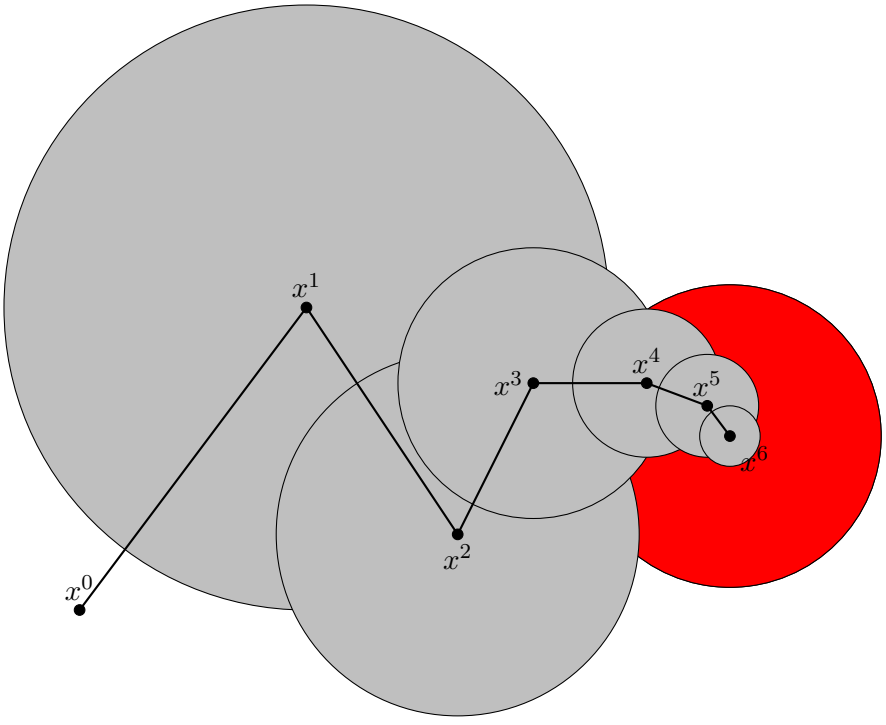
Einführung in die Numerik

Peter Bastian
Interdisziplinäres Zentrum für Wissenschaftliches Rechnen
Universität Heidelberg, Im Neuenheimer Feld 205, 69120 Heidelberg
Peter.Bastian@iwr.uni-heidelberg.de

3. April 2023



Inhaltsverzeichnis

Vorbemerkungen

Der Stoff der Vorlesung "Einführung in die Numerik" ist klassisch und umfasst die Themen

- Zahlendarstellung, Rundungsfehler, Fließkommaarithmetik,
- Direkte Lösung linearer Gleichungssysteme,
- Lösung nichtlinearer Gleichungssysteme,
- Eigenwerte und Eigenvektoren,
- Polynominterpolation,
- Numerische Integration,
- Extrapolationsprinzip,
- Approximation.

Dem entsprechend gibt es ein große Zahl von Standardwerken von denen hier nur vier genannt seien:

- Rannacher, Rolf: Numerik 0: Einführung in die Numerische Mathematik, Heidelberg: Heidelberg University Publishing, 2017. `https://doi.org/10.17885/heiup.206.281`.
- Stoer, Josef: Numerische Mathematik 1, Springer-Verlag.
- H. R. Schwarz, N. Klöckler: Numerische Mathematik, Teubner-Verlag.
- E. Süli, D. Mayers: An introduction to numerical analysis, Cambridge University Press.

Uneinigkeit besteht jedoch hinsichtlich der Reihenfolge der Präsentation des Stoffes. Hier haben sich zwei unterschiedliche Vorgehensweisen herausgebildet: der Beginn mit Rundungsfehlern und Polynominterpolation, klassisch nach Stoer, oder der Beginn mit der Lösung von linearen und dann nichtlinearen Gleichungssystemen wie in Schwarz/Klöckler oder Süli/Mayers.

Ich habe mich für die letztere Vorgehensweise entschieden. Die Gründe dafür sind: Der Beginn mit Zahlendarstellung und Rundungsfehlern ist elementar um Rechnungen am Computer durchzuführen. Allerdings wird die Rundungsfehleranalyse hauptsächlich noch für Algorithmen der linearen Algebra durchgeführt. Schließlich führt die naive Polynominterpolation nach der Schulmethode auf ein lineares Gleichungssystem. Daher scheint mir die Behandlung linearer Gleichungssysteme an zweiter Stelle zu stehen. Daran schließen sich nichtlineare algebraische Systeme natürlich an.

Für ein gutes Verständnis numerischer Methoden ist deren praktische Erprobung an konkreten Beispielen von zentraler Bedeutung. Da die Grundausbildung in der Programmierung in Heidelberg derzeit mit der Programmiersprache C++ erfolgt scheint es mir nur konsequent diese auch in der Numerikvorlesung einzusetzen. Dazu wird die Bibliothek HDNum eingesetzt, welche unter

`https://parcomp-git.iwr.uni-heidelberg.de/Teaching/hdnum.git`

mittels der Versionskontrollsoftware `git` herunter geladen werden kann. HDNum ist "header-only" und sollte mit jedem einigermaßen aktuellen C++-Compiler übersetzt werden können. Die Bibliothek erlaubt als eine Besonderheit Berechnungen mit beliebig hoher Genauigkeit.

Heidelberg, April 2020 *Peter Bastian*

Vorlesung 1

Einleitung

1.1 Grundlegende Aufgaben der Numerik

Die numerische Mathematik beschäftigt sich mit der Berechnung mathematischer Ausdrücke, wie z. B. $x = \sqrt{2}$ oder $I = \int_0^1 e^{z^2} dz$, dem Auflösen von Gleichungen, z.B. $x^2 e^x = 3$ oder dem Lösen von Differentialgleichungen, etwa z. B. $\frac{dy(t)}{dt} = ty(t)$, mit einem Startwert $y(1) = 1$. Charakteristisch ist dabei die Lösung dieser Aufgaben für *für konkrete Zahlenwerte*. Dies ist immer dann sinnvoll, wenn die Lösung einer Gleichung nicht in geschlossener Form darstellbar ist.

Selbst wenn so eine Darstellung möglich ist, enthält sie oft nichtlineare Funktionen, deren Werte nur mit einer Rechenmaschine (näherungsweise) berechenbar sind (oder früher in Tabellenwerken nachgeschlagen werden mussten). Generell sind numerische Berechnungen mit einer Reihe von *Fehlern* behaftet deren Abschätzung und Kontrolle einen zentralen Aspekt der *numerischen Analysis* darstellt.

Als erstes konkretes Beispiel betrachte den Ausdruck $x = \sqrt{2}$. Die Quadratwurzel $\sqrt{2}$ ist ein Symbol. Es definiert x als positive Lösung der quadratischen Gleichung $x^2 = 2$. Dieser Wert ist mit endlich vielen Grundoperationen $+, -, \cdot, /$ sowie Zahlen mit endlich vielen Ziffern nicht berechenbar. Stattdessen kann man nur eine Folge angeben, deren Glieder elementar berechenbar sind und die gegen $\sqrt{2}$ konvergiert. Ein Beispiel ist

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{2}{x_n}\right), \qquad x_0 = 1.$$

Diese Iteration erhält man durch Anwendung des Newton-Verfahrens auf die Gleichung $f(x) = x^2 - 2 = 0$. Die Folgenglieder x_n sind mit elementaren Operationen berechenbar. Die numerische Mathematik untersucht dann unter anderem folgende Fragestellungen:

1) Wie leitet man systematisch Iterationsverfahren zur Lösung von Gleichungen der Form $f(x) = 0$ her?

2) Unter welchen Bedingungen konvergiert so ein Iterationsverfahren?

3) Wie schnell konvergiert so ein Iterationsverfahren

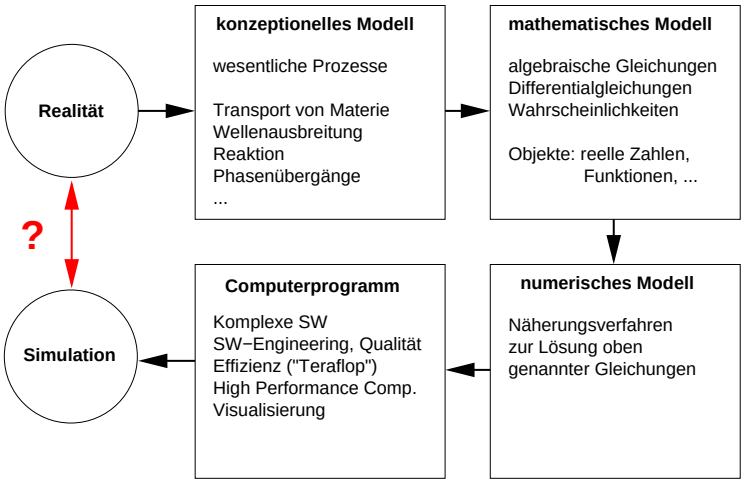
4) Wann kann man die Iteration abbrechen?

5) Welchen Einfluss hat die endliche Zahlendarstellung im Computer?

1.2 Wissenschaftliches Rechnen

Die numerische Mathematik ist in ein größeres Fachgebiet, dem *Wissenschaftlichen Rechnen*, eingebettet. Das Wissenschaftliche Rechnen verbindet Theorie

und Experiment, die klassischen Säulen der wissenschaftlichen Methode. Mithilfe einer Theorie bzw. eines Modells, in Form mathematischer Gleichungen, will man dabei eine Beobachtung, gegeben durch Messungen, erklären. Oft sind Modelle z.B. in der Form von Differentialgleichungen gegeben, welche nur sehr eingeschränkt in geschloßener Form lösbar sind. Als Beispiel dient das Phänomen des Klimawandels. Hier möchte man den Einfluss der Konzentration von Treibhausgasen auf die Temperatur der Atmosphäre beurteilen. In den vergangen Jahrzehnten hat das Wissenschaftliche Rechnen immer mehr an Einfluss gewonnen, da es undurchführbare Experimente ermöglicht (z.B. Galaxiensimulation, Klimawandel), teuere Experimente einspart (z.B. Windkanalexperimente) oder die Abschätzung von Unsicherheiten ermöglicht (z.B. atomares Endlager). Generell geht man im Wissenschaftlichen Rechnen wie folgt vor:



Im *konzeptionellen Modell* klärt man welche Prozesse zur Modellierung der Beobachtung relevant sind und welche Annahmen man trifft. Diese Entscheidung ist in der Regel nicht eindeutig und in Folge ergeben sich Modelle unterschiedlicher Aussagekraft und Genauigkeit. Ausgehend vom konzeptionellen Modell erstellt man ein *mathematisches Modell* in der Form von Gleichungen. Sehr oft spielen auch Unsicherheiten eine Rolle (wie etwa beim Klimawandel) und man zieht Methoden der Wahrscheinlichkeitsrechnung heran (dies werden wir aber in dieser Vorlesung nicht behandeln). Liegt das Modell fest tritt die eigentliche Numerik auf den Plan um mit Hilfe von Näherungsverfahren die mathematischen Gleichungen zu Lösen, man spricht vom *numerischen Modell*. Schließlich ist noch das Fachgebiet *Informatik* wesentlich beteiligt, da die Berechnungen teils einen sehr hohen Aufwand erfordern, der nur von parallelen Supercomputern bereit gestellt wird. Viele Probleme und Verfahren erfordern auch einen hohen Softwareentwicklungsaufwand. Schließlich liefert die Modellbildung und Simulation Zahlenwerte welche mit den Messungen verglichen werden können. In der Regel stimmen die berechneten Werte nicht exakt mit den Messungen überein und dann stellen sich mindestens folgende Fragen:

1) Welche Unterschiede zwischen Messung und Simulation sind akzeptabel?

2) Was sind die Ursachen für diese Unterschiede?

3) Wie können diese Unterschiede reduziert werden?

Zum zweiten Punkt sollen folgende Fehlerquellen genannt werden:

a) *Modellfehler*: Ein relevanter Prozess wurde nicht oder ungenau modelliert (z.B. Temperatur konstant, Luftwiderstand vernachlässigt, ...).

b) *Datenfehler*: Messungen von Anfangsbedingungen, Randbedingungen, Werte für Parameter sind fehlerbehaftet.

c) *Abschneidefehler*: Abbruch von Reihen oder Iterationsverfahren, Approximation von Funktionen (z.B. stückweise Polynome), diskreter statt kontinuierlicher Zeitverlauf.

d) *Rundungsfehler*: Reelle Zahlen werden im Rechner genähert dargestellt.

Mit den beiden letztgenannten Fehlerquellen werden wir uns in der Numerik ausführlich beschäftigen.

1.3 Modellierung der Ausbreitung des Coronavirus

Als Beispiel für die Untersuchung eines Prozesses mit numerischen Methoden betrachten wir die Ausbreitung des Coronavirus.

1.3.1 Ein Wachstumsmodell

Die Funktion $N : [t_0, \infty) \to \mathbb{N} \cup \{0\}$ beschreibe die Anzahl der infizierten Personen zur Zeit $t \geq t_0$. Wie könnte so eine Funktion aussehen? Als ersten Abstraktionsschritt beschreiben wir $N(t)$ durch eine Funktion $u(t)$ welche Werte aus den reellen Zahlen $u(t) \in \mathbb{R}$ annehmen darf. Dies erscheint zunächst ungewöhnlich, da ja eine Person entweder infiziert ist oder nicht. Der Vorteil dieser Verallgemeinerung wird sein, dass wir damit eine bestimmte Klasse von mächtigen mathematischen Modellen nutzen können, sogenannte *Differentialgleichungen*. Für hinreichend große Zahlen $N(t)$ wird die Näherung mit nichtganzen Zahlen nur einen kleinen (relativen) Fehler mit sich bringen.

Zur Modellierung der zeitlichen Entwicklung der Zahl der infizierten Personen machen wir die folgende Annahme: *Die Änderung der Zahl der infizierten Personen im Zeitintervall* $[t, t + \Delta t]$ *sei proportional zur Zahl der infizierten Personen zur Zeit* t *und zur Länge des Zeitintervalls* Δt. In Formeln:

$$u(t + \Delta t) = u(t) + \lambda \Delta t u(t). \tag{1.1}$$

Diese Annahme ist sinnvoll solange unbegrenzte Resourcen für das Wachstum zur Verfügung stehen, z.B. kann jede infizierte Person stets neue, nicht infizierte Personen treffen. Die Konstante $\lambda \in \mathbb{R}$ beschreibt die Wachstumsrate. Umstellen von Gleichung (1.1) liefert

$$\frac{u(t + \Delta t) - u(t)}{\Delta t} = \lambda u(t)$$

und schließlich liefert der Grenzübergang $\Delta t \to 0$ die lineare, gewöhnliche Differentialgleichung erster Ordnung

$$\frac{du}{dt}(t) = \lambda u(t). \tag{1.2}$$

Man kann zeigen, dass alle Lösungen dieser Gleichung die Form

$$u(t) = ce^{\lambda t}$$

mit einer Konstanten $c \in \mathbb{R}$ besitzen. Die Konstante c wird in der Regel durch einen Anfangswert $u(t_0) = u_0$ festgelegt. Dann spricht man von einer *Anfangswertaufgabe* deren Eigenschaften der folgende Satz zusammenfasst.

Satz 1.1. Die Anfangswertaufgabe

$$\frac{du}{dt}(t) = \lambda u(t), \qquad u(t_0) = u_0, \tag{1.3}$$

besitzt die Lösung

$$u(t) = u_0 e^{\lambda(t-t_0)}. \tag{1.4}$$

Beweis. Durch einsetzen und $u(t_0) = u_0 e^{\lambda 0} = u_0$. □

Das Modell (1.3) ist ein sehr einfaches Modell zur Beschreibung von Wachstumsprozessen welches viele Effekte, z.B. räumliche Abhängigkeiten, vernachlässigt. Trotzdem beschreibt es z.B. erstaunlich gut das Wachstum der Weltbevölkerung im Zeitraum 1700-1961, siehe Braun [1994]. Wir wollen nun dieses Modell zur Beschreibung der Ausbreitung des Coronavirus einsetzen und z.B. der Frage nachgehen wann und ob die zur Eindämmung der Ausbreitung getroffenen Maßnahmen greifen. Auf dem Weg werden wir einige grundlegende Aufgaben und Methoden der Numerik kennenlernen, welche im Rest der Vorlesung ausführlich behandelt werden.

1.3.2 Die Datenlage

Abbildung 1.1 zeigt die Daten der täglichen Entwicklung der mit dem Coronavirus infizierten Personen in Deutschland ab dem 24. Februar 2020, also

$$N_i = N(t_i), \qquad i = 0, \dots, m-1,$$

wobei die Zeiteinheit als "Tag" gewählt wurde, d.h. $t_i = i$. Die obere Kurve "sieht nach einem exponentiellen Verlauf" aus, aber ist das tatsächlich der Fall?

Würde der Verlauf dem Gesetz (1.4) folgen, so erhält man für die Logarithmen bei $t_i = i$:

$$\ln u_i = \ln u(t_i) = lnu_0 + \lambda i,$$

d.h. der (natürliche) Logarithmus der Fallzahlen sollte sich wie eine Gerade verhalten. In der unteren Grafik in Abbildung 1.1 sieht man dass dies nur eingeschränkt der Fall ist.

Um die Daten mit der Bibliothek HDNum zu analysieren müssen wir diese zunächst eingeben. Das gelingt mit den folgenden Anweisungen:

```
using Number = double;
using Vec = Vector<Number>;

// Corona Faelle Deutschland ab dem 24.2.2020
// https://de.wikipedia.org/wiki/COVID-19-Pandemie_in_Deutschland
Vec N = {16, 18, 21, 26, 53, 66, 117, 150, 188, 240,
  400, 639, 795, 902, 1139, 1296, 1567, 2369,
  3062, 3795, 4838, 6012, 7156, 8198, 10999, 13957,
  16662, 18610, 22672, 27436, 31554, 36508, 42288,
  48582,52547,57298,61913,67366,73522,79696,85778,91714,95391,
  99225,103228};
Vec t(N.size());
fill(t,0.0,1.0); // Zeitpunkt in Tagen
vector<string> tstring = {"24/02/2020","25/02/2020","26/02/2020",
```

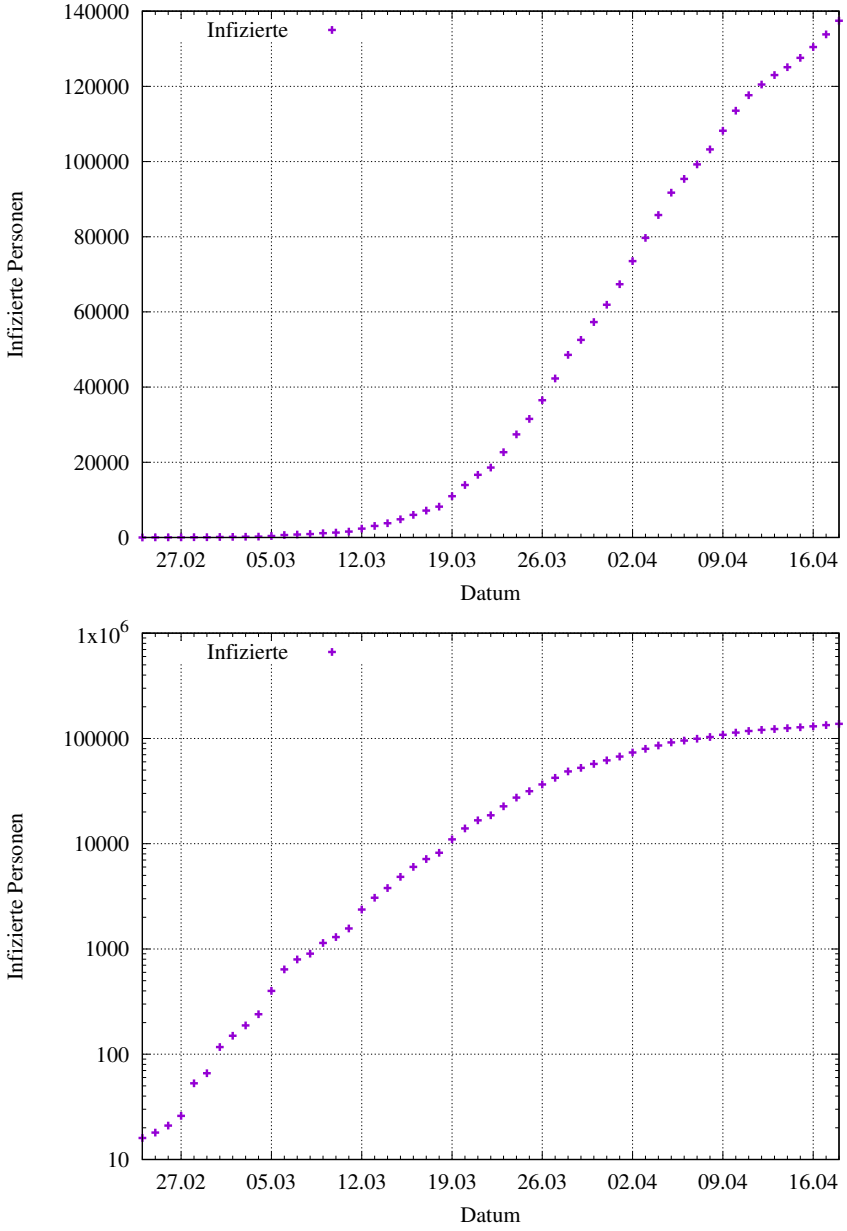


Abbildung 1.1: Tägliche Zahl der mit dem Coronavirus infizierten Personen ab dem 24. Februar 2020. Quelle: Wikipedia `https://de.wikipedia.org/wiki/COVID-19-Pandemie_in_Deutschland`. Oben in absoluter Darstellung, unten in logarithmischer Darstellung.

```
  "27/02/2020","28/02/2020","29/02/2020","01/03/2020","02/03/2020",
  "03/03/2020","04/03/2020","05/03/2020","06/03/2020","07/03/2020",
  "08/03/2020","09/03/2020","10/03/2020","11/03/2020","12/03/2020",
  "13/03/2020","14/03/2020","15/03/2020","16/03/2020","17/03/2020",
  "18/03/2020","19/03/2020","20/03/2020","21/03/2020","22/03/2020",
  "23/03/2020","24/03/2020","25/03/2020","26/03/2020","27/03/2020",
  "28/03/2020","29/03/2020","30/03/2020","31/03/2020","01/04/2020",
  "02/04/2020","03/04/2020","04/04/2020","05/04/2020","06/04/2020",
  "07/04/2020","08/04/2020"
}; // Datum zur Ausgabe
gnuplot("N.dat",tstring,N); // Ausgabe der Daten
```

1.3.3 Vorhersage und Parameterschätzung

Ein Anwendungsszenario von Modellbildung und Simulation ist die Vorhersage. So möchte man etwa die Zahl der mit dem Coronavirus infizierten Personen vorhersagen um sich auf Maßnahmen vorbereiten zu können. Auch beim Klimawandel spielt die Vorhersage verschiedener Szenarien ein große Rolle.

Voraussetzung für eine Vorhersage ist die Kenntnis von *Parametern* des Modells. In unserem Modell (1.4) sind die Parameter der Anfangswert u_0 und die Wachstumsrate λ. Hingegen ist die Startzeit t_0 in der Regel bekannt. Die übliche Vorgehensweise bei der Vorhersage besteht daher aus zwei Schritten:

1) *Parameterschätzung:* Bestimmung der Parameter des Modells aus der Literatur oder gemessenen Daten.

2) *Auswertung:* Berechnung des Modells mit den bestimmten Parametern.

In unserem Fall liegt das Modell in geschlossener (analytischer) Form vor. Die Auswertung erfordert daher nur die Berechnung der Formel (1.4). Hier werden wir später die Auswirkung von *Rundungsfehlern* näher betrachten. Wenden wir uns daher vor allem dem Parameterschätzproblem zu.

Interpolation Ein einfacher Fall liegt vor wenn genau zwei Datenpunkte (t_0, N_0), (t_i, N_i) zur Verfügung stehen. Dann können die Parameters u_0 und λ einfach berechnet werden:

Erweitern Sie die Herleitung auf den Fall $u(t_i) = N_i$, $u(t_j) = N_j$, $j \neq i$.

$$\left.\begin{array}{rcl} u_0 e^{\lambda 0} & = & N_0 \\ u_0 e^{\lambda(t_i - t_0)} & = & N_i \end{array}\right\} \quad \Rightarrow \quad u_0 = N_0, \lambda = \frac{\ln N_i - \ln N_0}{t_i - t_0}. \tag{1.5}$$

Offensichtlich muss $N_0 \neq 0$, $N_i \neq 0$ und $t_i \neq t_0$ gelten. Da die so bestimmte Funktion $u(t)$ die Datenpunkte (t_0, N_0), (t_i, N_i) exakt reproduziert spricht man von *Interpolation*. Folgende Funktion realisiert die Interpolation in HDNum:

```
Vec interpolation (const Vec& N, const Vec& t, int i)
{
  Vec p(2); // Ergebnis hat zwei Komponenten
  p[0] = N[0]; // u_0
  p[1] = (log(N[i])-log(N[0]))/(t[i]-t[0]); // λ
  return p;
}
```

Ausgleichsgerade Ein anderer einfacher Fall liegt vor, wenn wir versuchen die Logarithmen zu interpolieren. Dies führt auf das Gleichungssystem

$$\ln u_0 + \lambda(t_i - t_0) = \ln N_i \qquad i = 0, \dots, m-1.$$

Betrachten wir statt u_0 den Logarithmus $\ln u_0$ als Unbekannte, stellt dies ein lineares Gleichungssystem dar:

$$\begin{bmatrix} 1 & 0 \\ 1 & t_1 - t_0 \\ \vdots & \vdots \\ 1 & t_{m-1} - t_0 \end{bmatrix} \begin{bmatrix} \ln u_0 \\ \lambda \end{bmatrix} = \begin{bmatrix} \ln N_0 \\ \ln N_1 \\ \vdots \\ \ln N_{m-1} \end{bmatrix} \Leftrightarrow \; Ax = b. \tag{1.6}$$

Für $m > 2$ ist dieses Gleichungssystem allerdings überbestimmt und wir können in der Regel nicht erwarten, dass es eine Lösung besitzt. Statt dessen kann man versuchen die Lösung "so gut wie möglich" zu machen, indem man die Größe

$$J_{\text{lin}}(x) = \|Ax - b\|^2 = \sum_{i=0}^{m-1} (\ln u_0 + (t_i - t_0)\lambda - \ln N_i)^2 \tag{1.7}$$

zu minimieren. Dieses Verfahren wird auch *Gauß'sche Ausgleichsrechnung* genannt. Die resultierende Gerade $\ln u_0 + t\lambda$ bezeichnet man als *Ausgleichsgerade*. Sie approximiert die Logarithmen der Fallzahlen durch eine Gerade. Wir werden später zeigen, dass man die Lösung des Minimierungsproblems durch *Lösung* des folgenden Gleichungssystemes erhält:

$$A^T A x = A^T b. \tag{1.8}$$

Wiederholen Sie die Lösbarkeit linearer Gleichungssystem aus der Vorlesung Lineare Algebra.

In HDNum lässt sich das wie folgt realisieren:

```
Vec linear_fit (const Vec& N, const Vec& t)
{
  // Erstelle die Matrix A
  Mat A(N.size(),2);
  for (int i=0; i<N.size(); ++i) {
    A[i][0] = 1.0; A[i][1] = t[i]-t[0];
  }

  // Erstelle die rechte Seite
  Vec logN(N.size());
  for (int i=0; i<N.size(); ++i) logN[i] = log(N[i]);

  // berechne Ausgleichsgerade
  Mat AT(A.transpose()); // transponieren
  Mat ATA(2,2); ATA.mm(AT,A); // A^T*A
  Vec b(2);
  AT.mv(b,logN); // b = A^T * logN
  Vec p(2);
  linsolve(ATA,p,b);
  return p;
}
```

Nichtlineares Parameterschätzproblem Statt den Logarithmen kann man auch versuchen die Datenpunkte direkt zu interpolieren. D.h. wir setzen an:

$$u(t_i) = u_0 e^{\lambda(t_i - t_0)} = N_i \qquad i = 0, \ldots, m-1.$$

Für $m > 2$ führt dies auf ein überbestimmtes *nichtlineares* Gleichungssystem. Analog zum linearen Fall oben, kann man hier die Größe

$$J_{\mathrm{nl}}(u_0, \lambda) = \sum_{i=0}^{m-1} (u_0 e^{\lambda(t_i - t_0)} - N_i)^2 \tag{1.9}$$

minimieren und so die Parameter u_0 und λ aus m Datenpunkten bestimmen. Eine notwendige Bedingung für ein Minimum ist das Verschwinden der ersten partiellen Ableitungen von J_{nl} nach den Parametern:

$$F(u_0, \lambda) = \begin{bmatrix} \frac{\partial J_{\mathrm{nl}}}{\partial u_0}(u_0, \lambda) \\ \frac{\partial J_{\mathrm{nl}}}{\partial \lambda}(u_0, \lambda) \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{m-1} \left[(u_0 e^{\lambda(t_i - t_0)} - N_i) e^{\lambda(t_i - t_0)}\right] \\ \sum_{i=0}^{m-1} \left[(u_0 e^{\lambda(t_i - t_0)} - N_i)(t_i - t_0) e^{\lambda(t_i - t_0)}\right] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Das Gleichungssystem

$$F(x) = 0, \qquad x = (u_0, \lambda)^T,$$

kann mit dem Newton-Verfahren iterativ gelöst werden. Dieses lautet

$$x^{k+1} = x^k - \eta_k H^{-1}(x^k) F(x^k) \tag{1.10}$$

mit der Matrix

$$H(u_0, \lambda) = \begin{bmatrix} \frac{\partial F_0}{\partial u_0}(u_0, \lambda) & \frac{\partial F_0}{\partial \lambda}(u_0, \lambda) \\ \frac{\partial F_1}{\partial u_0}(u_0, \lambda) & \frac{\partial F_1}{\partial \lambda}(u_0, \lambda) \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 J_{nl}}{\partial u_0^2}(u_0, \lambda) & \frac{\partial \partial^2 J_{nl}}{\partial u_0 \partial \lambda}(u_0, \lambda) \\ \frac{\partial \partial^2 J_{nl}}{\partial \lambda \partial u_0}(u_0, \lambda) & \frac{\partial \partial^2 J_{nl}}{\partial \lambda^2}(u_0, \lambda) \end{bmatrix}.$$

H heisst *Jacobimatrix von F* bzw. *Hessematrix* von J_{nl}. Man benötigt also im Newtonverfahren insgesamt zweite Ableitungen von J_{nl}. Die Bereitstellung von Ableitungen kann bei komplexeren Modellen aufwändig werden. Daher werden in der Praxis auch *numerisches und automatisches Differenzieren* eingesetzt. Die Größe η_k in der Iteration (1.10) ist ein geeignet gewählter *Dämpfungsfaktor*, der für die Konvergenz des Verfahrens wichtig sein kann. Ebenfalls wichtig für die Konvergenz des Verfahrens ist ein geeignet gewählter *Startwert* x^0.

In HDNum lässt sich das Newtonverfahren wie folgt realisieren:

```
Vec nonlinear_fit (const Vec& N, const Vec& t, const Vec& p0)
{
  auto F = [&] (const Vec& p) // eine Lambdafunktion
  {
    Vec r(2,0.0);
    for (int i=0; i<N.size(); ++i) {
      r[0] += (p[0]*exp(p[1]*(t[i]-t[0]))-N[i])
                *exp(p[1]*(t[i]-t[0]));
      r[1] += (p[0]*exp(p[1]*(t[i]-t[0]))-N[i])*(t[i]-t[0])
                *exp(p[1]*(t[i]-t[0]));
    }
    return r;
  };
  auto nlp = getNonlinearProblem(F,p0);
```

```
  Newton newton;                // Ein Newtonobjekt
  newton.set_maxit(50);     // Setze diverse Parameter
  newton.set_verbosity(2);
  newton.set_reduction(1e-10);
  newton.set_abslimit(1e-100);
  newton.set_linesearchsteps(8);

  Vec p(p0);              // die Loesung
  newton.solve(nlp,p);    // Berechne Loesung
  return p;
}
```

Die Jacobimatrix von F wird in dieser Implementierung numerisch berechnet.

Abbildung 1.2 zeigt das Ergebnis der Vorhersage für die drei verschiedenen Parameterschätzmethoden. Dabei wurden die Fallzahlen für die letzten fünf Tage aus allen vorherigen Daten mit den drei verschiedenen Methoden vorher gesagt. Die Ausgleichsgerade überschätzt die Fallzahlen deutlich. Die Interpolation der Daten vom 24. Februar und sechstletzten Tag liegt schon besser und am besten liegt man mit der Lösung des nichtlinearen Parameterschätzproblems. Deutlich kann man auch die gute Annäherung der Datenpunkte vor dem Vorhersagezeitraum mit dieser Methode erkennen. Allerdings überschätzt auch die beste Methode die realen Daten deutlich. Dies liegt daran, dass die am 22. März getroffenen Eindämmungsmaßnahmen Wirkung zeigen. Das Modell mit einer konstanten Wachstumsrate λ ist kein gutes Modell für diesen Fall. Eine Verbesserung stellt eine zeitlich variable Wachstumsrate $\lambda(t)$ dar.

Können Sie dieses Ergebnis erklären?

1.3.4 Analyse der Verdopplungszeiten

Ziel der am 22. März veranlassten Maßnahmen war es die Ausbreitungsgeschwindigkeit des Virus zu verlangsamen. Als Maß für die Ausbreitungsgeschwindigkeit ziehen wir die *Verdopplungszeit*, d.h. die Zeitspanne innerhalb der sich die Anzahl der Infizierten Personen verdoppelt, heran. Aus unserem Modell erhalten wir die Verdopplungszeit mittels

$$u_0 e^{\lambda T_2} = 2u_0 \quad \Rightarrow \quad T_2 = \frac{ln2}{\lambda}. \tag{1.11}$$

Zur Analyse der Wirkung der Eindämmungsmaßnahmen postulieren wir dass λ (und damit die Verdopplungszeit) mit der Zeit variiert. Über einen Zeitraum weniger Tage können wir jedoch annehmen, dass λ annähernd konstant ist. Die Idee ist nun, zum Zeitpunkt t_k die Wachstumsrate $\lambda(t_k)$ aus den w Datenpunkten $(t_{k-w-1}, N_{k-w-1}), \ldots, (t_k, N_k)$ mit Hilfe der oben beschriebenen nichtlinearen Parameterschätzung zu bestimmen. w bezeichnet man dabei als *Fenstergröße*.

Eine naive Möglichkeit die Verdopplungszeit zu einem gegebenen N_k zu bestimmen ist natürlich

$$T_2 = k - \max\{i \,:\, N_i \leq N_k/2\}.$$

Dann erhält man jedoch nur ganze Tage.

