HELMUTH GRIEB

NUMERISCHE MATHEMATIK

Copyright Prof. Dr. Helmuth Grieb

Fachhochschule für Technik Esslingen

3 – 1996

Inhaltsverzeichnis: Seite

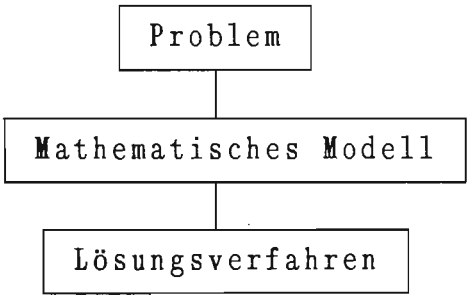
1. Einführung 4
   1.1. Übersicht über Angewandte und Numerische Mathematik 4
   1.2. Grundsätzliches über numerische Verfahren 5
2. Iterationsverfahren zur Lösung nichtlinearer Gleichungen 9
   2.1. Beispiele für nichtlineare Gleichungen 9
   2.2. Das Newton–Verfahren 10
   2.3. Das Sehnenverfahren (regula falsi) 13
   2.4. Bemerkungen 14
   2.5. Algebraische Gleichungen 16
3. Lineare Gleichungssysteme 17
   3.1. Problem und Lösungstheorie 17
   3.2. Der Gauß–Algorithmus 19
      3.2.1. Der Sonderfall einer Dreiecksmatrix 19
      3.2.2. Der Algorithmus zur Elimination 21
      3.2.3. Die Wahl des Pivotelements 24
      3.2.4. Kondition eines linearen Gleichungssystems 25
   3.3. Weitere direkte Lösungsverfahren bei regulärer n,n–Matrix A 26
      3.3.1. Lösung eines LGS durch Zerlegung in Dreiecksmatrizen 26
      3.3.2. Die Choleski – Zerlegung 27
      3.3.3. Vergleich des Rechenaufwands 27
   3.4. Nachiteration zur Verbesserung der Lösung eines LGS 28
   3.5. Iterationsverfahren zur Lösung eines LGS 29
      3.5.1. Prinzip 29
      3.5.2. Das Gesamtschrittverfahren (Jacobi – Verfahren) 30
      3.5.3. Das Einzelschrittverfahren (Gauß – Seidel) 31
4. Fehlerausgleich bei überbestimmten linearen Gleichungssystemen 32
   4.1. Problemstellung 32
   4.2. Fehlerausgleich nach der Methode der kleinsten Quadrate 33
   4.3. Fehlerausgleich mit Nebenbedingungen 37
5. Interpolation durch Polynome 38
   5.1. Problemstellung 38
   5.2. Der Satz vom Interpolationspolynom 39
   5.3. Das Interpolationspolynom nach Newton 40
   5.4. Abschließende Bemerkungen zu Interpolationspolynomen 44

6. Approximation von Funktionen 45
6.1. Einführung 45
6.2. Approximation einer Funktion durch ihr Taylorpolynom 45
6.3. Approximation von f(x) durch ein Stützpolynom 46
6.4. Approximation durch kubische Splines 47
6.5. Approximation durch Ausgleichspolynome 51
7. Periodische Funktionen und trigonometrische Polynome 56
7.1. Periodische Funktionen 56
7.2. Harmonische Schwingungen 56
7.3. Trigon. Polynome und Fourierreihen für 2π–periodische Funktionen 57
7.4. Diskrete Approximation und Numerische Fourierreihe 58
7.4.1. Approximation durch ein trigonometrisches Polynom 58
7.4.2. Trigonometrische Interpolation 61
7.4.3. Beispiel 62
7.5. Bemerkungen 64
8. Numerische Differentiation und Integration 65
8.1. Numerische Differentiation 65
8.2. Grundprinzipien der numerischen Integration 66
8.3. Grundformeln der numerischen Integration 67
8.4. Summierte Quadraturformeln 68
8.4.1. Die allgemeine Trapezregel (summierte Trapezregel) 68
8.4.2. Die Simpson–Formel 69
8.4.3. Beispiel 70
8.5. Fehlerverhalten 70
8.6. Das Romberg–Verfahren 71
8.7. Gauß–Integration 77
9. Differentialgleichungen 82
9.1. Anfangs– und Randwertprobleme 82
9.2. Aufbau von Näherungsverfahren bei Anfangswertproblemen 85
9.3. Einschrittverfahren bei Differentialgleichungen 1. Ordnung 86
9.3.1. Das Euler–Cauchysche Polygonzugverfahren 86
9.3.2. Das Verfahren von Heun 87
9.3.3. Das Runge–Kutta–Verfahren 88
9.4. Bemerkungen zur Schrittweite h 90
9.5. Das Runge–Kutta–Verfahren für Anfangswertprobleme bei Systemen und Differentialgleichungen höherer Ordnung 90
Literaturverzeichnis 93

1. Einführung

1.1. Übersicht über Angewandte und Numerische Mathematik

Zur Lösung eines naturwissenschaftlich–technischen Problems benötigt man zunächst ein "Mathematisches Modell". Im Rahmen dieses Modells wird dann ein Lösungsverfahren angestrebt.



Beispiele für mathematische Modelle:

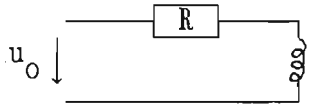
Formeln, Gleichungen, Gleichungssysteme, Differentialgleichungen, Statistische Modelle usw. .

Jedes Modell ist vom Anwender auf seine Zulässigkeit zu überprüfen; falsche Ergebnisse können auch trotz richtiger Lösung von einem falsche Modell verursacht sein.

Beispiel: Einschwingvorgang bei einer Induktivität

Bei einer Luftspule gilt für den Fluß $\phi = L \cdot i$ und $u_L = \frac{d\phi}{dt} = L \cdot \frac{di}{dt}$.

Dies führt bei einem Einschwingvorgang an einem seriellen R–L–Glied zu der linearen Differentialgleichung



$$L \cdot \frac{di}{dt} + R \cdot i = u_o \ ;$$

diese kann i.a. mit einfachen Standardverfahren gelöst werden.

Die Modellvorstellung $\phi = L \cdot i$ gilt aber für eine Spule mit Eisenkern nur noch in grober Näherung.

Genauer ist jetzt ein Ansatz $i = f(\phi)$ für den Strom und damit eine Differentialgleichung

$$\frac{d\phi}{dt} + R \cdot f(\phi) = u_o \ .$$

Je nach Ansatz für die Funktion $f(\phi)$ ergibt sich eine nichtlineare Differentialgleichung, die i.a. nicht mehr analytisch lösbar ist.

Die Angewandte Mathematik beschäftigt sich hauptsächlich mit folgenden Fragen:

- Unter welchen Voraussetzungen besitzt das mathematische Modell eine Lösung (Existenz) ?
- Gibt es eine oder mehrere Lösungen (Eindeutigkeit) ?
- Wie hängt die Lösung von den Eingangsdaten ab ?
- Wie läßt sich die Lösung berechnen (konstruktive Verfahren) ?

Wir beschäftigen uns im folgenden v.a. mit den konstruktiven Lösungsverfahren: Diese liefern zu Eingangsdaten konkrete Ergebnisdaten. Man unterscheidet dabei

- Analytische Verfahren: Benutzt werden Methoden aus der Algebra, und der Differential– und Integralrechnung
- Numerische Verfahren: Auf Rechner zugeschnittene Verfahren; Arbeitsgebiet der Numerischen Mathematik (Numerik)
- Statistische Verfahren: Verwendet werden Methoden aus der Wahrscheinlichkeitsrechnung und der Statistik.

1.2. Grundsätzliches über numerische Verfahren

Forderungen an ein numerisches Verfahren:

- Es muß darstellbar sein als "Algorithmus": Genau festgelegte Folge einzelner elementarer Rechenoperationen ("rezeptartig"), die aus den Eingabedaten nach endlich vielen Schritten Ergebnisdaten liefern.

Man unterscheidet dabei

- direkte Algorithmen (z.B. Lösungsformeln)
- iterative Algorithmen. Bei diesen wird durch wiederholte Anwendung des Algorithmus' die Genauigkeit schrittweise verbessert.

Anforderungen an einen Algorithmus:

- er darf weder logisch noch arithmetisch versagen;
- er soll keine computerabhängige Daten verwenden;
- der Fehler soll mit wachsender Ziffernlänge des Rechners gegen 0 gehen.

Wichtigste Fehlerarten bei Numerischen Verfahren:

– **Eingangsfehler**: Wichtig sind hier v.a.

  Fehler in den Eingangsdaten (z.B. Meßfehler, Größen wie die Erdbeschleunigung sind nicht exakt usw.),

  Konvertierungsfehler (Fehler bei der Umwandlung von Daten in Maschinenzahlen).

  Die Eingangsfehler lassen sich i.a. auf einen gewissen "unvermeidbaren Fehler" reduzieren.

– **Abschneide– und Rundungsfehler**: Dem Rechner stehen nur endlich viele Ziffern zur Verfügung.

– **Abbrechfehler**: Der Rechner muß jede Berechnung nach endlich vielen Schritten abbrechen.

  Beispiel: Eine unendliche Reihe kann nur durch eine endliche Summe angenähert werden. Dies kann im Rahmen der dem Rechner zur Verfügung stehenden Dezimalstellen zum richtigen Wert führen, das Ergebnis kann aber auch völlig falsch sein. So liefert etwa die divergente Reihe

$$\sum_{k=1}^{\infty} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \ldots$$

  auf jedem Rechner einen (falschen) endlichen Wert.

– **Diskretisierungsfehler**: Bei "Diskretisierungsverfahren" werden anstelle von Methoden aus der Analysis Ersatzoperationen verwendet (z.B. Differenzenquotient anstelle einer Ableitung). Die dadurch auftretenden Fehler bezeichnet man als Diskretisierungsfehler.

Kondition, Stabilität:

Kondition eines Problems: Wie wirken sich Eingangs– und Rundungsfehler auf die Genauigkeit der Ergebnisdaten aus ?

Geringe Auswirkung (Größenordnung des unvermeidbaren Fehlers): Gut konditioniertes Problem.

Starke Auswirkung: Schlecht konditioniertes Problem.

Beispiele:

1) Ein Produkt $P = x \cdot y$ ist gut konditioniert. Ändern sich etwa x und y um einen kleinen Wert h, so ändert sich P um $\Delta P = (x + h) \cdot (y + h) - x \cdot y = (x + y) \cdot h + h^2$. Dieser Wert bleibt in der Größenordnung von h .

2) Bei Gleitpunktarithmetik ist die Differenz $x - y$ für $x \approx y$ schlecht konditioniert. Berechnet man etwa (bei 10 Stellen) $1.234567893 - 1.234567892 = 1.000000000\ E-9$, so sind die Stellen hinter dem Dezimalpunkt beim Ergebnis "ausgelöscht" . Derartige Auslöschungseffekte können bei mehreren Summanden unterschiedlicher Größenordnung ein Ergebnis stark verfälschen.

3) Algebraische Gleichungen sind schlecht konditionierte Probleme.
So besitzt etwa die Gleichung $x^2 + 4x + 4 = (x + 2)^2 = 0$
die Lösungen $x_1 = x_2 = -2$.
Beim Problem $x^2 + 4x + 3.9999999 = 0$ tritt ein Fehler von $|\Delta c| \approx 10^{-7}$ in den Eingangsdaten auf. Die Lösungen $x_1^* = -1.9996837$, $x_2^* = -2.0003126$ besitzen einen Fehler der Größenordnung $|\Delta x_i| = |x_i^* - x_i| \approx 3 \cdot 10^{-4} = 3000 \cdot |\Delta c|$.

Stabilität von Algorithmen: Ein Algorithmus heißt numerisch stabil oder gutartig, wenn der Fehler des Ergebnisses in der Größenordnung des unvermeidbaren Fehlers ist.

Nicht jeder Algorithmus eines gut konditionierten Problems ist notwendigerweise stabil.

Beispiel: Der Algorithmus $D1 = \sqrt{a + h} - \sqrt{a}$ ist für $|h| << a$ instabil (Auslöschungseffekte), während der durch algebraische Umformung entstehende Algorithmus $D2 = \dfrac{h}{\sqrt{a + h} + \sqrt{a}}$ stabil ist .

Aufgaben der Numerischen Mathematik

- Entwicklung von Algorithmen für numerische Probleme (und gegebenenfalls von Rechnerprogrammen),
- Vergleich der Verfahren im Hinblick auf
  - Effektivität (Rechenzeit, Speicherplatz),
  - Stabilität,
- Angabe von Fehlerschranken für die Resultate.

Schwerpunkte des Skriptums: Entwicklung und Darstellung grundlegender Algorithmen. Fehleruntersuchungen können nur am Rande behandelt werden.

2. Iterationsverfahren zur Lösung nichtlinearer Gleichungen

2.1. Beispiele für nichtlineare Gleichungen

a) Algebraische Gleichungen n–ten Grades

$$p_n(x) = \sum_{k=0}^{n} a_k x^k = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0 = 0$$

Für $n \geq 5$ existieren keine analytischen Lösungsformeln mehr.

Beispiel: Ein Kapital K wird auf m Monate bei einer monatlichen Rückzahlungsrate r (Zins + Tilgung) ausgeliehen. Fragt man nach dem entsprechenden jährlichen Zinssatz p, so ist folgende Gleichung für $x = 1 + \frac{p}{1200}$ zu lösen:

$$K \cdot x^{m+1} - (K + r) \cdot x^m + r = 0 \ .$$

b) Transzendente Gleichungen, z.B. $\cos x = x$.

Diese Gleichung entspricht dem Schnitt der Cosinuskurve mit der Ursprungsgeraden $y = x$.

Im Allgemeinfall existieren für derartige nichtlineare Gleichungen keine analytischen Lösungsformeln mehr . Will man die Lösungen bestimmen, benötigt man Näherungsverfahren. In diesem Abschnitt wird einer der wichtigsten Typen derartiger Verfahren behandelt, die **Iterationsverfahren**:

Ausgehend von einem Startwert (Rohwert) x_0 wird eine Folge von Näherungswerten x_k konstruiert, die (unter bestimmten Voraussetzungen) gegen die gesuchte Lösung konvergiert.

Ein Startwert läßt sich häufig aus dem Verlauf von Funktionskurven bestimmen (vgl. Beispiel b).

Die zur Berechnung erforderliche Formel nennt man **Iterationsvorschrift**.

2.2. Das Newton–Verfahren

Das Newton–Verfahren ist ein Iterationsverfahren zur Berechnung von Nullstellen einer Funktion f(x). Eine nichtlineare Gleichung der Form $f_1(x) = f_2(x)$ ist vor Anwendung des Verfahrens in die Form $f(x) = f_1(x) - f_2(x) = 0$ umzuformen.

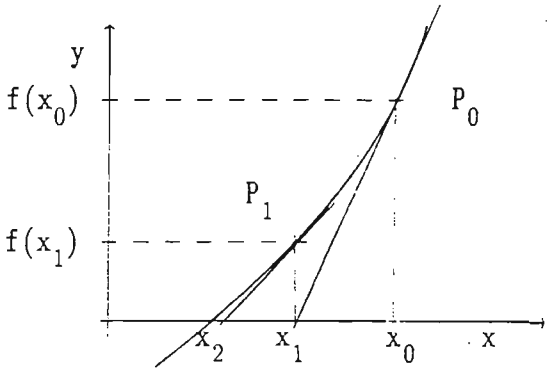
Es beruht auf folgendem Prinzip ("Tangenten – Näherungsverfahren"):

(1) Man wählt einen Startwert x_0 und approximiert die Kurve durch die Tangente t in $P_0(x_0|y_0)$. Dann berechnet man den Schnittpunkt $(x_1|0)$ der Tangente t mit der x–Achse. Bei geschickter Wahl von x_0 ergibt sich ein verbesserter Wert x_1.

Wiederholung (Iteration):

(2) x_1 wird neuer Startwert. Tangente in $P_1(x_1|f(x_1))$ → verbesserter Wert x_2.

(3) x_2 wird neuer Startwert. Tangente in $P_2(x_2|f(x_2))$ → verbesserter Wert x_3.
usw.



Die so konstruierte Folge $\{x_n\}$ konvergiert unter bestimmten Voraussetzungen gegen die gesuchte Lösung $\bar{x}$. Das Verfahren wird abgebrochen, wenn eine geforderte Genauigkeit erreicht ist.

Herleitung der Iterationsvorschrift:

Gleichung der Tangente in $P_n(x_n|y_n)$: $y - f(x_n) = f'(x_n)\cdot(x - x_n)$

Schnitt mit der x–Achse $(y = 0 \rightarrow x = x_{n+1})$:

$$-f(x_n) = f'(x_n)\cdot(x_{n+1} - x_n)$$

$$\rightarrow \quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}\,, \quad (n = 0, 1, 2, \ldots)\,.$$

Die rekursive Bauart des Newton–Verfahrens (dieselbe Prozedur wird Schritt für Schritt wiederholt, wenn auch mit anderen Zahlen) eignet sich hervorragend zur Programmierung.

Zusammenfassung: Newton–Verfahren zur Lösung der Gleichung $f(x) = 0$

Eine Näherung für die Lösung $\bar{x}$ der Gleichung $f(x) = 0$ läßt sich mit dem Newton – Verfahren in folgenden Schritten bestimmen:

- Festlegung eines Intervalls $I = [a,b]$ mit $\bar{x} \in I$
  (z.B. durch die Forderung $f(a) \cdot f(b) < 0$ bei stetigem $f(x)$).
- Wahl eines Startwerts $x_0 \in I$. Berechnung der Folge $\{x_n\}$ von Näherungswerten mit der Rekursionsformel (Iterationsvorschrift)

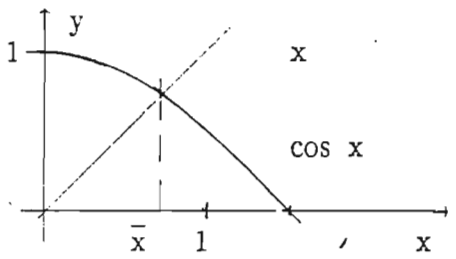
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad , \quad (n = 0, 1, 2, \ldots) .$$

- Abbruch der Berechnung, wenn eine geforderte Genauigkeit zu erwarten ist. Man wählt dazu i.a. $|x_{n+1} - x_n| < \epsilon$ oder $|f(x_n)| < \epsilon$; ϵ ist eine vom Problem abhängige Fehlerschranke.

Beispiele:

1) Nullstelle von $f(x) = x - \cos x$
   (siehe Bsp. 2.1 b)

Es gibt genau eine Lösung (vgl Skizze) in der Nähe von $x_0 = 0.7$ (Startwert).



Iterationsvorschrift:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n - \cos x_n}{1 + \sin x_n} \quad , \quad (n = 0, 1, 2, \ldots) .$$

Tabelle für die Ergebnisse des Newton–Verfahrens:

n	x_n	$x_n - x_{n-1}$	$f(x_n)$
0	0.7		0.064842187
1	0.739436498	0.039436498	– 0.000588094
2	0.739085161	– 0.000351337	– 0.000000046
3	0.739085133	– 0.000000027	0.000000000
4	0.739085133	0.000000000	0.000000000

Das Verfahren liefert auf 9 Dezimalstellen genau $\bar{x} = 0.739085133$.

An den beiden letzten Spalten der Tabelle kann man die schnelle Konvergenz des Newtonverfahrens erkennen (vgl. auch 2.4 d).

2) Beispiel zum Zinsproblem (siehe 2.1 a):

Ein Kredit für ein Auto in Höhe von $K = 20\,000$ DM wird für eine Laufzeit von $m = 36$ Monaten und einer Monatsrate von $r = 589$ DM angeboten. Wie hoch ist der entsprechende Zinssatz ?

Berechnung mit dem Newtonverfahren: Nach 2.1 gilt

$$f(x) = K\cdot x^{m+1} - (K + r)\cdot x^m + r \overset{!}{=} 0$$

$$f'(x) = K\cdot(m+1)\cdot x^m - (K + r)\cdot m\cdot x^{m-1}$$

Als Startwert für das Newtonverfahren $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ wählen wir etwa $x_0 = 1.01$. Dann ergeben sich folgende Iterationsverbesserungen:

$$x_1 = 1.0063051 \quad , \quad x_2 = 1.0043195 \quad , \quad x_3 = 1.0034453$$

$$x_4 = 1.0032129 \quad , \quad x_5 = 1.0031947 \quad , \quad x_6 = 1.0031946 = x_7$$

Der Zinssatz beträgt dann

$$p = (x - 1)\cdot 1200 = 3{,}83\ \% \ .$$

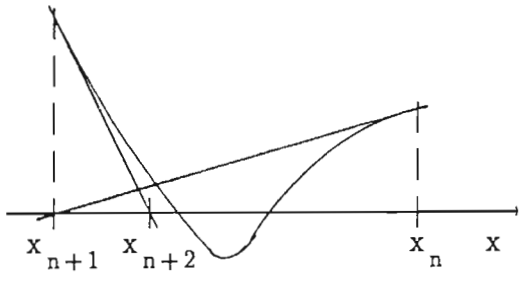
Bemerkungen:

Das Newton–Verfahren konvergiert nicht immer problemlos gegen die gesuchte Nullstelle $\bar{x}$.

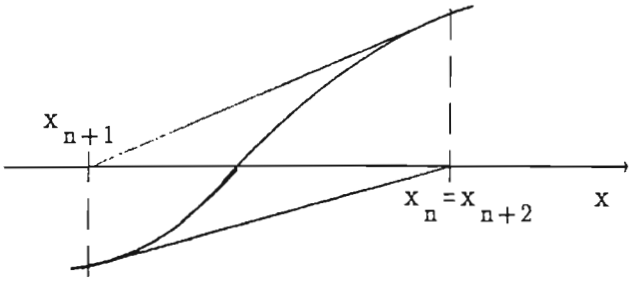
Beispiele dafür sind:

- Es darf kein Punkt $(x_n \mid f(x_n)$ mit waagrechter Tangente auftreten $(f'(x_n)) \neq 0)$.
- Hat f(x) mehrere Nullstellen, so kann bei schlechtem Startwert Konvergenz zu einer Nachbarnullstelle eintreten (Skizze a)).
- Für $x_{n+2} = x_n$ konvergiert das Verfahren nicht (Skizze b)).

Skizze a)



Skizze b)



Die Probleme lassen sich oft durch Änderung des Startwerts lösen.

2.3. Das Sehnenverfahren (regula falsi)

Gesucht ist wieder eine Nullstelle einer Funktion f(x) . Man ersetzt bei diesem Verfahren die Kurve f(x) zwischen 2 Punkten $P_n(x_n|f(x_n))$ und $P_{n+1}(x_{n+1}|f(x_{n+1}))$ durch die Sehne P_nP_{n+1} und berechnet deren Schnittstelle x_{n+2} mit der x–Achse. Als Startpunkte (n=0) P_0 und P_1 wählt man zweckmäßig Punkte mit $f(x_0)\cdot f(x_1) < 0$.

Herleitung der Iterationsformel: Gleichung der Sehne $\overline{P_nP_{n+1}}$:

Mit

$y_k = f(x_k)$ gilt $\qquad \dfrac{y - y_n}{x - x_n} = \dfrac{y_{n+1} - y_n}{x_{n+1} - x_n}$ $\qquad$ oder

$$y - y_n = \frac{y_{n+1} - y_n}{x_{n+1} - x_n}\cdot(x - x_n) \quad .$$

Schnitt mit der x–Achse ($y = 0$) ergibt

$$x = x_{n+2} = x_n - y_n\cdot\frac{x_{n+1} - x_n}{y_{n+1} - y_n} = \frac{x_n y_{n+1} - x_{n+1}y_n}{y_{n+1} - y_n} \quad .$$

Es ist darauf zu achten, daß keine Folgepunkte mit gleichem y auftreten ($y_{n+1} \neq y_n$) . Das Verfahren wird abgebrochen, wenn $|x_{n+2} - x_{n+1}|$ kleiner als eine bestimmte Fehlerschranke wird.

