|x^k - x^*\| \leq \|I - CA\|^k \cdot \|x^0 - x^*\|.$$

Das Verfahren konvergiert genau dann, wenn $\|I - CA\| < 1$ für eine beliebige Norm gilt.

8.2.2 Konvergenzgeschwindigkeit

Man möchte gerne wissen, *wie schnell* die Folge x^0, x^1, x^2, \dots gegen x^* konvergiert. Fehlerreduktionsrate im k -ten Schritt: $\frac{\|e^k\|}{\|e^{k-1}\|}$ und gemittelt über die ersten k Schritte

$$\sqrt[k]{\frac{\|e^k\|}{\|e^0\|}} =: \rho_k$$

Auch die Größe ρ_k hängt mit $\rho(I - CA)$ zusammen!

Beweis.

- Sei $I - CA$ wieder diagonalisierbar
- Eigenvektorbasis: v_1, v_2, \dots, v_n
- Nummerierung nach absteigenden Eigenwerten

$$|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$$

- Stelle den Anfangsfehler e^0 in der Eigenvektorbasis dar:

$$e^0 = \sum_{i=1}^n c_i v_i$$

- Sei o.B.d.A. $c_1 \neq 0$

- Es gilt

$$\begin{aligned}
 e^k &= (I - CA)^k \sum_{i=1}^n c_i v_i = \sum_{i=1}^n c_i \lambda_i^k v_i \\
 &= c_1 \lambda_1^k v_1 + \sum_{i=2}^n c_i \lambda_i^k v_i \\
 &= \lambda_1^k \left(c_1 v_1 + \underbrace{\sum_{i=2}^n c_i \left(\frac{\lambda_i}{\lambda_1} \right)^k v_i}_{=: r_k} \right) \\
 &= \lambda_1^k (c_1 v_1 + r_k).
 \end{aligned}$$

- Es gilt $c_1 \neq 0$ und $\left| \frac{\lambda_i}{\lambda_1} \right| \leq 1 \implies \exists$ Konstanten c_{\min}, c_{\max} (unabhängig von k), sodass

$$0 < c_{\min} < \|c_1 v_1 + r_k\| \leq c_{\max}$$

- Deshalb:

$$\rho_k = \sqrt[k]{\frac{\|e^k\|}{\|e^0\|}} = \frac{|\lambda_1| \|c_1 v_1 + r_k\|^{\frac{1}{k}}}{\|e^0\|^{\frac{1}{k}}} \xrightarrow{k \rightarrow \infty} |\lambda_1| = \rho(I - CA) \quad \square$$

Wie viele Schritte braucht man, um den Fehler um den Faktor $\frac{1}{e} \approx \frac{1}{2,718\dots}$ zu reduzieren?

$$(\rho_k)^k := \frac{\|e^k\|}{\|e^0\|} \approx \frac{1}{e} \iff k \approx \frac{1}{-\ln \rho_k}$$

- Interpretiere $-\ln \rho^k$ als Konvergenzgeschwindigkeit.
- Asymptotische Konvergenzgeschwindigkeit:

$$-\ln(\rho(I - CA))$$

- Für große k ist $\rho(I - CA)$ in etwa die gemittelte Fehlerreduktionsrate.

8.2.3 Die Wahl von C

Wie soll man die Matrix C wählen?

Ziel: $\rho(I - CA)$ soll möglichst klein sein.

- Ideal wäre $C = A^{-1} \implies \rho(I - CA) = 0$. Das Verfahren konvergiert dann in einem Schritt

$$x^1 = x^0 + A^{-1}(b - Ax^0) = A^{-1}b = x^*.$$

- Die Durchführung dieses Schrittes wäre aber sehr teuer. Es muss das LGS

$$A(x^1 - x^0) = b - Ax^0$$

gelöst werden.

Wir haben somit nichts gewonnen. Es ergibt sich folgendes Dilemma:

1. C soll A^{-1} möglichst gut approximieren
2. Die Operation $y \mapsto Cy$ soll möglichst billig sein

Beachte: Die Matrix C wird nie explizit ausgerechnet!

8.2.4 Das Jacobi-Verfahren

Wir nehmen im Folgenden an, dass $a_{ii} \neq 0$ für alle $i = 1, \dots, n$.

Betrachte das lineare Gleichungssystem:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1m}x_m &= b_1 \\ a_{12}x_1 + a_{22}x_2 + \dots + a_{2m}x_m &= b_2 \\ &\vdots \end{aligned}$$

Idee: Löse i -te Zeile nach x_i auf für $i = 1, \dots, n$.

$$\begin{aligned} x_1 &= \frac{1}{a_{11}} (b_1 - a_{12}x_2 - \dots - a_{1n}x_n) \\ x_2 &= \frac{1}{a_{22}} (b_2 - a_{21}x_1 - \dots - a_{2n}x_n) \\ &\vdots \end{aligned}$$

Mache daraus ein iteratives Verfahren:

$$\begin{aligned} x_1^{k+1} &= \frac{1}{a_{11}} (b_1 - a_{12}x_2^k - \dots - a_{1n}x_n^k) \\ x_2^{k+1} &= \frac{1}{a_{22}} (b_2 - a_{21}x_1^k - \dots - a_{2n}x_n^k) \\ &\vdots \end{aligned}$$

Für $i = 1, \dots, n$

$$x_i^{k+1} = \frac{1}{a_{ii}} (b_i - a_{i1}x_1^k - a_{i2}x_2^k - \dots - a_{i,i-1}x_{i-1}^k - a_{i,i+1}x_{i+1}^k - \dots - a_{in}x_n^k).$$

Oder, kompakter

$$\forall i \in \{1, 2, \dots, n\}: \quad x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}x_j^k \right)$$

Beachte:

- Die Rechnungen für die verschiedenen i sind voneinander unabhängig \implies leicht zu parallelisieren.
- In der Praxis gilt die Summe natürlich nur über die Einträge der i -ten Zeile von A , die $\neq 0$ sind.

Darstellung als lineares Verfahren

Sei $D := \text{diag}(a_{11}, \dots, a_{nn}) \in \mathbb{R}^{n \times n}$. Die Iterationsvorschrift lässt sich schreiben als

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

\implies lineares Verfahren mit $C = D^{-1}$.

Alternative Formulierung

- Sei $-L$ die Matrix aller Einträge von A unterhalb der Diagonalen
- Sei $-U$ die Matrix aller Einträge von A oberhalb der Diagonalen
- Also $A = D - L - U$
- Jacobi-Iteration:

$$Dx^{k+1} = (L + U)x^k + b$$

Konvergenz

Wir wenden das bekannte Konvergenzkriterium an:

Satz 8.2. *Das Jacobi Verfahren konvergiert genau dann, wenn $\rho(I - D^{-1}A) < 1$.*

Leider gilt diese Bedingung nicht immer.

Beispiel. Folgende Situation:

$$\begin{aligned} A &= \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \implies I - \underbrace{D^{-1}A}_{=I} = \begin{pmatrix} 0 & -2 \\ -2 & 0 \end{pmatrix} \\ \lambda_{1,2} &= \pm 2 \text{ sind Eigenwerte} \\ v_{1,2} &= \begin{pmatrix} \pm 1 \\ 1 \end{pmatrix} \text{ sind Eigenvektoren} \end{aligned}$$

$\implies \rho(I + D^{-1}A) = 2 > 1$. Das Verfahren konvergiert also *nicht*.

Es gibt schwächere Kriterien, die aber einfacher zu handhaben sind.

Definition. $A \in \mathbb{R}^{n \times n}$ heißt irreduzibel, falls es keine Permutationen der Zeilen und Spalten gibt, so dass A die Form

$$\begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{pmatrix}$$

bekommt, wobei $\tilde{A}_{11} \in \mathbb{R}^{k \times k}$, $1 \leq k < n$ (quadratisch) ist.

A heißt diagonaldominant, falls $|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|$ für $i = 1, \dots, n$ mit strikter Ungleichheit für mindestens ein i .

Satz 8.3. Das Jacobi-Verfahren konvergiert, falls mindestens eine der folgenden Bedingungen gilt:

- A ist symmetrisch positiv definit, und $2D - A$ ist auch symmetrisch positiv definit.
- A ist irreduzibel und diagonaldominant.

Anwendung auf das Poissonproblem

Die folgende Rechnung stammt aus Dahmen und Reusken [2, Beispiel 13.10].

Sei A die Matrix des Poisson-Problems. Gleichungssystem:

$$\frac{1}{h^2} (4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1}) = f_i.$$

Jacobi-Verfahren: $D = 4h^{-2}I$.

Wir versuchen jetzt, den Spektralradius von $I - CA = I - D^{-1}A$ abzuschätzen.

Dazu benutzen wir den *Rayleigh-Quotienten*

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

Für symmetrische A gilt $\lambda_{\min}(A) \leq R(A, x) \leq \lambda_{\max}(A)$, und diese Schranken werden angenommen, wenn x entsprechende Eigenvektoren sind.

Damit

$$\begin{aligned} \rho(I - D^{-1}A) &= \sup_{x \neq 0} \left| \frac{x^T (I - D^{-1}A) x}{\|x\|^2} \right| \\ &= \sup_{x \neq 0} \left| 1 - \frac{x^T D^{-1}A x}{\|x\|^2} \right| \\ &= \sup_{x \neq 0} \left| 1 - \frac{1}{4} h^2 \frac{x^T A x}{\|x\|^2} \right| \end{aligned}$$

Daraus folgt dass

$$\rho(I - D^{-1}A) = \sup \left\{ \left| 1 - \frac{1}{4} h^2 \lambda \right| : \lambda \text{ Eigenwert von } A \right\}$$

Für die spezielle Matrix können wir die Eigenwerte ausrechnen. Diese sind alle nichtnegativ.

Der größte Eigenwert von A ist [2, Kapitel 12.3.3]:

$$\lambda = \frac{8}{h^2} \sin^2 \left(\frac{1}{2} \pi h \right)$$

Es folgt, dass

$$\rho(I - D^{-1}A) = 1 - 2 \sin^2 \left(\frac{1}{2} \pi h \right) = \cos(\pi h)$$

Taylor-Entwicklung:

$$\cos(\pi h) = 1 - \frac{1}{2} \pi^2 h^2 + \dots$$

Also ist

$$\frac{\|e^{k+1}\|}{\|e^k\|} \approx \rho(I - D^{-1}A) \approx 1 - \frac{1}{2} \pi^2 h^2$$

Dieser Ausdruck geht „quadratisch“ (also ziemlich schnell) gegen 1 wenn $h \rightarrow 0$.

Wir wollen ausrechnen, wie viele Iterationen man ungefähr braucht, um den Anfangsfehler um einen Faktor R zu reduzieren. D.h., welches k soll man wählen, um ungefähr

$$\frac{\|e^k\|}{\|e^0\|} \leq \frac{1}{R}$$

zu erhalten?

Da $\frac{\|e^k\|}{\|e^0\|} \approx \rho^k$ erhält man

$$k = \log_{\rho} \frac{1}{R} = \frac{-\ln R}{\ln \rho(I - D^{-1}A)} \approx \frac{-\ln R}{\ln \left(1 - \frac{1}{2} \pi^2 h^2 \right)} \approx \frac{2}{\pi^2 h^2} \ln R,$$

da $\ln x \approx (x - 1) - \frac{1}{2}(x - 1)^2 + \dots$ ist.

Für ein doppelt so feines Gitter braucht man viermal so viele Iterationen (und diese sind natürlich auch noch teurer.).

Also ist dies kein so gutes Verfahren.

8.2.5 Das Gauß-Seidel-Verfahren

- Carl-Friedrich Gauß
- Philipp Ludwig von Seidel, 1821–1896, Mathematiker, Optiker, Astronom

Betrachte noch einmal die Jacobi-Rechenvorschrift:

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - a_{i1}x_1^k - a_{i2}x_2^k - \dots - a_{i,i-1}x_{i-1}^k - a_{i,i+1}x_{i+1}^k - \dots - a_{in}x_n^k \right).$$

Eigentlich haben wir für x_1, \dots, x_{i-1} schon bessere Werte als x_1^k, \dots, x_{i-1}^k , nämlich $x_1^{k+1}, \dots, x_{i-1}^{k+1}$.

Gauß-Seidel-Verfahren:

$$\begin{aligned} x_i^{k+1} &= \frac{1}{a_{ii}} \left(b_i - a_{i1}x_1^{k+1} - \dots - a_{i,i-1}x_{i-1}^{k+1} - a_{i,i+1}x_{i+1}^k - \dots - a_{in}x_n^k \right) \\ &= \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^n a_{ij}x_j^k \right). \end{aligned}$$

x_i^{k+1} hängt von x_{i-1}^{k+1} ab \implies keine Parallelisierung möglich.

Gauß-Seidel als lineares Verfahren

Seien $D, -L, -U$ Diagonale, linker, rechter Dreiecksteil von A .

$$x^{k+1} = D^{-1}(b + Lx^{k+1} + Ux^k)$$

Wir ziehen D und L auf die linke Seite

$$(D - L)x^{k+1} = Ux^k + b,$$

und lösen nach x^{k+1} auf

$$\begin{aligned} x^{k+1} &= (D - L)^{-1}Ux^k + (D - L)^{-1}b \\ &= x^k + [(D - L)^{-1}U - I]x^k + (D - L)^{-1}b \\ &= x^k + (D - L)^{-1} \underbrace{[U - D + L]}_{=-A} x^k + (D - L)^{-1}b \\ &= x^k + (D - L)^{-1}(b - Ax^k). \end{aligned}$$

\implies Lineares Verfahren mit $C = (D - L)^{-1}$.

Konvergenz

Wir erwarten bessere Konvergenzeigenschaften als für das Jacobi-Verfahren.

Und in der Tat:

Satz 8.4. *Das Gauß-Seidel-Verfahren konvergiert, wenn*

- *A symmetrisch und positiv definit ist und/oder*
- *A irreduzibel und diagonal dominant ist.*

Konvergenzgeschwindigkeit

Beispiel. Sei A die Matrix des Poisson-Problems für ein quadratisches Gebiet.

- Es gilt $\rho(I - (D - L)^{-1}A) = (\rho(I - D^{-1}A))^2$. Das heißt der Spektralradius der Jacobi-Methode ist das Quadrat des Spektralradius der Gauß-Seidel Methode. Laut Dahmen und Reusken [2] steht das bei [7].

- Deshalb

$$\rho(I - (D - L)^{-1}A) = \cos^2(\pi h)$$

- Taylor-Entwicklung für \cos^2

$$\cos^2 \pi h \approx \left(1 - \frac{1}{2}\pi^2 h^2 + \dots\right)^2 = 1 - 2\frac{1}{2}\pi^2 h^2 + \frac{1}{4}\pi^4 h^4 + \dots \approx 1 - \pi^2 h^2$$

- Fehlerreduktion

$$\frac{\|e^{k+1}\|}{\|e^k\|} \approx 1 - \pi^2 h^2$$

- Um den Startfehler um den Faktor R zu reduzieren, braucht man etwa

$$\frac{-\ln R}{\ln \rho(I - (D - L)^{-1}A)} \approx \frac{-\ln R}{\ln(1 - \pi^2 h^2)} \approx \frac{\ln R}{\pi^2 h^2}$$

Iterationen.

Faustregel: Das Gauß-Seidel-Verfahren braucht nur etwa halb so viele Iterationen wie das Jacobi-Verfahren.

8.2.6 Abbruchkriterien

Wie viele Iterationen soll man machen?

- Schätzungen wie „ $\frac{1}{\pi^2 h^2} \ln R$ “ sind nur für wenige Spezialfälle bekannt.

Idealerweise iteriert man so lange, bis $\|e^k\| < K$ mit K vorgegeben.

Wie sollte man $\|e^k\|$ ausrechnen/abschätzen?

Beliebter Ansatz: Betrachte das Residuum $r_k := b - Ax^k$. Es gilt

$$\|e^k\| = \|x^* - x^k\| = \|A^{-1}(b - Ax^k)\| = \|A^{-1}r_k\| \leq \|A^{-1}\| \cdot \|r_k\|.$$

Das Residuum schätzt den Fehler von oben ab, *wenn* $\|A^{-1}\|$ *bekannt ist.*

Abbruchbedingung $\|r_k\| < K$ ist nicht sinnvoll!

Stattdessen: Breche ab, sobald

$$\frac{\|r_k\|}{\|r^0\|} < K$$

Die Idee dahinter ist

$$\frac{\|e^k\|}{\|e^0\|} = \frac{\|A^{-1}r_k\|}{\|A^{-1}r^0\|} \approx \frac{\|r_k\|}{\|r^0\|}$$

Das ist aber nicht wirklich mathematisch zu rechtfertigen.

Ausgefilterter Ansatz: Um $\|e^k\|$ abzuschätzen:

- Berechne m weitere Iterationen,
- Schätze e^k durch $e^m - e^k$ ab.

8.3 Das Gradientenverfahren

(Auch bekannt als: Verfahren des steilsten Abstiegs, steepest descent, etc.)

Die folgenden Verfahren sind *nichtlinear*. Das heißt, dass die Fehlerfortpflanzung von einem Schritt zum nächsten nicht linear ist.

Der Inhalt dieses Kapitels ist weitestgehend dem Artikel von Shewchuk, *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain* [15] entnommen.

Ab jetzt sei A immer symmetrisch und positiv definit. Betrachte die Funktion

$$f: \mathbb{R}^n \rightarrow \mathbb{R}, \quad f(x) = \frac{1}{2}x^T Ax - bx.$$

Satz 8.5. Die Lösung x^* von $Ax = b$ ist eindeutiger Minimierer von f .

Beweis. 1. x^* ist stationärer Punkt von f , denn

$$f'(x) = \frac{1}{2}A^T x + \frac{1}{2}Ax - b = Ax - b \implies f'(x^*) = 0.$$

2. x^* ist Minimierer, denn für $p \neq x^*$ ergibt etwas Rechnen

$$f(p) = f(x^*) + \frac{1}{2} \underbrace{(p - x^*)^T A (p - x^*)}_{>0} > f(x^*),$$

da A positiv definit ist. □

Die Umformulierung des Gleichungssystems $Ax = b$ in ein Minimierungsproblem ermöglicht eine neue Sichtweise des Problems. Statt Lösungen eines Gleichungssystems können wir jetzt nach Minimierern einer Energie suchen.

8.3.1 Idee des Gradientenverfahrens

Idee: Ausgehend von x^k , mache einen Schritt in Richtung des steilsten Abstiegs.

Diese Richtung ist $-f'(x^k) = b - Ax^k = r_k$ (das Residuum).

Ein Schritt ist

$$x^{k+1} = x^k + \alpha^k r_k, \quad \alpha^k \in \mathbb{R} \text{ die Schrittlänge.} \quad (8.2)$$

Wie lang soll der Schritt sein?

Liniensuche: Minimiere f entlang der Suchrichtung.

Also

$$0 = \frac{d}{d\alpha} f(x^{k+1}) = f'(x^{k+1})^T \cdot \frac{dx^{k+1}}{d\alpha} = f'(x^{k+1})^T r_k.$$

Es ist aber $f'(x^{k+1}) = -r^{k+1}$, und deshalb

$$\begin{aligned} 0 &= (r^{k+1})^T r_k = (b - Ax^{k+1})^T r_k \\ &= (b - A(x^k + \alpha^k r_k))^T r_k = \underbrace{(b - Ax^k)^T}_{=r_k^T} r_k - \alpha^k (Ar_k)^T r_k \\ \implies \alpha^k &= \frac{r_k^T r_k}{r_k^T Ar_k} \end{aligned}$$

Die Berechnung von α^k besteht also (hauptsächlich) aus einer Matrix-Vektor-Multiplikation. Jeder Schritt ist orthogonal zu seinem Vorgänger.

8.3.2 Konvergenzanalyse

Zuerst ein einfacher Fall: Sei e^k Eigenvektor von A .

Dann ist r_k parallel zum Fehler e^k , denn

$$r_k = b - Ax^k = Ax^* - Ax^k = -Ae^k = -\lambda e^k$$

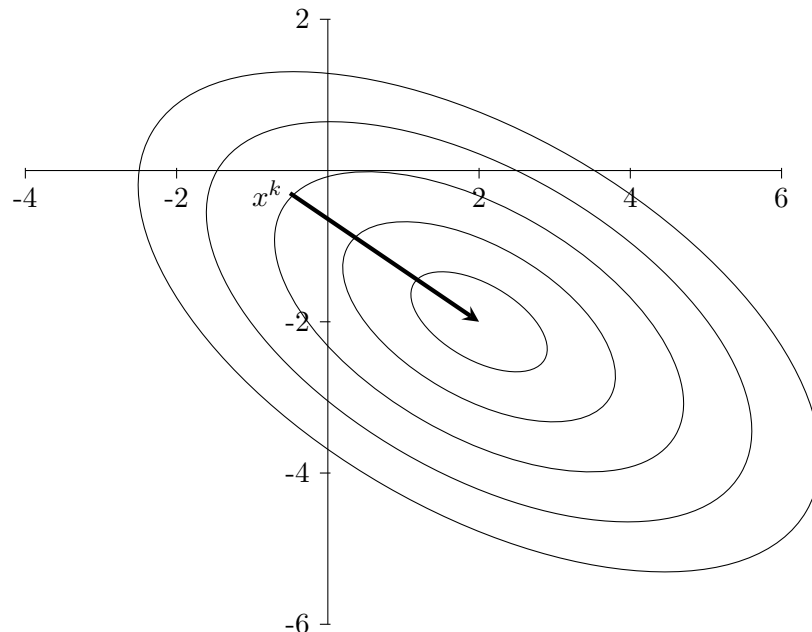
mit λ Eigenwert von A zu e^k .

Dann gilt:

$$e^{k+1} = e^k + \frac{r_k^T r_k}{r_k^T Ar_k} r_k = e^k + \frac{r_k^T r_k}{(r_k)^T \underbrace{A(-\lambda e^k)}_{=\lambda r_k}} \cdot (-\lambda e^k) = 0.$$

Das Verfahren konvergiert in einem Schritt.

Anschauung: x^k liegt auf einer Achse des Ellipsoids.



Allgemeiner: e^k ist Linearkombination von Eigenvektoren.
Sei $\{v_j\}$ Orthonormalbasis von Eigenvektoren

$$e^k = \sum_{j=1}^n \xi_j v_j.$$

(Zur Einfachheit lassen wir das k weg.)

Wir erhalten:

$$\begin{aligned} r^T r &= (-Ae)^T (-Ae) = \left(A \sum_i \xi_i v_i \right)^T \left(A \sum_j \xi_j v_j \right) \\ &= \left(\sum_i \xi_i \lambda_i v_i \right)^T \left(\sum_j \xi_j \lambda_j v_j \right) = \sum_j \xi_j^2 \lambda_j^2. \end{aligned}$$

Ebenso:

$$r^T A r = \sum_j \xi_j^2 \lambda_j^3$$

Der nächste Fehler ist damit

$$e^{k+1} = e^k + \frac{r_k^T r_k}{r_k^T A r_k} r_k = e^k + \frac{\sum_j \xi_j^2 \lambda_j^2}{\sum_j \xi_j^2 \lambda_j^3} r_k.$$

Beachte: Falls alle λ_j gleich sind, so sind wir wieder in einem Schritt fertig, da dann $r_k = -\lambda e^k$.

Anschauung dazu: Das Funktional ist dann kugelsymmetrisch.

Der folgende Satz beschreibt den allgemeinen Fall. Den Beweis findet man bei Shewchuk [15].

Satz 8.6 ([15, Kapitel 6]). *Sei κ die Kondition der Matrix A . Dann gilt nach k Schritten des Gradientenverfahrens*

$$\|e^k\|_A \leq \left(\frac{\kappa - 1}{\kappa + 1} \right)^k \|e^0\|_A,$$

wobei $\|\cdot\|_A$ die Energienorm ist.

8.4 Das Verfahren der konjugierten Gradienten (CG)

(Ursprünglich vorgestellt von Hestenes und Stiefel im Jahr 1952 [10])

Wir halten fest:

- Das Gradientenverfahren minimiert häufig mehrfach in ähnliche Richtungen.

Besser wäre doch:

1. n orthogonale Suchrichtungen d_1, \dots, d_n
2. Bei jedem Schritt könnten wir die richtige Schrittweite α^k bestimmen.

Damit hätten wir die exakte Lösung nach n Schritten.

Problem: 2) heißt gerade: e^{k+1} muss senkrecht auf der Suchrichtung d_k stehen. Bestimme α :

$$0 = d_k^T e^{k+1} = d_k^T (e^k + \alpha^k d_k) \iff \alpha^k = -\frac{d_k^T e^k}{d_k^T d_k}.$$

Geht nicht, denn e^k ist unbekannt!

Stattdessen: *Gute Idee Nr. 1:* Wähle stattdessen A -orthogonale Suchrichtungen (auch *konjugierte* Suchrichtungen).

Bestimme α^k so, dass d_k und e^{k+1} A -orthogonal sind:

$$\begin{aligned} d_k^T A e^{k+1} &= d_k^T A (e^k + \alpha^k d_k) = 0 \\ \implies \alpha^k &= -\frac{d_k^T A e^k}{d_k^T A d_k} = \frac{d_k^T r_k}{d_k^T A d_k}. \end{aligned}$$

Dabei haben wir benutzt dass

$$r_k = b - A x^k = A(A^{-1}b - x^k) = A(x^* - x^k) = -A e^k.$$

Lemma 8.1. *Auch mit A -orthogonalen Suchrichtungen ist man nach n Schritten fertig.*

Beweis. Schreibe Anfangsfehler e^0 als Linearkombination der Suchrichtungen

$$e^0 = \sum_{j=1}^n \delta_j d_j. \tag{8.3}$$

Gesucht ist eine Formel für δ_k . Multipliziere (8.3) mit $d_k^T A$. Man erhält

$$\begin{aligned} d_k^T A e^0 &= \sum_{j=1}^n \delta_j d_k^T A d_j = \delta_k d_k^T A d_k \\ \implies \delta_k &= \frac{d_k^T A e^0}{d_k^T A d_k} = \frac{d_k^T A (e^0 + \sum_{i=1}^{k-1} \alpha^i d_i)}{d_k^T A d_k} \\ &= \frac{d_k^T A e^k}{d_k^T A d_k} = -\alpha^k. \end{aligned}$$

Bei jedem Schritt wird genau ein Summand aus der Fehlerdarstellung (8.3) entfernt. \implies fertig nach n Schritten, da dann $e^n = 0$. \square

8.4.1 Das Gram-Schmidt-Verfahren

Wie erzeugt man A -orthogonale Richtungen?

Wir erklären jetzt, wie man n A -orthogonale Suchrichtungen konstruieren kann. Im CG-Verfahren passiert das synchron zum eigentlichen Suchen des Minimierers. Es wird *nicht* zuerst die Menge der Suchrichtungen konstruiert, und dann erst eine nach der anderen zur Suche genommen.

- Seien u_1, \dots, u_n linear unabhängige Vektoren.
- Setze:

$$d_i = u_i + \sum_{j=1}^{i-1} \beta_{ij} d_j, \quad \beta_{ij} = -\frac{u_i^T A d_j}{d_j^T A d_j}. \quad (8.4)$$

Funktioniert, aber:

1. Man muss sich alle d_j merken $\rightarrow \mathcal{O}(n^2)$ Speicherverbrauch
2. Benötigt $\mathcal{O}(n^3)$ Rechenoperationen. Das ist in etwa so viel wie wie das Gauß-Seidel-Verfahren, also zu viel.

8.4.2 Das Verfahren der konjugierten Gradienten

(Eigentlich ein schlechter Name: Es kommen keine konjugierten Gradienten vor.)

Gute Idee Nr. 2: Wähle $u_i = r^i$ für $i = 1, \dots, n$.

- Warum ist das eine gute Idee?
- Geht das überhaupt?
- Bilden die r^i eine linear unabhängige Menge?

Bemerkung: Wie kann das gehen? Zum Berechnen der Residuen r^i brauchen wir die Suchrichtungen, und jetzt sollen wir umgekehrt die Residuen zur Berechnung der Suchrichtungen brauchen? Der Trick: Wir berechnen beide abwechselnd. Aus der Startiterierten x^0 folgt das erste Residuum. Damit kann die erste Suchrichtung berechnet werden. Damit berechnen wir x^1 und damit das zweite Residuum. Damit dann die nächste Richtung usw.

Lemma 8.2. $d_l^T r_i = 0$ für alle $l < i$.

Beweis. Es gilt

$$e^i = e^0 + \sum_{j=0}^{i-1} \alpha^j d_j = \sum_{j=0}^n \delta_j d_j - \sum_{j=0}^{i-1} \delta_j d_j = \sum_{j=i}^n \delta_j d_j.$$

Multipliziere beide Seiten mit $-d_l^T A$:

$$-d_l^T A e^i = -\sum_{j=i}^n \delta_j d_l^T A d_j.$$

Die linke Seite ist $d_l^T r^i$.

Die rechte Seite ist 0, da die d_i A -orthogonal sind. \square

Nicht nur ist r^i orthogonal zu allen Suchrichtungen d_l mit $l < i$, es ist auch senkrecht auf allen Residuen r^l mit $l < i$! Deshalb bilden die r_k eine linear unabhängige Menge; das Gram-Schmidt-Verfahren kann also angewandt werden.

Lemma 8.3. r^i ist orthogonal zu r^l falls $l < i$.

Beweis. Gram-Schmidt-Formel

$$d_l = r^l + \sum_{k=0}^{l-1} \beta_{lk} d_k.$$

Multipliziere von rechts mit r^i :

$$\underbrace{d_l^T r^i}_{=0} = (r^l)^T r^i + \sum_{k=0}^{l-1} \beta_{lk} \underbrace{d_k^T r^i}_{=0}.$$

Es folgt

$$(r^l)^T r^i = 0.$$

Die Residuen r^1, \dots, r^n sind linear unabhängig. \square

Es passiert etwas magisches!

Lemma 8.4. Fast alle β_{ij} verschwinden! Gram-Schmidt wird billig.

Beweis. • $r^{j+1} = -Ae^{j+1} = -A(e^j + \alpha^j d_j) = r^j - \alpha^j A d_j$

- Multipliziere von links mit r_i^T

$$\alpha^j (r_i^T A d_j) = \underbrace{r_i^T r_j}_{=0 \Leftarrow i \neq j} - \underbrace{r_i^T r_{j+1}}_{=0 \Leftarrow i \neq j+1}$$

Wegen Lemma 8.3 ist der Term auf der rechten Seite fast immer Null! Genauer:

$$r_i^T A d_j = \begin{cases} \frac{1}{\alpha^i} (r^i)^T r^i & \text{falls } i = j, \\ -\frac{1}{\alpha^{i-1}} (r^i)^T r^i & \text{falls } i = j + 1, \\ 0 & \text{sonst.} \end{cases}$$

- Gram-Schmidt-Koeffizienten:

$$\beta_{ij} = -\frac{(r^i)^T Ad_j}{d_j^T Ad_j} = \begin{cases} \frac{1}{\alpha^{i-1}} \frac{(r^i)^T r^i}{d_j^T Ad_j} & \text{falls } i = j + 1, \\ 0 & \text{sonst.} \end{cases}$$

Der Fall $i = j$ tritt im Gram-Schmidt-Verfahren nicht auf! □

Formel (8.4) reduziert sich:

$$\text{Aus } d_i = u_i + \sum_{j=1}^{i-1} \beta_{ij} d_j \quad \text{wird} \quad d_i = u_i + \beta_{i,i-1} d_{i-1}. \quad (8.5)$$

- Fast alle β_{ij} sind Null!
- Deshalb: Einfachere Notation: Schreibe $\beta_{(i)}$ statt $\beta_{i,i-1}$.
- Es werden nicht mehr alle d_i benötigt, um die nächste Richtung auszurechnen.
- Speicheraufwand und Rechenzeit geht von $\mathcal{O}(n^2)$ nach $\mathcal{O}(n \cdot \text{Anzahl der Einträge von } A)$.

Wir können die Darstellung von $\beta_{(i)}$ noch weiter vereinfachen.

- Setze zunächst $\alpha^i = \frac{d_i^T r^i}{d_i^T Ad_i}$ in die Definition von $\beta_{i,i-1}$ ein. Man erhält

$$\beta_{(i)} = \frac{(r^i)^T r^i}{d_{i-1}^T r_{i-1}}$$

- Aus (8.5) folgt

$$d_{i-1}^T r_{i-1} = u_{i-1}^T r_{i-1} + \beta_{i-1,i-2} \underbrace{d_{i-2}^T r_{i-1}}_{=0} = u_{i-1}^T r_{i-1} = r_{i-1}^T r_{i-1}.$$

- Damit erhält man

$$\beta_{(i)} = \frac{(r^i)^T r^i}{r_{i-1}^T r_{i-1}}.$$

8.4.3 Das komplette Verfahren

- Berechne $d_0 = r_0 = b - Ax^0$.

- Für $k = 1, 2, 3, \dots$

$$\begin{aligned}\alpha^k &= \frac{r_k^T r_k}{d_k^T A d_k} \\ x^{k+1} &= x^k + \alpha^k d^k \\ r^{k+1} &= b - Ax^{k+1} = r_k - \alpha^k A d_k \\ \beta_{(k+1)} &= \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \\ d_{k+1} &= r_{k+1} + \beta_{(k+1)} d_k\end{aligned}$$

Direkter Löser mit Komplexität $\mathcal{O}(n \cdot \text{Anzahl der Einträge von } A)$. Einige Fakten zu CG

- Zuerst vorgeschlagen von Magnus Hestenes und Eduard Stiefel im Jahr 1952
- Toll: ein direkter Löser für dünnbesetzte lineare GS mit Komplexität $\mathcal{O}(n \cdot \text{Anzahl der Einträge von } A)$.
- Funktioniert in der Praxis aber schlecht: Rundungsfehler zerstören A -Orthogonalität der Richtungen, man hat nach n Iterationen also *nicht* die Lösung
- Geriet zwischenzeitlich in Vergessenheit
- Erlebte Revival als iterative Methode

8.4.4 Interpretation als Krylov-Verfahren

CG hat weitere interessante Eigenschaften.

Die Folge der Suchrichtungen d_0, d_1, \dots definiert eine Folge von Räumen

$$\begin{aligned}\mathcal{D}_0 &= \text{span}\{d_0\} \\ \mathcal{D}_1 &= \text{span}\{d_0, d_1\} \\ \mathcal{D}_2 &= \text{span}\{d_0, d_1, d_2\} \\ &\vdots\end{aligned}$$

Satz 8.7. *Das CG-Verfahren wählt x_k so aus dem Raum $e_0 + \mathcal{D}_k$, dass $\|e_k\|_A$ minimal ist.*

[Dies ist auch eine alternative Motivation des Verfahrens.]

Alternative Charakterisierung der \mathcal{D}_k

$$\begin{aligned}\mathcal{D}_k &= \text{span}\{d_0, A d_0, A^2 d_0, \dots, A^k d_0\} \\ &= \text{span}\{r_0, A r_0, A^2 r_0, \dots, A^k r_0\}.\end{aligned}$$

- Solche Räume heißen *Krylov-Räume*.¹
- CG heißt deshalb auch ein „Krylov-Verfahren“.
- Es gibt noch weitere Krylov-Raum-basierte Verfahren, z.B. BiCGStab, MinRes, GMRes.

8.4.5 Konvergenz des CG-Verfahren als iterativem Verfahren

Satz 8.8. Für alle $k = 1, \dots, n$ hat der Fehler e^k die Darstellung

$$e^k = \left(I + \sum_{j=1}^k \psi_j A^j \right) e^0,$$

wobei die Koeffizienten $\psi_1, \dots, \psi_k \in \mathbb{R}$ von α^i und $\beta_{(i)}$ für $i = 1, \dots, k$ abhängen.

Hauptidee:

- CG minimiert $\|e_k\|_A$
- Der Ausdruck in der Klammer ist ein Polynom in A , also

$$e^k = P_k(A)e^0$$

- Interpretation von CG:
 1. CG wählt die Koeffizienten $\alpha_i, \beta_{(i)}$
 2. CG konstruiert das Polynom $P_k(A)$

Wie wirkt $P_k(A)$ auf e^0 ?

- Schreibe e^0 in orthonormaler Eigenvektor-Basis

$$e^0 = \sum_{j=1}^n \xi_j v_j$$

- Daraus folgt

$$\begin{aligned} e^k &= \left(I + \sum_{i=1}^k \psi_i A^i \right) \sum_{j=1}^n \xi_j v_j \\ &= \sum_{j=1}^n \xi_j \left(I + \sum_{i=1}^k \psi_i A^i \right) v_j \\ &= \sum_{j=1}^n \xi_j \left(1 + \sum_{i=1}^k \psi_i \lambda_j^i \right) v_j \\ &= \sum_{j=1}^n \xi_j P_k(\lambda_j) v_j \end{aligned}$$

¹Nach Alexei Nikolajew Krylow, 1863-1945

- Multiplikation von links mit A

$$Ae_k = \sum_{j=1}^n \xi_j P_k(\lambda_j) \lambda_j v_j$$

$$\|e_k\|_A^2 = e_k^T (Ae_k) = \left(\sum_{i=1}^n \xi_i P_k(\lambda_i) v_i \right) \left(\sum_{j=1}^n \xi_j P_k(\lambda_j) \lambda_j v_j \right) = \sum_{j=1}^n \xi_j^2 (P_k(\lambda_j))^2 \lambda_j$$

- Das CG-Verfahren minimiert also

$$\|e_k\|_A^2 = \min_{\tilde{P}_k, \tilde{P}_k(0)=1} \sum_{j=1}^n \xi_j^2 \tilde{P}_k(\lambda_j)^2 \lambda_j$$

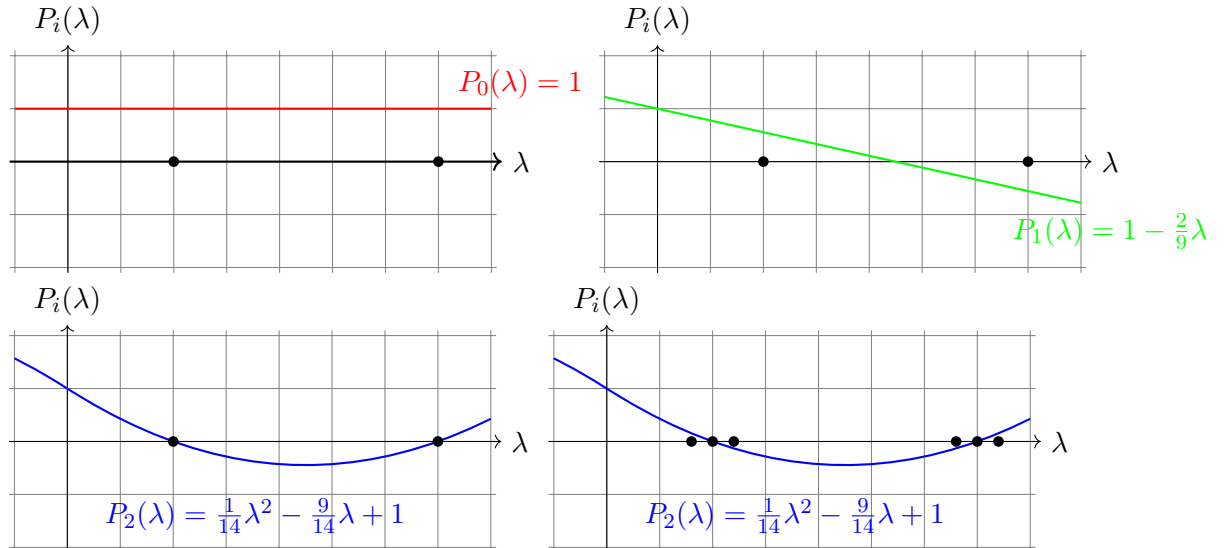
$$\leq \min_{\tilde{P}_k, \tilde{P}_k(0)=1} \max_{\lambda \in \sigma(A)} (\tilde{P}_k(\lambda))^2 \underbrace{\sum_{j=1}^n \xi_j^2 \lambda_j}_{\|e^0\|_A^2}$$

Dabei ist $\sigma(A)$ das *Spektrum* von A , d.h. die Menge aller Eigenwerte.

Die Aufgabe lautet also:

- Finde Polynom P_k mit $P_k(0) = 1$, dessen Werte für Eigenwerte λ möglichst klein sind.

Beispiel. $A \in \mathbb{R}^{2 \times 2}$ mit $\sigma(A) = \{2, 7\}$, also $\lambda_1 = 2, \lambda_2 = 7$.



Schnelle Konvergenz, wenn

- es ein Polynom niedrigen Grades gibt, dass bei allen Eigenwerten von A niedrige Werte annimmt.

- Eigenwerte von A in Haufen auftreten
- viele Eigenwerte mehrfach auftreten

Schlimmst-möglicher Fall:

- Eigenwerte sind gleichverteilt in $[\lambda_{\min}, \lambda_{\max}]$.
- Wenig doppelte Eigenwerte.

Allgemein ist zuwenig über die Eigenwerte von A bekannt.

Ansatz: Anstelle das Maximum von P_k über alle Eigenwerte von A zu minimieren, minimieren wir das Maximum von P_k auf ganz $[\lambda_{\min}, \lambda_{\max}]$.

$$\|e^k\|_A^2 \leq \min_{\tilde{P}_k, \tilde{P}_k(0)=1} \max_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} (\tilde{P}_k(\lambda))^2 \|e^0\|_A^2$$

Definition. Das Tschebyshev-Polynom vom Grad $i \in \mathbb{N}$ ist

$$T_i(s) = \frac{1}{2} \left[(s + \sqrt{s^2 - 1})^i + (s - \sqrt{s^2 - 1})^i \right].$$

Satz 8.9. Es gilt

$$|T_i(s)| \leq 1 \quad \text{für alle } s \in [-1, 1],$$

und T_i ist „maximal außerhalb von $[-1, 1]$ “ unter allen Polynomen mit dieser Eigenschaft.

Umskalieren:

Lemma 8.5. Das Polynom

$$\tilde{T}_i(\lambda) = \frac{T_i\left(\frac{\lambda_{\max} + \lambda_{\min} - 2\lambda}{\lambda_{\max} - \lambda_{\min}}\right)}{T_i\left(\frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}}\right)}$$

oszilliert auf $[\lambda_{\min}, \lambda_{\max}]$ zwischen

$$\pm T_i \left(\frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} \right)^{-1},$$

und erfüllt $\tilde{T}_i(0) = 1$.

Damit können wir den Fehler abschätzen:

$$\begin{aligned} \|e^k\|_A &\leq \min_{\tilde{P}_k, \tilde{P}_k(0)=1} \max_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} |P_k(\lambda)| \cdot \|e^0\|_A \\ &\leq \max_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} |\tilde{T}_k(\lambda)| \cdot \|e^0\|_A \\ &\leq T_i \left(\frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} \right)^{-1} \cdot \|e^0\|_A \end{aligned}$$

Da A s.p.d. ist gilt $\kappa = \kappa(A) = \frac{|\lambda_{\max}|}{|\lambda_{\min}|} = \frac{\lambda_{\max}}{\lambda_{\min}}$, und deshalb

$$\begin{aligned} \|e^k\|_A &= T_i \left(\frac{\kappa + 1}{\kappa - 1} \right)^{-1} \|e^0\|_A \\ &= 2 \left[\left(\frac{\sqrt{\kappa} + 1}{\sqrt{\kappa} - 1} \right)^k + \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \right]^{-1} \cdot \|e^0\|_A \end{aligned}$$

Der zweite Summand geht gegen 0 für $k \rightarrow \infty$. Deshalb

$$\|e^k\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \cdot \|e^0\|_A$$

- Vergleiche mit Gradientenverfahren. Dort:

$$\|e^k\|_A \leq \left(\frac{\kappa - 1}{\kappa + 1} \right)^k \|e^0\|_A$$

- Das ist langsamer als für das CG-Verfahren.
- Diese Abschätzung für CG ist aber sehr schwach. In vielen Fällen konvergiert das Verfahren deutlich besser!