Abbildung 1.3 zeigt die mit der Parameterschätzung berechneten Verdopplungszeiten für zwei Fenstergrößen sowie die naive Variante. Innerhalb einer Woche nach Einleitung der Maßnahmen ist die Verdopplungszeit von etwa 4 auf etwa 7-8 angestiegen. Dabei beobachten wir: größere Fenstergröße führt zu

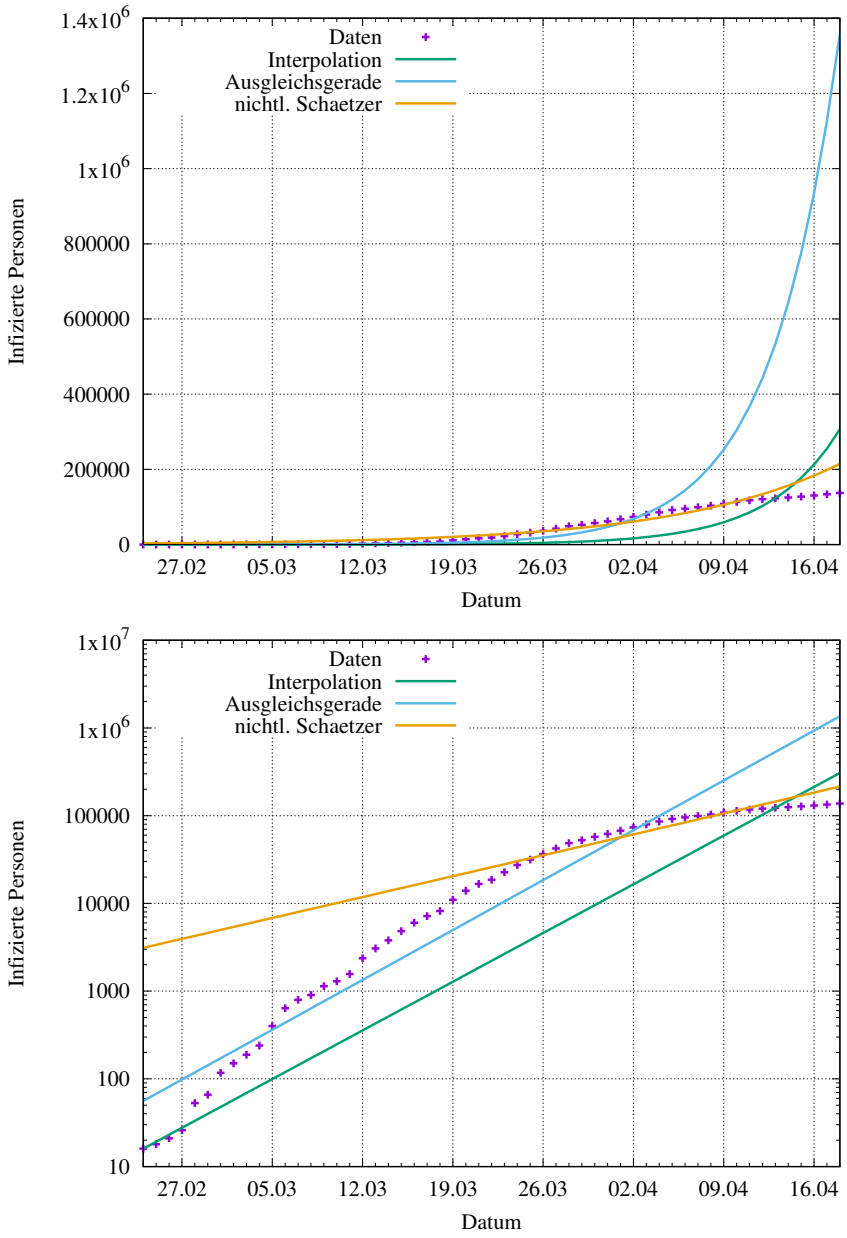


Abbildung 1.2: Vorhersage durch die drei verschiedene Parameterschätzmethoden in normaler und logarithmischer Darstellung. Die Daten bis zum sechstletzten Tag werden benutzt um die Fallzahlen für die letzten fünf Tage vorherzusagen.

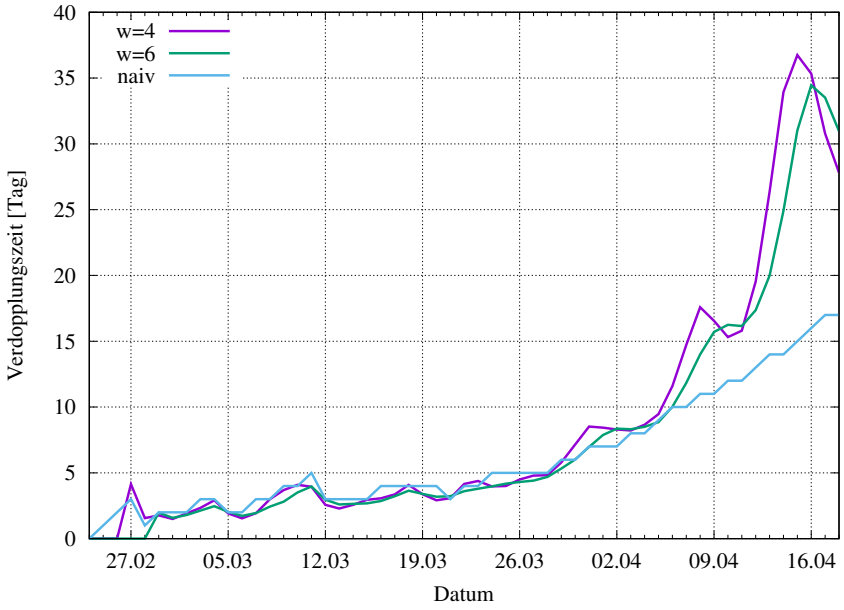


Abbildung 1.3: Verdopplungszeiten für verschiedene Fenstergrößen w.

einem glatteren Verlauf, allerdings auch zu absolut niedrigeren Zahlen. Die naive Methode ist nur bei annähernd exponentiell wachsendem Verlauf eine gute Näherung. Schön zu erkennen ist auch der Einbruch der Verdopplungszeiten nach dem Osterwochenende.

1.3.5 Ausblick

Das hier behandelte Wachstumsmodell ist sehr einfach und diente vor allem dazu verschiedene numerische Methoden, welche in dieser Vorlesung behandelt werden, zu illustrieren. Die in den Medien oft genannte *Reproduktionszahl* ist ein Parameter des sogenannten SIR-Modells, siehe Kermack et al. [1927]. Dieses Modell betrachtet drei Funktionen $S(t)$ (suszeptible) die ansteckbaren Individuen, $I(t)$ (infectious) die angesteckten Individuen, sowie $R(t)$ (removed) die genesenen bzw. gestorbenen Individuen und besteht aus einer nichtlinearen gekoppelten Differentialgleichung für diese Funktionen.

1.4 Zusammenfassung

Numerik ist ein wesentliches Teilgebiet des Wissenschaftlichen Rechnens. Mittels mathematischer Modelle und deren numerischer Simulation lassen sich lassen sich Hypothesen testen und Beobachtungen erklären. Am Beispiel der Ausbreitung des Coronavirus wurden Parameterschätzprobleme eingeführt, welche auf lineare bzw. nichtlineare Gleichungssysteme führen. Gauß'sche Ausgleichsrechnung und Newtonverfahren wurden als Lösungsverfahren kurz vorgestellt.

Vorlesung 2

Fließkommadarstellung

2.1 Einführendes Beispiel

Alle Programmiersprachen stellen elementare Datentypen zur Repräsentation von Zahlen zur Verfügung. Diese lauten mit ihrer mathematischen Entsprechung wie folgt:

Datentyp	Mathematik
`unsigned int, unsigned short, unsigned long`	$\mathbb{N}_0$
`int, short, long`	$\mathbb{Z}$
`float, double`	$\mathbb{R}$
`complex<float>, complex<double>`	$\mathbb{C}$

Natürlich sind diese Datentypen Idealisierungen der Zahlenmengen $\mathbb{N}_0, \mathbb{Z}, \mathbb{R}, \mathbb{C}$, da diese jeweils unendliche viele Elemente enthalten.

Bei `unsigned int`, `int` ... besteht die Idealisierung darin, dass es eine größte (bzw. kleinste) darstellbare Zahl gibt. Solange man diesen Bereich nicht überschreitet sind die Ergebnisse exakt.

Bei `float` und `double` kommt hinzu, dass die meisten innerhalb des darstellbaren Bereichs liegenden Zahlen nur *näherungsweise* dargestellt werden können. Wir betrachten in einem Beispiel welche Auswirkungen diese Näherung auf das Ergebnis einer Berechnung haben kann.

Beispiel 2.1. Die Funktion e^x lässt sich mit einer Potenzreihe berechnen

$$e^x = 1 + \sum_{i=1}^{\infty} \frac{x^i}{i!}.$$

Damit ist die Partialsumme $S_n = 1 + \sum_{i=1}^{n} \frac{x^i}{i!}$ eine Näherung. Praktisch erfolgt die Berechnung mit der Rekursionformel

$$y_n = \frac{x}{n} y_{n-1}, \qquad S_n = S_{n-1} + y_n$$

und dem Rekursionsanfang

$$y_1 = x, \qquad S_1 = 1 + y_1.$$

Wir untersuchen nun diese Rekursion für verschiedene Genauigkeiten und Werte von x.

Für den Wert $x = 1$ und `float`-Genauigkeit erhalten wir:

```
1.000000000000000e+00   1  2.000000000000000e+00
5.000000000000000e-01   2  2.500000000000000e+00
1.666666716337204e-01   3  2.666666746139526e+00
4.166666790843010e-02   4  2.708333492279053e+00
8.333333767950535e-03   5  2.716666936874390e+00
1.388888922519982e-03   6  2.718055725097656e+00
```

```
 1.984127011382952e-04    7  2.718254089355469e+00
 2.480158764228690e-05    8  2.718278884887695e+00
 2.755731884462875e-06    9  2.718281745910645e+00
 2.755731998149713e-07   10  2.718281984329224e+00
 0.000000000000000e+00  100  2.718281984329224e+00
                         ex  2.718281828459045e+00
```

Dabei stehen in der ersten Spalte die Summanden $y_n = x^n/n!$, in der zweiten Spalte der Index n und in der dritten Spalte die Größe S_n als Näherung von e^x. Zwischen $n = 10$ und $n = 100$ wurden die Werte der Übersichtlichkeit wegen weggelassen. Der Vergleich mit dem exakten Wert ergibt eine Genauigkeit von 7 Ziffern.

Für $x = 5$ erhält man entsprechend

```
 9.333108209830243e-06   21  1.484131774902344e+02
                         ex  1.484131591025766e+02
```

also keine Änderung in der Genauigkeit.

Gehen wir nun zu negativen Argumenten über. Für $x = -1$ und `float`-Genauigkeit erhält man das folgende Ergebnis

```
 2.755731998149713e-07   10  3.678794205188751e-01
-2.505210972003624e-08   11  3.678793907165527e-01
 2.087675810003020e-09   12  3.678793907165527e-01
                         ex  3.678794411714423e-01
```

also nur noch 6 gültige Ziffern.

Für $x = -5$ ergibt sich

```
-5.000000000000000e+00    1 -4.000000000000000e+00
 1.250000000000000e+01    2  8.500000000000000e+00
-2.083333396911621e+01    3 -1.233333396911621e+01
 2.604166793823242e+01    4  1.370833396911621e+01
-2.333729527890682e-02   15  1.118892803788185e-03
 7.292904891073704e-03   16  8.411797694861889e-03
 1.221854423194557e-10   28  6.737461313605309e-03
 0.000000000000000e+00  100  6.737461313605309e-03
                         ex  6.737946999085467e-03
```

also nur noch 4 gültige Ziffern!

Schließlich ergibt sich für $x = -20$ und `float`-Genauigkeit

```
-2.000000000000000e+01    1 -1.900000000000000e+01
 2.000000000000000e+02    2  1.810000000000000e+02
-1.333333374023438e+03    3 -1.152333374023438e+03
 6.666666992187500e+03    4  5.514333496093750e+03
-2.666666796875000e+04    5 -2.115233398437500e+04
-2.611609750000000e+06   31 -1.011914250000000e+06
 1.632256125000000e+06   32  6.203418750000000e+05
-9.892461250000000e+05   33 -3.689042500000000e+05
 5.819095000000000e+05   34  2.130052500000000e+05
-3.325197187500000e+05   35 -1.195144687500000e+05
 1.847331718750000e+05   36  6.521870312500000e+04
-4.473213550681976e-07   65  7.566840052604675e-01
 1.355519287926654e-07   66  7.566841244697571e-01
-4.046326296247571e-08   67  7.566840648651123e-01
 1.190095932912527e-08   68  7.566840648651123e-01
                         ex  2.061153622438557e-09
```

In diesem Fall ist *keine* Stelle im Ergebnis korrekt! Das Ergebnis ist absolut um 8 Größenordnungen falsch!

Nun liegt es nahe die Berechnungen mit einer genaueren Zahlendarstellung zu wiederholen. Für $x = -20$ und `double`-Genauigkeit erhält man die folgenden Werte

```
-1.232613988175268e+07    27  -5.180694836889297e+06
 8.804385629823344e+06    28   3.623690792934047e+06
 1.821561256740375e-24    94   6.147561828914626e-09
-3.834865803663947e-25    95   6.147561828914626e-09
                          ex   2.061153622438557e-09
```

Das Ergebnis hat immer noch keine korrekte Stelle, aber immerhin liegt es in der gleichen Größenordnung.

Erst mit „vierfacher Genauigkeit“ gegenüber dem Datentyp `float` erhält man

```
 118  2.06115362243855833927004587529 47e-09
  ex  2.06115362243855782796594038015 58e-09
```

also 15 gültige Ziffern im Ergebnis, bei ca 30 Ziffern „Rechengenauigkeit“.

Damit sind wir für die folgenden Fragen motiviert:

- Was bedeutet „Rechengenauigkeit“?
- Wann und wie entstehen Fehler in einer Berechnung?
- Können diese Fehler vermieden oder minimiert werden?

Im übrigen können die Fehler in diesem Beispiel einfach vermieden werden: Man nutzt einfach die Beziehung $e^x = 1/e^{-x}$ und führt damit die Berechnung der Exponentialfunktion für negative Argumente $x < 0$ auf die Berechnung mit einem positiven Argument zurück. □

2.2 Fließkommazahlen

Zur Zahlendarstellung hat sich schon vor Jahrtausenden das *Stellenwertsystem* herausgebildet. Eine natürliche Zahl $x \in \mathbb{N}$ wird dabei dargestellt als

$$x = m_n\beta^n + \ldots + m_1\beta + m_0 = \sum_{i=0}^{n} m_i\beta^i$$

mit den *Ziffern* $m_0, \ldots, m_n \in \{0, 1, \ldots, \beta - 1\}$ und der *Basis* $\mathbb{N} \ni \beta \geq 2$. Schon im alten Babylon hat man dieses System mit $\beta = 60$ verwendet. Die Basis 10 hat sich in Europa ab dem 16. Jahrhundert durchgesetzt und Blaise Pascal konnte zeigen, dass jede natürliche Zahl $\beta \geq 2$ als Basis geeignet ist, siehe [Knuth, 1998, p. 194]. Als ein Beispiel für die Zahlendarstellung ohne Stellenwertsystem seien die römischen Zahlen erwähnt. Diese Darstellung ist für arithmetische Operationen wenig geeignet.

Für negative und nichtganze Zahlen erweitert man das Stellenwertsystem einfach um ein Vorzeichen und negative Potenzen und erhält die *Festkommadarstellung*:

$$x = \pm m_n\beta^n + \ldots + m_1\beta + m_0 + m_{-1}\beta^{-1} + \ldots + m_{-k}\beta^{-k} = \pm \sum_{i=-k}^{n} m_i\beta^i.$$

Allerdings kommen in naturwissenschaftlichen Berechnungen sehr verschieden große Zahlen vor. So hat das Planck'sche Wirkungsquantum den Wert $6.6260693 \cdot 10^{-34}$Js und die Avogadrokonstante aus der Chemie hat den Wert $6.021415 \cdot 10^{23}\text{mol}^{-1}$. Die Darstellung dieser Zahlen in einem Festkommasystem würde eine sehr große Stellenzahl erfordern, z.B. mit der im Computer üblichen Basis $\beta = 2$ wären ungefähr 190 Binärstellen erforderlich. Da Zahlen mit solchen Größenunterschieden oft *nicht gleichzeitig* verarbeitet werden müssen führt man die effizientere Fließkommadarstellung ein.

Definition 2.2 (Normierte Fließkommadarstellung)**.** Sei $\beta, r, s \in \mathbb{N}$ und $\beta \geq 2$. $\mathbb{F}(\beta, r, s) \subset \mathbb{R}$ heißt *Menge der normierten Fließkommazahlen.* Ihre Elemente erfüllen die folgenden Eigenschaften

a) Für alle $x \in \mathbb{F}(\beta, r, s)$ gilt

$$x = m(x)\beta^{e(x)}, \quad m(x) = \pm \sum_{i=1}^{r} m_i \beta^{-i}, \quad e(x) = \pm \sum_{j=0}^{s-1} e_j \beta^j.$$

Dabei heißt $m(x)$ Mantisse von x und $e(x)$ Exponent von x.

b) Für alle $x \in \mathbb{F}(\beta, r, s)$ gilt: $x = 0 \vee m_1 \neq 0$. Dies *Normierungsbedingung* führt zu einer eindeutigen Fließkommadarstellung von $x \neq 0$ und erlaubt Abschätzungen über die Größe der Mantisse. □

Leiten Sie diese Abschätzungen her.

Aus dieser Definition ergeben sich folgende Abschätzungen für $x \neq 0$:

$$\frac{1}{\beta} \leq |m(x)| < 1, \qquad \beta^{e(x)-1} \leq |x| \leq \beta^{e(x)}. \tag{2.1}$$

Beispiel 2.3. a) $\mathbb{F}(10, 3, 1)$ besteht aus Zahlen der Form

$$x = \pm(m_1 \cdot 0.1 + m_2 \cdot 0.01 + m_3 \cdot 0.001) \cdot 10^{\pm e_0}$$

mit $0 \leq m_i, e_0 < 10$, $m_1 \neq 0$ sowie der Zahl 0, also insgesamt $9 \cdot 10 \cdot 10 \cdot 19 \cdot 2 + 1 = 34201$ Zahlen.

b) $\mathbb{F}(2, 2, 1)$ besteht aus Zahlen der Form

$$x = \pm(m_1 \cdot \frac{1}{2} + m_2 \cdot \frac{1}{4}) \cdot 2^{\pm e_0}$$

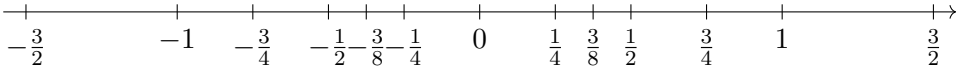
Wie lautet die Darstellung zur Basis 2?

mit $m_1 = 1$, $0 \leq m_2, e_0 \leq 1$ sowie der Null. Beachte, dass hier die Darstellung zur Basis 10 erfolgt! Damit ergibt sich

$$\mathbb{F}(2, 2, 1) = \left\{ -\frac{3}{2}, -1, -\frac{3}{4}, -\frac{1}{2}, -\frac{3}{8}, -\frac{1}{4}, 0, \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{3}{4}, 1, \frac{3}{2} \right\}.$$

Im Fall $\beta = 2$ bedeutet die Normierung, dass die Stelle $m_1 = 1$ festgelegt ist. Damit hat man eigentlich $r + 1$ Stellen in der Mantisse zur Verfügung ohne den Speicheraufwand zu erhöhen. Dies nennt man *hidden bit.*

Stellt man $\mathbb{F}(2, 2, 1)$ auf dem Zahlenstrahl dar, ergibt sich



Der minimale Abstand zweier Zahlen in $\mathbb{F}(2, 2, 1)$ ist $\frac{1}{8}$ allerdings wird dieser nicht bei der Null erreicht – ein Nachteil der Normierung. In praktischen Implementierung normierter Fließkommazahlen gibt man daher die Normierung nahe der Null auf. □

Zahlenbereich In einem gegebenen, normierten Fließkommasystem $\mathbb{F}(\beta, r, s)$ ist die größte bzw. kleinste darstellbare Zahl gegeben durch

$$X_{\pm} = \pm(\beta - 1)\left(\sum_{i=1}^{r} \beta^{-i}\right) \cdot \beta^{(\beta-1)\sum_{j=0}^{s-1}\beta^j}.$$

Die kleinste positive bzw. größte negative Zahl lautet

$$x_{\pm} = \pm\beta^{-1-(\beta-1)\sum_{j=0}^{s-1}\beta^j}$$

und damit gilt

$$\mathbb{F}(\beta, r, s) \subset D(\beta, r, s) = [X_-, x_-] \cup \{0\} \cup [x_+, X_+].$$

Abstände zwischen Fließkommazahlen Wie in obigem Beispiel 2.3 schon ersichtlich sind die Abstände zwischen Fließkommazahlen nicht gleich. Aufschluss gibt folgende Beobachtung.

Beobachtung 2.4. Es werde das Fließkommasystem $\mathbb{F}(\beta, r, s)$ betrachtet.

a) Für zwei Fließkommazahlen $x \neq x'$ im Bereich $\beta^{e-1} \leq x, x' \leq \beta^e$ gilt

$$|x - x'| \geq \beta^{e-r}$$

wobei $e = e(x) = e(x')$. Gilt $|x - x'| = \beta^{e-r}$ so bezeichnen wir die beiden Zahlen als *benachbart*.

b) Für zwei benachbarte Fließkommazahlen $x \neq x'$ im Bereich $\beta^{e-1} \leq x, x' \leq \beta^e$ gilt

$$\beta^{-r} < \frac{|x - x'|}{|x|} \leq \beta^{1-r}.$$

Dieser sogenannte *relative* Abstand zweier benachbarter Zahlen schwankt also maximal um einen Faktor β. Daher sind kleine Werte für die Basis β in dieser Hinsicht von Vorteil.

Beweis. a) Wegen $\beta^{e-1} \leq x < x' \leq \beta^e$ gilt $x = m(x)\beta^e$ und $x' = m(x')\beta^e$ mit gleichem Exponenten e, dabei erlauben wir ausnahmsweise auch die Mantisse $m(x') = 1$ im Fall $x' = \beta^e$ (eigentlich wäre dann $m(x') = \beta^{-1}$ und $e(x) = e + 1$ aber dies ändert nichts an der Zahl x', analog dürfte auch $x = \beta^e$ sein). Außerdem ist $m(x), m(x') > 0$. In diesem Bereich gilt $|m(x) - m(x')| \geq \beta^{-r}$ dabei ist β^{-r} die Wertigkeit der kleinsten Mantissenstelle. Somit erhalten wir

$$|x - x'| = |m(x)\beta^e - m(x')\beta^e| = |m(x) - m(x')|\beta^e \geq \beta^{e-r}.$$

b) Es sind $x < x'$ benachbarte Fließkommazahlen im Bereich $\beta^{e-1} \leq x, x' \leq \beta^e$. Mit dem Wissen von a) gilt

$$\frac{|x - x'|}{|x|} = \frac{\beta^{e-r}}{m(x)\beta^e} = \frac{\beta^{-r}}{m(x)}.$$

Aufgrund der Normierung gilt $\beta^{-1} \leq m(x) \leq 1 \Leftrightarrow 1 \geq m(x)^{-1} \leq \beta$ und damit gilt insgesamt

$$\beta^{-r} \geq \frac{|x - x'|}{|x|} \leq \beta^{1-r}.$$

□

2.3 IEEE 754 Standard

Dieser wurde 1985 verabschiedet um die Fließkommadarstellung und auch Arithmetik in Rechnern zu standardisieren. Mit dem Ziel, dass auch Programme die Fließkommazahlen verarbeiten auf verschiedenen Rechnern das gleiche Ergebnis liefern. Der Standard fordert $\beta = 2$ mit vier Genauigkeitsstufen:

Größe	single	single-ext	double	double-ext
e_{max}	127	≥ 1024	1023	≥ 16384
e_{min}	-126	≤ -1021	-1022	≤ -16381
Bits Exponent	8	≤ 11	11	15
Bits insgesamt	32	≥ 43	64	≥ 79

Betrachten wir die `double`-Genauigkeit genauer. Insgesamt stehen 64 Bit zur Verfügung, davon 11 Bit für den Exponenten. Der Exponent wird als vorzeichenlose Zahle $c \in [1, 2046]$ gespeichert. Es gilt dann $-1022 \leq e = c - 1023 \leq 1023$. Die Werte $c \in \{0, 2047\}$ werden zusammen mit der Mantisse anderweitig genutzt:

c	m	Situation
0	0	kodiert die Null
0	$\neq 0$	denormalisierte Darstellung
2047	$\neq 0$	kodiert not a number
2047	0	kodiert ∞ (Überlauf)

Für die Mantisse stehen $64 - 11 = 53$ Bit zur Verfügung. Ein Bit wird für das Vorzeichen benötigt, es bleiben 52 Bit. Wegen des hidden bit ist die Wertigkeit der kleinsten Mantissenstelle allerdings wieder 2^{-53}.

Außerdem stellt der Standard vier Rundungsarten zur Verfügung: Aufrunden, Abrunden, zur Null hin runden und die natürliche Rundung (siehe unten). Schließlich sind die Fließkommaoperationen exakt gerundet (wird ebenfalls unten erläutert).

2.4 Runden und Rundungsfehler

Um $x \in \mathbb{R}$ zu approximieren benötigen wir eine Abbildung

$$\mathrm{rd} : D(\beta, r, s) \to \mathbb{F}(\beta, r, s).$$

Die Abbildung setzt also voraus, dass x im prinzipiell darstellbaren Bereich liegt. Ansonsten sind ist der Parameter s entsprechend zu vergrößern. Eine sinnvolle Forderung für rd ist in jedem Fall

$$|x - \mathrm{rd}(x)| = \min_{y \in \mathbb{F}(\beta,r,s)} |x - y| \qquad \text{(Bestapproximation)}.$$

Wir setzen

$$l(x) = \max\{y \in \mathbb{F}(\beta, r, s)\} : y \leq x\}, \qquad r(x) = \min\{y \in \mathbb{F}(\beta, r, s)\} : y \geq x\}$$

und

$$\mathrm{rd}(x) = \begin{cases} x & l(x) = r(x), \text{ i.e. } x \in \mathbb{F} \\ l(x) & |x - l(x)| < |x - r(x)| \\ r(x) & |x - r(x)| < |x - l(x)| \\ ? & |x - r(x)| = |x - l(x)| \end{cases} \tag{2.2}$$

Im letzten Fall, wenn x genau zwischen $l(x)$ und $r(x)$ ist, gibt es verschiedene Möglichkeiten, von denen wir zwei weiter unten betrachten wollen.

Die Abbildung rd soll nun konkret im Computer realisiert werden. Da entsteht zunächst die Frage, wie die Zahl $x \in \mathbb{R}$, die gerundet werden soll, überhaupt in den Computer gelangt. Dort soll sie ja gerade in einem Fließkommasystem dargestellt werden. Daher wollen wir uns hier auf den Fall

$$\mathrm{rd} : \mathbb{F}(\beta, r', s) \to \mathbb{F}(\beta, r, s), \qquad r' > r,$$

beschränken, d.h es wird von einem Fließkommasystem mit mehr Mantissenstellen in eines mit weniger Mantissenstellen gerundet. Bei der Rundung transzendenter Zahlen wie e oder π kann das *Tabellenmacherdilemma* entstehen: Zunächst kann jede reelle Zahl als Fließkommazahl mit unendlicher Zahl von Mantissenstellen dargestellt werden, $x = \mathrm{sign}(x) \left(\sum_{i=1}^{\infty} m_i \beta^{-i}\right) \beta^e$, und es entsteht die Frage ob die Ziffern m_i der Reihe nach berechnet werden können. Nun werden transzendente Zahlen als Reihen entwickelt (siehe etwa Beispiel 2.1) und da könnte folgende Situation entstehen:

$$\begin{aligned} x_n &= 5.0835 \\ x_{n+1} &= 5.0835000 \\ x_{n+2} &= 5.083500000 \\ &\vdots \end{aligned}$$

und man kann a-priori nicht angeben nach wievielen Schritten $x < 5.0835$ oder $x > 5.0835$ entschieden werden kann. Wir nehmen also nun $x \in \mathbb{F}(\beta, r', s)$, $r' > r$ an und definieren die beiden Rundungsarten:

1) *Natürliche (bzw. kaufmännische) Rundung:* Hier setzt man

$$\mathrm{rd}(x) = \begin{cases} \mathrm{sign}(x) \left(\sum_{i=1}^{r} m_i \beta^{-i}\right) \beta^e & \text{falls } m_{r+1} < \beta/2, \\ \mathrm{sign}(x) \left[\left(\sum_{i=1}^{r} m_i \beta^{-i}\right) + \beta^{-r}\right] \beta^e & \text{falls } m_{r+1} \geq \beta/2. \end{cases}$$

   Zur Entscheidung wird die Stelle m_{r+1} herangezogen, es genügt also im Prinzip $r' = r + 1$. Im Sinne von Gleichung (2.2) entscheidet sich die natürliche Rundung im Fall $|x - r(x)| = |x - l(x)|$ immer für den betragsmäßig größeren Wert, d.h. positive Zahlen werden aufgerundet, negative Zahlen werden abgerundet.

2) *Gerade Rundung:* Hier setzt man β als gerade Zahl voraus und definiert

$$\mathrm{rd}(x) = \begin{cases} x & l(x) = r(x), \text{ i.e. } x \in \mathbb{F} \\ l(x) & |x - l(x)| < |x - r(x)| \\ r(x) & |x - r(x)| < |x - l(x)| \\ l(x) & |x - r(x)| = |x - l(x)| \wedge m_r \text{ gerade} \\ r(x) & |x - r(x)| = |x - l(x)| \wedge m_r \text{ ungerade} \end{cases}$$

   Hier rundet man im Fall $|x - r(x)| = |x - l(x)|$ sowohl auf als auch ab, in Abhängigkeit der Ziffer m_r. Sind diese statistisch gleichmäßig verteilt so rundet man genauso oft ab wie auf und vermeidet dadurch mögliche Verzerrungen von Statistiken die bei der kaufmännischen Rundung entstehen können. Die Entscheidung $|x - r(x)| = |x - l(x)|$ erfordert Kenntnis aller r' Mantissenstellen.

Beispiel 2.5. Gegeben sei $x_1 = 0.32642876$. Sowohl kaufmännisches als auch gerades Runden ergeben bei Rundung auf zwei Mantissenstellen 0.33 und bei Rundung auf drei Mantissenstellen 0.326.

Für $x_2 = 0.74500$ und Rundung auf 2 Mantissenstellen ergibt kaufmännisches Runden 0.75 und gerades Runden 0.74.

Für $x_3 = 0.75500$ und Rundung auf 2 Mantissenstellen ergibt kaufmännisches Runden 0.76 und gerades Runden ebenfalls 0.76. □

Über den maximal möglichen Fehler der bei *einer* Rundungsoperation entsteht gibt das folgende Lemma Auskunft. Dabei unterscheiden wir zwischen absolutem und relativem Fehler die wir zunächst einführen.

Definition 2.6 (Relativer und absoluter Fehler)**.** Wird eine Größe x durch eine Größe x' angenähert, dann bezeichnet

$$e_{\text{abs}} = x - x' \tag{2.3}$$

den absoluten Fehler und

$$e_{\text{rel}} = \frac{x - x'}{x} \qquad (x \neq 0) \tag{2.4}$$

den relativen Fehler in der Näherung. □

Relative Fehler sind in der Praxis oft relevanter als absolute Fehler. So ist bei einer Aussage “Der Unterschied beider Werte beträgt ein Meter"nicht klar ob dies ein Großer oder kleiner Unterschied ist. Beim Hochsprung wäre ein Unterschied von einem Meter relativ viel, bei der Entfernung Heidelberg-Hamburg relativ wenig.

Lemma 2.7 (Rundungsfehler)**.** Es sei $x \in D(\beta, r, s)$ und $\text{rd} : D(\beta, r, s) \to \mathbb{F}(\beta, r, s)$ eine der oben beschriebenen Rundungsarten. Dann gilt für den absoluten Rundungsfehler

$$|x - \text{rd}(x)| \leq \frac{1}{2}\beta^{e(x)-r} \tag{2.5}$$

und für den relativen Rundungsfehler

$$\frac{|x - \text{rd}(x)|}{|x|} \leq \frac{1}{2}\beta^{1-r} \qquad x \neq 0. \tag{2.6}$$

Beweis. Es sei $x = m(x)\beta^{e(x)}$ die normierte Fließkommadarstellung von x mit *möglicherweise unendlich großer Stellenzahl* (wir werden die genaue Mantisse $m(x)$ nicht benötigen).

$l(x)$ und $r(x)$ sind benachbarte Fließkommazahlen. Somit gilt wegen Beobachtung 2.4

$$|r(x) - l(x)| = \beta^{e(x)-r}.$$

Der maximale Rundungsfehler ergibt sich für $x = (l(x) + r(x))/2$, also

$$|x - \text{rd}(x)| \leq \left| \frac{l(x) + r(x)}{2} - l(x) \right| = \frac{1}{2}|r(x) - l(x)| = \frac{1}{2}\beta^{e(x)-r}.$$

Dies beweist die erste Aussage. Für den relativen Fehler bei $x \neq 0$ gilt

$$\frac{|x - \text{rd}(x)|}{|x|} \leq \frac{\frac{1}{2}\beta^{e(x)-r}}{|m(x)|\beta^{e(x)}} = \frac{1}{2}\frac{1}{|m(x)|}\beta^{-r} \leq \frac{1}{2}\beta^{1-r}.$$

Dabei haben wir die Normierung $|m(x)| \geq \beta^{-1}$ ausgenutzt. □

Die Zahl $\text{eps} = \frac{1}{2}\beta^{1-r}$ heißt *Maschinengenauigkeit*, *machine precision* oder *machine epsilon*.

Vorlesung 3

Rechnen mit Fließkommazahlen

3.1 Fließkommaarithmetik

Nun möchte man mit Fließkommazahlen natürlich auch rechnen, d.h. wir benötigen zweistellige Operationen

$$\circledast \ : \ \mathbb{F} \times \mathbb{F} \to \mathbb{F} \qquad \circledast \in \{\oplus, \ominus, \odot, \oslash\}, \tag{3.1}$$

welche den Grundrechenarten $+, -, \cdot, /$ auf den reellen Zahlen entsprechen. In diesem Kapitel steht $\mathbb{F}$ für ein beliebiges Fließkommasystem $\mathbb{F}(\beta, r, s)$. Die genauen Parameter werden nur dann spezifiziert wenn dies wichtig ist.

Die Problematik der Fließkommaoperationen ist, dass in der Regel selbst für zwei Fließkommazahlen $x, y \in \mathbb{F}$ das Ergebnis der exakten Verknüpfung $x * y$ nicht in notwendigerweise in $\mathbb{F}$ liegt. So wird sich in der Regel bei der Multiplikation die Zahl der Mantissenstellen verdoppeln. Auch das Ergebnis einer Addition zweier Zahlen mit verschiedenen Exponenten erfordert in der Regel mehr Mantissenstellen.

Ein naheliegende Definition für die Fließkommaarithmetik ist daher

$$x \circledast y = \mathrm{rd}(x * y) \qquad * \in \{+, -, \cdot, /\}. \tag{3.2}$$

Ein Fließkommaarithmetik mit dieser Eigenschaft nennt man *exakt gerundet*.

Eine triviale Realisierung exakter Rundung ist jedoch nicht effizient. Betrachten wir die Addition von $x = 10^{10}$ und $y = 10^{-10}$ in $\mathbb{F}(10, 4, 2)$, Bevor die Mantissen addiert werden können sind beide Summanden auf den gleichen Exponenten zu bringen, also $x = 0.1 \cdot 10^{11}$ und $y = 10^{-10} = 10^{-10-11} \cdot 10^{11} = 10^{-21} \cdot 10^{11}$. Dementsprechend bräuchte man 21 Nachkommstellen um in einem Zwischenschritt $x*y$ exakt zu berechnen. Die anschließende Rundung schneidet viele der zusätzlichen Stellen wieder ab, das ist sehr ineffizient. Die Rundung gleich nach Egalisierung der Exponenten durch zuführen ist allerdings gefährlich und kann zu großen relativen Fehlern führen. Es zeigt sich, dass folgende Vorgehensweise zum Ziel führt:

1) Bringe beide Argumente auf den gleichen Exponenten, also die betragsmäßig kleinere Zahl auf den Exponenenten der betragsmäßig Größeren. Diese ist dann nicht mehr normiert.

2) Runde die geschobene Zahl auf $r + 2$ Mantissenstellen.

3) Addiere beide Zahlen mit $r + 2$ Mantissenstellen. Diese zusätzlichen Stellen nennt man *guard digits*.

4) Runde das Ergebnis auf r Stellen.

Mit zwei zusätzlichen Stellen erreicht man so ein exakte Rundung. Im IEEE 754 Standard sind sowohl die Grundoperationen als auch die Berechnung der Quadratwurzel exakt gerundet.