Beispiel: $\qquad f(x) = x - \cos x = 0$.

Als Startpunkt wählen wir etwa $P_0(0.5|-0.37758)$ und $P_1(1|0.45970)$.

Das Verfahren lautet dann

$$x_{n+2} = \frac{x_n\cdot(x_{n+1} - \cos x_{n+1}) - x_{n+1}\cdot(x_n - \cos x_n)}{x_{n+1} - \cos x_{n+1} - x_n + \cos x_n} \quad .$$

Für n = 0 , 1 , 2 , ... erhält man dann aus den obigen Startwerten:

$x_2 = 0.725\,481\,587$

$x_3 = 0.738\,398\,620$

$x_4 = 0.739\,087\,211$

$x_5 = 0.739\,085\,133 = x_6$.

2.4. Bemerkungen

a) **Allgemeine Formulierung eines Iterationsverfahrens:**

Zu einer "Iterationsfunktion" $\varphi(x)$ definiert man das Verfahren

$$x_{n+1} = \varphi(x_n) .$$

Dieses konvergiert unter bestimmten Voraussetzungen gegen einen "Fixpunkt" $\bar{x}$ mit $\bar{x} = \varphi(\bar{x})$.

Beim Newton–Verfahren ergibt sich etwa $\varphi(x) = x - \frac{f(x)}{f'(x)}$. Iterationsverfahren lassen sich ganz allgemein definieren und auf Konvergenz untersuchen (vgl. Literatur).

b) **Fehleruntersuchung bei Iterationsverfahren** $x_{n+1} = \varphi(x_n)$:

Der Fehler der n–ten Näherung x_n zur Lösung $\bar{x}$ sei $\Delta x_n = x_n - \bar{x}$. Gesucht sind Fehlerabschätzungen der Form $|\Delta x_n| < c$.

Für das Newton–Verfahren läßt sich etwa zeigen (vgl. [10]):

Es sei f(x) im Nullstellenintervall [a,b] zweimal stetig differenzierbar und es existieren dort folgende Schranken für die ersten beiden Ableitungen

- $|f'(x)| \geq m > 0$ (d.h. einfache Nullstelle) und
- $|f''(x)| \leq M$.

Dann gelten für $n = 1, 2, 3, \ldots$ und $K = \frac{M}{2m}$ die Abschätzungen

(1) $|\Delta x_n| \leq K \cdot (x_n - x_{n-1})^2$ (im Nachhinein – a posteriori)

(2) $|\Delta x_n| \leq \frac{1}{K} \cdot [(b-a)K]^{2^n}$ (von vornherein – a priori).

Interessant ist bei Abschätzung (2), daß eine Fehleraussage für die n–te Iteration x_n bereits möglich ist, bevor überhaupt eine Iteration berechnet worden ist ("von vornherein").

Rechnerspezifische Fehlerquellen (z.B. Rundungsfehler) sind bei diesen Abschätzungen nicht berücksichtigt.

c) Für das praktische Rechnen an Computern geht man i.a. so vor:

Man bricht die Iteration ab, wenn im Rahmen einer vorzugebenden Genauigkeit der Fehler zwischen 2 Iterationen vernachlässigbar ist:

$$|x_{n+1} - x_n| < \epsilon ,$$

z. B. beim Rechnen auf 6 Dezimalstellen $|x_{n+1} - x_n| < 5 \cdot 10^{-6}$.

Dies garantiert allerdings nicht, daß das Ergebnis im Rahmen dieser Genauigkeit stimmt (vgl. dazu das Beispiel in d)).

d) **Konvergenzordnung eines Iterationsverfahrens:**

Berechnet man beim Beispiel $f(x) = x - \cos x$ in 2.2 die Korrekturen $\delta_n = |x_n - x_{n-1}|$, so ergibt sich

$\delta_1 = |x_1 - x_0| = 0.04$, $\delta_2 = 0.000\,4$, $\delta_3 = 0.000\,000\,03$ usw.

Offensichtlich zeigen die δ_n in der Umgebung der Lösung quadratisches Abklingverhalten

$\delta_{n+1} \approx c \cdot \delta_n^2$.

Definition: **Konvergenzordnung eines Iterationsverfahrens:**

Gilt für kleine $|\Delta x_n|$ (d.h. in der Umgebung einer Lösung $\bar{x}$)

$$|\Delta x_{n+1}| \leq c \cdot |\Delta x_n|^p ,$$

so heißt das Iterationsverfahren konvergent von der Ordnung p.

Je größer p ist, umso schneller konvergiert das Verfahren.

Als **quadratische Konvergenz** bezeichnet man den Fall $p = 2$,

d.h. für kleine $|\Delta x_n|$ gilt die Beziehung

$$|\Delta x_{n+1}| \leq c \cdot |\Delta x_n|^2 .$$

Unter **linearer Konvergenz** versteht man den Fall $p = 1$, $c < 1$.

Für das Newton–Verfahren läßt sich zeigen

Satz: Das Newton–Verfahren konvergiert in der Umgebung einer einfachen Nullstelle quadratisch. In mehrfachen Nullstellen konvergiert das Newton–Verfahren nur noch linear. Die Geschwindigkeit des Verfahrens nimmt dabei mit wachsender Vielfachheit der Nullstelle ab.

Beispiel: $f(x) = \frac{\pi}{2} - x - \cos x$ besitzt für $x = \frac{\pi}{2}$ eine dreifache Nullstelle. Versucht man diese mit dem Newton–Verfahren und Startwert $x_0 = 1.5$ zu berechnen, so ergibt sich

$x_1 = 1.523\,602\,668$, $x_2 = 1.539\,335\,360$,

$x_3 = 1.549\,823\,272$, $x_4 = 1.556\,816\,521$,

$x_5 = 1.561\,483\,111$, $x_6 = 1.564\,596\,034$,

$x_7 = 1.566\,677\,015$, $x_8 = 1.568\,091\,375$,

$x_9 = 1.569\,184\,749$, $x_{10} = x_{11} = 1.569\,954\,809$.

Man sieht deutlich die langsame Konvergenz des Verfahrens. Zudem bewirken Rundungsfehler, daß das Iterationsergebnis x_{10} (bei Rechnen auf 9 Dezimalstellen) nur in 3 Stellen mit dem exakten Wert $\frac{\pi}{2} = 1.570\,796$ übereinstimmt. Dies liegt daran, daß der Rechner im Rahmen seiner Stellenzahl schon bei x_{10} den Funktionswert als 0 interpretiert.

Numerisch zeigt eine Funktion bereits dann ein zu mehrfachen Nullstellen vergleichbares Verhalten, wenn die Tangente in einer Nullstelle sehr flach verläuft ("schleifende" Nullstelle).

2.5. Algebraische Gleichungen

Algebraische Gleichung vom Grad n:

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0 = 0 \ .$$

Beispiel: n = 2 ... Quadratische Gleichung.

Jede algebraische Gleichung n – ten Grades besitzt "im algebraischen Sinn" genau n Lösungen (d.h. reelle und komplexe Nullstellen sind zu zählen, mehrfache Nullstellen sind entsprechend ihrer Vielfachheit zu berücksichtigen).

Algebraische Gleichungen mit reellen Koeffizienten a_k:

Komplexe Lösungen treten stets als konjugiert komplexe Paare $x_{1,2} = u \pm jv$ (u, v reell) auf.

Numerische Berechnung von Lösungen algebraischer Gleichungen:

Für das Newton–Verfahren läßt sich ein vereinfachter Algorithmus auf der Basis des (erweiterten) Horner–Schemas angeben, der reelle und komplexe Lösungen liefert (bei komplexen Lösungen ist ein komplexer Startwert erforderlich).

Zur Berechnung komplexer Lösungen wird meist eine Variante des Newton–Verfahrens verwendet, das Bairstow–Verfahren. Faßt man bei einer komplexen Lösung die Faktorisierungsterme für ein Paar konjugiert komplexer Lösungen $x_1, x_2 = x_1^*$ zusammen, so entstehen quadratische Faktoren mit reellen Koeffizienten b und c :

$$(x - x_1) \cdot (x - x_1^*) = x^2 + bx + c \ .$$

Das Bairstow–Verfahren ermöglicht es, numerisch die Koeffizienten b und c solcher quadratischer Faktoren zu berechnen.

Die entsprechenden Verfahren und Algorithmen entnehme man der Literatur.

3. Lineare Gleichungssysteme (LGS)

3.1. Problem und Lösungstheorie

System von m linearen Gleichungen für n Unbekannte $x_1, x_2, \dots, x_n$

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n &= b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \dots + a_{3n}x_n &= b_3 \\
\dots & \\
a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \dots + a_{mn}x_n &= b_m
\end{aligned}
$$

Wir beschränken uns im folgenden auf reelle a_{ij}, b_i
(prinzipiell können auch komplexe Größen zugelassen werden).

Matrizentheoretische Formulierung: $\underline{A} \cdot \underline{x} = \underline{b}$ mit

$$
\underline{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \cdots a_{1n} \\ a_{21} & a_{22} & a_{23} \cdots a_{2n} \\ \dots & & \\ a_{m1} & a_{m2} & a_{m3} \cdots a_{mn} \end{bmatrix}, \quad \underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{bmatrix}.
$$

$\underline{A}$ heißt Koeffizientenmatrix. Als erweiterte Matrix $\underline{A}_e$ bezeichnet man

$$
\underline{A}_e = [\underline{A}, \underline{b}] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} \cdots & a_{2n} & b_2 \\ \dots & & & & \dots \\ a_{m1} & a_{m2} & a_{m3} \cdots & a_{mn} & b_m \end{array}\right].
$$

Für $\underline{b} \neq \underline{0} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \end{bmatrix}$ heißt ein LGS inhomogen, für $\underline{b} = \underline{0}$ homogen.

Ein homogenes LGS besitzt stets die triviale Lösung $\underline{x} = \underline{0}$.

Im Fall $m = n$ (d.h. Anzahl der Variablen = Anzahl der Gleichungen) besitzt das LGS $\underline{A} \cdot \underline{x} = \underline{b}$ genau dann eine eindeutige Lösung, wenn die Determinante der Koeffizientenmatrix $\underline{A}$ von 0 verschieden ist: $\det \underline{A} \neq 0$. In diesem Fall heißt $\underline{A}$ **regulär**.

Die allgemeine Lösungstheorie linearer Gleichungssysteme entnehme man der Literatur über Lineare Algebra.

Bei den meisten Anwendungsproblemen treten vor allem folgende Fälle auf:

a) m = n , inhomogen , det $\underline{A} \neq 0$ (A regulär, eindeutig lösbares LGS).

Gesucht sind Algorithmen zur Berechnung der Lösung.

b) m > n , d.h. mehr Gleichungen als Unbekannte, z.B. durch Kontrollmessungen.

Beispiel: Messung der Winkel in einem Dreieck

$$\alpha = 32{,}3^{\circ} \qquad (1)$$

$$\beta = 67{,}9^{\circ} \qquad (2)$$

$$\gamma = 80{,}1^{\circ} \qquad (3)$$

$$\alpha + \beta + \gamma = 180^{\circ} \qquad (4) \ .$$

(4) ist eine exakte Beziehung (Winkelsumme im Dreieck), eine der Messungen (1) bis (3) war theoretisch unnötig und hat die Eigenschaft einer Kontrollmessung. Das LGS ist überbestimmt. Der Versuch, ein überbestimmtes LGS exakt zu lösen, führt in der Regel zu einem Widerspruch (siehe obiges Beispiel).

Ziel: Man entwickle einen Algorithmus, der eine Lösung liefert, die allen Gleichungen "im Mittel" am besten genügt. Ein solches Verfahren wird als Fehlerausgleich bezeichnet (siehe 4.).

3.2. Der Gauß – Algorithmus

Wir beschränken uns im folgenden auf eine reguläre quadratische Koeffizientenmatrix $\underline{A}$ (d.h. $\det \underline{A} \neq 0$, eindeutig lösbares LGS).

3.2.1 Der Sonderfall einer Dreiecksmatrix

Fall 1: $\underline{A}$ sei speziell eine **obere (rechte) Dreiecksmatrix** $\underline{R}$, d.h.

$$\underline{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ 0 & r_{22} & r_{23} & \cdots & r_{2n} \\ \cdots & & & & \\ 0 & 0 & r_{k3} & \cdots & r_{kn} \\ \cdots & & & & \\ 0 & 0 & 0 & \cdots & r_{nn} \end{bmatrix} .$$

Das entsprechende LGS lautet dann

$$\begin{aligned} r_{11}x_1 + r_{12}x_2 + r_{13}x_3 + \ldots + r_{1n}x_n &= b_1 \\ r_{22}x_2 + r_{23}x_3 + \ldots + r_{2n}x_n &= b_2 \\ r_{33}x_3 + \ldots + r_{3n}x_n &= b_3 \\ \ldots & \\ r_{nn}x_n &= b_n \end{aligned}$$

Nach Voraussetzung ist

$$\det \underline{R} = r_{11}r_{22}r_{33}\ldots r_{nn} \neq 0 \quad ,$$

d.h. jedes Element r_{kk} der Hauptdiagonale ist von 0 verschieden. Die Lösungskomponenten x_k lassen sich jetzt durch Rückwärtseinsetzen berechnen (die auftretenden Nenner r_{kk} sind von 0 verschieden):

(n) $$x_n = \frac{b_n}{r_{nn}}$$

(n–1) $$x_{n-1} = \frac{1}{r_{n-1,n-1}}(b_{n-1} - r_{n-1,n}x_n)$$ (x_n aus (n))

...

(k) $$x_k = \frac{1}{r_{kk}}\left(b_k - \sum_{i=k+1}^{n} r_{ki}x_i\right)$$

...

(1) $$x_1 = \frac{1}{r_{11}}\left(b_1 - \sum_{i=2}^{n} r_{1i}x_i\right)$$

Beispiel:
$$\underline{R} = \begin{bmatrix} -1 & 1 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & 2 \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} 6 \\ 1 \\ 6 \end{bmatrix}.$$

LGS :
$$\begin{aligned} -x + y + z &= 6 \quad \rightarrow \quad x = -6 + y + z = -1 \\ -y + z &= 1 \quad \rightarrow \quad y = -1 + z = 2 \quad \uparrow \\ 2z &= 6 \quad \rightarrow \quad z = 3 \quad \uparrow \end{aligned}$$

Fall 2: $\underline{A}$ sei eine **untere (linke) Dreiecksmatrix** $\underline{L}$, d.h.

$$\underline{L} = \begin{bmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ \dots & & & & \\ l_{k1} & l_{k2} & l_{k3} & \dots & 0 \\ \dots & & & & \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{bmatrix}$$

Das entsprechende LGS

$$\begin{aligned} l_{11}x_1 &= b_1 \\ l_{21}x_1 + l_{22}x_2 &= b_2 \\ l_{31}x_1 + l_{32}x_2 + l_{33}x_3 &= b_3 \\ \dots & \\ l_{n1}x_1 + l_{n2}x_2 + l_{n3}x_3 + \dots \; l_{nn}x_n &= b_n \end{aligned}$$

läßt sich jetzt durch Vorwärtseinsetzen unmittelbar lösen.

3.2.2. Der Algorithmus zur Elimination

Ziel des Gauß – Algorithmus ist eine systematische Elimination der Variablen.

Das LGS $\underline{A}\cdot\underline{x} = \underline{b}$ wird dabei in ein LGS mit Dreiecksmatrix umgeformt (Lösung siehe 3.1).

Grundlage des Gauß – Algorithmus ist folgender Sachverhalt

(Additionsverfahren zur Elimination von Variablen):

Die Lösung eines LGS ändert sich nicht, wenn

- eine Gleichung zu einer anderen addiert wird,
- eine Gleichung mit einem konstanten Faktor multipliziert wird.

Die Elimination wird an der erweiterten Matrix

$$\underline{A}_e = [\,\underline{A}\,,\,\underline{b}\,] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} \cdots & a_{2n} & b_2 \\ \cdots & & & & \cdots \\ a_{n1} & a_{n2} & a_{n3} \cdots & a_{nn} & b_n \end{array}\right]$$

durchgeführt. Jede Zeile entspricht einer Gleichung des LGS.

Üblich ist Vorwärtselimination (d.h. sukzessive Elimination von $x_1, x_2, x_3, \ldots$) und

Umformung zu einer oberen Dreiecksmatrix $\underline{R}$.

Schritt1: Elimination von x_1

Man multipliziert dazu Zeile 1 (Gleichung (1)) mit $(-\frac{a_{21}}{a_{11}})$ und addiert das Resultat zu Zeile 2:

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \cdots & & & & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right] \cdot(-\frac{a_{21}}{a_{11}}) \qquad (a_{11} \neq 0)$$

$$\rightarrow \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}-\frac{a_{21}a_{12}}{a_{11}} & \cdots & a_{2n}-\frac{a_{21}a_{1n}}{a_{11}} & b_2-\frac{a_{21}b_1}{a_{11}} \\ \cdots & & & & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right].$$

Voraussetzung ist dabei $a_{11} \neq 0$. Dies ist keine Einschränkung. Im Fall $a_{11} = 0$ wird die 1. Zeile mit einer (sicher vorhandenen) k–ten Zeile mit $a_{k1} \neq 0$ vertauscht , die Zeilen werden dann neu nummeriert.

Für jede weitere Zeile erfolgt die Elimination von x_1 analog. Um allgemein x_1 aus der i–ten Zeile zu eliminieren, muß Zeile 1 mit $(-\frac{a_{i1}}{a_{11}})$ multipliziert und zu Zeile i addiert werden. Das Ergebnis ist dann folgende Umformung

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \cdots & & & & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right] \rightarrow \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & \bar{a}_{22} & \cdots & \bar{a}_{2n} & \bar{b}_2 \\ \cdots & & & & \cdots \\ 0 & \bar{a}_{n2} & \cdots & \bar{a}_{nn} & \bar{b}_n \end{array}\right] ;$$

die $\bar{a}_{ik}$ bzw. $\bar{b}_i$ berechnen sich wie oben für Zeile 2 beschrieben.

Im folgenden verwenden wir für $\bar{a}_{ik}$ bzw. $\bar{b}_i$ wieder die alten Bezeichnungen a_{ik} bzw. b_i (dies entspricht einem Überschreiben der alten Bezeichner).

Definition: Das Element a_{11} $(\neq 0)$ heißt 1. Pivotelement, Zeile 1 nennt man 1. Pivotzeile.

Die 1. Pivotzeile darf für die weiteren Eliminationen nicht mehr verwendet werden (sonst käme ja x_1 wieder dazu). Sie wird 1. Zeile der gesuchten Dreiecksmatrix $\underline{R}$.

Schritt 2: Elimination von x_2

Wir verwenden wieder die alten Bezeichnungen (z.B. a_{22} für $\bar{a}_{22}$).

Zweites Pivotelement wird a_{22} $(\neq 0$, sonst Zeilen vertauschen), die 2. Zeile wird neue Pivotzeile.

Multiplikation der 2. Pivotzeile mit $(-\frac{a_{i2}}{a_{22}})$ und Addition zur i–ten Zeile ergibt

(bei Überschreiben der bisherigen Koeffizienten a_{ij}, b_i)

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & \cdots a_{1n} & b_1 \\ 0 & a_{22} & a_{23} & \cdots a_{2n} & b_2 \\ 0 & 0 & a_{33} & \cdots a_{3n} & b_3 \\ \cdots & & & & \\ 0 & 0 & a_{n3} & \cdots a_{nn} & b_n \end{array}\right] \qquad \text{usw.}$$

k–ter Schritt:

Die k–te Zeile wird Pivotzeile, Pivotelement ist $a_{kk} \neq 0$ (gegebenenfalls Zeilen vertauschen).

Die k–te Zeile wird mit $(-\frac{a_{ik}}{a_{kk}})$ multipliziert und zur i–ten Zeile addiert $(1+k \leq i \leq n)$.

Bei regulärer n,n–Matrix $\underline{A}$ führen so (n–1) Eliminationsschritte zu einer oberen (rechten)

Dreiecksmatrix $\underline{R}$:

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \cdots & & & & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right] \rightarrow \left[\begin{array}{ccccccc|c} r_{11} & r_{12} & & \cdots & & & r_{1n} & c_1 \\ 0 & r_{22} & & \cdots & & & r_{2n} & c_2 \\ 0 & 0 & r_{33} & & & \cdots & r_{3n} & c_3 \\ 0 & 0 & \cdots & 0 & r_{kk} & \cdots & r_{kn} & c_k \\ \cdots & & & & & & & \\ 0 & 0 & 0 & \cdots & 0 & & r_{nn} & c_n \end{array}\right]$$

Diese liefert durch Rückwärtseinsetzen (vgl. 3.2.1) die Lösung des linearen Gleichungssystems.

Beispiel:

$$\begin{aligned} x_1 + x_2 + x_3 &= -1 \\ -x_1 + x_2 + x_3 &= 1 \\ 2x_1 + x_2 - x_3 &= 2 \end{aligned}$$

Darstellung als erweiterte Matrix und Gauß – Algorithmus:

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & 1 \\ 2 & 1 & -1 & 2 \end{array}\right] \rightarrow \left[\begin{array}{ccc|c} 1 & 1 & 1 & -1 \\ 0 & 2 & 2 & 0 \\ 0 & -1 & -3 & 4 \end{array}\right]$$

(Schritt 1: Multiplikation von Zeile 1 mit 1 und Addition zu Zeile 2

Multiplikation von Zeile 1 mit –2 und Addition zu Zeile 3)

$$\rightarrow \left[\begin{array}{ccc|c} 1 & 1 & 1 & -1 \\ 0 & 2 & 2 & 0 \\ 0 & 0 & -2 & 4 \end{array}\right] \rightarrow [\,\underline{R}\,,\underline{c}\,]$$

(Schritt2: Multiplikation von Zeile 2 mit $\frac{1}{2}$ und Addition zu Zeile 3)

Rückwärtseinsetzen ergibt:

$$\begin{aligned} -2x_3 = 4 \quad &\rightarrow \quad x_3 = -2 \\ 2x_2 + 2x_3 = 0 \quad &\rightarrow \quad x_2 = -x_3 = 2 \\ x_1 + x_2 + x_3 = -1 \quad &\rightarrow \quad x_1 = -1 - x_2 - x_3 = -1 \end{aligned}$$

Bemerkung: Manchmal wird auch in umgekehrter Reihenfolge

$x_n, x_{n-1}, \ldots$ eliminiert (Rückwärtselimination). Man erhält dann eine untere (linke) Dreiecksmatrix $\underline{L}$; die Lösung ergibt sich dann durch Vorwärtseinsetzen.

Beispiel (s.o.): $\left[\begin{array}{ccc|c} 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & 1 \\ 2 & 1 & -1 & 2 \end{array}\right] \rightarrow \left[\begin{array}{ccc|c} 3 & 2 & 0 & 1 \\ 1 & 2 & 0 & 3 \\ 2 & 1 & -1 & 2 \end{array}\right]$

(die 3. Zeile wird zur 2. bzw. zur 1. Zeile addiert)

$$\rightarrow \left[\begin{array}{ccc|c} 2 & 0 & 0 & -2 \\ 1 & 2 & 0 & 3 \\ 2 & 1 & -1 & 2 \end{array}\right] = [\,\underline{L}\,,\,\underline{d}\,]$$

(die 2. Zeile wird mit (–1) multipliziert und zur 1. Zeile addiert).