8.5 Vorkonditionierung

Die Konvergenzrate von CG (und diversen anderen Verfahren) hängt von der Kondition von A ab.

8.5.1 Idee der Vorkonditionierung

Wähle eine Matrix W , die A „gut approximiert“, und betrachte das Gleichungssystem

$$W^{-1}Ax = W^{-1}b.$$

- Gleiche Lösung wie $Ax = b$.
- Bei geschickt gewähltem W ist $\kappa(W^{-1}A) \ll \kappa(A)$

Löse $W^{-1}Ax = W^{-1}b$ mit dem CG-Verfahren.

Problem: Das CG-Verfahren kann nur verwendet werden, wenn $W^{-1}A$ s.p.d. ist, aber aus W und A s.p.d. folgt *nicht*, dass $W^{-1}A$ s.p.d. ist.

Trick: Wähle W s.p.d.

Dann existiert die Cholesky-Zerlegung

$$W = L_1 L_1^T = LDL^T$$

mit

- D Diagonalmatrix
- L untere Dreiecksmatrix mit Einsen auf der Diagonalen
- $L_1 = LD^{\frac{1}{2}}$ untere Dreiecksmatrix

Statt $Ax = b$ betrachte

$$\underbrace{L_1^{-1}AL_1^{-T}}_{=: \tilde{A}} \underbrace{L_1^T x}_{=: \tilde{x}} = \underbrace{L_1^{-1}b}_{=: \tilde{b}}$$

also

$$\tilde{A}\tilde{x} = \tilde{b}.$$

Die neue Matrix \tilde{A} ist s.p.d., es gilt sogar:

Satz 8.10. $W^{-1}A$ und $\tilde{A} = L_1^{-1}AL_1^{-T}$ haben die gleichen Eigenwerte.

Beweis. Sei v Eigenvektor von $W^{-1}A$, dann ist $L_1^T v$ Eigenvektor von \tilde{A} zum selben Eigenwert. \square

Da \tilde{A} s.p.d. ist können wir CG verwendet werden:

$$\tilde{d}_0 = \tilde{r}^0 = \tilde{b} - \tilde{A}\tilde{x}^0 = L_1^{-1}b - L_1^{-1}AL_1^{-T}\tilde{x}^0 = L_1^{-1}(b - Ax^0)$$

Für $k = 1, 2, 3, \dots$

$$\begin{aligned} \alpha^k &= \frac{\tilde{r}_k^T \tilde{r}_k}{\tilde{d}_k L_1^{-1} A L_1^{-T} \tilde{d}_k} \\ \tilde{x}^{k+1} &= \tilde{x}^k + \alpha^k \tilde{d}_k \\ \tilde{r}_{k+1} &= \tilde{r}_k - \alpha^k L_1^{-1} A L_1^{-T} \tilde{d}_k \\ \beta^{(k+1)} &= \frac{\tilde{r}_{k+1}^T \tilde{r}_{k+1}}{\tilde{r}_k^T \tilde{r}_k} \\ \tilde{d}_{k+1} &= \tilde{r}_{k+1} + \beta^{(k+1)} \tilde{d}_k \end{aligned}$$

Zu teuer: L_1 muss bekannt sein!

Stattdessen: Setze $d_k = L_1^{-T} \tilde{d}_k$.

Das umgeformte Verfahren ist:

$$d_0 = L_1^{-T} \tilde{d}_0 = L_1^{-T} L_1^{-1} (b - Ax^0) = W^{-1} r^0$$

Für $k = 1, 2, 3, \dots$

$$\begin{aligned} \alpha^k &= \frac{r_k^T W^{-1} r_k}{d_k^T A d_k} \\ x^{k+1} &= x^k + \alpha^k d^k \\ r_{k+1} &= r_k - \alpha^k A d_k \\ \beta^{(k+1)} &= \frac{r_{k+1}^T W^{-1} r_{k+1}}{r_k^T W^{-1} r_k} \\ d_{k+1} &= W^{-1} r_{k+1} + \beta^{(k+1)} d_k \end{aligned}$$

Mit einem Wort: Wir nehmen ersetzen im normalen CG-Verfahren einfach überall r durch $W^{-1}r$.

Dieses Verfahren heißt *Vorkonditioniertes CG-Verfahren*. Es funktioniert gut, wenn

1. $\kappa(W^{-1}A)$ klein ist
2. Die Lösung von $Wz = r_k$ billig berechnet werden kann.

8.5.2 Unvollständige Cholesky-Zerlegung (ICH,ILU,...)

Für die Wahl von W sind extrem viele verschiedene Möglichkeiten vorgeschlagen worden.

Wir zeigen zwei wichtige Ansätze.

Idee: Die beste Kondition bekäme man natürlich für $W = A$.

Zur Lösung von $Wz = Az = r_k$ Cholesky-Zerlegung $A = LDL^T$ berechnen (viele GS mit fester Matrix).

- Selbst für dünnbesetzte A ist L aber vollbesetzt
 \implies Lösen mit Cholesky-Zerlegung ist zu teuer und braucht zu viel Speicher.

Definition. Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit. Eine unvollständige Cholesky-Zerlegung von A ist $A \approx \tilde{L}\tilde{D}\tilde{L}^T$, wobei

- \tilde{L} : normierte untere Dreiecksmatrix
- \tilde{D} : Diagonalmatrix
- $\tilde{L}_{ij} = 0$ falls $A_{ij} = 0$

Vorkonditionierer: $W = \tilde{L}\tilde{D}\tilde{L}^T$

- häufig: $\kappa(W^{-1}A) \ll \kappa(A)$
- \tilde{L} ist dünnbesetzt: $\implies \tilde{L}\tilde{D}\tilde{L}^T z^k = r_k$ ist billig zu lösen.

Konstruktion der unvollständigen Cholesky-Zerlegung

Statt $\tilde{L}\tilde{D}\tilde{L}^T$ berechnen wir $\tilde{L}\tilde{R}$.

Dann ist nämlich $\tilde{R} = \tilde{D}\tilde{L}^T$ mit $\tilde{D} = \text{diag } \tilde{R}$.

Sei $A = LR$ mit L, R Dreiecksmatrizen, L normiert.

Dann ist

$$A_{ik} = \sum_{j=1}^n L_{ij}R_{jk} = \sum_{j=1}^i L_{ij}R_{jk} = \underbrace{L_{ii}}_{=1} R_{ik} + \sum_{j=1}^{i-1} L_{ij}R_{jk}$$

Damit kann man den Eintrag R_{ik} berechnen:

$$R_{ik} = A_{ik} - \sum_{j=1}^{i-1} L_{ij} R_{jk} \quad \text{da } L_{ii} = 1, \quad 1 \leq i < k \leq n$$

Ähnlich:

$$L_{ik} = R_{ii}^{-1} \left(A_{ik} - \sum_{j=1}^{k-1} L_{ij} R_{jk} \right) \quad 1 \leq k < i \leq n$$

Mit diesen Formeln kann man eine LR-Zerlegung berechnen.

```

1 input:  $A \in \mathbb{R}^{n \times n}$ 
2 Setze  $L = I \in \mathbb{R}^{n \times n}, R = 0 \in \mathbb{R}^{n \times n}$ 
3 for  $i = 1, 2, \dots, n$  do
4   for  $k = 1, \dots, i - 1$  do
5      $L_{ik} = R_{kk}^{-1} \left( A_{ik} - \sum_{j=1}^{k-1} L_{ij} R_{jk} \right)$ 
6   end
7   for  $k = i, \dots, n$  do
8      $R_{ik} = A_{ik} - \sum_{j=1}^{i-1} L_{ij} R_{jk}$ 
9   end
10 end

```

Um stattdessen eine *unvollständige* LR-Zerlegung zu berechnen lassen wir einfach alle Einträge i, j aus, für die $A_{ij} = 0$ gilt.

```

1 input:  $A \in \mathbb{R}^{n \times n}$ 
2 Setze  $\tilde{L} = I \in \mathbb{R}^{n \times n}, \tilde{R} = 0 \in \mathbb{R}^{n \times n}$ 
3 for  $i = 1, 2, \dots, n$  do
4   foreach  $k = 1, \dots, i - 1$  with  $A_{ik} \neq 0$  do
5      $\tilde{L}_{ik} = \tilde{R}_{kk}^{-1} \left( A_{ik} - \sum_{j=1}^{k-1} \tilde{L}_{ij} \tilde{R}_{jk} \right)$    Summe nur über das Muster
6   end
7   foreach  $k = i, \dots, n$  with  $A_{ik} \neq 0$  do
8      $\tilde{R}_{ik} = A_{ik} - \sum_{j=1}^{i-1} \tilde{L}_{ij} \tilde{R}_{jk}$    Summe nur über das Muster
9   end
10 end

```

Beispiel. Poisson-Problem. CG vs. CG mit ILR-Vorkonditionierer

| h | $\frac{1}{40}$ | $\frac{1}{80}$ | $\frac{1}{160}$ | $\frac{1}{320}$ |
|--------|----------------|----------------|-----------------|-----------------|
| CG | 65 | 130 | 262 | 525 |
| ILR-CG | 20 | 40 | 79 | 157 |

Tabelle: Anzahl der Iterationen um das Residuum um den Faktor 1000 zu reduzieren.
Es sind viele Varianten möglich!

8.5.3 Lineare Verfahren als Vorkonditionierer

Erinnerung: Lineares Verfahren

$$x^{k+1} = x^k + C(b - Ax^k)$$

Sei

1. A symmetrisch und positiv definit
2. C so, dass $\rho(I - CA) < 1$, d.h. das Verfahren konvergiert.

Wir hatten C als Approximation von A^{-1} interpretiert.

Idee: Wähle $W^{-1} = C$ als Vorkonditionierer für CG.

Praktische Umsetzung: Für CG müssen Ausdrücke der Form $z = W^{-1}r_k = Cr_k$ berechnet werden.

Problem: C ist i. A. nicht explizit gegeben.

Lösung: Betrachte das lineare Gleichungssystem $Az = r_k$

- Setze $z^0 = 0 \in \mathbb{R}^n$
- Ein Schritt des linearen Verfahrens:

$$z^1 = z^0 + C(r_k - Az^0) = Cr_k$$

Viele lineare Verfahren können als Vorkonditionierer verwendet werden!

- Z. B.: Der Jacobi-Vorkonditionierer: $C = \text{diag}(A)^{-1}$
- Viele lineare Verfahren werden überhaupt nur betrachtet, um als Vorkonditionierer zu dienen (z. B. Mehrgitterverfahren, Gebietszerlegungsverfahren).

Ein Problem noch: Wie steht es z. B. mit Gauß-Seidel

$$C = (D - L)^{-1} \quad ?$$

- Funktioniert nicht, denn $W = C^{-1} = D - L$ ist nicht symmetrisch.

Stattdessen: Symmetrischer Gauß-Seidel

- Abwechselnd

- Vorwärtsiteration:

$$x^{k+\frac{1}{2}} = x^k + (D - L)^{-1}(b - Ax^k)$$

- Rückwärtsiteration:

$$x^{k+1} = x^{k+\frac{1}{2}} + (D - R)^{-1}(b - Ax^{k+\frac{1}{2}})$$

- Einsetzen:

$$x^{k+1} = x^k + \underbrace{\left[(D - L)^{-1} + (D - R)^{-1} - (D - R)^{-1}A(D - L)^{-1} \right]}_{=C}(b - Ax^k)$$

Dieses C will man sicher nicht explizit ausrechnen.

- Aber: $C^{-1} = W$ ist s.p.d.!

Auch im linearen Verfahren selbst kann man C als Vorkonditionierer interpretieren.

- Lineares Gleichungssystem $Ax = b$

- Richardson-Verfahren

$$x^{k+1} = x^k + (b - Ax^k)$$

- Vorkonditionierung: Wähle ein W als Approximation von A . Betrachte

$$W^{-1}Ax = W^{-1}b$$

Statt W^{-1} schreibe C :

$$CAx = Cb$$

- Richardson-Iteration dafür:

$$x^{k+1} = x^k + (Cb - CAx^k) = x^k + C(b - Ax^k).$$

9 Direkte Lösungsverfahren für dünnbesetzte Gleichungssysteme

Problem: Sei $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$.

Finde $x \in \mathbb{R}^n$, sodass $Ax = b$.

Dabei ist A *sehr groß*, dafür aber dünnbesetzt.

Bisher: Iterative Verfahren. Funktionieren, im Prinzip, aber

- konvergieren nicht für jede invertierbare Matrix A ,
- Konvergenzgeschwindigkeit eventuell gering; hängt von der Kondition ab,
- Abbruchkriterien nötig, algebraischer Fehler

Alternative: direkte Löser:

- Bestimmen die exakte Lösung x nach endlich vielen Schritten (exakte Arithmetik vorausgesetzt)
- Beispiel: Gauß-Elimination; funktioniert, nützt aber die Dünnbesetztheit von A nicht aus. Deshalb:
 - Hohe Zeitkomplexität ($\mathcal{O}(n^3)$ Schritte)
 - Großer Speicheraufwand

Das Beste beider Welten: Direkte Löser für dünn besetzte Gleichungssysteme.

- Varianten von Gauß- und Cholesky-Elimination
- teilweise sehr kompliziert
- Zeit- und Speicheraufwand deutlich besser als bei Gauß-Verfahren
- Werden in der Praxis sehr häufig verwendet.
- Geeignet für Systeme bis ca. 10^5 Unbekannte, danach zu großer Speicheraufwand

9.1 Die Multifrontale Methode

Direktes Lösungsverfahren für dünnbesetzte Matrizen.

- Verfahren nach Duff und Reid [6] (1983)
- Implementiert z.B. in Matlab, Octave, UMFPack
- Wir behandeln nur den Fall das A symmetrisch und positiv definit ist.
- Unsere Darstellung folgt Liu: “The Multifrontal Method for Sparse Matrix Solution: Theory and Practice”, SIAM Review, 1992 [11]

9.1.1 Cholesky-Zerlegung

Die multifrontale Methode basiert auf der Cholesky-Zerlegung.

Sei A s.p.d. und dünnbesetzt.

Ziel: Berechne die Cholesky-Zerlegung

$$A = LL^T, \quad L \text{ untere Dreiecksmatrix,} \quad L_{ii} > 0 \quad \forall i = 1, \dots, n.$$

Wir wiederholen kurz die Cholesky-Zerlegung für vollbesetzte Matrizen.

Schreibe dafür A in Blockform

$$A = \begin{pmatrix} B & V^T \\ V & C \end{pmatrix}, \quad B \in \mathbb{R}^{(j-1) \times (j-1)}.$$

Dann existiert die Zerlegung

$$A = \begin{pmatrix} L_B & 0 \\ VL_B^{-T} & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & C - VB^{-1}V^T \end{pmatrix} \begin{pmatrix} L_B^T & L_B^{-1}V^T \\ 0 & I \end{pmatrix}.$$

Dabei ist L_B der Cholesky-Faktor von B .

- Dieser existiert, da B ja s.p.d. ist.

Lemma 9.1. Die $n \times (j-1)$ -Matrix $\begin{pmatrix} L_B \\ VL_B^{-T} \end{pmatrix}$ besteht aus den ersten $j-1$ Spalten von L .

Die Zerlegung lässt sich rekursiv fortsetzen, denn $C - VB^{-1}V^T$ ist ebenfalls s.p.d.

Beweis der Definitheit. Sei $u \in \mathbb{R}^{n-j+1}$, $u \neq 0$. Dann ist

$$u^T(C - VB^{-1}V^T)u = \begin{pmatrix} -B^{-1}V^T u \\ u \end{pmatrix}^T \begin{pmatrix} B & V^T \\ V & C \end{pmatrix} \begin{pmatrix} -B^{-1}V^T u \\ u \end{pmatrix} > 0. \quad \square$$

Wenn man $j = 2$ wählt erhält man

$$A = \begin{pmatrix} \sqrt{B} & 0 \\ \frac{V}{\sqrt{B}} & I \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & C - \frac{VV^T}{B} \end{pmatrix} \begin{pmatrix} \sqrt{B} & \frac{V^T}{\sqrt{B}} \\ 0 & I \end{pmatrix}$$

- Hier ist $B \in \mathbb{R}^{1 \times 1}$ eine Zahl. Wurzel und Divisions sind also wohldefiniert.
- Da \sqrt{B} und V/\sqrt{B} billig auszurechnen sind ist also *eine Spalte* von L billig auszurechnen.

Wegen Lemma 9.1 gilt

$$\begin{aligned} -VB^{-1}V^T &= -(VL_B^{-T})(L_B^{-1}V^T) \\ &= -\sum_{k=1}^{j-1} \begin{pmatrix} L_{jk} \\ \vdots \\ L_{nk} \end{pmatrix} \begin{pmatrix} L_{jk} & \dots & L_{nk} \end{pmatrix}. \end{aligned}$$

Daraus konstruieren wir einen Algorithmus:

1 for alle Spalten $j = 1, \dots, n$ do

2 | Definiere

$$F_j := \begin{pmatrix} A_{jj} & \dots & A_{jn} \\ \vdots & & \vdots \\ A_{nj} & \dots & A_{nn} \end{pmatrix} - \sum_{k=1}^{j-1} \begin{pmatrix} L_{jk} \\ \vdots \\ L_{nk} \end{pmatrix} \begin{pmatrix} L_{jk} & \dots & L_{nk} \end{pmatrix}$$

3 | Faktorisiere

$$F_j = \begin{pmatrix} L_{jj} & 0 & \dots & 0 \\ \vdots & & I & \\ L_{nj} & & & \end{pmatrix} \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & \tilde{U}_j & & \\ 0 & & & \end{pmatrix} \begin{pmatrix} L_{jj} & \dots & L_{nj} \\ 0 & & \\ \vdots & I & \\ 0 & & \end{pmatrix}$$

4 end

- In jedem Schritt wird eine weitere Spalte von L bestimmt.
- Die Matrix \tilde{U}_j ergibt sich aus der Faktorisierung. Sie wird aber bis auf weiteres nicht weiter verwendet.

Wir wollen diesen Algorithmus jetzt so modifizieren, dass er die Dünnbesetztheit von A ausnutzt.

9.1.2 Die Struktur von L

Zunächst machen wir uns klar, dass man die Struktur von L bestimmen kann, ohne L selbst bestimmen zu müssen.

Definition. Die Struktur \mathcal{S}_X einer Matrix $X \in \mathbb{R}^{n \times n}$ ist

$$\mathcal{S}_X := \{(i, j) : X_{ij} \neq 0\}.$$

- In der tatsächlichen Umsetzung von Matrixalgorithmen im Computer wird man statt der ganzen Matrix A nur deren Struktur \mathcal{S}_A speichern, sowie die dazugehörigen Einträge.
- Wie genau, dazu gibt es wieder verschiedene Varianten!

Wie sieht die Struktur des Cholesky-Faktors aus?

Satz 9.1. Sei $A \in \mathbb{R}^{n \times n}$ s.p.d. und L der Cholesky-Faktor.

1. Falls $i \leq j$ und $(j, i) \in \mathcal{S}_A$, so ist auch $(j, i) \in \mathcal{S}_L$.
2. Falls $i < j < k$ und $(j, i) \in \mathcal{S}_L$ und $(k, i) \in \mathcal{S}_L$, dann ist auch $(k, j) \in \mathcal{S}_L$.

Damit sind alle Einträge von \mathcal{S}_L beschrieben.

Beispiel.

$$A = \begin{pmatrix} \bullet & & & & \bullet \\ & \bullet & & & \\ \bullet & & \bullet & \bullet & \bullet \\ & & \bullet & \bullet & \\ \bullet & & \bullet & & \bullet \\ & \bullet & & \bullet & \\ \bullet & \bullet & & \bullet & \bullet \end{pmatrix}, \quad L = \begin{pmatrix} \bullet & & & & \\ & \bullet & & & \\ \bullet & & \bullet & & \\ & & \bullet & \bullet & \\ \bullet & & \bullet & \circ & \bullet \\ & \bullet & & \bullet & \circ & \bullet \\ \bullet & \bullet & & \circ & \bullet & \circ & \bullet \end{pmatrix}$$

Beweis. Betrachte wieder Darstellung

$$A = \begin{pmatrix} B & V^T \\ V & C \end{pmatrix}, \quad B \in \mathbb{R}^{1 \times 1} = \mathbb{R}.$$

1.
 - Da $L_{11} = \sqrt{B}$ und $B > 0$ enthält \mathcal{S}_L alle Diagonaleinträge.
 - Da die erste Spalte von L unterhalb der Diagonalen gleich $\frac{V}{\sqrt{B}}$ ist enthält \mathcal{S}_L alle Einträge von \mathcal{S}_A unterhalb der Diagonalen.
2.
 - Der Algorithmus berechnet die Cholesky-Zerlegung spaltenweise.
 - Im i -ten Schritt wird die i -te Spalte von L berechnet. Danach ändert sie sich nicht mehr.

$$\begin{pmatrix} 0 \\ \hline L_{ii} \\ \hline \tilde{l}_i \quad \tilde{A}_i - \tilde{l}_i \tilde{l}_i^T \end{pmatrix}$$

- Angenommen, $(j, i) \in \mathcal{S}_L$ und $(k, i) \in \mathcal{S}_L$
- Dann sind auch $(\tilde{l}_i)_j$ und $(\tilde{l}_i)_k$ Struktureinträge.
- Da $(\tilde{l}_i \tilde{l}_i^T)_{jk} = (\tilde{l}_i)_j \cdot (\tilde{l}_i)_k$ ist dann auch $(\tilde{l}_i \tilde{l}_i^T)_{jk}$ Struktureintrag.
- Also ist auch $(\tilde{A}_i - \tilde{l}_i \tilde{l}_i^T)_{jk}$ Struktureintrag.
- Mit 1) folgt die Aussage rekursiv. \square

Was passiert wenn $(\tilde{A}_i - \tilde{l}_i \tilde{l}_i^T)_{jk}$ zufällig gerade Null ergibt?

- Entgegen der Definition von S_L bezeichnet man (j, k) dann dennoch als Struktureintrag von L .
- Vorteil: Man kann dann die Struktur von L nur anhand der Struktur von A bestimmen.
- Vermutung: Dieser Fall kommt ohnehin selten vor.

9.1.3 Ausnutzen der Dünnbesetztheit, Teil 1

Wir wollen den Algorithmus jetzt so modifizieren, dass er die Dünnbesetztheit von A ausnutzt.

Idee: Die j -te Spalte von L hängt nur von der ersten Zeile und Spalte von F_j ab.

Modifizierter Algorithmus:

1 **for** alle Spalten $j = 1, \dots, n$ **do**

2 | Definiere

$$F_j := \begin{pmatrix} A_{jj} & \dots & \dots & A_{jn} \\ \vdots & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ A_{nj} & 0 & \dots & 0 \end{pmatrix} - \sum_{k=1}^{j-1} \begin{pmatrix} L_{jk} \\ \vdots \\ L_{nk} \end{pmatrix} \begin{pmatrix} L_{jk} & \dots & L_{nk} \end{pmatrix}$$

3 | Faktorisiere

$$F_j = \begin{pmatrix} L_{jj} & 0 & \dots & 0 \\ \vdots & & I & \\ L_{nj} & & & \end{pmatrix} \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & \hat{U}_j & & \\ 0 & & & \end{pmatrix} \begin{pmatrix} L_{jj} & \dots & L_{nj} \\ 0 & & \\ \vdots & I & \\ 0 & & \end{pmatrix}$$

4 **end**

Aus \tilde{U}_j ist eine andere Matrix \hat{U}_j geworden.

- Macht nichts: \tilde{U}_j wurde ohnehin nicht weiter verwendet.

Idee: Auch vom zweiten Summanden

$$- \sum_{k=1}^{j-1} \begin{pmatrix} L_{jk} \\ \vdots \\ L_{nk} \end{pmatrix} (L_{jk} \ \dots \ L_{nk})$$

ist nur die erste Zeile und Spalte relevant.

Es reicht also, über die Spalten $k = 1, \dots, j - 1$ zu addieren, für die $L_{jk} \neq 0$ ist.

Neuer Algorithmus:

1 **for** alle Spalten $j = 1, \dots, n$ **do**

2 | Definiere

$$F_j := \begin{pmatrix} A_{jj} & \dots & \dots & A_{jn} \\ \vdots & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ A_{nj} & 0 & \dots & 0 \end{pmatrix} - \sum_{\substack{k=1 \\ L_{jk} \neq 0}}^{j-1} \begin{pmatrix} L_{jk} \\ \vdots \\ L_{nk} \end{pmatrix} (L_{jk} \ \dots \ L_{nk})$$

3 | Faktorisiere

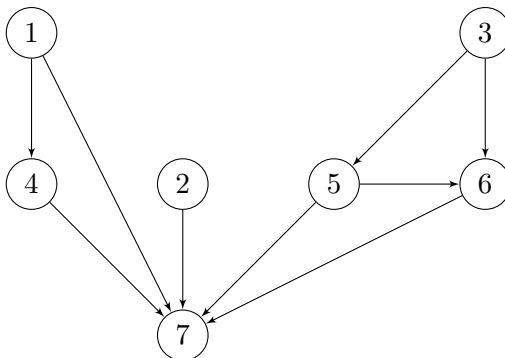
$$F_j = \begin{pmatrix} L_{jj} & 0 & \dots & 0 \\ \vdots & & I & \\ L_{nj} & & & \end{pmatrix} \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & \bar{U}_j & & \\ 0 & & & \end{pmatrix} \begin{pmatrix} L_{jj} & \dots & L_{nj} \\ 0 & & \\ \vdots & I & \\ 0 & & \end{pmatrix}$$

4 **end**

9.1.4 Graphen- und Baumdarstellung

Die Struktur der Dreiecksmatrix L lässt sich als gerichteter Graph $G = (V, E)$ darstellen mit $V = \{1, \dots, n\}$ und $E = \{(i, j) : (j, i) \in \mathcal{S}_L, i > j\}$.

Für die Beispielmatrix von oben erhält man:



Wir können also die letzte Version des Algorithmus umschreiben:

```

1 for alle Spalten  $j = 1, \dots, n$  do
2   Definiere
           
$$F_j := (\dots) - \sum_{\substack{k=1 \\ \exists \text{ Kante von } k \text{ nach } j}}^{j-1} \begin{pmatrix} L_{jk} \\ \vdots \\ L_{nk} \end{pmatrix} (L_{jk} \ \dots \ L_{nk})$$

3   Faktorisiere [...]
4 end
    
```

Der Eliminierungsbaum

Andererseits ist diese Darstellung teilweise redundant: zwischen zwei Knoten kann es mehr als einen Weg geben.

Denn: Satz 9.1 sagt, dass falls es die Kanten (i, j) und (i, k) gibt mit $j < k$, dann gibt es auch (j, k) .

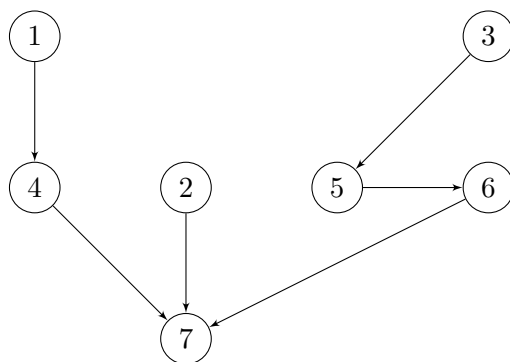
Entferne Kanten solange, bis jeder Knoten nur noch höchstens einen Nachfolger hat. Und zwar auf eine bestimmte Art:

Definition. Der Eliminierungsbaum $T(A)$ von A ist der Baum mit den Knoten $V = \{1, \dots, n\}$ und Kanten

$$E = \{(j, p) : \text{falls } p = \min\{i > j : (i, j) \in S_L\}\}.$$

Anschaulich: p ist die Zeile des ersten Eintrags von L in der j -ten Spalte (unter der Diagonalen).

Für das Beispiel von eben erhält man:



Satz 9.2. Falls $(j, k) \in S_L$, dann existiert ein Pfad $k \rightsquigarrow j$ in $T(A)$.

Definition. Ein Knoten i heißt Vorgänger von j , falls ein Weg $i \rightsquigarrow j$ in $T(A)$ existiert. Schreibe $T(j)$ für die Menge aller Vorgänger von j .

Mit dieser Terminologie können wir den Algorithmus nochmals umschreiben:

1 for alle Spalten $j = 1, \dots, n$ do

2 | Definiere

$$F_j := (\dots) - \sum_{k \in T(j) \setminus \{j\}}^{j-1} \begin{pmatrix} L_{jk} \\ \vdots \\ L_{nk} \end{pmatrix} (L_{jk} \ \dots \ L_{nk})$$

3 | Faktorisiere [...]

4 end

Die Umkehrung von Satz 9.2 gilt nicht unbedingt. So ist im Beispiel Knoten 3 Vorgänger von Knoten 7, der Eintrag $(3, 7)$ existiert aber nicht in L . Der Wechsel zum Eliminierungsbaum führt also erstmal wieder überflüssige Rechenoperationen ein. Im endgültigen Algorithmus sind die aber wieder verschwunden (s.u.).

9.1.5 Ausnutzen der Dünnbesetztheit, Teil 2

Idee: Jeder Summand in F_j ist für sich dünnbesetzt.

Aber: Jeder Summand hat doch vermutlich eine andere Besetzungsstruktur? Die *Summe* ist doch vermutlich dann doch relativ dicht?

Nein!

Satz 9.3. Seien i und j Knoten in $T(A)$, sodass ein Pfad $i \rightsquigarrow j$ existiert (also $i < j$). Dann sind alle Struktureinträge der i -ten Spalte von L (unterhalb der j -ten Zeile) in der Struktur der j -ten Spalte enthalten

$$(k, i) \in \mathcal{S}_L \implies (k, j) \in \mathcal{S}_L \quad (\text{falls } k \geq j \text{ und } i \rightsquigarrow j).$$

Seien also

$$j = i_0 < i_1 < \dots < i_r$$

die Zeilen der Einträge der j -ten Spalte von L .

Neuer Algorithmus:

1 for alle Spalten $j = 1, \dots, n$ do

2 | Definiere

$$F_j := \begin{pmatrix} A_{i_0 i_0} & \dots & \dots & A_{i_0 i_r} \\ \vdots & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ A_{i_r i_0} & 0 & \dots & 0 \end{pmatrix} - \sum_{k \in T(j) \setminus \{j\}}^{j-1} \begin{pmatrix} L_{i_0 k} \\ \vdots \\ L_{i_r k} \end{pmatrix} \begin{pmatrix} L_{i_0 k} & \dots & L_{i_r k} \end{pmatrix}$$

3 | Faktorisiere

$$F_j = \begin{pmatrix} L_{i_0 j} & 0 & \dots & 0 \\ \vdots & & I & \\ L_{i_r j} & & & \end{pmatrix} \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & U_j & & \\ 0 & & & \end{pmatrix} \begin{pmatrix} L_{i_0 j} & \dots & L_{i_r j} \\ 0 & & \\ \vdots & I & \\ 0 & & \end{pmatrix}$$

4 end

Die Matrizen $F_j \in \mathbb{R}^{(r+1) \times (r+1)}$ und $U_j \in \mathbb{R}^{r \times r}$ sind jetzt vollbesetzt.

- F_j nennt man j -te *Frontal-Matrix*
- U_j nennt man j -te *Update-Matrix*.

9.1.6 Matrix-Superposition (Der extend-add Operator)

Wir brauchen noch ein Werkzeug zum Arbeiten mit den neuen Matrizen.

- Die aktuelle Definition von F_j sagt, dass man den kompletten Teilbaum von $T(A)$ in j abläuft und für jeden Knoten Matrizen aufstellt.
- Das kann immer noch ziemlich teuer sein.
- Hier kommt der Trick: Man kann F_j effizient aus den Update-Matrizen der *direkten* Vorgänger zusammenbauen!
- Sei $R \in \mathbb{R}^{r \times r}$ mit $r \leq n$, $S \in \mathbb{R}^{s \times s}$ mit $s \leq n$.
- Jede Zeile/Spalte von R und S soll zu einer Zeile/Spalte der gegebenen Matrix A gehören
- Indextmengen: $i_1 < \dots < i_r$ für R ,
 $j_1 < \dots < j_s$ für S

1) Sei $k_1 < \dots < k_t$ die Vereinigung der beiden Indextmengen

2) Passe R und S an die Indextmenge $k_1 < \dots < k_t$ an, indem Nullzeilen und Nullspalten eingefügt werden.

3) Definiere $R \leftarrow \updownarrow \rightarrow S \in \mathbb{R}^{t \times t}$ als Summe der erweiterten Matrizen R, S .

Der Operator $\leftarrow \updownarrow \rightarrow$ wird in der englischsprachigen Literatur als „extend-add“ bezeichnet.
Beispiel.

$$R = \begin{pmatrix} p & q \\ u & v \end{pmatrix}, \quad S = \begin{pmatrix} w & x \\ y & z \end{pmatrix}$$

Indexmengen $\{5, 8\}$ bzw. $\{5, 0\}$.

Dann hat $R \leftarrow \updownarrow \rightarrow S$ die Indexmenge $\{5, 8, 9\}$, und

$$R \leftarrow \updownarrow \rightarrow S = \begin{pmatrix} p & q & 0 \\ u & v & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} w & 0 & x \\ 0 & 0 & 0 \\ y & 0 & z \end{pmatrix} = \begin{pmatrix} p+w & q & x \\ u & v & 0 \\ y & 0 & z \end{pmatrix}$$

Damit kann man eine billigere Formel für F_j finden.

Satz 9.4 (Liu [11, Thm. 4.1]). *Seien c_1, \dots, c_s die direkten Vorgänger des Knotens j im Eliminierungsbaum $T(A)$ von A . Dann ist*

$$F_j = \begin{pmatrix} A_{i_0 i_0} & A_{i_0 i_1} & \dots & A_{i_0 i_r} \\ A_{i_1 i_0} & & & \\ \vdots & & 0 & \\ A_{i_r i_0} & & & \end{pmatrix} \leftarrow \updownarrow \rightarrow U_{c_1} \leftarrow \updownarrow \rightarrow U_{c_2} \leftarrow \updownarrow \rightarrow \dots \leftarrow \updownarrow \rightarrow U_{c_s}.$$

Zum Beweis braucht man:

Satz 9.5 (Liu, Thm. 3.3). *Es gilt*

$$U_j = - \sum_{k \in T(j)} \begin{pmatrix} L_{i_1 k} \\ \vdots \\ L_{i_r k} \end{pmatrix} (L_{i_1 k} \quad \dots \quad L_{i_r k}).$$

Beweis. Umformen der Zerlegung von F_j gibt

$$F_j = \begin{pmatrix} L_{i_0 j} \\ \vdots \\ L_{i_r j} \end{pmatrix} (L_{i_0 j} \quad \dots \quad L_{i_r j}) + \begin{pmatrix} 0 & 0 \\ 0 & U_j \end{pmatrix}$$

Aber F_j und U_j unterscheiden sich nur in der ersten Zeile und Spalte.

Deshalb

$$\begin{aligned} & [F_j \text{ ohne erste Zeile und Spalte}] \\ & = [\bar{U}_j \text{ ohne erste Zeile und Spalte}] \\ & = \left[- \sum_{k \in T(j) \setminus \{j\}} \begin{pmatrix} L_{i_0 k} \\ \vdots \\ L_{i_r k} \end{pmatrix} (L_{i_0 k} \quad \dots \quad L_{i_r k}) \text{ ohne erste Zeile und Spalte} \right] \\ & = - \sum_{k \in T(j)} \begin{pmatrix} L_{i_1 k} \\ \vdots \\ L_{i_r k} \end{pmatrix} (L_{i_1 k} \quad \dots \quad L_{i_r k}) = \begin{pmatrix} L_{i_1 j} \\ \vdots \\ L_{i_r j} \end{pmatrix} (L_{i_1 j} \quad \dots \quad L_{i_r j}) + U_j. \end{aligned}$$

Daraus folgt die Behauptung. \square

Es fehlt noch der (kurze) Beweis von Satz 9.4.

9.1.7 Der endgültige Algorithmus

Berechne zunächst die Struktur von L .

Danach:

1 **for** alle Spalten $j = 1, \dots, n$ **do**

2 Seien i_0, \dots, i_r die Zeilenindizes der Einträge von L_{*j} . Nicht vergessen: $i_0 = j$

3 Seien c_1, \dots, c_s die direkten Vorgänger von j im Eliminationsbaum $T(A)$ von A .

4 Bilde Frontal-Matrix F_j wie in Satz 9.4

$$F_j = \begin{pmatrix} A_{i_0 i_0} & A_{i_0 i_1} & \dots & A_{i_0 i_r} \\ A_{i_1 i_0} & & & \\ \vdots & & 0 & \\ A_{i_r i_0} & & & \end{pmatrix} \leftarrow \rightleftharpoons U_{c_1} \leftarrow \rightleftharpoons U_{c_2} \leftarrow \rightleftharpoons \dots \leftarrow \rightleftharpoons U_{c_s}.$$

5 Faktorisiere

$$F_j = \begin{pmatrix} L_{i_0 i_0} & 0 & \dots & 0 \\ L_{i_1 i_0} & & & \\ \vdots & & I & \\ L_{i_r i_0} & & & \end{pmatrix} \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & U_j & \\ 0 & & & \end{pmatrix} \begin{pmatrix} L_{i_0 i_0} & L_{i_1 i_0} & \dots & L_{i_r i_0} \\ 0 & & & \\ \vdots & & & \\ 0 & & & I \end{pmatrix}$$

6 Man merke sich U_j , falls nötig.

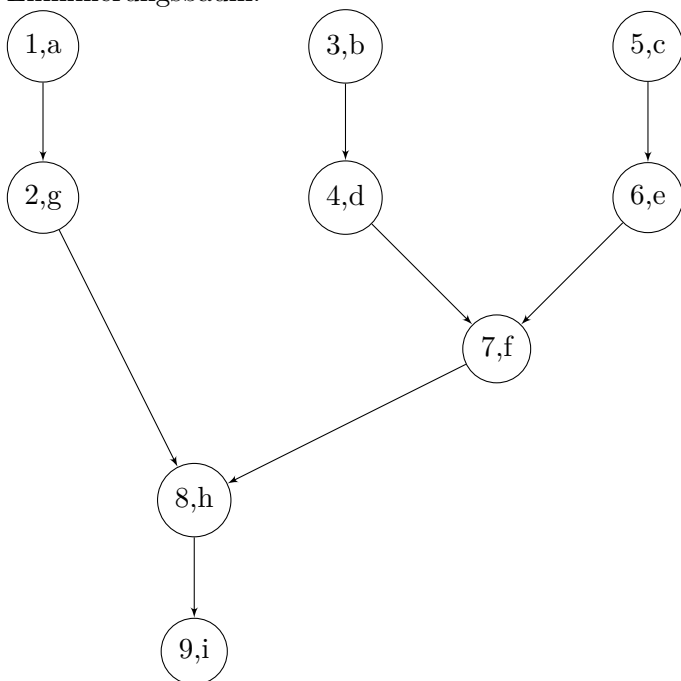
7 **end**

Das Ergebnis im j -ten Schritt ist also: $L_{i_0 i_0}, \dots, L_{i_r i_0}$ und U_j .

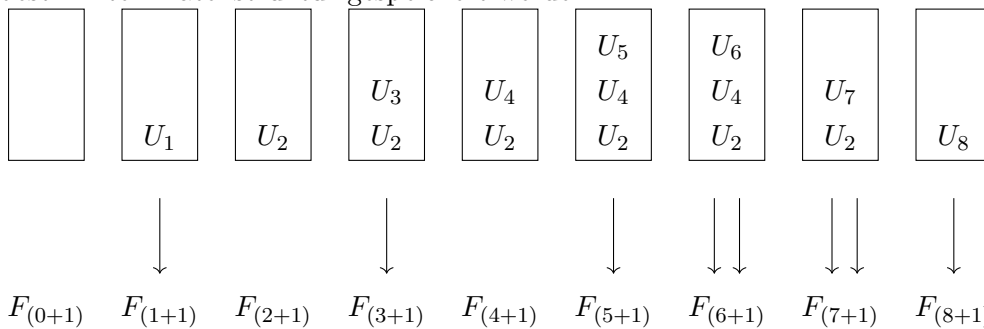
9.1.8 Umsortierungen der Matrix

- Die Matrixspalten werden von 1 bis n behandelt
- Die Update-Matrizen werden in eben dieser Reihenfolge erzeugt
- Wird eine Matrix U_j nicht für F_{j+1} gebraucht, so muss sie zwischengespeichert werden.
- Das verbraucht Speicher!

Eliminierungsbaum:



Mit dieser Nummerierung können die Matrizen U_j in einem Stapel (engl.: Stack), einer bestimmten Datenstruktur gespeichert werden.



Geht das immer?

Definition. Eine Ordnung, auf der Knotenmenge eines gerichteten Graphen heißt topologisch, wenn jeder Knoten vor seine Nachfolger einsortiert wird.

Satz 9.6. Sei $T(A)$ der Eliminationsbaum einer Matrix A . Jede topologische Ordnung von $T(A)$ erzeugt eine Umsortierung von A , die den E.-baum invariant lässt.

Definition. Eine topologische Ordnung eines Baumes $T(A)$ heißt Post-Ordnung, wenn alle Teilbäume konsekutiv durchnummeriert sind.

Die Zahlen $1, \dots, 9$ bilden in unserem Beispiel eine Post-Ordnung: Die Indexmengen jedes Teilbaums sind konsekutiv.

10 Numerik von gewöhnlichen Differentialgleichungen

Der Inhalt dieses Kapitels ist größtenteils dem Buch von Deuffhard und Bornemann [3] entnommen.

Gewöhnliche Differentialgleichung:

$$x'_i = f_i(t, x_1, \dots, x_d), \quad i = 1, \dots, d$$

wobei $(t, x) \in \mathbb{R} \times \mathbb{R}^d$ und $f_i: \Omega \rightarrow \mathbb{R}^d$, $\Omega \subseteq \mathbb{R} \times \mathbb{R}^d$ offen.

- Die Variable t ist häufig als Zeit interpretierbar, man spricht daher häufig von *Evolutionsproblemen*.
- x heißt *Zustandsvektor*.
- \mathbb{R}^d mit $x \in \mathbb{R}^d$ heißt *Zustandsraum*.
- $\mathbb{R} \times \mathbb{R}^d$ heißt *erweiterter Zustandsraum*.

Beispiele

1. (Radioaktiver Zerfall) Finde $x: \mathbb{R} \rightarrow \mathbb{R}$ so dass

$$x' = -kx$$

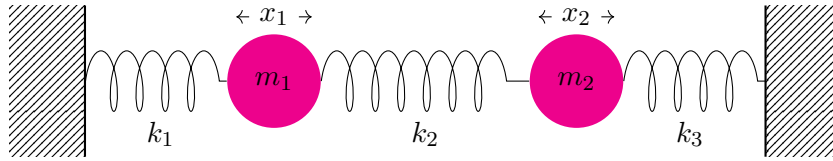
Achtung: Traditionell verwendet man das gleiche Symbol für Zustände $x \in \mathbb{R}^d$ und Funktionen in den Zustandsraum $x: \mathbb{R} \rightarrow \mathbb{R}^d$. Nicht verwirren lassen!