Die Fließkommaarithmetik unterscheidet sich von der Arithmetik in den reellen Zahlen, hier eine Liste relevanter Eigenschaften:

a) Es gibt Zahlen $y \in \mathbb{F}$ so dass $x \oplus y = x$ gilt.

b) Assoziativ und Distributivgesetz gelten nicht (unbedingt). Es kommt also auf die Reihenfolge der Operationen an. Z.B. gilt mit dem y aus a): $(x \oplus y) \ominus x = 0$ und $y \oplus (x \ominus x) = y$. Hier haben wir ausgenutzt, dass $x \ominus x = 0$.

c) Es gilt jedoch das Kommutativgesetz!

d) Auch folgende einfache Regeln gelten

$$\begin{aligned}(-x) \odot y &= -(x \odot y)\\ 1 \odot x &= x\\ x \odot y &= 0 \quad \text{nur dann wenn } x = 0 \text{ oder } y = 0\\ x \odot z &\leq y \odot z \quad \text{wenn } x \leq y \text{ und } z > 0\end{aligned}$$

3.2 Fehleranalyse

Wir kommen nun zu den Auswirkungen von Rundungsfehlern in Rechnungen die aus einer Folge von Grundoperation zusammengesetzt sind. Abstrakt betrachten wir die Berechnung als Auswertung einer vektorwertigen Funktion

$$F \;:\; \mathbb{R}^m \to \mathbb{R}^n.$$

Die Berechnung von F im Computer wird durch eine entsprechende Funktion

$$F' \;:\; \mathbb{F}^m \to \mathbb{F}^n$$

realisiert. F' wird durch endlich viele Elementaroperationen

$$F'(x') = \varphi_l(\dots \varphi_2(\varphi_1(x'))\dots)$$

realisiert. Jedes φ_i führt eine Grundoperation aus $\{\oplus, \ominus, \odot, \oslash\}$ durch und steuert einen unbekannten Teilfehler bei. Die numerische Realisierung F' von F ist in der Regel nicht eindeutig. Zum einen gibt es mathematisch äquivalente Möglichkeiten einen Ausdruck zu berechen, z.B $x(y+z) = xy + xz$ und selbst unterschiedlich Reihenfolgen der Zwischenschritte können zu unterschiedlichen Ergebnissen in $\mathbb{F}$ führen.

Im größeren praktischen Kontext sind auch die Eingaben der exakten Funktion F mit Unsicherheiten behaftet. Dann ist es interessant zu untersuchen wie sich Variationen in der Eingabe x gegenüber den akkumulierten Rundungsfehlern verhalten. Dies erreicht man mit der nun folgenden Analyse. Dazu betrachten wir die Aufspaltung

$$F(x) - F'(\mathrm{rd}(x)) = \underbrace{F(x) - F(\mathrm{rd}(x))}_{\text{Konditionsanalyse}} + \underbrace{F(\mathrm{rd}(x)) - F'(\mathrm{rd}(x))}_{\text{Rundungsfehleranalyse}}. \tag{3.3}$$

Die erste Differenz analysiert wie die exakte Abbildung F auf Änderungen in der Eingabe (hier Rundungsfehler) reagiert. Diese Analyse ist unabhängig von der Problematik der Fließkommaarithmetik. Die zweite Differenz analysiert wie sich die beiden Abbildungen F, F' bei der selben Eingabe Unterscheiden, also die eigentliche Rundungsfehleranalyse.

Schlussendlich wird man Fehlernormen analysieren. Dann folgt mittels Dreiecksungleichung

$$\|F(x) - F'(\mathrm{rd}(x))\| \leq \|F(x) - F(\mathrm{rd}(x))\| + \|F(\mathrm{rd}(x)) - F'(\mathrm{rd}(x))\|.$$

3.3 Differentielle Konditionsanalyse

Die folgende Analyse braucht den Satz von Tayler aus der Analysis. Da dieser Satz in der Vorlesung sehr häufig zitiert wird wollen wir ihn in zwei Varianten anführen.

Satz 3.1 (Satz von Taylor für Funktionen in einer Variable)**.** Sei $f : (a, b) \to \mathbb{R}$ $(r + 1)$ mal stetig differenzierbar und seien $x, z \in (a, b)$ gegeben. Dann gibt es ein $\xi \in (a, b)$ zwischen x und z (und abhängig von z), so dass gilt:

$$f(z) = \sum_{k=0}^{r} \frac{1}{k!} \frac{d^k f(x)}{dx^k} (z - x)^k + \frac{1}{(r+1)!} \frac{d^{r+1} f(\xi)}{dx^{r+1}} (z - x)^{r+1}. \tag{3.4}$$

Dabei heisst $t_n(x, z) = \sum_{k=0}^{r} \frac{1}{k!} \frac{d^k f(x)}{dx^k} (z-x)^k$ *Taylorpolynom* in z um den *Entwicklungspunkt* x und $R_r(x, z) = \frac{1}{(r+1)!} \frac{d^{r+1} f(\xi)}{dx^{r+1}} (z - x)^{r+1}$ *Lagranges Restglied.*

Beweis. Siehe [Rannacher, 2018a, Satz 5.8] □

Bemerkung 3.2. Oft setzt man $z = x + h$ und somit $z - x = h$ mit der Idee, dass $|h|$ "klein" ist. Damit lauten die Taylorformel und das Restglied alternativ

$$f(x + h) = \sum_{k=0}^{r} \frac{1}{k!} \frac{d^k f(x)}{dx^k} h^k + \frac{1}{(r+1)!} \frac{d^{r+1} f(\xi)}{dx^{r+1}} h^{r+1}. \tag{3.5}$$

Für Funktionen in höheren Dimensionen ist der Gradient ein nützliches Hilfsmittel, das wir im Folgenden wiederholt einsetzen werden.

Definition 3.3 (Gradient im $\mathbb{R}^n$)**.** Sei $D \subset \mathbb{R}^n$ eine offene Menge und $f : D \to \mathbb{R}$ partiell differenzierbar. Für $x \in D$ ist der Gradient von f definiert als

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix}.$$

Der Satz von Taylor kann auf $r + 1$ mal stetig partiell differenzierbare Funktionen $f : D \to \mathbb{R}$, mit D einer offenen Teilmenge des $\mathbb{R}^m$, erweitert werden. Um komplexe Notation zu vermeiden zitieren wir hier nur eine spezialisierte Variante des Satzes für $r = 1$. Mehr werden wir im folgenden nicht benötigen.

Satz 3.4 (Spezialfall des Satz von Taylor für Funktionen in mehreren Variablen)**.** Sei $D \subset \mathbb{R}^m$ eine offene Menge und $f : D \to \mathbb{R}$ zweimal stetig differenzierbar. Weiter seien $x, \eta \in \mathbb{R}^m$ derart, dass $x + s\eta \in D$ für alle $s \in [0, 1]$. Dann gibt es zu so einem η ein $\theta \in (0, 1)$ so dass

$$\begin{aligned} f(x + \eta) &= f(x) + \nabla f(x) \cdot \eta \\ &\quad + \frac{1}{2} \sum_{i=1}^{m} \frac{\partial^2 f(x + \theta\eta)}{\partial x_i^2} \eta_i^2 + \sum_{i=1}^{m} \sum_{i<j\leq m} \frac{\partial^2 f(x + \theta\eta)}{\partial x_i \partial x_j} \eta_i \eta_j. \end{aligned} \tag{3.6}$$

Dabei ist in der zweiten Zeile das Restglied $R_1(x, \eta) = \frac{1}{2} \sum_{i=1}^{m} \frac{\partial^2 f(x+\theta\eta)}{\partial x_i^2} \eta_i^2 + \sum_{i=1}^{m} \sum_{i<j\leq m} \frac{\partial^2 f(x+\theta\eta)}{\partial x_i \partial x_j} \eta_i \eta_j$ in Lagranger Form angegeben.

Beweis. Folgerung aus [Rannacher, 2018b, Satz 3.6] □

Bemerkung 3.5. Oft benötigen wir Abschätzungen des Restglieds. Mit $h := \max_{1\le i\le n} |\eta_i|$ gilt dann

$$|R_1(x,\eta)| \le \left(\frac{1}{2}\sum_{i=1}^{m} \left| \frac{\partial^2 f(x+\theta\eta)}{\partial x_i^2} \right| + \sum_{i=1}^{m} \sum_{i<j\le m} \left| \frac{\partial^2 f(x+\theta\eta)}{\partial x_i \partial x_j} \right| \right) h^2.$$

Für $h \to 0$ konvergiert der Wert in der Klammer und der Betrag des Restgliedes "geht wie h^2" gegen Null. Man schreibt hierfür auch "$|R_1(x,\eta)| = O(h^2)$".

Zur allgemeinen Quantifizierung der Konvergenzgeschwindigkeit dienen die

Definition 3.6 (Landau Symbole)**.** a) Man schreibt

$$g(t) = O(h(t)) \quad (t \to 0)$$

falls es ein $t_0 > 0$ und $c_0 \ge 0$ gibt so dass für alle $0 < t \le t_0$ die Abschätzung

$$|g(t)| \le c_0 |h(t)|$$

gilt. Man sagt "$g(t)$ geht mindestens wie $h(t)$ gegen Null".

b) Weiter schreibt man

$$g(t) = o(h(t)) \quad (t \to 0)$$

wenn es ein $t_0 > 0$ und eine Funktion $c(t)$ mit $\lim_{t\to 0} c(t) = 0$ gibt, so dass für alle $0 < t \le t_0$ die Abschätzung

$$|g(t)| \le c(t) |h(t)|$$

gilt. Geht $h(t)$ ebenfalls gegen Null dann drückt dies aus: "$g(t)$ geht schneller als $h(t)$ gegen Null".

c) Schließlich schreiben wir

$$g(t) \doteq h(t)$$

und sagen „$g(t)$ ist *in erster Näherung gleich* $h(t)$", wenn gilt

$$g(t) - h(t) = o(t).$$

□

Aus dem Satz von Taylor folgt mittels der Beobachtung $g(t) = O(t^2) \Rightarrow g(t) = o(t)$

$$f(x+\eta) \doteq f(x) + \sum_{i=1}^{m} \frac{\partial f(x)}{\partial x_i} \eta_i$$

Nach diesem Ausflug in die Analysis wenden wir uns nun wieder der Konditionsanalyse zu. Zur Analyse der Auswirkung von Änderungen in der Eingabe der Funktion F betrachten wir diese als zweimal stetig differenzierbare Abbildung. Nach dem Satz von Taylor 3.4 gilt damit für jedem Komponente F_i:

$$F_i(x+\Delta x) = F_i(x) + \sum_{j=1}^{m} \frac{\partial F_i}{\partial x_j}(x) \Delta x_j + R_{1,i}(x,\Delta x), \quad i = 1, \dots n, \qquad (3.7)$$

mit einem Restglied $R_{1,i}(x, \Delta x) = O(h^2)$ und $h = \max_{1 \le i \le n} |\Delta x_i|$. Wir formen um und gehen zur $\doteq$-Notation über:

$$F_i(x + \Delta x) - F_i(x) \doteq \sum_{j=1}^{m} \frac{\partial F_i}{\partial x_j}(x) \Delta x_j, \qquad i = 1, \dots n.$$

Für die relative Änderung erhalten wir dann

$$\frac{F_i(x + \Delta x) - F_i(x)}{F_i(x)} \doteq \sum_{j=1}^{m} \frac{\partial F_i}{\partial x_j}(x) \frac{\Delta x_j}{F_i(x)} = \sum_{j=1}^{m} \underbrace{\left(\frac{\partial F_i}{\partial x_j}(x) \frac{x_j}{F_i(x)} \right)}_{\text{Verstärkungsfaktor } k_{ij}(x)} \frac{\Delta x_j}{x_j}, \quad i = 1, \dots n.$$

Der Verstärkungsfaktor k_{ij} beschreibt wie stark sich die relative Änderung in der j-ten Komponente der Eingabe auf die relative Änderung in der i-ten Komponente des Ergebnisses auswirkt.

Definition 3.7. Wir nennen die Auswertung $y = F(x)$ "schlecht konditioniert" im Punkt x falls $|k_{ij}(x)| \gg 1$, andernfalls heißt die Auswertung "gut konditioniert". Bei $|k_{ij}(x)| < 1$ spricht man von Fehlerdämpfung, bei $|k_{ij}(x)| > 1$ von Fehlerverstärkung. □

In der Aufspaltung (3.3) entspricht $\Delta x = \mathrm{rd}(x) - x$, also gerade dem absoluten Rundungsfehler in der Eingabe. Damit gilt $\frac{\Delta x_j}{x_j} = \frac{1}{2}\beta^{1-r}$.

Beispiel 3.8 (Konditionierung der Grundoperationen)**.** a) Wir untersuchen die Konditionierung der Addition, also $F(x_1, x_2) = x_1 + x_2$. Offensichtlich gilt $\frac{\partial F}{\partial x_1} = 1$, $\frac{\partial F}{\partial x_2} = 1$ und damit nach obiger Formel

$$\frac{F(x_1 + \Delta x_1, x_2 + \Delta x_2) - F(x_1, x_2)}{F(x_1, x_2)} = \left(1 \cdot \frac{x_1}{x_1 + x_2} \right) \frac{\Delta x_1}{x_1} + \left(1 \cdot \frac{x_2}{x_1 + x_2} \right) \frac{\Delta x_2}{x_2}.$$

Für $x_1 \to -x_2$ werden beide Verstärkungsfaktoren sehr groß. In diesem Fall ist die Addition schlecht konditioniert. Dies gilt analog für die Subtraktion falls $x_1 \to x_2$.

b) Für die Multiplikation $F(x_1, x_2) = x_1 \cdot x_2$ gilt $\frac{\partial F}{\partial x_1} = x_2$, $\frac{\partial F}{\partial x_2} = x_1$ und damit

$$\frac{F(x_1 + \Delta x_1, x_2 + \Delta x_2) - F(x_1, x_2)}{F(x_1, x_2)} = \left(x_2 \cdot \frac{x_1}{x_1 \cdot x_2} \right) \frac{\Delta x_1}{x_1} + \left(x_1 \cdot \frac{x_2}{x_1 \cdot x_2} \right) \frac{\Delta x_2}{x_2}.$$

Die Verstärkungsfaktoren sind somit beide 1 unabhängig von x_1, x_2. Die Multiplikation ist eine gut konditionierte Operation!

c) Schließlich betrachten wir noch die Funktion $F(x_1, x_2) = x_1^2 - x_2^2$. Offensichtlich gilt $\frac{\partial F}{\partial x_1} = 2x_1$, $\frac{\partial F}{\partial x_2} = -2x_2$ und damit nach obiger Formel

$$\frac{F(x_1 + \Delta x_1, x_2 + \Delta x_2) - F(x_1, x_2)}{F(x_1, x_2)} = \left(2x_1 \cdot \frac{x_1}{x_1^2 - x_2^2} \right) \frac{\Delta x_1}{x_1} + \left(2x_2 \cdot \frac{x_2}{x_2^2 - x_1^2} \right) \frac{\Delta x_2}{x_2}.$$

3.4 Rundungsfehleranalyse

In der eigentlichen Rundungsfehleranalyse betrachten wir entsprechend der Aufspaltung (3.3) die Differenz $F(x) - F'(x)$ für Maschinenzahlen $x \in \mathbb{F}$. Entspricht F genau einer Rechenoperation $\circledast$ so gilt wegen der exakten Rundung

$$\frac{(x * y) - (x \circledast y)}{x * y} = \epsilon \qquad |\epsilon| \leq \text{eps} = \frac{1}{2}\beta^{1-r}$$

Der genaue Rundungsfehler ϵ hängt von den Argumenten x und y ab und ist damit für jede Operation verschieden. Umstellen der letzten Beziehung liefert

$$x \circledast y = (x * y)(1 + \epsilon) \qquad \text{für ein } |\epsilon(x, y)| \leq \text{eps}$$

In der Rundungsfehleranalyse lassen sich ebenfalls Verstärkungsfaktoren ähnlich wie in der Konditionsanalyse herleiten. Dies illustrieren wir mit Beispielen.

Beispiel 3.9. Wir untersuchen die Abbildung $F(x_1, x_2) = x_1^2 - x_2^2$ und zwei verschieden Realisierungen

$$F_a(x_1, x_2) = (x_1 \odot x_1) \ominus (x_2 \odot x_2), \; F_b(x_1, x_2) = (x_1 \ominus x_2) \odot (x_1 \oplus x_2).$$

a) In diesem Fall erhalten wir für die ersten beiden Operationen

$$\begin{aligned} u = x_1 \odot x_1 &= x_1^2(1 + \epsilon_1) && |\epsilon_1| \leq \text{eps}, \\ v = x_2 \odot x_2 &= x_2^2(1 + \epsilon_2) && |\epsilon_2| \leq \text{eps} \end{aligned}$$

und dann

$$\begin{aligned} F_a(x_1, x_2) &= u \ominus v \\ &= (x_1^2(1 + \epsilon_1) - x_2^2(1 + \epsilon_2))(1 + \epsilon_3) \\ &= (x_1^2 + \epsilon_1 x_1^2 - x_2^2 - \epsilon_2 x_2^2)(1 + \epsilon_3) \\ &= x_1^2 - x_2^2 + (\epsilon_1 + \epsilon_3)x_1^2 - (\epsilon_2 + \epsilon_3)x_2^2 + \epsilon_1\epsilon_3 x_1^2 - \epsilon_2\epsilon_3 x_2^2. \end{aligned}$$

Damit erhalten wir in erster Näherung für den relativen Rundungsfehler

$$\frac{F_a(x_1, x_2) - F(x_1, x_2)}{F(x_1, x_2)} \doteq \frac{x_1^2}{x_1^2 - x_2^2}(\epsilon_1 + \epsilon_3) + \frac{x_2^2}{x_2^2 - x_1^2}(\epsilon_2 + \epsilon_3).$$

Ein Vergleich mit Beispiel 3.8 ergibt bis auf Konstanten die gleichen Verstärkungsfaktoren.

b) Für die zweite Variante ergibt sich

$$\begin{aligned} u = x_1 \ominus x_2 &= (x_1 - x_2)(1 + \epsilon_1) && |\epsilon_1| \leq \text{eps}, \\ v = x_1 \oplus x_2 &= (x_1 + x_2)(1 + \epsilon_2) && |\epsilon_2| \leq \text{eps} \end{aligned}$$

und dann

$$\begin{aligned} F_b(x_1, x_2) &= u \odot v \\ &= (u \cdot v)(1 + \epsilon_3) \\ &= ((x_1 - x_2)(1 + \epsilon_1) \cdot (x_1 + x_2)(1 + \epsilon_2))(1 + \epsilon_3) \\ &= (x_1 - x_2)(x_1 + x_2)(1 + \epsilon_1)(1 + \epsilon_2)(1 + \epsilon_3) \\ &= (x_1^2 - x_2^2)(1 + \epsilon_1 + \epsilon_2 + \epsilon_3 + \ldots \epsilon_1\epsilon_2\epsilon_3). \end{aligned}$$

Damit erhalten wir in erster Näherung für den relativen Rundungsfehler

$$\frac{F_b(x_1, x_2) - F(x_1, x_2)}{F(x_1, x_2)} \doteq \epsilon_1 + \epsilon_2 + \epsilon_3.$$

Hier ist der Verstärkungsfaktor 1!. □

In der Gesamtschau sind Konditionsanalyse und Rundungsfehleranalyse gemeinsam zu betrachten. Dazu die folgende

Definition 3.10. Wir nennen einen numerischen Algorithmus *numerisch stabil*, wenn die Verstärkungsfaktoren aus der Rundungsfehleranalyse die aus der Konditionsanalyse nicht übersteigen.

Nach dieser Definition sind beide Realisierungen aus Beispiel 3.9 numerisch stabil.

3.5 Die quadratische Gleichung

Als ein weiteres, wichtiges Beispiel betrachten wir die Lösung der quadratischen Gleichung mit der p, q-Formel. Dieses Beispiel zeigt auch, dass die Minimierung von Rundungsfehlern für jedes Problem speziell betrachtet werden muss.

Die Gleichung

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

hat für $p^2/4 > q \neq 0$ die beiden reellen und verschiedenen Lösungen

$$x_{1/2} = f_\pm(p, q) = \frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}.$$

Die *Konditionsanalyse* der (beiden!) Abbildungen $f_\pm(p, q)$ liefert (Berechnung der partiellen Ableitungen):

$$\frac{f_\pm(p + \Delta p, q + \Delta q) - f(p, q)}{f(p, q)} \doteq$$
$$\left(1 \pm \frac{p}{2\sqrt{\frac{p^2}{4} - q}}\right) \frac{p}{p \pm 2\sqrt{\frac{p^2}{4} - q}} \frac{\Delta p}{p} \mp \frac{q}{\sqrt{\frac{p^2}{4} - q}\left(p \pm 2\sqrt{\frac{p^2}{4} - q}\right)} \frac{\Delta q}{q}.$$

Daraus ersehen wir die folgenden Fälle

1) Für $\frac{p^2}{4} \to q$ sind beide Lösungen schlecht konditioniert, da beide Verstärkungsfaktoren unweigerlich groß werden.

2) Für $q \to 0$ und $|p|$ weg von der Null kommt es auf das Vorzeichen von p an. Für $p < 0$ ist die negative Lösung $f_-(p, q) = \frac{p}{2} - \sqrt{\frac{p^2}{4} - q}$ in jedem Fall gut konditioniert und für $p > 0$ ist die positive Lösung $f_+(p, q) = \frac{p}{2} + \sqrt{\frac{p^2}{4} - q}$ in jedem Fall gut konditioniert.

Für die *Rundungsfehler* ist die Situation ähnlich. Für den ersten Fall oben kann man Probleme nicht vermeiden, es tritt unweigerlich Auslösung ein. Im zweiten Fall ist es vermeidet man Auslöschung in dem man die jeweils andere Lösung über den "Satz von Vieta"

$$x_1 + x_2 = p, \qquad x_1 \cdot x_2 = q,$$

ermittelt. D.h. man berechnet zunächst

$$w = \sqrt{(p \odot p) \oslash 4 \ominus q}.$$

Im Fall $p < 0$ rechnet man dann

$$x_2 = p \oslash 2 \ominus w, \qquad x_1 = q \oslash x_2$$

und im Fall $p > 0$ entsprechend

$$x_1 = p \oslash 2 \oplus w, \qquad x_2 = q \oslash x_1.$$

3.6 Auslöschung

Aus Beispiel 3.8 a) und Beispiel 3.9 a) entnehmen wir, dass die Addition (bzw. Subtraktion) die gefährlichen Operationen sind (da $x - y = x + (-y)$ sind Addition und Subtraktion keine verschiedenen Operationen). Das Grundübel ist die sogenannte Auslöschung, die immer dann auftritt wenn annähernd gleiche Zahlen von einander subtrahiert werden, bzw. wenn betragsmäßig annähernd gleiche Zahlen mit verschiedenen Vorzeichen addiert werden. Dabei treten Probleme erst dann auf, wenn die beiden Operanden selbst schon fehlerbehaftet sind (was aber in der Regel bei längeren Rechnungen der Fall ist). Wir illustrieren das mit einem Beispiel.

Beispiel 3.11 (Zur Auslöschung)**.** a) Für zwei beliebige *Maschinenzahlen* $x_1, x_2 \in \mathbb{F}$ gilt (unter der Bedingung $x_1 - x_2 \neq 0$) wegen der exakten Rundung der Fließkommaarithmetik (3.2):

$$\left| \frac{(x_1 \ominus x_2) - (x_1 - x_2)}{x_1 - x_2} \right| \leq \text{eps}.$$

b) Auslöschung tritt erst auf, wenn die beiden Argumente *selbst schon mit Fehlern behaftet sind.* Woher diese Fehler kommen ist dabei unerheblich. So kann in Beispiel 3.9 a) der Verstärkungsfaktor sehr groß werden wenn $|x_1 - x_2|$ klein ist. Sind $x_1 = m_1\beta^e, x_2 = (m_1 - \beta^{-r})\beta^e$ Maschinenzahlen, ist ein Verstärkungsfaktor der Größenordnung

$$\frac{x_1^2}{x_1^2 - x_2^2} = \frac{x_1^2}{(x_1 - x_2)(x_1 + x_2)} \approx \frac{m_1^2\beta^{2e}}{\beta^{-r}\beta^e 2m_1\beta^e} \approx \beta^{r-1}$$

möglich!

c) Betrachte die Berechnung $F'(\text{rd}(x_1), \text{rd}(x_2)) = \text{rd}(x_1) \ominus \text{rd}(x_2)$ in $\mathbb{F}(10, 4, 1)$ an einem konkreten Beispiel. Als Eingabe seien $x_1 = 0.11258762 \cdot 10^2$ und $x_2 = 0.11244891 \cdot 10^2$ gegeben. Offensichtlich gilt $x_1, x_2 \neq \mathbb{F}(10, 4, 1)$ und wir betrachten den relativen Fehler *inklusive Rundung* der Eingaben. Als exaktes Ergebnis erhalten wir

$$x_1 - x_2 = 0.13871 \cdot 10^{-1}.$$

Die (kaufmännische) Rundung der Eingaben liefert

$$\text{rd}(x_1) = 0.1126 \cdot 10^2, \quad \text{rd}(x_2) = 0.1124 \cdot 10^2$$

und die anschließende Fließkommaoperation liefert

$$\mathrm{rd}(x_1) \ominus \mathrm{rd}(x_2) = 0.2 \cdot 10^{-1}$$

wobei im übrigen *kein* weiterer Rundungsfehler eingeführt wird! Trotzdem ist keine Stelle im Gesamtergebnis korrekt und es ergibt sich als relativer Gesamtfehler

$$\frac{0.2 \cdot 10^{-1} - 0.13871 \cdot 10^{-1}}{0.13871 \cdot 10^{-1}} \approx 0.44 \approx 883 \frac{10^{-3}}{2} = 883\,\mathrm{eps}.$$

□

Als Regel kann man sich merken: *Setze die potentiell gefährlichen Operationen* $\oplus$, $\ominus$ *möglichst früh ein.*

Vorlesung 4

Analyse von Netzwerken

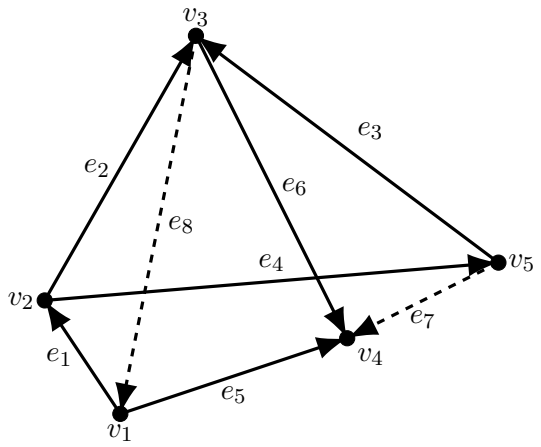
In dieser Vorlesung kommen wir noch einmal auf den Aspekt der Modellbildung zurück. Um die Wichtigkeit der Lösung (großer) linearer Gleichungssysteme zu demonstrieren stellen wir das Knotenpotentialverfahren zur Analyse der Flüssen in Netzwerken vor.

4.1 Rohrleitungsnetze

Wir betrachten ein Netzwerk wie es in Städten zur Wasser-, Gas- oder Stromversorgung vorkommt. Um ein konkretes Beispiel vor Augen zu haben beschränken wir uns auf das Wasserleitungsnetzwerk, das Verfahren kann aber auf die anderen Anwendungen übertragen werden.

Ein Wasserleitungsnetzwerk besteht aus Rohren unterschiedlichen Durchmessers, welche an Verbindungsstellen miteinander verknüpft sind. Wir nehmen an, dass die Rohre vollständig mit Wasser gefüllt sind und dass die Menge an Wasser in den Verbindungsstellen vernachlässigbar ist. Zusätzlich zu den normalen Rohrleitungen gibt es noch Pumpen die Wasser von einer Verbindungsstelle zu einer anderen Verbindungsstelle pumpen.

Abstrakt wird ein Wasserleitungsnetzwerk als ein Graph dargestellt. Folgende Abbildung gibt ein Beispiel:



Ein Graph $G = (V, E)$ besteht aus einer Menge von Knoten V und einer Menge von Kanten $E \subseteq V \times V$. Eine (gerichtete) Kante $e = (v, w)$ verbindet den Knoten v mit dem Knoten w. An einem Knoten werden mindestens zwei Rohre bzw. Pumpen miteinander verknüpft (d.h. wir erlauben keine Graphen mit Knoten an denen keine oder nur eine Kante anliegt). Wir unterscheiden außerdem zwei Arten von Kanten $E = E_R \cup E_P$, Rohre und Pumpen. Die Kanten sind gerichtet und es darf höchstens eine Kante zwischen zwei Knoten geben. Die Orientierung von Kanten ist wichtig um zu kodieren in welche Richtung das Wasser fließt (das Wasser kann aber in beide Richtungen fließen). Mit $n = \#V$ bezeichnen wir die Anzahl der Knoten in V und mit $m = \#E$

die Anzahl der Kanten, wobei $m = m_R + m_P$ die Anzahl Rohre und Anzahl Pumpen bezeichnet. In dem Beispiel oben gibt es fünf Knoten und acht Kanten. Die ersten sechs Kanten entsprechen Rohrleitungen und die Kanten e_7, e_8 entsprechen Pumpen.

Hagen-Poiseuille Gesetz Die Strömung in *einem* Rohr $e = (v, w) \in E_R$ wird in der Strömungsmechanik durch das Gesetz von Hagen-Poiseuille beschrieben:

$$q_e = \frac{\pi r_e^4}{8\eta l_e} \Delta p_e, \qquad L_e = \frac{\pi r_e^4}{8\eta l_e} \text{ “Leitfähigkeit”}, \tag{4.1}$$

mit den folgenden Größen:

q_e Gerichteter Volumenstrom in m^3/s. Das Vorzeichen von q_e kodiert die Richtung in die das Wasser fließt: $q_e > 0$ bedeutet, das Wasser fließt in Richtung der Orientierung der Kante e, $q_e < 0$ bedeutet das Wasser fließt entgegen der Orientierung der Kante e.

Δp_e Gerichtete Potentialdifferenz (oder auch manchmal auch Druckdifferenz) $\Delta p_e = p_v - p_w$ mit den Knotenpotentialen (oder Drücken) p_v und p_w in Pa. Δp_e hat das selbe Vorzeichen wie q_e, d.h. ein positiver Volumenstrom entspricht einem Fluss von hohem Potential zu niedrigem Potential.

r_e Radius des Rohres in m.

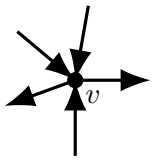
l_e Länge des Rohres in m.

η dynamische Viskosität von Wasser in $Pa\ s$.

Ziel des nun folgenden Verfahrens ist die Bestimmung aller Knotendrücke p_v sowie aller Kantenflüsse q_e aus den Daten r_e, l_e der einzelnen Rohre und den Stärken der Pumpen.

4.2 Kirchhoff'sche Gesetze

Knotenregel (1. Kirchhoff'sches Gesetz) Für jeden Knoten $v \in V$ definieren wir die ausgehenden Kanten E_v^+ sowie die eingehenden Kanten E_v^- als



$$E_v^+ = \{(u, w) \in E : u = v\}, \quad E_v^- = \{(u, w) \in E : w = v\}.$$

Dann gilt die Knotenregel

$$\sum_{e \in E_v^+} q_e - \sum_{e \in E_v^-} q_e = 0. \tag{4.2}$$

Die Knotenregel ist eine Folge der Massenerhaltung, denn in einem Knoten wird kein Wasser gespeichert. Alles Wasser das über ein Rohr in den Knoten hinein fließt muss ihn sofort über einen anderes Rohr wieder verlassen.

Die Gleichung (4.2) stellt eine lineare Beziehung zwischen den Kantenflüssen dar. Allerdings sind nur $n - 1$ dieser Beziehungen linear unabhängig. Denn

angenommen es gilt die Knotenregel für alle Knoten $v \in V \setminus \{w\}$ ausser im Knoten w, dann folgt

$$\begin{aligned} 0 &= \sum_{v \in V \setminus \{w\}} \left(\sum_{e \in E_v^+} q_e - \sum_{e \in E_v^-} q_e \right) = \sum_{v \in V \setminus \{w\}} \sum_{e \in E_v^+} q_e - \sum_{v \in V \setminus \{w\}} \sum_{e \in E_v^-} q_e \\ &= \sum_{e \in E_w^-} q_e - \sum_{e \in E_w^+} q_e, \end{aligned}$$

also die Knotenregel (mal -1) im Knoten w. Denn für jede Kante $e = (u, v)$ gilt genau einer der drei folgenden Fälle:

i) Es ist $u \neq w \wedge v \neq w$. Dann kommt q_e in jeder der beiden Summen am Ende der ersten Zeile genau einmal vor, einmal mit positivem und einmal mit negativem Vorzeichen. Somit kann man die beiden Flüsse weglassen.

ii) Es ist $u = w \wedge v \neq w$. Damit kommt q_e in den Summen genau einmal vor und zwar in E_v^+ als eingehende Kante, somit mit negativem Vorzeichen.

iii) Es ist $u \neq w \wedge v = w$. Damit kommt q_e in den Summen genau einmal vor und zwar in E_u^- als ausgehende Kante, somit mit positivem Vorzeichen.

Maschenregel (2. Kirchhoff'sches Gesetz) Es sei $C \subseteq E$ eine sogenannte Masche, d.h.

1) Die Kanten in C bilden einen geschlossenen Pfad.

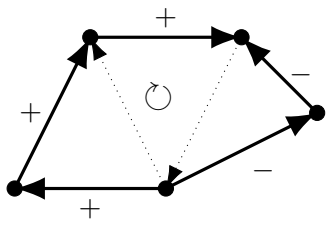
2) Dem Pfad C wird ein Umlaufsinn zugeordnet und wir setzen

$$\begin{aligned} C^+ &= \{e \in C \,:\, \text{e und C gleich orientiert}\}, \\ C^- &= \{e \in C \,:\, \text{e und C verschieden orientiert}\}. \end{aligned}$$

Dann gilt längs so einen Pfad die Maschenregel für die gerichteten Potentialdifferenzen:

$$\sum_{e \in C^+} \Delta p_e - \sum_{e \in C^-} \Delta p_e = 0. \tag{4.3}$$

Die Maschenregel ist eine Folge der Energieerhaltung. Die Größe $\int_v^w \Delta p_e dA$ ist eine Energie. Würde die Maschenregel nicht gelten wäre Energiegewinnung durch "im Kreis laufen" möglich.



Die Abbildung links zeigt eine Masche bestehend aus fünf Kanten. Davon haben drei dieselbe Orientierung wie die Masche und zwei die entgegengesetzte. Jede mögliche Masche in einem Graphen liefert somit eine lineare Gleichung der Form (4.3) für die Potentialdifferenzen. Allerdings zeigt sich, dass nur maximal $n-1$ dieser Gleichungen linear unabhängig sind. Somit sind die Maschen in der Praxis geeignet auszuwählen.

4.3 Knotenpotentialverfahren

Das Knotenpotentialverfahren nutzt das Gesetz von Hagen-Poiseuille und die beiden Kirchhoff'schen Gesetze um die Druckdifferenzen und Flüsse systematisch zu bestimmen.

Zunächst setzen wir, wie bereits oben bemerkt, für jede Kante $e = (v, w) \in E$

$$\Delta p_e = p_v - p_w, \qquad e \in E \tag{4.4}$$

dabei ist p_v das *Potential im Knoten* v.

Mit dieser Wahl ist die Maschenregel für jede Masche automatisch erfüllt. Denn sei $v \in V$ ein Knoten welcher in der Masche vorkommt, d.h. zwei Kanten der Masche liegen an v an. Sind beide Kanten mit der Masche orientiert so hebt sich p_v in der Gleichung auf. Analog gilt dies auch für alle anderen Fälle der Orientierung.

Da in der Maschenregel bzw. im Gesetz von Hagen-Poiseuille nur Potential*differenzen* auftauchen, kann das Potential in einem Knoten, dem sogenannten *Referenzknoten* beliebig festgelegt werden. D.h. wir wählen o.B.d.A. $r = v_n$ und setzen $p_{v_n} = 0$. Die Potentiale in den Knoten $v_1, \ldots, v_{n-1}$ sind die zu bestimmenden unbekannten Größen welche wir in einem Vektor $p \in \mathbb{R}^{n-1}$ zusammenfassen:

$$p = (p_{v_1}, \ldots, p_{v_{n-1}})^T. \tag{4.5}$$

Für alle Kanten $e = (v_i, v_j) \in E_R$, also der *Menge der Rohre* (!), können die Potentialdifferenzen ebenfalls in einem Vektor Δp zusammengefasst werden:

$$\Delta p = (\Delta p_{e_1}, \ldots, \Delta p_{e_{m_R}})^T, \tag{4.6}$$

wobei wir die Rohre von 1 bis m_R nummeriert haben.