Vorwärtseinsetzen ergibt:

$$2x_1 = -2 \qquad x_1 = -1$$

$$x_1 + 2x_2 = 3 \qquad x_2 = (3 - x_1)/2 = 2$$

$$2x_1 + x_2 - x_3 = 2 \qquad x_3 = -2 + 2x_1 + x_2 = -2$$

3.2.3. Die Wahl des Pivotelements

Es sei a_{kk} das beim k–ten Eliminationsschritt verwendete Pivotelement (wie vereinbart werden bei jeder Elimination die Matrixelemente mit den neuen Werten überschrieben). Theoretisch genügt für ein Pivotelement a_{kk} die Bedingung $a_{kk} \neq 0$. Am folgenden Beispiel soll gezeigt werden, daß die Wahl des Pivotelements wegen Rundungsfehlern zu falschen Ergebnissen führen kann.

Annahme: Gleitpunktrechnung mit 4 signifikanten Stellen

$$1.000 \cdot 10^{-4} x + 1.000\, y = 1.000$$

$$1.000\, x + 1.000\, y = 2.000$$

Bei "exakter" Rechnung ergibt sich $x = 1.00010$, $y = 0.99990$.

Gauß – Algorithmus für die vorgegebene Reihenfolge der Gleichungen:

$$\left[\begin{array}{ccc|c} 1.000\ \text{E–4} & 1.000 & 1.000 \\ 1.000 & 1.000 & 2.000 \end{array}\right] \rightarrow \left[\begin{array}{ccc|c} 1.000\ \text{E–4} & 1.000 & 1.000 \\ 0.000 & -1.000\ \text{E4} & -1.000\ \text{E4} \end{array}\right]$$

(wegen $-1.000\ \text{E4} + 1.000 = (-1.000 + 0.000_1)\ \text{E4} = -1.000\ \text{E4}$.)

Rückwärtseinsetzen ergibt $y = 1.000$, $x = 0.000$; das Ergebnis für x ist offensichtlich falsch.

Vertauscht man die Gleichungen und wendet den Gauß – Algorithmus an, so ergibt sich

$$\left[\begin{array}{ccc} 1.000 & 1.000 & 2.000 \\ 1.000\ \text{E–4} & 1.000 & 1.000 \end{array}\right] \rightarrow \left[\begin{array}{ccc} 1.000 & 1.000 & 2.000 \\ 0.000 & 1.000 & 1.000 \end{array}\right]$$

(wegen $-1.000\ \text{E–4} + 1.000 = (1.000 - 0.000_1)\ \text{E0} = 1.000$

bzw. $-2.000\ \text{E–4} + 1.000 = (1.000 - 0.000_2)\ \text{E0} = 1.000$)

Rückwärtseinsetzen ergibt jetzt $y = 1.000$, $x = 1.000$

(korrektes Ergebnis im Rahmen von 4 Stellen).

Offensichtlich liefert der Gauß – Algorithmus hier beim betragsgrößten Pivotelement das richtige Ergebnis. Dies läßt sich auch allgemein nachweisen.

Teilpivotisierung oder Spaltenpivotsuche:

Man suche im k–ten Eliminationsschritt das Element $\max\limits_{k \le j \le n} |a_{jk}|$. Dieses soll neues Pivotelement werden (und an die Position von a_{kk} gesetzt werden). Dazu ist gegebenenfalls die k–te Zeile mit der Zeile, in der das Maximum angenommen wird, zu vertauschen. Das Pivotelement a_{kk} ist jetzt das betragsgrößte Element der für den k–ten Eliminationsschritt in Frage kommenden Spaltenelemente. Die Lösung ist dann bei "gutartigen" (gut konditionierten, siehe 3.2.4) linearen Gleichungssystemen rundungsfehlerstabil.

3.2.4. Kondition eines linearen Gleichungssystems

Definition: Ein LGS heißt gut (bzw. schlecht) konditioniert, wenn kleine Fehler in den Koeffizienten von $\underline{A}$ und $\underline{b}$ zu kleinen (bzw. großen) Fehlern in der Lösung $\underline{x}$ führen.

Beispiel für ein schlecht konditioniertes LGS:

$$1.00\,x + 0.99\,y = 1.99$$

$$0.99\,x + 0.98\,y = 1.97$$

Offensichtlich ist das LGS eindeutig lösbar ($\det \underline{A} \neq 0$) und besitzt die Lösung $x = y = 1.00$.

Das geringfügig veränderte LGS (stimmt auf 2 Stellen mit dem vorherigen überein)

$$1.000\,x + 0.990\,y = 1.993$$

$$0.990\,x + 0.980\,y = 1.967 \quad ,$$

hat die Lösung $x = -58.1$, $y = 60.7$.

Bedingungen dafür, wann ein LGS gut bzw. schlecht konditioniert ist, entnehme man der Literatur.

3.3. Weitere direkte Lösungsverfahren bei regulärer n,n–Matrix $\underline{A}$

3.3.1. Lösung eines LGS durch Zerlegung von A in Dreiecksmatrizen

Der Gauß – Algorithmus liefert bei üblicher Vorwärtselimination eine obere (rechte) Dreiecksmatrix $\underline{R}$. Zu dieser läßt sich eine "korrespondierende" linke Dreiecksmatrix $\underline{L}$ bestimmen mit $\underline{L} \cdot \underline{R} = \underline{A}$.

($\underline{L}$ hat mit der durch Rückwärtselimination entstehenden unteren Dreiecksmatrix nichts zu tun !).

In der Hauptdiagonalen von $\underline{L}$ stehen dabei nur die Zahlen 1, zur Bestimmung der anderen Elemente von $\underline{L}$ gibt es verschiedene Algorithmen.

Beispiel: $\underline{A} \cdot \underline{x} = \underline{b}$ mit $\underline{A} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ 2 & 1 & -1 \end{bmatrix}$, $\underline{b} = \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix}$.

Der Gauß – Algorithmus liefert bei Vorwärtselimination (s.o.)

$\underline{R} \cdot \underline{x} = \underline{c}$ mit $\underline{R} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 2 \\ 0 & 0 & -2 \end{bmatrix}$ und $\underline{c} = \begin{bmatrix} -1 \\ 0 \\ 4 \end{bmatrix}$.

Macht man nun für $\underline{L}$ den Ansatz $\underline{L} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}$ und berechnet $\underline{L} \cdot \underline{R} = \underline{A}$, so lassen sich die fehlenden l_{ik} bestimmen und man erhält $\underline{L} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & -1/2 & 1 \end{bmatrix}$.

Bemerkung: Man kann $\underline{L}$ direkt aus den Multiplikatoren des Gauß – Algorithmus' bestimmen.

Es gilt $l_{ik} = a_{ik}/a_{kk} \quad (k+1 \leq i \leq n)$,

dabei sind a_{ik} und a_{kk} die Matrixkoeffizienten zu Beginn des k–ten Eliminationsschritts.

Weiter gilt $\det \underline{A} = \det \underline{L} \cdot \det \underline{R} = \det \underline{R} = (-1)^j \cdot r_{11} \cdot r_{22} \cdot \ldots \cdot r_{nn}$;

(j ist die Anzahl der erfolgten Zeilenvertauschungen.)

Beispiel (s.o.):

$i = 2 , k = 1 \quad \rightarrow l_{21} = a_{21}/a_{22} = -1/1 = -1$

$i = 3 , k = 1 \quad \rightarrow l_{31} = a_{31}/a_{22} = 2/1 = 2$

$i = 3 , k = 2$ (relevant ist jetzt die Matrix $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 2 \\ 0 & -1 & -3 \end{bmatrix}$)

$\rightarrow l_{32} = a_{32}/a_{22} = -\frac{1}{2}$.

Anwendung der LR – Zerlegung: Hat man zu einem vorgegebenen LGS $\underline{A}\cdot\underline{x} = \underline{b}$ die Matrizen $\underline{R}$ und $\underline{L}$ berechnet, so kann man leicht für beliebige rechte Seiten $\underline{b}$ die Lösungen $\underline{x}$ bestimmen.

Aus $\underline{A}\cdot\underline{x} = \underline{b}$ und $\underline{A} = \underline{L}\cdot\underline{R}$ folgt $\underline{L}\cdot(\underline{R}\cdot\underline{x}) = \underline{b}$.

Der Ansatz $\underline{y} = \underline{R}\cdot\underline{x}$ für Hilfsvariable $\underline{y}$ liefert 2 einfache lineare Gleichungssysteme mit Dreiecksmatrizen

$$\underline{L}\cdot\underline{y} = \underline{b} \quad (1) \qquad \text{und} \qquad \underline{R}\cdot\underline{x} = \underline{y} \quad (2).$$

Die Lösung von (1) erhält man durch Vorwärtseinsetzen, die von (2) durch Rückwärtseinsetzen.

Beispiel: $\underline{A}\cdot\underline{x} = \begin{bmatrix} 6 \\ 4 \\ 1 \end{bmatrix}$ mit $\underline{A} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ 2 & 1 & -1 \end{bmatrix}$ (s.o.).

(1) $\underline{L}\cdot\underline{y} = \underline{b}$ $\rightarrow$ $\left[\begin{array}{ccc|c} 1 & 0 & 0 & 6 \\ -1 & 1 & 0 & 4 \\ 2 & -1/2 & 1 & 1 \end{array}\right]$ $\rightarrow$ $\begin{aligned} y_1 &= 6 \\ y_2 &= 4 + y_1 = 10 \\ y_3 &= 1 - 2y_1 + \tfrac{1}{2}y_2 = -6 \end{aligned}$

(2) $\underline{R}\cdot\underline{x} = \underline{y}$ $\rightarrow$ $\left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 2 & 2 & 10 \\ 0 & 0 & -2 & -6 \end{array}\right]$ $\rightarrow$ $\begin{aligned} x_1 &= 6 - x_2 - x_3 = 1 \\ x_2 &= 5 - x_3 = 2 \\ -2x_3 &= -6 \rightarrow x_3 = 3 \end{aligned}$

3.3.2 Die Choleski – Zerlegung

Im Sonderfall einer symmetrischen und positiv definiten Matrix $\underline{A}$ (d.h. $\underline{x}^T\cdot\underline{A}\cdot\underline{x} > 0$ für beliebiges $\underline{x} \neq \underline{0}$) läßt sich eine spezielle Zerlegung $\underline{A} = \underline{C}^T\cdot\underline{C}$ mit einer oberen Dreiecksmatrix $\underline{C}$ und einer unteren Dreiecksmatrix $\underline{C}^T$ (transponierte Matrix von $\underline{C}$) angeben. Der Algorithmus für eine Choleski–Zerlegung ist wesentlich schneller als der für eine LR–Zerlegung. Für eine Choleski–Zerlegung geeignete Matrizen $\underline{A}$ treten häufig bei Verfahren mit Finiten Elementen auf.

3.3.3 Vergleich des Rechenaufwands

Die Anzahl der erforderlichen Rechenoperationen beträgt sowohl beim Gauß–Algorithmus wie auch bei der LR–Zerlegung etwa $\frac{2}{3}\cdot n^3$ Rechenoperationen. Kennt man die LR–Zerlegung, so werden für jede weitere rechte Seite nur noch $2n^2$–n Operationen für das Vorwärts– und Rückwärtseinsetzen benötigt. Für die Choleski–Zerlegung braucht man etwa $\frac{1}{3}\cdot n^3$ Operationen, zusätzlich sind noch n Quadratwurzeln zu berechnen.

3.4. Nachiteration zur Verbesserung der Lösung eines LGS

Beim Gauß – Algorithmus können Rundungsfehler die Genauigkeit der Lösung beeinflussen. In solchen Fällen läßt sich das Ergebnis durch Nachiteration verbessern.

Es sei $\underline{x} = \underline{x}^{(0)}$ die mit dem Gauß – Algorithmus berechnete Lösung. Für eine Verbesserung macht man folgenden Ansatz:

$$\underline{x}^{(1)} = \underline{x}^{(0)} + \underline{\Delta x} \quad \text{mit} \quad \underline{\Delta x} = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{bmatrix} .$$

$x^{(1)}$ setzt man in das LGS ein und erhält

$$\underline{A} \cdot \underline{x}^{(1)} = \underline{A} \cdot (\underline{x}^{(0)} + \underline{\Delta x}) = \underline{A} \cdot \underline{x}^{(0)} + \underline{A} \cdot \underline{\Delta x} = \underline{b} \quad ,$$

$$\underline{A} \cdot \underline{\Delta x} = \underline{b} - \underline{A} \cdot \underline{x}^{(0)} = \underline{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} .$$

Zur Berechnung der Δx_i ergibt sich die gleiche Koeffizientenmatrix $\underline{A}$. Die rechte Seite ergibt sich aus den Fehlern der einzelnen Gleichungen. Die einzelnen Eliminationsschritte zur Umwandlung in eine obere Dreiecksmatrix $\underline{R}$ sind bekannt, es braucht nur noch die neue rechte Seite $\underline{r}$ umgerechnet zu werden. Besonders einfach wird die Nachiteration, wenn eine LR– bzw. Choleski–Zerlegung bekannt ist.

Zu beachten ist bei Nachiteration folgendes: Die Fehler r_i sollten möglichst genau bestimmt werden; man berechnet die r_i in der Regel mit doppelter Genauigkeit der Rechenanlage.

3.5. Iterationsverfahren zur Lösung eines LGS

3.5.1. Prinzip

Bei verschiedenen Anwendungen (z.B. bei partiellen Differentialgleichungen) treten lineare Gleichungssysteme mit großer Variablen– bzw. Gleichungszahl auf. Hier bieten sich oft Iterationsverfahren zur Bestimmung der Lösung an. Das Grundprinzip für die wichtigsten Verfahren ist:

Man forme das LGS $\underline{A} \cdot \underline{x} = \underline{b}$ wie folgt um (Voraussetzung: $a_{kk} \neq 0$):

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n &= b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \ldots + a_{3n}x_n &= b_3 \\
&\ldots \\
a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n &= b_n
\end{aligned}
$$

$$
\begin{aligned}
x_1 &= \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3 - \ldots - a_{1n}x_n) \\
x_2 &= \frac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3 - \ldots - a_{2n}x_n) \\
&\ldots \\
x_k &= \frac{1}{a_{kk}}\left(b_k - \sum_{\substack{i=1 \\ i \neq k}}^{n} a_{ki}x_i\right) \\
&\ldots \\
x_n &= \frac{1}{a_{nn}}\left(b_n - \sum_{i=1}^{n-1} a_{ni}x_i\right)
\end{aligned}
$$

Ein Iterationsverfahren wird dann so konstruiert, daß auf der rechten Seite die Startwerte oder bereits berechnete Werte eingesetzt werden. Diese liefern dann auf der linken Seite die x_k als neue Näherungen.

Man unterscheidet 2 Varianten dieser Iterationsverfahren:

3.5.2. Das Gesamtschrittverfahren (Jacobi – Verfahren)

Zur Berechnung des Iterationsresultats $\underline{x}^{(p+1)}$ werden in die rechte Seite die Ausgangswerte $\underline{x}^{(p)}$ eingesetzt:

$$x_k^{(p+1)} = \frac{1}{a_{kk}} \left(b_k - \sum_{\substack{i=1 \\ i \neq k}}^{n} a_{ki} x_i^{(p)} \right) \qquad \begin{array}{l} k = 1, 2, \ldots, n \\ p = 0, 1, \ldots \end{array}$$

Als Startwerte kann man etwa $x_k^{(0)} = 0$ für alle k verwenden

(falls man keine weiteren Informationen über die Lösung besitzt).

Beispiel:

$$\begin{aligned} 10x_1 + x_2 + x_3 &= 11 \\ x_1 - 10x_2 - x_3 &= -18 \\ x_1 - x_2 + 10x_3 &= -11 \end{aligned}$$

Iterationsvorschrift:

$$\begin{aligned} x_1^{(p+1)} &= \frac{1}{10} (11 \quad - x_2^{(p)} - x_3^{(p)}) \\ x_2^{(p+1)} &= \frac{1}{10} (18 + x_1^{(p)} \quad - x_3^{(p)}) \\ x_3^{(p+1)} &= \frac{1}{10} (-11 - x_1^{(p)} + x_2^{(p)} \quad) \end{aligned}$$

Ergebnis bei Startwerten 0:

p	$x_1^{(p)}$	$x_2^{(p)}$	$x_3^{(p)}$
0	0.000000	0.000000	0.000000
1	1.100000	1.800000	–1.100000
2	1.030000	2.020000	–1.030000
3	1.001000	2.006000	–1.001000
4	0.999500	2.000200	–0.999500
5	0.999930	1.999900	–0.999930
6	1.000003	1.999986	–1.000003
7	1.000002	2.000001	–1.000002
8	1.000000	2.000000	–1.000000
9	1.000000	2.000000	–1.000000

3.5.3. Das Einzelschrittverfahren (Gauß – Seidel)

Man setzt im Unterschied zum Gesamtschrittverfahren jeweils die neuesten Iterationsergebnisse ein. Bei der Berechnung von $x_k^{(p+1)}$ sind zuvor bereits $x_1^{(p+1)}, x_2^{(p+1)}, \ldots, x_{k-1}^{(p+1)}$ bestimmt worden und können verwendet werden. Man erhält folgende Vorschrift:

$$x_k^{(p+1)} = \frac{1}{a_{kk}} \left(b_k - \sum_{i=1}^{k-1} a_{ki} x_i^{(p+1)} - \sum_{i=k+1}^{n} a_{ki} x_i^{(p)} \right) \qquad \begin{array}{l} k = 1, \ldots, n \\ p = 0, 1, \ldots \end{array}$$

Beispiel (s.o.):

Iterationsvorschrift:

$$\begin{aligned} x_1^{(p+1)} &= \frac{1}{10} (11 \qquad - x_2^{(p)} - x_3^{(p)}) \\ x_2^{(p+1)} &= \frac{1}{10} (18 + x_1^{(p+1)} \qquad - x_3^{(p)}) \\ x_3^{(p+1)} &= \frac{1}{10} (-11 - x_1^{(p+1)} + x_2^{(p+1)} \quad) \end{aligned}$$

Ergebnis bei Startwerten 0:

p	$x_1^{(p)}$	$x_2^{(p)}$	$x_3^{(p)}$
0	0.000000	0.000000	0.000000
1	1.100000	1.910000	–1.019000
2	1.010900	2.002990	–1.000791
3	0.999780	2.000057	–0.999972
4	0.999992	1.999996	–1.000000
5	1.000000	2.000000	–1.000000
6	1.000000	2.000000	–1.000000

Bemerkung zur Konvergenz beider Verfahren:

Im Fall von "Diagonaldominanz", d.h. $|a_{kk}| > \sum_{\substack{i=1 \\ i \neq k}}^{n} |a_{ki}|$, konvergieren beide Verfahren für beliebigen Startvektor $\underline{x}^{(0)}$ gegen die (eindeutig bestimmte) Lösung des LGS. Der Rechenaufwand beträgt für beide Verfahren etwa n^2 Rechenoperationen je Iterationsschritt.

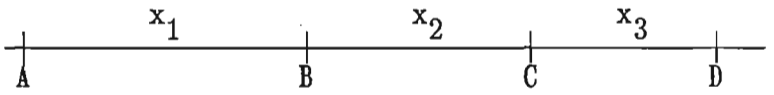
4. Fehlerausgleich bei überbestimmten linearen Gleichungssystemen

4.1. Problemstellung

Gegeben sei das LGS $\underline{A} \cdot \underline{x} = \underline{b}$ mit $\underline{A} = \underline{A}_{m,n}$ und $m > n$ (mehr Gleichungen als Unbekannte). Mindestens (n+1) Gleichungen sollen dabei voneinander unabhängig sein; das LGS ist also überbestimmt (exakte Lösung führt zu Widerspruch). Derartige lineare Gleichungssysteme treten häufig bei Kontrollmessungen auf.

Beispiele:

1) Längenmessung: Auf einer Geraden liegen 4 Punkte ABCD. Die Abstände x_i der Punkte sollen bestimmt werden.



Es bieten sich 6 Messungen an:

$\overline{AB} = 105.27$ m , $\overline{AC} = 177.95$ m , $\overline{AD} = 226.54$ m

$\overline{BC} = 72.83$ m , $\overline{BD} = 121.08$ m , $\overline{CD} = 48.31$ m .

Dies entspricht dem folgenden System von 6 linearen Gleichungen für die 3 Unbekannten x_1, x_2, x_3

$$
\begin{aligned}
x_1 &= 105.27 \\
x_2 &= 72.83 \\
x_3 &= 48.31 \\
x_1 + x_2 &= 177.95 \\
x_1 + x_2 + x_3 &= 226.54 \\
x_2 + x_3 &= 121.08
\end{aligned}
$$

2) Winkelmessung α , β , γ im Dreieck (siehe 3.1).

4.2 Fehlerausgleich nach der Methode der kleinsten Quadrate (Gauß)

Bezeichnungen: $\underline{a}_k$ sei der k–te Spaltenvektor der Matrix $\underline{A}$:

$$\underline{A} = \begin{bmatrix} a_{11} & a_{12} \cdots a_{1k} \cdots a_{1n} \\ a_{21} & a_{22} \cdots a_{2k} \cdots a_{2n} \\ \cdots & \\ a_{m1} & a_{m2} \cdots a_{mk} \cdots a_{mn} \end{bmatrix} = [\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_k, \ldots, \underline{a}_n]$$

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_m \end{bmatrix}.$$

Das LGS $\underline{A} \cdot \underline{x} = \underline{b}$ läßt sich i.a. ohne Widerspruch nicht exakt lösen; je nach Wahl von Zahlenwerten x_i ergibt sich in der j–ten Gleichung ein Fehler r_j. Man macht nun einen Ansatz mit derartigen Korrekturen r_j:

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 + r_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n &= b_2 + r_2 \\
&\ldots \\
a_{j1}x_1 + a_{j2}x_2 + a_{j3}x_3 + \ldots + a_{jn}x_n &= b_j + r_j \\
&\ldots \\
a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \ldots + a_{mn}x_n &= b_m + r_m
\end{aligned}$$

Ziel ist es , die x_i so zu bestimmen, daß die Fehler r_j $(j = 1,\ldots,m)$ "im Mittel" möglichst klein werden.

Das gegebene LGS wird durch die Korrekturen r_j in ein LGS

$$\underline{A} \cdot \underline{x} = \underline{b} + \underline{r} \quad \text{mit Fehlervektor } \underline{r} = \begin{bmatrix} r_1 \\ r_2 \\ .. \\ r_j \\ .. \\ r_m \end{bmatrix} \quad \text{umgeformt.}$$

Für diesen Fehlervektor $\underline{r}$ muß nun ein "Maß" angegeben werden, das minimiert werden soll.

Gauß schlägt dazu als mittleres Fehlermaß den Betrag des Vektors $\underline{r}$ vor.

Dies führt zu folgender Forderung:

Man bestimme die x_i so, daß die Fehlerquadratsumme

$$|\underline{r}|^2 = r_1^2 + r_2^2 + \ldots + r_j^2 + \ldots + r_m^2$$

minimal werden soll.

