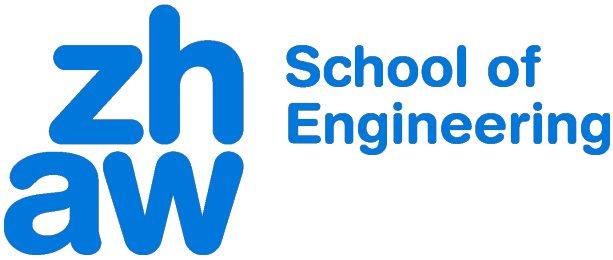
ZHAW Zurich University of Applied Sciences
Winterthur



Zusammenfassung NUM - CookBook Studienwochen 1-14

Written by: Severin Sprenger
June 17, 2026
Zf. NUM - CookBook SW 1-14

1 Lineare Gleichungssysteme (LGS)

1.1 Identifikation & Methodenwahl

- **Matrix A ist quadratisch ($n \times n$) und vollbesetzt:** LR-Zerlegung (LU) mit Pivotisierung.
- **Matrix A ist symmetrisch und positiv definit (spd):** Cholesky-Zerlegung (oft bei Normalengleichungen).
- **Matrix A ist überbestimmt ($m > n$):** QR-Zerlegung (Ausgleichsrechnung) zur Vermeidung schlechter Konditionierung.
- **Matrix A ist eine Tridiagonalmatrix:** Thomas-Algorithmus (FDM 1D, Splines).

1.2 Manuelles Rezept: LR-Zerlegung ($A = LR$)

1. Start: $R = A$, $L = I$ (Einheitsmatrix).
2. Für jede Spalte j (von 1 bis $n - 1$):
3. Finde Pivotelement (grösster Betrag in Spalte j ab Zeile j). Zeilen in L, R und Permutationsmatrix P tauschen.
4. Berechne Multiplikatoren: $l_{ij} = \frac{r_{ij}}{r_{jj}}$ für $i > j$. Trage l_{ij} in L ein.
5. Optimiere R: Subtrahiere $l_{ij}\times$ Zeile j von Zeile i in R.
6. **Edge Case:** Diagonalelement $r_{jj} = 0$ (oder sehr klein) $\Rightarrow$ Pivotisierung zwingend! Matrix evtl. singulär.

2 Nichtlineare Gleichungen & Systeme

2.1 Identifikation

Aufgabenstellung fragt nach "Nullstelle", "Schnittpunkt zweier Kurven", "Lösen Sie $f(x) = y$".

- **1D:** Skalares Newton-Verfahren.
- **Mehrdimensional:** Jacobi-Matrix und LGS-Lösung nötig.

2.2 Manuelles Rezept: Mehrdimensionales Newton-Verfahren

Ziel: Finde $\mathbf{x}$ für $\mathbf{F}(\mathbf{x}) = \mathbf{0}$.

1. Formuliere die Vektorfunktion $\mathbf{F}(\mathbf{x})$. Alle Terme auf eine Seite bringen!
2. Bilde die Jacobi-Matrix $J(\mathbf{x})$ durch partielle Ableitungen: $J_{ij} = \frac{\partial F_i}{\partial x_j}$.
3. **Iteration k:**
4. Werte $\mathbf{F}(\mathbf{x}^{(k)})$ und $J(\mathbf{x}^{(k)})$ an der aktuellen Stelle aus.
5. Löse das LGS $J(\mathbf{x}^{(k)}) \cdot \Delta\mathbf{x} = -\mathbf{F}(\mathbf{x}^{(k)})$.
6. Update: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta\mathbf{x}$.
7. **Sanity Check:** Ist $\|\mathbf{F}(\mathbf{x}^{(k+1)})\| < \|\mathbf{F}(\mathbf{x}^{(k)})\|$? Das Residuum muss schrumpfen!
8. **Edge Case:** $J(\mathbf{x})$ ist singulär (Determinante ≈ 0) $\Rightarrow$ Schrittweite kann nicht berechnet werden. Startwert ändern!

3 Ausgleichsrechnung (Curve Fitting)

3.1 Identifikation

"Bestimme Parameter a, b so, dass die Fehlerquadratsumme minimal wird." Mehr Messpunkte als Parameter.

- **Linear:** Modellfunktion ist linear in den Parametern (z.B. $f(x) = a \cdot x^2 + b \cdot e^x$).
- **Nichtlinear:** Parameter treten in nichtlinearer Form auf (z.B. $f(x) = a \cdot e^{bx}$). $\Rightarrow$ Gauss-Newton!