Nun können die Gleichungen (4.4) in Matrixform zusammengefasst werden:

$$\Delta p = Bp \tag{4.7}$$

mit der Matrix $B \in \mathbb{R}^{m_R \times n-1}$. Jede Kante $e \in E_R$ liefert dabei eine Zeile von B, d.h. die Einträge von B werden wie folgt bestimmt:

$$(B)_{ij} = \begin{cases} +1 & e_i = (v, w) \in E_R \wedge v = v_j, \\ -1 & e_i = (v, w) \in E_R \wedge w = v_j, \\ 0 & \text{sonst.} \end{cases} \tag{4.8}$$

Beachte, dass $1 \leq j < n$ da der Referenzknoten v_n das Potential Null hat und nicht vorkommt. Die Matrix B ist dünn besetzt, da jede Zeile höchstens zwei von Null verschiedene Elemente hat.

Das Gesetz von Hagen-Poiseuille (4.1) kann ebenfalls in Matrixform formuliert werden:

$$q = L\Delta p \tag{4.9}$$

wobei die Einträge der Diagonalmatrix $L \in \mathbb{R}^{m_R \times m_R}$ die Leitfähigkeiten $(L)_{ii} = L_{e_i}$ sind und der Vektor

$$q = (q_{e_1}, \ldots, q_{e_{m_R}})^T \tag{4.10}$$

die gerichteten Kantenflüsse durch alle Rohre zusammenfasst.

Schließlich bleibt noch die Knotenregel zu berücksichtigen. Hier spielen die Pumpen nun eine zentrale Rolle. Wie oben bemerkt sind auch nur $n-1$ Knotenregeln linear unabhängig, d.h. wir lassen den Referenzknoten v_n unberücksichtigt. Für jedes $v \in V \backslash \{v_n\}$ spalten wir die Knotenregel in Rohre in Pumpen auf und erhalten

$$\sum_{e \in E_v^+} q_e - \sum_{e \in E_v^-} q_e = \sum_{e \in E_v^+ \cap E_R} q_e - \sum_{e \in E_v^- \cap E_R} q_e + \sum_{e \in E_v^+ \cap E_P} q_e - \sum_{e \in E_v^- \cap E_P} q_e.$$

Da die (gerichteten) Pumpenströme $q_e, e \in E_P$ gegeben sind können wir diese auf die rechte Seite schaffen und erhalten für jedes $v \in V \setminus \{v_n\}$:

$$\sum_{e \in E_v^+ \cap E_R} q_e - \sum_{e \in E_v^- \cap E_R} q_e = \sum_{e \in E_v^- \cap E_P} q_e - \sum_{e \in E_v^+ \cap E_P} q_e =: b_v. \tag{4.11}$$

Auch diese Gleichungen können in Matrixform geschrieben werden:

$$\tilde{B} q = b \tag{4.12}$$

mit dem Vektor

$$b = (b_{v_1}, \ldots, b_{v_{n-1}})^T \in \mathbb{R}^{n-1} \tag{4.13}$$

und der $n-1 \times m_R$-Matrix

$$(\tilde{B})_{ij} = \begin{cases} +1 & e_j \in E_{v_i}^+ \cap E_R, \\ -1 & e_j \in E_{v_i}^- \cap E_R, \\ 0 & \text{sonst.} \end{cases} \tag{4.14}$$

Ein Vergleich mit (4.8) zeigt, dass $\tilde{B} = B^T$!

Setzt man die Beziehungen (4.12), (4.9) und (4.7) ineinander ein, so erhält man

$$b = B^T q = B^T L \Delta p = B^T L B p,$$

also ein lineares Gleichungssystem

$$Ap = b \tag{4.15}$$

mit der $n-1 \times n-1$-Matrix $A = B^T L B$. Die Matrix A hat folgende interessante Eigenschaften:

1) A is symmetrisch, d.h. $(A)_{ij} = (A)_{ji}$. Dies folgt unmittelbar aus der Darstellung $A = B^T L B$ und der Tatsache dass L eine Diagonalmatrix ist.

2) A ist positiv definit, d.h $x^T A x > 0$ für alle $x \neq 0$. Dies folgt aus der linearen Unabhängigkeit der Spalten von B und der Tatsache dass $L_{ii} > 0$, denn sei $x \neq 0$ so gilt $y = Bx \neq 0$ genau wenn die Spalten von B linear unabhängig sind. Damit gilt $x^T A x = y^T L y = \sum_i L_{ii} y_i^2 > 0$.

3) A ist eine dünn besetzte Matrix, d.h. es sind nur sehr wenige Einträge, typischerweise $O(n)$ ungleich Null.

4.4 Zusammenfassung

In dieser Vorlesung haben wir ein Verfahren zur systematischen Analyse der Strömung in Rohrleitungsnetzwerken kennen gelernt. Die Analyse führt auf die Lösung linearer Gleichungssysteme mit möglicherweise sehr vielen Unbekannten.

Das Verfahren lässt sich übertragen auf andere Situationen, z.B. die Analyse elektrischer Netzwerke bestehend aus Widerständen und Stromquellen. Dann gelten folgende Entsprechungen:

q_e	elektrischer Strom durch eine Kante
p_v	elektrische Spannung am Knoten relativ zur Masse
Hagen-Poiseuille	Ohmsches Gesetz

Eine Erweiterung dieses Verfahrens auf Netzwerke aus Widerständen, Spulen und Kondensatoren (sogenannte RLC-Netzwerke) mit harmonischer (sinusförmiger) Anregung führt auf komplexwertige, lineare Gleichungssysteme.

Vorlesung 5

Einige Grundlagen aus der linearen Algebra

Bevor wir uns intensiv mit der numerischen Lösung linearer Gleichungssysteme beschäftigen wollen wir einige Grundlagen aus der Linearen Algebra und (weniger) der Analysis wiederholen. Dabei liegt der Fokus auf Konzepten die für die Numerik besonders wichtig sind, wie Normen, sowie die Verbindung von Konzepten aus Numerik und Analysis. Die folgenden Begriffe setzen wir als bekannt voraus. Rufen sie sich deren Bedeutung ins Gedächtnis wenn sie unsicher sind:

- Vektorraum über einem Körper $\mathbb{K}$, wobei hier nur $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$ betrachtet wird. Wir werden in dieser Vorlesung nur endlichdimensionale Vektorräume $V = \mathbb{K}^n$, $n \in \mathbb{N}$ betrachten.
- Linearkombination und lineare Unabhängigkeit.
- Basis und Dimension. Wir werden uns in dieser Vorlesung nur mit endlichdimensionalen Vektorräumen beschäftigen.
- Untervektorraum und Summe von Vektorräumen.
- Lineare Abbildung zwischen Vektorräumen (Homomorphismen).
- Bild und Kern einer linearen Abbildung.
- Die Begriffe injektiv, surjektiv und bijektiv.
- Matrizen als Darstellung linearer Abbildungen zwischen Vektorräumen.
- Rang, Determinante und Inverse einer Matrix.

Wir wollen in diesem Kapitel insbesondere Zusammenhänge zwischen linearer Algebra und Analysis herstellen und auf Aspekte eingehen die für die Numerik besonders relevant sind.

5.1 Normen für Vektoren

Ein zentrales Anliegen der numerischen Analysis ist die Herleitung von Fehlerabschätzungen. Hierfür benötigt man in der Regel Normen.

Definition 5.1. Sei V ein Vektorraum über $\mathbb{K}$. Eine Abbildung $\|.\| : V \to \mathbb{R}_0^+ = \{x \in \mathbb{R} : x \geq 0\}$ heißt Norm auf V, falls gilt:

$$\|x\| > 0 \qquad V \ni x \neq 0 \qquad \text{(Definitheit)}, \tag{5.1a}$$

$$\|\alpha x\| = |\alpha|\|x\| \qquad x \in V, \alpha \in \mathbb{K} \qquad \text{(positive Homogenität)}, \tag{5.1b}$$

$$\|x + y\| \leq \|x\| + \|y\| \qquad x, y \in V \qquad \text{(Dreiecksungleichung)}. \tag{5.1c}$$

Beispiel 5.2. Häufig verwendete Normen für $V = \mathbb{K}^n$ sind

$$\|x\|_2 = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{\frac{1}{2}} \qquad \text{Euklidsche Norm, } l_2\text{-Norm,} \tag{5.2a}$$

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| \qquad l_1\text{-Norm,} \tag{5.2b}$$

$$\|x\|_\infty = \max_{i=1,\ldots,n} |x_i| \qquad \text{Maximumnorm, } l_\infty\text{-Norm.} \tag{5.2c}$$

Mit Hilfe von Normen lässt sich der Begriff der Konvergenz aus der Analysis auf Folgen von Vektoren (bzw. Elementen eines Vektorraumes) übertragen. Eine Folge $\{x^{(k)}\}_{k=1}^{\infty}$ von Elementen eines Vektorraumes V konvergiert gegen $x \in V$, falls

$$\forall \epsilon > 0 \; \exists N \in \mathbb{N} \; \forall k \geq N \; : \; \|x - x^{(k)}\| < \epsilon,$$

d.h. die Folge der Zahlen $z^{(k)} = \|x - x^{(k)}\|$ konvergiert gegen Null. Diese Formulierung der Konvergenz setzt die Kenntnis des Grenzelementes x voraus.

Eine Folge $\{x^{(k)}\}_{k=1}^{\infty}$ von Elementen eines Vektorraumes V heisst *Cauchy-Folge* falls gilt

$$\forall \epsilon > 0 \; \exists N \in \mathbb{N} \; \forall k, l \geq N \; : \; \|x^{(k)} - x^{(l)}\| < \epsilon.$$

Ein Vektorraum V heisst *vollständig* wenn jede Cauchy-Folge gegen ein Element aus V konvergiert. □

Die Vollständigkeit des K^n ergibt sich aus der Vollständigkeit von $\mathbb{K}$.

Eine weitere wichtige Folgerung ist die

Lemma 5.3 (Inverse Dreiecksungleichung)**.** Für alle $x, y \in V$ gilt

$$\|x - y\| \geq |\|x\| - \|y\|| \,.$$

Beweis. Es gilt

$$\|x\| = \|x - y + y\| \leq \|x - y\| + \|y\| \;\Rightarrow\; \|x - y\| \geq \|x\| - \|y\|.$$

Analog zeigt man durch vertauschen von x und y:

$$\|y\| = \|y - x + x\| \leq \|y - x\| + \|x\| \;\Rightarrow\; \|x - y\| \geq \|y\| - \|x\| = -(\|x\| - \|y\|).$$

Aus beiden Abschätzung zusammen folgt die Behauptung. □

Aus der inversen Dreiecksungleichung folgert man die Stetigkeit der Norm,

$$x^{(k)} \to x \quad \Rightarrow \quad \|x^{(k)}\| \to \|x\|,$$

lese dazu die inverse Dreiecksungleichung von "rechts nach links".

Die Konvergenz von Folgen basiert auf dem Normbegriff. Nun gibt es ja diverse Normen und damit stellt sich die Frage ob man für unterschiedliche Normen einen unterschiedlichen Konvergenzbegriff erhält. Dass dies für den $\mathbb{K}^n$ *nicht* der Fall ist zeigt der folgende Satz.

Satz 5.4 (Äquivalenz aller Normen)**.** Auf $\mathbb{K}^n$ sind alle Normen in folgendem Sinne äquivalent. Zu je zwei Normen $\|.\|$ und $\|.\|'$ gibt es Zahlen $m, M > 0$ aus $\mathbb{R}$ so dass gilt

$$m\|x\|' \leq \|x\| \leq M\|x\|' \qquad \forall x \in \mathbb{K}^n.$$

Beweis. Es genügt die Äquivalenz aller Normen zur Maximumnorm $\|.\|_\infty$ zu zeigen. Wir definieren die Menge von Vektoren

$$S = \{x \in \mathbb{K}^n \,:\, \|x\|_\infty = 1\} \subset \mathbb{K}^n,$$

auch als "Einheitskugel bezüglich $\|.\|_\infty$" bezeichnet. Die Menge S ist beschränkt (die Norm ist ja gerade 1) und abgeschlossen (Grenzwerte von Folgen werden angenommen) im $\mathbb{K}^n$. Daher nimmt die stetige Funktion $\|.\| : S \to \mathbb{R}_0^+$ ihr Minimum und Maximum an, d.h. es existieren $x^{\min}, x^{\max} \in S$ so dass

$$0 < \|x^{\min}\| \leq \|x\| \leq \|x^{\max}\| \qquad \forall x \in S.$$

Beachte dass $x \in S \Rightarrow x \neq 0$. Für ein beliebiges $x \in \mathbb{K}^n$, $x \neq 0$ gilt nun $\|x\|_\infty^{-1} x \in S$ und somit

$$0 < \|x^{\min}\| \leq \|\|x\|_\infty^{-1} x\| = \|x\|_\infty^{-1}\|x\| \leq \|x^{\max}\| \qquad \forall x \in \mathbb{K}^n, x \neq 0.$$

Damit folgt

$$\|x^{\min}\|\|x\|_\infty \leq \|x\| \leq \|x^{\max}\|\|x\|_\infty \qquad \forall x \in \mathbb{K}^n, x \neq 0.$$

□

5.2 Normen für Matrizen

Wir betrachten nun Matrizen $A \in \mathbb{K}^{m\times n}$. Diese können zum einen als Schema von mn Zahlen interpretiert werden

$$A = \begin{bmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & & \vdots \\ a_{m,1} & \dots & a_{m,n} \end{bmatrix}, \qquad a_{i,j} \in \mathbb{K}$$

und zum anderen als lineare Abbildung zwischen Vektorräumen

$$F : \mathbb{K}^n \to \mathbb{K}^m, \qquad F(x) = Ax.$$

Die Menge der Matrizen mit der Addition von Matrizen und der skalaren Multiplikation stellt selbst wieder einen Vektorraum dar, der mit dem $\mathbb{K}^{mn}$ identifiziert werden kann. Insofern kann mittels jeder Norm auf dem Vektorraum $\mathbb{K}^{mn}$ eine Norm auf dem Vektorraum der Matrizen definiert werden.

Andererseits bringt die Sicht als lineare Abbildung den Definitionsraum $\mathbb{K}^n$ und den Bildraum $\mathbb{K}^m$ ins Spiel und die Hintereinanderausführung von Abbildungen kann als Multiplikation auf Vektorräumen von Matrizen interpretiert werden. Es ist daher wünschenswert, dass ausser den Normeigenschaften zusätzliche Eigenschaften im Zusammenspiel der verschiedenen Konzepte gelten.

Definition 5.5. *Für alle* $n, m \in \mathbb{N}$ sei $\|.\|$ eine Vecktornorm auf $\mathbb{K}^n$ und $\|.\|$ eine Matrixnorm auf $\mathbb{K}^{m\times n}$. Aufgrund des Argumentes ist jeweils klar welche Norm gemeint ist.

Die Matrixnorm heißt *verträglich* zur Vektornorm falls gilt

$$\|Ax\| \leq \|A\|\|x\| \qquad x \in \mathbb{K}^n, A \in \mathbb{K}^{m\times n} \tag{5.3}$$

und sie heißt *submultiplikativ* falls gilt

$$\|AB\| \leq \|A\|\|B\| \qquad A \in \mathbb{K}^{m\times n}, B \in \mathbb{K}^{n\times k}. \tag{5.4}$$

Manchmal wird Submultiplikativität auch als zwingend notwendige Bedingung für eine Matrixnorm gefordert. □

Beispiel 5.6. Betrachte die *Frobeniusnorm*

$$\|A\|_{\mathrm{Fr}} = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{\frac{1}{2}}.$$

Die Frobeniusnorm ist verträglich mit der Euklidschen Norm. □

Nun stellt sich die Frage wie man zu einer Norm auf dem $\mathbb{K}^n$ eine geeignete Norm auf $\mathbb{K}^{m\times n}$ findet die obige Bedingungen erfüllen. Antwort gibt

Definition 5.7 (Zugeordnete Matrixnorm)**.** Es sei $\|.\|$ eine Vektornorm auf $\mathbb{K}^n$ für alle $n \in \mathbb{N}$. Dann heißt

$$\|A\| = \sup_{x\in\mathbb{K}^n\setminus\{0\}} \frac{\|Ax\|}{\|x\|}$$

zugeordnete oder *natürliche* Matrixnorm. □

Die zugeordnete Matrixnorm ist stets verträglich und submultiplikativ.

Die Supremumsbildung lässt sich auch anders formulieren. wegen $\|A(cx)\| = |c|\|Ax\|$ und $\|cx\| = |c|\|x\|$ für jede Norm gilt

$$\sup_{x\in\mathbb{K}^n\setminus\{0\}} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\|=1} \|Ax\|.$$

Auskunft über spezielle zugeordnete Matrixnormen gibt das Lemma

Lemma 5.8. Die zugeordneten Matrixnormen zu $\|.\|_\infty$ und $\|.\|_1$ sind

$$\|A\|_\infty = \max_{1\le i\le m} \sum_{j=1}^{n} |a_{ij}| \qquad \text{(Zeilensummennorm)}, \tag{5.5a}$$

$$\|A\|_1 = \max_{1\le j\le n} \sum_{i=1}^{m} |a_{ij}| \qquad \text{(Spaltensummennorm)} \tag{5.5b}$$

Beweis. Es wird nur der Beweis für die (wichtigere) Zeilensummennorm gegeben.

a) Zunächst rechnen wir die Verträglichkeit nach

$$\begin{aligned}\|Ax\|_\infty = \max_{1\le i\le m} \left| \sum_{j=1}^{n} a_{ij}x_j \right| &\le \max_{1\le i\le m} \sum_{j=1}^{n} |a_{ij}||x_j| \\ &\le \max_{1\le i\le m} \sum_{j=1}^{n} |a_{ij}| \left(\max_{1\le k\le n} |x_k| \right) = \|A\|_\infty \|x\|_\infty.\end{aligned}$$

b) Damit gilt nun zunächst mal

$$\sup_{x\in\mathbb{K}^n\setminus\{0\}} \frac{\|Ax\|_\infty}{\|x\|_\infty} \le \sup_{x\in\mathbb{K}^n\setminus\{0\}} \frac{\|A\|_\infty\|x\|_\infty}{\|x\|_\infty} = \|A\|_\infty.$$

c) Für $A = 0$ gilt trivialerweise $\sup_{x\in\mathbb{K}^n\setminus\{0\}} \frac{\|Ax\|_\infty}{\|x\|_\infty} = \|A\|_\infty = 0$. Sei also $A \neq 0$ und mithin $\|A\|_\infty > 0$. Dann gibt es einen Zeilenindex $k \in \{1, \ldots, m\}$ so dass

$$\sum_{j=1}^{n} |a_{kj}| = \max_{1\leq i\leq m} \sum_{j=1}^{n} |a_{ij}| = \|A\|_\infty.$$

Nun definiere den Vektor z mit den Komponenten

$$z_j = \begin{cases} \frac{|a_{kj}|}{a_{kj}} & \text{falls } a_{kj} \neq 0, \\ 0 & \text{sonst.} \end{cases}$$

Da die Einträge des Vektors z nur die Werte $\{-1, 0, +1\}$ annehmen können und ein a_{kj} ungleich Null sein muss, gilt $\|z\|_\infty = 1$. Für $v = Az$ gilt dann für die k-te Komponente

$$v_k = (Az)_k = \sum_{j=1}^{n} a_{kj} z_j = \sum_{j=1}^{n} |a_{kj}| = \|A\|_\infty.$$

Damit gilt

$$\|A\|_\infty = v_k \leq \|v\|_\infty = \|Az\|_\infty \leq \sup_{x\in\mathbb{K}^n\setminus\{0\}} \frac{\|Ax\|_\infty}{\|x\|_\infty}$$

und somit zusammen mit b) die Gleichheit.

□

5.3 Eigenwerte und Eigenvektoren

Definition 5.9. Sei $A \in \mathbb{K}^{n\times n}$ mit $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$. Die komplexe Zahl $\lambda \in \mathbb{C}$ heißt *Eigenwert* wenn es einen Vektor $e \in \mathbb{K}^n$, $e \neq 0$, gibt für den gilt

$$Ae = \lambda e.$$

e heißt *Eigenvektor* zum Eigenwert λ und (λ, e) heißt auch *Eigenpaar*. □

Zu jedem Eigenwert gibt es mindestens einen Eigenvektor. Mit jedem Eigenvektor e ist auch ce ein Eigenvektor für jedes $c \in \mathbb{K}$, die Eigenvektoren zu einem Eigenwert spannen deshalb einen Unterraum des $\mathbb{K}^n$ auf.

Definition 5.10 (Inverse)**.** Zu $A \in \mathbb{K}^{n\times n}$ nennt man $A^{-1} \in \mathbb{K}^{n\times n}$ *inverse Matrix* wenn gilt

$$AA^{-1} = I$$

mit I der Einheitsmatrix. Die Existenz der Inversen hängt eng mit der Lösbarkeit von linearen Gleichungssystemen zusammen, der wir uns weiter unten widmen. □

Definition 5.11 (Diagonalisierbarkeit)**.** Gibt es eine Basis bestehend aus Eigenvektoren nennt man A *diagonalisierbar*. Dann existiert eine Matrix $S \in \mathbb{K}^{n\times n}$ so dass

$$S^{-1}AS = \operatorname{diag}(\lambda_1, \ldots, \lambda_n) = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}.$$

Denn man setzt einfach $S = [e_1, \ldots, e_n]$ (spaltenweiser Aufbau der Matrix S aus den Vektoren e_i), dann gilt $AS = [\lambda_1 e_1, \ldots, \lambda_n e_n] = S\,\mathrm{diag}(\lambda_1, \ldots, \lambda_n)$. Das Produkt $S^{-1}AS$ heißt auch *Ähnlichkeitstransformation.* □

Die Eigenwerte sind Nullstellen des *charakteristischen Polynoms*

$$p(\lambda) = \det(A - \lambda I) = 0.$$

Nach dem Fundamentalsatz der Algebra hat jedes Polynom vom Grad n genau n Nullstellen in $\mathbb{C}$, wenn man die Vielfachheiten mitzählt. Bei reellwertigen Matrizen treten komplexe Eigenwerte stets in konjugiert komplexen Paaren auf, d.h. mit $a + ib$ ist auch $a - ib$ ein Eigenwert. In der Praxis bestimmt man die Eigenwerte jedoch nicht als Nullstellen des charakteristischen Polynoms da es hierfür keine stabilen numerischen Algorithmen gibt.

Eigenwerte sind in der Regel schwierig zu berechnen. Folgendes Lemma gibt eine einfache Abschätzung.

Lemma 5.12. Für jede zu einer Vektornorm $\|.\|$ verträglichen Matrixnorm gilt

$$\|A\| \geq \varrho(A) = \max_{1 \leq i \leq n} |\lambda_i|.$$

$\varrho(A)$ heißt *Spektralradius.*

Beweis. Zu $\lambda \in \mathbb{C}$ wähle einen Eigenvektor e mit $\|e\| = 1$. Dies ist immer möglich. Es gilt dann

$$|\lambda| = |\lambda|\|e\| = \|\lambda e\| = \|Ae\| \leq \|A\|\|e\| = \|A\|.$$

Somit gilt $\|A\| \geq |\lambda|$ für jeden Eigenwert und damit auch das Maximum. □

Einige Klassen von Matrizen sind für die Praxis besonders relevant.

Definition 5.13. Zu $A \in \mathbb{K}^{m \times n}$ heißt A^T *transponierte* Matrix wenn gilt $(A^T)_{ij} = a_{ji}$. Zu $A \in \mathbb{K}^{m \times n}$ heißt $\bar{A}$ *konjugiert komplexe* Matrix wenn gilt $(\bar{A})_{ij} = \bar{a}_{ij}$.

a) $A \in \mathbb{K}^{n \times n}$ heißt *hermitesch* falls $A = \bar{A}^T$ gilt. Manche Autoren kürzen $A^H = \bar{A}^T$ ab. Reelle hermitesche Matrizen heißen symmetrisch.

b) Eine komplexwertige Matrix $A \in \mathbb{C}^{n \times n}$ mit

$$A\bar{A}^T = \bar{A}^T A \qquad \text{heißt } \textit{normal}\text{, eine mit} \tag{5.6}$$

$$A^{-1} = \bar{A}^T \qquad \text{heißt } \textit{unitär}. \tag{5.7}$$

c) Eine reellwertige Matrix $A \in \mathbb{R}^{n \times n}$ mit

$$A^{-1} = A^T \qquad \text{heißt } \textit{orthogonal}. \tag{5.8}$$

Satz 5.14 (Diagonalisierbarkeit hermitescher Matrizen)**.** Eine Matrix $A \in \mathbb{C}^n$ ist genau dann *reell* diagonalisierbar, wenn A hermitesch ist. Die Transformationsmatrix Q ist in diesem Fall sogar unitär, d.h. es gilt $\bar{Q}^T A Q = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ mit $\lambda_i \in \mathbb{R}$.

Beweis. [Hackbusch, 1991, Satz 2.8.7] □

Aus diesem Satz folgern wir, dass reelle symmetrische Matrizen stets mit einer orthogonalen Transformationsmatrix reell diagonalisierbar sind.

5.4 Skalarprodukt

Definition 5.15. Sei $\mathbb{K} = \mathbb{C}$ oder $\mathbb{K} = \mathbb{R}$. Eine Abbildung $(.,.) : \mathbb{K}^n \times \mathbb{K}^n \to \mathbb{K}$ heißt *Skalarprodukt* falls sie folgende Eigenschaften besitzt:

$$(x, y) = \overline{(y, x)} \qquad x, y, \in \mathbb{K}^n \qquad \text{(Symmetrie)}, \tag{5.9a}$$

$$(\alpha x + \beta y, z) = \alpha(x, z) + \beta(y, z) \qquad x, y, z \in \mathbb{K}^n, \quad \alpha, \beta \in \mathbb{K} \qquad \text{(Linearität)}, \tag{5.9b}$$

$$(x, x) > 0 \qquad x \in \mathbb{K}^n \setminus \{0\} \qquad \text{(Definitheit)}. \tag{5.9c}$$

Bemerkung 5.16. Man beachte:

a) Aus der Symmetrie folgt wegen $(x, x) = \overline{(x, x)}$ dass (x, x) stets reell ist. Somit macht es Sinn die Definitheit zu fordern (auf $\mathbb{C}$ ist kein $>$ definiert).

b) Die Linearität im zweiten Argument gilt ebenfalls aber in der folgenden Form

$$\begin{aligned}(z, \alpha x + \beta y) &= \overline{(\alpha x + \beta y, z)} = \overline{\alpha(x, z) + \beta(y, z)} = \bar{\alpha}\overline{(x, z)} + \bar{\beta}\overline{(y, z)} \\ &= \bar{\alpha}(z, x) + \bar{\beta}(z, y).\end{aligned}$$

Es werden also die skalaren Faktoren zusätzlich konjugiert. □

Wegen der Definitheit kann man zu jedem Skalarprodukt in kanonischer Weise eine Norm konstruieren, die sogenannte *induzierte Norm*. Man rechnet nach, dass

$$\|x\| = \sqrt{(x, x)} \tag{5.10}$$

alle Eigenschaften einer Norm erfüllt. Für Skalarprodukt und induzierte Norm gilt stets die *Ungleichung von Cauchy-Schwarz*:

$$|(x, y)| \leq \|x\|\|y\|. \tag{5.11}$$

Diese Ungleichung ist neben der Dreiecksungleichung wohl die am häufigsten verwendete Ungleichung der numerischen Analysis.

Definition 5.17 (Euklidsches Skalarprodukt)**.** Das *Euklidsche Skalarprodukt* in $\mathbb{K}^n$ lautet

$$(x, y)_2 = \sum_{i=1}^{n} x_i \bar{y}_i. \tag{5.12}$$

□

Damit gilt für $A \in \mathbb{K}^n$

$$(Ax, y)_2 = (x, \bar{A}^T y)_2 \quad \forall x, y \in \mathbb{K}^n \tag{5.13}$$

denn

$$\begin{aligned}(Ax, y)_2 &= \sum_{i=1}^{n} \left(\sum_{j=1}^{n} a_{ij} x_j \right) \bar{y}_i = \sum_{i=1}^{n} \sum_{j=1}^{n} x_j \overline{\bar{a}_{ij} y_i} \\ &= \sum_{j=1}^{n} x_j \overline{\left(\sum_{i=1}^{n} \bar{a}_{ij} y_i \right)} = (x, \bar{A}^T y)_2.\end{aligned}$$

Damit kann man hermitesche Matrizen auch charakterisieren als

$$A = \bar{A}^T \quad \Leftrightarrow \quad (Ax, y)_2 = (x, Ay)_2 \quad \forall x, y \in \mathbb{K}^n. \tag{5.14}$$

Denn die Richtung $\Rightarrow$ folgt aus $(Ax, y)_2 = (x, \bar{A}^T y)_2 = (x, Ay)_2$ und die Richtung $\Leftarrow$ folgt wenn man die Koordinateneinheitsvektoren (nicht Eigenvektoren) $x = e_i$ und $y = e_j$ jeweils einsetzt.

Im allgemeinen heißt eine Matrix A^* *adjungiert* zu A bezüglich eines Skalarproduktes $(.,.)$ wenn

$$(Ax, y) = (x, A^* y) \; \forall x, y \in \mathbb{K}^n.$$

Gilt $A = A^*$ heißt A *selbstadjungiert* bezüglich $(.,.)$. Somit nennt man hermitesche Matrizen auch selbstadjungiert bezüglich des Euklidschen Skalarproduktes.

5.5 Spektralnorm

Oben haben wir in Lemma 5.8 zugeordnete Matrixnormen für die Maximumnorm und die 1-Norm charakterisieren können. Die der Euklischen Norm zugeordnete Matrixnorm

$$\|A\|_2 = \sup_{x \in \mathbb{K}^n \setminus \{0\}} \frac{\|Ax\|_2}{\|x\|_2},$$

die sogenannte *Spektralnorm*, wurde bisher noch nicht untersucht. Das folgende Lemma gibt hierzu Auskunft.

Lemma 5.18. Für hermitesche Matrizen $A \in \mathbb{K}^n$ gilt

$$\|A\|_2 = \varrho(A) = \max_{1 \le i \le n} |\lambda_i|,$$

d.h. in diesem Fall stimmt die Spektralnorm mit dem Spektralradius überein. Für jede Matrix $A \in \mathbb{K}^n$ gilt

$$\|A\|_2 = \sqrt{\varrho\left(\bar{A}^T A\right)}.$$

Beweis. a) Im ersten Fall ist A hermitesch, hat also nach Satz 5.14 n reelle Eigenwerte λ_i und einen vollständigen Satz orthonormaler Eigenvektoren e_i. Damit lässt sich jedes $x \in \mathbb{K}^n$ in der Eigenvektorbasis darstellen als $x = \sum_{i=1}^n \alpha_i e_i$ und es gilt $(e_i, e_j) = \delta_{ij}$ ($\delta_{ij} = 1$ wenn $i = j$, sonst 0). Dann gilt

$$\begin{aligned} \|x\|_2^2 = (x, x)_2 &= \left(\sum_{i=1}^n \alpha_i e_i, \sum_{j=1}^n \alpha_j e_j \right)_2 = \sum_{i=1}^n \sum_{j=1}^n \alpha_i \bar{\alpha}_j (e_i, e_j)_2 \\ &= \sum_{i=1}^n |\alpha|^2, \end{aligned}$$

$$\begin{aligned}\|Ax\|_2^2 = (Ax, Ax)_2 &= \left(\sum_{i=1}^n \alpha_i Ae_i, \sum_{j=1}^n \alpha_j Ae_j\right)_2 \\ &= \left(\sum_{i=1}^n \alpha_i \lambda_i e_i, \sum_{j=1}^n \alpha_j \lambda_j e_j\right)_2 = \sum_{i=1}^n \sum_{j=1}^n \alpha_i \lambda_i \bar{\alpha}_j \bar{\lambda}_j (e_i, e_j)_2 \\ &= \sum_{i=1}^n \lambda_i^2 |\alpha|^2.\end{aligned}$$

Somit

$$\|A\|_2^2 = \sup_{x \in \mathbb{K}^n \setminus \{0\}} \frac{\|Ax\|_2^2}{\|x\|_2^2} = \sup_{x \in \mathbb{K}^n \setminus \{0\}} \frac{\sum_{i=1}^n \lambda_i^2 |\alpha|^2}{\sum_{i=1}^n |\alpha|^2} \leq (\varrho(A))^2$$

also $\|A\|_2 \leq \varrho(A)$. Andererseits gilt wegen Lemma 5.12 für jede Matrixnorm $\|A\| \geq \varrho(A)$ und somit muss $\|A\|_2 = \varrho(A)$ gelten.

b) Zweiter Fall. Sei nun $A \in \mathbb{K}^n$ beliebig. Dann gilt unter Nutzung von Gleichung (5.13)

$$\|Ax\|_2^2 = (Ax, Ax)_2 = (\bar{A}^T Ax, x)_2.$$

Die Matrix $\bar{A}^T A$ ist hermitesch mit betragsgrößtem Eigenwert $\varrho(\bar{A}^T A)$. Also

$$\|A\|_2^2 = \sup_{x \in \mathbb{K}^n \setminus \{0\}} \frac{\|Ax\|_2^2}{\|x\|_2^2} = \sup_{x \in \mathbb{K}^n \setminus \{0\}} \frac{(\bar{A}^T Ax, x)_2}{\|x\|_2} = \varrho(\bar{A}^T A),$$

woraus die Behauptung folgt.

□

5.6 Positiv Definite Matrizen

Eine wichtige Klasse von Matrizen mit vorteilhaften Eigenschaften sind die positiv definiten Matrizen. Diese sind bereits in Kapitel 4 vorgekommen und sind wie folgt definiert.

Definition 5.19. Eine Matrix $A \in \mathbb{K}^{n \times n}$ heißt *positiv definit* wenn die beiden folgenden Bedingungen erfüllt sind:

$$(Ax, x)_2 \in \mathbb{R} \qquad \forall x \in \mathbb{K}^n, \tag{5.15a}$$

$$(Ax, x)_2 > 0 \qquad \forall x \in \mathbb{K}^n \setminus \{0\}. \tag{5.15b}$$

Im Fall $\mathbb{K} = \mathbb{R}$ ist die erste Bedingung trivialerweise immer erfüllt und nur die zweite Bedingung ist relevant. Im Fall $\mathbb{K} = \mathbb{C}$ ist die erste Bedingung wichtig damit die zweite Bedingung überhaupt Sinn macht. □

Damit stellt sich die Frage für welche Matrizen die Bedingung (5.15a) überhaupt Sinn macht. Antwort gibt

Lemma 5.20. Für $A \in \mathbb{C}^{n \times n}$ gilt: A ist hermitesch genau dann wenn $(Ax, x)_2 \in \mathbb{R}$ $\forall x \in \mathbb{C}^n$.