2. (Zwei-Massen-Schwinger) $d = 4$

$$\begin{aligned}x'_1 &= x_3 \\x'_2 &= x_4 \\x'_3 &= \frac{1}{m_1}(-k_1x_1 + k_2(x_2 - x_1)) \\x'_4 &= \frac{1}{m_2}(k_2(x_1 - x_2) - k_3x_2)\end{aligned}$$

Dürfen denn keine höheren Ableitungen vorkommen?

3. (Zwei-Massen-Schwinger physikalisch).



Massen m_1, m_2 , horizontale Positionen x_1, x_2 . Auf eine Masse wirken zwei Arten von Kräften:

- Trägheitskräfte: $m_i \ddot{x}_i(t)$
- Federkräfte: Für jede Feder Auslenkung \times Federkonstante

Mechanisches Prinzip: Alle wirkenden Kräfte addieren sich zu Null.

$$m_1 \ddot{x}_1(t) = -k_1 x_1 + k_2 (x_2 - x_1)$$

$$m_2 \ddot{x}_2(t) = k_2 (x_1 - x_2) - k_3 x_2$$

Kann auf obiges System erster Ordnung reduziert werden:

- Zusätzliche Variablen x_3, x_4
- Zusätzliche Gleichungen $\dot{x}_1 = x_3, \dot{x}_2 = x_4$

10.1 Anfangswertprobleme

Lösungen von gewöhnlichen Differentialgleichungen sind normalerweise nicht eindeutig.

Beispiel. Für alle $c \in \mathbb{R}$ löst $x(t) = ce^{-kt}$ die Gleichung

$$x' = -kx.$$

Deshalb: *Zusatzinformation.* Schreibe Anfangsbedingung

$$x(t_0) = x_0$$

vor. $x_0 \in \mathbb{R}^d$ heißt *Anfangswert*.

Bezeichnung:

Differentialgleichung + Anfangsbedingung = Anfangswertproblem (AWP/IVP)

10.2 Existenz und Eindeutigkeit

Sei die Notation wie oben.

- Der Definitionsbereich Ω von f sei offen,
- $(t_0, x_0) \in \Omega$.

Was meinen wir genau mit „Lösung des Anfangswertproblems“?

Definition. Sei $J \subset \mathbb{R}$ ein Intervall mit nichtleerem Inneren, und $t_0 \in J$. Eine Abbildung $x \in C^1(J, \mathbb{R}^d)$ heißt Lösung des AWP genau dann, wenn

$$\dot{x}(t) = f(t, x(t)) \quad \text{für alle } t \in J,$$

und $x(t_0) = x_0$ gilt.

- Es reichen schon Funktionen auf einem „kleinen“ Intervall.
- Wir wollen „große“ Intervalle I .

Es reichen schon wenige Zusatzinformationen, um zu erreichen, dass I größtmöglich ist.

Was heißt „größtmöglich“?

Definition (Maximale Fortsetzbarkeit). Eine Lösung $x \in C^1([t_0, t_1), \mathbb{R}^d)$ heißt (in der Zukunft) fortsetzbar bis an der Rand von Ω , wenn es eine Funktion $x^* \in C^1([t_0, t_+), \mathbb{R}^d)$ mit $t_1 \leq t_+ \leq \infty$ gibt, sodass

- $x(t) = x^*(t)$ für alle $t \in [t_0, t_1)$
- x^* ist ebenfalls Lösung,

und einer der drei folgenden Fälle vorliegt:

- 1) $t_+ = \infty$
- 2) $t_+ < \infty$ und $\lim_{t \uparrow t_+} |x^*(t)| = \infty$
- 3) $t_+ < \infty$ und $\lim_{t \uparrow t_+} \text{dist}((t, x^*(t)), \partial\Omega) = 0$

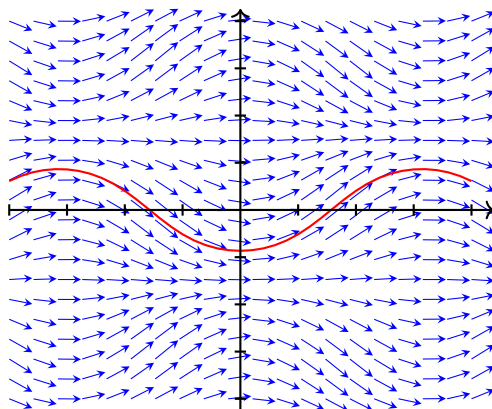
Beispiel (Fortsetzbarkeit). Betrachte das AWP $x' = -kx$, $x(0) = 1$. Eine Lösung davon ist

$$x: [0, 1] \rightarrow \mathbb{R}, \quad x(t) = e^{-kt}.$$

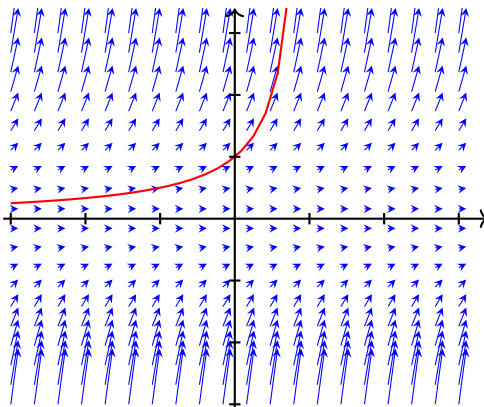
Die maximal fortgesetzte Lösung ist

$$x^*: [0, \infty) \rightarrow \mathbb{R}, \quad x^*(t) = e^{-kt}.$$

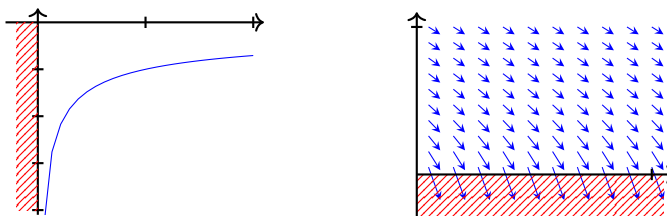
Beispiel. Für a) $f(t, x) = \sin t \cos x$



Für b) $f(t, x) = x^2$



Für c) $f(t, x) = -\frac{1}{\sqrt{x}}$



Wann sind Lösungen bis an den Rand fortsetzbar? Die Antwort ist erstaunlich einfach!
Gegeben sei das AWP

$$\dot{x} = f(t, x), \quad x(t_0) = x_0.$$

Satz 10.1 (Peano,1890). Sei $f: \Omega \subseteq \mathbb{R} \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ stetig im zweiten Argument. Dann hat das AWP für alle $(t_0, x_0) \in \Omega$ mindestens eine Lösung. Jede Lösung lässt sich bis an den Rand von Ω fortsetzen.

- Beweisskizze.*
1. Konstruiere eine numerische Approximation der vermuteten Lösung, mit Genauigkeitsparameter h .
 2. Die Folge dieser Approximationen für $h \rightarrow 0$ hat eine konvergente Teilfolge.
 3. Der Grenzwert dieser Teilfolge löst das AWP. □

Eindeutigkeit: Es kann mehr als eine Lösung geben.

Beispiel. Betrachte $x' = \sqrt{|x|}$, $x(0) = 0$.

- $f(x) = \sqrt{|x|}$ ist stetig auf \mathbb{R} , es existiert also eine Lösung.
- Z.B.: $x(t) = 0 \forall t$ ist Lösung
- Eine Lösung ist aber auch $x(t) = \frac{1}{4}t^2$ für $t > 0$

- Es kommt noch schlimmer:
 - Wähle ein $c > 0$
 - Definiere

$$\tilde{x}(t) := \begin{cases} 0 & \text{falls } 0 \leq t \leq c \\ \frac{1}{4}(t-c)^2 & \text{falls } c < t \end{cases}$$

- \tilde{x} löst das Problem.

Es gibt also unendlich viele Lösungen!

Für die Eindeutigkeit braucht man noch ein bisschen mehr.

Definition. Die Abbildung $f \in C(\Omega, \mathbb{R}^d)$ heißt auf Ω bzgl. x lokal Lipschitz-stetig, wenn zu jedem $(t_0, x_0) \in \Omega$ ein offener Zylinder

$$Z: (t_0 - \tau, t_0 + \tau) \times B_\rho(x_0) \subset \Omega$$

existiert, in dem eine Lipschitzbedingung

$$|f(t, x) - f(t, \bar{x})| \leq L|x - \bar{x}| \quad \forall (t, x), (t, \bar{x}) \in Z$$

mit Konstante L gilt.

Bemerkung. Falls $f(t, x)$ nach x ableitbar ist, dann ist es auch bzgl. x lokal Lipschitz-stetig.

Jetzt kommt der zentrale Satz zur Eindeutigkeit.

Benannt nach Émile Picard (1890) und Ernst Lindelöf (1894).

Satz 10.2 (Picard, Lindelöf). Betrachte das Anfangswertproblem

$$x' = f(t, x), \quad x(t_0) = x_0$$

auf dem erweiterten Zustandsraum $\Omega \subset \mathbb{R} \times \mathbb{R}^d$ mit $(t_0, x_0) \in \Omega$. f sei stetig, und bzgl. x lokal Lipschitz-stetig.

Dann besitzt das AWP eine bis an den Rand von Ω fortgesetzte Lösung. Sie ist eindeutig bestimmt, d.h. Fortsetzung jeder weiteren Lösung.

Beweisskizze. • Schreibe AWP als

$$x(t) = x_0 + \int_{t_0}^t f(s, x(s)) ds \quad \forall t \geq t_0$$

- Konstruiere dafür eine Fixpunktiteration

$$\varphi_0(t) = x_0 \quad \forall t \geq t_0$$

$$\varphi_{k+1}(t) = x_0 + \int_{t_0}^t f(s, \varphi_k(s)) ds \quad \text{„Picard-Iteration“}$$

- Banachscher Fixpunktsatz:

- 1) Die Iteration konvergiert gegen einen Fixpunkt.
- 2) Der Fixpunkt ist eindeutig.

- Der Fixpunkt löst das AWP. □

10.3 Evolution und Phasenfluss

Falls die Bedingungen des Satzes von Picard–Lindelöf gelten, so kann man eine elegante neue Notation einführen.

Sei $(t_0, x_0) \in \Omega$. Bezeichne mit $J_{\max}(t_0, x_0)$ das maximale Zeitintervall, auf dem eine Lösung des dazugehörigen AWP existiert.

Zu jedem Anfangswert (t_0, x_0) gibt es eine eindeutige Lösung, d.h. zu jedem AW (t_0, x_0) ist der Wert $x(t)$ für alle $J_{\max}(t_0, x_0)$ eindeutig bestimmt.

Definition. Für alle $t_0, t \in J_{\max}(t_0, x_0)$ heißt

$$\Phi^{t,t_0}: x_0 \mapsto x(t)$$

Evolution der Differentialgleichung $x' = f(t, x)$.

- Wohldefiniert, weil das AWP für jedes x_0 eine eindeutige Lösung hat.

Man kann also schreiben: $x(t) = \Phi^{t,t_0}x_0$.

Der Satz von Picard–Lindelöf erhält folgende schöne Form: Deuffhard und Bornemann [3, Lemma 2.9]

Satz 10.3 (Picard–Lindelöf). *Es mögen die Bedingungen des Satzes von Picard–Lindelöf gelten. Für alle $(t_0, x_0) \in \Omega$ gilt*

$$J_{\max}(t_0, x_0) = J_{\max}(t, \Phi^{t,t_0}x_0) \quad \forall t \in J_{\max}(t_0, x_0).$$

Außerdem

1. $\Phi^{t_0,t_0}x_0 = x_0$
2. $\Phi^{t,s}\Phi^{s,t_0}x_0 = \Phi^{t,t_0}x_0$ für alle $t, s \in J_{\max}(t_0, x_0)$.

Für autonome Gleichungen $x' = f(x)$ kann man die Abhängigkeit von t_0 weglassen (wähle immer $t_0 = 0$). Der Evolutionsoperator $\Phi^t x_0 = x(t)$ heißt dann „Phasenfluss“.

Aus den zwei Eigenschaften des vorigen Satzes wird dann

1. $\Phi^0 x_0 = x_0$
2. $\Phi^t \Phi^s x_0 = \Phi^{t+s} x_0$ (damit auch $\Phi^{-t} \Phi^t x_0 = \Phi^0 x_0 = x_0$).

Der Phasenfluss Φ hat also eine Gruppenstruktur.

10.4 Explizite Einschrittverfahren für AWP

Ziel: Finde eine numerische Approximation der Lösung $x \in C^1([t_0, T], \mathbb{R}^d)$ des AWP

$$x' = f(t, x), \quad x(t_0) = x_0$$

Vorgehensweise:

- Unterteile das Intervall $[t_0, T]$ durch $n + 1$ Zeitpunkte

$$t_0 < t_1 < t_2 < \dots < t_n = T. \quad (10.1)$$

- Die Menge der Zeitpunkte heißt *Gitter* $\Delta := \{t_0, t_1, \dots, t_n\}$
- Schrittweite: $\tau_j := t_{j+1} - t_j$ für $j = 0, \dots, n - 1$
- Maximale Schrittweite: $\tau_\Delta = \max_{j=0, \dots, n-1} \tau_j$

Wir suchen eine Gitterfunktion

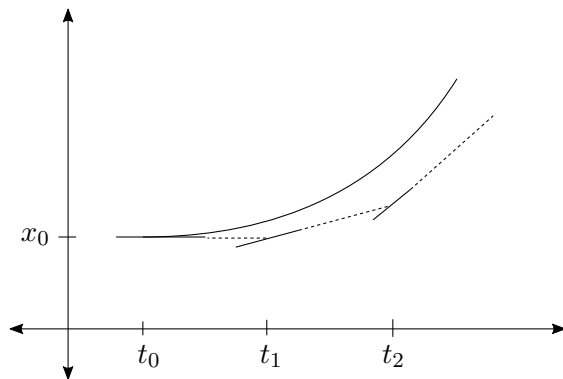
$$x_\Delta: \Delta \rightarrow \mathbb{R}^d,$$

welche die Lösung des AWP an den Gitterpunkten möglichst gut approximiert.

(Manchmal interpretieren wir so ein x_Δ auch als eine Funktion $[t_0, T] \rightarrow \mathbb{R}^d$, die die Werte an den Gitterpunkten linear interpoliert.)

10.4.1 Das explizite Euler-Verfahren

Nach L. Euler (1786), auch Eulersches Polygonzugverfahren genannt.



1. $x_\Delta(t_0) = x_0$

2. Für $t \in [t_j, t_{j+1}]$:

$$x_\Delta(t) = x_\Delta(t_j) + (t - t_j)f(t_j, x_\Delta(t_j))$$

3. Insbesondere:

$$x_{\Delta}(t_{j+1}) = x_{\Delta}(t_j) + \tau_j f(t_j, x_{\Delta}(t_j))$$

Keine Gleichungssysteme zu lösen \rightarrow das Verfahren ist explizit.

Beachte: Berechnung durch eine Zweiterm-Rekursion

$$1. x_{\Delta}(t_0) = x_0$$

$$2. x_{\Delta}(t_{j+1}) = \Psi^{t_{j+1}, t_j} x_{\Delta}(t_j), j = 0, 1, \dots, n-1$$

mit Ψ unabhängig von Δ .

- Die Funktion Ψ heißt *diskrete Evolution* des expliziten Euler-Verfahrens.

Hoffnung natürlich: Wenn das Gitter immer feiner wird (wenn also τ_{Δ} immer kleiner wird), wird der Unterschied zwischen x und x_{Δ} immer kleiner.

10.5 Konsistenz

Der Fehler der Lösung, also der Unterschied $x - x_{\Delta}$, besteht aus zwei Beiträgen:

- Jeder einzelne Schritt produziert einen Fehler.
- Da man nach dem ersten Schritt immer von einem fehlerbehafteten Wert startet, bekommt man auch falsche Ableitungen f .

Mit *Konsistenz* bezeichnet man das *lokale* Verhältnis zwischen der Evolution Φ und der diskreten Evolution Ψ .

Definition. Eine diskrete Evolution Ψ heißt *konsistent*, falls

$$\begin{aligned} \Psi^{t,t} x &= x && \text{für alle } (t, x) \in \Omega, \\ \frac{d}{d\tau} \Psi^{t+\tau, t} x \Big|_{\tau=0} &= f(x, t) && \text{für alle } (t, x) \in \Omega. \end{aligned}$$

Definition. Sei $(t, x) \in \Omega$. Die *Differenz*

$$\varepsilon(t, x, \tau) = \Phi^{t+\tau, t} x - \Psi^{t+\tau, t} x$$

heißt *Konsistenzfehler* von Ψ .

Man kann konsistente Evolutionen einfach charakterisieren.

Lemma 10.1 (Deuffhard und Bornemann [3, Lemma 4.4]). Die diskrete Evolution $\Psi^{t+\tau, t} x$ sei für jedes $(t, x) \in \Omega$ und hinreichend kleines τ bzgl. τ stetig differenzierbar. Dann ist äquivalent:

1. Ψ ist konsistent

2. Ψ hat die Darstellung

$$\Psi^{t+\tau,t}x = x + \tau\psi(t, x, \tau)$$

ψ heißt Inkrementfunktion. ψ ist stetig in $\tau = 0$, und

$$\psi(t, x, 0) = f(t, x).$$

3. Für den Konsistenzfehler gilt

$$\varepsilon(t, x, \tau) = o(\tau), \quad \tau \rightarrow 0 \quad (\text{also } \lim_{\tau \rightarrow 0} \frac{\varepsilon(\tau)}{\tau} = 0).$$

Beweis. Zunächst 1) \implies 3):

- Taylor-Entwicklung von Φ um $\tau = 0$

$$\begin{aligned} \Phi^{t+\tau,t}x &= \Phi^{t,t}x + \tau \cdot \frac{d}{d\tau} \Phi^{t+\tau,t}x|_{\tau=0} + o(\tau) \\ &= x + \tau \cdot f(t, x) + o(\tau) \end{aligned}$$

- Ebenso für Ψ :

$$\Psi^{t+\tau,t}x = x + \tau \cdot f(t, x) + o(\tau) \tag{10.2}$$

da Ψ konsistent ist.

- Subtraktion zeigt 1) \implies 3). Ebenso 3) \implies 1)

Zu 2):

- (10.2) bedeutet:

$$\Psi^{t+\tau,t}x = x + \tau f(t, x) + \eta(t, x, \tau)$$

mit einer Funktion η für die $\lim_{\tau \rightarrow 0} \frac{\eta(\tau)}{\tau} = 0$.

- Umformen ergibt

$$\frac{\Psi^{t+\tau,t}x - x}{\tau} = f(t, x) + \frac{\eta(t, x, \tau)}{\tau}.$$

Die Funktion

$$\psi(t, x, \tau) := f(t, x) + \frac{\eta(t, x, \tau)}{\tau}$$

ist also gerade die Inkrementfunktion aus 2).

- ψ stetig und $\psi(t, x, 0) = f(t, x)$, da $\eta \in o(\tau)$. □

Das vorige Lemma beschreibt Äquivalenzen. Insbesondere gilt: Wenn sich Ψ mit einer Inkrementfunktion schreiben lässt, so ist Ψ konsistent.

Beispiel. Das explizite Euler-Verfahren

$$\Psi^{t+\tau,t}x = x + \tau f(t, x) = x + \tau \psi(t, x, \tau)$$

ist konsistent.

Wir wollen eine quantitative Version von Konsistenz.

Definition. Eine diskrete Evolution Ψ besitzt Konsistenzordnung p , wenn es eine Konstante $C > 0$ unabhängig von t und x gibt, so dass

$$\varepsilon(t, x, \tau) \leq C\tau^{p+1}.$$

Ja, der Exponent ist wirklich $p + 1$!

Beispiel (Konsistenzordnung des Euler-Verfahrens). Das Euler-Verfahren ist

$$\Psi^{t+\tau,t}x = x + \tau f(t, x).$$

- Sei $f \in C^1(\Omega)$.
- Konsistenzfehler: $\varepsilon(t, x, \tau) = \Phi^{t+\tau,t}x - x - \tau f(t, x)$

Wir machen eine Taylor-Entwicklung von $\Phi^{t+\tau,t}x$ in $\tau = 0$.

- Erste Ableitung von Φ nach τ :

$$\frac{d}{d\tau}\Phi^{t+\tau,t}x = f(t + \tau, \Phi^{t+\tau,t}x)$$

- Warum? Sei $x(t) = \Phi^{t,t_0}x_0$ die Lösung des AWP. Dann ist $\Phi^{t+\tau,t}x = x(t + \tau)$, und

$$\begin{aligned} \frac{d}{d\tau}\Phi^{t+\tau,t}x &= x'(t + \tau) = f(t + \tau, x(t + \tau)) \\ &= f(t + \tau, \Phi^{t+\tau,t}x). \end{aligned}$$

- Zweite Ableitung (Kettenregel)

$$\frac{d^2}{d\tau^2}\Phi^{t+\tau,t}x = f_t(t + \tau, \Phi^{t+\tau,t}x) + f_x(t + \tau, \Phi^{t+\tau,t}x) \cdot \underbrace{\frac{d}{d\tau}\Phi^{t+\tau,t}x}_{=f(t+\tau, \Phi^{t+\tau,t}x)}$$

- Taylor-Entwicklung um $\tau = 0$:

$$\begin{aligned}\Phi^{t+\tau,t}x &= \Phi^{t+0,t}x + \tau \cdot \frac{d}{d\tau} \Phi^{t+\tau,t}x \Big|_{\tau=0} + \tau^2 \int_0^1 (1-\sigma) \frac{d^2}{d\tau^2} \Phi^{t+\sigma\tau,t}x d\sigma \\ &= x + \tau f(t, x) + \tau^2 \int_0^1 (1-\sigma) \left[f_t(t + \sigma\tau, \Phi^{t+\sigma\tau,t}x) \right. \\ &\quad \left. + f_x(t + \sigma\tau, \Phi^{t+\sigma\tau,t}x) f(t + \sigma\tau, \Phi^{t+\sigma\tau,t}x) \right] d\sigma\end{aligned}$$

- Deshalb ist

$$\begin{aligned}|\varepsilon(t, x, \tau)| &= |\Phi^{t+\tau,t}x - x - \tau f(t, x)| \\ &= \tau^2 \left| \int_0^1 \dots d\sigma \right| \\ &\leq \tau^2 \frac{1}{2} \max_{(s,z) \in K} |f_t(s, z) + f_x(s, z) f(s, z)|.\end{aligned}$$

- Das Maximum existiert, denn es wird über eine kompakte Menge K maximiert. Da wir nur an kleinen τ interessiert sind, können wir immer solch ein Kompaktum K , das alle relevanten (t, x) enthält (siehe Deuffhard und Bornemann [3, Beispiel 4.8]).
- Die Konsistenzordnung ist also 1.

10.6 Konvergenz

- Konsistenz ist ein lokales Phänomen, d.h. es betrachtet den Fehler nur in der Nähe eines festen t .
- Wir betrachten jetzt, wie gut eine Lösung $x \in C^1([t_0, T])$ insgesamt approximiert wird.
- Wir hoffen, dass für kleinere $\tau_\Delta = \max \tau_j$ die Approximation immer besser wird
- und das schnell!

Definition. Der Vektor der Approximationsfehler auf dem Gitter Δ

$$\varepsilon_\Delta: \Delta \rightarrow \mathbb{R}^d, \quad \varepsilon_\Delta(t) = x(t) - x_\Delta(t)$$

heißt Gitterfehler. Seine Norm

$$\|\varepsilon_\Delta\|_\infty = \max_{t \in \Delta} |\varepsilon_\Delta(t)|$$

heißt Diskretisierungsfehler.

Definition. Zu jedem Gitter Δ auf $[t_0, T]$ sei eine Gitterfunktion x_Δ gegeben. Die Familie dieser Gitterfunktionen konvergiert mit Ordnung $p \in \mathbb{N}$ gegen $x \in C^1([t_0, T])$, falls eine Konstante $C > 0$ existiert, so dass

$$\|\varepsilon_\Delta\|_\infty \leq C\tau_\Delta^p$$

für alle τ_Δ klein genug.

Alternative Notation: $\|\varepsilon_\Delta\|_\infty = \mathcal{O}(\tau_\Delta^p)$.

- Konvergenz hängt eng mit dem Konsistenzfehler zusammen.
- Das ist praktisch: Der Konsistenzfehler lässt sich häufig direkt am Verfahren ablesen.

Satz 10.4 (Deuffhard und Bornemann [3, Satz 4.10]). Sei ψ lokal Lipschitz-stetig in x . Die diskrete Evolution sei konsistent mit Ordnung p , d.h.

$$\varepsilon(t, x, \tau) := x(t + \tau) - \Psi^{t+\tau, t}x(t) = \mathcal{O}(\tau^{p+1}).$$

Dann definiert die diskrete Evolution Ψ für alle Gitter Δ mit hinreichend kleiner Zeitschrittweite τ_Δ eine Gitterfunktion x_Δ zum Anfangswert $x_\Delta(t_0) = x_0$. Die Familie dieser Gitterfunktionen konvergiert mit Ordnung p gegen die Lösung x des AWP, d.h.

$$\|\varepsilon_\Delta\|_\infty := \max_{t \in \Delta} |x(t) - x_\Delta(t)| = \mathcal{O}(\tau_\Delta^p).$$

Beweis. • Sei $K \subset \Omega$ kompakte Umgebung des Graphen der Lösung x .

- ψ ist lokal Lipschitz-stetig in x , d.h. es gibt $\tau_K, \Lambda_K > 0$ so dass

$$|\psi(t, x, \tau) - \psi(t, \bar{x}, \tau)| \leq \Lambda_K |x - \bar{x}|$$

für alle $(t, x), (t, \bar{x}) \in K$ und $\tau \in [0, \tau_K]$.

- Insbesondere ist dann $\Psi^{t+\tau, t}x$ für alle $(t, x) \in K, 0 \leq \tau \leq \tau_K$ definiert.
- Es gibt einen „Schlauch“ der Dicke $2\delta_K > 0$ um die Lösung x , der in K enthalten ist.

Genauer: Es gibt ein $\delta_K > 0$ so dass für alle $t \in [t_0, T]$

$$\text{Falls } |y - x(t)| \leq \delta_K, \text{ so folgt } (t, y) \in K.$$

- Sei Δ Gitter für $[t_0, T]$ mit $\tau_\Delta \leq \tau_K$.
- Setze voraus, dass x_Δ existiert und

$$|\varepsilon_\Delta(t)| = |x(t) - x_\Delta(t)| \leq \delta_K \quad \forall t \in \Delta.$$

- Betrachte ε_Δ genauer. Der Gitterfehler im Gitterpunkt t_{j+1} besteht aus zwei Teilen:

$$\begin{aligned}\varepsilon_\Delta(t_{j+1}) &= x(t_{j+1}) - x_\Delta(t_{j+1}) \\ &= x(t_{j+1}) - \Psi^{t_{j+1}, t_j} x_\Delta(t_j) \\ &= \underbrace{x(t_{j+1}) - \Psi^{t_{j+1}, t_j} x(t_j)}_{\text{Konsistenzfehler}} + \underbrace{\Psi^{t_{j+1}, t_j} x(t_j) - \Psi^{t_{j+1}, t_j} x_\Delta(t_j)}_{=:\varepsilon_j}.\end{aligned}$$

- ε_j kann als Propagation des Fehlers $\varepsilon_\Delta(t_j)$ durch Ψ zum Zeitpunkt t_{j+1} interpretiert werden.

Denn: Angenommen, Ψx sei linear in x (ist es nicht!). Dann wäre

$$\varepsilon_j = \Psi^{t_{j+1}, t_j}(x(t_j) - x_\Delta(t_j)) = \Psi^{t_{j+1}, t_j} \varepsilon_\Delta(t_j).$$

(Achtung: ε_j beschreibt eine Größe zur Zeit t_{j+1} !)

- Darstellung von ε_j mit der Inkrementfunktion:

$$\varepsilon_j = x(t_j) - x_\Delta(t_j) + \tau_j [\psi(t_j, x(t_j), \tau_j) - \psi(t_j, x_\Delta(t_j), \tau_j)]$$

- Lipschitz-Stetigkeit von ψ :

$$|\varepsilon_j| \leq |\varepsilon_\Delta(t_j)| + \tau_j \Lambda_K |x(t_j) - x_\Delta(t_j)| = (1 + \tau_j \Lambda_K) |\varepsilon_\Delta(t_j)|$$

Da wir die Abschätzung für alle j machen können folgt:

1. $|\varepsilon_\Delta(t_0)| = 0$
2. $|\varepsilon_\Delta(t_{j+1})| \leq C \tau_j^{p+1} + (1 + \tau_j \Lambda_K) |\varepsilon_\Delta(t_j)|$

Behauptung: Daraus folgt

$$|\varepsilon_\Delta(t)| \leq \tau_\Delta^p \frac{C}{\Lambda_K} (e^{\Lambda_K(t-t_0)} - 1) \quad (10.3)$$

für alle $t \in \Delta$.

Damit haben wir

$$\|\varepsilon_\Delta\|_\infty \leq \tau_\Delta^p \frac{C}{\Lambda_K} (e^{\Lambda_K(T-t_0)} - 1) = \mathcal{O}(\tau_\Delta^p).$$

Achtung: wir mussten *voraussetzen*, dass $\|\varepsilon_\Delta\|_\infty$ klein ist! □

Beweis der Behauptung (10.3). Induktion:

- Die Behauptung stimmt für $j = 0$.

- Angenommen die Behauptung ist richtig für $j < n$. Dann folgt

$$\begin{aligned} |\varepsilon_\Delta(t_{j+1})| &\leq C \underbrace{\tau_j^{p+1}}_{\leq \tau_\Delta^p \cdot \tau_j} + (1 + \tau_j \Lambda_K) \tau_\Delta^p \frac{C}{\Lambda_K} (e^{\Lambda_K(t_j - t_0)} - 1) \\ &\leq \tau_\Delta^p \frac{C}{\Lambda_K} [\tau_j \Lambda_K + (1 + \tau_j \Lambda_K)(e^{\Lambda_K(t_j - t_0)} - 1)] \\ &= \tau_\Delta^p \frac{C}{\Lambda_K} [(1 + \tau_j \Lambda_K)e^{\Lambda_K(t_j - t_0)} - 1]. \end{aligned}$$

- Es gilt aber $1 + \alpha \leq e^\alpha$, bzw. $1 + \tau_j \Lambda_K \leq e^{\tau_j \Lambda_K}$, also

$$(1 + \tau_j \Lambda_K) e^{\Lambda_K(t_j - t_0)} \leq e^{\Lambda_K \tau_j} e^{\Lambda_K(t_j - t_0)} = e^{\Lambda_K(t_{j+1} - t_0)}. \quad \square$$

Für die Profis hier noch die Argumentation, warum die Annahme dass $\|\varepsilon_\Delta\|_\infty$ klein ist, weggelassen werden kann.

- Bisher mussten wir voraussetzen, dass

$$|\varepsilon_\Delta(t)| \leq \delta_K \quad \text{für alle } t \in \Delta.$$

Jetzt haben wir aber (10.3)!

Damit zeigen wir, dass die Bedingung gilt, wenn τ_Δ klein genug ist.

- Wähle $\tau_* > 0$ so klein dass

$$\tau_*^p \frac{C}{\Lambda_K} (e^{\Lambda_K(T - t_0)} - 1) \leq \delta_K \quad \text{und} \quad \tau_* \leq \tau_K.$$

- Wähle ein Gitter Δ auf $[t_0, T]$ mit $\tau_\Delta \leq \tau_*$.
- Zeige mit der gleichen Induktion wie oben für alle j dass
 - die Abschätzung

$$|\varepsilon_\Delta(t_j)| \leq \tau_\Delta^p \frac{C}{\Lambda_K} (e^{\Lambda_K(t_j - t_0)} - 1) \leq \delta_K$$

gilt, und deshalb

- $x_\Delta(t_{j+1})$ existiert.

Beispiel (Explizites Euler-Verfahren).

$$x_{j+1} = x_j + \tau_j f(t_j, x_j)$$

- Konsistenzordnung 1 (der lokale Fehler verhält sich wie τ_Δ)
 - Inkrementfunktion $\psi(t, x, \tau) = f(t, x)$ ist lokal Lipschitz-stetig in x .
- \implies Verfahren konvergiert mit Ordnung 1, d.h. $\|\varepsilon_\Delta\|_\infty = \mathcal{O}(\tau_\Delta^1)$.
- \implies halber Fehler bedeutet doppelte Anzahl von Zeitschritten.
- \implies doppelte Anzahl von f -Auswertungen \implies doppelter Aufwand.

10.7 Explizite Runge–Kutta-Verfahren

[Nach: Carl Runge, 1856 (Bremen) – 1927 (Göttingen), Martin Wilhelm Kutta, 1867 (Pitschen/Oberschlesien) – 1944 (Fürstfeldbruck)]

Können wir Verfahren mit einer höheren Konsistenzordnung konstruieren?

Wiederholung: Wie lief der Beweis, dass das explizite Euler-Verfahren Konsistenzordnung 1 hat?

- Konsistenzfehler:

$$\begin{aligned}\varepsilon(t, x, \tau) &= \Phi^{t+\tau, t} x - \Psi^{t+\tau, t} x \\ &= (\Phi^{t+\tau, t} x - x) - \tau \psi(t, x, \tau)\end{aligned}$$

- Entwickle den Term in Klammern

$$\Phi^{t+\tau, t} x - x = \tau f(t, x) + \mathcal{O}(\tau^2)$$

- Euler-Verfahren: $\psi(t, x, \tau) = f(t, x)$ eliminiert gerade den ersten Term!

arrow Konsistenzfehler ist $\mathcal{O}(\tau^2)$

10.7.1 Taylor-Verfahren

Idee: Berechne weiteren Term der Taylor-Reihe:

$$\Phi^{t+\tau, t} x - x = \tau f(t, x) + \frac{\tau^2}{2} \left[\frac{df(t, x)}{dt} + \frac{df(t, x)}{dx} \cdot f(t, x) \right] + \mathcal{O}(\tau^3)$$

Inkrementfunktion für ein Verfahren mit Ordnung 2:

$$\psi^* = f(t, x) + \frac{\tau}{2} \left[f_t(t, x) + f_x(t, x) \cdot f(t, x) \right]$$

- So konstruierte Verfahren heißen „Taylor-Verfahren“.
- Im Prinzip für jede Ordnung möglich, solange f glatt genug ist.
- Man muss Ableitungen von f berechnen.
- Das geht sogar automatisch!
 - Schlagwort: Automatisches Differenzieren (AD)
 - de facto aber nur für Ableitungen niedriger Ordnung
- Wird deshalb in der Praxis kaum verwendet.

10.7.2 Idee der Runge-Kutta-Verfahren

Ziel: Hohe Ordnung ohne Ableitungen von f .

Idee: (Runge, 1893)

- Hauptsatz der Integralrechnung:

$$x: [t_0, T] \rightarrow \mathbb{R}^d \quad \text{löst} \quad x' = f(t, x),$$

also

$$x(t + \tau) = x(t) + \int_0^\tau f(t + \sigma, x(t + \sigma)) d\sigma$$

bzw.

$$\Phi^{t+\tau, t} x - x = \int_0^\tau f(t + \sigma, \Phi^{t+\sigma, t} x) d\sigma$$

- Approximiere das Integral numerisch, z.B. mit der Mittelpunktsregel / 1-Punkt Gauß-Legendre:

$$\int_0^\tau f(t + \sigma, \Phi^{t+\sigma, t} x) d\sigma = \tau f\left(t + \frac{\tau}{2}, \Phi^{t+\frac{\tau}{2}, t} x\right) + \mathcal{O}(\tau^3)$$

- Beachte: auch hier ist der Fehler in $\mathcal{O}(\tau^3)$!

Wie berechnet man aber $\Phi^{t+\frac{\tau}{2}, t} x$?

- Auf den ersten Blick nicht einfacher zu berechnen als $\Phi^{t+\tau, t} x$.
- *Glück:* $\Phi^{t+\frac{\tau}{2}, t} x$ taucht in einem Term auf, der mit τ multipliziert wird.
- Es reicht deshalb, $\Phi^{t+\frac{\tau}{2}, t} x$ bis auf $\mathcal{O}(\tau^2)$ zu berechnen.
- Das geht mit dem expliziten Euler-Verfahren:

$$\Phi^{t+\frac{\tau}{2}, t} x = x + \frac{\tau}{2} f(t, x) + \mathcal{O}(\tau^2)$$

- Man erhält das *Verfahren von Runge*:

$$\Psi^{t+\tau, t} x = x + \tau f\left(t + \frac{\tau}{2}, x + \frac{\tau}{2} f(t, x)\right)$$

mit Konsistenzordnung 2.

- Zwei Auswertungen von f pro Gitterpunkt, gegenüber 4 Auswertungen (von f oder seinen Ableitungen) beim entsprechenden Taylor-Verfahren.

Den Ansatz von Runge kann man verallgemeinern. Das war der Beitrag von Kutta 1901.

- Schreibe Verfahren von Runge in drei Schritten:

1. $k_1 := f(t, x)$
 2. $k_2 := f\left(t + \frac{\tau}{2}, x + \frac{\tau}{2}k_1\right)$
 3. $\Psi^{t+\tau, t}x = x + \tau k_2$
- Allgemein: s -stufiges explizites Runge-Kutta-Verfahren:
 1. $k_i := f\left(t + c_i\tau, x + \tau \sum_{j=1}^{i-1} a_{ij}k_j\right) \quad \forall i = 1, \dots, s$
 2. $\Psi^{t+\tau, t}x = x + \tau \sum_{i=1}^s b_i k_i.$
 - Die Größen $k_i = k_i(t, x, \tau)$ heißen *Stufen* des Verfahrens.

Koeffizienten:

$$\begin{aligned}
 b &= (b_1, \dots, b_s) \\
 c &= (c_1, \dots, c_s) \\
 A &= \begin{pmatrix} 0 & & & & \\ a_{21} & 0 & & & \\ \vdots & \ddots & & & \\ a_{s1} & a_{s2} & \dots & a_{s,s-1} & 0 \end{pmatrix} \in \mathbb{R}^{s \times s}
 \end{aligned}$$

Traditionell notiert man die Koeffizienten im Butcher-Schema (nach John C. Butcher, 1933 in Auckland)

$$\begin{array}{c|c}
 c^T & A \\
 \hline
 & b
 \end{array}$$

Beispiele:

$$\text{expl. Euler-Verfahren} \quad \begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

$$\text{Verfahren von Runge} \quad \begin{array}{c|cc} 0 & 0 & \\ \hline \frac{1}{2} & \frac{1}{2} & 0 \\ & 0 & 1 \end{array}$$

- Eine Funktionsauswertung pro Stufe
- $s + s + \frac{(s-1)s}{2}$ Parameter bei s Stufen

Wie müssen die Koeffizienten gewählt werden, um eine möglichst hohe Konsistenzordnung zu erreichen?

Zuerst: Konsistenz an sich (d.h., Konsistenz erster Ordnung)

- Einschrittverfahren

$$\Psi^{t+\tau,t}x = x + \tau\psi(t, x, \tau)$$

ist konsistent, wenn $\psi(t, x, 0) = f(t, x)$ ist.

- Runge–Kutta-Verfahren:

$$k_i(t, x, 0) = f\left(t + c_i \cdot 0, x + 0 \cdot \sum_{j=1}^{i-1} a_{ij}k_j\right) = f(t, x)$$

$$\psi(t, x, 0) = \sum_{i=1}^s b_i k_i = f(t, x) \sum_{i=1}^s b_i$$

Lemma 10.2. Ein explizites Runge–Kutta-Verfahren ist genau dann konsistent für alle $f \in C(\Omega, \mathbb{R}^d)$, wenn $\sum_{i=1}^s b_i = 1$.

10.7.3 Autonomisierung

- Taylor- und Runge–Kutta-Verfahren brauchen $f \in C^p(\Omega, \mathbb{R}^d)$, also gleiche Glattheit in Zeit und Zustand.
- Wir wollen die Notation vereinfachen, und nur noch autonome Gleichungen betrachten.
- Das geht erstaunlich einfach! Ersetze

$$x' = f(t, x), \quad x(t_0) = x_0$$

durch das erweiterte System

$$\begin{pmatrix} x'(t) \\ s'(t) \end{pmatrix} = \begin{pmatrix} f(s(t), x(t)) \\ 1 \end{pmatrix}, \quad \begin{pmatrix} x(0) \\ s(0) \end{pmatrix} = \begin{pmatrix} x_0 \\ t_0 \end{pmatrix}.$$

- Beide Systeme haben die gleiche Lösung.
- Runge–Kutta-Verfahren liefern möglicherweise *unterschiedliche Lösungen* für die beiden Gleichungen!
- Das ist natürlich häßlich!

Lemma 10.3. Ein explizites Runge–Kutta-Verfahren ist genau dann invariant unter Autonomisierung, wenn es konsistent ist, und

$$c_i = \sum_{j=1}^{i-1} a_{ij} \quad \text{für } i = 1, \dots, s$$

erfüllt.

Wir betrachten bis auf weiteres nur noch solche Runge–Kutta-Verfahren, und schreiben sie

$$(b, A).$$

- Nur noch autonome Probleme

$$x' = f(x), \quad x(0) = x_0$$

mit Phasenfluss Φ^t .

- Diskreter Fluss $\Psi^{t+\tau, t}$ vereinfacht sich zu

$$\Psi^t x = x + \tau \psi(x, \tau).$$

10.7.4 Konstruktion von Runge-Kutta-Verfahren

Wir wollen ein Runge–Kutta-Verfahren (b, A) der Ordnung p konstruieren.

- Wieviele Stufen brauche ich?
 - Mindestens p Stück
- Nach Autonomisierung sind noch $\frac{(s+1)s}{2}$ Koeffizienten zu bestimmen.

Schritt 1: Aufstellung von Bedingungsgleichungen an die Koeffizienten, die die gewünschte Ordnung liefern.

Schritt 2: Lösen dieser Gleichungen.

Wir behandeln den Fall $p = 4$.

- Autonome Gleichung $x' = f(x)$ mit $f \in C^4(\Omega_0)$.

Ansatz: Entwickle den Konsistenzfehler

$$\varepsilon(x, \tau) = \Phi^\tau x - \Psi^\tau x$$

in Taylor-Reihen bis auf $\mathcal{O}(\tau^5)$.

Konstruiere Ψ so, dass die ersten p Terme von ε verschwinden.

Taylor-Entwicklung des Phasenflusses Φ^τ

Taylor-Entwicklung von Φ^τ nach τ .

Vom Euler-Verfahren wissen wir schon dass

$$\Phi^\tau x = x + \tau f(x) + \frac{\tau^2}{2} f'(x) \cdot f(x) + \mathcal{O}(\tau^3).$$

Wie bekommen wir die nächsthöhere Ordnung?