Herleitung des Verfahrens für n = 2 Veränderliche x_1 , x_2

(die Herleitung für allgemeines n erfolgt analog):

Berechnung der r_j :

$$\begin{aligned} r_1 &= a_{11}x_1 + a_{12}x_2 - b_1 \\ r_2 &= a_{21}x_1 + a_{22}x_2 - b_2 \\ &\ldots \\ r_j &= a_{j1}x_1 + a_{j2}x_2 - b_j \\ &\ldots \\ r_m &= a_{m1}x_1 + a_{m2}x_2 - b_m \end{aligned}$$

Die Fehlerquadratsumme $|\underline{r}|^2$ ergibt sich als folgende Funktion $f(x_1,x_2)$ zweier Veränderlicher:

$$\begin{aligned} f(x_1,x_2) = &(a_{11}x_1 + a_{12}x_2 - b_1)^2 + (a_{21}x_1 + a_{22}x_2 - b_2)^2 + \ldots \\ &+ (a_{j1}x_1 + a_{j2}x_2 - b_j)^2 + \ldots + (a_{m1}x_1 + a_{m2}x_2 - b_m)^2 \end{aligned}$$

Zur Bestimmung des Minimums von $f(x_1,x_2)$ sind die partiellen Ableitungen von f nach x_1 bzw. x_2 null zu setzen; es ergeben sich so 2 Bedingungen für die 2 Veränderlichen x_1 und x_2 :

$$\begin{aligned} \frac{\partial f}{\partial x_1} &= 2(a_{11}x_1 + a_{12}x_2 - b_1)a_{11} + 2(a_{21}x_1 + a_{22}x_2 - b_2)a_{21} + \ldots \\ &+ 2(a_{j1}x_1 + a_{j2}x_2 - b_j)a_{j1} + \ldots + 2(a_{m1}x_1 + a_{m2}x_2 - b_m)a_{m1} = 0 \end{aligned}$$

$$\begin{aligned} \frac{\partial f}{\partial x_2} &= 2(a_{11}x_1 + a_{12}x_2 - b_1)a_{12} + 2(a_{21}x_1 + a_{22}x_2 - b_2)a_{22} + \ldots \\ &+ 2(a_{j1}x_1 + a_{j2}x_2 - b_j)a_{j2} + \ldots + 2(a_{m1}x_1 + a_{m2}x_2 - b_m)a_{m2} = 0 \end{aligned}$$

oder anders zusammengefaßt

$$(a_{11}^2+a_{21}^2+\ldots+a_{m1}^2)x_1 + (a_{11}a_{12}+a_{21}a_{22}+\ldots+a_{m1}a_{m2})x_2 = a_{11}b_1+a_{21}b_2+\ldots+a_{m1}b_m$$

$$(a_{11}a_{12}+a_{21}a_{22}+\ldots+a_{m1}a_{m2})x_1 + (a_{12}^2+a_{22}^2+\ldots+a_{m2}^2)x_2 = a_{12}b_1+a_{22}b_2+\ldots+a_{m2}b_m$$

Die im LGS auftretenden Koeffizienten lassen sich einfach durch Skalarprodukte der Spaltenvektoren $\underline{a}_k$ der Matrix $\underline{A}$ ausdrücken; es ergeben sich die "Gaußschen Normalgleichungen" für den Fehlerausgleich:

(1) $(\underline{a}_1 \cdot \underline{a}_1)x_1 + (\underline{a}_1 \cdot \underline{a}_2)x_2 = (\underline{a}_1 \cdot \underline{b})$

(2) $(\underline{a}_2 \cdot \underline{a}_1)x_1 + (\underline{a}_2 \cdot \underline{a}_2)x_2 = (\underline{a}_2 \cdot \underline{b})$

Man kann zeigen, daß die Lösungen x_1 , x_2 der Normalgleichungen (1),(2) die Funktion $f(x_1,x_2)$ tatsächlich minimieren.

Für n Variable liefert eine analoge Überlegung die n Normalgleichungen

(1) $(\underline{a}_1 \cdot \underline{a}_1)x_1 + (\underline{a}_1 \cdot \underline{a}_2)x_2 + \ldots + (\underline{a}_1 \cdot \underline{a}_n)x_n = (\underline{a}_1 \cdot \underline{b})$

(2) $(\underline{a}_2 \cdot \underline{a}_1)x_1 + (\underline{a}_2 \cdot \underline{a}_2)x_2 + \ldots + (\underline{a}_2 \cdot \underline{a}_n)x_n = (\underline{a}_2 \cdot \underline{b})$

...

(n) $(\underline{a}_n \cdot \underline{a}_1)x_1 + (\underline{a}_n \cdot \underline{a}_2)x_2 + \ldots + (\underline{a}_n \cdot \underline{a}_n)x_n = (\underline{a}_n \cdot \underline{b})$

Bemerkungen:

1) Matrizentheoretisch entsprechen die Normalgleichungen dem LGS $\underline{A}^T \cdot \underline{A} \cdot \underline{x} = \underline{A}^T \cdot \underline{b}$.

2) Wählt man bei einer Veränderlichen $x_1 = x$ die Koeffizienten $a_{j1} = 1$, so entspricht dies m Einzelmessungen

   $x = b_1$, $x = b_2$, ... $x = b_m$.

   Die Normalgleichungen liefern dann als Lösung für den Fehlerausgleich das arithmetische Mittel $x = \frac{1}{m} \cdot (b_1 + b_2 + \ldots + b_m)$.

3) Fehlerausgleich nach der Gaußschen Methode der kleinsten Quadrate bewirkt, daß einzelne "Ausrutscher" bei Messungen nicht so stark ins Gewicht fallen. Dies entspricht Meßfehlern, die statistisch einer Normalverteilung entsprechen. Als mittlerer Fehler der ausgeglichenen Werte wird der Ausdruck

   $$\rho = \sqrt{\frac{1}{m - n} \cdot (r_1^2 + r_2^2 + \ldots + r_m^2)}$$

   angegeben $(m - n$ ... Anzahl der überschüssigen Beobachtungen).
   Je kleiner ρ ist, umso besser ist der Ausgleich. Formeln über den mittleren Fehler der einzelnen Variablen x_i siehe Literatur über Fehler– und Ausgleichsrechnung.

4) Ein anderes Fehlermaß für $\underline{r}$ führt zu anderen Ausgleichsbedingungen. Bei Approximationsproblemen (siehe 6.) wird neben der Gaußschen Fehlerquadratsumme $|\underline{r}|^2$ häufig das Tschebyscheff– Maß $\max\limits_{j=1..m} |r_j|$ gewählt. Fehlerausgleich nach Tschebyscheff entspricht der Forderung garantierter Fehlertoleranz für jede Gleichung. Einzelne Ausrutscher fallen dabei stark ins Gewicht.

Lösung von Beispiel 1:

Spaltenvektoren:
$$\underline{a}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad \underline{a}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \underline{a}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix},$$

$$\underline{b} = \begin{bmatrix} 105.27 \\ 72.83 \\ 48.31 \\ 177.95 \\ 226.54 \\ 121.08 \end{bmatrix}$$

Koeffizienten der Normalgleichungen:

$(\underline{a}_1 \cdot \underline{a}_1) = 3$, $(\underline{a}_1 \cdot \underline{a}_2) = (\underline{a}_2 \cdot \underline{a}_1) = 2$, $(\underline{a}_1 \cdot \underline{a}_3) = (\underline{a}_3 \cdot \underline{a}_1) = 1$

$(\underline{a}_2 \cdot \underline{a}_2) = 4$, $(\underline{a}_2 \cdot \underline{a}_3) = (\underline{a}_3 \cdot \underline{a}_2) = 2$, $(\underline{a}_3 \cdot \underline{a}_3) = 3$,

$(\underline{a}_1 \cdot \underline{b}) = 509.76$, $(\underline{a}_2 \cdot \underline{b}) = 598.40$, $(\underline{a}_3 \cdot \underline{b}) = 395.93$.

Normalgleichungen:
$$\begin{aligned} 3x_1 + 2x_2 + x_3 &= 509.76 \\ 2x_1 + 4x_2 + 2x_3 &= 598.40 \\ x_1 + 2x_2 + 3x_3 &= 395.93 \end{aligned}$$

Dies ergibt die Lösungen
$$\begin{aligned} x_1 &= 105.28 \\ x_2 &= 72.78 \\ x_3 &= 48.37 \end{aligned}$$

und einen mittleren Fehler von $\rho = 0.108$.

4.3. Fehlerausgleich nach Gauß mit Nebenbedingungen

Beispiel 2 aus 4.1 war die Messung der 3 Winkel eines Dreiecks. Durch die Nebenbedingung $\alpha + \beta + \gamma = 180^{\circ}$ wird das LGS überbestimmt.

Diese Nebenbedingung ist eine exakte Beziehung, die nicht ausgeglichen werden darf. Man spricht hier von einer bedingten Beobachtung oder Messung.

Ein einfaches Verfahren zur Lösung des Problems ist die Reduzierung der Anzahl der Variablen durch Einsetzen der Nebenbedingung. Dann erst erfolgt der Fehlerausgleich nach der Methode der kleinsten Quadrate.

Lösung von Beispiel 2:

$$\begin{aligned} \alpha &= 32.3 \\ \beta &= 67.9 \\ \gamma &= 80.1 \quad \text{(alle Maße in Dez.Grad)} \end{aligned}$$

Man ersetzt in der 3.Gleichung γ durch $\gamma = 180 - \alpha - \beta$ und erhält folgende 3 Gleichungen für den Fehlerausgleich

$$\begin{aligned} \alpha &= 32.3 \\ \beta &= 67.9 \\ \alpha + \beta &= 180 - 80.1 = 99.9 \end{aligned}$$

Fehlerausgleich: Es ergeben sich folgende Spaltenvektoren

$$\underline{a}_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \quad , \quad \underline{a}_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \quad , \quad \underline{b} = \begin{bmatrix} 32.3 \\ 67.9 \\ 99.9 \end{bmatrix} .$$

Koeffizienten der Normalgleichungen

$$(\underline{a}_1 \cdot \underline{a}_1) = 2 \quad , \quad (\underline{a}_1 \cdot \underline{a}_2) = (\underline{a}_2 \cdot \underline{a}_1) = 1 \quad , \quad (\underline{a}_2 \cdot \underline{a}_2) = 2 \quad ,$$

$$(\underline{a}_1 \cdot \underline{b}) = 32.3 + 99.9 = 132.2 \quad , \quad (\underline{a}_2 \cdot \underline{b}) = 67.9 + 99.9 = 167.8 \ .$$

Die Normalgleichungen

$$\begin{aligned} 2\alpha + \beta &= 132.2 \\ \alpha + 2\beta &= 167.8 \end{aligned}$$

besitzen die Lösungen

$$\begin{aligned} \alpha &= 32.20^{\circ} \\ \beta &= 67.80^{\circ} \quad , \end{aligned}$$

und aus der Winkelsumme im Dreieck erhält man

$$\gamma = 80.00^{\circ} .$$

Man erkennt, daß die Korrekturen gleichmäßig auf alle 3 Winkel verteilt sind.

5. Interpolation durch Polynome

5.1. Problemstellung

Gegeben sind (n + 1) Stützpunkte

$P_0(x_0|y_0)$, $P_1(x_1|y_1)$, ... , $P_n(x_n|y_n)$

mit paarweise verschiedenen Stützstellen

$x_i \quad (i = 0 , 1 ,..., n)$;

die Stützpunkte können Meßpunkten entsprechen. Häufig werden die Stützstellen äquidistant gewählt, d.h. es gilt

$x_i = x_0 + i \cdot h$.

Ziel ist es, durch die Punkte P_i eine Funktionskurve f(x) zu legen; f(x) bezeichnet man als Interpolationsfunktion. Mit Hilfe von f(x) kann man z.B.

- an einer beliebigen Stelle $\bar{x}$ aus dem Intervall $[x_0,x_n]$ einen y–Wert $f(\bar{x})$ berechnen ("interpolieren");
- Ableitungswerte bestimmen (Numerische Differentiation);
- Bestimmte Integrale berechnen (Numerische Integration).

Gesucht sind möglichst einfache analytische Interpolationsfunktionen; die wichtigsten sind

- Polynome $p_n(x) = \sum_{k=0}^{n} a_k x^k$
- trigonometrische "Polynome" $Q_n(x) = \sum_{k=0}^{n} (a_k \cos kx + b_k \sin kx)$

  für periodische Zusammenhänge.

In diesem Abschnitt werden Grundzüge von Interpolationspolynomen behandelt. Approximation durch trigonometrische Polynome siehe Abschnitt 7.

5.2. Der Satz vom Interpolationspolynom

Satz: Zu (n+1) paarweise verschiedenen Stützstellen $x_0, x_1, ..., x_n$ mit Stützwerten $y_0, y_1, ..., y_n$ gibt es genau ein Interpolationspolynom $p_n(x)$ vom Grad $\leq n$ mit $p_n(x_i) = y_i$ für $i = 0, 1, ..., n$.

Beweis : Ansatz $\quad p_n(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n$.

Punktprobe für $P_0, P_1, ... P_n$ liefert (n+1) Gleichungen für die gesuchten Polynomkoeffizienten a_k

$$
\begin{aligned}
P_0: &\quad p_n(x_0) = a_0 + a_1 x_0 + a_2 x_0^2 + ... + a_n x_0^n = y_0 \\
P_1: &\quad p_n(x_1) = a_0 + a_1 x_1 + a_2 x_1^2 + ... + a_n x_1^n = y_1 \\
P_2: &\quad p_n(x_2) = a_0 + a_1 x_2 + a_2 x_2^2 + ... + a_n x_2^n = y_2 \\
&\quad \cdots \\
P_n: &\quad p_n(x_n) = a_0 + a_1 x_n + a_2 x_n^2 + ... + a_n x_n^n = y_n
\end{aligned}
$$

Es ergibt sich ein LGS von (n+1) Gleichungen für (n+1) Unbekannte $a_0, a_1, ..., a_n$.
Dieses hat die Systemdeterminante

$$
D = \begin{vmatrix}
1 & x_0 & x_0^2 & x_0^3 & ... & x_0^n \\
1 & x_1 & x_1^2 & x_1^3 & ... & x_1^n \\
\cdots & & & & & \\
1 & x_n & x_n^2 & x_n^3 & ... & x_n^n
\end{vmatrix} \quad .
$$

Man kann zeigen, daß eine solche "Vandermondesche Determinante" für paarweise verschiedene Stützstellen x_i stets $\neq 0$ ist; das LGS ist damit eindeutig lösbar.
Der Beweis liefert gleichzeitig ein Verfahren zur Bestimmung der Koeffizienten a_k des Interpolationspolynoms. Dabei ist ein LGS mit n+1 Gleichungen zu lösen. Dies ist oft etwas mühsam; es gibt daher verschiedene geschicktere Ansätze, die das Interpolationspolynom liefern (z.B. das Newtonsche IP, siehe nächster Abschnitt 5.3).

5.3. Das Interpolationspolynom nach Newton

Beim Newtonschen Ansatz wird das IP schrittweise Stützstelle um Stützstelle aufgebaut. Dies erfolgt durch den Ansatz

$$p_n(x) = c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1) + c_3(x-x_0)(x-x_1)(x-x_2) + \ldots$$
$$+ c_{n-1}(x-x_0)(x-x_1)\ldots(x-x_{n-2}) + c_n(x-x_0)(x-x_1)\ldots(x-x_{n-1}) \quad .$$

Punktprobe für die Stützstellen in aufsteigender Reihenfolge:

$P_0(x_0|y_0)$ $\qquad c_0 = y_0 \doteq f_0$

$P_1(x_1|y_1)$ $\qquad c_0 + c_1(x_1-x_0) = y_1 \qquad \rightarrow \qquad c_1 = \dfrac{y_1-y_0}{x_1-x_0} \doteq f_{10}$

(c_1 ... Steigung der Geraden P_0P_1)

$P_2(x_2|y_2)$ $\qquad c_0 + c_1(x_2-x_0) + c_2(x_2-x_0)(x_2-x_1) = y_2$.

Durch Umrechnen läßt sich zeigen: $\qquad c_2 = \dfrac{\dfrac{y_2-y_1}{x_2-x_1} - \dfrac{y_1-y_0}{x_1-x_0}}{x_2 - x_0} = \dfrac{f_{21} - f_{10}}{x_2 - x_0} \doteq f_{210}$.

Allgemein ergibt sich der k–te Koeffizient c_k durch die Rekursionsformel

$$c_k = f_{k,k-1,\ldots 2,1,0} = \frac{f_{k,k-1,\ldots,2,1} - f_{k-1,\ldots,1,0}}{x_k - x_0} \quad .$$

Die Zahlen $f_{k,k-1,\ldots 2,1,0}$ bezeichnet man als dividierte Differenzen (oder Steigungen) k–ter Ordnung; sie lassen sich mit Hilfe folgender Rekursionsformel berechnen:

$$f_{k,k-1,\ldots i} = \frac{f_{k,k-1,\ldots,i+1} - f_{k-1,\ldots,i}}{x_k - x_i} \quad .$$

Newtonsches Steigungsschema zur Darstellung:

x_0	$y_0=f_0$				
		f_{10}			
x_1	$y_1=f_1$		f_{210}		
		f_{21}		f_{3210}	
x_2	$y_2=f_2$		f_{321}		f_{43210}
		f_{32}		f_{4321}	...
x_3	$y_3=f_3$		f_{432}	...	
		f_{43}	...		
x_4	$y_4=f_4$	...			

Ein Vorteil des Ansatzes von Newton liegt darin, daß bei Hinzunahme eines weiteren Stützpunktes P_{n+1} nur eine zusätzliche Schrägzeile neu gerechnet werden muß, die dann den Koeffizienten c_{n+1} liefert (siehe Beispiele 1 und 2).

Beispiele: 1) Interpolationspolynom durch die Stützpunkte $(-2|8)$, $(-1|-1)$, $(0|-2)$, $(2|-4)$.

Steigungsschema:

x_i	y_i						
-2	8						
		-9	$\frac{-9}{1}=-9$				
-1	-1			8	$\frac{8}{2}=4$		
		-1	$\frac{-1}{1}=-1$			-4	$\frac{-4}{4}=-1$
0	-2			0	0		
		-2	$\frac{-2}{2}=-1$				
2	-4						

Interpolationspolynom:

$$p_3(x) = 8 - 9(x+2) + 4(x+2)(x+1) - (x+2)(x+1)x = -x^3 + x^2 + x - 2$$

Als Interpolationswert etwa an der Stelle $\overline{x} = 1$ ergibt sich $p_3(1) = -1$.

Beispiel 2: Wir nehmen zu den Punkten in Beispiel 1 noch einen fünften Punkt $(3|43)$ hinzu.

Das Steigungsschema ist jetzt um eine Schrägzeile zu ergänzen:

x_i	y_i								
-2	8								
		-9	$\frac{-9}{1}=-9$						
-1	-1			8	$\frac{8}{2}=4$				
		-1	$\frac{-1}{1}=-1$			-4	$\frac{-4}{4}=-1$		
0	-2			0	0			5	$\frac{5}{5}=1$
		-2	$\frac{-2}{2}=-1$			16	$\frac{16}{4}=4$		
2	-4			48	$\frac{48}{3}=16$				
		47	$\frac{47}{1}=47$						
3	43								

Als neues Interpolationspolynom ergibt sich

$$p_4(x) = p_3(x) + 1\cdot(x+2)(x+1)x(x-2) = x^4 - 3x^2 - 3x - 2$$

Beispiel 3: Interpolationspolynom durch 3 Punkte einer Kurve $y = f(x)$ an den Stützstellen a , a–h , a+h .

Steigungsschema:

x_i	y_i		
a–h	$f(a-h) \doteq y_1$		
		$\frac{y_2 - y_1}{h}$	
a	$f(a) \doteq y_2$		$\frac{y_3 - 2y_2 + y_1}{2h^2}$
		$\frac{y_3 - y_2}{h}$	
a+h	$f(a+h) \doteq y_3$		

Interpolationspolynom (Parabel):

$$p_2(x) = y_1 + \frac{y_2 - y_1}{h}\cdot(x - a + h) + \frac{y_3 - 2y_2 + y_1}{2h^2}\cdot(x - a + h)(x - a)$$

Dieses läßt sich umrechnen

$$\begin{aligned} p_2(x) &= y_1 + \frac{y_2 - y_1}{h}\cdot(x-a) + y_2 - y_1 + \frac{y_3 - 2y_2 + y_1}{2h^2}\cdot(x-a)^2 \\ &\quad + \frac{y_3 - 2y_2 + y_1}{2h}\cdot(x-a) \\ &= y_2 + \frac{y_3 - y_1}{2h}\cdot(x-a) + \frac{y_3 - 2y_2 + y_1}{2h^2}\cdot(x-a)^2 \\ &= f(a) + \frac{f(a+h) - f(a-h)}{2h}\cdot(x-a) + \frac{f(a+h) - 2f(a) + f(a-h)}{2h^2}\cdot(x-a)^2 \end{aligned}$$

Bemerkungen:

1) Für äquidistante Stützstellen gibt es modifizierte Ansätze, bei denen im Schema je Eintrag nur eine Differenz zu bilden ist. Aus dem Steigungsschema wird ein Differenzenschema (siehe Literatur).

2) Funktionswerte eines Polynoms berechnet man effizient mit dem Hornerschema (siehe etwa [15]). Dieses ergibt sich durch geschicktes Zusammenfassen (z.B. für n = 4)

$$\begin{aligned} p_4(x) &= a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 \\ &= [\{(a_4x + a_3)x + a_2\}x + a_1]x + a_0 \quad . \end{aligned}$$

Auswerten der Klammern (von innen nach außen) ergibt das Hornerschema:

	a_4	a_3	a_2	a_1	a_0
		$b_3\bar{x}$	$b_2\bar{x}$	$b_1\bar{x}$	$b_0\bar{x}$
$x = \bar{x}$	$a_4 \doteq b_3$	$a_3+b_3\bar{x}$ $\doteq b_2$	$a_2+b_2\bar{x}$ $\doteq b_1$	$a_1+b_1\bar{x}$ $\doteq b_0$	$a_0+b_0\bar{x} = p_4(\bar{x})$

Für allgemeines n lautet der Algorithmus für die b_k:

$$b_{n-1} = a_n \quad , \quad b_k = b_{k+1}\bar{x} + a_{k+1} \quad \text{für} \quad k = n-2, n-1, \ldots 2, 1, 0$$

Polynomwert an der Stelle $\bar{x}$:

$$p_n(\bar{x}) = a_0 + b_0\bar{x} \quad .$$

Dieser Algorithmus läßt sich bequem als Schleife programmieren.

Das Newtonsche Interpolationspolynom liefert keine nach Potenzen von x entwickelte Darstellung. Hier ist das Horner–Schema entsprechend zu modifizieren (z.B. für n = 4):

$$\begin{aligned} p_4(x) &= c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1) + c_3(x-x_0)(x-x_1)(x-x_2) \\ &\quad + c_4(x-x_0)(x-x_1)(x-x_2)(x-x_3) \\ &= c_0 + (x-x_0)\Big[c_1 + (x-x_1)[c_2 + (x-x_2)\{c_3 + (x-x_3)c_4\}]\Big] \end{aligned}$$

Schema:

$\bar{x}$		x_3	x_2	x_1	x_0
		$\bar{x}-x_3$	$\bar{x}-x_2$	$\bar{x}-x_1$	$\bar{x}-x_0$
	c_4	c_3	c_2	c_1	c_0
		$b_3(\bar{x}-x_3)$	$b_2(\bar{x}-x_2)$	$b_1(\bar{x}-x_1)$	$b_0(\bar{x}-x_0)$
	$c_4=b_3$	b_2	b_1	b_0	$p_4(\bar{x})$

Die b_k ergeben sich wieder durch Summation. Der Algorithmus zur Berechnung der b_k lautet für allgemeines n:

$$b_{n-1} = c_n \quad , \quad b_k = c_{k+1} + b_{k+1}(\bar{x}-x_{k+1}) \quad \text{für } k = n-2, \ldots, 2, 1, 0$$

Polynomwert an der Stelle $\bar{x}$:

$$p_n(\bar{x}) = c_0 + b_0(\bar{x}-x_0) \quad .$$

Beispiel 4: Wir rechnen vom Beispiel 1 den Interpolationswert $p_3(1) = -1$ nach:

$\bar{x} = 1$

x_k		0	−1	−2
$\bar{x} - x_k$		1	2	3
c_k	−1	4	−9	8
		$(-1)\cdot 1$	$3\cdot 2$	$(-3)\cdot 3$
	−1	3	−3	$-1 = p_3(1)$

5.4. Abschließende Bemerkungen zu Interpolationspolynomen

1) Neben dem Ansatz von Newton gibt es noch verschiedene andere, z.B. den Ansatz von Lagrange. Weiter berechnen verschiedene Algorithmen den Interpolationswert an einer Stelle x direkt aus den Stützpunkten (Aitken, Neville).