Beweis. Richtung $\Rightarrow$. $A \in \mathbb{C}^{n\times n}$ sei hermitesch. Dann gilt wegen (5.14) $(Ax,x)_2 = (x,Ax)_2$. Andererseits folgt aus der Symmetrie des Skalarproduktes $(Ax,x)_2 = \overline{(x,Ax)_2}$. Mithin gilt $(x,Ax)_2 = \overline{(x,Ax)_2}$, also $(Ax,x) \in \mathbb{R}$.
Für die Richtung $\Leftarrow$ muss man etwas mehr aufwenden. Man rechnet

$$(A(x+y),x+y)_2 = (Ax,x)_2 + (Ax,y)_2 + (Ay,x)_2 + (Ay,y)_2.$$

Umstellen liefert

$$(Ax,y)_2 + (Ay,x)_2 = (A(x+y),x+y)_2 - (Ax,x)_2 - (Ay,y)_2 \in \mathbb{R}$$

und deswegen muss $\mathrm{Im}(Ax,y)_2 = -\mathrm{Im}(Ay,x)_2$ sein. Nun rechne

$$(A(ix+y),ix+y)_2 = (A(ix),ix)_2 + (A(ix),y)_2 + (Ay,ix)_2 + (Ay,y)_2$$

und stelle um zu

$$\begin{aligned}(A(ix),y)_2 + (Ay,ix)_2 &= i(Ax,y)_2 - i(Ay,x)_2 = i\left((Ax,y)_2 - (Ay,x)_2\right)\\ &= (A(ix+y),ix+y)_2 - (A(ix),ix)_2 - (Ay,y)_2 \in \mathbb{R}.\end{aligned}$$

Da $i\left((Ax,y)_2 - (Ay,x)_2\right) \in \mathbb{R}$ muss $\mathrm{Re}((Ax,y)_2 - (Ay,x)_2) = 0$ sein also $\mathrm{Re}(Ax,y)_2 = \mathrm{Re}(Ay,x)_2$. Mit beiden Argumenten zusammen folgt

$$\begin{aligned}(Ax,y)_2 &= \mathrm{Re}(Ax,y)_2 + i\mathrm{Im}(Ax,y)\\ &= \mathrm{Re}(Ay,x)_2 - i\mathrm{Im}(Ay,x)_2\\ &= \overline{(Ay,x)_2}\\ &= (x,Ay)_2.\end{aligned}$$

□

Im Komplexen macht also die Bedingung $(Ax,x) > 0$ nur für hermitesche Matrizen Sinn. Im Reellen hingegen macht diese Bedingung für alle Matrizen Sinn und die Bedingung der Symmetrie stellt eine zusätzliche, unabhängige Bedingung dar.

Eine Charakterisierung der positiv definiten Matrizen über Eigenwerte liefert das folgende Lemma.

Lemma 5.21. Eine hermitesche Matrix A (im Reellen: symmetrische Matrix) ist genau dann positiv definit, wenn alle ihre (reellen) Eigenwerte positiv sind. Alle Hauptdiagonalelemente von A sind stets reell und positiv.

Beweis. a) A sei hermitesch mit nur positiven Eigenwerten $\lambda_i > 0$. Dann hat $x \in \mathbb{K}^n$ die Darstellung $x = \sum_{i=1}^n \alpha_i e_i$ mit orthonormalen Eigenvektoren e_i und es gilt

$$(Ax,x)_2 = \left(\sum_{i=1}^n \alpha_i Ae_i, \sum_{i=1}^n \alpha_i e_i\right)_2 = \sum_{i=1}^n \lambda_i |\alpha_i|^2 > 0.$$

b) A sei hermitesch und positiv definit. Dann gilt für *jeden* Eigenvektor e_i zum Eigenwert λ_i:

$$0 < (Ae_i,e_i)_2 = (\lambda_i e_i,e_i)_2 = \lambda_i (e_i,e_i)_2.$$

Da $(e_i,e_i)_2 > 0$ ist $\lambda_i > 0$.

c) Sei z_i der i-te Koordinateneinheitsvektor, dann gilt

$$0 < (Az_i, z_i)_2 = a_{ii} \in \mathbb{R}.$$

□

Speziell für reellwertige Matrizen gilt

Lemma 5.22. Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch und positiv definit. Dann liegt das betragsmäßig größte Element stets auf der Hauptdiagonale der Matrix.

Beweis. Sei z_k der k-te Koordinateneinheitsvektor. Angenommen a_{ij}, $i \neq j$ sei das betragsmäßig größte Element (also außerhalb der Hauptdiagonale). Wir rechnen

$$\begin{aligned} 0 &< (A(z_i - \operatorname{sgn}(a_{ij})z_j), z_i - \operatorname{sgn}(a_{ij})z_j)_2 \\ &= (Az_i, z_i)_2 - 2\operatorname{sgn}(a_{ij})(Az_i, z_j)_2 + \operatorname{sgn}(a_{ij})^2(Az_j, z_j)_2 \\ &= a_{ii} - 2\operatorname{sgn}(a_{ij})a_{ij} + \operatorname{sgn}(a_{ij})^2 a_{jj} \\ &= a_{ii} - 2|a_{ij}| + a_{jj} \end{aligned}$$

also $|a_{ij}| < \frac{1}{2}(a_{ii} + a_{jj}) = \frac{1}{2}(|a_{ii}| + |a_{jj}|)$ da Hauptdiagonalelemente stets positiv sind. Dies ist ein Widerspruch, da a_{ij} als betragsmäßig größtes Element angenommen war. □

Vorlesung 6

Konditionsanalyse für lineare Gleichungssysteme

Wir wenden uns nun der Lösung linearer Gleichungssysteme zu. Zunächst untersuchen wir in dieser Vorlesung die Konditionierung dieser Aufgabe.

6.1 Notation und Lösbarkeit linearer Gleichungsystem

Gegeben seien eine Matrix $A \in \mathbb{K}^{m\times n}$ und ein Vektor $b \in \mathbb{K}^m$

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}, \qquad b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}. \tag{6.1}$$

Elemente einer Matrix A werden mit a_{ij} oder $(A)_{ij}$ bezeichnet, entsprechend sind die Komponenten eines Vektors b dann b_i oder $(b)_i$. Die Schreibweise mit den runden Klammern bietet sich an wenn die Matrix oder der Vektor selbst schon einen Index tragen. Als Körper lassen wir $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$ zu.

In einem *linearen Gleichungssystem* ist ein Vektor $x \in \mathbb{K}^n$, $x = (x_1, \dots, x_n)^T$, gesucht, welcher

$$Ax = b \tag{6.2}$$

erfüllt. Das Gleichungssystem heißt

i) *unterbestimmt*, falls $m < n$,

ii) *quadratisch*, falls $m = n$ und

iii) *überbestimmt*, falls $m > n$.

Das lineare Gleichungssystem hat mindestens eine Lösung falls

$$\mathrm{Rang}(A) = \mathrm{Rang}([A|b]) = \mathrm{Rang}\left(\begin{bmatrix} a_{11} & \dots & a_{1n} & b_1 \\ \vdots & & & \vdots \\ a_{m1} & \dots & a_{mn} & b_m \end{bmatrix}\right)$$

d.h. b liegt im Bild der durch A gegebenen linearen Abbildung.

Für *quadratische* Matrizen (und mit diesen werden wir uns zunächst beschäftigen) sind folgende Aussagen äquivalent:

i) $Ax = b$ ist für jedes b eindeutig lösbar.

ii) $\mathrm{Rang}(A) = n$.

iii) $\det(A) \neq 0$.

iv) A hat keinen Eigenwert 0.

6.2 Kondition einer Matrix

In der Konditionsanalyse untersuchen wir die Empfindlichkeit von Abbildungen F bezüglich der Eingaben. Im Fall eines linearen Gleichungssystems $Ax = b$ sind A, b die Eingaben und x ist die Ausgabe. Es gilt also

$$x = F(A, b) = A^{-1}b.$$

Betrachten wir zunächst nur b als Eingabe, d.h. A sei nun fest gewählt. Wir untersuchen die Auswirkungen einer Änderung von b zu $b + \Delta b$. Dann ändert sich entsprechend x zu $x + \Delta x$ und es gilt

$$A(x + \Delta x) = b + \Delta b \qquad \Leftrightarrow \qquad x + \Delta x = A^{-1}(b + \Delta b).$$

Die absolute Änderung im Ergebnis ist somit

$$x + \Delta x - x = \Delta x = A^{-1}(b + \Delta b) - A^{-1}b = A^{-1}\Delta b.$$

Zur Untersuchung der *relativen* Kondition gehen wir zu Normen über:

$$\begin{aligned}\frac{\|\Delta x\|}{\|x\|} &= \frac{\|A^{-1}\Delta b\|}{\|A^{-1}b\|} \\ &\leq \frac{\|A^{-1}\|\|\Delta b\|}{\|A^{-1}b\|} = \frac{\|A^{-1}\|\|b\|}{\|A^{-1}b\|}\frac{\|\Delta b\|}{\|b\|} = \frac{\|A^{-1}\|\|AA^{-1}b\|}{\|A^{-1}b\|}\frac{\|\Delta b\|}{\|b\|} \\ &\leq \|A\|\|A^{-1}\|\frac{\|\Delta b\|}{\|b\|}.\end{aligned}$$

Dabei haben wir zwei mal ausgenutzt, dass die Normen auf Matrizen und Vektoren verträglich sind.

Wir können die Größe $\|A\|\|A^{-1}\|$ als Verstärkungsfaktor für die relativen Änderungen in der Eingabe b interpretieren. Die relative Kondition ist bei diesem Ansatz durch die Verwendung der Norm in einer einzigen Zahl zusammengefasst. Wir können also nicht mehr unterscheiden welche Komponente der Eingabe sich wie auf welche Komponente der Ausgabe auswirkt. Die Analyse ist in diesem Sinn ungenau und kann zu pessimistischen Ergebnissen führen („worst-case Analyse“).

Definition 6.1 (Konditionszahl)**.** Für jede invertierbare Matrix A heißt die Zahl

$$\mathrm{cond}(A) = \|A\|\|A^{-1}\|$$

Konditionszahl von A. Dabei kann $\|.\|$ irgendeine verträgliche Norm auf dem Vektorraum der Matrizen sein. □

Bemerkung 6.2 (Spektralkondition)**.** Für $\|.\|_2$ heißt

$$\mathrm{cond}_2(A) = \|A\|_2\|A^{-1}\|_2$$

Spektralkondition von A. Ist A symmetrisch und positiv definit dann gilt

$$\mathrm{cond}_2(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}.$$

Denn dann gilt nach Lemma 5.21 für die Eigenwerte $0 < \lambda_{\min}(A) \leq \lambda \leq \lambda_{\max}(A)$ und damit wegen Lemma 5.18 $\|A\|_2 = \lambda_{\max}$. Ist λ Eigenwerte von A so ist λ^{-1} ein Eigenwerte der Inversen A^{-1}. Daher ist der kleinste Eigenwert der Inversen $\lambda_{\max}^{-1}$, der größte Eigenwert $\lambda_{\min}^{-1}$ und $\|A^{-1}\|_2 = \lambda_{\min}^{-1}$. □

Dass die Konditionszahl nicht unbedingt etwas mit der Determinante zu tun hat zeigt folgendes

Beispiel 6.3. a) Betrachte die 2×2 Matrix für $0 < \epsilon < 1$

$$A = \begin{bmatrix} -1 & 1 \\ 1-\epsilon & -1 \end{bmatrix}, \qquad A^{-1} = \frac{1}{\epsilon} \begin{bmatrix} -1 & -1 \\ \epsilon - 1 & -1 \end{bmatrix}.$$

Es gilt $\det(A) = \epsilon$ und die Matrix wird für $\epsilon \to 0$ singulär. Für die Konditionszahl mit der Zeilensummennorm gilt

$$\|A\|_\infty = \max(2, 2-\epsilon) = 2, \quad \|A^{-1}\|_\infty = \frac{2}{\epsilon}, \quad \operatorname{cond}_\infty(A) = \frac{4}{\epsilon}.$$

Man könnte also vermuten dass eine kleine Determinante zu einer großen Kondition führt.

b) Dies ist jedoch nicht immer der Fall. Betrachte

$$B = \begin{bmatrix} 10^{-10} & 0 \\ 0 & 10^{-10} \end{bmatrix}, \qquad B^{-1} = \begin{bmatrix} 10^{10} & 0 \\ 0 & 10^{10} \end{bmatrix}.$$

Es ist $\det(B) = 10^{-20}$ aber für die Konditionszahl gilt

$$\operatorname{cond}_\infty(B) = \|B\|_\infty \|B^{-1}\|_\infty = 10^{-10} \cdot 10^{10} = 1.$$

c) Schließlich rechnet man nach:

$$\operatorname{cond}(\epsilon A) = \operatorname{cond}(A), \quad \text{aber} \quad \det(\epsilon A) = \epsilon^n \det(A).$$

Beide Größen skalieren also völlig unterschiedlich und haben in der Regel nichts miteinander zu tun.

6.3 Störungssatz

Wir wollen die Analyse von oben nun zusätzlich auf Änderungen in der Matrix A erweitern, d.h. es wird nun auch A zu $A + \Delta A$ geändert. Dann ändert sich entsprechend x zu $x + \Delta x$ und es gilt

$$(A + \Delta A)(x + \Delta x) = b + \Delta b \qquad \Leftrightarrow \qquad x + \Delta x = (A + \Delta A)^{-1}(b + \Delta b).$$

Auch wenn A als invertierbar angenommen wird, ist $A + \Delta A$ nicht notwendigerweise invertierbar. Wir untersuchen zunächst wann dies der Fall ist.

Lemma 6.4. $B \in \mathbb{K}^{n \times n}$ habe Norm $\|B\| < 1$. Dann ist $I + B$ regulär und es gilt

$$\left\|(I+B)^{-1}\right\| \leq \frac{1}{1 - \|B\|}. \tag{6.3}$$

Beweis. a) Für alle $x \in \mathbb{K}^n$ gilt $\|x\| = \|x + Bx - Bx\| \leq \|(I+B)x\| + \|Bx\|$ und somit

$$\|(I+B)x\| \geq \|x\| - \|Bx\| \geq \|x\| - \|B\|\|x\| = (1 - \|B\|)\|x\|.$$

Wegen $\|B\| < 1$ gilt $1 - \|B\| > 0$ und für jedes $x \neq 0$ muss $(I+B)x \neq 0$ gelten, somit ist $I + B$ regulär.

b) Wie oben zeigt man für beliebige A, B, $\|A\| \leq \|A+B\|+\|B\| \Rightarrow \|A+B\| \geq \|A\| - \|B\|$ (inverse Dreiecksungleichung). Somit

$$\begin{aligned} 1 = \|I\| &= \|(I+B)(I+B)^{-1}\| = \|(I+B)^{-1} + B(I+B)^{-1}\| \\ &\geq \|(I+B)^{-1}\| - \|B(I+B)^{-1}\| \geq \|(I+B)^{-1}\| - \|B\|\|(I+B)^{-1}\| \\ &= (1-\|B\|)\|(I+B)^{-1}\|. \end{aligned}$$

woraus die Abschätzung folgt.

□

Damit können wir den folgenden Satz formulieren

Satz 6.5 (Störungssatz)**.** Sei $A \in \mathbb{K}^{n\times n}$ regulär und $\|\Delta A\| < 1/\|A^{-1}\|$. Dann ist $A + \Delta A$ ebenfalls regulär und es gilt für die Lösung des gestörten Systems $(A + \Delta A)(x + \Delta x) = b + \Delta b$:

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\operatorname{cond}(A)}{1 - \operatorname{cond}(A)\frac{\|\Delta A\|}{\|A\|}} \left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|} \right). \tag{6.4}$$

Beweis. a) Zunächst zeigen wir dass $A+\Delta A$ regulär. $A+\Delta A = A(I+A^{-1}\Delta A)$ ist regulär wenn A und $I + A^{-1}\Delta A$ regulär sind. A ist regulär nach Voraussetzung und wegen $\|A^{-1}\Delta A\| \leq \|A^{-1}\|\|\Delta A\| < 1$ ist nach Lemma 6.4 auch $I + A^{-1}\Delta A$ regulär.

b) Wir rechnen zunächst nach

$$\begin{aligned} & & (A+\Delta A)(x+\Delta x) &= b + \Delta b \\ \Leftrightarrow \quad & Ax + (\Delta A)x + (A+\Delta A)\Delta x &= b + \Delta b \\ \Leftrightarrow \quad & (\Delta A)x + (A+\Delta A)\Delta x &= \Delta b \\ \Leftrightarrow \quad & \Delta x &= (A+\Delta A)^{-1}(\Delta b - (\Delta A)x). \end{aligned}$$

Nun gehen wir zu Normen über

$$\begin{aligned} \|\Delta x\| &= \|(A+\Delta A)^{-1}(\Delta b - (\Delta A)x)\| \\ &\leq \|(A+\Delta A)^{-1}\| \, (\|\Delta b\| + \|\Delta A\|\|x\|) && \text{Dreieck, verträglich} \\ &= \|[A(I+A^{-1}\Delta A)]^{-1}\| \, (\|\Delta b\| + \|\Delta A\|\|x\|) \\ &= \|(I+A^{-1}\Delta A)^{-1}A^{-1}\| \, (\|\Delta b\| + \|\Delta A\|\|x\|) \\ &\leq \|(I+A^{-1}\Delta A)^{-1}\|\|A^{-1}\| \, (\|\Delta b\| + \|\Delta A\|\|x\|) && \text{submultiplikativ} \\ &\leq \frac{\|A^{-1}\|}{1-\|A^{-1}\Delta A\|} \, (\|\Delta b\| + \|\Delta A\|\|x\|) && \text{Lemma 6.4} \\ &= \frac{\|A^{-1}\|}{1-\|A^{-1}\Delta A\|} \frac{\|A\|\|x\|}{\|A\|} \left(\frac{\|\Delta b\|}{\|x\|} + \|\Delta A\| \right) && \text{ausklammern, ergänzen} \\ &= \frac{\|A^{-1}\|\|A\|\|x\|}{1-\|A^{-1}\Delta A\|} \left(\frac{\|\Delta b\|}{\|A\|\|x\|} + \frac{\|\Delta A\|}{\|A\|} \right) \\ &\leq \frac{\|A^{-1}\|\|A\|\|x\|}{1-\|A^{-1}\|\|\Delta A)\|} \left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|} \right) && \text{verträglich, submultiplikativ} \\ &= \|x\| \frac{\operatorname{cond}(A)}{1-\operatorname{cond}(A)\frac{\|\Delta A\|}{\|A\|}} \left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|} \right) \end{aligned}$$

was die Aussage beweist.

□

Beispiel 6.6. Es sei die relativen Änderung in den Eingaben von der Größenordnung 10^{-k}, d.h. $\frac{\|\Delta A\|}{\|A\|} = 10^{-k}$ und $\frac{\|\Delta b\|}{\|b\|} = 10^{-k}$. Es sei weiter $\text{cond}(A) = 10^s$ aber $10^s \cdot 10^{-k} \ll 1$. Dann gilt

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{10^s}{1 - 10^s 10^{-k}} \cdot 2 \cdot 10^{-k} \approx 10^{s-k}.$$

Man verliert somit s Stellen an Genauigkeit (hier wird die Basis $\beta = 10$ betrachtet). Ein Zahlenbeispiel gefällig. Man ändert Matrixeinträge und rechte Seite relativ um 10^{-10}, die Kondition der Matrix sei 10^5, dann kann man relative Änderungen im Ergebnis von 10^{-5} erwarten. □

Allerdings zeigt dass folgende Beispiel, dass eine große Kondition nicht zwangsläufig zu großen Fehlern führen muss.

Beispiel 6.7. Betrachte die Diagonalmatrix

$$D = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}, \qquad D^{-1} = \begin{bmatrix} d_1^{-1} & 0 \\ 0 & d_2^{-1} \end{bmatrix}, \qquad d_1, d_2 \neq 0.$$

Es gilt

$$\text{cond}_\infty(D) = \frac{\max(|d_1|, |d_2|)}{\min(|d_1|, |d_2|)},$$

also etwa für $d_1 = 10^{10}$ und $d_2 = 1$ gilt $\text{cond}_\infty(D) = 10^{10}$. Andererseits sind alle Gleichungen in dem System $Dx = b$ unabhängig voneinander und die Berechnung $x_i = b_i/d_i$ ist gut konditioniert (Multiplikation und Division sind gut konditioniert). Das Problem hier ist, dass die Analyse mittels der Konditionszahl annimmt, dass Abhängigkeiten zwischen allen Komponenten der Eingabe und der Ausgabe bestehen. Dies ist hier nicht der Fall. □

Vorlesung 7

Eliminationsverfahren

Zunächst rekapitulieren wir das Gaußsche Eliminationsverfahren.

7.1 Dreieckssysteme

Die Lösung eines quadratischen linearen Gleichungssystemes $Ax = b$ gelingt besonders einfach wenn dieses Dreiecksgestalt hat:

$$\begin{aligned} a_{11}x_1 + a_{22}x_2 + \ldots + a_{nn}x_n &= b_1, \\ a_{22}x_2 + \ldots + a_{nn}x_n &= b_2, \\ &\vdots \\ a_{nn}x_n &= b_n. \end{aligned} \tag{7.1}$$

Dieses System ist regulär genau dann wenn für alle Koeffizienten $a_{ii} \neq 0$ gilt. Die Lösung gelingt dann mittels *rückwärts einsetzen*:

$$x_n = b_n/a_{nn}, \tag{7.2}$$

$$x_i = \left(b_i - \sum_{j=i+1}^{n} a_{ij}x_j \right) / a_{ii} \qquad i = n-1, \ldots, 1. \tag{7.3}$$

Der Rechenaufwand beträgt

$$N_{\text{Dreieck}}(n) = \sum_{i=0}^{n-1} (2i+1) = n^2$$

elementare Rechenoperationen (Addition, Multiplikation und Division werden hier nicht unterschieden obwohl sie möglicherweise unterschiedliche Rechenzeit benötigen).

Analog kann man auch *untere Dreieckssysteme* betrachten welche sich durch vorwärts einsetzen einfach lösen lassen.

7.2 Transformation auf Dreiecksgestalt

Hat ein Gleichungssystem keine Dreiecksform so lässt es sich mittels *elementarer Umformungen* in ein solches *transformieren*. Unter elementaren Umformungen verstehen wir:

i) Vertauschen zweier Gleichungen.

ii) Addition des Vielfachen einer Gleichung zu einer anderen Gleichung.

Das Gaußsche Eliminationsverfahren nutzt eine endliche Folge solcher elementarer Umformungen um das System auf Dreiecksgestalt zu bringen.

Algorithmus 7.1 (Gaußelimantion, 1. Formulierung)**.** Gegeben sei ein reguläres lineares Gleichungssystem $Ax = b$ mit einer $n \times n$ Matrix A. Das folgende Verfahren transformiert das System auf obere Dreiecksgestalt.

1) Das Verfahren konstruiert eine endliche Folge von Matrizen $A^{(k)} = \left(a_{i,j}^{(k)}\right)_{i,j=1}^{n}$ und Vektoren $b^{(k)} = \left(b_i^{(k)}\right)_{i=1}^{n}$. Zu Beginn setzen wir $A^{(0)} = A$, $b^{(0)} = b$ und $k = 1$.

2) Gilt $k = n$ so ist das Verfahren beendet.

3) Sonst ist nun $k < n$, also $n - k \geq 1$. Gehe wie folgt vor:

   a) Setze $\tilde{A}^{(k)} = A^{(k-1)}$ und $\tilde{b}^{(k)} = b^{(k-1)}$. Falls $\tilde{a}_{k,k}^{(k)} = 0$ finde einen Index $s \in \{k+1, \ldots, n\}$ so dass $\tilde{a}_{s,k}^{(k)} \neq 0$ und vertausche die Zeilen k und s in $\tilde{A}^{(k)}$ und die Einträge k und s in $\tilde{b}^{(k)}$. Am Ende dieses Schrittes gilt

   $$\tilde{a}_{k,k}^{(k)} \neq 0.$$

   Das Element $\tilde{a}_{k,k}^{(k)}$ heißt *Pivotelement*, die k-te Zeile heißt *Pivotzeile*.

   b) Setze $A^{(k)} = \tilde{A}^{(k)}$, $b^{(k)} = \tilde{b}^{(k)}$ und definiere die Vektoren

   $$l_i^{(k)} = \begin{cases} 0 & 1 \leq i \leq k \\ \tilde{a}_{i,k}^{(k)} / \tilde{a}_{k,k}^{(k)} & k+1 \leq i \leq n \end{cases}, \quad u_j^{(k)} = \begin{cases} 0 & 1 \leq j < k \\ \tilde{a}_{k,j}^{(k)} & k \leq j \leq n \end{cases} \tag{7.4}$$

   Für $k+1 \leq i \leq n$ *überschreibe* die Elemente von $A^{(k)}$ und $b^{(k)}$ wie folgt.

   $$a_{i,j}^{(k)} = \tilde{a}_{i,j}^{(k)} - l_i^{(k)} u_j^{(k)}, \qquad j \in \{k, \ldots, n\}, \tag{7.5}$$

   $$b_i^{(k)} = \tilde{b}_i^{(k)} - l_i^{(k)} \tilde{b}_k^{(k)}, \tag{7.6}$$

   d.h. man subtrahiert das $l_i^{(k)}$-fache der Zeile k von der Zeile $i > k$. Am Ende dieses Schrittes hat die Matrix die Gestalt

   $$A^{(k)} = \begin{bmatrix} a_{1,1}^{(k)} & \ldots & a_{1,k}^{(k)} & a_{1,k+1}^{(k)} & \ldots & a_{1,n}^{(k)} \\ 0 & \ddots & \vdots & & & \vdots \\ \vdots & & a_{k,k}^{(k)} & a_{k,k+1}^{(k)} & \ldots & a_{k,n}^{(k)} \\ \vdots & & 0 & a_{k+1,k+1}^{(k)} & \ldots & a_{k+1,n}^{(k)} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \ldots & 0 & a_{n,k+1}^{(k)} & \ldots & a_{n,n}^{(k)} \end{bmatrix}$$

   c) Setze $k = k + 1$ und gehe zu Schritt 2). □

Satz 7.2 (Gaußsches Eliminationsverfahren)**.** Das oben beschriebene Verfahren transformiert jedes quadratische, reguläre lineare Gleichungssystem der Dimension n in $n - 1$ Schritten auf obere Dreiecksgestalt.

Beweis. Würde sich im Schritt 3a) kein Pivotelement finden lassen, dann wären die Zeilen k bis n linear abhängig. Dies ist ein Widerspruch zur Voraussetzung. In Gleichung (7.5) gilt $a_{i,k}^{(k)} = \tilde{a}_{i,k}^{(k)} - l_i^{(k)} u_k^{(k)} = \tilde{a}_{i,k}^{(k)} - (\tilde{a}_{i,k}^{(k)} / \tilde{a}_{k,k}^{(k)}) \tilde{a}_{k,k}^{(k)} = 0$. Somit wird die k-te Spalte für die Zeilen $i \geq k+1$ zu Null eliminiert, was die Nullstruktur von $A^{(k)}$ erklärt. Wenn $k = n - 1$ erreicht ist, gilt offensichtlich $k + 1 = n$ und die Matrix $A^{(n-1)}$ hat obere Dreiecksgestalt. □

Bemerkung 7.3. 1) Der Schritt 3a) oben heißt *Pivotisierung*. In exakter Arithmetik ist es nur notwendig ein Nichtnullelement in der Spalte k zu finden. In Fließkommaarithmetik wird dieser Schritt eine wichtige Rolle spielen.

2) In einer praktischen Implementierung wird man für die Matrizen $A^{(k)}$ nicht jeweils extra Speicher reservieren sondern die Matrixelemente sukzessive überschreiben. Die Formulierung hier dient nur dazu präzise formulieren zu können welche Einträge die Matrix in jedem Schritt hat.

Beispiel 7.4. Folgendes Beispiel zeigt die Gauß-Elimination eines Gleichungssystemes mit $n = 4$. Wir fassen Matrix und rechte Seite in Form einer $n \times n+1$ Matrix $[A^{(k)}|b^{(k)}]$ zusammen. Das Beispiel ist so gewählt, dass keine Zeilenvertauuschungen nötig sind und Lösung und Matrix Einträge sind ganze Zahlen. Das Pivotelement ist jeweils durch einen Kasten gekennzeichnet.

$$[A^{(0)}|b^{(0)}] = \left[\begin{array}{cccc|c} \boxed{2} & 4 & 6 & 8 & 40 \\ 16 & 33 & 50 & 67 & 330 \\ 4 & 15 & 31 & 44 & 167 \\ 10 & 29 & 63 & 97 & 350 \end{array}\right], \quad [A^{(1)}|b^{(1)}] = \left[\begin{array}{cccc|c} 2 & 4 & 6 & 8 & 40 \\ 0 & \boxed{1} & 2 & 3 & 10 \\ 0 & 7 & 19 & 28 & 87 \\ 0 & 9 & 33 & 57 & 150 \end{array}\right],$$

$$[A^{(2)}|b^{(2)}] = \left[\begin{array}{cccc|c} 2 & 4 & 6 & 8 & 40 \\ 0 & 1 & 2 & 3 & 10 \\ 0 & 0 & \boxed{5} & 7 & 17 \\ 0 & 0 & 15 & 30 & 60 \end{array}\right], \qquad [A^{(3)}|b^{(3)}] = \left[\begin{array}{cccc|c} 2 & 4 & 6 & 8 & 40 \\ 0 & 1 & 2 & 3 & 10 \\ 0 & 0 & 5 & 7 & 17 \\ 0 & 0 & 0 & 9 & 9 \end{array}\right].$$

Schließlich liefert Rückwärtseinsetzen:

$$x_4 = 9/9 = \boxed{1}, \qquad x_3 = (17 - 7 \cdot 1)/5 = \boxed{2},$$
$$x_2 = (10 - 2 \cdot 2 - 3 \cdot 1)/1 = \boxed{3}, \quad x_1 = (40 - 4 \cdot 3 - 6 \cdot 2 - 8 \cdot 1)/2 = \boxed{4}.$$

□

Lemma 7.5. Das Gaußsche Eliminationsverfahren benötigt

$$N_{\text{Gauß}}(n) = \frac{2}{3}n^3 + O(n^2)$$

Rechenoperationen um eine $n \times n$ Matrix auf obere Dreiecksgestalt zu transformieren.

Beweis. Wir zählen die Anzahl der Operationen im Schritt 3b.

$$\begin{aligned} N_{\text{Gauß}}(n) &= \sum_{k=1}^{n-1} \left\{ \underbrace{n-k}_{l_i^{(k)}} + \underbrace{(n-k)}_{\#\text{ Zeilen}} \underbrace{[(n-k)\cdot 2 + 2]}_{(7.5),(7.6)} \right\} \\ &= 2\sum_{k=1}^{n-1}(n-k)^2 + 3\sum_{k=1}^{n-1}(n-k) = \frac{2}{3}n^3 + O(n^2). \end{aligned}$$

□

Somit lässt sich ein lineares Gleichungssystem mit insgesamt $\frac{2}{3}n^3 + O(n^2)$ Operationen lösen.

7.3 Matrixformulierung der Gauß-Elimination

Das Gaußsche Eliminationsverfahren kann auf verschiedene Weisen kompakt formuliert werden wenn man zu einer Matrixform übergeht.

Definition 7.6 (Permutationsmatrizen)**.** Es seien Zahlen $r, s, n \in \mathbb{N}$ mit $1 \leq r \leq s \leq n$ gegeben. Die $n \times n$ Matrix $P_{r,s}$ heißt *Permutationsmatrix* wenn gilt

$$(P_{r,s})_{i,j} = \begin{cases} 1 & i = j \wedge i \neq r \wedge j \neq s \\ 1 & (i = r \wedge j = s) \vee (i = s \wedge j = r) \\ 0 & \text{sonst} \end{cases}$$

□

Permutationsmatrizen haben also die Form

$$P_{rs} = \begin{bmatrix} 1 & & & & & & & & & \\ & \ddots & & & & & & & & \\ & & 1 & & & & & & & \\ & & & 0 & & & 1 & & & \\ & & & & 1 & & & & & \\ & & & & & \ddots & & & & \\ & & & & & & 1 & & & \\ & & & 1 & & & & 0 & & \\ & & & & & & & & 1 & \\ & & & & & & & & & \ddots \\ & & & & & & & & & & 1 \end{bmatrix}$$

Lemma 7.7. Für Permutationsmatrizen $P_{r,s}$ gilt

i) Für eine $n \times n$ Matrix A enthält das Produkt $P_{r,s}A$ die Matrix A mit den Zeilen r und s vertauscht.

ii) Für eine $n \times n$ Matrix A enthält das Produkt $AP_{r,s}$ die Matrix A mit den Spalten r und s vertauscht.

iii) Es gilt $P_{r,s} = P_{r,s}^T$.

iv) Es gilt $P_{r,s}^{-1} = P_{r,s}$. Damit ist $P_{r,s}$ eine orthogonale Matrix.

Beweis. Alle Eigenschaften beruhen auf der speziellen Form der Matrizen $P_{r,s}$ und der Definition der Matrixmultiplikation.

i) Für $\tilde{A} = P_{r,s}A$ gilt $\tilde{a}_{i,j} = \sum_{k=1}^{n} (P_{r,s})_{i,k} a_{k,j}$. Damit gilt

$$\begin{aligned} i \neq r \wedge i \neq s \quad &\Rightarrow \quad (P_{r,s})_{i,i} = 1 \quad \Rightarrow \quad \tilde{a}_{i,j} = a_{ij}, \\ i = r \quad &\Rightarrow \quad (P_{r,s})_{i,s} = 1 \quad \Rightarrow \quad \tilde{a}_{i,j} = a_{sj}, \\ i = s \quad &\Rightarrow \quad (P_{r,s})_{i,r} = 1 \quad \Rightarrow \quad \tilde{a}_{i,j} = a_{rj}. \end{aligned}$$

ii) $\tilde{A} = AP_{r,s}$ zeigt man analog, nur gilt $\tilde{a}_{i,j} = \sum_{k=1}^{n} a_{i,k} (P_{r,s})_{k,j}$.

iii) Folgt direkt aus der Definition.

iv) Wegen i) und der Gestalt von $P_{r,s}$ gilt $P_{r,s}P_{r,s} = I$ also $P_{r,s}^{-1} = P_{r,s}$. Wegen iii) gilt $P_{r,s}^{-1} = P_{r,s}^T$.

□

Der Vertauschungsschritt 3a in der Gaußelimination lässt sich mittels einer Permutationsmatrix formulieren. Für den Eliminationsschritt 3b betrachten wir folgende Matrizen. Seien l und u Vektoren der Länge n, dann ist $A = lu^T$ eine $n \times n$ Matrix mit den Einträgen $a_{i,j} = l_i u_j$. Da alle Zeilen Vielfache voneinander sind hat A den Rang 1. Für den Eliminationsschritt 3b erhalten wir daher mit den Definitionen von dort $A^{(k)} = \tilde{A}^{(k)} - l^{(k)} \left(u^{(k)}\right)^T$. Damit erhalten wir die folgende, kompaktere Formulierung des Gaußschen Eliminationsverfahrens.

Algorithmus 7.8 (Gaußelimination, Formulierung in Matrixform)**.**

1) Setze $A^{(0)} = A$, $b^{(0)} = b$ und $k = 1$.

2) Ist $k = n$ so ist das Verfahren beendet.