Trick:

$$\frac{d}{d\tau} \Phi^\tau x = x'(\tau) = f(x(\tau), \tau) = f(\Phi^\tau x)$$

Einsetzen der Reihe für Φ^τ :

$$\frac{d}{d\tau} \Phi^\tau x = f\left(x + \tau f(x) + \frac{\tau^2}{2} f'(x) \cdot f(x) + \mathcal{O}(\tau^3)\right)$$

Taylor-Entwicklung davon:

$$\begin{aligned} \frac{d}{d\tau} \Phi^\tau x &= f(x) + f'(x) \cdot \left(\tau f(x) + \frac{\tau^2}{2} f'(x) f(x) + \mathcal{O}(\tau^3)\right) + \frac{1}{2!} f''(x)(\tau f(x), \tau f(x)) + \mathcal{O}(\tau^3) \\ &= f + f'f + \frac{\tau^2}{2} (f'f'f + f''(f, f)) + \mathcal{O}(\tau^3) \end{aligned}$$

Wir wollen aber eine Reihendarstellung von $\Phi^\tau x$.

- Hauptsatz:

$$\begin{aligned} \Phi^\tau x &= x + \int_0^\tau \frac{d}{d\sigma} \Phi^\sigma x d\sigma \\ &= x + \int_0^\tau \left[f + \sigma f'f + \frac{\sigma^2}{2} (f'f'f + f''(f, f)) + \mathcal{O}(\sigma^3) \right] d\sigma \\ &= x + \tau f + \frac{\tau^2}{2} f'f + \frac{\tau^3}{3!} (f'f'f + f''(f, f)) + \mathcal{O}(\tau^4) \end{aligned}$$

- Eine Ordnung mehr!
- Und nochmal!

$$\begin{aligned} f(\Phi^\tau x) &= f + \tau f'f + \frac{\tau^2}{2} (f'f'f + f''(f, f)) \\ &\quad + \frac{\tau^3}{3!} [f'''(f, f, f) + 3f''(f'f, f) + f'f''(f, f) + f'f'f'f] + \mathcal{O}(\tau^4) \end{aligned}$$

- Nochmal den Hauptsatz benutzen

$$\begin{aligned} \Phi^\tau x &= x + \tau f + \frac{\tau^2}{2} f'f + \frac{\tau^3}{3!} (f'f'f + f''(f, f)) \\ &\quad + \frac{\tau^4}{4!} [f'''(f, f, f) + 3f''(f'f, f) + f'f''(f, f) + f'f'f'f] + \mathcal{O}(\tau^5) \end{aligned}$$

Taylorentwicklung des diskreten Flusses Ψ^τ

Ähnliches Spiel mit den Runge–Kutta Gleichungen

$$k_i := f\left(x + \tau \sum_{j=1}^{i-1} a_{ij} k_j\right) \quad \text{für } i = 1, \dots, s.$$

1. f ist stetig, also auf Kompakta beschränkt
 $\implies k_i$ beschränkt, $k_i = \mathcal{O}(1)$

2. Einsetzen:

$$k_i = f\left(x + \tau \sum_{j=1}^{i-1} a_{ij} \mathcal{O}(1)\right) = f(x + \tau \mathcal{O}(1)) = f(x + \mathcal{O}(\tau))$$

Taylor-Entwicklung um x :

$$k_i = f(x) + \mathcal{O}(\tau)$$

3. Einsetzen:

$$\begin{aligned} k_i &= f\left(x + \tau \sum_{j=1}^{i-1} a_{ij} (f + \mathcal{O}(\tau))\right) = f\left(x + \tau \sum_{j=1}^{i-1} a_{ij} f + \mathcal{O}(\tau^2)\right) \\ &= f(x + \tau c_i f + \mathcal{O}(\tau^2)) \quad \text{mit } c_i = \sum_{j=1}^{i-1} a_{ij}. \end{aligned}$$

Taylor-Entwicklung:

$$k_i = f + \tau c_i f' f + \mathcal{O}(\tau^2) \quad \forall i = 1, \dots, s$$

4. Nochmal:

$$\begin{aligned} k_i &= f\left(x + \tau \sum_{j=1}^{i-1} a_{ij} (f + \tau c_j f' f) + \mathcal{O}(\tau^3)\right) \\ &= f + \tau c_i f' f + \tau^2 \sum_j a_{ij} c_j f' f' f + \frac{\tau^2}{2} c_i^2 f''(f, f) + \mathcal{O}(\tau^3) \end{aligned}$$

5. Nochmal:

$$\begin{aligned} k_i &= f\left[x + \tau c_i f + \tau^2 \sum_{j=1}^{i-1} a_{ij} c_j f' f + \tau^3 \sum_{jk} a_{ij} a_{jk} c_k f' f' f \right. \\ &\quad \left. + \frac{\tau^3}{2} \sum_j a_{ij} c_j^2 f''(f, f) + \mathcal{O}(\tau^4)\right] \\ &= f + \tau c_i f' f + \tau^2 \sum_j a_{ij} c_j f' f' f + \frac{\tau^2}{2} c_i^2 f''(f, f) \\ &\quad + \tau^3 \sum_{jk} a_{ij} a_{jk} c_k f' f' f' f + \frac{\tau^3}{2} \sum_j a_{ij} c_j^2 f' f''(f, f) \\ &\quad + \tau^3 \sum_j c_i a_{ij} c_j f''(f' f, f) + \frac{\tau^3}{6} c_i^3 f'''(f, f, f) + \mathcal{O}(\tau^4) \end{aligned}$$

Einsetzen in $\Psi^\tau x = x + \tau \sum_i b_i k_i$:

$$\begin{aligned} \Psi^\tau x &= x + \tau \sum_{i=1}^s b_i f + \frac{\tau^2}{2} \left(2 \sum_i b_i c_i f' f \right) \\ &+ \frac{\tau^3}{3!} \left[3 \sum_i b_i c_i^2 f''(f, f) + 6 \sum_{i,j} b_i a_{ij} c_j f' f' f \right] \\ &+ \frac{\tau^4}{4!} \left[4 \sum_i b_i c_i^3 f'''(f, f, f) + 24 \sum_{i,j} b_i c_i a_{ij} c_j f''(f' f, f) \right. \\ &\left. + 12 \sum_{i,j} b_i a_{ij} c_j^2 f' f''(f, f) + 24 \sum_{i,j,k} b_i a_{ij} a_{jk} c_k f' f' f' f \right] + \mathcal{O}(\tau^5) \end{aligned}$$

Koeffizientenvergleich ergibt die Ordnungsbedingungen:

Satz 10.5 (Deuffhard und Bornemann [3, Satz 4.18]). *Ein Runge–Kutta-Verfahren (b, A) besitzt genau dann*

- *Konsistenzordnung $p = 1$, falls*

$$\sum_{i=1}^s b_i = 1,$$

- *Konsistenzordnung $p = 2$, falls zusätzlich*

$$\sum_{i=1}^s b_i c_i = \frac{1}{2},$$

- *Konsistenzordnung $p = 3$, falls zusätzlich*

$$\sum_i b_i c_i^2 = \frac{1}{3} \quad \text{und} \quad \sum_{i,j} b_i a_{ij} c_j = \frac{1}{6},$$

- *Konsistenzordnung 4, falls zusätzlich*

$$\begin{aligned} \sum_i b_i c_i^3 &= \frac{1}{4} & \sum_{i,j} b_i c_i a_{ij} c_j &= \frac{1}{8} \\ \sum_{i,j} b_i a_{ij} c_j^2 &= \frac{1}{12} & \sum_{i,j,k} b_i a_{ij} a_{jk} c_k &= \frac{1}{24} \end{aligned}$$

gelten.

(Eigentlich muss man noch zeigen, dass diese Bedingungen notwendig und nicht nur hinreichend sind.)

Lösen der Gleichungen

Wir wollen ein Verfahren vierter Ordnung.

- Reichen $s = 3$ Stufen?
 - Nein! 8 Gleichungen, aber nur $\frac{(s+1)s}{2} = 6$ Unbekannte.
 - Das Gleichungssystem ist überbestimmt \rightarrow keine Lösung
- $s = 4$? $\frac{(s+1)s}{2} = 10$ Unbekannte, \rightarrow könnte gehen!

Unbekannte: $b_1, b_2, b_3, b_4, a_{21}, a_{31}, a_{32}, a_{41}, a_{42}, a_{43}$

1. $b_1 + b_2 + b_3 + b_4 = 1$
2. $b_1c_1 + b_2c_2 + b_3c_3 + b_4c_4 = \frac{1}{2}$
3. $b_2c_2^2 + b_3c_3^2 + b_4c_4^2 = \frac{1}{3}$
4. $b_3a_{32}c_2 + b_4(a_{42}c_2 + a_{43}c_3) = \frac{1}{6}$
5. $b_2c_2^3 + b_3c_3^3 + b_4c_4^3 = \frac{1}{4}$
6. $b_3c_3a_{32}c_2 + b_4c_4(a_{42}c_2 + a_{43}c_3) = \frac{1}{8}$
7. $b_3a_{32}c_2^2 + b_4(a_{42}c_2^2 + a_{43}c_3^2) = \frac{1}{12}$
8. $b_4a_{43}a_{32}c_2 = \frac{1}{24}$

Außerdem $c_i = \sum_j a_{ij}$, $i = 1, \dots, 4$, insbesondere $c_1 = 0$.

Was nun?

- Gute Idee:

$$\int_0^1 1 dx = 1, \quad \int_0^1 x dx = \frac{1}{2}, \quad \int_0^1 x^2 dx = \frac{1}{3}, \quad \int_0^1 x^3 dx = \frac{1}{4}.$$

- Approximiere Integral durch 4-Punkt-Formel mit Stützstellen c_1, c_2, c_3, c_4 , Gewichten b_1, b_2, b_3, b_4
- Falls Formel exakt für kubische Polynome ist, erhält man die Gleichungen 1, 2, 3, und 5.

Simpson-Regel: Stützstellen $0, \frac{1}{2}, 1$ und Gewichte $\frac{1}{6}, \frac{2}{3}, \frac{1}{6}$.

Verdopple mittleren Punkt:

$$c = (0, \frac{1}{2}, \frac{1}{2}, 1), \quad b = (\frac{1}{6}, \frac{1}{3}, \frac{1}{3}, \frac{1}{6})$$

Damit kann man die fehlenden a_{ij} bestimmen.

Man erhält das klassische Runge-Kutta-Verfahren:

| | | | | |
|---------------|---------------|---------------|---------------|---------------|
| 0 | | | | |
| $\frac{1}{2}$ | $\frac{1}{2}$ | | | |
| $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | | |
| 1 | 0 | 0 | 1 | |
| | $\frac{1}{6}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{6}$ |

10.8 Lineare Mehrschrittverfahren

10.8.1 Einführung

Gegeben sei ein AWP

$$x'(t) = f(t, x(t)), \quad t \in [t_0, T], \quad x(t_0) = x_0.$$

Approximation auf Gitter $\Delta = \{t_0, \dots, t_n\}$ mit

$$t_0 < t_1 < \dots < t_n = T$$

durch Gitterfunktion x_Δ mit dem Ziel $x_\Delta(t_i) \approx x(t_i)$ für $i = 1, \dots, n$.

Bisher: Einschrittverfahren

- Berechne $x_\Delta(t_{j+1})$ ausschließlich aus der Kenntnis des vorangegangenen Zustands $x_\Delta(t_j)$.
- Formalisierung: Diskrete Evolution

$$x_\Delta(t_{j+1}) = \Psi^{t_{j+1}, t_j} x_\Delta(t_j).$$

Eigenschaften der Einschrittverfahren:

- Konsistenztheorie eher schwierig
- Konvergenztheorie eher einfach
- Änderung der Zeitschrittweite jederzeit möglich.
- Verfahren hoher Ordnung sind teuer: Für Konsistenzordnung p braucht man (z.B. mit einem expliziten Runge-Kutta-Verfahren) mindestens $s \geq p$ Auswertungen der Funktion f .

Idee der Mehrschrittverfahren: Berechne den neuen Wert $x_\Delta(t_{j+1})$ aus den letzten k Werten, also

$$x_\Delta(t_{j-k+1}), \dots, x_\Delta(t_j) \mapsto x_\Delta(t_{j+1}).$$

Vorteile:

- nur eine einzige Auswertung von f pro Schritt, für beliebig hohe Ordnung.

Nachteile:

- Konvergenztheorie komplizierter,
- nicht so flexibel bei Änderungen der Schrittweite, schwierig auf nicht uniformen Gittern,
- k -Schritt-Verfahren benötigen neben Startwert $x_\Delta(t_0)$ noch $(k - 1)$ zusätzliche Startwerte $x_\Delta(t_1), \dots, x_\Delta(t_{k-1})$.

10.8.2 Mehrschrittverfahren für äquidistante Gitter

Wir beschränken uns auf äquidistante (auch: uniforme) Gitter

$$t_j = t_0 + j\tau, \quad j = 0, 1, \dots, n$$

also

$$\tau = \frac{T - t_0}{n}.$$

Dies ist nicht nur zur Bequemlichkeit. Mehrschrittverfahren auf unregelmäßigen Gittern sind deutlich komplizierter!

Beispiel: Die explizite Mittelpunktsregel

- Integriere die Differentialgleichung über $[t_{j-1}, t_{j+1}]$:

$$x(t_{j+1}) = x(t_{j-1}) + \int_{t_{j-1}}^{t_{j+1}} x'(\sigma) d\sigma = x(t_{j-1}) + \int_{t_{j-1}}^{t_{j+1}} f(\sigma, x(\sigma)) d\sigma$$

- Approximiere das Integral durch die Mittelpunktsregel

$$\int_{t_{j-1}}^{t_{j+1}} f(\sigma, x(\sigma)) d\sigma = 2\tau f(t_j, x(t_j)) + \mathcal{O}(\tau^3).$$

- Mehrschrittverfahren

$$x_{\Delta}(t_{j+1}) = x_{\Delta}(t_{j-1}) + 2\tau f(t_j, x_{\Delta}(t_j)), \quad j = 1, \dots, n-1$$

- Verfahren ist explizit: Die gesuchte Größe $x_{\Delta}(t_{j+1})$ taucht nur links des Gleichheitszeichens auf.
- Fehler der Integralapproximation ist $\mathcal{O}(\tau^3)$ (für f hinreichend glatt). Wir vermuten deshalb Konsistenzordnung $p = 2$.

Wie bekommt man höhere Konsistenzordnung?

Idee: Nimm genauere Quadraturformel!

Beispiel: Die Simpson-Regel

$$\int_{t_{j-1}}^{t_{j+1}} f(\sigma, x(\sigma)) d\sigma = \frac{\tau}{3} \left[f(t_{j+1}, x(t_{j+1})) + 4f(t_j, x(t_j)) + f(t_{j-1}, x(t_{j-1})) \right] + \mathcal{O}(\tau^5)$$

- Damit konstruiert man das Milne–Simpson-Verfahren

$$x_{\Delta}(t_{j+1}) = x_{\Delta}(t_{j-1}) + \frac{\tau}{3} \left[f(t_{j+1}, x_{\Delta}(t_{j+1})) + 4f(t_j, x_{\Delta}(t_j)) + f(t_{j-1}, x_{\Delta}(t_{j-1})) \right]$$

- Konsistenzordnung $p = 4$
- Dennoch nur *eine* f -Auswertung pro Zeitschritt!
- Verfahren ist *implizit*: Zur Bestimmung von $x_{\Delta}(t_{j+1})$ muss ein Gleichungssystem gelöst werden.

Allgemeine lineare k -Schritt-Verfahren:

- Beide Beispiel-Verfahren sind linear in f .
- Definiere $f_\tau(t_i)$ als Abkürzung von $f(t_i, x_\Delta(t_i))$.

Definiere das Verfahren

$$\begin{aligned} \alpha_k x_\tau(t_{j+k}) + \alpha_{k-1} x_\tau(t_{j+k-1}) + \dots + \alpha_0 x_\tau(t_j) \\ = \tau \left[\beta_k f_\tau(t_{j+k}) + \beta_{k-1} f_\tau(t_{j+k-1}) + \dots + \beta_0 f_\tau(t_j) \right] \end{aligned} \quad (10.4)$$

mit $|\alpha_0| + |\beta_0| > 0$ und $\alpha_k \neq 0$, (sonst ist es ein $(k-1)$ -Schritt-Verfahren).

- Das Verfahren ist explizit, falls $\beta_k = 0$. Ansonsten ist es implizit.

Existiert eine eindeutige Gitterlösung?

- Explizites IMSV, also $\beta_k = 0$ klar (sogar ohne Einschränkung an τ).

Lemma 10.4. Sei $\beta_k \neq 0$ und $f : \mathbb{R} \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ genüge der Lipschitz-Bedingung

$$\|f(t, x) - f(t, \bar{x})\| \leq L \|x - \bar{x}\| \quad \forall x, \bar{x} \in \mathbb{R}^d, t \in \mathbb{R}.$$

Dann existiert für $\tau < \frac{|\alpha_k|}{|\beta_k|L}$ zu beliebigen Startwerten $x_\Delta(t_0), \dots, x_\Delta(t_{k-1})$ eine eindeutige Gitterfunktion x_Δ des IMSV.

Beweis. • Löse (10.4) nach $x_\Delta(t_{j+k})$ auf:

$$x_\Delta(t_{j+k}) = \tau \frac{\beta_k}{\alpha_k} f(t_{j+k}, x_\Delta(t_{j+k})) + \text{sonstige Terme}$$

- Dies ist eine Fixpunktgleichung.
- $\tau \frac{\beta_k}{\alpha_k} f$ ist Lipschitz-stetig mit Konstante $L^* = \tau \frac{\beta_k}{\alpha_k} L$
- Der Banachsche Fixpunktsatz liefert die Existenz eines eindeutigen Fixpunkts, wenn diese Lipschitzkonstante L^* kleiner 1 ist, wenn also

$$\tau < \frac{|\alpha_k|}{|\beta_k|L}. \quad \square$$

Darstellung durch Polynome: Zum Mehrschrittverfahren (10.4) definiere die Polynome

$$\begin{aligned} \rho(\xi) &= \alpha_k \xi^k + \alpha_{k-1} \xi^{k-1} + \dots + \alpha_0 \\ \sigma(\xi) &= \beta_k \xi^k + \beta_{k-1} \xi^{k-1} + \dots + \beta_0. \end{aligned}$$

Zu den Beispielen:

- explizite Mittelpunktsregel: $\rho(\xi) = \xi^2 - 1$ und $\sigma(\xi) = 2\xi$
- Milne-Simpson: $\rho(\xi) = \xi^2 - 1$ und $\sigma(\xi) = \frac{1}{3}(\xi^2 + 4\xi + 1)$

10.8.3 Konsistenz

Wir brauchen einen neuen Konsistenzbegriff.

- Beschreibt den Zusammenhang zwischen Differenzengleichung und Differentialgleichung.

Plan:

- Ersetzen von x_τ durch die Funktion x , die die DGL erfüllt.
- Lokaler Diskretisierungsfehler $L(x, t, \tau)$ (hier wegen $f(t_j, x(t_j)) = x'(t_j)$)

$$L(x, t, \tau) := \alpha_k x(t + k\tau) + \alpha_{k-1} x(t + (k-1)\tau) + \dots + \alpha_0 x(t) - \tau \left[\beta_k x'(t + k\tau) + \beta_{k-1} x'(t + (k-1)\tau) + \dots + \beta_0 x'(t) \right] \quad (10.5)$$

Definition. Das LMSV (10.4) besitzt die Konsistenzordnung p , wenn

$$L(x, t, \tau) = \mathcal{O}(\tau^{p+1})$$

für alle $x \in C^\infty([t_0, T], \mathbb{R}^d)$ und $\tau \rightarrow 0$ gleichmäßig für alle t, τ gilt.

Diese Definition ist eine Verallgemeinerung des bisherigen Konsistenzbegriffs.

Beispiel: Explizites Euler-Verfahren

- Lineares 1-Schritt-Verfahren mit definierenden Polynomen $\rho(\xi) = \xi - 1$ und $\sigma(\xi) = 1$.
- Zugehöriger lokaler Diskretisierungsfehler

$$\begin{aligned} L(x, t, \tau) &= x(t + \tau) - x(t) - \tau x'(t) \\ &= x(t + \tau) - [x(t) + \tau f(t, x(t))] \\ &= \Phi^{t+\tau, t} x - \Psi^{t+\tau, t} x. \end{aligned}$$

Lemma 10.5. Das lineare Mehrschrittverfahren (10.4) hat genau dann die Konsistenzordnung p , wenn eine der folgenden äquivalenten Bedingungen erfüllt ist:

1. Für beliebige $x \in C^{p+1}([t_0, T], \mathbb{R}^d)$ gilt $L(x, t, \tau) = \mathcal{O}(\tau^{p+1})$ gleichmäßig in allen zulässigen t und τ .
2. $L(Q, 0, \tau) = 0$ für alle Polynome Q von Grad höchstens p
3. $L(\exp, 0, \tau) = \rho(e^\tau) - \tau \sigma(e^\tau) = \mathcal{O}(\tau^{p+1})$

4. Für alle $l = 1, \dots, p$ gilt

$$\sum_{j=0}^k \alpha_j = 0, \quad \sum_{j=0}^k \alpha_j j^l = l \sum_{j=0}^k \beta_j j^{l-1}$$

(dabei gelte $0^0 = 1$).

Beweis. Wir zeigen $1) \implies 2) \implies 3) \implies 4) \implies 1)$.

- 1) \implies 2):
- Für $Q \in \Pi_p$ ist $L(Q, 0, \tau)$ ein Polynom in τ .
 - Dessen Grad ist durch den von Q , also p , beschränkt.
 - Gleichzeitig ist $L(Q, 0, \tau) = \mathcal{O}(\tau^{p+1})$.
 - Also ist $L(Q, 0, \tau) = 0$.

- 2) \implies 3)

$$\exp(\tau) = Q(\tau) + \mathcal{O}(\tau^{p+1})$$

mit Q Taylorpolynom von \exp an der Stelle 0.

Aus der Linearität von L bzgl. des ersten Arguments folgt

$$L(\exp, 0, \tau) = L(Q, 0, \tau) + \mathcal{O}(\tau^{p+1})$$

Wegen 2) ist $L(Q, 0, \tau) = 0$

- 3) \implies 4) Taylor-Entwicklung (nach τ) an der Stelle 0 liefert

$$\begin{aligned} L(\exp, 0, \tau) &= \sum_{l=0}^p \frac{1}{l!} \sum_{j=0}^k \alpha_j j^l \tau^l - \sum_{l=0}^{p-1} \frac{1}{l!} \sum_{j=0}^k \beta_j j^l \tau^{l+1} + \mathcal{O}(\tau^{p+1}) \\ &= \sum_{l=0}^p \frac{1}{l!} \sum_{j=0}^k \alpha_j j^l \tau^l - \sum_{l=1}^p \frac{1}{(l-1)!} \sum_{j=0}^k \beta_j j^{l-1} \tau^l + \mathcal{O}(\tau^{p+1}) \end{aligned}$$

Koeffizientenvergleich bzgl. der τ -Potenzen!

- 4) \implies 1) Taylor-Entwicklung (nach τ) an der Stelle 0 liefert für $x \in C^{p+1}$:

$$L(x, t, \tau) = \sum_{l=0}^p \frac{1}{l!} \sum_{j=0}^k \alpha_j j^l \tau^l x^{(l)}(t) + \mathcal{O}(\tau^{p+1}) - \tau \left[\sum_{l=0}^{p-1} \frac{1}{l!} \sum_{j=0}^k \beta_j j^l \tau^l x^{(l+1)}(t) + \mathcal{O}(\tau^p) \right]$$

Koeffizientenvergleich liefert mit 4) ($= 0$), also folgt die Aussage. \square

Beispiel: $p = 0$.

- Wegen 2) bedeutet das, dass konstante Funktionen exakt behandelt werden.

- D.h., das Anfangswertproblem

$$x' = 0, \quad x(t_0) = x_0$$

wird exakt gelöst.

- Nach 4) ist die dazugehörige Bedingungsgleichung

$$\sum_{j=0}^k \alpha_j = 0, \quad \text{bzw.} \quad \rho(1) = 0.$$

Beispiel: $p = 1$.

- Zusätzlich die Bedingungsgleichung

$$\sum_{j=0}^k \alpha_j j = \sum_{j=0}^k \beta_j$$

- In Polynomform

$$\rho'(1) = \sigma(1).$$

Diese Bedingungen entsprechen dem $\sum_{j=1}^s b_j = 1$ bei Runge–Kutta-Verfahren.

10.8.4 Stabilität

Bei Einschrittverfahren galt:

- Wenn ein Verfahren mit einer gewissen Ordnung konsistent ist, dann konvergiert es auch mit dieser Ordnung.

So einfach ist es bei Mehrschrittverfahren leider nicht!

Beispiel. Betrachte das (bis auf Normierung $\alpha_k = 1$) eindeutige explizite Zweischnittverfahren der Konsistenzordnung $p = 3$

$$\rho(\xi) = \xi^2 + 4\xi - 5, \quad \sigma(\xi) = 4\xi + 2. \quad (10.6)$$

- Anwendung auf

$$x'(t) = 0, \quad x(0) = 1$$

mit den Startwerten

$$x_{\Delta}(0) = 1, \quad x_{\Delta}(\tau) = 1 + \tau\varepsilon.$$

- Man erhält die Gitterfunktion

$$x_{\Delta}(n\tau) = 1 + \tau\varepsilon \frac{(1 - (-5)^n)}{6}$$

(Beachte: 1 und -5 sind gerade die Nullstellen von ρ !)

- Es folgt für alle $\varepsilon \neq 0$ dass

$$\lim_{n \rightarrow \infty} |x_{\Delta}(n\tau)| = \infty$$

- obwohl für den Startwert gilt, dass

$$\lim_{\tau \rightarrow 0} x_{\Delta}(\tau) = 1$$

und die exakte Lösung $x(t) = 1$ beliebig glatt ist.

Das heißt: keine Konvergenz, sogar ganz anderes Verhalten.

- Das Mehrschrittverfahren ist instabil.
- Es ist komplett unbenutzbar!

Definition. Ein lineares Mehrschrittverfahren heißt stabil (oder nullstabil oder D-stabil), wenn die lineare homogene Differenzgleichung

$$\sum_{j=0}^k \alpha_j x_{\tau}(t_{j+k}) = 0, \quad k = 0, 1, \dots$$

bei beliebigen Startwerten stabil ist. Das wiederum heißt: Die Folge bleibt für alle Startwerte beschränkt.

Beachte: Die homogene Differenzgleichung ist gerade das Mehrschrittverfahren, angewandt auf $x' = 0$.

Satz 10.6 (Dahlquistsche Wurzelbedingung). Ein lineares Mehrschrittverfahren ist genau dann stabil, wenn die Nullstellen ξ von ρ die Dahlquist'sche Wurzelbedingung erfüllen, d.h.

- $|\xi| \leq 1$,
- wenn $|\xi| = 1$, dann ist ξ einfache Nullstelle.

Beispiel. Für das IMSV (10.6) ist $\rho(\xi) = \xi^2 + 4\xi - 5 = (\xi - 1)(\xi + 5)$, d.h. dieses Verfahren ist nicht stabil.

Beispiel. Für die explizite Mittelpunktsregel sowie für das Milne–Simpson-Verfahren ist $\rho(\xi) = \xi^2 - 1 = (\xi + 1)(\xi - 1)$. Beide sind also stabil.

Beispiel. Einschrittverfahren:

- Sofern sie konsistent sind muss $\rho(1) = 0$, und deshalb $\rho(\xi) = \xi - 1$ gelten.
- Alle konsistenten Einschrittverfahren sind deshalb automatisch stabil.

- Beim Konvergenzbeweis für Einschrittverfahren mussten wir auf Stabilität deshalb gar nicht eingehen!

Wenn man sich die Bestimmungsgleichungen in 4) genauer anschaut, erkennt man:

- Es gibt k -Schritt-Verfahren mit Ordnung $2k$, aber keine mit höherer Ordnung.
- Es gibt explizite k -Schritt-Verfahren mit Ordnung $2k - 1$, aber keine mit höherer Ordnung.

Wenn man sich auf *stabile* Verfahren beschränkt verliert man in etwa die halbe Ordnung.

Satz 10.7. *Die Konsistenzordnung p eines stabilen linearen k -Schrittverfahrens unterliegt der Beschränkung*

1. $p \leq k + 2$, wenn k gerade ist,
2. $p \leq k + 1$, wenn k ungerade ist,
3. $p \leq k$, wenn $\frac{\beta_k}{\alpha_k} \leq 0$ ist, also insbesondere auch für explizite Verfahren.

Diese Aussage nennt man die erste Dahlquistsche Schranke.

Man kann zeigen dass sie scharf ist.

10.8.5 Konvergenz

Definition. Sei $x \in C^1([t_0, T], \mathbb{R}^d)$ die Lösung eines Anfangswertproblems $x' = f(t, x)$, $x(t_0) = x_0$. Ein Mehrschrittverfahren konvergiert gegen diese Lösung, wenn

$$\lim_{\tau \rightarrow 0} x_{\Delta}(t) = x(t) \quad \text{für alle } t \in \Delta_{\tau} \cap [t_0, T]$$

gilt, sobald die Startwerte

$$\lim_{\tau \rightarrow 0} x_{\Delta_{\tau}}(t_0 + j\tau) = x_0 \quad j = 0, \dots, k - 1$$

erfüllen. Wenn ein lineares Mehrschrittverfahren für beliebige Anfangswertprobleme mit hinreichend glatter rechter Seite konvergiert, so heißt es konvergent.

Das Phänomen aus dem obigen Beispiel ist von grundsätzlicher Natur, wie folgender Satz zeigt.

Satz 10.8. *Ein konvergentes lineares Mehrschrittverfahren ist notwendigerweise stabil und konsistent, speziell gilt*

$$\rho'(1) = \sigma(1) \neq 0.$$

Die Umkehrung gilt aber auch:

Satz 10.9. *Ein stabiles und konsistentes lineares Mehrschrittverfahren ist konvergent.*

Dies ist ein gutes Beispiel dessen, was manche den Hauptsatz der Numerik nennen:

Konsistenz und Stabilität implizieren Konvergenz.

Aussagen zur Konvergenzordnung sind natürlich auch möglich, gehen aber über den Rahmen dieser Vorlesung hinaus.

11 Steife Differentialgleichungen und implizite Verfahren

11.1 Wiederholung: Gewöhnliche Differentialgleichungen und Anfangswertprobleme

Das folgende Unterkapitel wiederholt ein paar wichtige Konzepte aus dem vorangegangenen Kapitel. Es existiert hauptsächlich als Vorlage für eine Vorlesung, die mit steifen Differentialgleichungen anfängt, und deshalb etwas Vorwissen wiederholen muss. Beim Lesen dieses Dokuments kann es übersprungen werden.

11.1.1 Gewöhnliche Differentialgleichungen

Gewöhnliche Differentialgleichung:

$$x'_i = f_i(t, x_1, \dots, x_d), \quad i = 1, \dots, d$$

wobei $(t, x) \in \mathbb{R} \times \mathbb{R}^d$ und $f_i: \Omega \rightarrow \mathbb{R}^d$, $\Omega \subseteq \mathbb{R} \times \mathbb{R}^d$ offen.

- Die Variable t ist häufig als Zeit interpretierbar, man spricht daher häufig von *Evolutionsproblemen*.
- x heißt *Zustandsvektor*.
- \mathbb{R}^d mit $x \in \mathbb{R}^d$ heißt *Zustandsraum*.
- $\mathbb{R} \times \mathbb{R}^d$ heißt *erweiterter Zustandsraum*.

Achtung: Traditionell verwendet man das gleiche Symbol für Zustände $x \in \mathbb{R}^d$ und Funktionen in den Zustandsraum $x: \mathbb{R} \rightarrow \mathbb{R}^d$. Nicht verwirren lassen!

Beispiele

1. (Lineare skalare Differentialgleichung) Finde $x: \mathbb{R} \rightarrow \mathbb{R}$ so dass

$$x' = -kx$$

(beschreibt z.B. radioaktiven Zerfall)

2. Allgemeiner: Finde $x: \mathbb{R} \rightarrow \mathbb{R}^d$ so dass

$$x' = -Ax, \quad A \in \mathbb{R}^{d \times d}.$$

3. Höhere Ableitungen können wegtransformiert werden.

11.1.2 Anfangswertprobleme

Lösungen von gewöhnlichen Differentialgleichungen sind normalerweise nicht eindeutig.

Beispiel. Für alle $c \in \mathbb{R}$ löst $x(t) = ce^{-kt}$ die Gleichung

$$x' = -kx.$$

Deshalb: *Zusatzinformation.* Schreibe Anfangsbedingung

$$x(t_0) = x_0$$

vor. $x_0 \in \mathbb{R}^d$ heißt *Anfangswert*.

Bezeichnung:

Differentialgleichung + Anfangsbedingung = Anfangswertproblem (AWP/IVP)

11.1.3 Existenz und Eindeutigkeit

Auch AWP's können mehr als eine Lösung haben.

Beispiel. Betrachte $x' = \sqrt{|x|}$, $x(0) = 0$.

- $f(x) = \sqrt{|x|}$ ist stetig auf \mathbb{R} , es existiert also eine Lösung, z.B. $x(t) = 0 \forall t$
- Aber:
 - Wähle ein $c > 0$
 - Definiere

$$\tilde{x}(t) := \begin{cases} 0 & \text{falls } 0 \leq t \leq c \\ \frac{1}{4}(t-c)^2 & \text{falls } c < t \end{cases}$$

- \tilde{x} löst das Problem.

Es gibt also unendlich viele Lösungen!

Mit einer einfachen Zusatzforderung an f kann man Eindeutigkeit erhalten.

Satz 11.1 (Picard, Lindelöf). *Betrachte das Anfangswertproblem*

$$x' = f(t, x), \quad x(t_0) = x_0$$

auf dem erweiterten Zustandsraum $\Omega \subset \mathbb{R} \times \mathbb{R}^d$ mit $(t_0, x_0) \in \Omega$. f sei stetig, und bzgl. x lokal Lipschitz-stetig.

Dann besitzt das AWP eine eindeutige Lösung.

11.1.4 Evolution und Phasenfluss

Falls die Bedingungen des Satzes von Picard–Lindelöf gelten, so kann man eine elegante neue Notation einführen.

Sei $(t_0, x_0) \in \Omega$. Bezeichne mit $J_{\max}(t_0, x_0)$ das maximale Zeitintervall, auf dem eine Lösung des dazugehörigen AWP existiert.

Zu jedem Anfangswert (t_0, x_0) gibt es eine eindeutige Lösung, d.h. zu jedem Anfangswert (t_0, x_0) ist der Wert $x(t)$ für alle $J_{\max}(t_0, x_0)$ eindeutig bestimmt.

Definition. Für alle $t_0, t \in J_{\max}(t_0, x_0)$ heißt

$$\Phi^{t, t_0}: x_0 \mapsto x(t)$$

Evolution der Differentialgleichung $x' = f(t, x)$.

- Wohldefiniert, weil das AWP für jedes x_0 eine eindeutige Lösung hat.

Man kann also schreiben: $x(t) = \Phi^{t, t_0} x_0$.

Für autonome Gleichungen $x' = f(x)$ kann man die Abhängigkeit von t_0 weglassen. Der Evolutionsoperator $\Phi^t x_0 = x(t)$ heißt dann „Phasenfluss“.

11.1.5 Das explizite Euler-Verfahren

Ziel: Finde eine numerische Approximation der Lösung $x \in C^1([t_0, T], \mathbb{R}^d)$ des AWP

$$x' = f(t, x), \quad x(t_0) = x_0$$

Vorgehensweise:

- Unterteile das Intervall $[t_0, T]$ durch $n + 1$ Zeitpunkte

$$t_0 < t_1 < t_2 < \dots < t_n = T.$$

- Die Menge der Zeitpunkte heißt *Gitter* $\Delta := \{t_0, t_1, \dots, t_n\}$
- Schrittweite: $\tau_j := t_{j+1} - t_j$ für $j = 0, \dots, n - 1$
- Maximale Schrittweite: $\tau_\Delta = \max_{j=0, \dots, n-1} \tau_j$

Wir suchen eine Gitterfunktion

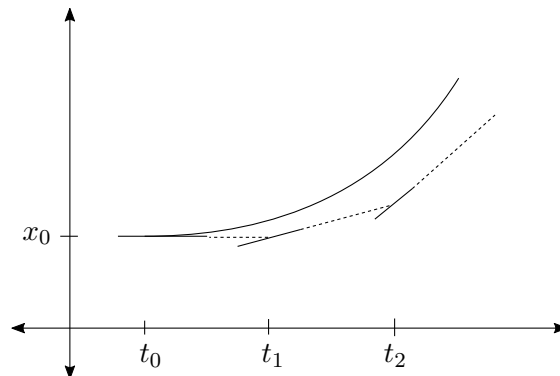
$$x_\Delta: \Delta \rightarrow \mathbb{R}^d,$$

welche die Lösung des AWP an den Gitterpunkten möglichst gut approximiert.

(Manchmal interpretieren wir so ein x_Δ auch als eine Funktion $[t_0, T] \rightarrow \mathbb{R}^d$, die die Werte an den Gitterpunkten linear interpoliert.)

Hoffnung natürlich: Wenn das Gitter immer feiner wird (wenn also τ_Δ immer kleiner wird), wird der Unterschied zwischen x und x_Δ immer kleiner.

Das explizite Euler-Verfahren Nach L. Euler (1786), auch Eulersches Polygonzugverfahren genannt.



1. $x_{\Delta}(t_0) = x_0$

2. Für $t \in [t_j, t_{j+1}]$:

$$x_{\Delta}(t) = x_{\Delta}(t_j) + (t - t_j)f(t_j, x_{\Delta}(t_j))$$

3. Insbesondere:

$$x_{\Delta}(t_{j+1}) = x_{\Delta}(t_j) + \tau_j f(t_j, x_{\Delta}(t_j))$$

Keine Gleichungssysteme zu lösen \rightarrow das Verfahren ist explizit.

Beachte: Berechnung durch eine Zweiterm-Rekursion

1. $x_{\Delta}(t_0) = x_0$

2. $x_{\Delta}(t_{j+1}) = \Psi^{t_{j+1}, t_j} x_{\Delta}(t_j), \quad j = 0, 1, \dots, n-1$

mit Ψ unabhängig von Δ .

- Die Funktion Ψ heißt *diskrete Evolution* des expliziten Euler-Verfahrens.

11.1.6 Konsistenz

Der Fehler der Lösung, also der Unterschied $x - x_{\Delta}$, besteht aus zwei Beiträgen:

- Jeder einzelne Schritt produziert einen Fehler.
- Da man nach dem ersten Schritt immer von einem fehlerbehafteten Wert startet, bekommt man auch falsche Ableitungen f .

Mit *Konsistenz* bezeichnet man das *lokale* Verhältnis zwischen der Evolution Φ und der diskreten Evolution Ψ .

Definition. Sei $(t, x) \in \Omega$. Die Differenz

$$\varepsilon(t, x, \tau) = \Phi^{t+\tau, t} x - \Psi^{t+\tau, t} x$$

heißt Konsistenzfehler von Ψ .

Eine diskrete Evolution heißt *konsistent*, wenn der Konsistenzfehler keine konstanten oder linearen Terme in τ hat. Formaler definiert man das wie folgt:

Definition. Eine diskrete Evolution Ψ heißt *konsistent*, falls

$$\begin{aligned} \Psi^{t, t} x = x &= \Phi^{t, t} x && \text{für alle } (t, x) \in \Omega, \\ \frac{d}{d\tau} \Psi^{t+\tau, t} x \Big|_{\tau=0} = f(t, x) &= \frac{d}{d\tau} \Phi^{t+\tau, t} x \Big|_{\tau=0} && \text{für alle } (t, x) \in \Omega. \end{aligned}$$

Beispiel. Das explizite Euler-Verfahren

$$\Psi^{t+\tau, t} x = x + \tau f(t, x)$$

ist konsistent, denn $\frac{\partial \Psi}{\partial \tau} = \frac{\partial}{\partial \tau} (x + \tau f(t, x)) = f(t, x)$.

Wir wollen eine quantitative Version von Konsistenz.

Definition. Eine diskrete Evolution Ψ besitzt *Konsistenzordnung* p , wenn es eine Konstante $C > 0$ unabhängig von t und x gibt, so dass

$$\varepsilon(t, x, \tau) \leq C\tau^{p+1}.$$

Ja, der Exponent ist wirklich $p + 1$!

Beispiel. Das Euler-Verfahren hat die Konsistenzordnung 1.

11.1.7 Konvergenz

- Konsistenz ist ein lokales Phänomen, d.h. es betrachtet den Fehler nur in der Nähe eines festen t .
- Wir betrachten jetzt, wie gut eine Lösung $x \in C^1([t_0, T])$ insgesamt approximiert wird.
- Wir hoffen, dass für kleinere $\tau_\Delta := \max \tau_j$ die Approximation immer besser wird.
- und das schnell!

Definition. Der Vektor der Approximationsfehler auf dem Gitter Δ

$$\varepsilon_\Delta: \Delta \rightarrow \mathbb{R}^d, \quad \varepsilon_\Delta(t) = x(t) - x_\Delta(t)$$

heißt Gitterfehler. Seine Norm

$$\|\varepsilon_\Delta\|_\infty = \max_{t \in \Delta} |\varepsilon_\Delta(t)|$$

heißt Diskretisierungsfehler.

Definition. Zu jedem Gitter Δ auf $[t_0, T]$ sei eine Gitterfunktion x_Δ gegeben. Die Familie dieser Gitterfunktionen konvergiert mit Ordnung $p \in \mathbb{N}$ gegen $x \in C^1([t_0, T])$, falls eine Konstante $C > 0$ existiert, so dass

$$\|\varepsilon_\Delta\|_\infty \leq C\tau_\Delta^p$$

für alle τ_Δ klein genug.

Alternative Notation: $\|\varepsilon_\Delta\|_\infty = \mathcal{O}(\tau_\Delta^p)$.

- Konvergenz hängt eng mit dem Konsistenzfehler zusammen.
- Das ist praktisch: Der Konsistenzfehler lässt sich häufig direkt am Verfahren ablesen.

Satz 11.2 (Deuffhard und Bornemann [3, Satz 4.10]). Sei

$$\Psi^{t+\tau, t}x = x + \tau\psi(t, x, \tau),$$

und ψ lokal Lipschitz-stetig in x . Die diskrete Evolution sei konsistent mit Ordnung p . Dann definiert die diskrete Evolution Ψ für alle Gitter Δ mit hinreichend kleiner Zeitschrittweite τ_Δ eine Gitterfunktion x_Δ zum Anfangswert $x_\Delta(t_0) = x_0$. Die Familie dieser Gitterfunktionen konvergiert mit Ordnung p gegen die Lösung x des AWP, d.h.

$$\|\varepsilon_\Delta\|_\infty := \max_{t \in \Delta} |x(t) - x_\Delta(t)| = \mathcal{O}(\tau_\Delta^p).$$

Beispiel (Explizites Euler-Verfahren).

$$x_{j+1} = x_j + \tau_j f(t_j, x_j)$$

- Konsistenzordnung 1 (der lokale Fehler verhält sich wie τ_Δ)
- Inkrementfunktion $\psi(t, x, \tau) = f(t, x)$ ist lokal Lipschitz-stetig in x .

⇒ Verfahren konvergiert mit Ordnung 1, d.h. $\|\varepsilon_\Delta\|_\infty = \mathcal{O}(\tau_\Delta^1)$.