2) Mehrdimensionale Interpolation dient zur Darstellung von Funktionen mehrerer Veränderlicher (siehe Literatur).

3) Mit wachsender Stützstellenzahl nimmt normalerweise der Grad des Interpolationspolynoms zu. Polynome mit hohem Grad werden jedoch unhandlich und "wellig" (d.h. sie versuchen Extremstellen anzunehmen). Bei der Approximation von Funktionen durch Polynome (siehe 6.) wächst die Qualität ab einem bestimmten Polynomgrad nicht mehr. Man approximiert dann z.B. segmentweise durch Polynome (vgl. Splines, 6.4).

6. Approximation von Funktionen

6.1. Einführung

Eine Funktion f(x) sei durch ihre Funktionsvorschrift oder durch Stützpunkte $P_i(x_i|y_i)$ vorgegeben (etwa aus Messungen).

Ziel: Man stelle f(x) möglichst gut durch einfach zu berechnende Funktionen dar (Annäherung von f(x) durch einfache Funktionen, Approximation).

Die wichtigsten Näherungsfunktionen sind Polynome oder trigonometrische Polynome.

Wir verwenden im folgenden zur Approximation endlich viele Stellen x_i aus dem Definitionsbereich von f(x) ; dies bezeichnet man als diskrete Approximation.

6.2. Approximation einer Funktion durch ihr Taylorpolynom

Die Funktion f(x) sei n+1 mal stetig differenzierbar in einer Umgebung der Stelle x_0 . Dann läßt sich f(x) nach dem Satz von Taylor an der Stelle x_0 in ein Taylorpolynom $T_n(x)$ entwickeln:

$$f(x) = T_n(x) + R_n = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!} \cdot (x-x_0)^k + R_n$$

$$= f(x_0) + f'(x_0) \cdot (x-x_0) + \frac{f''(x_0)}{2!} \cdot (x-x_0)^2 + \frac{f'''(x_0)}{3!} \cdot (x-x_0)^3$$

$$+ \ldots + \frac{f^{(n)}(x_0)}{n!} \cdot (x-x_0)^n + R_n$$

mit dem Restglied (Fehler)

$$R_n = \frac{f^{(n+1)}(u)}{(n+1)!} \cdot (x-x_0)^{n+1} \quad ;$$

dabei ist u eine (i.a. unbekannte) Stelle im Intervall $[x_0,x]$.

Mit R_n kann der Fehler $|f(x) - T_n(x)|$ abgeschätzt werden.

Beispiel: Berechnung von e^x im Intervall $0 \le x \le 1$ bei Entwicklung um $x_0 = 0$.

Wegen $f^{(k)}(x) = e^x$ gilt $f^{(k)}(0) = 1$ für alle k.

Das Taylorpolynom für e^x lautet daher

$$T_n(x) = \sum_{k=0}^{n} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots + \frac{x^n}{n!} \quad .$$

Abschätzung des Restglieds R_n :

Für $f^{(k+1)}(u) = e^u$ gilt im Bereich $0 \le u \le 1$

$$1 \le e^u \le e^1 < 3 \quad ,$$

weiter gilt $0 \le x^{n+1} \le 1^{n+1} \le 1$. Daraus folgt

$$|R_n| = |\frac{f^{(n+1)}(u)}{(n+1)!} \cdot x^{n+1}| \le \frac{3}{(n+1)!}$$

Für Approximation durch die ersten 10 Glieder gilt daher

$$|R_{10}| \le 3/11! = 7.5 \cdot 10^{-8} < 5 \cdot 10^{-7} \ ,$$

T_{10} approximiert e^x im Intervall [0,1] auf 6 Dezimalstellen genau.

Verdoppelung von $(x-x_0)$ bzw. im obigen Beispiel von x bedeutet allein wegen des Faktors x^{n+1} eine Vervielfachung des Fehlers R_n um das 2^{n+1}– fache. Legt man für e^x etwa das Intervall [0,2] zugrunde, so ergeben sich die Abschätzungen

$1 \le e^u \le e^2 < 8$, $0 \le x^{n+1} \le 2^{n+1}$ und damit

$$|R_n| \le 8 \cdot \frac{2^{n+1}}{(n+1)!} \ .$$

Für n = 10 ergibt sich jetzt

$|R_{10}| \le 8 \cdot 2^{11}/11! = 0.00041$,

T_{10} approximiert e^x im Intervall [0,2] nur auf 3 Dezimalstellen genau. Taylorpolynome eignen sich daher vor allem zur Approximation einer Funktion in der Umgebung der Entwicklungs – stelle x_0.

6.3. Approximation von f(x) durch ein Stützpolynom

Man bestimmt durch Vorgabe von Stützstellen x_i Stützpunkte $P_i(x_i|f(x_i)$ und legt durch diese ein Interpolationspolynom (siehe 4.). In den Stützstellen $x_0, x_1, ..., x_n$ approximiert das IP $p_n(x)$ die Funktion f(x) exakt. Im Fall einer (n+1) mal stetig differenzierbaren Funktion f(x) gilt folgende zum Restglied eines Taylorpolynoms analoge Fehlerformel:

$$R = f(x) - p_n(x) = \frac{f^{(n+1)}(u)}{(n+1)!} \cdot (x-x_0)(x-x_1)\cdots(x-x_n) \quad ,$$

u ist dabei eine fest existierende Stelle im Interpolationsbereich.

Zu viele Stützstellen erhöhen den Grad des Interpolationspolynoms und bringen keine Verbesserung der Approximationsgüte. Hier empfehlen sich im wesentlichen eine Approximation durch Splines (siehe 6.4) oder durch Ausgleichspolynome (siehe 6.5).

6.4 Approximation durch kubische Splines

Approximiert werden soll eine Funktion f(x) im Intervall [a,b] durch Vorgabe der Stützstellen $a=x_0 , x_1 , x_2 , \dots , x_n = b$. Wir beschränken uns im folgenden auf äquidistante Stützstellen

$$x_k = x_0 + k \cdot h \quad , 0 \leq k \leq n \; .$$

Für h gilt dann $h = \frac{b - a}{n}$.

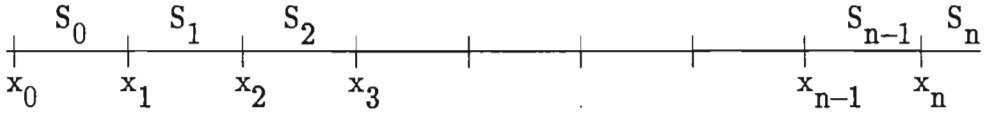
Bei der Approximation durch Splines sucht man in jedem Teilintervall ein (i.a. verschiedenes) Näherungspolynom $S_k(x)$ und sorgt dafür, daß die Polynome an den Intervallenden "glatt" aneinander anschließen (segmentierte Polynomdarstellung). Üblich ist die Forderung nach Übereinstimmung von Funktionswert sowie 1. und 2. Ableitung (d.h. Tangente und Krümmung). Dies ist möglich mit Polynomen ab Grad 3.

Definition: Als kubischen Spline (kubische Splinefunktion) bezeichnet man eine zweimal stetig differenzierbare Kurve, die aus Polynomsegmenten vom Grad 3 aufgebaut ist.

Bemerkung: Die Approximation einer Funktion durch eine kubische Splinefunktion entspricht der Biegelinie eines elastischen Balkens (engl. Spline: elastische Latte, die als Kurvenlineal im Schiffsbau zum Zeichnen von Biegelinien benutzt wird).

Berechnung kubischer Splinefunktionen

Es sei $S_k(x)$ das Splinepolynom im Intervall $[x_k,x_{k+1}]$



Das Polynom $S_n(x)$ wird formal zur Berechnung definiert.

Für $S_k(x)$ empfiehlt sich folgender Ansatz:

$$S_k(x) = a_k + b_k(x-x_k) + c_k(x-x_k)^2 + d_k(x-x_k)^3 \qquad 0 \leq k \leq n \; .$$

Für die im Approximationsbereich auftretenden n Polynome S_0 bis S_{n-1} sind insgesamt 4n Koeffizienten zu bestimmen.

Bedingungen für die Splinepolynome:

(a) Interpolation am linken Rand: $S_k(x_k) = y_k = f(x_k)$ $0 \le k \le n$

(b) Übergangsbedingungen:

(b1) Stetigkeit $S_k(x_k) = S_{k-1}(x_k)$ $1 \le k \le n$

(b2) 1. Ableitung $S_k'(x_k) = S_{k-1}'(x_k)$ $1 \le k \le n-1$

(b3) 2. Ableitung $S_k''(x_k) = S_{k-1}''(x_k)$ $1 \le k \le n$

In den Bedingungen (a), (b1) und (b3) wurde das Hilfspolynom $S_n(x)$ miteinbezogen. Dadurch erreicht man die Interpolationsbedingung am rechten Rand (Bedingungen (a) und (b1)) sowie eine formale Übereinstimmung der Krümmung am rechten Rand. Dies erfordert andererseits die Festlegung der Koeffizienten $a_n = S_n(x_n)$ und $c_n = \frac{1}{2} \cdot S_n''(x_n)$ des Hilfspolynoms; b_n und d_n werden nicht benötigt. Insgesamt sind daher 4n+2 Koeffizienten der $S_k(x)$ zu bestimmen.
Die Bedingungen (a) und (b) liefern aber nur 4n Gleichungen. Die restlichen 2 Gleichungen legt man durch Randbedingungen an den Stellen $a = x_0$ und $b = x_n$ fest.
Im folgenden beschränken wir uns auf sogenannte **natürliche Splines**:
Man fordert hier zusätzlich am Rand den Übergang der Splinefunktionen in Geraden (so wie ein Balken frei weiterlaufen würde). Dies entspricht folgenden Zusatzbedingungen:

(c) $S_0''(x_0) = S_n''(x_n) = 0$.

Es läßt sich zeigen, daß mit diesen 4n+2 Bedingungen die Koeffizienten der Splinepolynome S_0 bis S_{n-1} eindeutig berechnet werden können.

Bemerkungen:

1) Neben den natürlichen kubischen Splines gibt es auch andere Möglichkeiten zur Festlegung der beiden Bedingungen. Andere Bedingungen ermöglichen etwa Periodizität oder Festlegung der 2. Ableitung.
2) Die Erweiterung der Bedingung (b3) auf k = n setzt i.a. die Annahme eines natürlichen Splines voraus.

Aufstellung der Gleichungen für die Koeffizienten:

$$S_k(x) = a_k + b_k(x-x_k) + c_k(x-x_k)^2 + d_k(x-x_k)^3$$
$$S_k'(x) = b_k + 2c_k(x-x_k) + 3d_k(x-x_k)^2$$
$$S_k''(x) = 2c_k + 6d_k(x-x_k)$$

Die entsprechenden Bedingungen für k−1 lauten

$$S_{k-1}(x) = a_{k-1} + b_{k-1}(x-x_{k-1}) + c_{k-1}(x-x_{k-1})^2 + d_{k-1}(x-x_{k-1})^3$$
$$S_{k-1}'(x) = b_{k-1} + 2c_{k-1}(x-x_{k-1}) + 3d_{k-1}(x-x_{k-1})^2$$
$$S_{k-1}''(x) = 2c_{k-1} + 6d_{k-1}(x-x_{k-1})$$

Bedingung (a): $a_k = y_k \qquad 0 \le k \le n \qquad (1)$

Bedingung (c): $c_0 = c_n = 0 \qquad (2)$

Bedingung (b3): $2c_k = 2c_{k-1} + 6d_{k-1}(x_k-x_{k-1}) = 2c_{k-1} + 6d_{k-1}h$

$$d_{k-1} = \frac{c_k - c_{k-1}}{3h} \quad \text{oder mit } k \to k+1$$
$$d_k = \frac{1}{3h}\cdot(c_{k+1} - c_k) \qquad 0 \le k \le n-1 \qquad (5)$$

Bedingung (b1):

$$a_k = a_{k-1} + b_{k-1}(x_k-x_{k-1}) + c_{k-1}(x_k-x_{k-1})^2 + d_{k-1}(x_k-x_{k-1})^3$$
$$= a_{k-1} + b_{k-1}h + c_{k-1}h^2 + d_{k-1}h^3$$

Umformen und Einsetzen von (5) ergibt

$$b_{k-1}h = a_k - a_{k-1} - h^2(c_{k-1} + d_{k-1}h)$$
$$= a_k - a_{k-1} - h^2(c_{k-1} + \tfrac{1}{3}(c_k - c_{k-1}))$$
$$b_{k-1} = \frac{1}{h}(a_k - a_{k-1}) - \frac{h}{3}(c_k + 2c_{k-1})$$

und mit $k \to k+1$

$$b_k = \frac{1}{h}\cdot(a_{k+1} - a_k) - \frac{h}{3}\cdot(c_{k+1} + 2c_k) \qquad 0 \le k \le n-1 \qquad (4)$$

Bedingung (b2): (und Einsetzen von (5))

$$b_k = b_{k-1} + 2c_{k-1}(x_k-x_{k-1}) + 3d_{k-1}(x_k-x_{k-1})^2$$
$$= b_{k-1} + 2c_{k-1}h + 3\cdot\frac{c_k - c_{k-1}}{3h}\cdot h^2 = b_{k-1} + h\cdot(c_{k-1} + c_k)$$
$$b_k - b_{k-1} - h\cdot(c_k + c_{k-1}) = 0$$

und mit (4)

$$\frac{1}{h}\cdot(a_{k+1} - 2a_k + a_{k-1}) - \frac{h}{3}\cdot(c_{k+1} + c_k - 2c_{k-1} + 3c_k + 3c_{k-1}) = 0$$
$$\frac{3}{h}\cdot(a_{k+1} - 2a_k + a_{k-1}) = h\cdot(c_{k+1} + 4c_k + c_{k-1}) \qquad 1 \le k \le n-1 \qquad (3)$$

Zusammenfassend ergeben sich folgende 5 Bedingungen, die der Reihe nach auszuwerten sind:

$$a_k = y_k \qquad 0 \le k \le n \qquad (1)$$

$$c_0 = c_n = 0 \qquad (2)$$

$$\frac{3}{h}\cdot(a_{k+1} - 2a_k + a_{k-1}) = h\cdot(c_{k+1} + 4c_k + c_{k-1}) \qquad 1 \le k \le n-1 \qquad (3)$$

$$b_k = \frac{1}{h}\cdot(a_{k+1} - a_k) - \frac{h}{3}\cdot(c_{k+1} + 2c_k) \qquad 0 \le k \le n-1 \qquad (4)$$

$$d_k = \frac{1}{3h}\cdot(c_{k+1} - c_k) \qquad 0 \le k \le n-1 \qquad (5)$$

Die Gleichungssysteme (1) bis (5) sind eindeutig lösbar und numerisch gutartig.

Beispiel:

Man approximiere die Funktion $f(x) = \dfrac{2}{1 + x^2}$ im Intervall $[-1,1]$ mit Schrittweite $h = 1$.
Bestimmt werden soll der Interpolationswert an der Stelle 0.5 (exakter Wert $f(0.5) = 1.60$).

Stützstellen: $x_0 = -1 \quad x_1 = 0 \quad x_2 = 1 \quad \rightarrow n = 2$

$y_0 = 1 \quad y_1 = 2 \quad y_2 = 1$

Ansatz für die Splinepolynome:

$$S_0(x) = a_0 + b_0(x + 1) + c_0(x + 1)^2 + d_0(x + 1)^3$$

$$S_1(x) = a_1 + b_1x + c_1x^2 + d_1x^3$$

relevante Koeffizienten von $S_2(x)$: $a_2 = y_2 = 1$, $c_2 = 0$.

Auswertung der Bedingungen (1) bis (5):

(1) $a_0 = 1$, $a_1 = 2$, $a_2 = 1$

(2) $c_0 = c_2 = 0$

(3) liefert nur eine Bedingung für $k = 1$

$h\cdot(c_2 + 4c_1 + c_0) = \frac{3}{h}\cdot(a_2 - 2a_1 + a_0)$ und mit (1), (2), $h = 1$

$4c_1 = 3\cdot(1 - 4 + 1) = -6$, $c_1 = -\frac{3}{2}$

(4) $k = 0$: $b_0 = \frac{1}{h}\cdot(a_1 - a_0) - \frac{h}{3}\cdot(c_1 + 2c_0) = 1 - \frac{1}{3}\cdot(-\frac{3}{2}) = \frac{3}{2}$

$k = 1$: $b_1 = \frac{1}{h}\cdot(a_2 - a_1) - \frac{h}{3}\cdot(c_2 + 2c_1) = -1 - \frac{1}{3}\cdot(-3) = 0$

(5) $k = 0$: $d_0 = \frac{1}{3h}\cdot(c_1 - c_0) = \frac{1}{3}\cdot(-\frac{3}{2}) = -\frac{1}{2}$

$k = 1$: $d_1 = \frac{1}{3h}\cdot(c_2 - c_1) = \frac{1}{3}\cdot\frac{3}{2} = \frac{1}{2}$

Es ergeben sich die beiden Splinepolynome

$$S_0(x) = 1 + \frac{3}{2}\cdot(x + 1) - \frac{1}{2}\cdot(x + 1)^3 \qquad \text{für } -1 \le x \le 0$$

$$S_1(x) = 2 - \frac{3}{2}\cdot x^2 + \frac{1}{2}\cdot x^3 \qquad \text{für } 0 \le x \le 1$$

Rechnet man $S_0(x)$ um in $S_0(x) = -\frac{1}{2}\cdot x^3 - \frac{3}{2}\cdot x^2 + 2$, so sieht man, daß die Achsensymmetrie von f(x) für den Spline erhalten bleibt.

Interpolationswert an der Stelle $\bar{x} = \frac{1}{2}$: $S_1(0.5) = 1.69$.

Vergleich des Splines mit dem klassischen Interpolationspolynom $p_2(x)$:

$p_2(x) = 1 + (x+1) - (x+1)x = 2 - x^2$ $\qquad$ $p_2(0.5) = 1.75$

Lineare Interpolation liefert den Wert 1.50 .

6.5. Approximation durch Ausgleichspolynome

Bei Versuchen ist häufig ein funktionaler Zusammenhang vorgegeben (Mathematisches Modell). Eine Messung liefert eine Vielzahl von Punkten; durch diese soll die Kurve gelegt werden.

Beispiel:

1) Für die Auslenkung einer Federwaage gelte im Elastizitätsbereich

$F = c\cdot s$ $\qquad$ (s ... Auslenkung, c ... Federkonstante, F ... Kraft).

Eine Messung ergibt

F (in N)	0	1	2	3	4	5
s (in cm)	0	5.1	10.1	15.2	20.0	25.1

2) Für eine beschleunigte Bewegung eines Körpers gilt

$s = s(t) = s_0 + v_0 t + \frac{a}{2}\cdot t^2$

(s_0 ... Weg zur Zeit 0 , v_0 ... Anfangsgeschwindigkeit , a ... Beschleunigung)

Eine Messung beim Bremsen eines Eisenbahnzugs ergibt

t (in sec)	0	5	10	15	20	25	30
s (in m)	1000	1091	1165	1221	1260	1281	1285

Einsetzen der Stützpunkte in die physikalischen Formelgleichungen liefert ein überbestimmtes Gleichungssystem für die Koeffizienten.

Eine Lösung durch ein Interpolationspolynom ist nicht brauchbar, da dessen Grad durch die Anzahl der Stützstellen festgelegt ist und so dem Modell nicht entspricht.

Problem: Die Art der Kurve ist durch das Modell vorgegeben (in den Beispielen eine Gerade bzw. Parabel). Exakt können die Meßpunkte nicht auf die Modellkurve verteilt werden. Gesucht ist eine Verteilung, bei der die Fehler im Mittel minimal sind.

Das Problem entspricht der Lösung eines überbestimmten linearen Gleichungssystems nach der Gaußschen Methode der kleinsten Quadrate. Wir beschränken uns im folgenden auf Modelle ganzrationaler Funktionen (Polynome); die durch Fehlerausgleich sich ergebenden Kurven bezeichnet man dann als Ausgleichspolynome. Liegt speziell eine Gerade vor (vgl. Beispiel 1), so bezeichnet man diese als Ausgleichsgerade oder Regressionsgerade (lineare Regression). Regressionsgeraden treten häufig auch in der Statistik auf.

Bestimmung von Ausgleichspolynomen:

Gegeben sind N Stützstellen $x_1 , x_2 , \ldots x_N$ und die entsprechenden Stützpunkte $P_k(x_k | y_k=f(x_k))$. Weiter vorgegeben sei der Grad n des Ausgleichspolynoms durch ein mathematisches Modell.

Ansatz für ein Ausgleichspolynom vom Grad n :

$$A_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n \quad .$$

Gesucht sind die n+1 Koeffizienten $a_0 , a_1 , \ldots a_n$.

Punktprobe für die Stützpunkte ergibt N Gleichungen für n+1 Unbekannte a_k (mit $N > n+1$):

$$\begin{aligned}
P_1: &\quad a_0 + a_1 x_1 + a_2 x_1^2 + + a_3 x_1^3 \ldots + a_n x_1^n &= y_1 \\
P_2: &\quad a_0 + a_1 x_2 + a_2 x_2^2 + + a_3 x_2^3 \ldots + a_n x_2^n &= y_2 \\
P_3: &\quad a_0 + a_1 x_3 + a_2 x_3^2 + + a_3 x_3^3 \ldots + a_n x_3^n &= y_3 \\
&\quad \ldots \\
P_N: &\quad a_0 + a_1 x_N + a_2 x_N^2 + + a_3 x_N^3 \ldots + a_n x_N^n &= y_N
\end{aligned}$$

Fehlerausgleich nach der Gaußschen Methode der kleinsten Quadrate:

Es liegt das LGS $\underline{C}\cdot\underline{a} = \underline{b}$ vor mit Koeffizientenmatrix

$$\underline{C} = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & x_2^3 & \dots & x_2^n \\ \dots & & & & & \\ 1 & x_N & x_N^2 & x_N^3 & \dots & x_N^n \end{bmatrix} \quad \text{und} \quad \underline{a} = \begin{bmatrix} a_0 \\ a_1 \\ \dots \\ a_n \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix}..$$

Normalgleichungen für den Fehlerausgleich:

$$(\underline{c}_0\cdot\underline{c}_0)a_0 + (\underline{c}_0\cdot\underline{c}_1)a_1 + (\underline{c}_0\cdot\underline{c}_2)a_2 + \dots + (\underline{c}_0\cdot\underline{c}_n)a_n = (\underline{c}_0\cdot\underline{b})$$
$$(\underline{c}_1\cdot\underline{c}_0)a_0 + (\underline{c}_1\cdot\underline{c}_1)a_1 + (\underline{c}_1\cdot\underline{c}_2)a_2 + \dots + (\underline{c}_1\cdot\underline{c}_n)a_n = (\underline{c}_1\cdot\underline{b})$$
$$\dots$$
$$(\underline{c}_n\cdot\underline{c}_0)a_0 + (\underline{c}_n\cdot\underline{c}_1)a_1 + (\underline{c}_n\cdot\underline{c}_2)a_2 + \dots + (\underline{c}_n\cdot\underline{c}_n)a_n = (\underline{c}_n\cdot\underline{b})$$

Aus $\underline{c}_0 = \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix}$, $\underline{c}_1 = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_N \end{bmatrix}$, $\underline{c}_2 = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \dots \\ x_N^2 \end{bmatrix}$, $\dots$, $\underline{c}_n = \begin{bmatrix} x_1^n \\ x_2^n \\ \dots \\ x_N^n \end{bmatrix}$, $\underline{b} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix}$

folgt $(\underline{c}_i\cdot\underline{c}_j) = \sum_{k=1}^{N} x_k^{i+j}$, $(\underline{c}_i\cdot\underline{b}) = \sum_{k=1}^{N} x_k^i\cdot y_k$.