3) Sonst ist nun $k < n$, also $n - k \geq 1$.

   a) Finde einen Index $s \in \{k, \ldots, n\}$ so dass $a_{s,k}^{(k-1)} \neq 0$ und setze $\tilde{A}^{(k)} = P_{k,s}A^{(k-1)}$, $\tilde{b}^{(k)} = P_{k,s}b^{(k-1)}$

   b) Definiere wie oben die Vektoren der Länge n

$$l_i^{(k)} = \begin{cases} 0 & 1 \leq i \leq k \\ \tilde{a}_{i,k}^{(k)}/\tilde{a}_{k,k}^{(k)} & k+1 \leq i \leq n \end{cases}, \; u_j^{(k)} = \begin{cases} 0 & 1 \leq j < k \\ \tilde{a}_{k,j}^{(k)} & k \leq j \leq n \end{cases}$$

   und setze $A^{(k)} = \tilde{A}^{(k)} - l^{(k)} \left(u^{(k)}\right)^T$, $b^{(k)} = \tilde{b}^{(k)} - l^{(k)}\tilde{b}_k^{(k)}$. Man sagt $A^{(k)}$ entsteht aus $\tilde{A}^{(k)}$ durch ein *Rang-1-Update*. *Achtung*: Wie beschrieben führt dieser Schritt *mehr* Rechenoperationen aus wie der ursprüngliche Algorithmus, da $l^{(k)}$, $u^{(k)}$ die Länge n haben. In einer Implementierung würde man ausnutzen, dass nur $(n-k-1)^2$ der Updates wirklich durchzuführen sind. wirklich auszuführen

   c) Setze $k = k + 1$ und gehe zu Schritt 2). □

Eine weitere Umformulierung gelingt mittels der

Definition 7.9 (Frobeniusmatrizen)**.** Für $n, k \in \mathbb{N}$, $k \leq n$, sei I_n die $n \times n$ Einheitsmatrix, $e^{(k)}$ der k-te Koordinateneinheitsvektor und g ein Vektor der Länge n mit $g_i = 0$ für $i \leq k$. Dann heißt

$$G_k(g) = I_n + g\left(e^{(k)}\right)^T \tag{7.7}$$

Frobeniusmatrix. □

Frobeniusmatrizen bestehen also aus der Identität mit einer Rang-1-Modifikation

und haben die Form

$$G_k(g) = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & & \\ & & & g_{k+1} & 1 & \\ & & & \vdots & & \ddots \\ & & & g_n & & 1 \end{bmatrix}.$$

Lemma 7.10. Für Frobeniusmatrizen der Form $G_k(g) = I_n + g\left(e^{(k)}\right)^T$ gelten die folgenden Eigenschaften.

i) Sei $\tilde{A}$ eine $n \times n$ Matrix. Für das Produkt $A = G_k(g)\tilde{A}$ mit einer Frobeniusmatrix von links gilt

$$a_{ij} = \begin{cases} \tilde{a}_{i,j} & i \leq k \\ \tilde{a}_{i,j} + g_i^{(k)} \tilde{a}_{k,j} & i > k \end{cases}.$$

Es wird also gerade das $g_i^{(k)}$-fache der Zeile k auf die Zeile i addiert.

ii) Für die Inverse gilt $G_k^{-1}(g) = I_n - g\left(e^{(k)}\right)^T = G_k(-g)$.

iii) Für das Produkt der Inversen von Frobeniusmatrizen gilt mit $k < n$

$$G_1^{-1}\left(g^{(1)}\right) G_2^{-1}\left(g^{(2)}\right) \ldots G_k^{-1}\left(g^{(k)}\right) = I_n - \sum_{j=1}^{k} g^{(j)} \left(e^{(j)}\right)^T.$$

Das Produkt ist demnach eine untere Dreiecksmatrix mit 1 auf der Hauptdiagonale.

iv) Für alle $k < \alpha \leq \beta \leq n$ gilt

$$P_{\alpha,\beta} G_k(g) = G_k(P_{\alpha,\beta} g) P_{\alpha,\beta}.$$

Beweis. i) Wir rechnen

$$A = G_k(g)\tilde{A} = \left(I_n + g\left(e^{(k)}\right)^T\right)\tilde{A} = \tilde{A} + g\left(e^{(k)}\right)^T \tilde{A}.$$

Nun extrahiert $(e^{(k)})^T \tilde{A}$ gerade die k-te Zeile von $\tilde{A}$ und $g\left(e^{(k)}\right)^T \tilde{A}$ ist somit gerade die Rang-1-Matrix welche aus $g_i^{(k)}$-fachen der k-ten Zeile von $\tilde{A}$ besteht, was behauptet wurde. Da $g_i^{(k)} = 0$ für $i \leq k$, nach Definition der Frobeniusmatrizen, gilt $\tilde{a}_{i,j} = a_{i,j}$ für $i \leq k$.

ii) Wir rechnen los

$$\left(I_n - g\left(e^{(k)}\right)^T\right)\left(I_n + g\left(e^{(k)}\right)^T\right) = I_n - g\left(e^{(k)}\right)^T + g\left(e^{(k)}\right)^T - g\left(e^{(k)}\right)^T g\left(e^{(k)}\right)^T = 0$$

da im letzten Term $\left(e^{(k)}\right)^T g = 0$ da $g_i = 0$ für $i \leq k$.

iii) Beweis durch Induktion. Für $k = 1$ hat $G_1^{-1}(g)$ offensichtlich die angegebene Form. Sei die Behauptung bis $k-1$ bewiesen. Dann gilt

$$\begin{aligned} G_1^{-1}\left(g^{(1)}\right) G_2^{-1}\left(g^{(2)}\right) \dots G_k^{-1}\left(g^{(k)}\right) &= \\ &= \left(I_n - \sum_{j=1}^{k-1} g^{(j)} \left(e^{(j)}\right)^T\right) \left(I_n - g^{(k)} \left(e^{(k)}\right)^T\right) \\ &= I_n - \sum_{j=1}^{k-1} g^{(j)} \left(e^{(j)}\right)^T - g^{(k)} \left(e^{(k)}\right)^T + \sum_{j=1}^{k-1} \left(g^{(j)} \left(e^{(j)}\right)^T g^{(k)} \left(e^{(k)}\right)^T\right) \\ &= I_n - \sum_{j=1}^{k} g^{(j)} \left(e^{(j)}\right)^T \end{aligned}$$

da $\left(e^{(j)}\right)^T g^{(k)} = 0$ für alle $j < k$.

iv) Wir rechnen

$$\begin{aligned} P_{\alpha,\beta} G_k(g) &= P_{\alpha,\beta}\left(I_n + g\left(e^{(k)}\right)^T\right) \\ &= P_{\alpha,\beta} + P_{\alpha,\beta} g \left(e^{(k)}\right)^T P_{\alpha,\beta} P_{\alpha,\beta} \\ &= \left(I_n + P_{\alpha,\beta} g \left(e^{(k)}\right)^T P_{\alpha,\beta}\right) P_{\alpha,\beta} \\ &= \left(I_n + P_{\alpha,\beta} g \left(e^{(k)}\right)^T\right) P_{\alpha,\beta} = G_k(P_{\alpha,\beta} g) P_{\alpha,\beta} \end{aligned}$$

da $\left(e^{(k)}\right)^T P_{\alpha,\beta} = \left(e^{(k)}\right)^T$ wegen $k < \alpha \le \beta$ (die k-te Zeile von $P_{\alpha,\beta}$ entspricht der k-ten Zeile der Einheitsmatrix I_n).

□

Mit den Frobeniusmatrizen können wir nun zu Schritt 3b in Algorithmus 7.8 zurückkehren. Der Vektor $l^{(k)}$ erfüllt $l_i^{(k)} = 0$ für alle $i \le k$ und kann somit als g-Vektor in einer Frobeniusmatrix eingesetzt werden. Wegen $\tilde{a}_{k,j}^{(k)} = 0$ für $j < k$ darf man $\left(u^{(k)}\right)^T$ durch $\left(e^{(k)}\right)^T \tilde{A}^{(k)}$ ersetzen und es gilt

$$A^{(k)} = \tilde{A}^{(k)} - l^{(k)} \left(u^{(k)}\right)^T = \tilde{A}^{(k)} - l^{(k)} \left(e^{(k)}\right)^T \tilde{A}^{(k)} = G_k\left(-l^{(k)}\right) \tilde{A}^{(k)}.$$

Algorithmus 7.11 (Gaußelimination, Formulierung mit Frobeniusmatrizen)**.** 1) $A^{(0)} = A$, $b^{(0)} = b$ und $k = 1$.

2) Ist $k = n$ so ist das Verfahren beendet.

3) Sonst ist nun $k < n$, also $n - k \ge 1$.

a) Finde einen Index $s \in \{k, \dots, n\}$ so dass $a_{s,k}^{(k-1)} \neq 0$ und setze $\tilde{A}^{(k)} = P_{k,s} A^{(k-1)}$, $\tilde{b}^{(k)} = P_{k,s} b^{(k-1)}$

b) Definiere den Vektor $l_i^{(k)} = \begin{cases} 0 & 1 \le i \le k \\ \tilde{a}_{i,k}^{(k)} / \tilde{a}_{k,k}^{(k)} & k+1 \le i \le n \end{cases}$ und setze $A^{(k)} = G_k\left(-l^{(k)}\right) \tilde{A}^{(k)}$, $b^{(k)} = G_k\left(-l^{(k)}\right) \tilde{b}^{(k)}$.

c) Setze $k = k + 1$ und gehe zu Schritt 2). □

Vorlesung 8

LU-Zerlegung

Im letzten Kapitel haben wir das Gaußsche Eliminationsverfahren mittels Permutations- und Frobeniusmatrizen in Matrixform geschrieben. Dies erlaubt eine geschickte Implementierung des Verfahrens welches üblicherweise in der numerischen Lösung von Gleichungssystemen Verwendung findet.

8.1 Herleitung der Zerlegung

Unten treten häufig Produkte von Matrizen auf. Dabei verwenden wir die folgende Notation. Für $a, b \in \mathbb{N}_0$, ist

$$\prod_{i=a}^{b} M_i = M_b M_{b-1} \dots M_a. \tag{8.1}$$

Man beachte die Reihenfolge der Faktoren. Der Faktor mit Index a ist immer rechts und der Faktor mit Index b ist immer links unabhängig davon ob $a \leq b$ oder $a > b$. Zum Beispiel

$$\prod_{i=1}^{3} M_i = M_3 M_2 M_1, \qquad \text{und} \qquad \prod_{i=3}^{1} M_i = M_1 M_2 M_3.$$

Mit den Vorarbeiten aus dem letzten Kapitel können wir das Hauptresultat direkt angehen.

Satz 8.1. Sei $A \in \mathbb{K}^{n\times n}$ regulär. Dann gibt es eine Zerlegung

$$PA = LU \tag{8.2}$$

in ein Produkt von $n \times n$ Matrizen mit folgenden Eigenschaften

i) $P = \prod_{k=1}^{n-1} P_{k,s_k}$ ist ein Produkt von Permutationsmatrizen wobei $s_k \geq k$ die Indizes für die Zeilentausche aus Schritt 3a des Gauß-Algorithmus sind.

ii) L ist eine untere Dreiecksmatrix mit den Einträgen

$$L = \begin{bmatrix} 1 & 0 & \dots & 0 \\ l_{2,1} & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ l_{n,1} & \dots & l_{n,n-1} & 1 \end{bmatrix} = I_n + \sum_{k=1}^{n-1} \left(\prod_{j=k+1}^{n-1} P_{j,s_j} l^{(k)} \left(e^{(k)}\right)^T \right)$$

mit den Vektoren $l^{(k)}$ aus Schritt 3b des Gauß-Algorithmus.

iii) U ist eine obere Dreiecksmatrix

$$U = \begin{bmatrix} u_{1,1} & u_{1,2} & \dots & u_{1,n} \\ 0 & u_{2,2} & & u_{2,n} \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & u_{n,n} \end{bmatrix}, \qquad u_{i,j} = a_{i,j}^{(n-1)} \quad 1 \leq i \leq j \leq n,$$

d.h. die Einträge von U stimmen mit denen der Matrix $A^{(n-1)}$ am Ende des Gauß-Algorithmus überein.

Der Name der Zerlegung (8.2) ergibt sich aus den englischen Bezeichnungen für untere (lower) und obere (upper) Dreiecksmatrizen (triangular matrices).

Beweis. Aus der dritten Formulierung der Gaußelimination in Algorithmus 7.11 entnehmen wir, dass in jedem Schritt des Verfahrens das Gleichungssystem erst mit einer Permutationsmatrix und dann mit einer Frobeniusmatrix von links multipliziert wird:

$$\begin{aligned}
& Ax = b \\
\Leftrightarrow \quad & G_1\left(-l^{(1)}\right) P_{1,s_1} Ax = G_1\left(-l^{(1)}\right) P_{1,s_1} b \\
\Leftrightarrow \quad & G_2\left(-l^{(2)}\right) P_{2,s_2} G_1\left(-l^{(1)}\right) P_{1,s_1} Ax = G_2\left(-l^{(2)}\right) P_{2,s_2} G_1\left(-l^{(1)}\right) P_{1,s_1} b \\
& \vdots \\
& \underbrace{G_{n-1} P_{n-1,s_{n-1}} \dots G_2 P_{2,s_2} G_1 P_{1,s_1} A}_{A^{(n-1)}=U} x = G_{n-1} P_{n-1,s_{n-1}} \dots G_2 P_{2,s_2} G_1 P_{1,s_1} b
\end{aligned}$$

(Um Platz zu sparen wurden die Argumente der Frobeniusmatrizen in der letzten Zeile weggelassen). Nach $n-1$ Schritten hat die Matrix links obere Dreiecksgestalt und wir nennen diese Matrix U, also:

$$G_{n-1} P_{n-1,s_{n-1}} \dots G_2 P_{2,s_2} G_1 P_{1,s_1} A = U. \tag{8.3}$$

Nach Lemma 7.10 Nr. iv) kann man eine Frobeniusmatrix G_k und eine Permutationsmatrix $P_{\alpha,\beta}$ tauschen wenn $\alpha > k$, etwa

$$P_{2,s_2} G_1\left(-l^{(1)}\right) = G_1\left(-P_{2,s_2} l^{(1)}\right) P_{2,s_2}.$$

Auf diese Weise kann man alle Permutationsmatrizen sukzessive „nach rechts wandern“ lassen bis Gleichung (8.3) die folgende Form hat:

$$G_{n-1}\left(-l^{(n-1)}\right) \dots G_1\left(\left(\prod_{k=2}^{n-1} P_{k,s_k}\right)\left(-l^{(1)}\right)\right) \underbrace{\left(\prod_{k=1}^{n-1} P_{k,s_k}\right)}_{P} A = U. \tag{8.4}$$

Nun nutzen wir die Eigenschaft $G_k^{-1}(g) = G_k(-g)$ aus Lemma 7.10 Nr. ii) um die Frobeniusmatrizen auf die rechte Seite zu schaffen und erhalten

$$PA = \underbrace{G_1\left(\left(\prod_{k=2}^{n-1} P_{k,s_k}\right) l^{(1)}\right) \dots G_{n-1}\left(l^{(n-1)}\right)}_{=L} U \tag{8.5}$$

wobei wir auf der rechten Seite den Faktor L identifizieren. Schließlich zeigt Lemma 7.10 Nr. iii), dass L die behauptete Form hat. □

Wie kann man diese Zerlegung nun nutzen um eine lineares Gleichungssystem $Ax = b$ zu lösen. Dazu rechne

$$Ax = b \;\Leftrightarrow\; PAx = Pb \;\Leftrightarrow\; LUx = Pb \;\Leftrightarrow\; Ly = Pb \wedge Ux = y.$$

Algorithmus 8.2. Satz 8.1 kann man folgendermaßen nutzen um ein lineares Gleichungssystem $Ax = b$ zu lösen:

1) Berechne die LU-Zerlegung $PA = LU$, also die Matrizen L und U.

2) Berechne $\tilde{b} = Pb$.

3) Löse das Dreieckssystem $Ly = \tilde{b}$ durch vorwärts einsetzen.

4) Löse das Dreieckssystem $Ux = y$ durch rückwärts einsetzen.

Bemerkung 8.3. a) Die LU-Zerlegung ist eine (geschickte) Umformulierung der Gaußelimination bei der die Transformation von A auf obere Dreiecksgestalt und die Modifikation der rechten Seite b separiert werden. Löst man ein Gleichungssystem für genau eine rechte Seite so werden exakt die gleichen Operationen wie in der Gaußelimination durchgeführt, nur in anderer Reihenfolge.

b) Der Vorteil der LU-Zerlegung kommt dann zum tragen wenn ein Gleichungssystem mit der selben Matrix A für *mehrere* rechte Seiten gelöst werden soll. Der Aufwand zur Berechnung der LU-Zerlegung ist im wesentlichen gleich der der Gaußelimination, also $\frac{2}{3}n^3 + O(n^2)$, während der Aufwand zur Lösung der beiden Dreieckssysteme nur $2n^2$ beträgt. Somit erhöht sich der Aufwand für eine weitere rechte Seite nur um $2n^2$.

c) Will man beispielsweise die Inverse A^{-1} einer Matrix berechnen, so kann man dies als n simultane Gleichungssysteme mit den Koordinateneinheitsvektoren als rechte Seiten sehen. Mit den Überlegungen von oben kann man somit die Inverse mit dem Aufwand $\frac{8}{3}n^3 + O(n^2)$ berechnen.

8.2 Vollpivotisierung

Aufgabe des Schrittes 3a in der Gaußelimination ist es sicher zu stellen, dass $\tilde{a}_{k,k}^{(k)}$ ungleich Null ist. Bisher haben wir dazu, falls notwendig, einen Zeilentausch mit einer Zeile $s_k > k$ durchgeführt. Ebenso könnte man allerdings auch einen Spaltentausch durchführen. Angenommen es sei $\tilde{a}_{k,r_k}^{(k)} \neq$ mit $r_k > k$, dann tauscht $\tilde{A}^{(k)} P_{k,r_k}$ die k-te und die r_k-te Spalte in $\tilde{A}^{(k)}$. Im ersten Schritt verwendet man das so:

$$Ax = b \qquad \Leftrightarrow \qquad AP_{1,r_1} P_{1,r_1} x = b.$$

$P_{1,r_1} x$ entspricht einer Vertauschung der Komponenten x_k und x_{r_k} im Lösungsvektor.

Danach kann die Elimination mittels der Frobeniusmatrix erfolgen wie bisher. Schließlich kann sowohl ein Zeilentausch als auch ein Spaltentausch vorgenommen werden um eine beliebiges Element $\tilde{a}_{s_k,r_k}^{(k)} \neq$ zum Pivotelement zu machen. Dies führt zu der folgenden Variante der LU-Zerlegung.

Satz 8.4 (LU-Zerlegung mit Vollpivotisierung)**.** Sei $A \in \mathbb{K}^{n\times n}$ regulär. Dann gibt es eine Zerlegung

$$PAQ = LU \tag{8.6}$$

in ein Produkt von $n \times n$ Matrizen mit folgenden Eigenschaften

i) $P = \prod_{k=1}^{n-1} P_{k,s_k}$ und $Q = \prod_{k=n-1}^{1} P_{k,r_k}$ sind Produkte von Permutationsmatrizen (man beachte die Reihenfolge) wobei in Schritt 3a des Gauß-Algorithmus das Element mit Index (s_k, r_k) zum Pivotelement erklärt wird.

ii) L ist eine untere Dreiecksmatrix mit den Einträgen

$$L = \begin{bmatrix} 1 & 0 & \dots & 0 \\ l_{2,1} & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ l_{n,1} & \dots & l_{n,n-1} & 1 \end{bmatrix} = I_n + \sum_{k=1}^{n-1} \left(\prod_{j=k+1}^{n-1} P_{j,s_j} l^{(k)} \left(e^{(k)}\right)^T \right)$$

mit den Vektoren $l^{(k)}$ aus Schritt 3b des Gauß-Algorithmus der um den zusätzlichen Spaltentausch modifiziert wurde.

iii) U ist eine obere Dreiecksmatrix

$$U = \begin{bmatrix} u_{1,1} & u_{1,2} & \dots & u_{1,n} \\ 0 & u_{2,2} & & u_{2,n} \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & u_{n,n} \end{bmatrix}, \qquad u_{i,j} = a_{i,j}^{(n-1)} \quad 1 \le i \le j \le n,$$

d.h. die Einträge von U stimmen mit denen der Matrix $A^{(n-1)}$ am Ende des Gauß-Algorithmus mit dem zusätzlichen Spaltentausch überein.

Beweis. Wir wenden Schrittweise die Zeilen- und Spaltentausche sowie die Eliminationen an und erhalten:

$$\begin{aligned} Ax &= b \\ G_1 P_{1,s_1} A P_{1,r_1} P_{1,r_1} x &= G_1 P_{1,s_1} b \\ G_2 P_{2,s_2} G_1 P_{1,s_1} A P_{1,r_1} P_{2,r_2} P_{2,r_2} P_{1,r_1} x &= G_1 P_{1,s_1} b \\ &\vdots \\ \underbrace{G_{n-1} P_{n-1,s_{n-1}} \dots G_1 P_{1,s_1} A P_{1,r_1} \dots P_{n-1,r_{n-1}}}_{A^{(n-1)}=U} P_{n-1,r_{n-1}} \dots P_{1,r_1} x &= G_{n-1} P_{n-1,s_{n-1}} \dots G_1 P_{1,s_1} b. \end{aligned}$$

Wie in der Variante mit den reinen Spaltentauschen kann man wieder die Permutationsmatrizen mit den Frobeniusmatrizen tauschen und man erhält mit der eingeführten Notation

$$\left(\prod_{k=1}^{n-1} G_k\right) \left(\prod_{k=1}^{n-1} P_{k,s_k}\right) A \left(\prod_{k=n-1}^{1} P_{k,r_k}\right) \left(\prod_{k=1}^{n-1} P_{k,r_k}\right) = \left(\prod_{k=1}^{n-1} G_k\right) \left(\prod_{k=1}^{n-1} P_{k,s_k}\right) b,$$

also

$$\left(\prod_{k=1}^{n-1} G_k\right) PAQ = U \qquad \Leftrightarrow \qquad PAQ = LU.$$

□

Die Anwendung der Zerlegung (8.6) basiert dann auf der Äquivalenz

$$\begin{aligned} Ax = b \Leftrightarrow\ & PAQQ^{-1}x = Pb \ \Leftrightarrow\ LUQ^{-1}x = Pb \\ \Leftrightarrow\ & Lz = Pb \ \wedge\ Uy = z \ \wedge\ Q^{-1}x = y \end{aligned}$$

und führt zu folgendem Algorithmus

Algorithmus 8.5. 1) Berechne die LU-Zerlegung $PAQ = LU$, also die Matrizen L, U sowie die Permutations Matrizen P und Q.

2) Berechne $\tilde{b} = Pb$.

3) Löse das Dreieckssystem $Lz = \tilde{b}$ durch vorwärts einsetzen.

4) Löse das Dreieckssystem $Uy = z$ durch rückwärts einsetzen.

5) Berechne $x = Qy$.
Es ist somit nur ein zusätzlicher Permutationsschritt am Ende nötig. □

8.3 Praktische Implementierung

Nun einige praktische Hinweise zur Implementierung der LU-Zerlegung.

Wenden wir uns erst mal den Permutationsmatrizen zu. Diese werden nicht wie gewöhnliche Matrizen als n^2 Zahlen gespeichert und auch die Multiplikation mit Permutationsmatrizen wird nicht als gewöhnliche Matrix-Vektor-Multiplikation bzw. Matrixmultiplikation durchgeführt da dies zuviel Speicher und Rechenzeit in Anspruch nehmen würde. Stattdessen benötigt die Speicherung einer Permutationsmatrix nur die Dimension n sowie die Indizes r, s. In der LU-Zerlegung treten die Matrizen zudem immer als Sequenz auf, mit den speziellen Indizes k, s_k. Daher muss man zur Realisierung der Permutationsmatrix $P = \prod_{k=1}^{n-1} P_{k,s_k}$ nur einen Vektor von natürlichen Zahlen $p = (s_1, s_2, \ldots)^T$ speichern. Folgende Funktion zeigt die Anwendung der P-Matrix auf einen Vektor aus der HDNum-Bibliothek:

```
template<class T>
void permute_forward (const Vector<std::size_t>& p, Vector<T>& b)
{
  if (b.size()!=p.size())
    HDNUM_ERROR("permutation␣vector␣incompatible␣with␣rhs");

  for (std::size_t k=0; k<b.size()-1; ++k)
    if (p[k]!=k) std::swap(b[k],b[p[k]]);
}
```

Die Speicherung der Matrizen L und U erfolgt üblicherweise im Speicherbereich der zu zerlegenden Matrix A, welche dadurch überschrieben wird. Die Einträge (i, j), $j \geq i$ enthalten gehören zu U, die Einträge (i, j), $j < i$ gehören zu L und die Hauptdiagonalelemente von L werden nicht gespeichert da diese immer 1 sind. Folgende Funktion realisiert Algorithmus 8.5 in HDNum.

```
template<class T>
void linsolve (DenseMatrix<T>& A, Vector<T>& x, Vector<T>& b)
{
  if (A.rowsize()!=A.colsize() || A.rowsize()==0)
    HDNUM_ERROR("need␣square␣and␣nonempty␣matrix");
  if (A.rowsize()!=b.size())
    HDNUM_ERROR("right␣hand␣side␣incompatible␣with␣matrix");

  Vector<std::size_t> p(x.size());
  Vector<std::size_t> q(x.size());
  lr_fullpivot(A,p,q);
  permute_forward(p,b);
  solveL(A,b,b);
  solveR(A,x,b);
  permute_backward(q,x);
}
```

Die Integer-Vektoren `p` und `q` realisieren die Matrizen P, bzw. Q die während der LU-Zerlegung von A in `lr_fullpivot` berechnet werden (in HDNum heißt die Zerlegung nach ihrem deutschen Namen LR-Zerlegung). `permute_forward` und `permute_backward` realisieren deren Anwendung auf einen Vektor. Schließlich realisieren `solveL` und `solveR` die Lösung der Dreiecksfaktoren.

8.4 Lineare Integralgleichungen

Wir wollen nun ein größeres Beispiel behandeln um die bisher behandelten Konzepte zu illustrieren. Dies tun wir anhand der numerischen Lösung von Integralgleichungen.

Wir suchen eine unbekannte Funktion $u : (\alpha, \beta) \to \mathbb{R}$ die folgende Gleichung erfüllt

$$\lambda(x)u(x) + \int_\alpha^\beta K(x,y)u(y)\,dy = f(x) \qquad \forall x \in (\alpha,\beta). \tag{8.7}$$

Eine solche Gleichung nennt man eine Integralgleichung, da die unbekannte Funktion u unter einem Integral vorkommt (so wie man Differentialgleichung sagt, wenn Ableitungen von u in einer Gleichung für u vorkommen). Dabei sind $\lambda(x)$, $f(x)$ und $K(x,y)$ gegebene Funktionen auf dem Intervall (α,β). Integralgleichungen haben viele Anwendung, z.B. in der Mechanik aber auch in der Computergrafik. Dort berechnet man die Lichtintensität welche aus einer bestimmten Richtung von einer Szene beim Betrachter ankommt. Beim Ray-Tracing-Verfahren werden Lichtstrahlen auf der Oberfläche von Objekten nach der Regel „ Einfallswinkel gleich Ausfallswinkel“ reflektiert. Dies führt unter anderem zu sehr harten Schatten. Bei der Radiosity-Methode wird das aus einer gegebenen Richtung auf einen Punkt einer Oberfläche treffende Licht in alle Richtungen reflektiert, allerdings unterschiedlich stark. Wieviel Licht aus einer Richtung x in eine Richtung y reflektiert wird beschreibt die oben eingeführte Funktion $K(x,y)$, die sogenannte *Kernfunktion*, in der Integralgleichung.

Integralgleichungen kommen in diversen Varianten vor, einige davon sind:

a) Obige Integralgleichung nennt man linear, da der Ausdruck auf der linken Seite linear in der Funktion u ist. Eine nichtlineare Integralgleichung hat z.B. die Form $\int_\alpha^\beta K(x,y,u(y))\,dy = f(x)$.

b) Weiter heißt eine lineare Integralgleichung von „1. Art“ wenn $\lambda(x) = 0$.

c) Schließlich heißt eine Integralgleichung „Fredholmsch“ wenn, wie im obigen Fall, die Integralgrenzen fest sind. Man spricht von einer „Volteraschen“ Integralgleichung wenn der Integrationsbereich von x abhängt.

Im folgenden wollen wir die lineare Fredholmsche Integralgleichung 1. Art

$$\int_\alpha^\beta K(x,y)u(y)\,dy = f(x) \qquad \forall x \in (\alpha,\beta). \tag{8.8}$$

betrachten. Ein typisches Beispiel wäre die Kernfunktion

$$K(x,y) = \frac{1}{|x-y|^\gamma}, \qquad 0 < \gamma < 1.$$

Für den angegebenen Bereich des Parameters γ ist der Kern zwar singulär (dies ist typisch für Integralgleichungen) aber integrierbar. Man spricht von einer *schwach singulären* Kernfunktion.

Die Frage nach Existenz, Eindeutigkeit sowie weiterer Eigenschaften der Lösungen von Integralgleichungen sind Gegenstand der Funktionalanalysis. *An dieser Stelle ein kleiner Ausflug für Mathematiker:* Sei u eine Funktion aus einer gegebenen Menge U (z.B. ein Banachraum) von Funktionen. Dann definiert

$$I_u(x) = \int_\alpha^\beta K(x,y)u(y)\,dy$$

eine Funktion aus einer Menge V (z.B. ein anderer Banachraum). Zu jedem $u \in U$ gehört also eine Funktion $I_u \in V$. Diese Zuordnung $A : U \to V$, $A(u) = I_u$ ist, wie man nachrechnet, eine lineare Abbildung zwischen den Räumen U und V. In dieser abstrakten Formulierung lautet die Integralgleichung: Finde ein $u \in U$ so dass

$$A(u) = f,$$

also eine lineare Gleichung in Funktionenräumen.

Numerische Lösung mit der Kollokationsmethode

Wir betrachten nun ein einfache numerische Methode zur Lösung von Integralgleichungen, die sogenannte *Kollokationsmethode*.

Dazu unterteilen wir das endliche Intervall $(\alpha, \beta) \subset \mathbb{R}$ in n gleichgroße Teilintervalle der Länge $h = (\beta - \alpha)/n$ durch einführen der Punkte

$$x_i = \alpha + ih, \qquad 0 \le i \le n.$$

Zusätzlich bezeichnen wir die Mittelpunkte der Teilintervalle mit

$$c_i = \frac{x_i + x_{i+1}}{2} = \alpha + \left(i + \frac{1}{2}\right) h, \qquad 0 \le i < n.$$

Auf dieser Zerlegung des Intervalls (α, β) führen wir nun die folgenden n Funktionen ein

$$\varphi_i(x) = \begin{cases} 1 & x_i < x < x_{i+1} \\ 0 & \text{sonst} \end{cases}, \qquad 0 \le i < n,$$

und machen den Ansatz

$$u_h(x) = \sum_{j=0}^{n-1} z_j \varphi_j(x).$$

Die Funktion u_h ist ein Element der Menge der *stückweise konstante Funktion* auf der Zerlegung gegeben durch die x_i und ist festgelegt durch die n Koeffizenten z_i. Die Menge der stückweise Konstanten Funktionen ist ein n-dimensionaler Vektorraum der in eineindeutiger Weise dem $\mathbb{R}^n$ entspricht:

$$u_h = \sum_{j=0}^{n-1} z_j \varphi_j(x) \qquad \Leftrightarrow \qquad z_j = u_h(c_j).$$

Die Kollokationsmethode bestimmt nun die n unbekannten Koeffizienten z_i aus den n Gleichungen

$$\int_\alpha^\beta K(c_i, y) u_h(y)\, dy = f(c_i) \qquad 0 \le i < n. \tag{8.9}$$

Einsetzen der Darstellung von u_h in den Kollokationsansatz liefert dann

$$\begin{aligned} \int_\alpha^\beta K(c_i, y) u_h(y)\, dy &= \int_\alpha^\beta K(c_i, y) \left(\sum_{j=0}^{n-1} z_j \varphi_j(y) \right) dy \\ &= \sum_{j=0}^{n-1} z_j \int_{x_j}^{x_{j+1}} K(c_i, y)\, dy \\ &= \sum_{j=0}^{n-1} z_j \int_{c_j - h/2}^{c_j + h/2} K(c_i, y)\, dy = f(c_i). \end{aligned} \tag{8.10}$$

Wir erhalten demnach ein lineares Gleichungssystem

$$Az = b$$

für die unbekannten Koeffizienten $z = (z_0, \ldots, z_{n-1})^T$ (hier nummeriert ab 0!) mit den Einträgen

$$a_{i,j} = \int_{c_j - h/2}^{c_j + h/2} K(c_i, y)\, dy, \qquad\qquad b_i = f(c_i). \tag{8.11}$$

Eine weitere Methode ist die *Galerkinmethode* bei der man fordert

$$\int_\alpha^\beta \int_\alpha^\beta K(x, y) u_h(y)\, dy \varphi_i(x)\, dx = \int_\alpha^\beta f(c_i) \varphi_i(x)\, dx \qquad 0 \le i < n. \tag{8.12}$$

Auch in diesem Fall wird man ein lineares Gleichungssystem erhalten. Wir wollen nun ein spezielles Beispiel betrachten.

Beispiel 8.6. Wir betrachten die oben angegebene schwach singuläre Kernfunktion

$$K(x, y) = \frac{1}{|x - y|^\gamma}, \qquad 0 < \gamma < 1.$$

und erhalten damit für die Koeffizienten des linearen Gleichungssystemes:

$$a_{i,j} = \int_{c_j - h/2}^{c_j + h/2} \frac{1}{|c_i - y|^\gamma}\, dy = \begin{cases} \frac{2}{1-\gamma} \left(\frac{h}{2}\right)^{1-\gamma} & c_i = c_j \\ \frac{1}{1-\gamma} \left[\left(|c_i - c_j| + \frac{h}{2} \right)^{1-\gamma} - \left(|c_i - c_j| - \frac{h}{2} \right)^{1-\gamma} \right] & \text{sonst.} \end{cases} \tag{8.13}$$

Hier kann man das Integral über die Kernfunktion geschlossen angeben. Später werden wir auch numerische Integration kennenlernen. Speziell wählen wir nun das Einheitsintervall $(\alpha, \beta) = (0, 1)$ sowie die rechte Seite

$$f(x) = \frac{1}{1.1 + \sin(7\pi x)}.$$

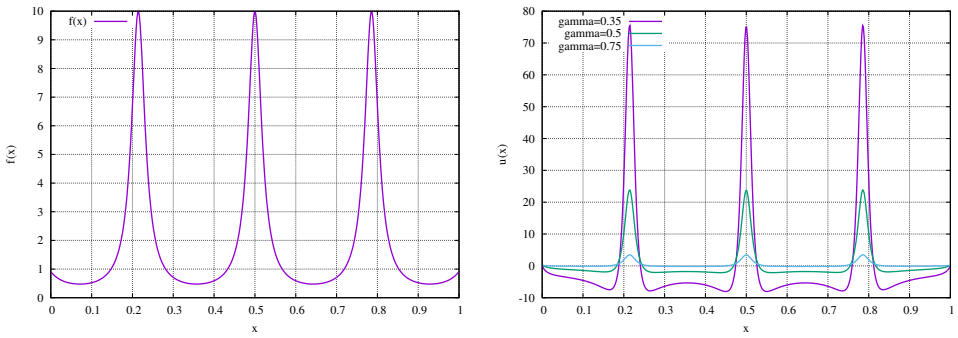


Abbildung 8.1: Rechte Seite $f(x)$ sowie numerisch berechnete Lösungen $u(x)$ zu den Parametern $\gamma \in \{0.35, 0.5, 0.75\}$.

Abbildung 8.1 zeigt diese Funktion sowie numerisch berechnete Lösungen $u(x)$ zu den Parametern $\gamma \in \{0.35, 0.5, 0.75\}$.

Die folgende Tabelle enthält Konditionszahlen $\text{cond}_\infty(A)$ für verschiedene Parameter γ und Gitterweiten $h = 1/n$. Diese Zahlen wurden mit sehr hoher Genauigkeit (mehr als hundert Nachkommstellen im Zehnersystem) berechnet.

γ/n	8	32	128	512
0.9	1.48	1.85	2.28	2.77
0.5	6.48	14.03	28.99	58.90
0.1	66.30	234.67	819.37	2855.07
1e-2	774.13	3076.99	12145.30	47915.50
1e-3	7862.13	31618.50	126349.00	504711.00
1e-4	78743.10	317047.00	1.26e+06	5.07e+06
1e-5	787553.00	3.17e+06	1.26e+07	5.07e+07

Wir beobachten, dass die Konditionszahl für $\gamma \to 1$ nur schwach mit n wächst und nahe bei 1 liegt. Für $\gamma \to 0$ werden die Konditionszahlen jedoch sehr groß und steigen linear mit n an. Schon für $n = 8$ liegt die Kondition nahe 10^6.

Wir wollen nun die Größe der Rundungsfehler in Abhängigkeit der Koinditionszahl untersuchen. Dazu bestimmen wir zu A die Inverse A^{-1} und berechnen die Fehlernorm $\|AA^{-1} - I\|_\infty$. Wegen $\|I\|_\infty = 1$ ist hier absoluter gleich relativem Fehler.

Bei Berechnung mit `double`-Genauigkeit erhalten wir die folgende Tabelle:

γ/n	8	32	128	512
0.9	4.95e-16	1.06e-15	2.58e-15	7.63e-15
0.5	6.62e-16	4.20e-15	2.61e-14	1.37e-13
0.1	8.02e-15	5.53e-14	7.69e-13	8.25e-12
1e-2	9.41e-14	1.52e-12	9.35e-12	2.00e-10
1e-3	8.81e-13	2.59e-11	1.85e-10	1.18e-09
1e-4	1.14e-11	1.68e-10	1.64e-09	3.19e-08
1e-5	8.73e-11	1.07e-09	2.28e-08	3.86e-07

Bei γ nahe 1 erhalten wir die volle Genauigkeit welche wir von `double` erwarten können. Bei $\gamma = 10^{-5}$ und $n = 512$ hingegen verlieren wir ungefähr die Hälfte der zur Verfügung stehenden Rechengenauigeit, was ungefähr dem Ergebnis aus dem Störungssatz 6.5 und den Überlegungen aus Beispiel 6.6 entspricht.