⇒ halber Fehler bedeutet doppelte Anzahl von Zeitschritten.

⇒ doppelte Anzahl von f -Auswertungen ⇒ doppelter Aufwand.

11.1.8 Explizite Runge–Kutta-Verfahren

[Nach: Carl Runge, 1856 (Bremen) – 1927 (Göttingen), Martin Wilhelm Kutta, 1867 (Pitschen/Oberschlesien) – 1944 (Fürstfeldbruck)]

Klassische Konstruktion von Verfahren mit einer höheren Konsistenzordnung.

Insbesondere: Hohe Ordnung ohne Ableitungen von f .

s -stufiges explizites Runge-Kutta-Verfahren:

$$1. k_i := f\left(t + c_i\tau, x + \tau \sum_{j=1}^{i-1} a_{ij}k_j\right) \quad \forall i = 1, \dots, s$$

$$2. \Psi^{t+\tau,t}x = x + \tau \sum_{i=1}^s b_i k_i.$$

Die Größen $k_i = k_i(t, x, \tau)$ heißen *Stufen* des Verfahrens.

Koeffizienten:

$$b = (b_1, \dots, b_s)$$

$$c = (c_1, \dots, c_s)$$

$$A = \begin{pmatrix} 0 & & & & & \\ a_{21} & 0 & & & & \\ \vdots & \ddots & & & & \\ a_{s1} & a_{s2} & \dots & a_{s,s-1} & 0 & \end{pmatrix} \in \mathbb{R}^{s \times s}$$

Traditionell notiert man die Koeffizienten im Butcher-Schema (nach John C. Butcher, 1933 in Auckland)

$$\begin{array}{c|c} c^T & A \\ \hline & b \end{array}$$

Beispiele:

expl. Euler-Verfahren $\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$

Verfahren von Runge $\begin{array}{c|cc} 0 & 0 & \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array}$

Das klassische 4-stufige Runge-Kutta-Verfahren:

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

- Eine Funktionsauswertung pro Stufe
- $s + s + \frac{(s-1)s}{2}$ Parameter bei s Stufen

Bei geschickter Wahl der Koeffizienten erhält man Verfahren höherer Ordnung.

Lemma 11.1. Ein explizites Runge–Kutta-Verfahren ist genau dann konsistent für alle $f \in C(\Omega, \mathbb{R}^d)$, wenn $\sum_{i=1}^s b_i = 1$.

Außerdem betrachten wir nur folgende Vereinfachung:

Lemma 11.2. Ein explizites Runge–Kutta-Verfahren ist genau dann invariant unter Autonomisierung, wenn es konsistent ist, und

$$c_i = \sum_{j=1}^{i-1} a_{ij} \quad \text{für } i = 1, \dots, s$$

erfüllt.

Diese Verfahren sind *Invariant unter Autonomisierung*.

11.2 Steife Differentialgleichungen

Explizites Euler-Verfahren

$$x_{k+1} = x_k + \tau f(t_k, x_k)$$

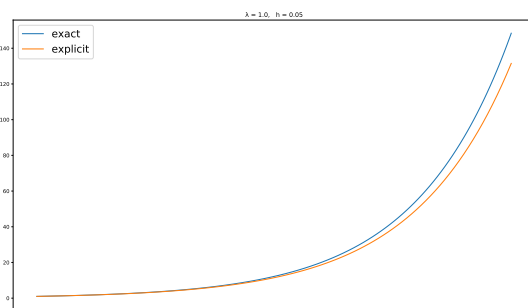
- Konvergent mit Ordnung 1.

Löse damit das Anfangswertproblem

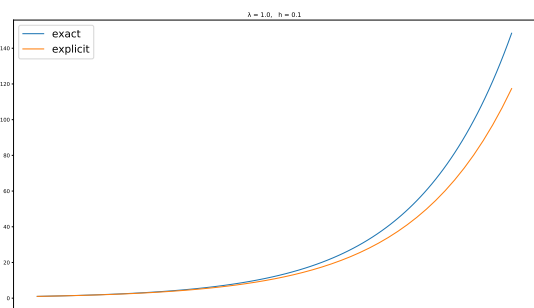
$$x' = \lambda x, \quad x(0) = 1, \quad \lambda \in \mathbb{R}.$$

Die folgenden Bilder zeigen Rechnung für $\lambda = 1$, mit verschiedenen Schrittweiten:

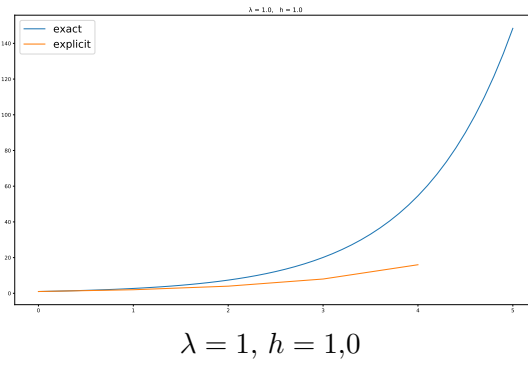
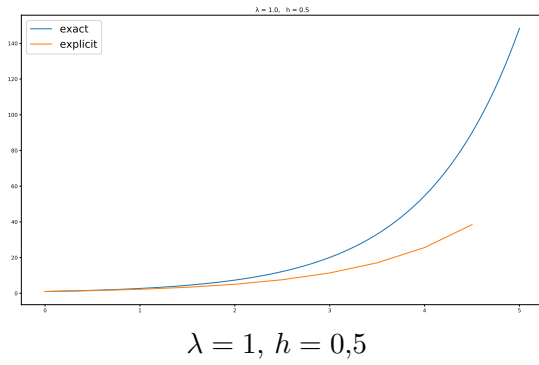
(Die Tatsache dass die orange Linien nicht bis zum Ende des Zeitintervalls gehen sind Zeichen schlechter Programmierung, aber kein mathematisches Problem.)



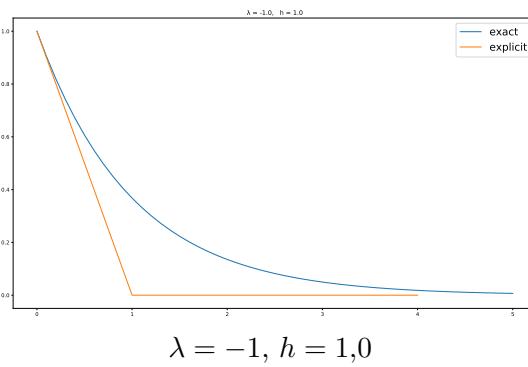
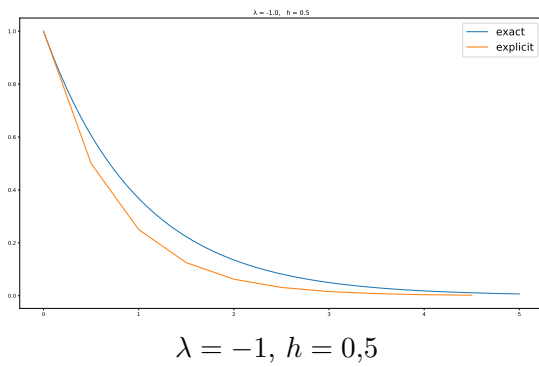
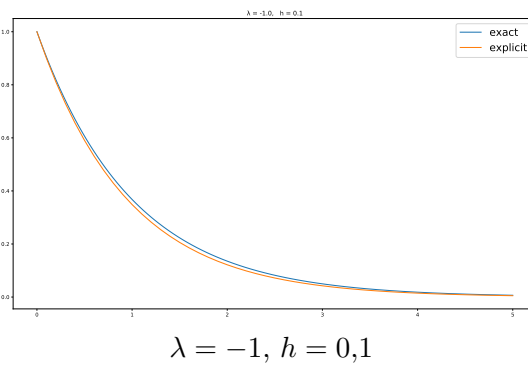
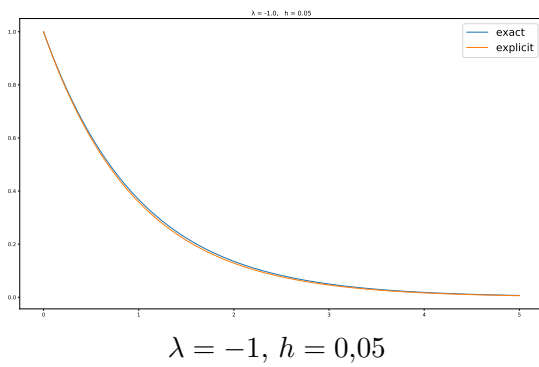
$\lambda = 1, h = 0,05$



$\lambda = 1, h = 0,1$

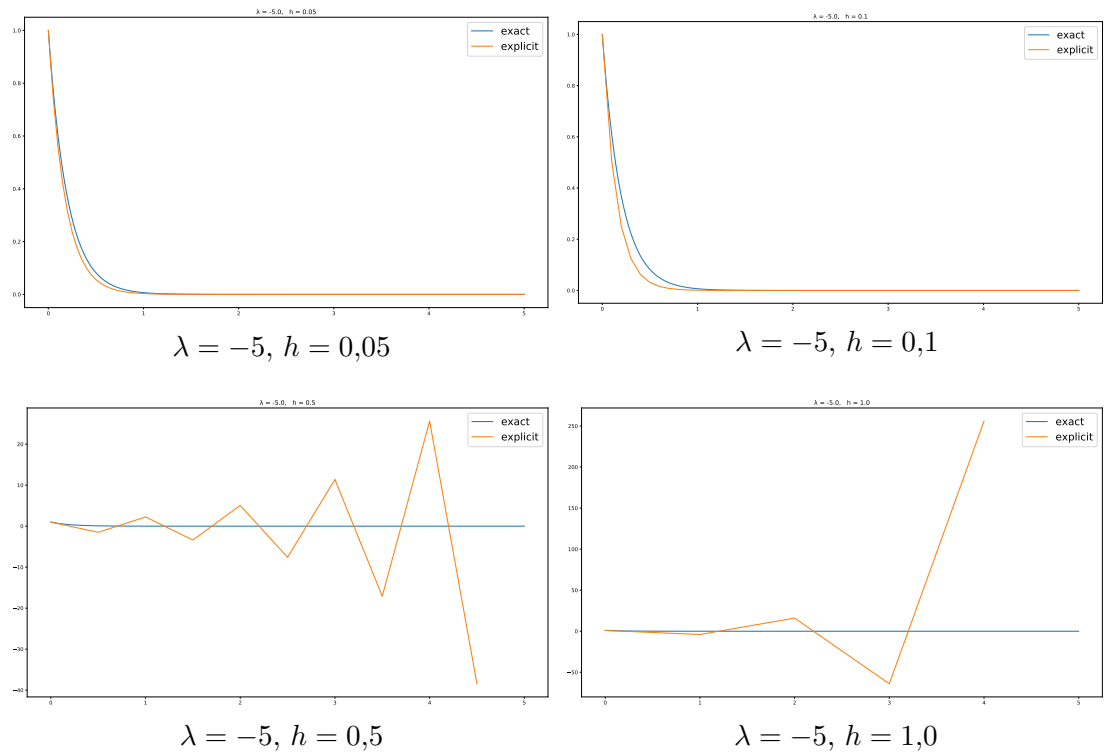


Als nächstes probieren wir ein negatives λ , hier z.B. $\lambda = -1$:



Die letzte Rechnung sieht ein wenig seltsam aus, aber der Rest ist okay.

Probieren wir noch $\lambda = -5$:



Fazit:

- Keine Überraschungen, falls $\lambda \geq 0$
- Falls $\lambda < 0$, dann produziert das explizite Euler-Verfahren nur dann qualitativ richtige Ergebnisse, falls der Zeitschritt $\tau < 1/|\lambda|$ ist.
- Für $\lambda < 0$, $\tau > 1/|\lambda|$ ist das Verfahren instabil.

Nachrechnen:

Expliziter Euler:

$$x_{k+1} = x_k + \tau f(t_k, x_k) = x_k + \tau \lambda x_k = (1 + \lambda \tau) x_k = (1 + \lambda \tau)^{k+1} x_0$$

Fall 1: $\lambda > 0$

- Lösung $x(t) = \exp(\lambda t)$ monoton steigend in t
- Diskrete Lösung: monoton steigend in k , da $1 + \lambda \tau > 1$.

Fall 2: $\lambda < 0$

- Lösung $x(t) = \exp(\lambda t)$ monoton fallend und positiv
- x_k monoton fallend und positiv nur dann, wenn $0 < 1 + \lambda \tau < 1 \iff \tau \leq 1/|\lambda|$.

Falls $\lambda < 0$ und $\tau > \frac{1}{|\lambda|}$

- Diskrete Lösung oszilliert.

Falls $\lambda < 0$ und sogar $\tau > \frac{2}{|\lambda|}$

- Diskrete Lösung ist unbeschränkt.

11.2.1 Steifheit und Kondition

Für Euler- und Runge-Kutta-Verfahren hatten wir Konvergenzaussagen der Form

$$\|\varepsilon_\Delta\|_\infty \leq C\tau_\Delta^p$$

bewiesen.

Problem: Diese Aussagen sind *asymptotisch*.

- Der Fehler wird kleiner, wenn wir ein hinreichend kleines τ_Δ weiter verkleinern.
- Die Konstante C ist aber unbekannt: Wir wissen nicht, *wie* klein τ_Δ sein muss, um eine vernünftige Genauigkeit zu erzielen.

Man kann C nur in seltenen Fällen exakt ausrechnen.

Stattdessen: *Qualitativ* verstehen, wann C groß sein kann.

Angenommen, wir erhalten für ein gegebenes τ_Δ eine gute Approximation der Lösung.

Dann können wir davon ausgehen, dass das nicht zufällig so ist: Für einen leicht gestörten Anfangswert $x_0 + \delta x_0$ erwarten wir dann auch eine gute Approximation der gestörten Lösung.

Erinnerung: Intervallweise Kondition eines AWP:

$$x' = f(t, x) \quad x(t_0) = x_0 \quad t \in [t_0, T].$$

Störung der Eingabedaten $x_0 \mapsto x_0 + \delta x_0$ führt zu einer Störung der Lösung $x(t) \mapsto x(t) + \delta x(t)$ für alle $t \in [t_0, T]$.

Die intervallweise Kondition $\kappa[t_0, T]$ ist die kleinste Zahl, für die

$$\|\delta x\|_\infty \leq \kappa[t_0, T] \cdot \|\delta x_0\|.$$

Analog führen wir eine diskrete Kondition κ_Δ ein: die Auswirkung einer Störung des Anfangswerts auf eine von einem numerischen Verfahren erzeugte Gitterfunktion

$$\|\delta x_\Delta\|_\infty \leq \kappa_\Delta \cdot \|\delta x_0\|.$$

Wenn ein Verfahren für x_0 und $x_0 + \delta x_0$ (mit δx_0 klein) vernünftige Lösungen liefert, dann muss

$$\kappa_\Delta \approx \kappa[t_0, T]$$

gelten.

Umgekehrt bedeutet $\kappa_\Delta \gg \kappa[t_0, T]$ dass das Verfahren völlig unbrauchbar ist, denn es reagiert auf kleine Störungen völlig anders als das eigentliche Problem.

⇒ Das Gitter ist dann noch zu grob, da für jedes konvergente Verfahren

$$\kappa_\Delta \rightarrow \kappa[t_0, T] \quad \text{für } \tau_\Delta \rightarrow 0.$$

Die Beziehung

$$\kappa_\Delta \approx \kappa[t_0, T]$$

ist eine qualitative Minimalforderung an ein Verfahren + Wahl des Zeitschritts.

- Für die bisher vorgestellten Verfahren gibt es Anfangswertprobleme, für die $\kappa_\Delta \approx \kappa[t_0, T]$ erst für sehr kleine τ_Δ gilt.
- Solche Probleme nennt man *steif*.

Ungewöhnlich:

- Es gibt keine mathematisch präzise Definition des Begriffs „steif“.
- Eine Verfahrensklasse klassifiziert die Probleme!

11.2.2 Beispiel: Wieder das Modellproblem

Wieder das Modellproblem

$$x' = \lambda x, \quad x(0) = 1.$$

Wie ist die Kondition dieses AWP?

Lösung des AWP:

$$x(t) = 1 \cdot e^{\lambda t}.$$

Betrachte jetzt stattdessen einen gestörten Startwert: $x(0) = 1 + d$.

- Lösung davon:

$$x(t) = (1 + d) \exp(\lambda t) = \exp(\lambda t) + d \exp(\lambda t),$$

also ist $\delta x = d \exp(\lambda t)$ die Störung des Resultats.

Es folgt

- $\kappa[0, T] = e^{\lambda T}$ falls $\lambda \geq 0$,
- $\kappa[0, T] = 1$ falls $\lambda \leq 0$.

Diskrete Kondition des expliziten Euler-Verfahrens:

$$x_\Delta(t_{k+1}) = (1 + \tau\lambda)x_\Delta(t_k) = (1 + \tau\lambda)^{k+1}x_0$$

ist linear in x_0 , deshalb gilt

$$\kappa_\Delta = \max_{0 \leq k \leq n-1} |1 + \tau\lambda|^{k+1}$$

Fall 1: $\lambda \geq 0$. Dann ist $\kappa_\Delta = (1 + \tau\lambda)^n$. Wegen $1 + \tau\lambda \leq e^{\tau\lambda}$ gilt

$$\kappa_\Delta = (1 + \tau\lambda)^n \leq \exp(n\tau\lambda) = e^{\lambda T}.$$

Also ist $\kappa_\Delta \approx \kappa[0, T]$, das AWP ist nichtsteif.

Fall 2: $\lambda < 0$

$$\kappa_\Delta = \max_{0 \leq k \leq n-1} |1 - \tau\lambda|^k = |1 - \tau\lambda|^{n-1}.$$

Falls $\tau < 2/|\lambda|$ so ist $\kappa_\Delta \leq 1 = \kappa[0, T]$.

Andererseits gilt für $\tau\Delta \gg 2/|\lambda|$

$$\kappa_\Delta = |1 - \tau\lambda|^n \gg 1 = \kappa[0, T].$$

Das Problem ist steif.

11.2.3 Stabilität

Wir betrachten noch einmal das vorige gestörte AWP

$$x' = \lambda x, \quad x(0) = 1 + d.$$

Wie verhält sich die Störung $d \exp(\lambda t)$? Drei Fälle:

1. $\lambda > 0$: Die Störung wächst exponentiell mit t . Lösen der Gleichung für große t ist kaum sinnvoll, bzw. sehr schwierig.
2. $\lambda = 0$: Die Störung bleibt für alle t in konstanter Größe erhalten.
3. $\lambda < 0$: Für große t wird die Störung „von alleine“ immer kleiner!

Betrachtet man die Auswirkung von Störungen nicht auf ein beschränktes Intervall $[t_0, T]$, sondern für alle Zeiten $[t_0, \infty)$, dann spricht man statt von Kondition meistens von *Stabilität*.

Die obige Dreiteilung ist typisch. Wir machen daraus eine Definition.

Definition. Sei (t_0, x_0) so, dass $\Phi^{t, t_0} x_0$ für alle $t \geq t_0$ existiert. Die Lösung des AWP's heißt

- 2) (Lyapunov)-stabil, falls zu jedem $\varepsilon > 0$ ein $\delta > 0$ existiert, so dass

$$\|\Phi^{t, t_0} x - \Phi^{t, t_0} x_0\| \leq \varepsilon$$

für alle $t \geq t_0$ und $\|x - x_0\| \leq \delta$,

- 3) asymptotisch stabil, falls es zusätzlich ein $\delta_0 > 0$ gibt, so dass

$$\lim_{t \rightarrow \infty} \|\Phi^{t, t_0} x - \Phi^{t, t_0} x_0\| = 0$$

falls $\|x - x_0\| \leq \delta_0$,

- 1) instabil, falls weder 2) noch 3) gelten.

Achtung: Dieser Stabilitätsbegriff hat nichts mit der Stabilität von Algorithmen zu tun. Es kann anspruchsvoll bis zu schwierig sein, die Stabilität von DGLn zu bestimmen.

11.2.4 Das implizite Euler-Verfahren

Expliziter Euler:

$$x_{k+1} = x_k + \tau f(t_k, x_k)$$

Impliziter Euler:

$$x_{k+1} = x_k + \tau f(t_{k+1}, x_{k+1})$$

Implizit bedeutet: In jedem Schritt muss ein Gleichungssystem gelöst werden.
Betrachte wieder das AWP

$$x' = \lambda x, \quad x(0) = 1, \quad \lambda \in \mathbb{R}$$

Implizites Euler-Verfahren:

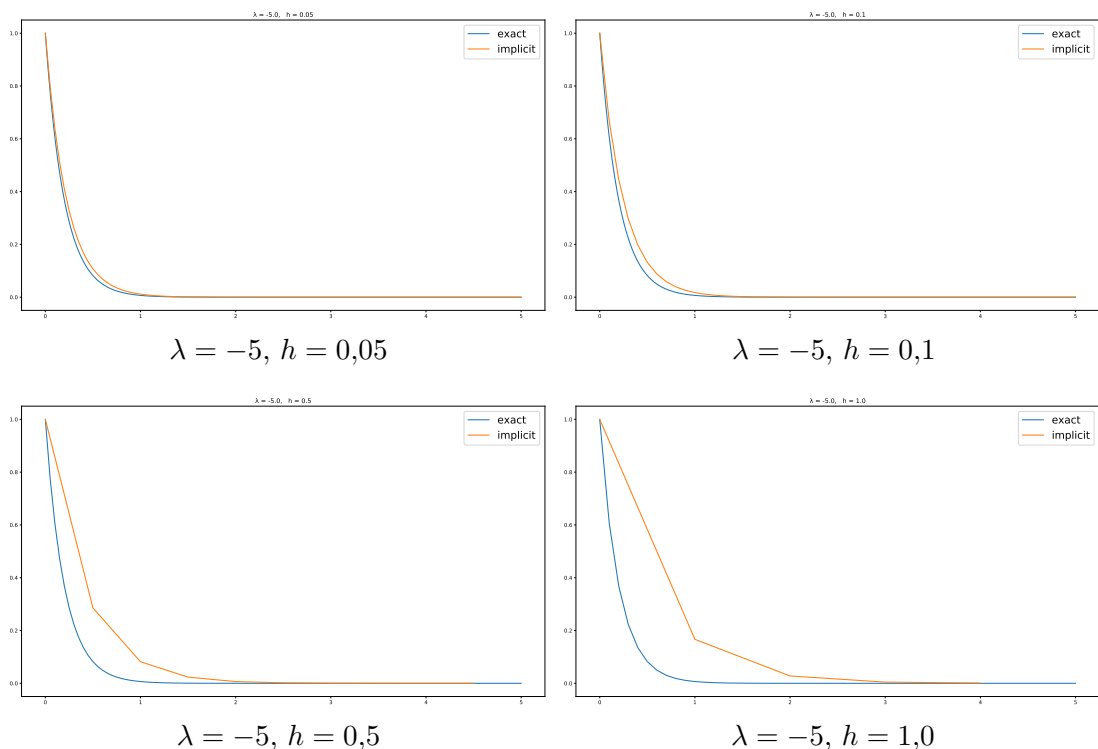
$$\begin{aligned} x_{k+1} &= x_k + \tau f(t_{k+1}, x_{k+1}) = x_k + \tau \lambda x_{k+1} \\ \implies x_{k+1} &= \frac{x_k}{1 - \tau \lambda} = \left(\frac{1}{1 - \tau \lambda} \right)^{k+1} x_0 \end{aligned}$$

Wenn $\lambda < 0$, so ist

$$0 < \frac{1}{1 - \tau \lambda} < 1$$

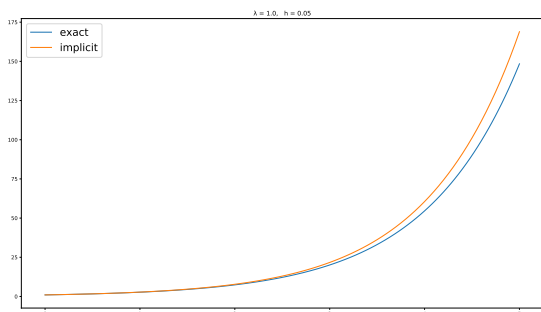
für alle $\tau > 0$. Das Verfahren ist für alle $\tau > 0$ stabil.

Das probieren wir wieder numerisch aus. Hier ist das implizite Euler-Verfahren für $x' = \lambda x$ mit $\lambda = -5$:

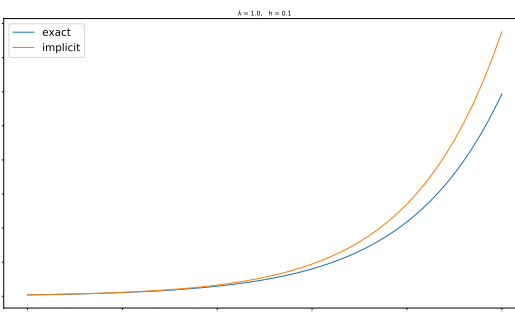


Die letzte Rechnung (mit $h = 1,0$) ist zwar nicht mehr sonderlich präzise, aber stabil ist sie.

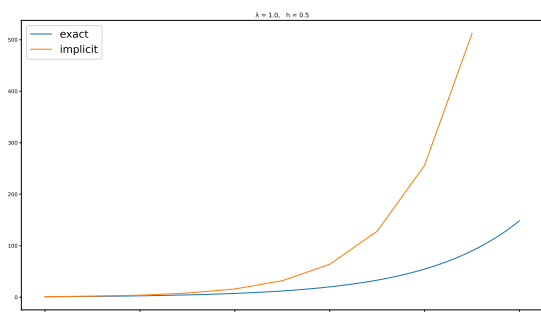
Ist jetzt alles gut? Nein, denn es steht zu vermuten dass wir für positive λ Probleme kriegen. Und in der Tat, für $\lambda = 1$:



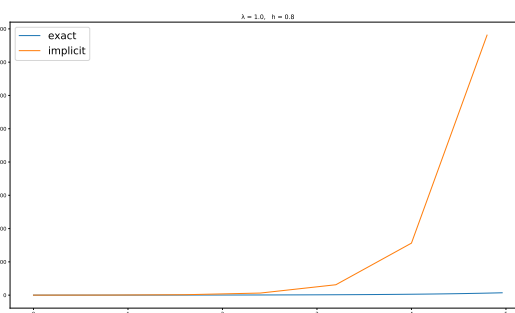
$\lambda = 1, h = 0,05$



$\lambda = 1, h = 0,1$



$\lambda = 1, h = 0,5$



$\lambda = 1, h = 0,8$

Man beachte hier die wechselnde Skalierung der vertikalen Achse. Die Zeitschrittweite im letzten Bild ist 0,8 statt wie bisher 1,0, weil der Wert 1,0 im Verfahren zu einer Division durch Null führt.

11.3 Stabilität von Einschrittverfahren

Das explizite Euler-Verfahren wird für die lineare Gleichung

$$x' = \lambda x, \quad x(t_0) = x_0$$

instabil, wenn $\lambda < 0$ und der Zeitschritt τ zu groß ist.

Wir verallgemeinern das jetzt und betrachten *lineare*, autonome, homogene Systeme

$$x' = Ax, \quad x(0) = x_0 \in \mathbb{R}^d, \quad A \in \mathbb{R}^{d \times d}$$

Satz 11.3. Die Lösung dieses AWP's ist

$$x(t) = \exp(tA)x_0$$

wobei

$$\exp(tA) := \sum_{k=0}^{\infty} \frac{(tA)^k}{k!}$$

Diese Reihe konvergiert gleichmäßig auf jedem kompakten Zeitintervall.

Stabilität heißt: Störungen im Startwert führen auf beschränkte Störungen in der Lösung (für $t \rightarrow \infty$).

- Für lineare Gleichungen $x' = Ax$, $x(0) = x_0 + \delta_0$ ist die Lösung

$$x_\delta(t) = \exp(tA) (x_0 + \delta_0) = \exp(tA)x_0 + \exp(tA)\delta_0$$

d.h. die Störung löst das AWP $x' = Ax$, $x(0) = \delta_0$.

Lemma 11.3 (Deuffhard und Bornemann [3, Lemma 3.20]). *Die Lösung x eines linearen, homogenen AWP ist genau dann stabil, wenn*

$$\sup_{t \geq 0} \|x(t)\| < \infty.$$

Sie ist asymptotisch stabil, falls $\|x(t)\| \xrightarrow{t \rightarrow \infty} 0$.

Beispiel. $x' = \lambda x$ ist stabil für $\lambda \leq 0$, und asymptotisch stabil für $\lambda < 0$.

Satz 11.4 (Deuffhard und Bornemann [3, Satz 3.23]). *Die Lösung des AWP*

$$x' = Ax, \quad x(0) = x_0, \quad A \in \mathbb{C}^{d \times d}$$

ist genau dann stabil, wenn

- *der Realteil aller Eigenwerte nicht positiv ist und*
- *falls λ ein Eigenwert von A mit $\operatorname{Re}(\lambda) = 0$, so hat λ die gleiche algebraische und geometrische Vielfachheit.*

Die Lösung ist asymptotisch stabil, falls $\operatorname{Re}(\lambda) < 0$ für alle Eigenwerte λ von A .

Wunsch: Die von einem numerischen Verfahren erzeugte Folge x_k soll diese Stabilitätseigenschaften *erben*. Beim expliziten Euler-Verfahren:

$$x_{k+1} = \Psi^\tau x_k = x_k + \tau A x_k = (I + \tau A) x_k$$

war das nicht der Fall. Beim impliziten Euler-Verfahren

$$x_{k+1} = x_k + \tau A x_{k+1} \implies x_{k+1} = (I - \tau A)^{-1} x_k$$

jedoch schon!

Verallgemeinere: Betrachte Verfahren der Form

$$x_{k+1} = \Psi^\tau x_k = R(\tau A) x_k$$

mit P, Q Polynomen, sodass

$$R(\tau A) = \frac{P(\tau A)}{Q(\tau A)}$$

x_{k+1} berechnet sich als Lösung des linearen(!) Gleichungssystems

$$Q(\tau A)x_{k+1} = Q(\tau A)(\Psi^\tau x_k) = P(\tau A)x_k$$

Die rationalen Funktionen werden als Approximationen der Evolution $\Phi^\tau = \exp(tA)$ verwendet.

Damit $R(A)$ wohldefiniert ist muss $Q(A)$ invertierbar sein.

- $Q(A)$ darf also nicht den Eigenwert Null haben.

Verallgemeinerung der entsprechenden Bedingung für rationale Funktionen in \mathbb{C} :

Lemma 11.4. *Eine rationale Funktion $r : z \mapsto \frac{p(z)}{q(z)}$ ist genau dort nicht definiert (bzw. hat genau dort Polstellen), wo $q(z) = 0$ ist.*

Satz 11.5 (Deuffhard und Bornemann [3, Satz 3.42]). *Für eine Matrix $A \in \mathbb{C}^{d \times d}$ ist $R(A)$ genau dann definiert, wenn kein Eigenwert von A Pol von R ist.*

11.3.1 Wann sind Einschrittverfahren R stabil?

Satz 11.6 (Deuffhard und Bornemann [3, Satz 3.33]). *Die lineare Iteration $x_{k+1} = Bx_k$ mit $B \in \mathbb{C}^{d \times d}$ ist genau dann stabil, wenn*

- $|\lambda| \leq 1$ für alle Eigenwerte λ von B und
- Falls λ Eigenwert von B mit $|\lambda| = 1$, so hat λ gleiche algebraische und geometrische Vielfachheit.

Die Iteration ist asymptotisch stabil, falls $|\lambda| < 1$ für alle Eigenwerte λ von B .

Wir müssen also die Eigenwerte von

$$B = R(\tau A)$$

betrachten. Kann man sie als Funktion von τ und den Eigenwerte von A berechnen? Kurioserweise gilt:

Satz 11.7 (Deuffhard und Bornemann [3, Satz 3.42, Forts.]). *Sei $\sigma(A)$ das Spektrum von A . Dann ist*

$$\sigma(R(A)) = R(\sigma(A)).$$

Mit anderen Worten: λ ist ein Eigenwert von A genau dann wenn $R(\lambda)$ ein Eigenwert von $R(A)$ ist.

Dabei ist jetzt $R(\lambda)$ die formale rationale Funktion R angewandt auf die komplexe Zahl λ .

Definition. Für ein gegebenes Einschrittverfahren heißt die dazugehörige Funktion $R : \mathbb{C} \rightarrow \mathbb{C}$ Stabilitätsfunktion des Verfahrens.

Vererbung von Stabilität heißt damit: Wenn

$$\operatorname{Re}(\lambda) \leq 0 \quad \text{für alle Eigenwerte } \lambda \text{ von } A$$

dann soll auch

$$|R(\tau\lambda)| \leq 1 \quad \text{für alle Eigenwerte } \lambda \text{ von } A$$

(plus Zusatzbedingungen für den Fall $\operatorname{Re} \lambda = 0$).

Definition. Die Menge

$$S = \{z \in \mathbb{C} \mid |R(z)| \leq 1\}$$

heißt Stabilitätsgebiet von R .

Beispiel. Explizites Euler-Verfahren:

$$x_{k+1} = \underbrace{(I + \tau A)}_{R(\tau A)} x_k \implies R(z) = 1 + z$$

Stabilitätsgebiet: $S = \{z \in \mathbb{C} : |1 + z| \leq 1\}$.

Damit ein Verfahren stabil ist muss $\tau\lambda \in S$ für alle $\lambda \in \sigma(A)$ sein.

Im Falle der skalaren Gleichung mit $\mathbb{R} \ni \lambda < 0$ und dem Euler-Verfahren führt das auf die Bedingung $\tau \leq \frac{2}{|\lambda|}$.

Für eine graphische Darstellung der Stabilitätsgebiete expliziter Runge-Kutta-Verfahren siehe Deuffhard und Bornemann [3, Seite 238].

Folgendes Detail sticht ins Auge.

Lemma 11.5 (Deuffhard und Bornemann [3, Lemma 6.5]). *Für jede konsistente rationale Approximation von \exp gilt $0 \in \partial S$.*

Also:

- Die Lösung $x(t) = \exp(tA)x_0$ ist stabil, wenn alle Eigenwerte von A in der linken Halbebene von \mathbb{C} liegen.
- Ein numerisches Verfahren $\Psi^\tau = R(\tau A)$ ist stabil, wenn alle Eigenwerte von τA in S liegen (plus Zusatzbedingungen am Rand von S).

Folgende Eigenschaft ist deshalb wünschenswert:

Definition. Ein Einschrittverfahren heißt A -stabil, falls sein Stabilitätsgebiet die negative komplexe Halbebene enthält.

In diesem Fall gibt es keine Schrittweitenbeschränkung!

Explizite Verfahren können aber nicht A-stabil sein.

Lemma 11.6. *Die Flüsse aller expliziten Runge-Kutta-Verfahren (für lineare Gleichungen) sind Polynome in τA , also $\Psi^\tau x = P(\tau A)x$.*

Beweis. Wir zeigen dass für jedes $i \leq s$ der Ausdruck τk_i ein formales Polynom in τA ist.

Dann folgt die Behauptung aus

$$\Psi^{t+\tau, t} x = x + \tau \sum_{i=1}^s b_i k_i = (\tau A)^0 x + \sum_{i=1}^s b_i \tau k_i.$$

Beweis mit vollständiger Induktion: RK-Verfahren für $x' = Ax$:

$$k_i = f(t + c_i \tau, x + \tau \sum_{j=1}^{i-1} a_{ij} k_j) = A \left[x + \tau \sum_{j=1}^{i-1} a_{ij} k_j \right]$$

- $\tau k_1 = \tau Ax$ ist Polynom in τA .
- Seien τk_j Polynome für alle $j < i$. Dann ist

$$k_i = Ax + \sum_{j=1}^{i-1} a_{ij} \tau k_j$$

Polynom in τA . □

Lemma 11.7 (Deuffhard und Bornemann [3, Lemma 6.11]). *Das Stabilitätsgebiet von Polynomen ist kompakt.*

Beweis. Für jedes Polynom P vom Grad ≥ 1 gilt $|P(z)| \xrightarrow{z \rightarrow \infty} \infty$. Also ist S beschränkt. □

Implizite Verfahren können A-stabil sein: z.B. Implizites Euler-Verfahren:

$$R(z) = \frac{1}{1-z}$$

Stabilitätsgebiet: $S = \{z \in \mathbb{C} \mid |1-z| \geq 1\} \supset \mathbb{C}$.

Das Ziel für die Zukunft lautet jetzt, A-stabile Verfahren hoher Ordnung zu konstruieren.

11.4 Implizite Runge–Kutta-Verfahren

Wir betrachten jetzt wieder allgemeine nichtlineare, nicht-autonome Anfangswertprobleme

$$x' = f(t, x), \quad x(t_0) = x_0.$$

Wie können wir stabile Verfahren hoher Konsistenzordnung konstruieren?

Definition (Butcher 1964). *Unter einem allgemeinen Runge-Kutta-Verfahren (kurz: RK-Verfahren) verstehen wir ein Verfahren der Form*

$$(i) \quad k_i = f\left(t + c_i\tau, x + \tau \sum_{j=1}^s a_{ij}k_j\right), \quad i = 1, \dots, s,$$

$$(ii) \quad \Psi^{t+\tau, t}x = x + \tau \sum_{j=1}^s b_jk_j.$$

- Unterschied zu expliziten RK-Verfahren: Die Summe in (i) geht über alle s Stufen, nicht nur über die ersten $i - 1$.

Man fasst die Koeffizienten wieder in zwei Vektoren $b, c \in \mathbb{R}^s$ und eine Matrix $\mathcal{A} \in \mathbb{R}^{s \times s}$ zusammen.

Darstellung der Koeffizienten wieder im Butcher-Schema:

$$\begin{array}{c|c} c & \mathcal{A} \\ \hline & b^T \end{array}$$

Beispiel (Implizites Euler-Verfahrens).

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

- Verfahren ist explizit, wenn \mathcal{A} eine strikte untere Dreiecksmatrix ist.
- Ansonsten muss in jedem Schritt ein nichtlineares Gleichungssystem gelöst werden.

Frage: Unter welchen Bedingungen ist dieses Gleichungssystem eindeutig lösbar (für hinreichend kleine τ ?) Nur dann kann ja von einem wohldefinierten Zeitintegrationsverfahren gesprochen werden.

Um diese Frage zu klären schreiben wir das Verfahren zunächst in der sogenannten *symmetrischen Form* auf:

Definiere:

$$g_i = x + \tau \sum_{j=1}^s a_{ij}k_j, \quad i = 1, \dots, s.$$

Dann gilt

$$g_i = x + \tau \sum_{j=1}^s a_{ij} f(t + c_j \tau, g_j) \quad i = 1, \dots, s$$

$$\Psi^{t+\tau, t} x = x + \tau \sum_{j=1}^s b_j f(t + c_j \tau, g_j).$$

Satz 11.8 ([3, Satz 6.28]). Die Abbildung $f \in C(\Omega, \mathbb{R}^d)$ sei auf $\Omega \subset \mathbb{R} \times \mathbb{R}^d$ bezüglich x lokal Lipschitz-stetig. Für ein implizites RK-Verfahren gibt es zu jedem $(x, t) \in \Omega$ ein $\tau_* > 0$, und eindeutige stetige Funktionen

$$g_i : (-\tau_*, \tau_*) \rightarrow \mathbb{R}^d, \quad i = 1, \dots, s,$$

so dass:

1. $g_i(0) = x$ für $i = 1, \dots, s$
2. Für $|\tau| < \tau_*$ genügen die Vektoren $g_i(\tau)$, $i = 1, \dots, s$, den Bestimmungsgleichungen des impliziten RK-Verfahrens.

Für den Beweis brauchen wir einen besonderen, parameterabhängigen Fixpunktsatz. Den Beweis dazu findet man bei Dieudonné [5, Satz 10.1.1].

Satz 11.9. Es seien E_1 und E_2 zwei Banach-Räume. U und V seien offene Kugeln in E_1 (bzw. E_2) jeweils um 0; der Radius von V sei β . Sei F eine stetige Abbildung von $U \times V$ nach E_2 , so dass $\|F(\tau, y_1) - F(\tau, y_2)\| \leq \theta \cdot \|y_1 - y_2\|$ für $\tau \in U$, $y_1, y_2 \in V$, und θ eine Konstante mit $0 \leq \theta < 1$.

Dann, falls $\|F(\tau, 0)\| < \beta(1 - \theta)$ für alle $\tau \in U$, existiert eine eindeutige Abbildung $g : U \rightarrow V$ so dass

$$g(\tau) = F(\tau, g(\tau))$$

für alle $\tau \in U$, und g ist stetig in U .

Beweis von Satz 11.8. • Sei $(t_0, x_0) \in \Omega$ fest gewählt.

- f ist lokal Lipschitz-stetig bezüglich x . D.h. es gibt Parameter $\tau_1, \rho, L > 0$ so dass

$$|f(t, x) - f(t, \bar{x})| < L|x - \bar{x}|$$

für alle $(t, x), (t, \bar{x}) \in (t_0 - \tau_1, t_0 + \tau_1) \times B_\rho(x_0) \subset \Omega$.

- Weiterhin gilt $|f(t, x_0)| < M$ für alle $t \in (t_0 - \tau_1, t_0 + \tau_1)$ (zur Not wird dafür τ_1 verkleinert).
- Wir wählen jetzt ein $0 < \theta < 1$ (Das wird das θ aus dem Satz von Dieudonné).

- Schreibe das RK-System als parameterabhängige Fixpunktgleichung

$$g(\tau) = F(\tau, g(\tau))$$

mit

$$\begin{aligned} g &:= (g_1, \dots, g_s)^T, \\ F(\tau, g) &:= (F_1(\tau, g), \dots, F_s(\tau, g))^T, \\ F_i(\tau, g) &= x_0 + \tau \sum_{j=1}^s a_{ij} f(t_0 + c_j \tau, g_j), \quad i = 1, \dots, s. \end{aligned}$$

- g ist aus $E_2 = \mathbb{R}^{s \cdot d}$. Wähle dort die Norm

$$\|g\| = \max_{1 \leq i \leq s} |g_i|.$$

- Notation: $g_* := (x_0, \dots, x_0) \in \mathbb{R}^{s \cdot d}$.
- Jetzt definieren wir die Kugeln in $E_1 = \mathbb{R}$ und $E_2 = \mathbb{R}^{s \cdot d}$

$$U = (-\tau_*, \tau_*), \quad V = \{g \in \mathbb{R}^{s \cdot d} \mid \|g - g_*\| < \rho_*\}.$$

- Damit ist

$$F : U \times V \rightarrow \mathbb{R}^{s \cdot d}$$

wohldefiniert und stetig. (Welche Eigenschaft von τ_* ist hier relevant?)

- Wir zeigen als nächstes die Lipschitz-Stetigkeit von F im zweiten Argument: Für alle $(\tau, g), (\tau, \bar{g}) \in U \times V$ gilt

$$\begin{aligned} \|F(\tau, g) - F(\tau, \bar{g})\| &= \|(F_1(\tau, g) - F_1(\tau, \bar{g}), \dots, F_s(\tau, g) - F_s(\tau, \bar{g}))^T\| \\ &= \max_{1 \leq i \leq s} |F_i(\tau, g) - F_i(\tau, \bar{g})| \\ &= \max_{1 \leq i \leq s} \left| x_0 + \tau \sum_{j=1}^s a_{ij} f(t_0 + c_j \tau, g_j) - x_0 - \tau \sum_{j=1}^s a_{ij} f(t_0 + c_j \tau, \bar{g}_j) \right| \\ &\leq \tau_* \|\mathcal{A}\|_\infty \max_{1 \leq j \leq s} |f(t_0 + c_j \tau, g_j) - f(t_0 + c_j \tau, \bar{g}_j)|. \end{aligned}$$

Den letzten Schritt genauer ausführen!