Die Normalgleichungen lauten für die wichtigsten Fälle von Ausgleichsgerade bzw. Ausgleichsparabel:

n = 1 : Ausgleichsgerade $A_1(x) = a_0 + a_1x$

(1) $$N\cdot a_0 + (\sum_{k=1}^{N} x_k)\cdot a_1 = \sum_{k=1}^{N} y_k$$

(2) $$(\sum_{k=1}^{N} x_k)\cdot a_0 + (\sum_{k=1}^{N} x_k^2)\cdot a_1 = \sum_{k=1}^{N} x_k y_k$$

n = 2 : Ausgleichsparabel $A_2(x) = a_0 + a_1 x + a_2 x^2$

$$(1) \qquad N \cdot a_0 + \left(\sum_{k=1}^{N} x_k\right) \cdot a_1 + \left(\sum_{k=1}^{N} x_k^2\right) \cdot a_2 = \sum_{k=1}^{N} y_k$$

$$(2) \qquad \left(\sum_{k=1}^{N} x_k\right) \cdot a_0 + \left(\sum_{k=1}^{N} x_k^2\right) \cdot a_1 + \left(\sum_{k=1}^{N} x_k^3\right) \cdot a_2 = \sum_{k=1}^{N} x_k y_k$$

$$(3) \qquad \left(\sum_{k=1}^{N} x_k^2\right) \cdot a_0 + \left(\sum_{k=1}^{N} x_k^3\right) \cdot a_1 + \left(\sum_{k=1}^{N} x_k^4\right) \cdot a_2 = \sum_{k=1}^{N} x_k^2 y_k$$

Lösung von Beispiel 1 (Federkonstante (s.o.)):

Die Messung wurde in der Form $s = d \cdot F$ vorgenommen; es empfiehlt sich (wegen der günstigeren Zahlen) auch eine Ausgleichsgerade $y = A_1(x)$ mit $F = x$ und $s = y$. Die Federkonstante berechnet sich dann durch $c = 1/d$.

Ansatz $y = A_1(x) = a_0 + a_1 x$:

Koeffizienten der Normalgleichungen (Summationsgrenzen sind 1 und 6):

$$N = 6 \;,\; \sum_{k=1}^{6} x_k = 15 \;,\; \sum_{k=1}^{6} x_k^2 = 55 \;,\; \sum_{k=1}^{6} y_k = 75.5 \;,\; \sum_{k=1}^{6} x_k y_k = 276.4 \;.$$

Normalgleichungen:

$$6a_0 + 15a_1 = 75.5 \qquad (1)$$

$$15a_0 + 55a_1 = 276.4 \qquad (2)$$

Als Lösung von (1) , (2) erhält man $a_0 = 0.06$, $a_1 = 5.01$.

Ausgleichsgerade: $y = A_1(x) = 0.06 + 5.01 \cdot x$.

Die ausgeglichene Anfangsauslenkung 0.06 kann vernachlässigt werden und man erhält

$y = s = 5.01 \cdot F = 5.01 \cdot x$, also

$d = 5.01$ oder $c = \frac{1}{d} = 0.20$ N/cm .

Lösung von Beispiel 2:

Wir substituieren zweckmäßig (wegen kleinerer Zahlen x) $x = \frac{t}{5}$ und erhalten folgende Tabelle:

x	0	1	2	3	4	5	6
s = y	1000	1091	1165	1221	1260	1281	1285

Koeffizienten der Normalgleichungen für die Ausgleichsparabel

$$y = A_2(x) = a_0 + a_1 x + a_2 x^2$$

(Summationsgrenzen sind 1 bis 7):

$$N = 7 \;,\; \sum_{k=1}^{7} x_k = 21 \;,\; \sum_{k=1}^{7} x_k^2 = 91 \;,\; \sum_{k=1}^{7} x_k^3 = 441 \;,\; \sum_{k=1}^{7} x_k^4 = 2275 \;,$$

$$\sum_{k=1}^{7} y_k = 8303 \;,\; \sum_{k=1}^{7} x_k y_k = 26239 \;,\; \sum_{k=1}^{7} x_k^2 y_k = 115185 \;.$$

Normalgleichungen:

$$7a_0 + 21a_1 + 91a_2 = 8303 \qquad (1)$$

$$21a_0 + 91a_1 + 441a_2 = 26239 \qquad (2)$$

$$91a_0 + 441a_1 + 2275a_2 = 115185 \qquad (3)$$

Als Lösungen von (1), (2) und (3) erhält man (Gauß–Algorithmus)

$$a_0 = 995.95 \;,\; a_1 = 99.93 \;,\; a_2 = -8.74 \;.$$

Ausgleichspolynom: $y = A_2(x) = 999.95 + 99.93 \cdot x - 8.74 \cdot x^2$.

Umrechnen in s(t) durch $y = s$, $x = \frac{t}{5}$ ergibt

$$s = 999.95 + 19.99 \cdot t - 0.35 \cdot t^2 = s_0 + v_0 t + \frac{a}{2} \cdot t^2 \;.$$

Der Zug beginnt also (gerundet) eine Bremsung beim Punkt $s_0 = 1000$ m ,
einer Anfangsgeschwindigkeit von 20 m/sec = 72 km/h und einer Bremsbeschleunigung
von 0,70 m/sec^2 .

8. Numerische Differentiation und Integration

8.1. Numerische Differentiation

Gesucht ist ein Näherungswert für die Ableitung einer differenzierbaren Funktion f(x) an der Stelle $\bar{x}$. Dabei kann f(x) analytisch (d.h. durch einen Formelausdruck) oder durch die Stützstellen einer Wertetabelle vorgegeben sein.

Grundprinzip der numerischen Differentiation: Man approximiert f(x) in der Umgebung der Stelle $\bar{x}$ durch ein Näherungspolynom p(x) und verwendet $p'(\bar{x})$ als Näherungswert für $f'(\bar{x})$. Im folgenden werden zwei der wichtigsten Formeln zur numerischen Differentiation angegeben.

a) **Einseitige Differenz**

Man approximiert $f'(\bar{x})$ durch die Steigung der Tangente zu einem Nachbarpunkt mit

$x = \bar{x} + h$

$$f'(\bar{x}) = \frac{f(\bar{x}+h) - f(\bar{x})}{h} + r_1$$

Für den Fehler r_1 gilt nach dem Satz von Taylor $r_1 = -\frac{1}{2!} h\, f''(u)$,

u ist eine (i.a. nicht bekannte) Zwischenstelle im Intervall $[\,\bar{x}\,;\,\bar{x}+h\,]$

b) **Zweiseitige zentrale Differenz**

Nach der Taylorschen Formel gilt

$$f(\bar{x}+h) = f(\bar{x}) + h\cdot f'(\bar{x}) + \frac{h^2}{2!}\cdot f''(\bar{x}) + \frac{h^3}{3!}\cdot f'''(\bar{x}) + \ldots$$

$$f(\bar{x}-h) = f(\bar{x}) - h\cdot f'(\bar{x}) + \frac{h^2}{2!}\cdot f''(\bar{x}) - \frac{h^3}{3!}\cdot f'''(\bar{x}) + \ldots$$

Subtrahiert man beide Summen und teilt das Ergebnis durch 2, so ergibt sich

$$f'(\bar{x}) = \frac{f(\bar{x}+h) - f(\bar{x}-h)}{2h} + r_2$$

Diese Formel ist wesentlich genauer wie die Formel in a), für den Fehler r_2 gilt

$r_2 = \frac{-1}{3!} h^2 f'''(u)$, u ist eine Zwischenstelle im Intervall $[\,\bar{x}-h\,;\,\bar{x}+h\,]$

c) **Näherungsformeln für die 2. Ableitung**

Aus den Ergebnissen in a) bzw. b) lassen sich folgende Näherungen für die 2. Ableitung an der Stelle $\bar{x}$ angeben:

$$f''(\bar{x}) \approx \frac{f(\bar{x}+h) - 2\,f(\bar{x}) + f(\bar{x}-h)}{h^2} \qquad \text{bzw.}$$

$$f''(\bar{x}) \approx \frac{-f(\bar{x}+2h) + 16\,f(\bar{x}+h) - 30\,f(\bar{x}) + 16\,f(\bar{x}-h) - f(\bar{x}-2h)}{12\,h^2}$$

8.2. Grundprinzipien der numerischen Integration

Bestimmte Integrale löst man mit analytischen Verfahren über den Hauptsatz der Differential– und Integralrechnung: Man bestimmt eine Stammfunktion (unbestimmtes Integral)

$F(x) = \int f(x)\,dx$; dann berechnet sich das bestimmte Integral als Differenz

$$\int_a^b f(x)\,dx = F(b) - F(a) \quad .$$

Dieses analytische Verfahren versagt, wenn die Stammfunktion F(x) nicht als elementare Funktion existiert (z.B. $\frac{\sin x}{x}$, e^{-x^2}), oder wenn F(x) sehr kompliziert ist.

Beispiel: Für die Funktion $f(x) = \frac{1}{x^5 + x + 1}$ kann etwa F(x) analytisch berechnet werden (Faktorisierung des Nenners, Partialbruchzerlegung).

Zur Faktorisierung benötigt man jedoch die Nullstellen des Nenners; die einzige reelle Nullstelle von $x^5 + x + 1$ läßt sich aber nur über ein Iterationsverfahren bestimmen ($x_1 = -0.754877666$). Faktorisierung durch Abspalten von $x - x_1$, Bestimmung der komplexen Nullstellen, Partialbruchzerlegung und Integration führen hier zu umfangreichen und rundungsfehlerbehafteten Berechnungen.

Ziel: Näherungsverfahren für bestimmte Integrale:

(1) $$\int_a^b f(x)\,dx = Q + R$$

$Q = Q(a,b)$ ... Numerische Integrationsformel (Quadraturformel)

R ... Fehler der Formel Q (Fehler des Verfahrens) .

Grundgedanke zur Bestimmung einer Quadraturformel Q(a,b):

Man approximiere f(x) durch eine Funktion $\varphi(x)$, die einfach zu integrieren ist:

$$Q = Q(a,b) = \int_a^b \varphi(x)\,dx \quad .$$

Dazu teilt man das Intervall [a,b] in n Teilintervalle $a = x_0, x_1, x_2, \ldots, x_n = b$;

$\varphi(x)$ ist i.a. in den Teilintervallen abschnittsweise definiert. Wir beschränken uns im folgenden auf Polynome zur Approximation.

Für die Aufteilung in Teilintervalle gibt es 2 wichtige Prinzipien:

a) äquidistante Zerlegung $x_k = a + k \cdot h \quad , \quad 0 \leq k \leq n$

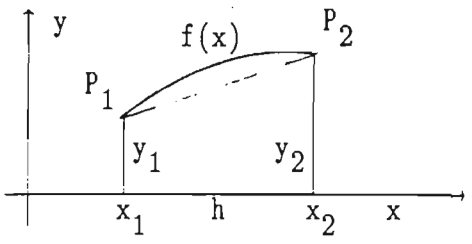
mit $x_n = a + n \cdot h = b \quad , \quad h = \frac{b - a}{n}$.

b) Zerlegung nach Kriterien optimaler Integration von Polynomen. Diese Aufteilung ist nicht mehr äquidistant und liefert z.B. die Gauß–Integration.

8.3. Grundformeln der numerischen Integration

a) Die Sehnentrapezformel

Man ersetzt die Funktionskurve $y = f(x)$ im Intervall $[x_1, x_2]$ durch die Verbindungssehne zwischen den Punkten $P_1(x_1|y_1)$ und $P_2(x_2|y_2)$. Die Fläche des entstehenden Trapezes ist ein Näherungswert Q für die Fläche zwischen Kurve f(x) und x–Achse:

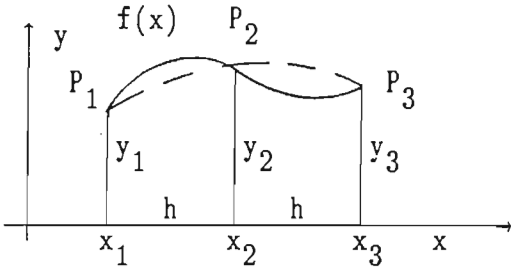


$$(2) \qquad Q = Q(x_1, x_2) = \frac{h}{2}(y_1 + y_2) \qquad \text{mit} \quad h = x_2 - x_1 \ .$$

b) Die Keplersche Faßregel

Wir betrachten 3 Punkte $P_1(x_1|y_1)$, $P_2(x_2|y_2)$ und $P_3(x_3|y_3)$ mit äquidistanten x–Werten x_1 , $x_2 = x_1 + h$, $x_3 = x_1 + 2h$ und $y_i = f(x_i)$.

Approximiert man die Kurve f(x) durch eine Parabel $p_2(x)$ durch die Punkte P_1, P_2 und P_3, so ist das Integral $Q = \int_{x_1}^{x_3} p_2(x)\,dx$ ein Näherungswert



für die Fläche zwischen Kurve f(x) und x–Achse in $[x_1, x_3]$. Das Polynom $p_2(x)$ läßt sich wie in 5.3 Beispiel 3 mit $x_i = a + (i-1) \cdot h$ bestimmen :

$$p_2(x) = y_2 + \frac{y_3 - y_1}{2h}(x - x_1 - h) + \frac{y_3 - 2y_2 + y_1}{2h^2}(x - x_1 - h)^2 \ .$$

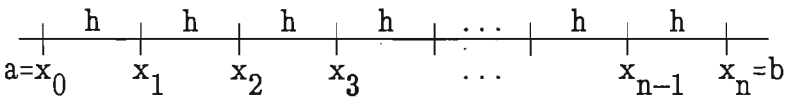
Für die Integration über dem Intervall $[x_1,x_3]$ substituiert man zweckmäßig $t = x - x_1 - h$. Die Integrationsgrenzen gehen dann über in –h und h , und es läßt sich folgendes Resultat nachrechnen:

(3) $$Q = \frac{h}{3}(y_1 + 4y_2 + y_3) \quad .$$

Diese Formel bezeichnet man als Keplersche Faßregel oder Simpson– Formel für ein Doppelintervall.

8.4. Summierte Quadraturformeln

Man zerlegt das Intervall [a,b] in n äquidistante Teilintervalle $x_k = x_0 + kh$, $0 \le k \le n$, der Länge $h = \frac{b-a}{n}$:



Dann wendet man die Grundformeln aus 8.2 an.

8.4.1. Die allgemeine Trapezregel (summierte Trapezregel)

Man wendet in jedem Teilintervall der Länge h die Trapezformel (2) an:

$$T = \frac{h}{2}[(y_0+y_1)+(y_1+y_2)+(y_2+y_3)+ \dots +(y_{n-2}+y_{n-1})+(y_{n-1}+y_n)]$$

(4) $$T = T(h) = \frac{h}{2}(y_0 + 2y_1 + 2y_2 + 2y_3 + \dots + 2y_{n-1} + y_n)$$

$$= \frac{h}{2}\left(y_0 + y_n + 2\sum_{k=1}^{n-1} y_k \right) \quad .$$

Formel (4) heißt allgemeine (oder summierte) Trapezregel.

Trapezformel bei Schrittweitenhalbierung:

Um die Güte von Näherungen mit der Trapezformel (4) beurteilen zu können, berechnet man T(h) für verschiedene Schrittweiten h. Dazu sucht man Strategien, um den Rechenaufwand klein zu halten. Hauptaufwand erfordert die Berechnung der Funktionswerte $y_k = f(x_k)$.

Eine dieser Strategien besteht in schrittweiser Halbierung der Intervalle:

Man berechnet T für eine Anfangsschrittweite $h = h_0$ (häufig $h_0 = b - a$). Dann halbiert man sukzessiv die Schrittweite. Neu berechnet werden müssen dann jeweils nur die Funktionswerte an den Halbierungsstellen.

Nach m Halbierungen beträgt die Schrittweite $h_m = h_0/2^m$. Zwischen den Trapeznäherungen $T(h_m)$ und $T(h_{m+1}) = T(h_m/2)$ besteht folgender Zusammenhang:

$$T(h_{m+1}) = \tfrac{1}{2}\cdot T(h_m) + h_{m+1}[f(a+h_{m+1}) + f(a+3h_{m+1}) + \ldots + f(b-h_{m+1})] \ .$$

In der eckigen Klammer stehen die bei der Intervallhalbierung neu zu berechnenden Funktionswerte.

Beweis: Bei Schrittweite h_{m+1} ist (wegen Halbierung) auf jeden Fall eine geradzahlige Anzahl von Teilintervallen vorhanden; diese Anzahl sei 2r.

Stützstellen: $a = x_0 , x_1 , x_2 , \ldots , x_{2r} = b$.

Trapeznäherungen:

$$\begin{aligned} T(h_m) &= \frac{h_m}{2}\cdot[y_0 + 2y_2 + 2y_4 + \ldots + 2y_{2r-2} + y_{2r}] \\ T(h_{m+1}) &= \frac{h_{m+1}}{2}\cdot[y_0 + 2y_1 + 2y_2 + 2y_3 + \ldots + 2y_{2r-2} + 2y_{2r-1} + y_{2r}] \\ &= \frac{h_m/2}{2}\cdot[y_0 + 2y_2 + 2y_4 + \ldots + 2y_{2r-2} + y_{2r}] \\ &\quad + \frac{h_{m+1}}{2}\cdot[2y_1 + 2y_3 + 2y_5 + \ldots + 2y_{2r-1}] \\ &= \tfrac{1}{2}\cdot T(h_m) + h_{m+1}[f(a+h_{m+1}) + f(a+3h_{m+1}) + \ldots + f(b-h_{m+1})] \end{aligned}$$

8.4.2. Die Simpson–Formel

Um die Keplersche Faßregel auf Teilintervalle anwenden zu können, benötigt man eine Zerlegung in $n = 2m$ Teile (gerade Zahl von Teilintervallen) $x_0 , x_1 , x_2 , x_3 , \ldots , x_{2m}$.
In jedem Doppelintervall $[x_0,x_2]$, $[x_2,x_4]$, ... $[x_{2m-2},x_{2m}]$ wendet man nun die Keplersche Faßregel (3) an:

$$S = \frac{h}{3}\cdot[(y_0+4y_1+y_2)+(y_2+4y_3+y_4)+(y_4+4y_5+y_6)+ \ldots +(y_{2m-2}+4y_{2m-1}+y_{2m})]$$

$$\begin{aligned} (5) \qquad S &= \frac{h}{3}\cdot(y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + \ldots + 2y_{2m-2} + 4y_{2m-1} + y_{2m}) \\ &= \frac{h}{3}\cdot\Big(y_0 + y_{2m} + 2\cdot\sum_{k=1}^{m-1} y_{2k} + 4\cdot\sum_{k=1}^{m} y_{2k-1} \Big) \ . \end{aligned}$$

Formel (5) bezeichnet man als Simpsonformel.

8.4.3. Beispiel: Man berechne $\int_1^2 \frac{1}{x}\,dx = \ln 2 = 0.69315$ mit Trapez– bzw. Simpsonformel.

Trapezformel mit Schrittweite $h = 0.2$:

$$T = T(0.2) = \frac{0.2}{2}\left[1 + \frac{1}{2} + 2\cdot\left(\frac{1}{1.2} + \frac{1}{1.4} + \frac{1}{1.6} + \frac{1}{1.8}\right)\right] = 0.69563$$

Trapezformel mit Schrittweite $h = 0.1$:

$$T = T(0.1) = \frac{0.1}{2}\left[1 + \frac{1}{2} + 2\cdot\left(\frac{1}{1.1} + \frac{1}{1.2} + \frac{1}{1.3} + \ldots + \frac{1}{1.9}\right)\right]$$
$$= 0.69377\,.$$

$T(0.1)$ läßt sich schneller berechnen mit der Rekursionsformel

$$T(0.1) = \frac{1}{2}\cdot T(0.2) + 0.1\cdot\left(\frac{1}{1.1} + \frac{1}{1.3} + \frac{1}{1.5} + \frac{1}{1.7} + \frac{1}{1.9}\right)$$

Simpsonformel mit der Schrittweite $h = 0.1$:

$$S = S(0.1) = \frac{0.1}{3}\left[1 + \frac{1}{2} + 2\cdot\left(\frac{1}{1.2} + \frac{1}{1.4} + \frac{1}{1.6} + \frac{1}{1.8}\right) + 4\left(\frac{1}{1.1} + \frac{1}{1.3} + \frac{1}{1.5} + \frac{1}{1.7} + \frac{1}{1.9}\right)\right]$$
$$= 0.69315$$

8.5. Fehlerverhalten

Bei Verkleinerung von h nimmt die Güte der Näherungsformeln (4) bzw. (5) zu, d.h. für den Fehler R in (1) gilt $\lim_{h\to 0} R = 0$.

Fehlerabschätzungen (vgl. [8]) liefern für

die summierte Trapezregel (4) $\quad R \le c\cdot h^2$

die Simpsonformel (5) $\quad R \le c\cdot h^4$

(f(x) wird dabei als 2 bzw. 4 mal stetig differenzierbar vorausgesetzt).

Die Potenz der Schrittweite h bezeichnet man als "Ordnung des Verfahrens"; die summierte Trapezregel ist also ein Verfahren der Ordnung 2, die Simpsonformel ein Verfahren der Ordnung 4.

Die Qualität eines Näherungsverfahrens wächst mit seiner Ordnung.

Rundungsfehler sind bei diesen Abschätzungen nicht berücksichtigt.

8.6. Das Romberg – Verfahren

8.6.1. Grundlagen

Ausgehend vom Trapezverfahren hat Romberg 1955 ein Verfahren angegeben, das beim n–ten Schritt die Fehlerordnung 2n+2 besitzt (entspricht h^{2n+2}). Grundlage für das Romberg–Verfahren ist eine Formel, die den Fehler bei der summierten Trapezregel beschreibt:

Satz (Euler – Mac–Laurinsche Summenformel):

Für eine (2m+2) mal stetig differenzierbare Funktion f(x) gilt für den Fehler E(h) der summierten Trapezformel T(h) aus 8.3.1

$$E(h) = T(h) - \int_a^b f(x)\,dx = T(h) - I$$

$$= c_1 h^2 + c_2 h^4 + c_3 h^6 + \ldots + c_m h^{2m} + \alpha_{m+1}(h)\cdot h^{2m+2} \quad .$$

Dabei sind die Koeffizienten c_k konstante, von h unabhängige Zahlen, die nur von dem Verhalten der Funktion f(x) an den Intervallenden a, b abhängen:

$$c_k = \frac{B_{2k}}{(2k)!}\cdot\left[\, f^{(2k-1)}(b) - f^{(2k-1)}(a)\,\right] \quad , \qquad k = 1, 2, 3, \ldots, m \quad .$$

Die Zahlen B_{2k} sind sog. "Bernoullische Zahlen" mit bekanntem (aber kompliziertem) Bildungsgesetz; z.B. gilt

$$B_2 = \frac{1}{6} \quad , \quad B_4 = B_8 = -\frac{1}{30} \quad , \quad B_6 = \frac{1}{42} \quad , \quad \ldots \quad .$$

Die Funktion $\alpha_{m+1}(h)$ ist beschränkt in [a,b], d.h. es gilt

$$|\alpha_{m+1}(h)| \leq K \quad .$$

Folgende Idee führt nun auf ein Integrationsverfahren höherer Ordnung:

Man berechne T(h) für verschiedene Schrittweiten h und kombiniere die Ergebnisse so, daß der Fehler E(h) von höherer Ordnung wird (Ordnung 4 , 6 , 8 , ...).