Führen wir die Berechnungen mit `float`-Genauigkeit durch dann erhalten wir die folgende Tabelle:

γ/n	8	32	128	512
0.9	1.92e-07	5.83e-07	2.07e-06	8.53e-06
0.5	4.24e-07	2.47e-06	1.49e-05	1.73e-04
0.1	5.54e-06	5.70e-05	4.16e-04	1.78e-02
1e-2	5.72e-05	1.33e-03	8.07e-03	4.08e-01
1e-3	4.88e-04	1.10e-02	2.22e-01	2.95e+00
1e-4	3.17e-03	1.60e-01	2.09e+00	1.58e+01
1e-5	4.68e-02	2.11e+00	1.33e+01	3.24e+02

Hier sieht man ein vergleichbares Verhalten. Im gut konditionierten Fall erhält man einen Fehler entsprechend der Genauigkeit von `float`. Für die schlecht konditionierten Probleme kann mit `float` kein zufriedenstellendes Ergebnis berechnet werden. □

Vorlesung 9

Rundungsfehleranalyse der LU-Zerlegung

Wir erinnern uns an die Analyse in Kapitel 3. Dort waren $F : \mathbb{R}^m \to \mathbb{R}^n$ und $F' : \mathbb{F}^m \to \mathbb{F}^n$ einander entsprechende Abbildungen in dem Sinne dass F' die Abbildung F durch $k \in \mathbb{N}$ arithmetische Grundoperationen in Fließkommaarithmetik realisiert. In der Rundungsfehleranalyse geht es um die Differenz $F(x) - F'(x)$ für $x \in \mathbb{F}^m$.

In der, z.B. in Beispiel 3.9 durchgeführten, sogenannten *Vorwärtsanalyse* waren wir in der Lage eine Beziehung der Form

$$F'(x) = F(x) + E(x, \epsilon_1, \ldots, \epsilon_k)$$

herzuleiten mit $\epsilon_1, \ldots, \epsilon_k$ den Rundungsfehlern aus den einzelnen Rechenoperationen. Für den Fehler gilt $E(x, \epsilon_1, \ldots, \epsilon_k) = 0$, wenn $\epsilon_1 = \ldots = \epsilon_k = 0$ weil ohne Rundungsfehler ja $F'(x) = F(x)$ gilt. In der Regel betrachten wir nur die führende Ordnung

$$E(x, \epsilon_1, \ldots, \epsilon_k) \doteq A(x) \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_k \end{bmatrix}$$

mit einer $n \times k$ Matrix A der Koeffizienten Ausdrücke in den Eingaben x sind.

In diesem Kapitel werden wir uns mit der sogenannten *Rückwärtsanalyse* beschäftigen. Dabei versucht man eine Beziehung der Form

$$F'(x) = F(x + H(x, \epsilon_1, \ldots, \epsilon_k)), \qquad H(x, \epsilon_1, \ldots, \epsilon_k) \doteq B(x) \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_k \end{bmatrix}$$

zu finden, d.h. die numerische Berechnung $F'(x)$ erscheint als Ergebnis der exakten Berechnung F mit modifizierten Eingaben $x + H(x, \epsilon_1, \ldots, \epsilon_k)$, wobei die Modifikationen in erster Näherung linear in den gemachten Rundungsfehlern sind. Also: Wie muss man die Eingabe der exakten Rechnung modifizieren um das selbe Ergebnis wie in der Fließkommaarithmetik zu erhalten. Daher rührt der Name „Rückwärtsanalyse“.

Dieses Kapitel folgt [Golub and Van Loan, 2013, Kapitel 3].

9.1 Notation

In diesem Kapitel betrachten wir die Rückwärtsanalyse für Probleme der linearen Algebra. Dafür ist es gut etwas Notation einzuführen umd die Resultate kompakter ausdrücken zu können.

Für Vektoren $x \in \mathbb{R}^n$ und Matrizen $A \in \mathbb{R}^{m \times n}$ vereinbaren wir die Absolutwertnotation:

$$|x| = \begin{bmatrix} |x_1| \\ \vdots \\ |x_n| \end{bmatrix}, \qquad |A| = \begin{bmatrix} |a_{1,1}| & \cdots & |a_{1,n}| \\ \vdots & & \vdots \\ |a_{m,1}| & \cdots & |a_{m,n}| \end{bmatrix}.$$

Beachte, dass der Absolutwert keine Norm ist sondern elementweise den Betrag nimmt!

Ebenso wird die Rundungsoperation $\mathrm{rd} : \mathbb{R}^{m\times n} \to \mathbb{F}^{m\times n}$ komponentenweise auf Vektoren und Matrizen erweitert:

$$\mathrm{rd}(x) = \begin{bmatrix} \mathrm{rd}(x_1) \\ \vdots \\ \mathrm{rd}(x_n) \end{bmatrix}, \qquad \mathrm{rd}(A) = \begin{bmatrix} \mathrm{rd}(a_{1,1}) & \cdots & \mathrm{rd}(a_{1,n}) \\ \vdots & & \vdots \\ \mathrm{rd}(a_{m,1}) & \cdots & \mathrm{rd}(a_{m,n}) \end{bmatrix}.$$

Und schließlich können wir auch Relationen wie $=$ oder $\leq$ auf alle Komponenten erweitern:

$$A \leq B \qquad \Leftrightarrow \qquad a_{i,j} \leq b_{i,j}, \quad 1 \leq i \leq m, 1 \leq j \leq n.$$

Beachte dass dies ein System von nm simultanen Ungleichungen darstellt!

Damit können wir etwa kurz schreiben:

$$\mathrm{rd}(A) = A + A' \qquad \text{mit} \qquad |A'| \leq |A| \text{ eps}.$$

Schließlich vereinbaren wir die fl-Notation. Diese überführt einen gegebenen Ausdruck bestehend aus arithmetischen Grundoperationen in einen entsprechenden Ausdruck mit Fließkommaoperationen. Mit den oben eingeführten Bezeichungen gilt also

$$F'(x) = \mathrm{fl}(F)(x).$$

Als ein Beispiel betrachte das Euklidsche Skalarprodukt zweier (reellwertiger) Vektoren $x \cdot y = x^T y = \sum_{i=1}^n x_i y_i$. Dann gilt

$$\mathrm{fl}(x \cdot y) = x_1 \odot y_1 \oplus \ldots \oplus x_n \odot y_n.$$

Im Prinzip kann diese Notation auch auf andere Funktionen erweitert werden, so wäre $\mathrm{fl}(\sqrt{x})$ eine Berechnung von $\sqrt{x}$ in Fließkommaarithmetik.

Kommen wir nun zurück zur Rückwärtsanalyse. Angewandt auf das Problem „Lineares Gleichungssystem Lösen“ würde die Vorwärtsanalyse eine Beziehung der Form

$$\hat{x} - x = E(A, b, \epsilon_1, \ldots, \epsilon_k)$$

herleiten, wobei $\hat{x} \in \mathbb{F}^n$ die Lösung des linearen Gleichungssystemes in Fließkommaarithmetik ist. Die Rückwärtsanalyse dagegen würde die Lösung darstellen als

$$(A + H(A, b, \epsilon_1, \ldots, \epsilon_k))\hat{x} = b$$

Dies hat den Vorteil, dass wir nun die Auswirkung der Rundungsfehler mittels des Störungssatzes 6.5 aus der Konditionsanalyse abschätzen können:

$$\frac{\|\hat{x} - x\|}{\|x\|} \leq \frac{\mathrm{cond}(A)}{1 - \mathrm{cond}(A)\frac{\|H\|}{\|A\|}} \frac{\|H\|}{\|A\|}.$$

9.2 Dreieckssysteme

Wir beginnen mit einem Hilfssatz.

Lemma 9.1. Für gegebene Vektoren $x, y \in \mathbb{F}^n$ gilt für das Skalarprodukt

$$\mathrm{fl}(x \cdot y) = (x + f) \cdot y, \qquad |f| \leq n \text{ eps } |x| + O(\text{eps}^2).$$

Beweis. Wir beweisen mittels Induktion.
$\underline{n = 1.}$ Wegen exakt gerundeter Fließkommaarithmetik gilt $\mathrm{fl}(x \cdot y) = \mathrm{fl}(x_1 y_1) = x_1 y_1 (1 + \epsilon_1) = (x_1 + x_1 \epsilon_1) y_1$, also $|f_1| = |\epsilon_1||x_1| \leq \text{ eps } |x_1|$.
$\underline{n \geq 2.}$ Die Behauptung gelte schon für Vektoren der Länge $n-1$. Es seien also $x, y \in \mathbb{F}^n$ und $\tilde{x}, \tilde{y} \in \mathbb{F}^{n-1}$ mit $\tilde{x}_i = x_i$, $\tilde{y}_i = y_i$. Dann gilt

$$\begin{aligned}
\mathrm{fl}(x \cdot y) &= \mathrm{fl}(\tilde{x} \cdot \tilde{y} + x_n y_n) \\
&= (\mathrm{fl}(\tilde{x} \cdot \tilde{y}) + \mathrm{fl}(x_n y_n))(1 + \epsilon_n) \\
&= ((\tilde{x} + \tilde{f}) \cdot \tilde{y} + x_n y_n (1 + \delta_n))(1 + \epsilon_n) \\
&= (\tilde{x} + \tilde{f} + \epsilon_n \tilde{x} + \epsilon_n \tilde{f}) \cdot \tilde{y} + (x_n + (\epsilon_n + \delta_n) x_n + \epsilon_n \delta_n x_n) y_n \\
&= (x + f) \cdot y
\end{aligned}$$

mit

$$f = \begin{bmatrix} \tilde{f} + \epsilon_n \tilde{x} + \epsilon_n \tilde{f} \\ (\epsilon_n + \delta_n) x_n + \epsilon_n \delta_n x_n \end{bmatrix}.$$

Für die Abschätzung gilt

$$|\tilde{f} + \epsilon_n \tilde{x} + \epsilon_n \tilde{f}| \leq (n-1) \text{ eps } |\tilde{x}| + \text{ eps } |\tilde{x}| + O(\mathrm{eps}^2) = n \text{ eps } |\tilde{x}| + O(\mathrm{eps}^2)$$

und

$$|(\epsilon_n + \delta_n) x_n + \epsilon_n \delta_n x_n| \leq 2 \text{ eps } |x_n| + O(\mathrm{eps}^2).$$

Wegen $n \geq 2$ gilt somit $|f| \leq n \text{ eps } |x| + O(\mathrm{eps}^2)$ wie behauptet. □

Dieses Lemma zeigt, dass sich die Rundungsfehler im Skalarprodukt im schlimmsten Fall alle addieren. Dies ist allerdings sehr pessimistisch.

Satz 9.2. Es seien $\hat{x} \in \mathbb{F}^n$ bzw. $\hat{y} \in \mathbb{F}^n$ die numerischen Lösungen der unteren bzw. oberen Dreieckssysteme $Lx = b$ bzw. $Uy = c$ in Fließkommaarithmetik. Dann gibt es Matrizen F bzw. G so dass gilt

$$\begin{aligned}
(L + F)\hat{x} &= b, & |F| &\leq n \text{ eps } |L| + O(\mathrm{eps}^2), \\
(U + G)\hat{y} &= c, & |G| &\leq n \text{ eps } |U| + O(\mathrm{eps}^2).
\end{aligned}$$

Beweis. Wir beweisen mittels Induktion über die Größe der Matrizen n. Dabei betrachten wir nur das vorwärts einsetzen, also L. Der Beweis für das System mit U geht analog.
$\underline{n = 1.}$ In diesem Fall gilt $l_{1,1} x_1 = b_1$, also $\hat{x}_1 = \mathrm{fl}(b_1 / l_{1,1}) = (b_1 / l_{1,1})(1 + \epsilon_1)$. Dies können wir nun mittels Taylorentwicklung schreiben als

$$\begin{aligned}
& & \frac{l_{1,1}}{1 + \epsilon_1} \hat{x}_1 &= b_1 \\
\Leftrightarrow & & \left(1 - \epsilon_1 + R(\epsilon_1^2)\right) l_{1,1} \hat{x}_1 &= b_1 \\
\Leftrightarrow & & \left(l_{1,1} - \epsilon_1 l_{1,1} + l_{1,1} R(\epsilon_1^2)\right) \hat{x}_1 &= b_1.
\end{aligned}$$

Somit gilt die Darstellung mit $f_1 = -\epsilon_1 l_{1,1} + l_{1,1} R(\epsilon_1^2)$ und damit

$$|f_1| \leq \text{ eps } |l_{1,1}| + O(\mathrm{eps}^2).$$

$\underline{n \geq 2.}$ Wir schreiben das untere Dreieckssystem in Blockform:

$$\begin{bmatrix} L_1 & 0 \\ v^T & \alpha \end{bmatrix} \begin{bmatrix} x_1 \\ w \end{bmatrix} = \begin{bmatrix} b_1 \\ \beta \end{bmatrix}$$

mit der $n-1 \times n-1$-Matrix L_1 sowie den $n-1$-Vektoren v, x_1 und b_1. Sei nun $\hat{x}_1$ die numerische Lösung von $L_1 x_1 = b_1$ für die die Induktionsvoraussetzung $(L_1 + F_1)\hat{x}_1 = b_1$ gilt. Dann berechnen wir $\hat{w}$ als

$$\begin{aligned}\hat{w} &= \mathrm{fl}((\beta - v^T\hat{x}_1)/\alpha) \\ &= (\mathrm{fl}(\beta - v^T\hat{x}_1)/\alpha)(1+\epsilon_n) \\ &= ((\beta - \mathrm{fl}(v^T\hat{x}_1))(1+\delta_n)/\alpha)(1+\epsilon_n) \\ &= ((\beta - (v+f)^T\hat{x}_1)/\alpha)(1+\delta_n+\epsilon_n+\delta_n\epsilon_n).\end{aligned}$$

wobei wir im letzten Schritt den Hilfssatz 9.1 benutzt haben. Wie oben benutzen wir die Taylorentwicklung

$$\frac{1}{1+\delta_n+\epsilon_n+\delta_n\epsilon_n} = 1 - (\delta_n+\epsilon_n) + R((\delta_n+\epsilon_n+\delta_n\epsilon_n)^2)$$

und erhalten

$$\begin{aligned}&\left(1 - (\delta_n+\epsilon_n) + R((\delta_n+\epsilon_n+\delta_n\epsilon_n)^2)\right)\hat{w} = (\beta - (v+f)^T\hat{x}_1)/\alpha \\ \Leftrightarrow\ &(v+f)^T\hat{x}_1 + \left(1 - (\delta_n+\epsilon_n) + R((\delta_n+\epsilon_n+\delta_n\epsilon_n)^2)\right)\alpha\hat{w} = \beta.\end{aligned}$$

Mit Hilfe der Induktionsvoraussetzung erhalten wir dann

$$\begin{bmatrix} L_1 + F_1 & 0 \\ (v+f)^T & \left(1 - (\delta_n+\epsilon_n) + R((\delta_n+\epsilon_n+\delta_n\epsilon_n)^2)\right)\alpha \end{bmatrix}\begin{bmatrix}\hat{x}_1 \\ \hat{w}\end{bmatrix} = \begin{bmatrix}\hat{b}_1 \\ \beta\end{bmatrix}$$

und daraus identifizieren wir

$$F = \begin{bmatrix} F_1 & 0 \\ f^T & -(\delta_n+\epsilon_n) + R((\delta_n+\epsilon_n+\delta_n\epsilon_n)^2)\alpha \end{bmatrix}.$$

Für die Abschätzung erhalten wir nach Induktionsvoraussetzung

$$|F_1| \leq (n-1)\ \mathrm{eps}\ |L_1| + O(\mathrm{eps}^2)$$

und für die beiden anderen Teile bekommen wir

$$\begin{aligned}|f_1|^T &\leq (n-1)\ \mathrm{eps}\ |v|^T + O(\mathrm{eps}^2) \\ |-(\delta_n+\epsilon_n) + R((\delta_n+\epsilon_n+\delta_n\epsilon_n)^2)\alpha| &\leq 2\ \mathrm{eps}\ |\alpha| + O(\mathrm{eps}^2)\end{aligned}$$

und somit insgesamt wegen $n \geq 2$ dann $|F| \leq n\ \mathrm{eps}\ |L| + O(\mathrm{eps}^2)$ wie behauptet. □

9.3 LU-Zerlegung und Lösen eines LGS

Nach diesen Vorarbeiten können wir uns nun dem Hauptresultat zuwenden.

Satz 9.3. Es sei $A \in \mathbb{F}^{n\times n}$ regulär. Wir nehmen an die exakte LU-Zerlegung $A = LU$ sei *ohne* Zeilentausch möglich. Dann gilt für die numerisch berechneten Faktoren $\hat{L}$ und $\hat{U}$ die Darstellung

$$A + H = \hat{L}\hat{U} \qquad \text{mit} \qquad |H| \leq 3(n-1)\ \mathrm{eps}\ \left(|A| + |\hat{L}||\hat{U}|\right) + O(\mathrm{eps}^2)$$

Beweis. Wieder beweisen wir mittels Induktion über n.
$\underline{n = 1.}$ In diesem Fall gilt $l_{1,1} = 1$ und damit auch in Fließkommaarithmetik $1 \odot u_{1,1} = a_{1,1}$, also $H = 0$.
$\underline{n \geq 2.}$ Wir schreiben A in 2×2 Blockform:

$$A = \begin{bmatrix} \alpha & w^T \\ v & B \end{bmatrix}.$$

Dann geht die LU-Zerlegung folgendermaßen vor:

a) Berechne den $n-1$-Vektor $\hat{z} = \mathrm{fl}(v/\alpha)$.

b) Berechne die Modifikation als Rang-1-Update: $\hat{A}_1 = \mathrm{fl}(B - \hat{z}w^T)$

c) Berechne *rekursiv* die LU-Zerlegung von $\hat{A}_1$, hier können wir die Induktionsvoraussetzung nutzen.

Diese Schritte arbeiten wir nun ab

a) $\hat{z} = \mathrm{fl}(v/\alpha) = v/\alpha + f$ mit $|f| \leq \mathrm{eps}\ |v|/|\alpha|$.

b)

$$\begin{aligned}
\hat{A}_1 &= \mathrm{fl}(B - \hat{z}w^T) \\
&= B - \mathrm{fl}(\hat{z}w^T) + G && |G| \leq \mathrm{eps}\ |B - \mathrm{fl}(\hat{z}w^T)| \\
&= B - (\hat{z}w^T + G') + G && |G'| \leq \mathrm{eps}\ |\hat{z}w^T| \leq \mathrm{eps}\ |\hat{z}||w^T| \\
&= B - \hat{z}w^T + F && |F| = |-G' + G| \leq |G'| + |G| \\
& && \leq \mathrm{eps}\ |\hat{z}||w^T| + \mathrm{eps}\ |B - \mathrm{fl}(\hat{z}w^T)| \\
& && \leq 2\ \mathrm{eps}\ \left(|B| + |\hat{z}||w^T|\right) + O(\mathrm{eps}^2).
\end{aligned}$$

c) Nun wird $\hat{A}_1$ LU-zerlegt und es gilt die Induktionsannahme

$$\hat{A}_1 + H_1 = \hat{L}_1\hat{U}_1 \qquad \text{mit} \qquad |H_1| \leq 3(n-2)\ \mathrm{eps}\ \left(|\hat{A}_1| + |\hat{L}_1||\hat{U}_1|\right) + O(\mathrm{eps}^2).$$

Die LU-Zerlegung, welche der numerische Algorithmus zurückliefert, lautet dann in Blockform

$$\begin{aligned}
\hat{L}\hat{U} &= \begin{bmatrix} 1 & 0 \\ \hat{z} & \hat{L}_1 \end{bmatrix} \begin{bmatrix} \alpha & w^T \\ 0 & \hat{U}_1 \end{bmatrix} = \begin{bmatrix} \alpha & w^T \\ \alpha\hat{z} & \hat{z}w^T + \hat{L}_1\hat{U}_1 \end{bmatrix} \\
&= \begin{bmatrix} \alpha & w^T \\ \alpha\left(\frac{v}{\alpha} + f\right) & \hat{z}w^T + \hat{A}_1 + H_1 \end{bmatrix} \\
&= \begin{bmatrix} \alpha & w^T \\ v + \alpha f & \hat{z}w^T + B - \hat{z}w^T + F + H_1 \end{bmatrix} \\
&= \begin{bmatrix} \alpha & w^T \\ v & B \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \alpha f & F + H_1 \end{bmatrix} = A + H.
\end{aligned}$$

Nun ist noch $|H|$ abzuschätzen. Dazu rechnen wir mit b) von oben

$$\begin{aligned}
|\hat{A}_1| &= |B - \hat{z}w^T + F| \leq |B| + |\hat{z}||w^T| + |F| \\
&\leq |B| + |\hat{z}||w^T| + 2\ \mathrm{eps}\ \left(|B| + |\hat{z}||w^T|\right) + O(\mathrm{eps}^2) \\
&= (1 + 2\ \mathrm{eps})\left(|B| + |\hat{z}||w^T|\right) + O(\mathrm{eps}^2)
\end{aligned}$$

und

$$\begin{aligned}
|F + H_1| &\leq |F| + |H_1| \\
&\leq 2 \text{ eps} \left(|B| + |\hat{z}||w^T|\right) + 3(n-2) \text{ eps} \left(|\hat{A}_1| + |\hat{L}_1||\hat{U}_1|\right) + O(\text{eps}^2) \\
&\leq 2 \text{ eps} \left(|B| + |\hat{z}||w^T|\right) \\
&\quad + 3(n-2) \text{ eps} \left((1 + 2 \text{ eps}) \left(|B| + |\hat{z}||w^T|\right) + |\hat{L}_1||\hat{U}_1|\right) + O(\text{eps}^2) \\
&\leq 3(n-1) \text{ eps} \left(|B| + |\hat{z}||w^T| + |\hat{L}_1||\hat{U}_1|\right) + O(\text{eps}^2)
\end{aligned}$$

da $3(n-2) + 2 = 3n - 4 \leq 3(n-1)$. Damit nun zum letzten Streich:

$$\begin{aligned}
|H| &= \begin{bmatrix} 0 & 0 \\ |\alpha f| & |F + H_1| \end{bmatrix} \\
&\leq \begin{bmatrix} 0 & 0 \\ \text{eps}|v| & 3(n-1) \text{ eps} \left(|B| + |\hat{z}||w^T| + |\hat{L}_1||\hat{U}_1|\right) \end{bmatrix} + O(\text{eps}^2) \\
&\leq 3(n-1) \text{ eps} \begin{bmatrix} 0 & 0 \\ |v| & \left(|B| + |\hat{z}||w^T| + |\hat{L}_1||\hat{U}_1|\right) \end{bmatrix} + O(\text{eps}^2) \\
&\leq 3(n-1) \text{ eps} \left(\begin{bmatrix} |\alpha| & |w^T| \\ |v| & |B| \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ |\hat{z}| & |\hat{L}_1| \end{bmatrix} \begin{bmatrix} |\alpha| & |w|^T \\ 0 & |\hat{U}_1| \end{bmatrix} \right) + O(\text{eps}^2) \\
&= 3(n-1) \text{ eps} \left(|A| + |\hat{L}||\hat{U}|\right) + O(\text{eps}^2).
\end{aligned}$$

wobei einige Terme hinzugefügt wurden.

□

Nun können wir alle Resultate zusammen nutzen um den Rundungsfehler bei der Lösung eines linearen Gleichungssystemes abzuschätzen.

Satz 9.4. $A \in \mathbb{F}^{n \times n}$ und $b \in \mathbb{F}^n$ seien gegebene Matrix und rechte Seite eines linearen Gleichungssystemes. $\hat{L}$ und $\hat{U}$ seien die in Fließkommaarithmetik berechneten Faktoren in der LU-Zerlegung von A. Weiter sei $\hat{y}$ die in Fließkommaarithmetik berechnete Lösung von $\hat{L}y = b$ und $\hat{x}$ die in Fließkommaarithmetik berechnete Lösung von $\hat{U}x = \hat{y}$. Dann gilt für $\hat{x}$:

$$(A + E)\hat{x} = b, \qquad |E| \leq n \text{ eps } \left(3|A| + 5|\hat{L}||\hat{U}|\right) + O(\text{eps}^2).$$

Beweis. Nach Satz 9.2 gilt

$$\begin{aligned}
(\hat{L} + F)\hat{y} &= b, & |F| &\leq n \text{ eps } |\hat{L}| + O(\text{eps}^2), \\
(\hat{U} + G)\hat{x} &= \hat{y}, & |G| &\leq n \text{ eps } |\hat{U}| + O(\text{eps}^2),
\end{aligned}$$

also

$$(\hat{L} + F)(\hat{U} + G)\hat{x} = (\hat{L}\hat{U} + F\hat{U} + \hat{L}G + FG)\hat{x} = b.$$

Nach Satz 9.3 gilt für die numerisch berechnete LU-Zerlegung von A

$$A + H = \hat{L}\hat{U} \qquad |H| \leq 3(n-1) \text{ eps } \left(|A| + |\hat{L}||\hat{U}|\right) + O(\text{eps}^2)$$

und somit

$$(A+E)\hat{x} = b \qquad\qquad E = H + F\hat{U} + \hat{L}G + FG.$$

Bleibt noch $|E|$ abzuschätzen:

$$\begin{aligned}
|E| &= |H + F\hat{U} + \hat{L}G + FG| \\
&\leq |H| + |F||\hat{U}| + |\hat{L}||G| + O(\text{eps}^2) \\
&\leq 3(n-1)\ \text{eps}\ \left(|A| + |\hat{L}||\hat{U}|\right) + n\ \text{eps}\ |\hat{L}||\hat{U}| + |\hat{L}|n\ \text{eps}\ |\hat{U}| + O(\text{eps}^2) \\
&\leq n\ \text{eps}\ \left(3|A| + 5|\hat{L}||\hat{U}|\right) + O(\text{eps}^2).
\end{aligned}$$

□

9.4 Pivotisierung

Wir wollen uns nun der Frage zuwenden in wie weit die Gaußelimination ein numerisch stabiles Verfahren zur Lösung eines linearen Gleichungssystemes ist. Dazu erinnern wir uns: ein Verfahren heißt numerisch stabil falls die Verstärkungsfaktoren aus der Rundungsfehleranalyse die Verstärkungsfaktoren aus der Konditionsanalyse nicht übersteigen.

Fassen wir die Situation nochmal zusammen:

1) $A \in \mathbb{F}^{n\times n}$ und $b \in \mathbb{F}^n$ sind eine gegebene Matrix und eine rechte Seite in bereits in Fließkommazahlen. Die exakte Lösung bezeichnen wir $x \in \mathbb{R}^n$, also $Ax = b$.

2) Die numerisch bestimmte, also mit Rundungsfehlern behaftete, Lösung $\hat{x}$ des Gleichungssystemes bezeichnen wir mit $\hat{x} \in \mathbb{F}^n$. Für diese gilt nach Satz 9.4 dann $(A+E)\hat{x} = b$ mit $|E| \leq n\ \text{eps}\ \left(3|A| + 5|\hat{L}||\hat{U}|\right) + O(\text{eps}^2)$.

3) Nun betrachten wir A und b zusätzlich als *gerundete* Varianten von $\check{A} \in \mathbb{R}^{n\times n}$ und $\check{b} \in \mathbb{R}^n$. Damit gilt $A = \check{A} - \Delta A$ und $b = \check{b} - \Delta b$ mit $|\Delta A| \leq \text{eps}\ A$ und $|\Delta b| \leq \text{eps}\ b$. Mit $\check{x}$ bezeichnen wir die exakte Lösung des ungerundeten Systems, also $\check{A}\check{x} = \check{b} \Leftrightarrow (A + \Delta A)\check{x} = b + \Delta b$.

In der nun folgenden Analyse benutzen wir die Norm $\|.\|_\infty$ für Vektoren und die zugeordnete Matrixnorm. Für diese (und auch die 1-Norm) gelten $\|A\|_\infty = \||A|\|_\infty$ bzw. $\|b\|_\infty = \||b|\|_\infty$.

Bei der numerischen Stabilität müssen wir Konditions- und Rundungsfehleranalyse vergleichen. Betrachten wir daher zunächst die Konditionsanalyse, d.h. welchen Einfluss die Rundungsfehler bei der Eingabe auf die Lösung haben. Mit dem Störungssatz 6.5 und den oben getroffenen Vereinbarungen gilt

$$\begin{aligned}
\frac{\|\check{x} - x\|_\infty}{\|x\|_\infty} &\leq \frac{\text{cond}_\infty(A)}{1 - \text{cond}_\infty(A)\frac{\|\Delta A\|_\infty}{\|A\|_\infty}} \left(\frac{\|\Delta A\|_\infty}{\|A\|_\infty} + \frac{\|\Delta b\|_\infty}{\|b\|_\infty}\right) \\
&\leq \frac{\text{cond}_\infty(A)}{1 - \text{cond}_\infty(A)\frac{\|\Delta A\|_\infty}{\|A\|_\infty}} 2\ \text{eps}\ .
\end{aligned} \tag{9.1}$$

Für die bei der numerischen Lösung entstehenden Rundungsfehler erhalten wir mit Rückwärtsanalyse und Störungssatz:

$$\begin{aligned}\frac{\|\hat{x}-x\|_\infty}{\|x\|_\infty} &\leq \frac{\operatorname{cond}_\infty(A)}{1-\operatorname{cond}_\infty(A)\frac{\|E\|_\infty}{\|A\|_\infty}}\frac{\|E\|_\infty}{\|A\|_\infty} \\ &\leq \frac{\operatorname{cond}_\infty(A)}{1-\operatorname{cond}_\infty(A)\frac{\|E\|_\infty}{\|A\|_\infty}}\ n\ \text{eps}\ \left(3\frac{\|A\|_\infty}{\|A\|_\infty}+5\frac{\|\hat{L}\|_\infty\|\hat{U}\|_\infty}{\|A\|_\infty}+O(\text{eps}^2)\right).\end{aligned}$$

Wenn wir annehmen, dass $\|E\|_\infty \ll \|A\|_\infty$ dann ist der erste Faktor ungefähr $\operatorname{cond}_\infty(A)$. Dann fällt im Vergleich zur Abschätzung (9.1) der Faktor n auf und schließlich ist die zentrale Frage wie sich die Normen $\|\hat{L}\|_\infty$ und $\|\hat{U}\|_\infty$ verhalten.

Wir beobachten, dass die Einträge von $\hat{L}$ durch die Wahl des Pivotelementes beeinflusst werden können.

Definition 9.5. Die Wahl des Pivotelementes $\tilde{a}_{k,k}^{(k)}$ als betragsmäßig Größtes Element der Spalte k im Schritt 3a der LU-Zerlegung, also

$$\tilde{a}_{k,k}^{(k)} \geq \tilde{a}_{i,k}^{(k)}, \qquad\qquad k \leq i \leq n,$$

bezeichnet man als *Spaltenpivotisierung*. Hingegen bezeichnet man die Wahl

$$\tilde{a}_{k,k}^{(k)} \geq \tilde{a}_{i,j}^{(k)}, \qquad\qquad k \leq i,j \leq n,$$

als *Totalpivotisierung*. Für Spaltenpivotisierung sind nur Zeilentausche notwendig, während für die Totalpivotisierung Zeilen- und Spaltentausche notwendig sind. □

Die Rückwärtsanalyse kann auf die LU-Zerlegung mit Pivotisierung erweitert werden, siehe Golub and Van Loan [2013]. Mittels beider Arten der Pivotisierung erreicht man $l_{ik} = \tilde{a}_{i,k}^{(k)}/\tilde{a}_{k,k}^{(k)} \leq 1$ und mithin $\|\hat{L}\|_\infty \leq n$. Damit erhalten wir dann in Fortführung von oben:

$$\begin{aligned}\frac{\|\hat{x}-x\|_\infty}{\|x\|_\infty} &\leq \frac{\operatorname{cond}_\infty(A)}{1-\operatorname{cond}_\infty(A)\frac{\|E\|_\infty}{\|A\|_\infty}}\ n\ \text{eps}\ \left(3+5n\frac{\|\hat{U}\|_\infty}{\|A\|_\infty}+O(\text{eps}^2)\right) \\ &\lesssim \frac{\operatorname{cond}_\infty(A)}{1-\operatorname{cond}_\infty(A)\frac{\|E\|_\infty}{\|A\|_\infty}}\ 5n^2\ \text{eps}\ \frac{\|\hat{U}\|_\infty}{\|A\|_\infty}.\end{aligned} \tag{9.2}$$

Nun stellt sich die Frage wie sich der Faktor $\|\hat{U}\|_\infty/\|A\|_\infty$ verhält. Im Fall der Spaltenpivotisierung kann man Beispiele angeben in denen $\|\hat{U}\|_\infty/\|A\|_\infty = O(2^n)$ gilt, siehe [Golub and Van Loan, 2013, Kapitel 3.4.5] bzw. Übung. Für die Totalpivotisierung kann eine deutlich kleinere obere Schranke für die Einträge von U gezeigt werden [Golub and Van Loan, 2013, Gl. 3.4.10]:

$$|a_{i,j}^{(k)}| \leq k^{\frac{1}{2}}\left(2\cdot 3^{\frac{1}{2}}\cdots k^{\frac{1}{k-1}}\right)^{\frac{1}{2}}\max|a_{i,j}|$$

Ist die Gaußelimination (bzw. LU-Zerlegung) mit Pivotisierung nun ein numerisch stabiler Algorithmus? Streng genommen nein, da es Beispiele gibt in denen, insbesondere bei Spaltenpivotisierung, der Faktor $\|\hat{U}\|_\infty/\|A\|_\infty$ stark anwachsen kann. Allerdings ist das in der Praxis extrem selten. Golub und van Loan schreiben in [Golub and Van Loan, 2013, S. 131]:

> *Although there is still more to understand about ρ [i.e. $\|\hat{U}\|_\infty/\|A\|_\infty$], the consensus is that serious element growth in Gaussian elimination with partial pivoting [Spaltenpivotisierung] is extremely rare. The method can be used with confidence.*

Beispiel 9.6. Zur Illustration der Abschätzung (9.2) betrachten wir noch einmal die Systeme aus der Lösung von linearen Integralgleichungen aus Beispiel 8.6. Dort können wir die Konditionierung des Systems durch die Parameter γ und n nahezu beliebig einstellen und dadurch versuchen den Einfluss der Konditionszahl und der Systemgröße zu trennen. Das System wird numerisch mit der Genauigkeit `double` erstellt und gelöst, d.h. es gilt eps $= 2^{-53} \approx 1,11 \cdot 10^{-16}$. Mittels Berechnung mit Mantissenlänge 1024 können wir sowohl Rundungsfehler als auch Konditionsfehler bestimmen, letzeren in dem wir die Einträge von A und b zufällig stören.

Bei konstantem $n = 20$ und steigender Kondition erhalten wir folgende Tabelle:

n	γ	cond_∞	$\frac{\|x-\hat{x}\|_\infty}{\|x\|_\infty}$	$\frac{\|x-\check{x}\|_\infty}{\|x\|_\infty}$	$\frac{\|\hat{U}\|_\infty}{\|A\|_\infty}$
20	1.575e-1	84.06	5.33e-16	3.89e-16	0.917
20	1.575e-2	1207.76	4.25e-15	5.57e-15	0.991
20	1.575e-3	12525.20	6.52e-14	5.96e-14	0.999
20	1.575e-4	125709.00	6.02e-13	6.00e-13	0.999
20	1.575e-5	1.25e+06	3.48e-12	6.00e-12	0.999
20	1.575e-6	1.25e+07	6.26e-11	6.00e-11	0.999

Rundungsfehler (4. Spalte) und Konditionsfehler (5. Spalte) verhalten sich sehr ähnlich und sind nahezu identisch. Es findet praktisch kein Wachstum des U Faktors relativ zu A statt.