- Da f lokal Lipschitz-stetig im zweiten Argument ist, gilt

$$\begin{aligned} \|F(\tau, g) - F(\tau, \bar{g})\| &\leq \tau_* \|\mathcal{A}\|_\infty L \max_{1 \leq i \leq s} |g_j - \bar{g}_j| \\ &= \tau_* \|\mathcal{A}\|_\infty L \|g - \bar{g}\| \\ &\leq \theta \|g - \bar{g}\| \end{aligned}$$

wenn $\tau_* \leq \frac{\theta}{L \|\mathcal{A}\|_\infty}$.

- Ähnlich zeigt man

$$\begin{aligned}
 \|F(\tau, g_*) - g_*\| &= \|F(\tau, g_*) - F(0, g_*)\| \\
 &= \left| x_0 + \tau \sum_{j=1}^s a_{ij} f(t_0 + c_j \tau, x_0) - x_0 - 0 \right| \\
 &= \left| \tau \sum_{j=1}^s a_{ij} f(t_0 + c_j \tau, x_0) \right| \\
 &\leq \tau_* \|\mathcal{A}\|_\infty \max_{1 \leq j \leq s} |f(t_0 + c_j \tau, x_0)| \\
 &< \tau_* \|\mathcal{A}\|_\infty M \\
 &\leq \rho(1 - \theta),
 \end{aligned}$$

$$\text{falls } \tau_* \leq \frac{\rho(1-\theta)}{\|\mathcal{A}\|_\infty M}.$$

Wende jetzt den parameterabhängigen Fixpunktsatz 11.9 an! Er liefert

- Es existieren eindeutige $g(\tau) \in V$ für alle $\tau \in U = (-\tau_*, \tau_*)$, so dass

$$g(\tau) = F(\tau, g(\tau)).$$

- $g(\tau)$ ist stetig, insbesondere ist $g(0) = g_*$. □

Implizite RK-Verfahren sind also für kleine τ wohldefiniert.

Wie sieht es mit Konsistenz und Stabilität aus?

Erinnerung: Konsistenztheorie für explizite RK-Verfahren

- Entwickle Φ und Ψ als Taylorreihen
- Wähle die Koeffizienten b, c, \mathcal{A} so, dass möglichst viele Terme aus der Taylorreihe von Φ reproduziert werden.

Im Prinzip funktioniert das für implizite Verfahren genauso.

- Alles etwas komplizierter: Es müssen mehr Koeffizienten bestimmt werden.
- Alles etwas einfacher: Für eine gegebene Ordnung p erhält man die gleiche Anzahl von Bestimmungsgleichungen wie im expliziten Fall [3, Satz 4.24]. Man hat aber mehr Freiheitsgrade, um diese zu erfüllen.

Satz 11.10. *Unter den Bedingungen des obigen Satzes gilt:*

- Die Evolution Ψ ist genau dann konsistent, wenn

$$\sum_{i=1}^s b_i = 1.$$

- Ist $f \in C^p(\Omega, \mathbb{R}^d)$, so ist auch $\Psi^{t+\tau, t}x$ in τ p -fach differenzierbar.
- Wie für explizite Verfahren zeigt man: Ein implizites Verfahren ist genau dann invariant unter Autonomisierung, wenn es konsistent ist, und

$$c_i = \sum_{j=1}^s a_{ij} \quad \text{für } i = 1, \dots, s.$$

Wie bestimmt man jetzt die Koeffizienten? Eine Technik kommt gleich!
Welche Ordnung kann man maximal erzielen?

- Dafür braucht man die Stabilitätsfunktion.

Lemma 11.8. Die Stabilitätsfunktion R eines s -stufigen RK-Verfahrens (b, \mathcal{A}) ist durch

$$R(z) = 1 + zb^T(I - z\mathcal{A})^{-1}(1, \dots, 1)^T$$

gegeben. Die Funktion R kann in eindeutiger Weise als

$$R(z) = \frac{P(z)}{Q(z)}$$

dargestellt werden, wobei P, Q teilerfremde Polynome höchstens s -ten Grades mit $P(0) = Q(0) = 1$ sind.

Beweis. Übung. □

Erinnerung: Ein s -stufiges explizites RK-Verfahren hat höchstens die Konsistenzordnung $p \leq s$.

Lemma 11.9. Ein s -stufiges implizites RK-Verfahren besitze für alle $f \in C^\infty(\Omega, \mathbb{R}^d)$ die Konsistenzordnung $p \in \mathbb{N}$. Dann gilt $p \leq 2s$.

Beweis. • Betrachte das AWP $x' = x, x(0) = 1$.

- Lösung: $x(\tau) = e^\tau$, RK-Approximation Ψ^τ .
- Das Verfahren ist konsistent mit Konsistenzordnung p , also gilt

$$\Psi^\tau 1 - \Phi^\tau 1 = R(\tau) - e^\tau = \mathcal{O}(\tau^{p+1}).$$

- $R = P/Q$ ist Quotient zweier Polynome mit Ordnung jeweils $\leq s$.
- Es folgt $p \leq \deg P + \deg Q \leq 2s$

Warum?

- Angenommen es gäbe Polynome P, Q mit

$$\deg P \leq k, \quad \deg Q \leq j,$$

und $k + j < p$.

- Das heie $P(z)/Q(z) - e^z = \mathcal{O}(z^{k+j+2})$.
- Multiplikation mit $Q(z)$: $P(z) - Q(z)e^z = \mathcal{O}(z^{k+j+2})$.
- Daraus folgt $P = Q = 0$. Widerspruch!

Beweis davon: bung. [3, Lemma 6.4] **Hier ausarbeiten!**

11.5 Kollokationsverfahren

Wie konstruiert man jetzt konkrete RK-Verfahren?

- Die folgende Idee ist unabhngig von den RK-Verfahren entwickelt worden.
- Dass man dadurch implizite RK-Verfahren erhlt wurde erst in den 1970ern entdeckt.

Betrachte

$$x' = f(t, x)$$

- Seien $(t, x) \in \Omega$ und eine Schrittweite τ gegeben.
- Gesucht: Ein Schritt einer diskreten Evolution $\Psi^{t+\tau, t}x$.

Idee:

- Whle s Sttzstellen im Intervall $(t, t + \tau)$

$$t + c_i\tau, \quad 0 \leq c_1 < c_2 < \dots < c_s \leq 1$$

- Konstruiere ein Polynom $u \in P_s^d$, das
 1. den Anfangswert $u(t) = x$ erfllt und
 2. die Differentialgleichung an den Sttzstellen erfllt

$$u'(t + c_i\tau) = f(t + c_i\tau, u(t + c_i\tau)), \quad i = 1, \dots, s.$$

3. Setze

$$\Psi^{t+\tau, t}x := u(t + \tau).$$

Ein Bild!

Diese Bedingungen nennen wir *Kollokationsbedingungen*.

Einziger Parameter des Verfahrens: Die Sttzstellen c_1, \dots, c_s . Wir haben $s + 1$ Bedingungen an ein Polynom s -ten Grades.

- Wir *vermuten*, dass ein eindeutiges u existiert (zumindest fr kleine τ).
- Klar ist das nicht, denn die Gleichungen fr u sind nichtlinear!

Einfacher Ausweg: Wir interpretieren das Verfahren als implizites RK-Verfahren. Dann liefert Satz 11.8 Existenz und Eindeutigkeit.

- Angenommen es existiere eine Lösung $u \in P_s^d$.
- Sei $\{L_1, \dots, L_s\}$ die Lagrange-Basis von P_{s-1} bezüglich der c_i , also

$$L_i(c_j) = \delta_{ij}, \quad i, j = 1, \dots, s.$$

- u' ist in P_{s-1}^d , und hat Lagrange-Darstellung

$$u'(t + \theta\tau) = \sum_{j=1}^s \underbrace{u'(t + c_j\tau)}_{k_j} L_j(\theta) = \sum_{j=1}^s k_j L_j(\theta). \quad (11.1)$$

- Wir integrieren und nutzen die Kollokationsbedingung 1: $u(t) = x$

$$\begin{aligned} u(t + c_i\tau) &= u(t) + \int_t^{t+c_i\tau} u'(s) ds \\ &= x + \tau \int_0^{c_i} u'(t + \theta\tau) d\theta \\ &= x + \tau \int_0^{c_i} \sum_{j=1}^s k_j L_j(\theta) d\theta \\ &= x + \tau \sum_{j=1}^s k_j \underbrace{\int_0^{c_i} L_j(\theta) d\theta}_{a_{ij}:=} \\ &= x + \tau \sum_{j=1}^s a_{ij} k_j. \end{aligned}$$

- Das setzen wir in die Kollokationsbedingung 2 ein:

$$k_i = f\left(t + c_i\tau, x + \tau \sum_{j=1}^s a_{ij} k_j\right), \quad i = 1, \dots, s.$$

- Abschließend benutzen wir die Kollokationsbedingung 3:

$$\begin{aligned} \Psi^{t+\tau, t} x &= u(t + \tau) \\ &= x + \tau \int_0^1 u'(t + \theta\tau) d\theta \\ &= x + \tau \int_0^1 \sum_{j=1}^s k_j L_j(\theta) d\theta \quad (\text{wegen (11.1)}) \\ &= x + \tau \sum_{j=1}^s b_j k_j \end{aligned}$$

mit

$$b_j = \int_0^1 L_j(\theta) d\theta, \quad j = 1, \dots, s.$$

Wir erhalten tatsächlich ein RK-Verfahren mit

$$\begin{aligned} a_{ij} &= \int_0^{c_i} L_j(\theta) d\theta, & L_i(\theta) &= \frac{\prod_{j \neq i} (c_j - \theta)}{\prod_{j \neq i} (c_j - c_i)} & i, j &= 1, \dots, s \\ b_j &= \int_0^1 L_j(\theta) d\theta, & & & j &= 1, \dots, s \\ k_i &= u'(t + c_i \tau), & i &= 1, \dots, s \end{aligned}$$

Diese Größen hängen nur von den c_i ab.

- Die Stufen k_i sind gerade die Ableitungen von u an den Stützstellen c_i .
- Durch Satz 11.8 bekommen wir die Existenz einer eindeutigen Lösung für das Kollokationsproblem!
- Deutliche Reduktion der Komplexität: Nur noch s Freiheitsgrade c_1, \dots, c_s statt bisher $2s + s^2$ Freiheitsgrade c, b, \mathcal{A} .

Aber sind alle Kollokations-Verfahren auch *gute* RK-Verfahren?

Erinnerung: Satz 4.18 aus Deuffhard und Bornemann [3]: Ein RK-Verfahren besitzt

- genau dann die Konsistenzordnung 1, wenn

$$\sum_{i=1}^s b_i = 1$$

- genau dann die Konsistenzordnung 2, wenn zusätzlich

$$\sum_{i=1}^s b_i c_i = \frac{1}{2}.$$

Lemma 11.10 (Dd & Bo, 6.37). *Die Koeffizienten eines durch Kollokation definierten RK-Verfahrens (b, c, \mathcal{A}) erfüllen*

$$\sum_{j=1}^s b_j c_j^{k-1} = \frac{1}{k} \quad k = 1, \dots, s. \quad (11.2)$$

Insbesondere sind solche Verfahren also konsistent ($k = 1$) und von mindestens zweiter Ordnung wenn $s \geq 2$.

Beweis. • Nach Definition der b_j gilt

$$\sum_{j=1}^s b_j c_j^{k-1} = \sum_{j=1}^s \int_0^1 c_j^{k-1} L_j(\theta) d\theta.$$

- $\sum_{j=1}^s c_j^{k-1} L_j(\theta)$ ist gerade die Lagrange-Darstellung des Polynoms θ^{k-1} . Deshalb

$$\sum_{j=1}^s b_j c_j^{k-1} = \int_0^1 \theta^{k-1} d\theta = \frac{1}{k}. \quad \square$$

Es besteht ein enger Zusammenhang zwischen RK-Verfahren und Quadraturformeln.

Lemma 11.11. *Interpretiere die b_j , $j = 1, \dots, s$ als Gewichte einer Quadraturformel mit Stützstellen c_j . Aus (11.2) folgt, dass diese Quadraturformel für Polynome höchstens $(s-1)$ -ten Grades exakt ist.*

Beweis. Sei $\pi \in P_{s-1}$, $\pi(\theta) = \sum_{j=0}^{s-1} \alpha_j \theta^j$. Dann ist

$$\begin{aligned} \sum_{i=1}^s b_i \pi(c_i) &= \sum_{i=1}^s b_i \sum_{j=1}^s \alpha_{j-1} c_i^{j-1} \\ &= \sum_{j=1}^s \alpha_{j-1} \sum_{i=1}^s b_i c_i^{j-1} = \sum_{j=1}^s \alpha_{j-1} \frac{1}{j} = \sum_{j=1}^s \alpha_{j-1} \int_0^1 \theta^{j-1} d\theta \\ &= \int_0^1 \sum_{j=1}^s \alpha_{j-1} \theta^{j-1} d\theta = \int_0^1 \pi(\theta) d\theta. \quad \square \end{aligned}$$

Wir werden sehen:

- Die Konsistenzordnung eines Kollokationsverfahrens wird im Wesentlichen durch die Eigenschaften dieser Quadraturformel bestimmt.
- Man kann aus bekannten Quadraturformeln Kollokationsverfahren gleicher Konsistenzordnung konstruieren.

Satz 11.11 (Dd & Bo, 6.40). *Ein durch Kollokation erzeugtes implizites RK-Verfahren (b, c, \mathcal{A}) besitzt die Konsistenzordnung p für rechte Seiten $f \in \mathcal{C}^p(\Omega, \mathbb{R}^d)$ genau dann, wenn die durch Stützstellen c und Gewichte b gegebene Quadraturformel die Ordnung p besitzt.*

Beweisskizze. Teil 1) RK-Verfahren hat Ordnung $p \Rightarrow$ Quadraturformel hat Ordnung p

- (b, c, \mathcal{A}) besitze Konsistenzordnung $p \Rightarrow$ Der Fehler bei einem Schritt ist $\mathcal{O}(\tau^{p+1})$.
- Insbesondere können wir Gleichungen der Form $x' = f(t)$ integrieren.
- Lösung davon ist $x(t + \tau) = x(t) + \int_t^{t+\tau} f(s) ds$.

- Runge-Kutta-Verfahren dafür:

$$\Psi^{t+\tau,t}x = x + \tau \sum_{i=1}^s b_i k_i = x + \tau \sum_{i=1}^s b_i f(t + c_i \tau)$$

- Konsistenzfehler

$$\Psi^{t+\tau,t}x - \Phi^{t+\tau,t}x = x + \tau \sum_{i=1}^s b_i f(t + c_i \tau) - x - \int_t^{t+\tau} f(s) ds$$

hat Ordnung p .

- \Rightarrow Klassische Integrale werden mit Ordnung p approximiert.

Teil 2) Quadraturformel hat Ordnung $p \Rightarrow$ RK hat Ordnung p

- (jetzt wieder allgemeine $x' = f(t, x)$)
- Sei τ so klein, dass das Kollokationspolynom $u \in P_s$ existiert.
- Betrachte u als Lösung einer Störung des AWP

$$x'(\bar{t}) = f(\bar{t}, x(\bar{t})), \quad x(t) = x.$$

Dazu wird die rechte Seite gestört!

- Konkret löst u das AWP

$$u'(\bar{t}) = f(\bar{t}, u(\bar{t})) + \underbrace{[u'(\bar{t}) - f(\bar{t}, u(\bar{t}))]}_{=: \delta f(\bar{t})}, \quad u(t) = x.$$

Plan: Schließe aus der Größe von δf auf den Fehler $x(t + \tau) - u(t + \tau)$. Das ist aber gerade der Konsistenzfehler $\Psi^{t+\tau,t}x - \Phi^{t+\tau,t}x$!

Ideen dabei:

- δf verschwindet an den Stützstellen der Quadraturformel.
- Wird auch an den anderen Punkten klein bleiben.

Wir benutzen ein allgemeines Resultat aus der Störungstheorie gewöhnlicher Differentialgleichungen.

Satz 11.12 (Aleksejew, Gröber (Satz 3.4 in Deuffhard und Bornemann [3])). *Es existiert eine beliebig häufig differenzierbare matrixwertige Funktion $M(\bar{t}, \sigma)$, so dass*

$$x(t + \tau) - u(t + \tau) = \int_t^{t+\tau} M(t + \tau, \sigma) \delta f(\sigma) d\sigma$$

- Schätze das Integral mit der Quadraturformel ab

$$x(t + \tau) - u(t + \tau) = \tau \sum_{j=1}^s b_j M(t + \tau, t + c_j \tau) \delta f(t + c_j \tau) + \mathcal{O}(\tau^{p+1})$$

- u ist aber Kollokationspolynom. Deshalb ist

$$\delta f(t + c_j \tau) = u'(t + c_j \tau) - f(t + c_j \tau, u(t + c_j \tau)) = 0 \quad \forall j = 1, \dots, s.$$

Also folgt

$$x(t + \tau) - u(t + \tau) = \mathcal{O}(\tau^{p+1}). \quad \square$$

Bei diesem Argument muss man aber vorsichtig sein:

- Die Konstante in $\mathcal{O}(\tau^{p+1})$ hängt von höheren Ableitungen von $M(t + \tau, s)\delta f(s)$ nach s ab.
- Dieser Ausdruck hängt aber von u ab.
- Und u wiederum hängt von τ ab!

Es geht aber trotzdem alles gut:

(Lemma 6.41 in Dd & Bo, ca. 2 Seiten lang)

Was haben wir gelernt?

- In jedem Kollokationsverfahren steckt eine Quadraturformel der Ordnung $s \leq p \leq 2s$.
- Diese Ordnung wird an des Kollokations-Verfahren vererbt.
- Das betrifft nur den Fehler am Ende eines Zeitschritts, d.h.,

$$x(t + \tau) - u(t + \tau) = \mathcal{O}(\tau^{p+1})$$

- Gleichzeitig hat man in Form von u auch eine Approximation für $x(\sigma)$ für alle σ zwischen t und $t + \tau$.
- Für die gilt: $\max_{t \leq \sigma \leq t + \tau} |x(\sigma) - u(\sigma)| = \mathcal{O}(\tau^{s+1})$ (Deuffhard und Bornemann [3], Lemma 6.41)
- Also schlechter als am Intervallende (da $s \leq p$)
- Diesen Effekt nennt man Superkonvergenz.

11.5.1 Gauß-Verfahren

Wir bauen ein implizites RK-Verfahren hoher Ordnung:

- Wähle eine Quadraturformel mit s Stützstellen, welche für Polynome vom Grad $p - 1$ exakt ist.
- Konstruiere das dazugehörige Kollokationsverfahren.

Wir versuchen gleich das Maximum: $p = 2s$.

Erinnerung: Ist eine Quadraturformel

$$\int_0^1 \phi(t) dt \approx \sum_{i=1}^s b_i \phi(c_i)$$

exakt für Polynome des Grades $2s - 1$, so sind die Stützstellen

$$0 < c_1 < \dots < c_s < 1$$

eindeutig definiert als die Nullstellen des s -ten Legendre-Polynoms P_s .

Definition.

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ P_2(x) &= \frac{1}{2}(3x^2 - 1) \\ P_3(x) &= \frac{1}{2}(5x^3 - 3x) \\ &\vdots \end{aligned}$$

Dies sind die Standarddarstellungen bzgl. des Intervalls $[-1, 1]$. Für unsere Zwecke muss noch auf $[0, 1]$ umtransformiert werden.

Kollokationsverfahren mit diesen Stützstellen werden Gauß-Verfahren genannt.

Aus unserem Satz 11.11 folgt direkt das folgende Resultat zur Konsistenzordnung von Gauß-Verfahren.

Satz 11.13 (Deuffhard und Bornemann [3], Satz 6.43). *Für $f \in \mathcal{C}^{2s}(\Omega, \mathbb{R}^d)$ besitzt das s -stufige Gauß-Verfahren die Konsistenzordnung $p = 2s$.*

Warum 2s, und nicht 2s-1?

Zusätzlich wollen wir aber auch A -Stabilität.

Satz 11.14 (Deuffhard und Bornemann [3], Satz 6.44). *Jedes Gauß-Verfahren ist A -stabil.*

Das beweisen wir auf einem kleinen Umweg.

11.6 Dissipative Differentialgleichungen

Wir müssen noch beweisen, dass Gauß-Verfahren A -stabil sind.

Das machen wir über einen Umweg:

- Wir führen einen neuen, stärkeren Stabilitätsbegriff ein, der direkt auf nichtlineare Gleichungen zielt.
- Dann zeigen wir, dass Gauß-Verfahren sogar in diesem stärkeren Sinne stabil sind.

Bei bisherigen Stabilitätsuntersuchungen haben wir uns auf lineare Differentialgleichungen

$$x' = \lambda x \quad (11.3)$$

beschränkt. Die waren genau dann stabil, wenn $\lambda \leq 0$.

Das kann man anders formulieren: Die Gleichung (11.3) ist stabil, wenn die rechte Seite $f(t, x) = \lambda x$ monoton fallend in x ist.

Das verallgemeinern wir jetzt für allgemeine, autonome Gleichungen

$$x' = f(x) \quad \text{mit } x(t) \in \mathbb{R}^d.$$

Definition. Eine Abbildung $f : \Omega_0 \rightarrow \mathbb{R}^d$ heißt monoton fallend oder dissipativ bzgl. eines Skalarproduktes $\langle \cdot, \cdot \rangle$, wenn für alle $x, \bar{x} \in \Omega_0$

$$\langle f(x) - f(\bar{x}), x - \bar{x} \rangle \leq 0$$

gilt.

Als nächstes definieren wir eine Variante des Begriffs „stabile Differentialgleichung“.

Definition. Ein Phasenfluss $\Phi^t : \Omega_0 \rightarrow \Omega_0$ heißt nichtexpansiv, wenn

$$|\Phi^t x - \Phi^t \bar{x}| \leq |x - \bar{x}|$$

für alle $x, \bar{x} \in \Omega_0$ und alle zulässigen t .

Lemma 11.12. Sei $x' = f(x)$ Differentialgleichung auf Ω_0 mit lokal Lipschitz-stetigem f . Der Phasenfluss Φ ist genau dann nichtexpansiv, wenn f dissipativ ist.

Beweis. • Betrachte die Funktion

$$\chi(t) := |\Phi^t x - \Phi^t \bar{x}|^2 = \langle \Phi^t x - \Phi^t \bar{x}, \Phi^t x - \Phi^t \bar{x} \rangle$$

- Ableiten nach t :

$$\begin{aligned} \chi'(t) &= \langle (\Phi^t x)' - (\Phi^t \bar{x})', \Phi^t x - \Phi^t \bar{x} \rangle + \langle \Phi^t x - \Phi^t \bar{x}, (\Phi^t x)' - (\Phi^t \bar{x})' \rangle \\ &= 2 \langle (\Phi^t x)' - (\Phi^t \bar{x})', \Phi^t x - \Phi^t \bar{x} \rangle \\ &= 2 \langle f(\Phi^t x) - f(\Phi^t \bar{x}), \Phi^t x - \Phi^t \bar{x} \rangle. \end{aligned}$$

I) f ist dissipativ $\Rightarrow \Phi$ nichtexpansiv

- Sei f dissipativ. Dann ist

$$\chi(t) = \chi(0) + \int_0^t 2 \underbrace{\langle f(\Phi^s x) - f(\Phi^s \bar{x}), \Phi^s x - \Phi^s \bar{x} \rangle}_{\leq 0} ds \leq \chi(0).$$

II) „ \Leftarrow “ Sei Φ nichtexpansiv

$\Rightarrow \chi(t) \leq \chi(0)$ für alle hinreichend kleinen t

$\Rightarrow \chi$ ist monoton fallend bei $t = 0$

$\Rightarrow \chi'(0) = 2 \langle f(x) - f(\bar{x}), x - \bar{x} \rangle \leq 0.$ □

Nichtexpansivität ist eine Eigenschaft, die man eventuell vererben möchte.

Definition (Butcher 1975). *Ein Verfahren heißt B-stabil, wenn es für dissipative, hinreichend glatte rechte Seiten einen nichtexpansiven diskreten Phasenfluss erzeugt, also*

$$|\Psi^\tau x - \Psi^\tau \bar{x}| \leq |x - \bar{x}|$$

für alle zulässigen x, \bar{x}, τ .

Dieses Konzept ist stärker als A-Stabilität.

Lemma 11.13 (Deuffhard und Bornemann [3], Satz 6.50). *B-stabile Runge-Kutta-Verfahren sind A-stabil.*

Beweis. Betrachte das komplexe AWP

$$x' = \lambda x, \quad x(0) = 1, \quad \lambda \in \mathbb{C}, \quad \operatorname{Re} \lambda \leq 0.$$

(Bei Systemen steht dieses AWP stellvertretend für einen Eigenwert.) Das AWP ist stabil. Ist die rechte Seite dissipativ?

- Reellifizierung: $x = u + iv, \lambda = \alpha + i\beta$

$$x' = \lambda x \quad \Leftrightarrow \quad \begin{pmatrix} u \\ v \end{pmatrix}' = \underbrace{\begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix}}_{=:A} \begin{pmatrix} u \\ v \end{pmatrix}$$

- Test auf Dissipativität von $f(x) = Ax$:

$$\begin{aligned} \langle Ax - A\bar{x}, x - \bar{x} \rangle &= \langle A\tilde{x}, \tilde{x} \rangle = (\tilde{u}, \tilde{v}) \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix} \begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} \\ &= \alpha(\tilde{u}^2 + \tilde{v}^2) \\ &\leq 0, \quad \text{da } \alpha = \operatorname{Re} \lambda \leq 0, \quad f \text{ ist dissipativ!} \end{aligned}$$

Das Verfahren ist B-stabil. Also erhält man für diese dissipative rechte Seite einen nichtexpansiven diskreten Phasenfluss

$$\begin{aligned} |x - \bar{x}| &\geq |\Psi^\tau x - \Psi^\tau \bar{x}| = \underbrace{|R(\tau A)x - R(\tau A)\bar{x}|}_{(R: \text{Stabilitätsfunktion des Verfahrens})} \\ &= |R(\tau\lambda)(x - \bar{x})| = |R(\tau\lambda)| \cdot |x - \bar{x}| \end{aligned}$$

($|\cdot|$ ist multiplikativ – komplexer Betrag!)

Daraus folgt $|R(\tau\lambda)| \leq 1$ für alle $\tau \geq 0$.

- Für $\tau = 1$ erhält man $|R(\lambda)| \leq 1$, also

$$\lambda \in S := \{z \in \mathbb{C} : |R(\lambda)| \leq 1\}, \text{ das Stabilitätsgebiet.}$$

- λ ist in \mathbb{C}_- beliebig

\Rightarrow Das Verfahren ist A-stabil. □

Statt der A-Stabilität von Gauß-Verfahren zeigen wir:

Satz 11.15 (Dd & Bo 6.51). *Gauß-Verfahren sind B-stabil.*

Beweis. • Die rechte Seite f sei dissipativ und hinreichend glatt.

- Zu zeigen: Der diskrete Phasenfluss eines Gauß-Verfahrens ist nichtexpansiv.
- Wähle $x, \bar{x} \in \Omega_0$.
- Sofern τ klein genug ist, existieren die Kollokationspolynome $u, \bar{u} \in P_s$, mit

$$u(0) = x, \quad u(\tau) = \Psi^\tau x, \quad \bar{u}(0) = \bar{x}, \quad \bar{u}(\tau) = \Psi^\tau \bar{x}.$$

- Betrachte die Differenz

$$q(\theta) = |u(\theta\tau) - \bar{u}(\theta\tau)|^2$$

- Hauptsatz der Integralrechnung:

$$\begin{aligned} |\Psi^\tau x - \Psi^\tau \bar{x}|^2 &= q(1) \\ &= q(0) + \int_0^1 q'(\theta) d\theta \\ &= |x - \bar{x}|^2 + \int_0^1 q'(\theta) d\theta \end{aligned}$$

- Es ist also zu zeigen, dass

$$\int_0^1 q'(\theta) d\theta \leq 0.$$

- Aber $q(\theta) = |u(\theta\tau) - \bar{u}(\theta\tau)|^2$ ist ein Polynom in θ vom Grad höchstens $2s$.

- Also ist q' ein Polynom vom Grad $2s - 1$.
- Dafür ist Gauß-Quadratur exakt:

$$\int_0^1 q'(\theta) d\theta = \sum_{j=1}^s b_j q'(c_j).$$

- Wir zeigen jetzt, dass $q'(c_j) \leq 0 \forall j = 1, \dots, s$.
Da für alle Gauß-Quadraturformeln $b_j \geq 0 \forall j$ gilt, folgt dann die Behauptung.
- Es gilt $q'(\theta) = 2\langle u'(\theta\tau) - \bar{u}'(\theta\tau), u(\theta\tau) - \bar{u}(\theta\tau) \rangle$
- Da u, \bar{u} Kollokationspolynome sind, folgt weiter:

$$q'(c_j) = 2\langle f(u(c_j\tau)) - f(\bar{u}(c_j\tau)), u(c_j\tau) - \bar{u}(c_j\tau) \rangle \forall j = 1, \dots, s$$

- Diese Ausdrücke sind alle ≤ 0 , da f dissipativ ist. □

Achtung: Nicht alle A-stabilen Verfahren sind B-stabil!

Beispiel: Die Stabilitätsfunktion $R(z) = \frac{1+\frac{z}{2}}{1-\frac{z}{2}}$ beschreibt A-stabile Verfahren.

Bild von $\{ |R(z)|=1 \}$

- Diese Stabilitätsfunktion gehört zur impliziten Mittelpunktsregel

$$\Psi^{t+\tau, t} x = \xi, \quad \xi = x + \tau f\left(t + \frac{\tau}{2}, \frac{x + \xi}{2}\right).$$

- Das ist ein Gauß-Verfahren ($s = 1$), also B-stabil.
- Die selbe Stabilitätsfunktion gehört außerdem zur impliziten Trapezregel

$$\Psi^{t+\tau, t} x = \xi, \quad \xi = x + \frac{\tau}{2} [f(t + \tau, \xi) + f(t, x)].$$

Dieses Verfahren ist *nicht* B-stabil!

Beweisskizze.

- Betrachte $x' = f(x)$ (skalar) mit $f(x) = \begin{cases} |x|^3, & x \leq 0 \\ -x^2, & x > 0 \end{cases}$
- f ist C^1 , monoton fallend, also dissipativ.
- $x \equiv 0$ ist Fixpunkt der Gleichung und der impliziten Trapezregel.
- Wenn die impl. Tr.Regel B-stabil sein soll, muss also

$$|\Psi^\tau x - \underbrace{\Psi^\tau 0}_{=0}| \leq |x - 0|$$

für alle $x \in \mathbb{R}$, $\tau > 0$ gelten (Das Verfahren existiert $\forall \tau$.)

- Allerdings erhält man für $x = -2$, $\tau = \frac{36}{7}$ gerade $\Psi^\tau x = 2,5$.

⇒ Widerspruch! □

11.7 Linear-implizite Einschrittverfahren

Wir haben Verfahren konstruiert, die hohe Ordnung haben, und trotzdem A -stabil sind.

- Gauß-Verfahren, es gibt aber noch andere.

Diese Verfahren sind implizit. Zum Berechnen des nächsten Zeitschritts muss ein Gleichungssystem gelöst werden.

- Falls f linear ist, so ist dieses Gleichungssystem linear. Das ist okay.
- Falls f nichtlinear ist, so ist Gleichungssysteme ebenfalls nichtlinear. Das kann ganz schön teuer werden!

Können wir A -stabile Verfahren konstruieren, für die bei jedem Schritt nur ein lineares Gleichungssystem gelöst werden muss, selbst wenn f nichtlinear ist?

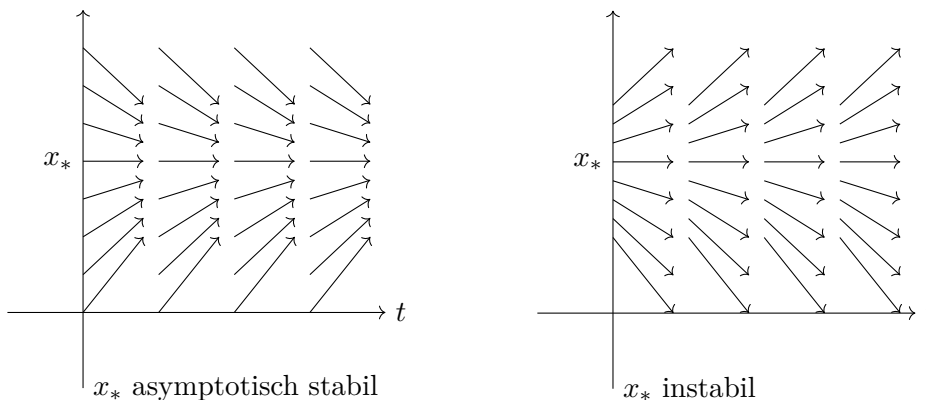
11.7.1 Stabilität von Fixpunkten

Wir wollen einen alternativen Stabilitätsbegriff für autonome nichtlineare Differentialgleichungen $x' = f(x)$ untersuchen.

Definition. Ein Zustand $x_* \in \Omega_0$ heißt Fixpunkt der Gleichung, wenn $f(x_*) = 0$, bzw. wenn $\Phi^t x_* = x_*$ für alle t ist.

Definition. Ein Fixpunkt x_* heißt asymptotisch stabil, wenn ein $\epsilon > 0$ existiert, so dass $\lim_{t \rightarrow \infty} \Phi^t x_0 = x_*$ für alle $x_0 \in \Omega_0$ mit $\|x_* - x_0\| < \epsilon$.

Beispiel:



Man erkennt an den Bildern, dass die asymptotische Stabilität von x_* mit der Ableitung von f in (der Nähe von) x_* zusammenhängt.

Satz 11.16 (Dd & Bo 3.30). Sei $x_* \in \Omega_0$ Fixpunkt von $x' = f(x)$, und f sei stetig differenzierbar. Falls

$$\nu(Df(x_*)) < 0,$$

so ist x_* asymptotisch stabiler Fixpunkt

(Erinnerung: ν ist die Spektralabzisse, der größte Realteil aller Eigenwerte.)

Zwischenfazit: Um die asymptotische Stabilität von Fixpunkten zu untersuchen, reicht es, sich die Linearisierung um x_* anzuschauen!

Wir betrachten jetzt zusätzlich die um x_* linearisierte Differentialgleichung

$$(x - x_*)' = x' = Df(x_*)(x - x_*). \quad (11.4)$$

Idee: Wenn $Df(x_*)$ das Stabilitätsverhalten von x_* qualitativ richtig beschreibt, dann enthält die *lineare* Gleichung (11.4) vielleicht schon alle „schwierigen“ (im Sinne der Stabilität) Aspekte von $x' = f(x)$ in der Nähe von x_* ?

Betrachte ein beliebiges Einschrittverfahren:

- Ψ^τ diskreter Fluss für das Ausgangsproblem
- Ψ_*^τ diskreter Fluss für das linearisierte Problem $x' = Df(x_*)(x - x_*)$.

Definition. Ein Einschrittverfahren heißt invariant gegen Linearisierung um einen Fixpunkt x_* , wenn

1. $\Psi^\tau x_* = x_* \quad \forall \tau > 0$ (τ zulässig)
→ der Fixpunkt der Differentialgleichung ist auch Fixpunkt des numerischen Verfahrens für die nichtlineare Gleichung.
2. $\Psi_*^\tau x = x_* + R(\tau Df(x_*))(x - x_*)$ mit einer rationalen Funktion R , die nur vom Verfahren abhängt
→ D.h.: Für das linearisierte Problem degeneriert das Verfahren zu einer rationalen Approximation der Exponentialfunktion.
3. $D_x \Psi^\tau x|_{x=x_*} = \Psi_*^\tau$ für alle zulässigen τ
→ Ψ_*^τ ist Linearisierung von Ψ^τ .

Zum Beispiel sind alle expliziten RK-Verfahren in diesem Sinne invariant. Solch ein Verfahren heißt *A-stabil*, falls R *A-stabil* ist.

Invariante Verfahren retten den Zusammenhang zwischen der asymptotischen Stabilität eines Fixpunkts x_* und der Linearisierung dort ins Diskrete:

Satz 11.17 (Dd & Bo 6.23). Sei Ψ^τ, Ψ_*^τ ein gegen Linearisierung invariantes Einschrittverfahren. Sei $\tau_c \geq 0$ die maximale Schrittweite, so dass Ψ_*^τ die asymptotische Stabilität vererbt. Dann ist x_* asymptotisch stabiler Fixpunkt der Rekursion

$$x_{n+1} = \Psi^\tau x_n \quad n = 0, 1, 2, \dots$$

für alle $\tau < \tau_c$.

Bsp: Skalare Differentialgleichung

$$x' = \lambda(1 - x^2), \quad \lambda > 0.$$

- Fixpunkte:
 $x_s = 1$ (asymptotisch stabil)
 $x_u = -1$ instabil
- Linearisierte Gleichung in x_s :

$$x' = f'(x_s)(x - x_s) = -2\lambda x_s(x - x_s) = -2\lambda(x - 1)$$

- Explizites Euler-Verfahren dafür
 – stabil, wenn $\tau < 1/\lambda$
- Es folgt: x_s ist auch asymptotisch stabiler Fixpunkt des expl. Euler-Verfahrens für die nichtlineare Gleichung

$$x_{n+1} = x_n + \tau f(x_n) = x_n + \tau\lambda(1 - x_n^2).$$

Aber wie gesagt nur falls $\tau < 1/\lambda$.

Und nicht vergessen: x_s ist nur dann Attraktor, wenn man nah genug dran startet. Für dieses Beispiel heißt das:

- Kontinuierlich: $x_0 > -1$
- Euler: $x_0 \in [0, 5/4]$.

11.7.2 Linear-implizite Runge–Kutta-Verfahren

Idee: Behandle nur den linearen Teil von f implizit.

Für festes $\bar{x} \in \Omega_0$ schreibe die Differentialgleichung als

$$x'(t) = Jx(t) + (f(x(t)) - Jx(t)), \quad J = Df(\bar{x}) \in \mathbb{R}^{d \times d}$$

(Hier ist \bar{x} beliebig; in der Praxis linearisiert man um den Zustand zum vorigen Zeitschritt.) Wende das implizite Euler-Verfahren auf den ersten Term an, und das explizite Euler-Verfahren auf den Rest.

$$\Psi^\tau x = \xi + \tau(f(x) - Jx), \quad \xi = x + \tau J\xi$$

Das ist das linear-implizite oder semi-implizite Euler-Verfahren.

- Nur ein *lineares* Gleichungssystem pro Schritt
- Trotzdem A -stabil

Jetzt allgemein: Linear-implizite Runge-Kutta-Verfahren

$$\Psi^\tau x = x + \tau \sum_{j=1}^s b_j k_j$$

$$k_i = J\left(x + \tau \sum_{j=1}^i \beta_{ij} k_j\right) + \left[f\left(x + \tau \sum_{j=1}^{i-1} \alpha_{ij} k_j\right) - J\left(x + \tau \sum_{j=1}^{i-1} \alpha_{ij} k_j\right) \right]$$

für $i = 1, \dots, s$.

Beachte: Der obere Summationsindex des impliziten Teils ist i , nicht s .

- Dadurch kann der Phasenfluss durch wiederholtes Lösen *linearer* Gleichungssysteme berechnet werden.

1. $J = Df(x)$

2. $(I - \tau \beta_{ii} J) k_i = \tau \sum_{j=1}^{i-1} (\beta_{ij} - \alpha_{ij}) J k_j + f\left(x + \tau \sum_{j=1}^{i-1} \alpha_{ij} k_j\right)$ für $i = 1, \dots, s$

3. $\Psi^\tau x = x + \tau \sum_{j=1}^s b_j k_j$

Solche Verfahren heißen *lineare-implizite RK-Verfahren* oder *Rosenbrock-Verfahren*.

Koeffizienten: $A = (\alpha_{ij}) \in \mathbb{R}^{s \times s}$, $B = (\beta_{ij}) \in \mathbb{R}^{s \times s}$, $b = (b_1 \dots, b_s)$

- Wählt man die β_{ij} alle gleich, so haben die s Gleichungssysteme in 2) alle die gleiche Matrix
 \Rightarrow Es reicht eine LR-Zerlegung, um alle s Gleichungssysteme zu lösen.

Die Frage, ob sich die linearen Gleichungssysteme tatsächlich immer lösen lassen, ist einfacher als für den allgemeinen impliziten Fall:

Lemma 11.14. Sei $\beta \geq 0$ und $J \in \mathbb{R}^{d \times d}$. Die Matrix $I - \tau \beta J$ ist für alle $0 \leq \tau \leq \tau_*$ invertierbar. Dabei hängt τ_* von der Spektralabzisse $\nu(J)$ ab:

$$\tau_* = \infty \text{ für } \nu(J) \leq 0, \quad \tau_* = \frac{1}{\beta \nu(J)} \text{ für } \nu(J) > 0.$$

Beweis. Zu zeigen: Unter den gegebenen Voraussetzungen hat $I - \tau \beta J$ nicht den Eigenwert 0.

- Nach Satz (11.7) über rationale Funktionen ist aber

$$\sigma(I - \tau \beta J) = 1 - \tau \beta \sigma(J).$$

Deshalb zu zeigen: J hat keinen Eigenwert λ mit $1 - \tau\beta\lambda = 0$.

Fall 1: $\nu(J) \leq 0$, d.h. insbesondere $\operatorname{Re}(\lambda) \leq 0$:

$$\operatorname{Re}(1 - \tau\beta\lambda) = 1 - \tau\beta \operatorname{Re}(\lambda) \geq 1 \Rightarrow 1 - \tau\beta\lambda \neq 0.$$

Fall 2: $0 < \operatorname{Re}(\lambda) \leq \nu(J)$:

$$\operatorname{Re}(1 - \tau\beta\lambda) = 1 - \tau\beta \operatorname{Re}(\lambda) \geq 1 - \tau\beta\nu(J).$$

Also > 0 wenn $\tau < \frac{1}{\beta\nu(J)}$. □

Der Satz sagt: Die steifen Anteile der Differentialgleichung (d.h., die nichtpositiven Eigenwert von J), beeinflussen nicht Lösbarkeit des Gleichungssystems.

Für autonome *lineare* Probleme ist das Verfahren offensichtlich äquivalent zum impliziten Runge-Kutta-Verfahren $(b, (\beta_{ij}))$. Es hat also die selbe Stabilitätsfunktion.

Konstruktion der Bedingungsgleichungen: Ähnlich wie bei expl. RK-Verfahren.

11.8 Erhalt erster Integrale

Betrachte die autonome Differentialgleichung

$$x' = f(x)$$

auf einem Phasenraum Ω_0 .

Definition. Eine Funktion $\mathcal{E} : \Omega_0 \rightarrow \mathbb{R}$ heißt erstes Integral, wenn

$$\mathcal{E}(\Phi^t x) = \mathcal{E}(x)$$

für alle $x \in \Omega_0$ und alle zulässigen t gilt.