Man geht dazu von einer Anfangsschrittweite h_0 aus (üblich ist $h_0 = b - a$). Für andere Schrittweiten h soll nun bei möglichst geringem Rechenaufwand T(h) berechnet werden (Hauptaufwand: Berechnung der Funktionswerte von f(x) an den Stützstellen).
Romberg schlägt dazu schrittweise Halbierung des Ausgangsintervalls vor:

$$h_1 = h_0/2 \quad , \quad h_2 = h_1/2 = h_0/4 \quad , \quad \ldots \quad , \quad h_k = h_0/2^k \qquad \text{("Romberg– Folge")}$$

Für die Trapezregel gilt dann nach 8.3.1 die Rekursionsformel

$$T(h_k) = \frac{1}{2}\cdot T(h_{k-1}) + h_k\cdot[f(a+h_k) + f(a+3h_k) + f(a+5h_k) + \ldots + f(b-h_k)]$$

8.6.2. Beschreibung des Romberg–Verfahrens

Es sei $h = h_0$. Dann berechnet man $T(h) \doteq R_0(h)$.

Die Euler–Mac–Laurinsche Summenformel ergibt:

$$R_0(h) = T(h) = I + c_1h^2 + c_2h^4 + c_3h^6 + \ldots + c_mh^{2m} + \ldots \qquad (1a).$$

1. Rombergschritt:

Man halbiert die Schrittweite h und berechnet $T(\frac{h}{2})$.

Euler–Mac–Laurinsche Summenformel:

$$T(\tfrac{h}{2}) = I + c_1(\tfrac{h}{2})^2 + c_2(\tfrac{h}{2})^4 + c_3(\tfrac{h}{2})^6 + \ldots + c_m(\tfrac{h}{2})^{2m} + \ldots \qquad (1b)$$

Aus (1a) und (1b) eliminiert man h^2: $4\cdot(1b)-(1a) \rightarrow$

$$4\cdot T(\tfrac{h}{2}) - T(h) = 3I + c_2h^4(\tfrac{1}{4}-1) + c_3h^6(\tfrac{1}{16}-1) + \ldots$$
$$= 3I + (-\tfrac{3}{4}c_2)h^4 + (-\tfrac{15}{16}c_3)h^6 + \ldots$$

Division durch 3 liefert ein Verfahren zur Bestimmung von I mit Fehlerordnung 4 (ohne daß zusätzliche Funktionswerte berechnet worden sind !):

$$R_1(h) = \frac{4\cdot T(\frac{h}{2}) - T(h)}{3} = I + (-\tfrac{1}{4}c_2)h^4 + (-\tfrac{5}{16}c_3)h^6 + \ldots$$
$$= I + \bar{c}_2h^4 + \bar{c}_3h^6 + \bar{c}_4h^8 + \ldots \qquad (2a)$$

2. Rombergschritt:

Weitere Halbierung des Intervalls ergibt die Schrittweite $\frac{h}{4}$; zu dieser berechnet man $T(\frac{h}{4})$.

Dann ergibt die Euler–Mac–Laurinsche Summenformel:

$$T(\tfrac{h}{4}) = I + c_1(\tfrac{h}{4})^2 + c_2(\tfrac{h}{4})^4 + c_3(\tfrac{h}{4})^6 + \ldots + c_m(\tfrac{h}{4})^{2m} + \ldots \qquad (1c)$$

$4\cdot(1c)-(1b)$ und Division durch 3 ergibt wie beim 1. Rombergschritt

$$R_1(\tfrac{h}{2}) = \frac{4\cdot T(\frac{h}{4}) - T(\frac{h}{2})}{3} = I + \bar{c}_2(\tfrac{h}{2})^4 + \bar{c}_3(\tfrac{h}{2})^6 + \bar{c}_4(\tfrac{h}{2})^8 + \ldots \qquad (2b)$$

Aus (2a) und (2b) läßt sich nun h^4 eliminieren durch $4^2\cdot(2b)-(2a)$:

$$4^2\cdot R_1(\tfrac{h}{2}) - R_1(h) = (4^2-1)I + \bar{c}_3h^6(\tfrac{1}{4}-1) + \bar{c}_4h^8(\tfrac{1}{16}-1) + \ldots$$

Division durch (4^2-1) liefert ein Verfahren der Ordnung 6 für I :

$$R_2(h) = \frac{4^2\cdot R_1(\frac{h}{2}) - R_1(h)}{4^2 - 1} = I + (-\tfrac{1}{20}\bar{c}_3)h^6 + (-\tfrac{1}{16}\bar{c}_4)h^8 + \ldots$$
$$= I + \bar{\bar{c}}_3h^6 + \bar{\bar{c}}_4h^8 + \ldots \qquad (3a)$$

Analog rechnet man weitere Rombergschritte. Beim n–ten Rombergschritt (n = 1, 2, 3, ...) erhält man eine Schrittweite $h/2^n$ für das Trapezverfahren und eine Rombergverbesserung $R_n(h)$ mit Fehlerordnung 2(n+1) .

Die Rekursionsformel für den n–ten Rombergschritt lautet:

$$R_n(h) = \frac{4^n \cdot R_{n-1}(\frac{h}{2}) - R_{n-1}(h)}{4^n - 1}$$

Das Verfahren läßt sich an einem "Rombergtableau" verdeutlichen:

Romberg – Tableau:

Schritt–weite	Fehlerordnung 2	4	6	8
h	$T(h) = R_0(h)$			
		$R_1(h) = \frac{4R_0(\frac{h}{2}) - R_0(h)}{4-1}$		
$\frac{h}{2}$	$T(\frac{h}{2}) = R_0(\frac{h}{2})$		$R_2(h) = \frac{4^2 R_1(\frac{h}{2}) - R_1(h)}{4^2-1}$	
		$R_1(\frac{h}{2}) = \frac{4R_0(\frac{h}{4}) - R_0(\frac{h}{2})}{4-1}$		$R_3(h) = \frac{4^3 R_2(\frac{h}{2}) - R_2(h)}{4^3-1}$
$\frac{h}{4}$	$T(\frac{h}{4}) = R_0(\frac{h}{4})$		$R_2(\frac{h}{2}) = \frac{4^2 R_1(\frac{h}{4}) - R_1(\frac{h}{2})}{4^2-1}$	
		$R_1(\frac{h}{4}) = \frac{4R_0(\frac{h}{8}) - R_0(\frac{h}{4})}{4-1}$		
$\frac{h}{8}$	$T(\frac{h}{8}) = R_0(\frac{h}{8})$		usw.	

Berechnet werden jeweils Schrägzeilen des Romberg–Tableaus.

8.6.3. Beispiele

Beispiel 1: $\int_1^2 \frac{1}{x^2}\,dx = \frac{1}{2}$ (zur Demonstration)

Schritt-weite	Rombergtableau (9 Dezimalstellen)			
1	0.625000000			
		0.504629630		
$\frac{1}{2}$	0.534722222		0.500136810	
		0.500417612		0.500001923
$\frac{1}{4}$	0.508993764		0.500004030	0.500000011
		0.500029879		0.500000018
$\frac{1}{8}$	0.502270850		0.500000081	
		0.500001943		
$\frac{1}{16}$	0.500569170			

Beim nächsten (d.h. 5–ten) Rombergschritt ergibt sich das richtige Ergebnis in allen 9 Dezimalstellen.

Beispiel 2: $\int_0^1 \frac{dx}{x^3 + x + 1}$

Rombergtableau bei 9 Dezimalstellen für Schrittweiten $1, \frac{1}{2}, \frac{1}{4}, \ldots \frac{1}{32}$

0.666666667

0.632478633

0.641025641 0.630396803

0.630526918 0.630320397

0.633151599 0.630321591 0.630319335

0.630334424 0.630319339 0.630319323

0.631038718 0.630319374 0.630319323

0.630320315 0.630319323

0.630499916 0.630319324

0.630319386

0.630364518

8.6.4 Bemerkungen

1) Bei jedem Rombergschritt wird eine Schrägzeile des Romberg – Tableaus berechnet. Überschreibt man hier Zwischenergebnisse, die für die weitere Berechnung nicht mehr benötigt werden, so genügt zur Speicherung ein eindimensionales Feld R:

Romberg–schritt		1	2	3
Romberg–ergebnisse	$R_0(h)$	$R_0(\frac{h}{2})$ $R_1(h)$	$R_0(\frac{h}{4})$ $R_1(\frac{h}{2})$ $R_2(h)$	$R_0(\frac{h}{8})$ $R_1(\frac{h}{4})$ $R_2(\frac{h}{2})$ $R_3(h)$
Feldelement	R(1)	R(2) R(1)	R(3) R(2) R(1)	R(4) R(3) R(2) R(1)

usw.

2) Man bricht das Rombergverfahren entweder nach einer bestimmten Anzahl von Halbierungen ab oder wenn sich die Rombergnäherungen im Rahmen einer bestimmten Genauigkeit nicht mehr unterscheiden.

3) Die 1. Rombergnäherung R_1 entspricht der Simpson–Formel S,
und zwar gilt $R_1(h) = S(\frac{h}{2})$.
Dies läßt sich leicht nachrechnen:

m Teilintervalle mit Schrittweite $h = \frac{b - a}{m}$ → $T(h)$

2m Teilintervalle mit Schrittweite $\frac{h}{2}$ → $T(\frac{h}{2})$.

Stützstellen: $x_i = a + i \cdot \frac{h}{2}$, $i = 0, 1, 2, \ldots, 2m$

mit $x_0 = a$, $x_{2m} = b$.

Funktionswerte an den Stützstellen: $y_i = f(x_i)$.

$$T(h) = \frac{h}{2} \cdot [y_0 + 2y_2 + 2y_4 + \ldots + 2y_{2m-2} + y_{2m}]$$

$$T(\tfrac{h}{2}) = \frac{h}{4} \cdot [y_0 + 2y_1 + 2y_2 + 2y_3 + 2y_4 + \ldots + 2y_{2m-2} + 2y_{2m-1} + y_{2m}]$$

$$R_1(h) = \frac{4 \cdot T(\frac{h}{2}) - T(h)}{3}$$

$$= \frac{h}{6} \cdot [y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + \ldots + 2y_{2m-2} + 4y_{2m-1} + y_{2m}]$$

$$= S(\tfrac{h}{2}) .$$

4) Jede der Formeln R_k integriert ein Polynom vom Grad 2k+1 exakt, d.h. für

$$p(x) = a_{2k+1}x^{2k+1} + a_{2k}x^{2k} + \ldots + a_1x + a_0 \quad \text{gilt}$$

$$R_k(h) = \int_a^b p(x)\,dx \quad .$$

Insbesondere integriert die Trapezregel lineare Funktionen exakt, die Simpsonformel Polynome bis zum Grad 3.

5) Die Euler–Mac–Laurinsche Summenformel stellt eine Entwicklung der Trapezformel nach geradzahligen Potenzen von h dar. Man kann nun durch die "Stützpunkte" $(h\,|\,T(h))$ und $(\frac{h}{2}\,|\,T(\frac{h}{2}))$ ein Interpolationspolynom $p_2(t)$ mit geradzahligen Potenzen von t legen:

$$p_2(t) = a_0 + a_1t^2 \quad .$$

Die beiden Stützpunkte liefern die Bedingungen

$$p_2(h) = a_0 + a_1h^2 = T(h)$$
$$p_2(\tfrac{h}{2}) = a_0 + a_1(\tfrac{h}{2})^2 = T(\tfrac{h}{2})$$

Diese 2 linearen Gleichungen für a_0 und a_1 besitzen die Lösung

$$a_1 = 4\cdot[T(h) - T(\tfrac{h}{2})]/(3h^2) \quad , \quad a_0 = \frac{4\cdot T(\frac{h}{2}) - T(h)}{3} \quad .$$

Dieses Interpolationspolynom $p_2(t)$ hat an der Stelle 0 den "Extrapolationswert" (0 liegt außerhalb des Intervalls $[\frac{h}{2}\,;\,h]$)

$$p_2(0) = a_0 + a_1\cdot 0 = a_0 = \frac{4\cdot T(\frac{h}{2}) - T(h)}{3} = R_1(h)\,.$$

Das Verfahren bezeichnet man deshalb auch als "Richardson – Extrapolation".
Die weiteren Rombergformeln lassen analog durch Extrapolation entsprechender Interpolationspolynome (geradzahliger Potenzen) durch die Stützpunkte $(h\,|\,T(h))$, $(\frac{h}{2}\,|\,T(\frac{h}{2}))$, $(\frac{h}{4}\,|\,T(\frac{h}{4}))$ ... herleiten.

6) Neben der Rombergfolge h , $\frac{h}{2}$, $\frac{h}{4}$, $\frac{h}{8}$, ... lassen sich noch numerisch günstigere Folgen angeben; vgl. dazu [8] .

8.7. Gauß – Integration

Formeln zur numerischen Integration (vgl. 8.2 bis 8.5) haben prinzipiell folgende Bauart:

$$\sum_{k=1}^{n} w_k f(x_k) \quad ;$$

x_k sind dabei n Stützstellen bei n–1 Teilintervallen. Die Größen w_k bezeichnet man als Gewichte.

Beispiel: Summierte Trapezformel

$$w_1 = w_n = \frac{h}{2} \ , \ w_2 = w_3 = \ldots = w_{n-1} = h$$

bei n äquidistanten Stützstellen

$$x_k = a + (k-1)\cdot h \ , \ h = \frac{b-a}{n-1} \ , \ k = 1, 2, \ldots, n \ .$$

Bisher waren die Stützstellen von vornherein im Ansatz des Verfahrens festgelegt worden (äquidistant), bestimmt wurden dazu entsprechende Gewichte. Bei der Gauß–Integration werden nun Gewichte und Stützstellen bestimmt.

Allgemeine Formulierung des Problems für n Stützstellen x_k:

$$(*) \qquad I = \int_a^b f(x)\,dx = \sum_{k=1}^{n} w_k f(x_k) + R \quad ;$$

zu bestimmen sind n Gewichte w_k und n Stützstellen x_k. R ist der Fehler des Verfahrens.

Nach dem Satz vom Interpolationspolynom (5.2) benötigt man zur Festlegung eines IPs vom Grad m genau m+1 Bedingungen; diese wurden dort allein durch Funktionswerte an den Stützstellen realisiert. Hieraus abgeleitet wird folgende

Forderung von Gauß: Stützstellen x_k und Gewichte w_k sind so zu wählen, daß Polynome bis zum Grad 2n–1 mit der Quadraturformel (*) exakt integriert werden ($R = 0$).

Beispiel: Eine Stützstelle ($n = 1$)

Ansatz: $\int_a^b f(x)\,dx = w_1 f(x_1) + R$

Gesucht: w_1, x_1 .

Exakt integriert werden (R = 0) soll eine lineare Funktion

$$f(x) = p_1(x) = cx + d \quad (c, d \text{ beliebig}) .$$

$$\int_a^b (cx + d)\,dx = [c\cdot\frac{x^2}{2} + d\cdot x]_a^b = c\cdot\frac{b^2 - a^2}{2} + d\cdot(b - a)$$

$$\overset{!}{=} w_1 f(x_1) = w_1\cdot(cx_1 + d) = c\cdot w_1 x_1 + d\cdot w_1$$

für beliebige c, d. Koeffizientenvergleich für c, d ergibt 2 Bedingungen für w_1 und x_1:

$$w_1 = b - a \qquad \text{(Intervall–Länge)}$$

$$w_1 x_1 = \frac{1}{2}(b^2 - a^2) \qquad \rightarrow$$

$$x_1 = \frac{1}{2}(b + a) \qquad \text{(Mittelpunkt des Intervalls [a,b])} .$$

Diese Formeln lassen sich auch einfach dadurch bestimmen, daß man exakte Integration für die Potenzfunktionen $x^0 = 1$ und $x^1 = x$ fordert und die entsprechenden Bedingungen auswertet.

Zur Herleitung von Formeln für allgemeines n transformiert man das Ausgangsintegral $\int_a^b f(x)\,dx$ in ein Integral mit den Grenzen -1 und 1.

Substitution: $$t = \frac{2}{b - a}\cdot(x - \frac{a + b}{2}) \text{ bzw. } x = \frac{(b - a)t + b + a}{2}$$

$$dx = \frac{b - a}{2}\,dt \;,\; f(x) = f(\frac{(b - a)t + b + a}{2}) \doteq g(t) .$$

Es ergibt sich folgendes Integral:

$$I = \int_a^b f(x)\,dx = \frac{b - a}{2}\cdot\int_{-1}^{1} g(t)\,dt = \frac{b - a}{2}\cdot I_1 \text{ mit } I_1 = \int_{-1}^{1} g(t)\,dt .$$

<u>Formeln zur Gauß–Integration</u>: Bestimmt werden Näherungsformeln G_n für das Integral

$I_1 = \int_{-1}^{1} g(t)\,dt = G_n + R_n$ nach dem Gaußschen Prinzip. Dazu werden für Zahlen

n = 1, 2, 3, ... die Stützstellen t_i und entsprechenden Gewichte w_i (i=1, ..., n) bestimmt.

n = 1 (eine Stützstelle):

Gefordert wird exakte Integration für $t^0 = 1$ und $t^1 = t$.

1: $\int_{-1}^{1} 1\,dt = 2 = w_1 \cdot 1 \quad \rightarrow \quad w_1 = 2$ (Intervall–Länge)

t: $\int_{-1}^{1} t\,dt = 0 = w_1 t_1 \quad \rightarrow \quad t_1 = 0$ (Intervallmitte)

Näherungsformel G_1 für $n = 1$: $\quad G_1 = 2g(0)$

Das Ergebnis entspricht dem obigen Beispiel.

n = 2 (2 Stützstellen):

Ansatz: $\quad I_1 = \int_{-1}^{1} g(t)\,dt = G_2 + R_2$.

mit $\quad G_2 = w_1 g(t_1) + w_2 g(t_2)$.

Gefordert wird exakte Integration ($R_2 = 0$) für die Potenzen
$t^0 = 1$, t , t^2 und t^3 .

1: $\int_{-1}^{1} 1\,dt = 2 = w_1 + w_2$ (1)

t: $\int_{-1}^{1} t\,dt = 0 = w_1 t_1 + w_2 t_2$ (2)

t^2: $\int_{-1}^{1} t^2 dt = \frac{2}{3} = w_1 t_1^2 + w_2 t_2^2$ (3)

t^3: $\int_{-1}^{1} t^3 dt = 0 = w_1 t_1^3 + w_2 t_2^3$ (4)

Man kann zeigen, daß die 4 nichtlinearen Gleichungen (1) bis (4) eindeutig lösbar sind.

Die Lösungen lauten

$w_1 = w_2 = 1 \quad , \quad t_2 = \frac{1}{\sqrt{3}} = 0.57735\,02692 \quad , \quad t_1 = -t_2$.

Allgemein läßt sich zeigen:

Satz: Die Stützstellen t_k liegen symmetrisch zum Nullpunkt, bei ungeradem n ist $t = 0$ eine der Stützstellen. In symmetrischen Stützstellen sind die Gewichte gleich.

Stützstellen und Gewichte lassen sich aus Tabellen in mathematischen Handbüchern entnehmen. Für Werte von n zwischen 1 und 5 erhält man etwa

n	Gewichte w_k	Stützstellen t_k
1	$w_1 = 2$	$t_1 = 0$
2	$w_1 = w_2 = 1$	$t_2 = -t_1 = 0.57735\ 02692$
3	$w_1 = w_3 = \frac{5}{9} = 0.55555\ 55556$	$t_3 = -t_1 = 0.77459\ 66692$
	$w_2 = \frac{8}{9} = 0.88888\ 88889$	$t_2 = 0$
4	$w_1 = w_4 = 0.34785\ 48451$	$t_4 = -t_1 = 0.86113\ 63116$
	$w_2 = w_3 = 0.65214\ 51549$	$t_3 = -t_2 = 0.33998\ 10436$
5	$w_1 = w_5 = 0.23692\ 68851$	$t_5 = -t_1 = 0.90617\ 98459$
	$w_2 = w_4 = 0.47862\ 86705$	$t_4 = -t_2 = 0.53846\ 93101$
	$w_3 = 0.56888\ 88889$	$t_3 = 0$

Beispiel:

$$I = \int_0^1 \frac{dx}{x^3 + x + 1}$$ (vgl. entsprechendes Beispiel beim Rombergverfahren)

Schritt 1: Umrechnung in ein Integral über $[-1;1]$:

$a = 0$, $b = 1 \rightarrow x = \frac{t+1}{2}$ bzw. $t = 2\cdot(x - \frac{1}{2})$.

$$f(x) = \frac{1}{x^3 + x + 1} = \frac{1}{(\frac{t+1}{2})^3 + \frac{t+1}{2} + 1} = \frac{8}{t^3 + 3t^2 + 7t + 13}$$

$$I = \int_0^1 f(x)\,dx = \frac{1}{2}\int_{-1}^1 \frac{8}{t^3 + 3t^2 + 7t + 13}\,dt = 4\int_{-1}^1 \frac{dt}{t^3 + 3t^2 + 7t + 13} .$$

Zu berechnen ist mit den Gauß–Formeln das Integral

$$\int_{-1}^1 \frac{dt}{t^3 + 3t^2 + 7t + 13} .$$

Ergebnisse für

n = 1 :	$G_1 = 2\cdot g(0) = \frac{2}{13}$	$I = \frac{8}{13} = 0.615384615$
n = 2 :	$G_2 = g(-1/\sqrt{3}) + g(1/\sqrt{3}) = 0.157237937$	$I = 0.628951747$
n = 3 :	$G_3 = 0.157561922$	$I = 0.630247688$
n = 4 :	$G_4 = 0.157579234$	$I = 0.630316935$
n = 5 :	$G_5 = 0.157579777$	$I = 0.630319109$

Das Ergebnis stimmt auf 6 Dezimalstellen.

Bemerkungen:

1) Das Gaußsche Verfahren liefert im Vergleich zum Rechenaufwand die genauesten Resultate der numerischen Integrationsverfahren, die in 8.2 bis 8.6 beschrieben wurden.

2) Die Stützstellen t_k ergeben sich als Nullstellen der "Legendre – Polynome" (siehe Literatur, z.B. [8]).

3) Die Methode der Gauß–Integration läßt sich auf Integrale der Form $\int_a^b w(x)f(x)\,dx$ mit speziellen Gewichtsfunktionen $w(x)$ erweitern. Dadurch können sogar spezielle Typen uneigentlicher Integrale numerisch ausgewertet werden (siehe z.B. [8],[12])

9. Differentialgleichungen

9.1. Anfangs– und Randwertprobleme

Gewöhnliche Differentialgleichung n–ter Ordnung:

$F(x, y(x), y'(x), ..., y^{(n)}(x)) = 0$ (implizite Darstellung)

$y^{(n)}(x) = f(x, y(x), ..., y^{(n-1)}(x))$ (explizite Darstellung).

Gesucht sind Lösungsfunktionen $y(x)$.