3.2 Manuelles Rezept: Lineare Ausgleichsrechnung (Normalengleichung)

1. Stelle die Designmatrix A auf. Jede Zeile ist ein Messpunkt x_i, jede Spalte repräsentiert die Auswertung der jeweiligen Basisfunktion an x_i.
2. Rechte Seite $\mathbf{b}$ ist der Vektor der Messwerte y_i.
3. Berechne $A^T A$ (ergibt eine quadratische, spd Matrix).
4. Berechne $A^T \mathbf{b}$.
5. Löse $A^T A\mathbf{x} = A^T \mathbf{b}$ (von Hand oft mit Cholesky oder Gauss).
6. **Edge Case:** Polynom-Fitting hohen Grades $\Rightarrow A^T A$ wird extrem schlecht konditioniert (Vandermonde-Matrix). Lösung: QR-Zerlegung direkt auf $A\mathbf{x} = \mathbf{b}$ anwenden (Löse $R\mathbf{x} = Q^T \mathbf{b}$).

4 Gewöhnliche Differentialgleichungen (AWP)

4.1 Identifikation

"Gegeben sei die DGL $y' = \ldots, y(0) = \ldots$ Führen Sie einen Schritt mit Verfahren X aus."

- **Prüfung auf Steifheit:** Wenn explizite Verfahren (Euler, RK4) explodieren, implizites Verfahren nutzen.

4.2 Manuelles Rezept: Runge-Kutta 4 (Klassisch)

1. Identifiziere $f(t, y)$, Startwerte t_0, y_0 und Schrittweite h.
2. Berechne $k_1 = f(t_0, y_0)$.
3. Berechne $k_2 = f(t_0 + \frac{h}{2}, y_0 + \frac{h}{2}k_1)$. (*Achtung: y_0 updaten, nicht nur t!*)
4. Berechne $k_3 = f(t_0 + \frac{h}{2}, y_0 + \frac{h}{2}k_2)$.
5. Berechne $k_4 = f(t_0 + h, y_0 + hk_3)$.
6. Finales Update: $y_1 = y_0 + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$.
7. **Sanity Check:** Für $f(t, y)$ linear, vergleiche mit analytischer Taylor-Entwicklung (RK4 stimmt bis h^4 exakt überein).

4.3 Manuelles Rezept: Impliziter Euler (1 Schritt)

1. Ansatz: $y_{n+1} = y_n + h \cdot f(t_{n+1}, y_{n+1})$.
2. **Linearer Fall ($f(t, y) = \lambda y$):** Gleichung algebraisch nach y_{n+1} auflösen: $y_{n+1} = \frac{y_n}{1 - h\lambda}$.
3. **Nichtlinearer Fall:** Definiere Nullstellenproblem $F(z) = z - y_n - h \cdot f(t_{n+1}, z) = 0$. Führe Newton-Verfahren für z (das ist y_{n+1}) aus. Startwert für Newton: Expliziter Euler-Schritt!

5 Partielle Differentialgleichungen & Randwertprobleme

5.1 Identifikation

Ableitungen nach Ort (u'', u_{xx}), Randwerte an zwei Enden ($u(a) = x, u(b) = y$).

5.2 Manuelles Rezept: 1D Poisson ($u'' = f(x)$) mit FDM

1. Bestimme Schrittweite $h = \frac{b-a}{N+1}$.
2. Diskretisiere die Ableitung: $\frac{u_{i-1}-2u_i+u_{i+1}}{h^2} = f_i$.
3. Isoliere die Unbekannten u_i. Bringe f_i auf die rechte Seite.
4. Randwerte eintragen: Für $i = 1$ ist u_0 bekannt, bringe es als Konstante auf die rechte Seite!
5. Stelle Matrix auf: Diagonale ist oft -2, Nebendiagonalen 1.
6. **Edge Case (Neumann Randbedingung $u'(a) = c$):** Nutze zentralen Differenzenquotienten am Rand $\frac{u_1-u_{-1}}{2h} = c$. Löse nach dem "Geisterpunkt" $u_{-1} = u_1 - 2hc$ auf und setze dies in die FDM-Gleichung für $i = 0$ ein.

6 Spline-Interpolation

6.1 Manuelles Rezept: Kubische Splines (Natürliche Ränder)

1. Überprüfe die Intervalle $h_i = x_{i+1} - x_i$.
2. Stelle das LGS für die Momente M_i (zweite Ableitungen an den Knoten) auf.
3. Nutze die Formel: $\frac{h_{i-1}}{6}M_{i-1} + \frac{h_{i-1}+h_i}{3}M_i + \frac{h_i}{6}M_{i+1} = \frac{y_{i+1}-y_i}{h_i} - \frac{y_i-y_{i-1}}{h_{i-1}}$.
4. Setze Randbedingungen: $M_0 = 0$ und $M_n = 0$ (natürlich). Dies reduziert das LGS!
5. Löse nach den inneren M_i auf (Tridiagonalsystem).
6. Berechne Spline-Parameter a_i, b_i, c_i, d_i aus $y_i, y_{i+1}, M_i, M_{i+1}$.
7. **Sanity Check:** Sind die berechneten $c_i = \frac{M_i}{2}$? Prüfe am ersten Intervall, ob die Interpolationsbedingung $S_0(x_0) = y_0$ gilt.