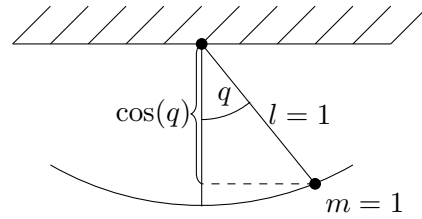
Alternative Bezeichnungen:

- Invariante
- Erhaltungsgröße
- engl.: constant of motion

Beispiel: Mathematisches (Faden-)pendel

Bewegungsgleichungen für Winkel q :

$$\ddot{q} + \frac{g}{l} \sin q = 0$$



bzw. als System erster Ordnung:

$$\begin{aligned}\dot{p} &= -mgl \sin q \\ \dot{q} &= \frac{1}{ml^2} p.\end{aligned}$$

Erhält $\mathcal{E}(p, q) = \frac{1}{2} \frac{1}{ml^2} p^2 - mgl \cos q$ (die totale Energie).

Beispiel: Betrachte ein System mit N Partikeln

- $q_i \in \mathbb{R}^3$, $i = 1, \dots, N$ Positionen, $p_i \in \mathbb{R}$, $i = 1, \dots, N$ Impulse
- m_i : Massen
- Paarweise Interaktion über Kräfte, die vom Abstand abhängen.

Bewegungsgleichungen:

$$q'_i = \frac{p_i}{m_i}, \quad p'_i = \sum_{j=1}^N \nu_{ij}(q_i - q_j)$$

mit

$$\nu_{ij}(y) = -\nu_{ji}(-y)$$

daraus folgt insbesondere dass $\nu_{ii} = 0$.

Die Bewegungsgleichungen erhalten den Gesamtimpuls $P = \sum_{i=1}^N p_i$, denn

$$\frac{d}{dt} \sum_{i=1}^N p_i = \sum_{i=1}^N p'_i = \sum_{i=1}^N \sum_{j=1}^N \nu_{ij}(q_i - q_j) = 0$$

Ebenso: Der Gesamtdrehimpuls $L = \sum_{i=1}^N q_i \times p_i$

$$\begin{aligned}\frac{d}{dt} \sum_{i=1}^N q_i \times p_i &= \sum_{i=1}^N q'_i \times p_i + \sum_{i=1}^N q_i \times p'_i \\ &= \sum_{i=1}^N \frac{1}{m_i} \underbrace{p_i \times p_i}_{=0} + \sum_{i=1}^N \sum_{j=1}^N q_i \times \nu_{ij}(q_i - q_j) \\ &= 0.\end{aligned}$$

Klassifikation der Erhaltungsgrößen: (für diese Beispiele)

- Impuls: linear
- Drehimpuls: quadratisch
- Energie beim Fadenpendel: nichtlinear

Man hätte gerne numerische Verfahren, die erste Integrale erhalten.

Zunächst eine einfache Charakterisierung mit Hilfe von f :

Lemma 11.15 (Dd & Bo 6.56). *Sei f lokal Lipschitz-stetig. Eine Funktion $\mathcal{E} \in C^1(\Omega_0, \mathbb{R})$ ist genau dann erstes Integral, wenn*

$$\nabla \mathcal{E}(x) \cdot f(x) = 0$$

für alle $x \in \Omega_0$.

Beweis. Kettenregel:

$$0 = \frac{d}{dt} \mathcal{E}(\Phi^t x) = \nabla \mathcal{E}(\Phi^t x) \cdot \frac{d}{dt} \Phi^t x = \nabla \mathcal{E}(\Phi^t x) \cdot f(\Phi^t x). \quad \square$$

Beweise damit Energieerhaltung des Fadenpendels!

Satz 11.18 (Hairer, Lubich und Wanner [8, Thm. IV.1.5]). *Alle Runge-Kutta-Verfahren erhalten lineare Invarianten.*

Beweis.

- Sei \mathcal{E} lineare Invariante, also $\mathcal{E}(x) = d^T x$ mit festem Vektor d .
- Nach dem vorigen Satz ist dann $d^T f(x) = 0$ für alle $x \in \Omega_0$.
- Für eine Stufe k_i eines beliebigen RK-Verfahrens ist dann

$$d^T k_i = d^T f\left(x + \tau \sum_{j=1}^s a_{ij} k_j\right) = 0.$$

- Also ist

$$\mathcal{E}(x_{k+1}) = d^T x_{k+1} = d^T \left(x_k + \tau \sum_{i=1}^s b_i k_i\right) = d^T x_k = \mathcal{E}(x_k). \quad \square$$

Für die quadratischen Invarianten betrachten wir zunächst einen wichtigen Spezialfall:

Frage: Für welche linearen autonomen Differentialgleichungen

$$x' = Ax$$

erhält der Phasenfluss Φ^t die Euklidische Norm

$$\|\Phi^t x\|_2 = \|x\|_2 \quad \forall t?$$

\Rightarrow Genau dann, wenn $\Phi^t = \exp(tA)$ eine orthogonale Matrix ist.

Satz 11.19 (Dd & Bo 6.18). Sei $A \in \mathbb{R}^{d \times d}$. Die Matrix $\exp(tA)$ ist genau dann orthogonal, wenn A schiefssymmetrisch ist.

Beweis. Teil I) $\exp(tA) \in O(d) \Rightarrow A = -A^T$

- Sei $\exp(tA) \in O(d)$ für alle t .
- Dann ist

$$I = \exp(tA)^T \exp(tA) = \exp(tA^T) \exp(tA).$$

- Differenziere nach t und betrachte $t = 0$

$$0 = \left(A^T \exp(tA^T) \exp(tA) + \exp(tA^T) A \exp(tA) \right) \Big|_{t=0} = A^T + A$$

Teil II) $A = -A^T \Rightarrow \exp(tA) \in O(d)$

$$\begin{aligned} I &= \exp(tA - tA) = \exp(tA) \cdot \exp(-tA) \quad (\text{da } A \text{ mit } A \text{ kommutiert}) \\ &= \exp(tA) \cdot \exp(tA^T) \\ &= \exp(tA) \cdot \exp(tA)^T. \end{aligned}$$

□

Zentral ist anscheinend die Eigenschaft

$$\exp(z) \cdot \exp(-z) = 1 \quad \forall z \in \mathbb{C}.$$

Das nennt man *Reversibilität*.

Man hätte diese Eigenschaft gerne auch für diskrete Verfahren.

Definition. Eine diskrete Evolution Ψ heißt reversibel, wenn

$$\Psi^{t,t+\tau} \Psi^{t+\tau,t} x = x$$

für alle $(t, x) \in \Omega$ und hinreichend kleine τ .

Beispiel: Das explizite Euler-Verfahren ist nicht reversibel.

Reversible rationale Approximationen der Exponentialfunktion erzeugen normerhaltende diskrete Flüsse.

Satz 11.20 (Db & Bo 6.21). Sei R eine rationale, konsistente, reversible Approximation der Exponentialfunktion. Dann gilt für eine Matrix $A \in \mathbb{R}^{d \times d}$

$$R(\tau A) \in O(d) \quad \forall \tau > 0$$

genau dann, wenn $A = -A^T$.

Beweis. Weitestgehend wie bei Satz 11.19. □

Beispiel:

$$R(z) = \frac{1 + \frac{z}{2}}{1 - \frac{z}{2}} = 1 + z + \frac{z^2}{2} + \frac{z^3}{4} + \mathcal{O}(z^4) = e^z + \mathcal{O}(z^3)$$

- Die entsprechende Matrix-Abbildung heißt *Cayley-Transformation*.
- Stabilitätsfunktion insbesondere der impliziten Mittelpunktsregel
→ dem einfachsten Gauß-Verfahren.

Gauß-Verfahren erhalten sogar beliebige quadratische Invarianten!

Satz 11.21 (Dd & Bo 6.58). Die D.Gl. $x' = f(x)$ mit lokal Lipschitz-stetigem f besitze das quadratische erste Integral \mathcal{E} , d.h.

$$\mathcal{E}(x) = x^T E x + e^T x + \eta$$

mit $E \in \mathbb{R}^{d \times d}$, $e \in \mathbb{R}^d$, $\eta \in \mathbb{R}$. Jedes Gaus-Verfahren erzeugt einen Phasenfluss Ψ , der \mathcal{E} erhält, d.h.

$$\mathcal{E}(\Psi^\tau x) = \mathcal{E}(x)$$

für alle $x \in \Omega_0$ und zulässige τ .

Beweis. Ganz ähnlich wie der Beweis der B -Stabilität.

- Sei $x \in \Omega_0$, und τ so klein, dass das Kollokationspolynom

$$u \in P_s, \quad u(0) = x, \quad u(\tau) = \Psi^\tau x$$

existiert.

- \mathcal{E} ist quadratisch. Deshalb ist

$$q(\theta) := \mathcal{E}(u(\theta\tau))$$

ein Polynom in P_{2s} .

- Hauptsatz der Integralrechnung:

$$\mathcal{E}(\Psi^\tau x) = q(1) = q(0) + \int_0^1 q'(\theta) d\theta = \mathcal{E}(x) + \int_0^1 q'(\theta) d\theta.$$

- Zu zeigen ist also $\int_0^1 q'(\theta) d\theta = 0$.
- Nutze Quadraturformel des Gauß-Verfahrens.
Diese ist für Polynome in P_{2s-1} exakt:

$$\int_0^1 q'(\theta) d\theta = \sum_{j=1}^s b_j q'(c_j).$$

- Es sind aber alle $q'(c_j) = 0$, denn

$$\begin{aligned}q'(c_j) &= (\mathcal{E}(u(c_j\tau)))' \\ &= \tau \nabla \mathcal{E}(u(c_j\tau)) \cdot u'(c_j\tau) \quad (\text{Kettenregel}) \\ &= \tau \nabla \mathcal{E}(u(c_j\tau)) \cdot f(u(c_j\tau)) \quad (\text{Kollokationseigenschaft}) \\ &= 0 \quad (\text{da } \mathcal{E} \text{ eine Invariante ist}).\end{aligned} \quad \square$$

Was ist mit der Energieerhaltung des Fadenpendels?

- Das behandeln wir später.
- Mit der Theorie der Hamiltonschen Systeme.

12 Hamilton-Systeme

12.1 Hamilton-Systeme

Extrem wichtige Klasse von Differentialgleichungen

- klassische Mechanik, Quantenmechanik, relativistische Mechanik
- Dazugehörige spezielle numerische Verfahren
- Schöne Mathematik

“Vereinigendes Prinzip”: Bringt ganz unterschiedliche Gleichungen auf eine gemeinsame Form

Beispiel: Mathematisches Pendel (Fadenpendel)

- Koordinate: Winkel α
- Masse m , Fadenlänge l , Erdbeschleunigung g

Bewegungsgleichungen

$$\ddot{\alpha} + \frac{g}{l} \sin \alpha = 0$$

Beispiel: Teilchen in einem Kraftfeld $F(x)$

$$m\ddot{x} = F(x) \quad (\text{Newtons Gesetz})$$

Beispiel: 1d-Wellengleichung

Longitudinale Auslenkung einer elastischen Schnur

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} &= \frac{\partial^2 u}{\partial t^2} & x \in [a, b], t \geq 0 \\ u(a, t) &= u(b, t) = 0 & \forall t \geq 0 \end{aligned}$$

12.1.1 Die Lagrange-Gleichungen

Mechanisches System mit d Freiheitsgraden $q = (q_1, \dots, q_d)$

- Kinetische Energie

$$T = T(q, \dot{q})$$

(häufig: $T(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q}$ mit $M(q)$ s.p.d.)

- Potentielle Energie

$$U = U(q)$$

Definition. Die Lagrange-Funktion eines mechanischen Systems ist

$$L = T - U.$$

Das mechanische System löst die Lagrange-Gleichungen

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) = \frac{\partial L}{\partial q}$$

Warum? \rightarrow Es gilt das Prinzip der stationären Wirkung.

Definition (Prinzip der stationären Wirkung / Hamiltonsches Prinzip). Sei $q : [t_0, t_1] \rightarrow \mathbb{R}^d$ eine Trajektorie eines mechanischen Systems. Für die in der Natur vorkommenden Trajektorien ist die Wirkung

$$W := \int_{t_0}^{t_1} L(q(t), \dot{q}(t)) dt$$

stationär.

Sei q eine Trajektorie, und δq eine Variation davon, die die Endpunkte fest lässt, also $\delta q(t_0) = \delta q(t_1) = 0$.

- Stationarität von q heißt dann, dass für alle solche δq

$$\frac{d}{d\epsilon} S(q + \epsilon \delta q)|_{\epsilon=0} = 0.$$

- Ausrechnen

$$\begin{aligned} \frac{d}{d\epsilon} \int_{t_0}^{t_1} L(q + \epsilon \delta q, \dot{q} + \epsilon \delta \dot{q}) dt |_{\epsilon=0} &= \int_{t_0}^{t_1} \frac{\partial L}{\partial q} \delta q + \frac{\partial L}{\partial \dot{q}} \delta \dot{q} dt \\ &= \int_{t_0}^{t_1} \left(\frac{\partial L}{\partial q} \delta q - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \delta q \right) dt \quad (\text{partielle Integration}) \\ &= \int_{t_0}^{t_1} \left(\frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \right) \delta q dt. \end{aligned}$$

Da dieser Ausdruck für alle hinreichend glatten Funktionen δq gleich Null sein muss, erhält man die Lagrange-Gleichung

$$\delta W = 0 = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q}.$$

Beispiel: Pendel

- Kinetische Energie

$$T = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) = \frac{1}{2}ml^2\dot{\alpha}^2$$

- Potentielle Energie

$$U = mgy = -mgl \cos \alpha$$

- Lagrange-Funktion

$$L(\alpha, \dot{\alpha}) = \frac{1}{2}ml^2\dot{\alpha}^2 + mgl \cos \alpha$$

- Lagrange-Gleichung

$$0 = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = \frac{d}{dt}(ml^2\dot{\alpha}) + mgl \sin \alpha = ml^2\ddot{\alpha} + mgl \sin \alpha.$$

Beispiel: Teilchen in einem Kraftfeld

Angenommen das Kraftfeld ist *konservativ*, d.h. es gibt ein $U : \mathbb{R}^3 \rightarrow \mathbb{R}$, so dass $F(x) = -\nabla U(x)$.

- Kinetische Energie

$$T(x, \dot{x}) = \frac{1}{2}m\langle \dot{x}, \dot{x} \rangle$$

- Potentielle Energie

$$U$$

- Lagrange-Gleichung

$$0 = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = \frac{d}{dt}(m\dot{x}) + \nabla U(x) = m\ddot{x} - F(x)$$

Beispiel: Wellengleichung

Unendlich-dimensionales System: wird nicht beschrieben durch d Freiheitsgrade (q_1, \dots, q_d) , sondern durch die Funktion $u : [a, b] \rightarrow \mathbb{R}$.

- Kinetische Energie

$$T(u, \dot{u}) = \frac{1}{2} \int_a^b m(x) \dot{u}(x)^2 dx$$

- Potentielle Energie

$$U(u) = \int_a^b m(x) u(x)^2 dx$$

- Lagrange-Funktion

$$L(u, \dot{u}) = T(u, \dot{u}) - U(u)$$

- Lagrange-Gleichung

Fehlt

12.1.2 Die Hamiltonschen Gleichungen

Eine Transformation der Lagrange-Gleichung; quasi „die andere Seite der Medaille“

- Definiere die Impulse

$$p_k := \frac{\partial L}{\partial \dot{q}_k}(q, \dot{q}) \quad \text{für } k = 1, \dots, d$$

Diese Abbildung heißt *Legendre-Transformation*.

Definition. Die *Hamilton-Funktion* ist

$$H(p, q) := p^T \dot{q} - L(q, \dot{q}).$$

Dabei geht man natürlich davon aus, dass die Legendre-Transformation eine C^1 -Bijektion $\dot{q} \leftrightarrow p$ darstellt.

Beispiel: kinetische Energie ist quadratisch:

$$T = \frac{1}{2} \dot{q}^T M \dot{q} \quad \text{mit } M \text{ s.p.d.}$$

- Legendre-Transformation: Für festes q hat man

$$p = M \dot{q}.$$

Transformation ist also tatsächlich glatte Bijektion.

- Hamilton-Funktion

$$\begin{aligned} H(p, q) &= p^T \dot{q} - L(q, \dot{q}) \\ &= p^T M^{-1} p - L(q, M^{-1} p) \\ &= p^T M^{-1} p - T(q, M^{-1} p) + U(q) \\ &= p^T M^{-1} p - \frac{1}{2} (M^{-1} p)^T M (M^{-1} p) + U(q) \\ &= \frac{1}{2} (M^{-1} p)^T M (M^{-1} p) + U(q) \\ &= T + U \end{aligned}$$

Die Hamilton-Funktion ist die Gesamtenergie!

Auch mit Hilfe der Hamilton-Funktion kann man das Verhalten des mechanischen Systems einfach ausdrücken.

Satz 12.1 (Hairer, Lubich und Wanner [8, Thm. VI.1.3]). *Die Lagrange-Gleichung ist äquivalent zu den Hamilton-Gleichungen*

$$\dot{p}_k = -\frac{\partial H}{\partial q_k}(p, q), \quad \dot{q}_k = \frac{\partial H}{\partial p_k}(p, q), \quad k = 1, \dots, d.$$

Beweis. Lagrange \implies Hamilton (die andere Richtung ist ähnlich)

$$\frac{\partial H}{\partial q} = \frac{\partial}{\partial q} (p^T \dot{q} - L(q, \dot{q})) \quad (\text{Def. von } H)$$

$$= p^T \frac{\partial \dot{q}}{\partial q} - \frac{\partial L}{\partial q} - \underbrace{\frac{\partial L}{\partial \dot{q}}}_{=p^T} \frac{\partial \dot{q}}{\partial q} \quad (\text{Kettenregel})$$

$$= -\frac{\partial L}{\partial q} \quad (\text{Def. von } p = \frac{\partial L}{\partial \dot{q}})$$

$$= -\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) \quad (\text{Lagrange-Gleichung})$$

$$= -\dot{p} \quad (\text{Def. von } p)$$

Und:

$$\begin{aligned} \frac{\partial H}{\partial p} &= \frac{\partial}{\partial p} (p^T \dot{q} - L(q, \dot{q})) \\ &= \dot{q} + p^T \frac{\partial \dot{q}}{\partial p} - \underbrace{\frac{\partial L}{\partial \dot{q}}}_{=p^T} \frac{\partial \dot{q}}{\partial p} \quad (\text{Produktregel; } q \text{ hängt nicht von } p \text{ ab}) \\ &= \dot{q} \quad \square \end{aligned}$$

Sowohl die Lagrangesche als auch die Hamiltonsche Formulierungen haben ihre Daseinsberechtigung.

- Die Lagrange-Formulierung ist besonders fundamental: sie beruht auf Variationsprinzipien
- Die Hamilton-Formulierung ist besonders fundamental: sie beruht auf der Gesamtenergie des Systems.

Beispiel: Pendel (mit $q = \alpha$)

- Kinetische Energie

$$T = \frac{1}{2} m l^2 \dot{q}^2$$

- Potentielle Energie

$$U = -mgl \cos q$$

- Impuls

$$p := \frac{\partial L}{\partial \dot{q}} = \frac{\partial}{\partial \dot{q}} \left(\frac{1}{2} m l^2 \dot{q}^2 + mgl \cos q \right) = m l^2 \dot{q}$$

- Kinetische Energie ist quadratisch, also

$$\begin{aligned} H(p, q) &= T(p, q) + U(q) \\ &= \frac{1}{2} m l^2 \dot{q}^2 - mgl \cos q \\ &= \frac{1}{2} \frac{1}{m l^2} p^2 - mgl \cos q \end{aligned}$$

- Bewegungsgleichungen:

$$\begin{aligned}\dot{p} &= -\frac{\partial H}{\partial q} \Leftrightarrow \dot{p} = -mgl \sin q \\ \dot{q} &= \frac{\partial H}{\partial p} \Leftrightarrow \dot{q} = \frac{1}{ml^2} p\end{aligned}$$

In der letzten Vorlesung hatten wir gesehen, dass das Pendel die Größe

$$\frac{1}{2} \frac{1}{ml^2} p^2 - mgl \cos q = T(p, q) + U(p)$$

erhält. Das ist kein Zufall.

Satz 12.2. Die Hamilton-Funktion H ist Invariante des Flusses der Hamiltonschen Gleichung.

Beweis.

$$\begin{aligned}\frac{d}{dt} H(p, q) &= \frac{\partial H}{\partial p} \dot{p} + \frac{\partial H}{\partial q} \dot{q} && \text{(Kettenregel)} \\ &= \frac{\partial H}{\partial p} \left(-\frac{\partial H}{\partial q} \right) + \frac{\partial H}{\partial q} \left(\frac{\partial H}{\partial p} \right) && \text{(Hamiltonsche Gl.)} \\ &= 0 && \square\end{aligned}$$

Diese sehr allgemeine Erhaltungseigenschaft wollen wir natürlich ins Diskrete übertragen!

12.2 Symplektizität

Flüsse von Hamiltonschen Systemen haben eine weitere wichtige Erhaltungseigenschaft,

- die sog. Symplektizität
- ähnlich wie Volumenerhaltung im Phasenraum

Beispiel: Volumenerhaltung beim mathematischen Pendel

Bild

Betrachte die Hamiltonschen Gleichungen

$$\dot{p} = -\frac{\partial H}{\partial q}(p, q), \quad \dot{q} = \frac{\partial H}{\partial p}(p, q)$$

Umschreiben:

$$\begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}^{-1} \begin{pmatrix} \frac{\partial H}{\partial p} \\ \frac{\partial H}{\partial q} \end{pmatrix}$$

Diese Beziehung wollen wir jetzt abstrakter betrachten.

Bemerkung: die harmlos aussehende Matrix $\begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$ hat eine besondere Eigenschaft. Es gilt nämlich

$$\begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}^2 = - \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}.$$

- Verhält sich also wie die imaginäre Einheit i .
- Erzeugt eine komplexe Struktur auf \mathbb{R}^{2d} .

Wir betrachten 2-dimensionale Parallelogramme in \mathbb{R}^{2d} .

- Aufgespannt durch Vektoren

$$\xi = \begin{pmatrix} \xi^p \\ \xi^q \end{pmatrix}, \quad \eta = \begin{pmatrix} \eta^p \\ \eta^q \end{pmatrix}.$$

Falls $d = 1$, so ist die orientierte Fläche des Parallelogramms gerade

$$\det \begin{pmatrix} \xi^p & \eta^p \\ \xi^q & \eta^q \end{pmatrix} = \xi^p \eta^q - \xi^q \eta^p = (\xi^p \quad \xi^q) \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \eta^p \\ \eta^q \end{pmatrix}.$$

Das verallgemeinern wir jetzt für höhere Dimensionen.

Definition (Symplektische Form). Die symplektische Form $\omega : \mathbb{R}^{2d} \times \mathbb{R}^{2d} \rightarrow \mathbb{R}$ ist

$$\omega(\xi, \eta) := \sum_{i=1}^d \det \begin{pmatrix} \xi_i^p & \eta_i^p \\ \xi_i^q & \eta_i^q \end{pmatrix} = \sum_{i=1}^d (\xi_i^p \eta_i^q - \xi_i^q \eta_i^p).$$

- Bilineare Form
- Interpretation: Summe der orientierten Flächen der Projektionen auf die Koordinatenebenen (p_i, q_i) .
- Matrixdarstellung

$$\omega(\xi, \eta) = (\xi^{pT} \quad \xi^{qT}) \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \begin{pmatrix} \eta^p \\ \eta^q \end{pmatrix}.$$

Da die Matrix $\begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$ wichtig zu sein scheint geben wir ihr den Namen J .

Eine wichtige Eigenschaft von Hamiltonschen Systemen ist nun, dass ihre Flüsse

$$\Phi^t : \mathbb{R}^{2d} \rightarrow \mathbb{R}^{2d}$$

die symplektische Form erhalten.
Das muss man natürlich erklären.

Definition (Lineare symplektische Abbildung). Eine lineare Abbildung $A : \mathbb{R}^{2d} \rightarrow \mathbb{R}^{2d}$ heißt symplektisch, wenn

$$\omega(A\xi, A\eta) = \omega(\xi, \eta) \quad \forall \xi, \eta \in \mathbb{R}^{2d}.$$

Alternativ: Wenn

$$A^T J A = J.$$

- Für $d = 1$ bedeutet das gerade, dass A flächenerhaltend ist.

Definition (Differenzierbare symplektische Abbildung). Sei U eine offene Teilmenge von \mathbb{R}^{2d} . Eine differenzierbare Abbildung $g : U \rightarrow \mathbb{R}^{2d}$ heißt symplektisch, falls die Jacobi-Matrix $\nabla g(p, q)$ für alle $(p, q) \in U$ symplektisch ist.

Jetzt der zentrale Satz: die Flüsse Φ^t von Hamiltonschen Systemen erhalten die symplektische Form:

Satz 12.3 (Poincaré, 1899). Sei $H(p, q)$ zweimal stetig differenzierbar auf $U \subset \mathbb{R}^{2d}$. Sei Φ^t der Phasenfluss der Differentialgleichung

$$\dot{y} = J^{-1} \nabla H(y)$$

mit $y = (p, q)$.

Für jedes feste t ist Φ^t eine symplektische Abbildung.

Beweis. Der Beweis erfolgt in zwei Schritten:

1. Φ^0 ist symplektisch.
2. Die „Abweichung von der Symplektizität“ hängt nicht von t ab.

Zu 1):

- Φ^0 ist symplektisch, wenn seine erste Ableitung an jedem Punkt $y_0 = (p_0, q_0)$ symplektisch ist.
- Da $\Phi^0 y_0 = y_0$ gilt

$$\left(\frac{\partial \Phi^0 y_0}{\partial y_0} \right)^T J \left(\frac{\partial \Phi^0 y_0}{\partial y_0} \right) = I^T J I = J.$$

- Also ist Φ^0 symplektisch.

Zu 2):

- Wir müssen die Ableitung $\frac{\partial \Phi^t y_0}{\partial y_0}$ untersuchen.
- Linearisierte Störung der Lösung bei einer Störung des Startwerts.
- Also gerade die Wronski-Matrix Ξ

- Löst die Gleichung

$$\dot{\Xi} = J^{-1} \underbrace{\nabla^2 H(\Phi^t(y_0))}_{\text{Hesse-Matrix von } H} \Xi$$

- Konkret heißt das hier

$$\frac{d}{dt} \frac{\partial \Phi^t}{\partial y_0} = J^{-1} \nabla^2 H(\Phi^t y_0) \frac{\partial \Phi^t}{\partial y_0} \tag{12.1}$$

- Produktregel:

$$\frac{d}{dt} \left[\left(\frac{\partial \Phi^t}{\partial y_0} \right)^T J \left(\frac{\partial \Phi^t}{\partial y_0} \right) \right] = \left(\frac{d}{dt} \frac{\partial \Phi^t}{\partial y_0} \right)^T J \left(\frac{\partial \Phi^t}{\partial y_0} \right) + \left(\frac{\partial \Phi^t}{\partial y_0} \right)^T J \left(\frac{d}{dt} \frac{\partial \Phi^t}{\partial y_0} \right)$$

- Dort wird jetzt (12.1) eingesetzt:

$$\frac{d}{dt} \left[\left(\frac{\partial \Phi^t}{\partial y_0} \right)^T J \left(\frac{\partial \Phi^t}{\partial y_0} \right) \right] = \left(\frac{\partial \Phi^t}{\partial y_0} \right)^T \nabla^2 H(\Phi^t y_0)^T J^{-T} J \left(\frac{\partial \Phi^t}{\partial y_0} \right) + \left(\frac{\partial \Phi^t}{\partial y_0} \right)^T J J^{-1} \nabla^2 H(\Phi^t y_0) \left(\frac{\partial \Phi^t}{\partial y_0} \right)$$

- Aber $J^T = -J$, also $J^{-T} J = -I$, und ∇^H ist symmetrisch.

- Deshalb ist

$$\frac{d}{dt} \left[\left(\frac{\partial \Phi^t}{\partial y_0} \right)^T J \left(\frac{\partial \Phi^t}{\partial y_0} \right) \right] = 0. \quad \square$$

Es gilt sogar die Umkehrung des Satzes: *nur* Hamiltonsche Systeme haben symplektische Flüsse!

Definition (lokal Hamiltonsch). *Eine Differentialgleichung $x' = f(x)$ heißt lokal Hamiltonsch, wenn für jedes $x_0 \in U$ eine Nachbarschaft existiert, in der*

$$f(x) = J^{-1} \nabla H(x)$$

für eine Funktion H .

Satz 12.4 (Hairer, Lubich und Wanner [8, Satz VI.2.6]). *Sei $f : U \rightarrow \mathbb{R}^{2d}$ stetig differenzierbar. Dann ist $x' = f(x)$ genau dann lokal Hamiltonsch, wenn der Fluss $\Phi^t x$ für alle $x \in U$ und alle t hinreichend klein symplektisch ist.*

12.3 Symplektische Verfahren

Wir wollen Verfahren entwickeln, die die Symplektizität von Hamiltonschen Flüssen erben.

Definition. *Ein Einschrittverfahren heißt symplektisch, falls der diskrete Fluss*

$$\Psi^t : \mathbb{R}^{2d} \rightarrow \mathbb{R}^{2d}$$

symplektisch ist, wenn das Verfahren auf ein Hamiltonsches System angewendet wird.

Die einfachsten symplektischen Verfahren sind die symplektischen Euler-Verfahren

$$\begin{aligned} p_{k+1} &= p_k - \tau H_q(p_{k+1}, q_k) \\ q_{k+1} &= q_k + \tau H_p(p_{k+1}, q_k) \end{aligned}$$

und

$$\begin{aligned} p_{k+1} &= p_k - \tau H_q(p_k, q_{k+1}) \\ q_{k+1} &= q_k + \tau H_p(p_k, q_{k+1}). \end{aligned}$$

Satz 12.5 (Hairer, Lubich und Wanner [8, Satz VI.3.3]). *Die symplektischen Euler-Verfahren sind symplektisch.*

Beweis. Beweis für die erste Methode:

- Methode ist symplektisch, wenn

$$\frac{\partial \Psi^\tau y}{\partial y} \in \mathbb{R}^{2d \times 2d}$$

für alle $y = (p, q)$ die symplektisch Form erhält, wenn also

$$\left(\frac{\partial \Psi^\tau y}{\partial y} \right)^T J \left(\frac{\partial \Psi^\tau y}{\partial y} \right) = J. \quad (12.2)$$

- Wir bestimmen die vier Komponenten von $\frac{\partial \Psi^\tau y}{\partial y}$:

1) Erste Gleichung des Verfahrens:

$$p_{k+1} = p_k - \tau H_q(p_{k+1}, q_k)$$

Ableiten nach p_k :

$$\frac{\partial p_{k+1}}{\partial p_k} = I - \tau H_{qp}(p_{k+1}, q_k) \cdot \frac{\partial p_{k+1}}{\partial p_k}$$

\Leftrightarrow

$$\frac{\partial p_{k+1}}{\partial p_k} (I + \tau H_{qp}) = I$$

Ebenso z.B.

$$\frac{\partial p_{k+1}}{\partial q_k} (I + \tau H_{qp}) = -\tau H_{qq}$$

etc.

Zusammen erhält man

$$\begin{pmatrix} I + \tau H_{qp}^T & 0 \\ -\tau H_{pp} & I \end{pmatrix} \begin{pmatrix} \frac{\partial p_{k+1}}{\partial p_k} & \frac{\partial p_{k+1}}{\partial q_k} \\ \frac{\partial q_{k+1}}{\partial p_k} & \frac{\partial q_{k+1}}{\partial q_k} \end{pmatrix} = \begin{pmatrix} I & -\tau H_{qq} \\ 0 & I + \tau H_{qp} \end{pmatrix},$$

also

$$\frac{\partial \Psi^\tau y}{\partial y} = \begin{pmatrix} I + \tau H_{qp}^T & 0 \\ -\tau H_{pp} & I \end{pmatrix}^{-1} \begin{pmatrix} I & -\tau H_{qq} \\ 0 & I + \tau H_{qp} \end{pmatrix}.$$

Damit kann man die Erhaltungseigenschaft (12.2) direkt nachrechnen. \square

Die symplektischen Euler-Verfahren sind *keine* RK-Verfahren.
Stattdessen gehören sie zu den sog. *partitionierten* RK-Verfahren.
Betrachte Differentialgleichungen der Form

$$y' = f(y, z), \quad z' = g(y, z),$$

wobei $y \in \mathbb{R}^{n_1}$ und $z \in \mathbb{R}^{n_2}$

Idee: Nimm für y und z zwei verschiedene RK-Verfahren.

Details bei Hairer, Lubich und Wanner [8, Kapitel II.2]

Es gibt auch ein „einfaches“ Verfahren zweiter Ordnung, das symplektisch ist.

Satz 12.6 (Hairer, Lubich und Wanner [8, Satz VI.3.5]). *Die implizite Mittelpunktsregel*

$$y_{k+1} = y_k + \tau J^{-1} \nabla H \left(\frac{y_{k+1} + y_k}{2} \right)$$

ist symplektisch.

Beweis. Wir leiten wieder ab

$$\begin{aligned} \frac{\partial \Psi^\tau y_k}{\partial y_k} &= \frac{\partial y_{k+1}}{\partial y_k} \\ &= I + \tau J^{-1} \nabla^2 H \left(\frac{y_{k+1} + y_k}{2} \right) \cdot \left(\frac{1}{2} \frac{\partial y_{k+1}}{y_k} + \frac{1}{2} \right). \end{aligned}$$

Umformen ergibt

$$\frac{\partial y_{k+1}}{\partial y_k} = \left(I - \frac{\tau}{2} J^{-1} \nabla^2 H \right)^{-1} \left(I + \frac{\tau}{2} J^{-1} \nabla^2 H \right).$$

Dann kann man direkt nachrechnen dass $\left(\frac{\partial y_{k+1}}{\partial y_k} \right)^T J \frac{\partial y_{k+1}}{\partial y_k} = J$.

Ein paar Details zu dieser Rechnung einfügen!

12.3.1 Symplektische RK-Verfahren

- Relativ neue Verfahren, wurden erst Ende der 1980er Jahre systematisch untersucht.

Wir interessieren uns wieder für die Ableitung

$$\Xi(t) = \frac{\partial \Phi^t y_0}{\partial y_0}.$$

Diese löst bekanntlich eine lineare Differentialgleichung.

Lemma 12.1 (Hairer, Lubich und Wanner [8, Lemma VI.4.1]). *Das folgende Diagramm kommutiert für alle Runge-Kutta-Verfahren und alle partitionierten Runge-Kutta-Verfahren:*

$$\begin{array}{ccc}
 \dot{y} = f(y), y(0) = y_0 & \xrightarrow{\frac{\partial}{\partial y_0}} & \begin{array}{l} \dot{y} = f(y), \quad y(0) = y_0 \\ \dot{\Xi} = f'(y)\Xi, \quad \Xi(0) = I \end{array} \\
 \downarrow \text{RK-Verfahren} & & \downarrow \text{RK-Verfahren} \\
 \{y_k\} & \xrightarrow{\frac{\partial}{\partial y_0}} & \{y_k, \Xi_k\}
 \end{array}$$

Beweisidee: Betrachte exemplarisch das explizite Euler-Verfahren

$$y_{k+1} = y_k + \tau f(y_k).$$

Ableiten nach y_0 ergibt

$$\Xi_{k+1} = \Xi_k + \tau f'(y_k)\Xi_k.$$

Das ist gerade das explizite Euler-Verfahren für die Gleichung

$$\dot{\Xi} = f'(y)\Xi.$$

Startwert passt auch, denn $I = \frac{\partial y_0}{\partial y_0} = \Xi_0.$ □

Idee: Die Symplektizitätsbedingung ist eine quadratische invariante des erweiterten Systems für die Variablen y, Ξ .

Satz 12.7. *Alle Verfahren die quadratische Invarianten erhalten sind symplektisch.*

Beweis. Der quadratische Ausdruck $\Xi^T J \Xi$ is Invariante der Gleichung

$$\dot{\Xi} = J^{-1} \nabla^2 H(y) \Xi,$$

denn

$$\begin{aligned}
 \frac{d}{dt} (\Xi^T J \Xi) &= \dot{\Xi}^T J \Xi + \Xi^T J \dot{\Xi} \\
 &= (J^{-1} \nabla^2 H \Xi)^T J \Xi + \Xi^T J J^{-1} \nabla^2 H \Xi \\
 &= \Xi^T \nabla^2 H \underbrace{J^{-T} J}_{=-I} + \Xi^T \underbrace{J J^{-1}}_{=I} \nabla^2 H \Xi \\
 &= 0.
 \end{aligned}$$
□

Korollar. Gauß-Verfahren sind symplektisch.

12.3.2 Reversibilität vs. Symplektizität

Es gibt reversible Verfahren, die nicht symplektisch sind.

Es gibt symplektische Verfahren, die nicht reversibel sind.

Für quadratische Hamilton-Funktionen ist das anders.

Satz 12.8 (Hairer, Lubich und Wanner [8, Satz VI.4.9.]). *Für RK-Verfahren sind die folgenden Aussagen äquivalent:*

i) Die Methode ist reversibel für lineare Probleme

$$\dot{y} = Ly$$

ii) Die Methode ist symplektisch für Hamilton-Gleichungen mit quadratischer Hamilton-Funktion

$$H(y) = \frac{1}{2}y^T Cy. \quad C \text{ s.p.d.}$$

Beweis. ii) \rightarrow i)

- Die Hamilton-Gleichungen haben die Form

$$\dot{y} = J^{-1}\nabla H(y) = J^{-1}Cy,$$

sind also linear.

- Das Runge-Kutta-Verfahren dafür hat also die Form

$$\Psi^\tau y = R(\tau J^{-1}C)y$$

wobei R die Stabilitätsfunktion ist.

- Da das Verfahren symplektisch ist, gilt

$$R(\tau J^{-1}C)^T J R(\tau J^{-1}C) = J.$$

- Da $R = PQ^{-1}$ für Polynome P, Q erhält man

$$P(\tau J^{-1}C)^T J P(\tau J^{-1}C) = Q(\tau J^{-1}C)^T J Q(\tau J^{-1}C). \quad (12.3)$$

Die folgende Rechnung muss nochmal im Detail überprüft werden!

- Betrachte das Produkt „Polynom in $J^{-1}C$ “ mit J .

- Für jedes Monom $(J^{-1}C)^k$ gilt (C ist symmetrisch)

$$((J^{-1}C)^k)^T J = (C^T J^{-T})^k J = J(-(J^{-1}C)^k) \quad \forall k = 0, 1, 2, \dots$$

- Also folgt aus (12.3)

$$P(-\tau J^{-1}C) \cdot P(\tau J^{-1}C) = Q(-\tau J^{-1}C) \cdot Q(\tau J^{-1}C)$$

bzw.

$$R(-\tau J^{-1}C) \cdot R(\tau J^{-1}C) = I.$$

- Das ist gerade die Symmetrie des Verfahrens. □

12.4 Energieerhaltung

- Wir haben einige Mühe in Verständnis und Erhaltung der Symplektizität gesteckt.
- Aber Symplektizität ist eine sehr abstrakte Eigenschaft. Wozu soll die gut sein?
- Hier komme eine etwas konkretere Rechtfertigung.

Betrachte das mathematische Pendel.

- Kinetische Energie:

$$T(q, \dot{q}) = \frac{ml^2}{2} \dot{q}^2$$

(q ist der Winkel)

- Potentielle Energie:

$$U(q) = -mgl \cos q$$

- Bewegungsgleichungen:

$$\ddot{q} + \frac{g}{l} \sin q = 0$$

- Gesamtenergie:

$$E = \frac{ml^2}{2} \dot{q}^2 - mgl \cos q$$

Dies entspricht der Hamilton-Funktion

$$H(p, q) = \frac{1}{2ml^2} p^2 - mgl \cos q$$

Damit ist die Gesamtenergie eine Erhaltungsgröße!

- Aber: weder linear noch quadratisch. Wird daher nicht automatisch von z.B. Gauß-Verfahren erhalten.

Wird die Energie von symplektischen Verfahren erhalten?

Nein! Aber fast...

Satz 12.9 (Benettin und Giorgilli [1]; Hairer, Lubich und Wanner [8, Thm. IX.8.1]). *Betrachte ein Hamilton-System mit analytischer Hamilton-Funktion $H : D \rightarrow \mathbb{R}$, ($D \subset \mathbb{R}^{2d}$), und wende ein symplektisches Verfahren Ψ^τ mit Schrittweite τ an. Wenn die numerische Lösung in einer kompakten Menge $K \subset D$ bleibt, dann existiert ein τ_0 , so dass*

$$H(y_n) = H(y_0) + O(\tau^p)$$

für exponentiell lange Zeitintervalle $n\tau \leq e^{\frac{\tau_0}{2\tau}}$.

Symplektische Verfahren erhalten also *nicht* die Hamilton-Funktion bzw. die Gesamtenergie. Aber die numerische Energie bleibt „in der Nähe“ der exakten Energie!

12.5 Variationelle Integratoren

Mit dem jetzt Gelernten können wir Zeitschrittverfahren auf eine ganz neue Art konstruieren.

Siehe Marsden und West [13] für eine detailliertere Übersicht.

Wir erinnern an das Prinzip der stationären Wirkung (auch Hamiltonsches Prinzip genannt)

- Lagrange-Funktion

$$L(q, \dot{q}) = T(q, \dot{q}) - U(q)$$

Definition. Die Wirkung einer Trajektorie $q: t \mapsto (q(t), \dot{q}(t))$ ist

$$S(q) := \int_{t_0}^{t_1} L(q(t), \dot{q}(t)) dt.$$

Wir betrachten nur Trajektorien mit gegebenem festen Start- und Endpunkt

$$q(t_0) = q_0, \quad q(t_1) = q_1.$$

Definition (Hamiltonsches Prinzip). Die tatsächlich vorkommenden Trajektorien sind die, die die Wirkung stationär machen.

Sei q eine Trajektorie, und δq eine Variation davon, die die Endpunkte fest lässt, also $\delta q(t_0) = \delta q(t_1) = 0$.

- Stationarität von q heißt dann, dass für alle solche δq

$$\left. \frac{d}{d\epsilon} S(q + \epsilon \delta q) \right|_{\epsilon=0} = 0.$$

Wie schon in Kapitel 12.1.1 gezeigt ist dies äquivalent zur Euler–Lagrange-Gleichung

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} = \frac{\partial L}{\partial q}.$$

Wir betrachten jetzt das Wirkungsintegral S als Funktion der Start- und Endposition

$$S(q_0, q_1) = \int_{t_0}^{t_1} L(q(t), \dot{q}(t)) dt.$$

Dabei ist q die zu q_0, q_1 gehörige Lösung der Lagrange-Gleichung.

Exkurs Anfang: Erzeugendenfunktionen

Wir brauchen ein weiteres Kriterium für Symplektizität:

- Betrachte ein gegebenes Hamiltonsches System H auf einem festen Zeitintervall $[t_0, t_1]$
- Seien $p_0 \in \mathbb{R}^d$ und $q_0 \in \mathbb{R}^d$ die Startwerte zur Zeit t_0
- Bezeichne die Werte zur Zeit t_1 mit $p_1 \in \mathbb{R}^d$ und $q_1 \in \mathbb{R}^d$
- Es gibt eine Abbildung $\Phi^{t_0, t_1}(p_0, q_0) = (p_1, q_1)$.