Lösungstheorie: Unter bestimmten Voraussetzungen existieren Lösungen einer Differentialgleichung. Die Mathematik stellt dafür Existenz– und Eindeutigkeitssätze zur Verfügung. Bei der allgemeinen Lösung einer DGL n–ter Ordnung ist i.a. mit n frei wählbaren Parametern zu rechnen. Es gibt für eine Vielzahl von Differentialgleichungen analytische Verfahren zur Bestimmung der allgemeinen Lösung (v.a. für lineare DGLen).
Bei vielen Differentialgleichungen existiert allerdings in den meisten Fällen kein analytischer Lösungsweg mehr; häufig ist eine analytische Lösung auch recht kompliziert und unpraktikabel. Hier benötigt man numerische Verfahren.
Numerische Verfahren lassen sich bei einer eindeutig bestimmten Lösungskurve anwenden und liefern zu vorgegebenen x–Werten die y–Werte der Lösungskurve. Eine eindeutig bestimmte Lösungskurve einer DGL ergibt sich durch zusätzliche Vorgabe von Anfangs– bzw. Randbedingungen. Wir beschränken uns im folgenden auf explizit vorgegebene DGLen.
Man unterscheidet

a) Anfangswertprobleme (AWP)

DGL: $y^{(n)}(x) = f(x, y(x), ..., y^{(n-1)}(x))$ (1)

Anfangsbedingungen an einer Stelle x_0:

$y(x_0) = c_1 \ , \ y'(x_0) = c_2 \ , \ ... \ , \ y^{(n-1)}(x_0) = c_n$ (2)

Beispiele: 1) AWP für DGL 1. Ordnung $y' = f(x,y) \ , \ y(x_0) = y_0$

Anwendungsbeispiel: Einschwingvorgang an einem seriellen R–L–Glied.

Für den Strom i(t) gilt $L \cdot \frac{di}{dt} + R \cdot i = u_0(t) \ , \ i(0) = i_0$.

2) AWP für DGL 2. Ordnung $\quad y'' = f(x,y,y') \;,\; y(x_0) = c$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad y'(x_0) = d \;.$

Anwendungsbeispiel: Für die Biegelinie eines fest eingespannten Balkens gilt

$$y'' = \frac{-M_y(x)}{E \cdot I_y(x)} \cdot [1 + y'(x)^2]^{3/2} \qquad\qquad y(x_0) = c \;,\; y'(x_0) = d \;.$$

b) **Randwertprobleme (RWP)**

Bei Differentialgleichungen der Ordnung $n \geq 2$ werden Bedingungen an mindestens 2 Stellen $x_1 = a \;,\; x_2 = b$ vorgeschrieben.

Beispiel: RWP für eine DGL 2. Ordnung:

$$y'' = f(x,y,y') \quad,\quad y(a) = c \quad,\quad y(b) = d \;.$$

Anwendungsbeispiel: Biegelinie eines in P(a|c) und Q(b|d) aufliegenden Balkens. Zusätzlich zur DGL in a), Beispiel 2 gelten die obigen Randbedingungen.

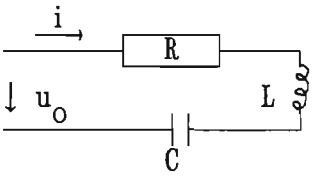
Bemerkung: Anfangs– bzw. Randwertprobleme lassen sich manchmal nur unter bestimmten Bedingungen (d.h. für bestimmte "Eigenwerte") lösen.
Wir beschränken uns im folgenden auf Verfahren zur Lösung von Anfangswertproblemen.
Numerische Verfahren zur Behandlung von Rand– und Eigenwertproblemen siehe Literatur.

Systeme (1. Ordnung) von Differentialgleichungen

Als (gekoppeltes) System bezeichnet man die folgenden n Differentialgleichungen 1. Ordnung zur Bestimmung der Funktionen $y_1(x)$ bis $y_n(x)$:

$$\left.\begin{aligned} y_1'(x) &= f_1(x,y_1(x),y_2(x),\ldots,y_n(x)) \\ y_2'(x) &= f_2(x,y_1(x),y_2(x),\ldots,y_n(x)) \\ &\ldots \\ y_n'(x) &= f_n(x,y_1(x),y_2(x),\ldots,y_n(x)) \end{aligned}\right\} \quad (3)$$

Beispiel: Bei einem Reihenschwingkreis ergibt sich folgendes DGL–System für i(t) und $u_c(t)$:



$$L \cdot \frac{di}{dt} + R \cdot i + u_c = u_o(t)$$

$$i - C \cdot \frac{du_c}{dt} = 0 \quad .$$

Numerische Verfahren für Anfangswertprobleme bei Differentialgleichungen 1. Ordnung lassen sich leicht auf DGL–Systeme übertragen (siehe 9.4).

Differentialgleichungen höherer Ordnung rechnet man durch Einführen zusätzlicher Funktionen in DGL–Systeme um:

DGL n–ter Ordnung: $y^{(n)}(x) = f(x, y(x), ..., y^{(n-1)}(x))$.

Substitution:

$$y_1(x) \doteq y(x)$$
$$y_2(x) \doteq y'(x) = y_1'(x)$$
$$y_3(x) \doteq y''(x) = y_2'(x)$$
$$\dots$$
$$y_{n-1}(x) \doteq y^{(n-2)}(x) = y_{n-2}'(x)$$
$$y_n(x) \doteq y^{(n-1)}(x) = y_{n-1}'(x)$$

DGL:
$$y_n'(x) = y^{(n)}(x) = f(x, y_1, y_2, \dots, y_n)$$

Dies führt zu folgendem DGL–System für $y_1(x)$ bis $y_n(x)$:

$$\left.\begin{aligned} y_1'(x) &= y_2(x) \\ y_2'(x) &= y_3(x) \\ &\dots \\ y_{n-2}'(x) &= y_{n-1}(x) \\ y_{n-1}'(x) &= y_n(x) \\ y_n'(x) &= f(x, y_1, y_2, \dots, y_n) \end{aligned}\right\} \quad (4)$$

Für n = 2 verwendet man häufig eine vereinfachte Schreibweise:

DGL: $y'' = f(x,y,y')$

Man setzt $z(x) = y'$ und erhält wegen $y'' = z'$ folgendes DGL–System:

$$\left.\begin{aligned} y'(x) &= z(x) \\ z'(x) &= f(x,y(x),z(x)) \end{aligned}\right\} \quad (4a)$$

Anfangsbedingungen rechnet man entsprechend um,

z.B. $y(x_0) = c\ ,\ y'(x_0) = d$ ergibt

$y(x_0) = c\ ,\ z(x_0) = d\ .$

9.2. Aufbau von Näherungsverfahren bei Anfangswertproblemen

DGL 1. Ordnung: Ausgehend vom Startpunkt $(x_0 | y_0 = y(x_0))$ werden an Fortschreitstellen x_k Näherungswerte y_k für die Lösung $y(x_k)$ berechnet. Wir beschränken uns im folgenden auf äquidistante Werte x_k :

$x_k = x_0 + k \cdot h \quad , \quad k = 0, 1, 2, \ldots \quad ;$

die Schrittweite h muß vorgegeben werden. Man unterscheidet:

a) Einschrittverfahren: Zur Berechnung von y_{k+1} werden nur Daten der vorhergehenden Stelle x_k verwendet (schrittweises Rechnen von x_0 zu $x_1, x_2, \ldots$) .

b) Mehrschrittverfahren: Es werden Daten mehrerer (oder aller) vorhergehenden Stellen in die Berechnung einbezogen.

Wir beschränken uns im folgenden auf Einschrittverfahren.

9.3. Einschrittverfahren für Anfangswertprobleme bei Differentialgleichungen 1. Ordnung

$y' = f(x,y(x))$, $y(x_0) = y_0$

Vorgegeben sei weiter eine Schrittweite h.

Gesucht sind Näherungswerte $y_k \approx y(x_k)$ an den Stellen $x_k = x_0 + kh$.

9.3.1 Das Euler–Cauchysche Polygonzugverfahren

Man ersetzt im Punkt $P_k(x_k|y_k)$, $k = 0, 1, 2, \ldots$, die Kurve durch ihre Tangente.

Deren Steigung ist aus der DGL bekannt ($m = y'$). Dies entspricht einer Linearisierung der Kurve $y(x)$ an der Stelle x_k :

$$y(x) \approx y(x_k) + y'(x_k)\cdot(x - x_k) \quad \text{mit} \quad y'(x_k) = f(x_k,y_k) \ .$$

Für $x_{k+1} = x_k + h$ ergibt sich folgendes Näherungsverfahren

(5) $y_{k+1} = y_k + h\cdot f(x_k,y_k)$, $k = 0, 1, 2, \ldots$

Die Kurve wird bei diesem Verfahren durch einen Polygonzug ersetzt; das Verfahren bezeichnet man daher als Euler–Cauchysches Polygonzugverfahren.

Beispiel: $y' = f(x,y) = -xy$, $y(0) = 1$.

Gesucht sei ein Näherungswert für $y(1)$.

Rechnung mit Schrittweite $h = 0.2$ ergibt

n	x_n	y_n
0	0	1
1	0.2	1
2	0.4	0.960000
3	0.6	0.883200
4	0.8	0.777216
5	1.0	0.652861

Rechnung mit $h = 0.1$ ergibt den Näherungswert 0.628156 für $y(1)$.

(zum Vergleich: die exakte Lösung ist $y(x) = e^{-\frac{1}{2}\cdot x^2}$;

Lösungswert: $y(1) = e^{-0.5} = 0.606531$)

9.3.2. Das Verfahren von Heun

Idee: Man integriere die DGL und wende auf das Integral über f(x,y(x)) ein Näherungsverfahren an (beim Heun–Verfahren die Trapezformel).

Berechnung für das Intervall $[x_k,x_{k+1}]$ ergibt: $y' = f(x,y(x))$

$$\int_{x_k}^{x_{k+1}} y'(x)\,dx = y(x_{k+1}) - y(x_k) = \int_{x_k}^{x_{k+1}} f(x,y(x))\,dx$$

Als Startwert beim k–ten Schritt wird $y(x_k) \approx y_k$ verwendet. Die Trapeznäherung für das Integral über f liefert einen Näherungswert y_{k+1} für $y(x_{k+1})$. Mit $x_{k+1} - x_k = h$ ergibt sich

$$y_{k+1} - y_k = \frac{h}{2}\cdot[f(x_k,y_k) + f(x_{k+1},y_{k+1})] \quad .$$

Diese Beziehung wäre nach der gesuchten Variablen y_{k+1} aufzulösen. Dies ist nur in Sonderfällen möglich. Man ändert daher das Verfahren ab:

Man bestimme durch eine "Anlaufrechnung" einen Näherungswert y^*_{k+1} für y_{k+1}, diesen setze man auf der rechten Seite in den Ausdruck $f(x_{k+1},y_{k+1})$ ein. Zur Berechnung von y^*_{k+1} verwendet man das Euler–Cauchy–Verfahren. Ergebnis ist das

Verfahren von Heun: Man berechne für $k = 0, 1, 2, \ldots$

Schritt 1 (Anlaufrechnung):

(6a) $$y^*_{k+1} = y_k + h\cdot f(x_k,y_k) \quad ,$$

Schritt 2: Für $y(x_{k+1})$ ergibt sich dann der Näherungswert

(6b) $$y_{k+1} = y_k + \frac{h}{2}\cdot[f(x_k,y_k) + f(x_{k+1},y^*_{k+1})]$$

Beispiel (s.o.): $y' = -xy$, $y(0) = 1$

Berechnung einer Näherung für y(1) mit dem Heun–Verfahren, Schrittweite $h = 0.2$:

n	x_n	y^*_n	y_n
0	0		1
1	0.2	1.000000	0.980000
2	0.4	0.940800	0.922768
3	0.6	0.848947	0.834920
4	0.8	0.734730	0.726047
5	1.0	0.609879	0.606975

Berechnung mit $h = 0.1$ liefert $y(1) \approx 0.606718$, beide Ergebnisse unterscheiden sich erst in der 4. Dezimalstelle. Vergleich mit dem exakten Wert: $y(1) = 0.606531$.

Das Verfahren ist wesentlich genauer als das Euler–Cauchy–Verfahren.

9.3.3. Das Runge–Kutta–Verfahren

Dieses Verfahren ist sehr genau und wird häufig benutzt. Die exakte Herleitung übersteigt den Rahmen dieser Einführung. Um aber die Qualität des Verfahrens abschätzen zu können, wird im folgenden die Grundidee beschrieben.

Wie in 9.3.2 wird integriert; das Integral der rechten Seite über $f(x,y(x))$ wird mit der Keplerschen Faßregel ausgewertet. Dazu benötigt man ein Doppelintervall $[x_k\ ,\ x_k + \frac{h}{2}\ ,\ x_{k+1} = x_k + h]$. Man erhält

$$\int_{x_k}^{x_{k+1}} y'(x)\,dx \;=\; y_{k+1} - y_k \;\approx\; \frac{h}{6}\cdot[f(x_k,y_k) + 4f(x_k+\tfrac{h}{2}\ ,\ y(x_k+\tfrac{h}{2})) + f(x_k+h\ ,\ y(x_k+h))]$$

Das Problem ist nun, durch Anlaufrechnungen Näherungswerte $y_{k+1/2}$, y_{k+1} für $y(x_k+\frac{h}{2})$ bzw. $y(x_k+h)$ zu erhalten. Man verwendet dazu für $y_{k+1/2}$ das Ergebnis des Heun–Verfahrens an der Stelle $(x_k|y_k)$. Dann legt man durch die Punkte $P_1(x_k|y_k)$ und $P_2(x_k+\frac{h}{2}\ |\ y_{k+1/2})$ ein Polynom 3. Grades $p_3(x)$, dessen Tangenten in P_1 und P_2 die durch die DGL vorgegebenen Steigungen besitzen. Als Wert y_{k+1} wird $p_3(x_{k+1})$ bestimmt. Die Keplersche Faßregel liefert dann ein Verfahren, das leicht abgeändert zum Runge–Kutta–Verfahren führt.

Eine ausführliche Herleitung findet man in [4].

Runge–Kutta–Verfahren: Man berechne für $j = 0, 1, 2, \ldots$

$$\left.\begin{aligned} k_1 &= h\cdot f(x_j,y_j) \\ k_2 &= h\cdot f(x_j+\tfrac{h}{2}\ ,\ y_j+\tfrac{1}{2}\cdot k_1) \\ k_3 &= h\cdot f(x_j+\tfrac{h}{2}\ ,\ y_j+\tfrac{1}{2}\cdot k_2) \\ k_4 &= h\cdot f(x_j+h\ ,\ y_j+k_3) \\ y_{j+1} &= y_j + \tfrac{1}{6}\cdot(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned}\right\} \quad (7)$$

Die Hilfsgrößen k_1, k_2 entsprechen dem Euler – bzw. Heun – Verfahren, die Formel für y_{j+1} der Keplerschen Faßregel.

Beispiel (s.o.): $y' = -xy$, $y(0) = 1$. Gesucht sei wieder $y(1)$.

Berechnung mit Schrittweite $h = 0.2$:

x_0 = 0.000 y_0 = 1.000000

k1 = 0.000000
k2 = − 0.020000
k3 = − 0.019800
k4 = − 0.039208

x_1 = 0.200 y_1 = 0.980199

k1 = − 0.039208
k2 = − 0.057636
k3 = − 0.057083
k4 = − 0.073849

x_2 = 0.400 y_2 = 0.923116

k1 = − 0.073849
k2 = − 0.088619
k3 = − 0.087881
k4 = − 0.100228

x_3 = 0.600 y_3 = 0.835270

k1 = − 0.100232
k2 = − 0.109922
k3 = − 0.109243
k4 = − 0.116164

x_4 = 0.800 y_4 = 0.726149

k1 = − 0.116184
k2 = − 0.120250
k3 = − 0.119884
k4 = − 0.121253

x_5 = 1.000 y_5 = 0.606531

Berechnung mit Schrittweite $h = 0.1$ liefert auf 6 Dezimalstellen dasselbe Ergebnis. Vergleich mit dem exakten Wert $y(1)$ bestätigt alle 6 Dezimalstellen.

9.4. Bemerkungen zur Schrittweite h

Bei Anwendung der Verfahren muß die Schrittweite vorgegeben werden. Je kleiner h ist, umso kleiner wird der Fehler bei jedem Einzelschritt, aber umso mehr Einzelschritte sind durchzuführen. Bei zu vielen Einzelschritten wachsen (neben der Rechenzeit) auch die Rundungsfehler wieder an. Eine optimale Schrittweite h ist klein genug, um die Fehler bei jedem Einzelschritt hinreichend klein zu halten, andererseits aber auch nicht zu klein. Die Wahl einer optimalen Schrittweite ist problematisch; es gibt verschiedene Vorschläge zur Schrittweitensteuerung (vgl. Literatur).

Ein einfaches Verfahren ist folgendes: Man beginne mit einer nicht zu kleinen Schrittweite (z.B. 0.2) , wiederhole die Berechnung mit kleineren Schrittweiten (etwa durch Halbieren von h) und vergleiche die Ergebnisse.

9.5. Das Runge–Kutta–Verfahren für Anfangswertprobleme bei Systemen und Differentialgleichungen höherer Ordnung

Im folgenden werden die Formeln für Systeme von 2 gekoppelten Funktionen y(x) und z(x) sowie die entsprechenden Formeln für eine DGL. 2. Ordnung angegeben. Die Formeln für mehr Funktionen (bzw. DGLen höherer Ordnung) lauten analog.

AWP für ein DGL–System 1. Ordnung zur Bestimmung von y(x) und z(x) :

$$y'(x) = f_1(x,y(x),z(x)) \qquad y(x_0) = y_0$$
$$z'(x) = f_2(x,y(x),z(x)) \qquad z(x_0) = z_0$$

Vorgegeben sei eine Schrittweite h.

Runge–Kutta–Verfahren: Man berechne für $j = 0, 1, 2, \ldots$ zeilenweise

$$k_1 = h \cdot f_1(x_j,y_j,z_j) \qquad l_1 = h \cdot f_2(x_j,y_j,z_j)$$
$$k_2 = h \cdot f_1(x_j+\tfrac{h}{2}, y_j+\tfrac{k_1}{2}, z_j+\tfrac{l_1}{2}) \qquad l_2 = h \cdot f_2(x_j+\tfrac{h}{2}, y_j+\tfrac{k_1}{2}, z_j+\tfrac{l_1}{2})$$
$$k_3 = h \cdot f_1(x_j+\tfrac{h}{2}, y_j+\tfrac{k_2}{2}, z_j+\tfrac{l_2}{2}) \qquad l_3 = h \cdot f_2(x_j+\tfrac{h}{2}, y_j+\tfrac{k_2}{2}, z_j+\tfrac{l_2}{2})$$
$$k_4 = h \cdot f_1(x_j+h, y_j+k_3, z_j+l_3) \qquad l_4 = h \cdot f_2(x_j+h, y_j+k_3, z_j+l_3)$$

$$y_{j+1} = y_j + \tfrac{1}{6}\cdot(k_1 + 2k_2 + 2k_3 + k_4)$$
$$z_{j+1} = z_j + \tfrac{1}{6}\cdot(l_1 + 2l_2 + 2l_3 + l_4)$$

AWP für eine DGL 2. Ordnung: Man bestimme $y(x)$ aus

$$y'' = f(x,y(x),y'(x)) \quad , \quad y(x_0) = y_0 = c \quad , \quad y'(x_0) = d \; .$$

Setzt man (siehe 9.1)

$$z(x) = y'(x) \quad , \quad z(x_0) = y'(x_0) = d \; , \qquad \text{oder}$$

$$y'(x) = f_1(x,y(x),z(x)) = z(x)$$

$$z'(x) = f_2(x,y(x),z(x)) = y''(x) = f(x,y(x),y'(x)) \; ,$$

so lassen sich die Runge–Kutta–Formeln für Systeme anwenden.

Runge–Kutta–Verfahren für ein AWP einer DGL 2. Ordnung:

$$y'' = f(x,y(x),y'(x)) \quad , \quad y(x_0) = y_0 = c \quad , \quad y'(x_0) = d \; .$$

Man berechne für $j = 0, 1, 2, \ldots$ zeilenweise

$$k_1 = h \cdot z_j \qquad l_1 = h \cdot f(x_j, y_j, z_j)$$

$$k_2 = h \cdot (z_j + \frac{l_1}{2}) \qquad l_2 = h \cdot f(x_j + \frac{h}{2}, y_j + \frac{k_1}{2}, z_j + \frac{l_1}{2})$$

$$k_3 = h \cdot (z_j + \frac{l_2}{2}) \qquad l_3 = h \cdot f(x_j + \frac{h}{2}, y_j + \frac{k_2}{2}, z_j + \frac{l_2}{2})$$

$$k_4 = h \cdot (z_j + l_3) \qquad l_4 = h \cdot f(x_j + h, y_j + k_3, z_j + l_3)$$

$$y_{j+1} = y_j + \frac{1}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4)$$

$$z_{j+1} = z_j + \frac{1}{6} \cdot (l_1 + 2l_2 + 2l_3 + l_4)$$

Beispiel: $y'' = f(x,y,y') = 4e^{-x} - 2y' - 5y$, $y(0) = y'(0) = 0$.

Gesucht ist ein Näherungswert für $y(1)$.

Der Ansatz $z(x) = y'(x)$ liefert das AWP für Systeme:

$$y' = z \qquad y(0) = y_0 = 0$$
$$z' = f(x,y,z) = 4e^{-x} - 5y - 2z \qquad z(0) = y'(0) = 0 \ .$$

Runge–Kutta–Verfahren mit Schrittweite $h = 0.2$

(die k– bzw. l–Werte werden für den ersten Schritt angegeben) :

$x_0 = 0$ $\qquad y_0 = 0$ $\qquad z_0 = 0$

$k_1 = h \cdot z_0 = 0$ $\qquad l_1 = h \cdot f(0,0,0) = 0.2 \cdot 4 = 0.8$

$k_2 = 0.2 \cdot (0 + \frac{1}{2} \cdot 0.8) = 0.08$ $\qquad l_2 = 0.2 \cdot f(0.1\ ,\ 0\ ,\ 0.4) = 0.563870$

$k_3 = 0.056387$ $\qquad l_3 = 0.571096$

$k_4 = 0.114219$ $\qquad l_4 = 0.370159$

$x_1 = 0.2$	$y_1 = 0.064499$	$z_1 = 0.573348$
$x_2 = 0.4$	$y_2 = 0.203167$	$z_2 = 0.758980$
$x_3 = 0.6$	$y_3 = 0.349894$	$z_3 = 0.673726$
$x_4 = 0.8$	$y_4 = 0.462521$	$z_4 = 0.436366$
$x_5 = 1.0$	$y_5 = 0.521157$	$z_5 = 0.148360$

z_5 wird nicht mehr benötigt. Der gesuchte Näherungswert für $y(1)$ ist $y_5 = 0.521157$.

Berechnung mit $h = 0.1$ liefert den Wert 0.520984.

Vergleich mit dem exakten Wert (einfach lösbare inhomogene lineare DGL mit konstanten Koeffizienten): $y(1) = 0.520971$.

Literaturverzeichnis:

[1] Engeln–Müllges/Reutter, Numerische Mathematik für Ingenieure
BI Wissenschaftsverlag 1987

[2] Engeln–Müllges/Reutter, Formelsammlung zur Num. Mathematik
BI Wissenschaftsverlag 1985

[3] Böhm/Gose/Kahmann Methoden der Numerischen Mathematik
Vieweg 1985

[4] Stiefel Einführung in die Numerische Mathematik
Teubner 1976

[5] Zurmühl Praktische Mathematik
Springer 1965

[6] Henrici Elemente der Numerischen Mathematik
BI Wissenschaftsverlag 1972

[7] Isaacson/Keller Analyse Numerischer Verfahren
Harri Deutsch 1973

[8] Stoer Einführung in die Num. Mathematik 1
Springer 1972

[9] Bulirsch/Stoer Einführung in die Num. Mathematik 2
Springer 1973

[10] Stummel/Hainer Praktische Mathematik
Teubner 1971

[11] Becker/Dreyer/Haacke/Nabert Num. Mathematik für Ingenieure
Teubner 1977

[12] Selder Einführung in die Numerische Mathematik für Ingenieure
Hanser 1973

[13] Scheid Numerical Analysis
Schaum's Outline Series
McGraw–Hill Book Company, New York

[14] Schwarz Elementare Darstellung der schnellen Fouriertransformation
Computing 18, 107–116 (1977)