Wie wir wissen, ist diese symplektisch.

[Achtung: der folgende Satz enthält überdurchschnittlich viel didaktische Reduktion]

Satz 12.10 (Hairer, Lubich und Wanner [8, Satz VI.5.1]). *Eine Abbildung $\varphi: (p_0, q_0) \mapsto (p_1, q_1)$ ist genau dann symplektisch, wenn lokal eine Funktion*

$$S: (q_0, q_1) \mapsto S(q_0, q_1) \in \mathbb{R}$$

existiert, so dass

$$\nabla S = \begin{pmatrix} \frac{\partial S}{\partial q_0} \\ \frac{\partial S}{\partial q_1} \end{pmatrix} = \begin{pmatrix} -p_0 \\ p_1 \end{pmatrix}. \quad (12.4)$$

- Wenn man eine symplektische Abbildung $(p_0, q_0) \mapsto (p_1, q_1)$ hat, dann sie durch (12.4) aus der Funktion S rekonstruiert werden.
- Aber der obige Satz ist „genau dann, wenn“. Es gilt also auch die Umkehrung:
 - Jede hinreichen glatte (und in einem gewissen Sinne nicht degenerierte) Funktion S erzeugt via (12.4) eine symplektische Abbildung $(p_0, q_0) \mapsto (p_1, q_1)$!
- Man kann also auf systematische Art symplektische Abbildungen erzeugen. Die Funktion S heißt deshalb Erzeugendenfunktion.

Exkurs Ende

Große Überraschung: Diese Funktion S ist gerade die Erzeugendenfunktion einer symplektischen Abbildung!

- Berechne partielle Ableitung:

$$\frac{\partial S}{\partial q_0} = \int_{t_0}^{t_1} \left(\frac{\partial L}{\partial q} \frac{\partial q}{\partial q_0} + \frac{\partial L}{\partial \dot{q}} \frac{\partial \dot{q}}{\partial q_0} \right) dt$$

Partielle Integration des zweiten Terms in der Klammer liefert

$$\begin{aligned}
 &= \left. \frac{\partial L}{\partial \dot{q}} \frac{\partial q}{\partial q_0} \right|_{t_0}^{t_1} + \int_{t_0}^{t_1} \underbrace{\left(\frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \right)}_{=0} \frac{\partial q}{\partial q_0} dt \\
 &= \frac{\partial L(q_1, \dot{q}_1)}{\partial \dot{q}} \cdot \underbrace{\frac{\partial q_1}{\partial q_0}}_{=0} - \frac{\partial L(q_0, \dot{q}_0)}{\partial \dot{q}} \cdot \underbrace{\frac{\partial q_0}{\partial q_0}}_{=1} \\
 &= - \frac{\partial L(q_0, \dot{q}_0)}{\partial \dot{q}} \\
 &= -p_0 \quad (\text{Def. des Impulses}).
 \end{aligned}$$

Ebenso berechnet man

$$\frac{\partial S}{\partial q_1} = p_1.$$

Wir erhalten also

$$\nabla S = \begin{pmatrix} \frac{\partial S}{\partial q_0} \\ \frac{\partial S}{\partial q_1} \end{pmatrix} = \begin{pmatrix} -p_0 \\ p_1 \end{pmatrix}. \quad (12.5)$$

Dies ist gerade die Formel (12.4) für Erzeugendenfunktionen von symplektischen Abbildungen.

Daraus folgt dass die entsprechende Abbildung $(p_0, q_0) \mapsto (p_1, q_1)$ symplektisch ist.

12.5.1 Idee der variationellen Integratoren

Wir ersetzen das Integral im Hamiltonschen Prinzip durch eine diskrete Approximation:

- Führe ein Zeitgitter ein

$$t_0 < t_1 < \dots < t_N = T.$$

- Führe die approximative Wirkung ein

$$L_h(q_k, q_{k+1}) \approx \int_{t_k}^{t_{k+1}} L(q(t), \dot{q}(t)) dt$$

(z.B. durch eine Quadraturformel)

Hier könnte man denken dass L_h ein schlechtes Symbol ist, weil es sich ja schließlich um eine Wirkung handelt. Andererseits fungiert L_h später bei der Definition der diskreten Impulse wie eine Lagrange-Funktion (siehe (12.6)).

q ist hier die Lösung der Lagrange-Gleichung auf $[t_k, t_{k+1}]$ mit gegebenen Start- und Endwerten q_k, q_{k+1} .

- Definiere das diskrete Wirkungsfunktional

$$S_h(\{q_k\}_{k=0}^N) := \sum_{k=0}^{N-1} L_h(q_k, q_{k+1}).$$

Definition (Diskretes Hamilton-Prinzip). Finde $\{q_k\}_{k=0}^N$ mit gegebenen q_0, q_N , so dass S_h stationär wird.

Wie kann man L_h wählen?

Beispiel: (MacKay [12], 1992)

- Approximiere q auf $[t_k, t_{k+1}]$ als linear Interpolierende von q_k und q_{k+1} .
- Approximiere das Integral durch die Trapezregel

$$L_h(q_k, q_{k+1}) = \tau \cdot \frac{1}{2} \left[L\left(q_k, \frac{q_{k+1} - q_k}{\tau}\right) + L\left(q_{k+1}, \frac{q_{k+1} - q_k}{\tau}\right) \right].$$

Beispiel: (Wendlandt und Marsden [16], 1997)

- Nimm statt der Trapezregel die Mittelpunktsregel

$$L_h(q_k, q_{k+1}) = \tau \left[L\left(\frac{q_{k+1} + q_k}{2}, \frac{q_{k+1} - q_k}{\tau}\right) \right].$$

Wie kommen wir an stationäre Punkte von S_h ?

- Ableitung ausrechnen und gleich Null setzen!
Partielle Ableitung für $k = 1, \dots, N - 1$:

$$\frac{\partial S_h}{\partial q_k} = \frac{\partial}{\partial q_k} \sum_{i=0}^{N-1} L_h(q_i, q_{i+1}) = \frac{\partial}{\partial q_k} L_h(q_{k-1}, q_k) + \frac{\partial}{\partial q_k} L_h(q_k, q_{k+1}).$$

- Dieser Ausdruck = 0 sind die *diskreten Euler-Lagrange-Gleichungen*.
- Ein System von algebraischen Gleichungen (mit Bandstruktur).

Fragen:

- Wann kriege ich mit diesem Ansatz symplektische Integratoren?
- Kann ich das Verfahren so umschreiben dass ich wieder einen Zeitschritt nach dem anderen berechnen kann?

12.5.2 Variationelle Integratoren sind symplektisch

Wir schreiben die diskrete Wirkung jetzt wieder als Funktion von Anfangs- und Endzustand

$$S_h(q_0, q_N) = \sum_{k=0}^{N-1} L_h(q_k, q_{k+1}).$$

Dabei ist $\{q_k\}$ die dazugehörige Lösung des variationellen Integrators.

- Wir rechnen wieder die partiellen Ableitungen aus. Es bezeichne $\frac{\partial L_h}{\partial x}$, $\frac{\partial L_h}{\partial y}$ die partiellen Ableitungen von L_h nach dem ersten bzw. zweiten Argument:

$$\begin{aligned} \frac{\partial S_h}{\partial q_0} &= \sum_{k=0}^{N-1} \left[\frac{\partial L_h}{\partial x} \cdot \frac{\partial q_k}{\partial q_0} + \frac{\partial L_h}{\partial y} \cdot \frac{\partial q_{k+1}}{\partial q_0} \right] \\ &= \frac{\partial L}{\partial x}(q_0, q_1) \cdot \underbrace{\frac{\partial q_0}{\partial q_0}}_{=1} \\ &\quad + \sum_{k=1}^{N-1} \underbrace{\left[\frac{\partial L_h}{\partial y}(q_{k-1}, q_k) \cdot \frac{\partial q_k}{\partial q_0} + \frac{\partial L_h}{\partial x}(q_k, q_{k+1}) \cdot \frac{\partial q_k}{\partial q_0} \right]}_{=0, \text{ wg. diskreter Lagrange-Gleichung}} \\ &\quad + \frac{\partial L_h}{\partial y}(q_{N-1}, q_N) \cdot \underbrace{\frac{\partial q_N}{\partial q_0}}_{=0} \\ &= \frac{\partial L}{\partial x}(q_0, q_1). \end{aligned}$$

- Ebenso

$$\frac{\partial S_h}{\partial q_N} = \frac{\partial L_h}{\partial y}(q_{N-1}, q_N).$$

Jetzt führen wir die *diskreten Impulse* durch eine *diskrete Legendre-Transformation* ein:

$$p_k := -\frac{\partial L_h}{\partial x}(q_k, q_{k+1}) = \frac{\partial L_h}{\partial y}(q_{k-1}, q_k) \quad (12.6)$$

Die Gleichheit ist gerade die diskrete Euler-Lagrange-Gleichung.
[Vergleiche: $p = \frac{\partial L}{\partial \dot{q}}(q, \dot{q})$]

- Für $k = N$ erhalten wir

$$p_N = \frac{\partial L_h}{\partial y}(q_{N-1}, q_N).$$

Zusammen also:

$$\nabla S_h = \begin{pmatrix} \frac{\partial S_h}{\partial q_0} \\ \frac{\partial S_h}{\partial q_N} \end{pmatrix} = \begin{pmatrix} \frac{\partial L_h}{\partial x}(q_0, q_1) \\ \frac{\partial L_h}{\partial y}(q_{N-1}, q_N) \end{pmatrix} = \begin{pmatrix} -p_0 \\ p_N \end{pmatrix}$$

Nach Satz 12.10 ist S_h also eine Erzeugendenfunktion für die symplektische Abbildung

$$(p_0, q_0) \mapsto (p_N, q_N).$$

12.5.3 Variationelle Integratoren als klassische Einschrittverfahren

Jetzt bauen wir uns ein klassisches Zeitschrittverfahren:

Angenommen die diskrete Legendre-Transformation (12.6) sei eine Bijektion zwischen p_k und q_{k+1} .

Einschrittverfahren:

$$\begin{array}{c}
 (p_k, q_k) \\
 \downarrow \text{inv. disk. Legendre-Transformation} \\
 (q_k, q_{k+1}) \\
 \downarrow \text{diskrete Euler-Lagrange-Gleichung} \\
 (q_{k+1}, q_{k+2}) \\
 \downarrow \text{diskrete Legendre-Transformation} \\
 (p_{k+1}, q_{k+1})
 \end{array}$$

Schritt 2 und 3 schreibt man als

$$p_{k+1} = \frac{\partial L_h}{\partial y}(q_k, q_{k+1}).$$

Satz 12.11. *Das diskrete Hamilton Prinzip erzeugt das Zeitschrittverfahren*

$$(p_k, q_k) \mapsto (p_{k+1}, q_{k+1}), \quad p_k = -\frac{\partial L_h}{\partial x}(q_k, q_{k+1}), \quad p_{k+1} = \frac{\partial L_h}{\partial y}(q_k, q_{k+1}).$$

(Die erste Gleichung ist dabei als implizite Gleichung für q_{k+1} zu verstehen.)

- i) Dieses Verfahren ist symplektisch.
- ii) Jedes symplektische Verfahren lässt sich auf diese Art darstellen.

Beweis.

- i) L_h ist Erzeugendenfunktion für die symplektische Abbildung $(p_k, q_k) \mapsto (p_{k+1}, q_{k+1})$

- ii) Jede symplektische Abbildung hat eine Erzeugendenfunktion. Wähle diese als diskrete Lagrange-Funktion. \square

Beispiel: Das Verfahren von MacKay [12]:

$$L_h(q_k, q_{k+1}) = \frac{\tau}{2} L(q_k, v_{k+\frac{1}{2}}) + \frac{\tau}{2} L(q_{k+1}, v_{k+\frac{1}{2}})$$

mit $v_{k+\frac{1}{2}} := \frac{1}{\tau}(q_{k+1} - q_k)$.

Man erhält das Verfahren:

$$\begin{aligned} p_k &= -\frac{\partial L_h}{\partial x}(q_k, q_{k+1}) \\ &= -\frac{\tau}{2} \left[\frac{\partial L}{\partial q}(q_k, v_{k+\frac{1}{2}}) + \frac{\partial L}{\partial \dot{q}}(q_k, v_{k+\frac{1}{2}}) \cdot \left(-\frac{1}{\tau}\right) \right. \\ &\quad \left. + \frac{\partial L}{\partial q}(q_{k+1}, v_{k+\frac{1}{2}}) \cdot \underbrace{\frac{\partial q_{k+1}}{\partial q_n}}_{=0} + \frac{\partial L}{\partial \dot{q}}(q_{k+1}, v_{k+\frac{1}{2}}) \cdot \left(-\frac{1}{\tau}\right) \right] \\ &= -\frac{\tau}{2} \frac{\partial L}{\partial q}(q_k, v_{k+\frac{1}{2}}) + \frac{1}{2} \frac{\partial L}{\partial \dot{q}}(q_k, v_{k+\frac{1}{2}}) + \frac{1}{2} \frac{\partial L}{\partial \dot{q}}(q_{k+1}, v_{k+\frac{1}{2}}), \end{aligned}$$

sowie

$$\begin{aligned} p_{k+1} &= \frac{\partial L_h}{\partial y}(q_k, q_{k+1}) \\ &= \frac{\tau}{2} \frac{\partial L}{\partial q}(q_{k+1}, v_{k+\frac{1}{2}}) + \frac{1}{2} \frac{\partial L}{\partial \dot{q}}(q_k, v_{k+\frac{1}{2}}) + \frac{1}{2} \frac{\partial L}{\partial \dot{q}}(q_{k+1}, v_{k+\frac{1}{2}}). \end{aligned}$$

Wir betrachten das mechanische System

$$L(q, \dot{q}) = \frac{1}{2} \dot{q}^T M \dot{q} - U(q) \quad \text{mit } M \text{ s.p.d}$$

Damit ist

$$\begin{aligned} \frac{\partial L}{\partial \dot{q}}(q_k, v_{k+\frac{1}{2}}) &= M v_{k+\frac{1}{2}} = \frac{\partial L}{\partial \dot{q}}(q_{k+1}, v_{k+\frac{1}{2}}) \\ \frac{\partial L}{\partial q}(q_k, v_{k+\frac{1}{2}}) &= -\nabla U(q_k) = F(q_k) \quad (\text{Kraftfeld an der Stelle } q_k) \end{aligned}$$

Das Verfahren wird also zu:

$$\begin{aligned} p_k &= -\frac{\tau}{2} F(q_k) + \frac{1}{2} M v_{k+\frac{1}{2}} + \frac{1}{2} M v_{k+\frac{1}{2}} \\ p_{k+1} &= \frac{\tau}{2} F(q_{k+1}) + \frac{1}{2} M v_{k+\frac{1}{2}} + \frac{1}{2} M v_{k+\frac{1}{2}} \end{aligned}$$

Umschreiben:

$$Mv_{k+\frac{1}{2}} = p_k + \frac{\tau}{2}F(q_k) \quad (\text{erste Gleichung})$$

$$q_{k+1} = q_k + \tau v_{k+\frac{1}{2}} \quad (\text{Definition von } v_{k+\frac{1}{2}})$$

$$p_{k+1} = Mv_{k+\frac{1}{2}} + \frac{\tau}{2}F(q_{k+1}) \quad (\text{zweite Gleichung})$$

Das ist gerade das Störmer-Verlet-Verfahren!

Diskrete Lagrange-Gleichung in diesem Fall:

$$0 = \frac{\partial L_h}{\partial y}(q_{k-1}, q_k) + \frac{\partial L_h}{\partial x}(q_k, q_{k+1})$$

$$\Leftrightarrow M \underbrace{\frac{(q_{k+1} - 2q_k + q_{k-1}))}{\tau^2}}_{\approx \ddot{q}_k} = F(q_k)$$

Kann also als direkte Diskretisierung der Bewegungsgleichung $M\ddot{q} = F(q)$ interpretiert werden!

12.5.4 Variationelle Integratoren höherer Ordnung

Wie können wir variationelle Integratoren höherer Ordnung konstruieren?

Bessere Approximation von

$$L_h(q_k, q_{k+1}) := \int_{t_k}^{t_{k+1}} L(q(t), \dot{q}(t)) dt$$

heißt:

- Approximation von q höherer Ordnung,
- Quadraturformel höherer Ordnung.

Idee: (Marsden und West [13])

$$L_h(q_k, q_{k+1}) := \tau \sum_{i=1}^s b_i L(u(c_i\tau), \dot{u}(c_i\tau)) \quad (12.7)$$

- Quadraturformel mit s Stützstellen c_1, \dots, c_s , Gewichten b_1, \dots, b_s
- u ist Polynom vom Grad höchstens s mit
 - $u(0) = q_k, \quad u(\tau) = q_{k+1}$
 - u macht die rechte Seite von (12.7) stationär im Raum aller Polynome vom Grad höchstens s .

Tatsächlich werden von u nur die Werte und Ableitungen an den Stützstellen $c_i\tau$ gebraucht.

Definiere deshalb:

$$Q_i := u(c_i\tau), \quad \dot{Q}_i := \dot{u}(c_i\tau).$$

Die Q_i können durch die \dot{Q}_i ausgedrückt werden:

$$\begin{aligned} Q_i &= u(c_i\tau) = u(0) + \int_0^{c_i} \dot{u}(\sigma\tau) d\sigma \\ &= q_k + \tau \int_0^{c_i} \sum_{j=1}^s L_j(\sigma) \dot{u}(c_j\tau) d\sigma \quad (\text{Lagrange-Darstellung}) \\ &= q_k + \tau \sum_{j=1}^s a_{ij} \dot{Q}_j \quad \text{mit} \quad a_{ij} = \int_0^{c_i} L_j(\sigma) d\sigma. \end{aligned} \quad (12.8)$$

Die b_i sind Quadraturgewichte. Wähle deshalb

$$b_i = \int_0^1 L_i(\sigma) d\sigma.$$

Wir wählen die \dot{Q}_i so, dass der Ausdruck

$$L_h(q_k, q_{k+1}) = \tau \sum_{i=1}^s b_i L(Q_i(\dot{Q}_1, \dots, \dot{Q}_s), \dot{Q}_i)$$

stationär wird.

Allerdings brauchen wir zusätzlich die Nebenbedingung

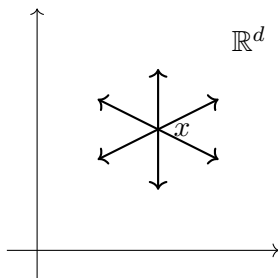
$$q_{k+1} = u(\tau) = u(0) + \tau \int_0^1 \dot{u}(\tau\sigma) d\sigma = q_k + \tau \sum_{i=1}^s b_i \dot{Q}_i. \quad (12.9)$$

Exkurs: Stationarität unter Gleichheitsnebenbedingungen

Betrachte eine Funktion $f : \mathbb{R}^d \rightarrow \mathbb{R}$.

- Wir suchen einen stationären Punkt von f .
- D.h., einen Punkt x , in dem die Richtungsableitung in alle Richtungen v verschwindet

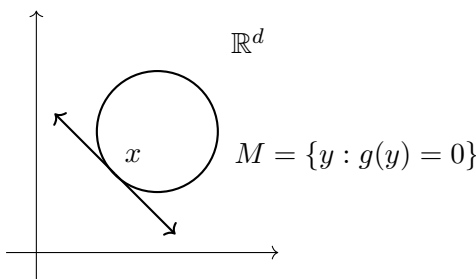
$$\frac{df}{dv} = 0 \quad \forall v \in \mathbb{R}^d, v \neq 0.$$



- D.h. $\nabla f(x) = 0$.

Betrachte jetzt eine weitere Funktion $g : \mathbb{R}^d \rightarrow \mathbb{R}$.

- Wir suchen ein x mit $g(x) = 0$, so dass f in x stationär ist bzgl. der Menge $M = \{y \in \mathbb{R}^d : g(y) = 0\}$.
- D.h. $\frac{df}{dv} = 0$ for alle Richtungen v , die tangential zu M sind.



- $\nabla f(x)$ ist nicht zwangsläufig null!
- Aber $\nabla f(x)$ steht senkrecht auf M .
- $\nabla g(x)$ steht ebenfalls senkrecht auf M .
- Gesucht werden $x \in \mathbb{R}^d$, $\lambda \in \mathbb{R}$, so dass

$$\nabla f(x) = \lambda \nabla g(x).$$

Solch eine Variable λ heißt *Lagrange-Multiplikator*.

Umschreiben: Definiere die Lagrange-Funktion

$$\mathcal{L}(x, \lambda) := f(x) - \lambda g(x).$$

- Der Gradient davon ist

$$\nabla \mathcal{L}(x, y) = \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial x} \\ \frac{\partial \mathcal{L}}{\partial \lambda} \end{pmatrix} = \begin{pmatrix} \nabla f(x) - \lambda \nabla g(x) \\ -g(x) \end{pmatrix}.$$

- Die gesuchten Punkte sind also gerade die stationären Punkte von \mathcal{L} (ohne Nebenbedingungen).

Exkurs Ende

Diese Technik wenden wir auf die Nebenbedingung

$$q_{k+1} = q_k + \tau \sum_{i=1}^s b_i \dot{Q}_i$$

an. Da diese Nebenbedingung d -wertig ist, ist auch der Lagrange-Multiplikator λ aus \mathbb{R}^d .

Wir suchen also stationäre Punkte von

$$\mathcal{L}(\dot{Q}_1, \dots, \dot{Q}_d, \lambda) = \tau \sum_{i=1}^s b_i L(Q_i, \dot{Q}_i) - \left\langle \lambda, \left(q_k - q_{k+1} + \tau \sum_{i=1}^s b_i \dot{Q}_i \right) \right\rangle.$$

Wir berechnen hiervon die partiellen Ableitungen nach den \dot{Q}_j (die part. Ableitungen nach λ sind klar).

$$\frac{\partial \mathcal{L}}{\partial \dot{Q}_j} = \tau \sum_{i=1}^s b_i \left[\frac{\partial L}{\partial q} \cdot \frac{\partial Q_i}{\partial \dot{Q}_j} + \frac{\partial L}{\partial \dot{q}} \cdot \frac{\partial \dot{Q}_i}{\partial \dot{Q}_j} \right] - \underbrace{\left\langle \lambda, \left(\tau \sum_{i=1}^s b_i \frac{\partial \dot{Q}_i}{\partial \dot{Q}_j} \right) \right\rangle}_{=\tau b_j \lambda}.$$

Da

$$\frac{\partial Q_i}{\partial \dot{Q}_j} = \frac{\partial}{\partial \dot{Q}_j} \left(q_k + \tau \sum_{l=1}^s a_{il} \dot{Q}_l \right) = \tau a_{ij} I_{d \times d}$$

folgt

$$\frac{\partial \mathcal{L}}{\partial \dot{Q}_j} = \tau \sum_{i=1}^s b_i \frac{\partial L}{\partial q} \cdot \tau a_{ij} + \tau b_j \frac{\partial L}{\partial \dot{q}} - \tau b_j \lambda.$$

Stationäre Punkte von \mathcal{L} erfüllen also

$$\sum_{i=1}^s b_i \frac{\partial L}{\partial q}(Q_i, \dot{Q}_i) \cdot \tau a_{ij} + b_j \frac{\partial L}{\partial \dot{q}}(Q_j, \dot{Q}_j) = b_j \lambda. \quad (12.10)$$

Wir führen wieder die konjugierten Impulse ein:

$$P_i = \frac{\partial L}{\partial \dot{q}}(Q_i, \dot{Q}_i),$$

und schreiben außerdem formal

$$\dot{P}_i = \frac{\partial L}{\partial q}(Q_i, \dot{Q}_i).$$

Damit vereinfacht sich die Bedingung (12.10) zu

$$\tau \sum_{i=1}^s b_i \dot{P}_i a_{ij} + b_j P_j = b_j \lambda. \quad (12.11)$$

Das allgemeine variationelle Integrationsverfahren hatte die Form

$$p_k = -\frac{\partial L_h}{\partial x}(q_k, q_{k+1}), \quad p_{k+1} = \frac{\partial L_h}{\partial y}(q_k, q_{k+1}).$$

Das rechnen wir jetzt für den konkreten Fall aus.

$$\begin{aligned} p_k &= -\frac{\partial L_h}{\partial q_k}(q_k, q_{k+1}) \\ &= -\tau \sum_{i=1}^s b_i \frac{\partial}{\partial q_k} L(Q_i, \dot{Q}_i) \\ &= -\tau \sum_{i=1}^s b_i \left[\underbrace{\frac{\partial}{\partial x} L(Q_i, \dot{Q}_i)}_{=\dot{P}_i} \cdot \frac{\partial Q_i}{\partial q_k} + \underbrace{\frac{\partial}{\partial y} L(Q_i, \dot{Q}_i)}_{=P_i} \cdot \frac{\partial \dot{Q}_i}{\partial q_k} \right] \\ &= -\tau \sum_{i=1}^s b_i \left[\dot{P}_i \left(I + \tau \sum_{j=1}^s a_{ij} \frac{\partial \dot{Q}_j}{\partial q_k} \right) + P_i \frac{\partial \dot{Q}_i}{\partial q_k} \right] \quad (\text{wg. } Q_i = q_k + \tau \sum_{j=1}^s a_{ij} \dot{Q}_j) \\ &= -\tau \sum_{i=1}^s b_i \dot{P}_i - \tau \sum_{j=1}^s \tau \underbrace{\sum_{i=1}^s b_i \dot{P}_i a_{ij}}_{=b_j \lambda - b_j P_j} \frac{\partial \dot{Q}_j}{\partial q_k} - \tau \sum_{i=1}^s b_i P_i \frac{\partial \dot{Q}_i}{\partial q_k} \\ &= -\tau \sum_{i=1}^s b_i \dot{P}_i - \tau \sum_{j=1}^s (b_j \lambda - b_j P_j) \frac{\partial \dot{Q}_j}{\partial q_k} - \tau \sum_{i=1}^s b_i P_i \frac{\partial \dot{Q}_i}{\partial q_k} \\ &= -\tau \sum_{i=1}^s b_i \dot{P}_i - \tau \sum_{j=1}^s b_j \lambda \frac{\partial \dot{Q}_j}{\partial q_k}. \end{aligned}$$

Differenzieren der Nebenbedingung $q_{k+1} = q_k + \tau \sum_{i=1}^s b_i \dot{Q}_i$ ergibt

$$0 = I + \tau \sum_{i=1}^s b_i \frac{\partial \dot{Q}_i}{\partial q_k}.$$

Deshalb ist

$$p_k = -\tau \sum_{i=1}^s b_i \dot{P}_i + \lambda. \quad (12.12)$$

Ganz ähnlich erhält man

$$p_{k+1} = \frac{\partial L_h}{\partial y}(q_k, q_{k+1}) = \lambda. \quad (12.13)$$

Lemma 12.2. *Es gilt*

1. $p_{k+1} = p_k + \tau \sum_{i=1}^s b_i \dot{P}_i$
2. $q_{k+1} = q_k + \tau \sum_{i=1}^s b_i \dot{Q}_i$

$$3. P_i = p_k + \tau \sum_{j=1}^s \underbrace{(b_j - b_j a_{ji}/b_i)}_{=: \hat{a}_{ij}} \dot{P}_j$$

$$4. Q_i = q_k + \tau \sum_{j=1}^s a_{ij} \dot{Q}_j$$

Beweis. 1. ist (12.12) mit (12.13)

2. ist gerade die Nebenbedingung (12.9), d.h. $u(\tau) = q_{k+1}$.

3. Aus 1. und $p_{k+1} = \lambda$ folgt

$$0 = p_k + \tau \sum_{j=1}^s b_j \dot{P}_j - \lambda.$$

Multiplizieren mit einem b_i ($\neq 0$):

$$0 = b_i p_k + \tau \sum_{j=1}^s b_i b_j \dot{P}_j - b_i \lambda.$$

Addiere $b_i P_i$ auf beiden Seiten:

$$b_i P_i = b_i p_0 + \tau \sum_{j=1}^s b_i b_j \dot{P}_j + \underbrace{b_i P_i - b_i \lambda}_{= -\tau \sum_{j=1}^s b_j a_{ji} \dot{P}_j \text{ (wg. (12.11))}}$$

$$\implies b_i P_i = b_i p_0 + \tau \sum_{j=1}^s (b_j - b_j a_{ji}) \dot{P}_j$$

4. ist (12.8) (Darstellung der Werte Q über Hauptsatz und Lagrange-Darstellung). \square

Die vier Gleichungen aus dem Lemma bilden ein partitioniertes Runge–Kutta-Verfahren $(p_k, q_k) \mapsto (p_{k+1}, q_{k+1})$ für die Gleichungen

$$\dot{p} = \frac{\partial L}{\partial q}(q, \dot{q}), \quad \dot{q} = \frac{\partial L}{\partial p}(q, \dot{q}) \quad \left(\text{bzw. } \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) = \frac{\partial L}{\partial q} \right).$$

Erinnerung: Partitionierte RK-Verfahren für ein System

$$\begin{aligned} \dot{y} &= f(y, z) & \dot{z} &= g(y, z) \\ y_{k+1} &= y_k + \tau \sum_{i=1}^s \hat{b}_i k_i & z_{k+1} &= z_k + \tau \sum_{i=1}^s b_i l_i \\ k_i &= f(y_k + \tau \sum_{j=1}^s \hat{a}_{ij} k_j, z_k + \tau \sum_{j=1}^s a_{ij} l_j) & l_i &= g(y_k + \tau \sum_{j=1}^s \hat{a}_{ij} k_j, z_k + \tau \sum_{j=1}^s a_{ij} l_j) \end{aligned}$$

Symmetrische Form

$$y_{k+1} = y_k + \tau \sum_{i=1}^s \hat{b}_i f(g_i, h_i) \quad z_{k+1} = z_k + \tau \sum_{i=1}^s b_i g(g_i, h_i)$$

$$g_i = y_k + \tau \sum_{i=1}^s \hat{a}_{ij} f(g_i, h_i) \quad h_i = z_k + \tau \sum_{i=1}^s a_{ij} g(g_i, h_i)$$

Wir haben ein Verfahren dieser Bauart, mit

$$\begin{aligned} \dot{P}_i &= f(g_i, h_i) & \dot{Q}_i &= g(g_i, h_i) \\ P_i &= g_i & Q_i &= h_i \\ & & \hat{b}_i &= b_i. \end{aligned}$$

und insbesondere

$$\hat{a}_{ij} = b_j - b_j a_{ji} / b_i.$$

Diese letzte Relation hat eine besondere Eigenschaft:

Satz 12.12 (Hairer, Lubich und Wanner [8, Thm. VI.4.6]). *Wenn für die Koeffizienten eines partitionierten Runge-Kutta-Verfahrens gilt:*

$$\begin{aligned} b_i \hat{a}_{ij} + \hat{b}_j a_{ji} &= b_i \hat{b}_j & i, j &= 1, \dots, s \\ b_i &= \hat{b}_i & i &= 1, \dots, s, \end{aligned}$$

dann ist das Verfahren symplektisch.

12.6 Mechanische Systeme mit Nebenbedingungen

- auch: Zwangsbedingungen
- auch: differential-algebraische Gleichungen (DAEs)

Betrachte System mit Positionskoordinaten q_1, \dots, q_d .

- Nebenbedingung: sei $g : \mathbb{R}^d \rightarrow \mathbb{R}^m$ mit $m < d$.
- Nur Positionen $q \in \mathbb{R}^d$ mit $g(q) = 0$ sind erlaubt.

Wie sehen Bewegungsgleichungen aus, wenn es solche eine Nebenbedingung gibt? Formulierung dieser Nebenbedingung wieder mit Lagrange-Multiplikator.

- Kinetische Energie: $T(q, \dot{q}) = \frac{1}{2} \dot{q}^T M \dot{q}$
- Potentielle Energie: $U(q)$

- Lagrange-Funktion

$$L(q, \dot{q}) = T(q, \dot{q}) - U(q) - g(q)^T \lambda$$

mit Lagrange-Multiplikatoren $\lambda_1, \dots, \lambda_m$.

Stationäre Punkte von

$$S(q, \lambda) := \int L(q(t), \dot{q}(t), \lambda(t)) dt$$

erfüllen dann

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} &= 0 \\ g(q(t)) &= 0 \quad \forall t \end{aligned}$$

Bewegungsgleichungen:

$$\frac{\partial L}{\partial \dot{q}} = M\dot{q}, \quad \frac{\partial L}{\partial q} = -\nabla U - (\nabla g)^T \lambda$$

$$\begin{aligned} M\ddot{q} + \nabla U(q) + (\nabla g(q))^T \lambda &= 0 \\ g(q) &= 0 \end{aligned}$$

System erster Ordnung:

$$\begin{aligned} v &= \dot{q} \\ M\dot{v} &= -\nabla U(q) - (\nabla g(q))^T \lambda \\ g(q) &= 0 \end{aligned}$$

Beispiel: Das Kugelpendel

- Wieder ein Fadenpendel, aber das Pendel darf sich in drei Dimensionen bewegen
- Beschreibung mit *zwei* Winkeln: Möglich, aber technisch
- Alternative: Kartesische Koordinaten q_1, q_2, q_3
- Nebenbedingung: Die Länge des Pendels ist fest:

$$g(q) = q_1^2 + q_2^2 + q_3^2 - l^2 = 0$$

- Kinetische Energie:

$$T = \frac{m}{2} (\dot{q}_1^2 + \dot{q}_2^2 + \dot{q}_3^2)$$

- Potentielle Energie:

$$U = mgq_3$$

Bewegungsgleichungen:

$$\begin{aligned} v_1 &= \dot{q}_1, & v_2 &= \dot{q}_2, & v_3 &= \dot{q}_3 \\ m\dot{v}_1 &= -2q_1\lambda, & m\dot{v}_2 &= -2q_2\lambda, & m\dot{v}_3 &= -mg - 2q_3\lambda \\ 0 &= q_1^2 + q_2^2 + q_3^2 - l^2 \end{aligned}$$

Der Lagrange-Multiplikator λ kann als die Spannung im Faden interpretiert werden. Betrachte ein mechanisches System mit Positionskoordinaten q_1, \dots, q_d .

- Nebenbedingung: Sei $g : \mathbb{R}^d \rightarrow \mathbb{R}^m$ mit $m < d$.
- Es sind nur solche Positionen $q \in \mathbb{R}^d$ erlaubt, für die $g(q) = 0$ gilt.

Betrachte ein mechanisches System mit

- Kinetischer Energie: $T(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q}$.
- Potenzieller Energie $U(q)$

Lagrange-Funktion $L(q, \dot{q}, \lambda) = T(q, \dot{q}) - U(q) - g(q)^T \lambda$ mit Lagrange-Multiplikatoren $\lambda_1, \dots, \lambda_m$.

Wir suchen Lösungen der Lagrange-Gleichung:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = 0$$

Dazu rechnen wir:

$$\begin{aligned} \frac{\partial L}{\partial \dot{q}} &= \frac{\partial}{\partial \dot{q}} \left(\frac{1}{2} \dot{q}^T M(q) \dot{q} \right) = M(q) \dot{q} \\ \frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{q}} \right) &= \dot{q}^T \frac{\partial M}{\partial q} \dot{q} + M(q) \ddot{q} \\ \frac{\partial L}{\partial q} &= \frac{\partial}{\partial q} T(q, \dot{q}) - \frac{\partial U}{\partial q} - \frac{\partial g}{\partial q} \lambda \end{aligned}$$

Schreiben als System erster Ordnung:

$$\begin{aligned} \dot{q} &= v \\ M(q)\dot{v} &= \underbrace{-v^T \frac{\partial M}{\partial q} v + \frac{\partial}{\partial q} T(q, v)}_{:=f(q,v)} - \frac{\partial U}{\partial q} - \underbrace{\frac{\partial g^T}{\partial q}}_{:=G(q)} \lambda \\ 0 &= g(q) \end{aligned}$$

Geometrische Interpretation: Wir betrachten jetzt die Menge

$$M = \{q \in \mathbb{R}^d : g(q) = 0\}$$

als geometrisches Objekt.

Wenn g hinreichend freundlich ist, dann ist M eine glatte, gekrümmte, geschlossene $(d - m)$ -dimensionale Fläche in \mathbb{R}^d .

Der Profi sagt: M ist eine $(d - m)$ -dimensionale Mannigfaltigkeit.

M ist genau die Menge aller Werte, die q annehmen darf.

- Man könnte jetzt also auf die Idee kommen, die Differentialgleichung für q nicht mehr „in \mathbb{R}^d “ sondern „auf M “ zu definieren.
- Und die restlichen Punkte $\mathbb{R}^d \setminus M$ komplett vergessen!

Aber: in welchem Raum leben die Geschwindigkeiten v ?

- Sie leben *nicht* in M !

Definition. Der Tangentialraum $T_q M$ von M in einem Punkt q ist die Menge aller Tangentialvektoren zu M im Punkt q .

- Ein Vektorraum! (für festes q)

Tangential zu M heißt in unserem Fall gerade $d \perp M|_q \Leftrightarrow \nabla g(q)^T d = 0$. Betrachte jetzt die Nebenbedingung

$$g(q) = 0$$

Sei $q := [-\epsilon, \epsilon] \rightarrow M$ ein Pfad in M . Ableiten von

$$g(q) = 0$$

ergibt

$$\nabla g(q)^T \dot{q} = 0$$

Ergo: $\dot{q} = v$ ist immer tangential zu M . Soweit galt das nur an einem festen Punkt auf M . Jetzt betrachten wir alle Punkte.

Definition. Das Tangentialbündel TM von M ist die disjunkte Vereinigung aller Tangentialräume von M :

$$TM := \{(q, v) : q \in M, v \in T_q M\}$$

- Kein linearer Raum!
- Aber eine $2(d - m)$ -dimensionale Mannigfaltigkeit.

Wir können also die Bewegungsgleichungen als ein System von gewöhnlichen Differentialgleichungen erster Ordnung auf der Mannigfaltigkeit TM auffassen.

Hamilton-Formulierung

- Definiere Impulse (wie gehabt)

$$p = \frac{\partial L}{\partial \dot{q}} = M(q)\dot{q}$$

- Hamilton-Funktion

$$\begin{aligned}\tilde{H}(p, q) &:= p^T \dot{q} - L(q, \dot{q}) \\ &= p^T \dot{q} - \frac{1}{2} \dot{q}^T M(q) \dot{q} + U(q) + g(q)^T \lambda \\ &= \underbrace{\frac{1}{2} \dot{q}^T M(q) \dot{q} + U(q)}_{=H(p, q)} + g(q)^T \lambda\end{aligned}$$

- Hamilton-System

$$\begin{aligned}\dot{q} &= \frac{\partial H}{\partial p} \\ \dot{q} &= -\frac{\partial H}{\partial q} - \nabla g(q)^T \lambda \\ 0 &= g(q)\end{aligned}$$

Statt der Tangentialitätsbedingung $\nabla g(q)^T v = 0$ haben wir dismal

$$\nabla g(q)^T \frac{\partial H}{\partial p} = 0$$

Ausblick: Dies sind Gleichungen auf dem Kotangentenbündel T^*M von M :

$$T^*M := \{(q, l) : q \in M, l \text{ ist lineares Funktional auf } T_q M\}$$

- Die Hamilton-Funktion H wird weiterhin erhalten.
- Die Flüsse sind weiterhin symplektisch.

Einfaches Verfahren erster Ordnung

Basis: symplektisches Euler-Verfahren

- p implizit, q explizit

$$\begin{aligned}\hat{p}_{k+1} &= p_k - \tau \left(\frac{\partial H}{\partial q}(\hat{p}_{k+1}, q_k) + \nabla g(q_k)^T \lambda_{k+1} \right) \\ q_{k+1} &= q_k + \tau \frac{\partial H}{\partial p}(\hat{p}_{k+1}, q_k) \\ 0 &= g(q_{k+1})\end{aligned}$$

Der neue Wert (\hat{p}_{k+1}, q_{k+1}) erfüllt $0 = g(q_{k+1})$, aber nicht $\nabla g(q_{k+1}) \frac{\partial H}{\partial p}(p_{k+1}, q_{k+1})$.
Deshalb: Projektionsschritt

$p_{k+1} = \hat{p}_{k+1} - \tau \nabla g(q_{k+1})^T \mu_{k+1}$ TODO Handgeschrieben steht hier eindeutig $\mu \dots$

$$0 = \nabla g(q_{k+1}) \frac{\partial H}{\partial p}(p_{k+1}, q_{k+1})$$

- Wohldefiniert für kleine τ
- Konsistent erster Ordnung
- Symplektisch

Literatur

- [1] G. Benettin und A. Giorgilli. “On the Hamiltonian interpolation of near-to-the identity symplectic mappings with application to symplectic integration algorithms”. In: *Journal of Statistical Physics* 74 (März 1994), S. 1117–1143. DOI: 10.1007/BF02188219.
- [2] W. Dahmen und A. Reusken. *Numerik für Ingenieure und Naturwissenschaftler*. Springer, 2008.
- [3] P. Deuffhard und F. Bornemann. *Numerische Mathematik 2 – Gewöhnliche Differentialgleichungen*. de Gruyter, 2008.
- [4] P. Deuffhard und A. Hohmann. *Numerische Mathematik 1*. de Gruyter, 1993.
- [5] J. Dieudonné. *Foundations of Modern Analysis*. Academic Press, 1960.
- [6] I. S. Duff und J. K. Reid. “The Multifrontal Solution of Indefinite Sparse Symmetric Linear Equations”. In: *ACM Transactions on Mathematical Software (TOMS)* 9.3 (1983), S. 302–325.
- [7] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations*. Springer, 1994.
- [8] E. Hairer, C. Lubich und G. Wanner. *Geometric Numerical Integration—Structure-Preserving Algorithms for Ordinary Differential Equations*. Zweite Auflage. Springer, 2016.
- [9] C. A. Hall und W. W. Meyer. “Optimal Error Bounds for Cubic Spline Interpolation”. In: *Journal of Approximation Theory* 16 (1976), S. 105–122.
- [10] M. R. Hestenes und E. Stiefel. “Methods of Conjugate Gradients for Solving Linear Systems”. In: *Journal of Research of the National Bureau of Standards* 49 (1952), S. 409–436.
- [11] J. Liu. “The Multifrontal Method for Sparse Matrix Solution: Theory and Practice”. In: *SIAM Review* 34.1 (1992), S. 82–109.
- [12] R. MacKay. “Some Aspects of the Dynamics of Hamiltonian Systems”. In: *The Dynamics of Numerics and the Numerics of Dynamics*. Hrsg. von D. Broomhead und A. Iserles. Clarendon Press, Oxford, 1992, S. 137–193.
- [13] J. E. Marsden und M. West. “Discrete mechanics and variational integrators”. In: *Acta Numerica* 10 (2001), S. 357–514. DOI: 10.1017/S096249290100006X.
- [14] J. Nocedal und S. J. Wright. *Numerical Optimization*. Springer, 2006.
- [15] J. R. Shewchuk. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Techn. Ber. Pittsburgh, PA, USA, 1994.

- [16] J. Wendlandt und J. Marsden. “Mechanical Integrators Derived From a Discrete Variational Principle”. In: *Physica D* 106 (1997), S. 223–246.