Nun variieren wir n und passen γ so an, dass die Kondition konstant auf 10^7 gehalten wird:

n	γ	cond_∞	$\frac{\|x-\hat{x}\|_\infty}{\|x\|_\infty}$	$\frac{\|x-\check{x}\|_\infty}{\|x\|_\infty}$	$\frac{\|\hat{U}\|_\infty}{\|A\|_\infty}$
8	7.87e-07	1.000e+07	6.21e-11	3.76e-10	1.000
16	1.57e-06	1.005e+07	3.26e-11	1.23e-10	0.999
32	3.15e-06	1.006e+07	1.84e-11	6.18e-11	0.999
64	6.30e-06	1.007e+07	1.46e-11	4.33e-11	0.999
128	1.26e-05	1.007e+07	8.65e-12	2.94e-11	0.999
256	2.52e-05	1.007e+07	6.96e-12	2.12e-11	0.999
512	5.04e-05	1.006e+07	5.28e-12	1.63e-11	0.999
1024	1.00e-04	1.006e+07	4.03e-12	1.39e-11	0.999

Wir beobachten keinerlei Abhängigkeit beider Fehler von der Systemgröße und wiederum kein Wachstum des U Faktors relativ zu A. Zumindest in diesem Beispiel erweist sich die n^2 Abhängigkeit in Abschätzung (9.2) als sehr pssimistisch. □

Beispiel 9.7. Aus Golub and Ortega [1996].

a) Das folgende Beispiel demonstriert eine extreme Fortpflanzung eines einzigen Rundungsfehlers ohne Spaltenpivotisierung. Wir betrachten das 2×2 System

$$\begin{bmatrix} -10^{-5} & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

In exakter Arithmetik führt das Gaußsche Verfahren nach Elimination von a_{21} auf

$$\begin{bmatrix} -10^{-5} & 1 \\ 0 & 1+2\cdot 10^5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2\cdot 10^5 \end{bmatrix}$$

mit der Lösung

$$x_1 = -0.4999975, \qquad x_2 = 0.999995 \quad .$$

Nun führen wir das Verfahren in $\mathbb{F}(10,4,1)$ durch. Beim Multiplikator

$$q_{21} = (0.2\cdot 10^1) \oslash (-0.1\cdot 10^{-4}) = -0.2\cdot 10^6$$

ergibt sich kein Rundungsfehler. Für das neue a_{22} ergibt sich

$$\begin{aligned} a_{22}^{(1)} &= 0.1\cdot 10^1 \ominus (-0.2\cdot 10^6) \odot (0.1\cdot 10^1) \\ &= 0.1\cdot 10^1 \oplus 0.2\cdot 10^6 = \boxed{0.2\cdot 10^6}. \end{aligned}$$

Hier wurde auf vier Stellen gerundet. Damit ergibt sich (ohne Fehler)

$$b_2^{(1)} = -(-0.2\cdot 10^6) \odot (0.1\cdot 10^1) = 0.2\cdot 10^6$$

und

$$\begin{aligned} x_2 &= b_2^{(1)} \oslash a_{22}^{(1)} = 0.2\cdot 10^6 \oslash 0.2\cdot 10^6 = \boxed{1}, \\ x_1 &= (0.1\cdot 10^1 \ominus 0.1\cdot 10^1 \odot 1) \oslash (-0.1\cdot 10^{-4}) = \boxed{0} \quad . \end{aligned}$$

Es ist also *keine* Stelle im Ergebnis korrekt obwohl nur an einer *einzigen* Stelle (in der Berechnung von $a_{22}^{(1)}$) ein Rundungsfehler eingeführt wurde.

Im Prinzip haben wir in Fließkommaarithmetik das System

$$\begin{bmatrix} -10^{-5} & 1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

exakt gelöst, was eine völlig andere Lösung hat als das ursprüngliche System (Rückwärtsanalyse).

Der große Multiplikator kann ganz einfach vermieden werden indem man eine Zeilenvertauschung durchführt, d. h. wir lösen

$$\begin{bmatrix} 2 & 1 \\ -10^{-5} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Nun erhält man

$$\begin{aligned} q_{21} &= -0.1\cdot 10^{-4} \oslash 0.2\cdot 10^1 = -0.5\cdot 10^{-5}, \\ a_{22}^{(1)} &= 0.1\cdot 10^1 \ominus (-0.5\cdot 10^{-5}) \odot 0.1\cdot 10^1 = 0.1\cdot 10^1 \oplus 0.5\cdot 10^{-5} = 0.1\cdot 10^1, \\ b_2^{(1)} &= 0.1\cdot 10^1 \ominus 0.5\cdot 10^{-5} \odot 0 = 0.1\cdot 10^1, \\ x_2 &= 0.1\cdot 10^1 \oslash 0.1\cdot 10^1 = \boxed{1}, \\ x_1 &= (0 \ominus 0.1\cdot 10^1 \odot 0.1\cdot 10^1) \oslash 0.2\cdot 10^1 = \boxed{-0.5}, \end{aligned}$$

was in $\mathbb{F}(10,4,1)$ völlig in Ordnung ist.

b) Wir betrachten das 2×2 System

$$\begin{bmatrix} 10 & -10^6 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -10^6 \\ 0 \end{bmatrix}.$$

welches aus (a) durch Multiplikation der ersten Zeile mit -10^6 entsteht.

Die Spaltenpivotisierung erfordert keine Vertauschung. Allerdings entsteht für $a_{22}^{(1)} = 1 + 2 \cdot 10^5$ genau dasselbe Problem wie oben!

Wir sehen also dass (Spalten-)pivotisierung durchaus eine Verbesserung erzielen kann aber nicht muss. □

Das Rundungsfehlerverhalten der Gaußelimination im Teil b) des letzten Beispiels kann durch eine *Zeilenäquilibrierung* verbessert werden, siehe [Rannacher, 2017, S. 115]. Dazu multipliziert man das lineare Gleichungssystem mit einer Diagonalmatrix von links:

$$Ax = b \quad \rightarrow \quad DAx = Db, \qquad d_{i,i} = \left(\sum_{j=1}^{n} |a_{i,j}| \right)^{-1}.$$

Vorlesung 10

Spezielle Gleichungssysteme

Wir betrachten in diesem Abschnitt Klassen von Matrizen deren LU-Zerlegung besondere Eigenschaften besitzt.

10.1 Symmetrisch Positiv Definite Matrizen

Diese Klasse von Matrizen haben wir bereits kennengelernt. Sie erlaubt eine besondere Behandlung der LU-Zerlegung wie der folgende Satz zeigt.

Satz 10.1. Eine symmetrisch positiv definite Matrix $A \in \mathbb{R}^{n\times n}$ ist stets *ohne* Pivotisierung stabil LU-zerlegbar. Für die Diagonalelemente von U (die Pivotelemente) gilt

$$u_{k,k} \geq \lambda_{\min}(A), \qquad 1 \leq k \leq n.$$

Beweis. Zunächst zeigen wir dass alle im Laufe der LU-Zerlegung entstehenden Untermatrizen symmetrisch und positiv definit sind. Dazu nutzen wir die rekursive Formulierung und starten mit einer 2×2 Blockzerlegung:

$$A = \begin{bmatrix} \alpha & v^T \\ v & B \end{bmatrix},$$

mit $\alpha \in \mathbb{R}$, $v \in \mathbb{R}^{n-1}$ and $B \in \mathbb{R}^{n-1\times n-1}$. Elimination der Spalte v liefert dann

$$\begin{bmatrix} 1 & 0 \\ -\frac{v}{\alpha} & I \end{bmatrix} \begin{bmatrix} \alpha & v^T \\ v & B \end{bmatrix} = \begin{bmatrix} \alpha & v^T \\ 0 & B - \frac{1}{\alpha} vv^T \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & B - \frac{1}{\alpha} vv^T \end{bmatrix} \begin{bmatrix} 1 & \frac{v^T}{\alpha} \\ 0 & I \end{bmatrix}.$$

Multiplikation mit $X = \begin{bmatrix} 1 & \frac{v^T}{\alpha} \\ 0 & I \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -\frac{v^T}{\alpha} \\ 0 & I \end{bmatrix}$ von rechts liefert die Identität

$$\begin{bmatrix} 1 & 0 \\ -\frac{v}{\alpha} & I \end{bmatrix} \begin{bmatrix} \alpha & v^T \\ v & B \end{bmatrix} \begin{bmatrix} 1 & -\frac{v^T}{\alpha} \\ 0 & I \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & B - \frac{1}{\alpha} vv^T \end{bmatrix} =: \tilde{A},$$

also $X^T AX = \tilde{A}$. Die Matrix X hat offensichtlich vollen Rang und A ist symmetrisch und positiv definit nach Voraussetzung. Damit ist auch $\tilde{A}$ symmetrisch und positiv definit und damit auch $B - \frac{1}{\alpha} vv^T$. Somit ist die zu zerlegende Restmatrix auch stets symmetrisch und positiv definit.

Nun schätzen wir die Diagonalelemente nach unten ab. Dazu nutzen wir die Ungleichung

$$(Ax, x)_2 \geq \lambda_{\min}(A)(x, x)_2, \qquad \forall x \in \mathbb{R}^n$$

und jede symmetrische und positiv definite Matrix A. Denn dazu schreibe x ein einer Basis aus orthonormalen Eigenvektoren (die existiert) $x = \sum_{i=1}^n \beta_i e_i$ und setze ein: $(Ax, x)_2 = \sum_{i=1}^n \lambda_i \beta_i^2 \geq \lambda_{\min} \sum_{i=1}^n \beta_i^2 = \lambda_{\min}(A)(x, x)_2$.

Für alle Diagonalelemente von A und insbesondere $\alpha = a_{1,1}$ gilt dann mit den kartesischen Koordinateneinheitsvektoren $z_k^i = \delta_{i,k}$

$$a_{i,i} = (Az^{(i)}, z^{(i)})_2 \geq \lambda_{\min}(A)(z^{(i)}, z^{(i)})_2 = \lambda_{\min}(A).$$

Für die Diagonalelemente der Restmatrix $B - \frac{1}{\alpha}vv^T$ gilt

$$\begin{aligned}\left(B - \frac{1}{\alpha}vv^T\right)_{i,i} &= \left(\tilde{A}z^{(i+1)}, z^{(i+1)}\right)_2 = \left(X^T AXz^{(i+1)}, z^{(i+1)}\right)_2 \\ &= \left(AXz^{(i+1)}, Xz^{(i+1)}\right)_2 \\ &\geq \lambda_{\min}(A)\left(Xz^{(i+1)}, Xz^{(i+1)}\right)_2 = \lambda_{\min}(A)\left(1 + \frac{v_i^2}{\alpha^2}\right) \\ &\geq \lambda_{\min}(A).\end{aligned}$$

Somit ist gezeigt, dass alle Pivotelemente stets auch ohne Pivotisierung durch den minimalen Eigenwert der Ausgangsmatrix A nach unten beschränkt sind und mithin nicht Null werden können.

Bezüglich der numerischen Stabilität beobachten wir, dass die spektrale Konditionszahl $\mathrm{cond}_2(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$ sich wie $1/\lambda_{\min}(A)$ verhält. Anderseits sind die Einträge von L auch von der Form $a_{i,k}/a_{k,k} \geq a_{i,k}/\lambda_{\min}(A)$, können also nicht schlimmer wachsen als die Konditionszahl. □

Satz 10.2 (Cholesky-Zerlegung)**.** Für eine symmetrisch positiv definite Matrix A existiert eine Zerlegung

$$A = LDL^T$$

mit einer unteren Dreiecksmatrix L deren Einträge 1 auf der Hauptdiagonale sind sowie einer Diagonalmatrix D mit positiven Einträgen auf der Diagonale.

Alternativ existiert eine Zerlegung der Form

$$A = \tilde{L}\tilde{L}^T$$

mit unterer Dreiecksmatrix $\tilde{L}$. Diese Zerlegung heisst Cholesky-Zerlegung. Beide Zerlegungen können mit Aufwand $n^3/3 + O(n^2)$ berechnet werden und benötigen einen Speicheraufwand $n(n+1)/2$.

Beweis. Wir zeigen zunächst die erste Zerlegung mittels Induktion über n. Zunächst sei $n = 1$. Die Matrix A besteht aus einem Element $a_{1,1} > 0$. Mit $l_{1,1} = 1$ und $d_{1,1} = a_{1,1}$ gilt dann $A = LDL^T$ und L bzw. D haben die behaupteten Eigenschaften.

Induktionsschritt $n - 1 \to n$. Nun sei A eine $n \times n$ Matrix. Aus dem Beweis von Satz 10.1 und der Induktionsvoraussetzung ersehen wir dass gilt

$$\begin{aligned}A &= \begin{bmatrix}\alpha & v^T \\ v & B\end{bmatrix} = \begin{bmatrix}1 & 0 \\ \frac{v}{\alpha} & I\end{bmatrix}\begin{bmatrix}\alpha & 0 \\ 0 & B - \frac{1}{\alpha}vv^T\end{bmatrix}\begin{bmatrix}1 & \frac{v^T}{\alpha} \\ 0 & I\end{bmatrix} \\ &= \begin{bmatrix}1 & 0 \\ \frac{v}{\alpha} & I\end{bmatrix}\begin{bmatrix}\alpha & 0 \\ 0 & L_{n-1}D_{n-1}L_{n-1}^T\end{bmatrix}\begin{bmatrix}1 & \frac{v^T}{\alpha} \\ 0 & I\end{bmatrix} \\ &= \begin{bmatrix}1 & 0 \\ \frac{v}{\alpha} & I\end{bmatrix}\begin{bmatrix}1 & 0 \\ 0 & L_{n-1}\end{bmatrix}\begin{bmatrix}\alpha & 0 \\ 0 & D_{n-1}\end{bmatrix}\begin{bmatrix}1 & 0 \\ 0 & L_{n-1}^T\end{bmatrix}\begin{bmatrix}1 & \frac{v^T}{\alpha} \\ 0 & I\end{bmatrix} \\ &= \begin{bmatrix}1 & 0 \\ \frac{v}{\alpha} & L_{n-1}\end{bmatrix}\begin{bmatrix}\alpha & 0 \\ 0 & D_{n-1}\end{bmatrix}\begin{bmatrix}1 & \frac{v^T}{\alpha} \\ 0 & L_{n-1}^T\end{bmatrix}.\end{aligned}$$

Da $\alpha > 0$ gelten die behaupteten Eigenschaften.

Der Hauptaufwand zur Berechnung der Zerlegung steckt in $B - \frac{1}{\alpha}vv^T$. Aufgrund der Symmetrie des Rang-1-Updates muss nur die Hälfte der Elemente berechnet und auch gespeichert werden. Der Speicheraufwand ist n für

die Diagonale und $(n^2-n)/2 = n(n-1)/2$ für L. Zusammen ergibt sich $(n^2-n)/2+n = (n^2+n)/2 = n(n+1)/2$.

Mit der Matrix $D^{1/2} = \operatorname{diag}(\sqrt{d_{1,1}}, \ldots, \sqrt{d_{n,n}})$ ergibt sich die Cholesky-Zerlegung $A = LD^{1/2}D^{1/2}L^T = \tilde{L}\tilde{L}^T$ mit $\tilde{L} = LD^{1/2}$. □

10.2 Diagonaldominante Matrizen

Hier ist noch eine Klasse von Matrizen die sich ohne Pivotisierung LU-zerlegen lassen.

Definition 10.3. Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt *diagonaldominant* falls

$$\sum_{j=1, j\neq i}^{n} |a_{i,j}| \leq |a_{i,i}|, \qquad 1 \leq i \leq n.$$

Manche Autoren unterscheiden zwischen starker und schwacher Diagonaldominanz, daher genau auf die Definition achten. □

Satz 10.4. Sei $A \in \mathbb{R}^{n\times n}$ regulär und diagonaldominant. Dann exisitiert eine LU-Zerlegung ohne Pivotisierung.

Beweis. Wir nutzen wieder die 2×2 Blockzerlegung von $A = \begin{bmatrix} \alpha & w^T \\ v & B \end{bmatrix}$.

Wir zeigen $|\alpha| > 0$: Entweder ist schon $\sum_{j=1}^{n-1} |w_j| > 0$ und damit $|\alpha| > 0$ wegen Diagonaldominanz oder, falls $\sum_{j=1}^{n-1} |w_j| = 0$, gilt $|\alpha| > 0$ wegen der Regularität von A.

Mit der Diagonaldominanz von A folgern wir:

$$\sum_{j=1}^{n-1} |w_j| \leq |\alpha| \;\Leftrightarrow\; \sum_{j=1}^{n-1} \frac{|w_j|}{|\alpha|} \leq 1 \qquad \text{Zeile 1,}$$

$$|v_i| + \sum_{j=1, j\neq i}^{n-1} |b_{i,j}| \leq |b_{i,i}| \qquad \text{Zeilen } i = 2 \ldots n.$$

Ein Schritt der LU-Zerlegung lautet

$$\begin{bmatrix} 1 & 0 \\ -v/\alpha & I \end{bmatrix} \begin{bmatrix} \alpha & w^T \\ v & B \end{bmatrix} = \begin{bmatrix} \alpha & w^T \\ 0 & B - \frac{1}{\alpha} v w^T \end{bmatrix}.$$

Für die Restmatrix $B - \frac{1}{\alpha} v w^T$ rechnen wir nach:

$$\begin{aligned}
\sum_{j=1}^{n-1} \left| b_{i,j} - \frac{v_i w_j}{\alpha} \right| &\leq \sum_{j=1, j\neq i}^{n-1} \left| b_{i,j} - \frac{v_i w_j}{\alpha} \right| + |b_{i,i}| + \left| \frac{v_i w_i}{\alpha} \right| \\
&\leq |b_{i,i}| + \sum_{j=1, j\neq i}^{n-1} |b_{i,j}| + |v_i| \sum_{j=1, j\neq i}^{n-1} \frac{|w_j|}{|\alpha|} + |v_i| \frac{|w_i|}{|\alpha|} \\
&= |b_{i,i}| + \sum_{j=1, j\neq i}^{n-1} |b_{i,j}| + |v_i| \sum_{j=1}^{n-1} \frac{|w_j|}{|\alpha|} \\
&\leq |b_{i,i}| + \sum_{j=1, j\neq i}^{n-1} |b_{i,j}| + |v_i| \leq |b_{i,i}| + |b_{i,i}|
\end{aligned}$$

$$\Leftrightarrow \quad \sum_{j=1, j\neq i}^{n-1} \left| b_{i,j} - \frac{v_i w_j}{\alpha} \right| \leq |b_{i,i}| - \left| \frac{v_i w_i}{\alpha} \right|.$$

Mit der inversen Dreiecksungleichung erhalten wir

$$\sum_{j=1,j\neq i}^{n-1} \left| b_{i,j} - \frac{v_i w_j}{\alpha} \right| \leq |b_{i,i}| - \left| \frac{v_i w_i}{\alpha} \right| \leq \left| b_{i,i} - \frac{v_i w_i}{\alpha} \right| = \left| \left(B - \frac{vw^T}{\alpha} \right)_{i,i} \right|.$$

Somit ist die Restmatrix ebenfalls diagonaldominant und das Argument kann rekursiv fortgesetzt werden. □

10.3 Bandmatrizen

Die folgende Klasse von Matrizen kann mit reduziertem Aufwand LU-zerlegt werden.

Definition 10.5. Sei $A \in \mathbb{R}^{n\times n}$ und $m \in \mathbb{N}$. A heißt *Bandmatrix* mit *Bandbreite* m wenn gilt

$$|i-j| > m \quad \Rightarrow \quad a_{i,j} = 0.$$

Satz 10.6. Sei $A \in \mathbb{R}^{n\times n}$ Bandmatrix mit Bandbreite m. Weiter sei A ohne Pivotisierung LU-zerlegbar. Dann sind ihre Dreiecksfaktoren L und U ebenfalls Bandmatrizen mit Bandbreite m, der Aufwand zur LU-Zerlegung beträgt

$$N_{\text{Band}}(n,m) \leq 2nm^2 + nm$$

und der Speicheraufwand beträgt asymptotisch $2nm + n$.

Beweis. Wir beweisen mit Induktion über n dass die Bandstruktur bei der LU-Zerlegung erhalten bleibt. Im Fall $n \leq m$ ist die Bandmatrix voll besetzt und es besteht kein Unterschied zu Nicht-Bandmatrizen.

Nun sei $n > m$ und die Aussage sei bis $n-1$ bewiesen. Dann können wir die $n \times n$-Matrix A als 3×3 Blockmatrix schreiben bei der ein Schritt der LU-Zerlegung durchgeführt wird:

$$A = \begin{bmatrix} \alpha & w^T & 0 \\ v & B & C \\ 0 & D & E \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} \alpha & w^T & 0 \\ 0 & B - \frac{1}{\alpha} v w^T & C \\ 0 & D & E \end{bmatrix}.$$

Die Größen der Blöcke betragen der Reihe nach 1, $m-1$ und $n-m$. Die erste Spalte und erste Zeile haben Bandbreite m und werden Teil von L bzw. U. Die Untermatrix B der Größe $m-1 \times m-1$ wird modifiziert, die Untermatrizen C, D und E bleiben unverändert. Somit ist die zu zerlegende Restmatrix

$$A_{n-1} = \begin{bmatrix} B - \frac{1}{\alpha} v w^T & C \\ D & E \end{bmatrix}$$

wieder Bandmatrix mit Bandbreite m welche laut Induktionsvoraussetzung entsprechend mit Bandstruktur LU-zerlegt werden kann.

Die Aufwände können wir direkt abschätzen. In jedem der $n-1$ Schritte der LU-Zerlegung beträgt der Aufwand zur Berechnung von v/α bzw. $B - \frac{1}{\alpha} v w^T$, $m-1$ bzw. $(m-1)^2$ Rechenoperationen, also:

$$N_{\text{Band}}(n,m) \leq \sum_{k=1}^{n-1} (m - 1 + (m-1)^2) \leq nm^2 + nm.$$

Da jede Zeile höchstens $2m+1$ Nichtnullelemente hat beträgt der Speicheraufwand höchstens $2nm + n$. □

Vorlesung 11

Nichtreguläre Systeme

11.1 Motivation: Ausgleichsrechnung

Gegeben seien m Datenpunkte

$$(x_i, y_i) \in \mathbb{R} \times \mathbb{R}, \qquad 1 \leq i \leq m,\ x_i \neq x_j\ \forall i \neq j$$

und n Funktionen

$$u_j : \mathbb{R} \to \mathbb{R}, \qquad 1 \leq j \leq n.$$

Gesucht sind dann n Koeffizienten z_j so dass die Funktion

$$u(x) = \sum_{j=1}^{n} z_j u_j(x)$$

die m gegebenen Datenpunkte möglichst gut repräsentiert. Beispiele für die Funktionen u_j sind etwa

$$u_j(x) = x^{j-1}, \qquad\qquad u_j(x) = \sin(j\pi x).$$

Ein naheliegender Ansatz dies zu tun ist es die quadratische Abweichung zu minimieren:

$$\sum_{i=1}^{m} (y_i - u(x_i))^2 \quad \to \quad \min.$$

Dies nennt man die *Methode der kleinsten Quadrate* bzw. *least squares method*. Sie wurde von Gauß bzw. Legendre eingeführt. Wir haben das Verfahren in der ersten Vorlesung verwendet um Modellparameter zu schätzen.

Schreibt man den Ansatz für $u(x)$ aus ergibt sich

$$\sum_{i=1}^{m} \left(y_i - \sum_{j=1}^{n} z_j \underbrace{u_j(x_i)}_{a_{i,j}} \right)^2 = \|y - Az\|_2^2 = (y - Az)^T (y - Az)$$

$$= z^T A^T A z - 2 z^T A^T y + y^T y,$$

mit der $m \times n$ Matrix A, $a_{i,j} = u_j(x_i)$. Dies ist ein quadratisches Minimierungsproblem in den Koeffizienten z.

Für die Lösbarkeit des Minimierungsproblems unterscheiden wir drei Fälle, wobei wir im folgenden annehmen wollen, dass der Rang von A maximal ist, d.h. $\text{Rang}(A) = \min(m, n)$.

i) $m < n$, d.h. weniger Datenpunkte als Koeffizienten. Dann gibt es mindestens ein z so dass $Az = y$ (beachte $\text{Rang}(A) = m$). und somit $\|y - Az\|_2 = 0$. Da $\text{Kern}(A) \neq \{0\}$ gibt es noch weitere, unendlich viele, Lösungen.

ii) $m = n$. Da $\text{Rang}(A) = n$ gibt es genau eine Lösung welche $Az = y$ und damit $\|y - Az\|_2 = 0$ erfüllt.

iii) $m > n$, d.h. mehr Datenpunkte als Koeffizienten. Dann gibt es *nicht* unbedingt ein z so dass $Az = y$ (nur falls $y \in \text{Bild}(A)$) und die Minimierungsaufgabe macht Sinn.

In diesem Kapitel sind wir in der Regel an der Lösung der Minimierungsaufgabe für $m \geq n$ interessiert.

11.2 Lösbarkeit und Normalengleichung

Wir wenden uns zunächst allgemein der Frage der Lösbarkeit des Minimierungsproblems zu. Dazu betrachten wir die allgemeinste Situation, d.h. $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, m, n beliebig und $\text{Rang}(A)$ beliebig.

Wir erinnern an die folgenden Begriffe aus der linearen Algebra:

$$\begin{aligned}\text{Bild}(A) &= \{y \in \mathbb{R}^m \,:\, y = Ax, x \in \mathbb{R}^n\} \subseteq \mathbb{R}^m,\\ \text{Kern}(A) &= \{x \in \mathbb{R}^n \,:\, Ax = 0\} \subseteq \mathbb{R}^n,\\ \text{Rang}(A) &= \dim(\text{Bild}(A)) = \text{Rang}(A^T) = \dim(\text{Bild}(A^T)).\end{aligned}$$

Ist V ein Unterraum des Vektorraumes W, dann ist das orthogonale Komplement von V gegeben durch

$$V^\perp = \{w \in W \,:\, (w, v)_2 = 0 \ \forall v \in V\}.$$

Schließlich gilt

$$\begin{aligned}y \in \text{Bild}(A)^\perp &\Leftrightarrow (y, w)_2 = 0 \ \forall w \in \text{Bild}(A)\\ &\Leftrightarrow (y, Ax)_2 = 0 \ \forall x \in \mathbb{R}^n\\ &\Leftrightarrow (A^T y, x)_2 = 0 \ \forall x \in \mathbb{R}^n\\ &\Leftrightarrow A^T y = 0\\ &\Leftrightarrow y \in \text{Kern}(A^T),\end{aligned}$$

womit wir $\text{Bild}(A)^\perp = \text{Kern}(A^T)$ gezeigt haben. Dies zeigt dann wiederum

$$m = \dim(\text{Bild}(A)) + \dim(\text{Bild}(A)^\perp) = \dim(\text{Bild}(A)) + \dim(\text{Kern}(A^T)).$$

Wir zeigen den folgenden allgemeinen Satz über die Lösung des Least-Sqares-Problemes.

Satz 11.1. Es sei $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, m, n beliebig und $\text{Rang}(A)$ beliebig. Dann gelten folgende Aussagen:

a) Es existiert ein $x^* \in \mathbb{R}^n$ so dass

$$\|Ax^* - b\|_2 = \min_{x\in\mathbb{R}^n} \|Ax - b\|_2.$$

b) Diese Bedingung ist äquivalent dazu dass x^* Lösung von

$$A^T A x^* = A^T b, \tag{11.1}$$

der sogenannten *Normalengleichung*.

c) Falls $\text{Rang}(A) = n$ (und damit $m \geq n$), ist x^* eindeutig bestimmt. Sonst hat jede weitere Lösung die Form $x^* + y$ mit $y \in \text{Kern}(A)$.

Beweis. Wir zeigen zunächst $(b) \Rightarrow (a)$. Sei x^* eine Lösung der Normalengleichung (11.1). Für ein Beliebiges $x \in \mathbb{R}^n$ gilt dann

$$\begin{aligned}
\|Ax-b\|_2^2 &= \|A(x-x^*+x^*)-b\|_2^2 \\
&= (A(x-x^*+x^*)-b, A(x-x^*+x^*)-b)_2 \\
&= ((Ax^*-b)+A(x-x^*), (Ax^*-b)+A(x-x^*))_2 \\
&= (Ax^*-b, Ax^*-b)_2 + 2(Ax^*-b, A(x-x^*))_2 + (A(x-x^*), A(x-x^*))_2 \\
&= \|Ax^*-b\|_2^2 + \|A(x-x^*)\|_2^2 \\
&\geq \|Ax^*-b\|_2^2,
\end{aligned}$$

also nimmt $\|Ax-b\|_2^2$ sein Minimum in $x = x^*$ an. Der gemischte Term ist stets Null, da $(Ax^*-b, A(x-x^*))_2 = (A^T(Ax^*-b), x-x^*)_2$ und $A^T(Ax^*-b) = 0$ wegen $A^TAx^* = A^Tb$.

Nun zeigen wir $(a) \Rightarrow (b)$. Betrachte die Abbildung $F : \mathbb{R}^n \to \mathbb{R}$ gegeben durch $F(x) = \|Ax-b\|^2$. Notwendige Bedingung für ein Minimum von F in x^* ist, dass die partiellen Ableitungen verschwinden, d.h.

$$\frac{\partial F}{\partial x_k}(x^*) = 0, \qquad 1 \leq k \leq n.$$

Die Ableitungen rechnen wir brute-force nach:

$$\begin{aligned}
\frac{\partial F}{\partial x_k}(x) &= \frac{\partial}{\partial x_k} \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{i,j}x_j - b_i \right)^2 = \sum_{i=1}^{m} 2 \left(\sum_{j=1}^{n} a_{i,j}x_j - b_i \right) a_{i,k} \\
&= 2 \sum_{i=1}^{m} (A^T)_{k,i} \left(Ax-b\right)_i = 2 \left(A^T(Ax-b)\right)_k
\end{aligned}$$

Damit folgt

$$\frac{\partial F}{\partial x_k}(x^*) = 0,\ 1 \leq k \leq n \quad \Leftrightarrow \quad 2A^T(Ax^*-b) = 0 \quad \Leftrightarrow \quad A^TAx^* = A^Tb.$$

Aus der Betrachtung oben folgt, dass der Extremalpunkt x^* tatsächlich ein Minimum ist.

Schließlich betrachten wir die Lösbarkeit der Normalengleichung. Wegen

$$\mathbb{R}^m = \text{Bild}(A) \oplus \text{Bild}(A)^\perp$$

kann jedes $b \in \mathbb{R}^m$ eindeutig in zerlegt werden in

$$b = s + r, \qquad s \in \text{Bild}(A), \qquad r \in \text{Bild}(A)^\perp = \text{Kern}(A^T).$$

Zu $s \in \text{Bild}(A)$ gibt es dann ein $\bar{x} \in \mathbb{R}^n$ mit $A\bar{x} = s$. Für dieses gilt dann

$$A^TA\bar{x} = A^Ts = A^Ts + A^Tr = A^Tb,$$

d.h. dieses $\bar{x}$ ist auch Lösung der Normalengleichung.

Für $\text{Rang}(A) = n$ und damit $m \geq n$ untersuchen wir das homogene System $A^TAx = 0$ also $\text{Kern}(A^TA)$. Wegen $\text{Rang}(A) = n$ ist $\text{Kern}(A) = \{0\}$, d.h. für jedes $x \neq 0$ gilt $\text{Bild}(A) \ni Ax \neq 0$. Andererseits ist $\text{Kern}(A^T) = \text{Bild}(A)^\perp$ da aber $Ax \in \text{Bild}(A)$ muss also $A^TAx \neq 0$ gelten. Damit ist $\text{Kern}(A^TA) = \{0\}$ und die Lösung der Normalengleichung ist eindeutig bestimmt.

Nun sei $\text{Rang}(A) < n$ und x_1, x_2 seien zwei Lösungen der Normalengleichung, also $A^T A x_1 = A^T b$ und $A^T A x_2 = A^T b$. Subtraktion beider Gleichungen liefert

$$A^T A(x_1 - x_2) = 0.$$

Wegen $\text{Kern}(A^T) = \text{Bild}(A)^\perp$ muss schon $A(x_1 - x_2) = 0$ sein um diese Gleichung erfüllen zu können. Alle Lösungen haben also die Gestalt $\bar{x} + y$ wobei $\bar{x}$ eine Lösung der Normalengleichung ist und $y \in \text{Kern}(A)$ gilt. □

Der letzte Satz zeigt, dass das Problem der kleinsten Quadrate mit Hilfe der Normalengleichung gelöst werden kann. Die dabei auftretende Matrix $A^T A$ ist in jedem Fall symmetrisch und positiv *semidefinit*, d.h. $x^T A^T A x = \|Ax\|_2^2 \geq 0$.

Im Fall $\text{Rang}(A) = n$ ist A positiv definit. Dann kann man das System mit LU-Zerlegung lösen. Dies hat jedoch zwei Nachteile:

- Die Berechnung der Matrix $A^T A$ hat Aufwand $O(mn^2)$. Falls A sehr viele Nullelemente hat $A^T A$ möglicherweise nicht mehr viele Nullelemente.

- Die Kondition der Matrix $A^T A$ ist die quadrierte Kondition von A. Die Normalengleichung ist also möglicherweise sehr viel schlechter konditioniert als A selbst. Dies sieht man zumindest für symmetrische positiv definite Matrizen B schnell ein. Nach Lemma 5.18 gilt in diesem Fall $\|B\|_2 = \varrho(B) = \sqrt{\varrho(B^T B)}$, also $\varrho(B^T B) = \varrho^2(B)$. Für die Inversen gilt das analog und somit auch für die Kondition.

Im folgenden leiten wir Methoden her die die Normalengleichung ohne Aufstellung von $A^T A$ lösen.

11.3 Schmale QR-Zerlegung

Wir fassen $A \in \mathbb{R}^{m \times n}$, $A = [a_1, \ldots, a_n]$ als n Spaltenvektoren der Länge m auf. Dann gilt

$$\text{Bild}(A) = \text{span}\{a_1, \ldots, a_n\}.$$

Im folgenden nehmen wir $n \leq m$ und $\text{Rang}(A) = n$ an, d.h. die a_1 bis a_n sind linear unabhängig.

Algorithmus 11.2 (Gram-Schmidt)**.** Das Orthogonalisierungsverfahren von *Gram-Schmidt* berechnet eine Orthonormalbasis $q_1, \ldots, q_n$ von $\text{Bild}(A)$:

1) Setze $q_1' = a_1$ und $k = 2$.

2) Falls $k > n$ gehe zu 4), sonst

3) Setze

$$q_k' = a_k - \sum_{j=1}^{k-1} \frac{(a_k, q_j')_2}{(q_j', q_j')_2} q_j'$$

und erhöhe $k = k + 1$. Gehe zu Schritt 2).

4) Für $i = 1, \ldots, n$ setze $q_i = \frac{q_i'}{\|q_i'\|_2}$.

Die Schritte 1 bis 3 führen die Orthogonalisierung durch. Offensichtlich ist $\text{span}\{q_1'\} = \text{span}\{a_1\}$. Für $k > 1$ sei bereits $\text{span}\{a_1, \ldots, a_{k-1}\} = \text{span}\{q_1', \ldots, q_{k-1}'\}$ und $(q_i', q_j')_2 = 0$, $1 \le i < j \le k-1$. Dann setzt man an

$$q_k' = a_k - \sum_{j=1}^{k-1} \alpha_{k,j} q_j'$$

und bestimmt die $k-1$ Koeffizient $\alpha_{k,j}$ aus den Gleichungen $(q_k', q_i') = 0$, $i < k$:

$$(q_k', q_i')_2 = \left(a_k - \sum_{j=1}^{k-1} \alpha_{k,j} q_j', q_i' \right)_2 = (a_k, q_i')_2 - \alpha_{k,i}(q_i', q_i')_2 = 0$$

also

$$\alpha_{k,i} = \frac{(a_k, q_i')_2}{(q_i', q_i')_2}.$$

Damit ist q_k' orthogonal zu allen vorherigen q_j'. Da $\text{Rang}(A) = n$, gilt $(a_k, q_i')_2 \neq 0$ denn die q_i' sind Linearkombinationen der a_j für $j \le i < k$. Im Schritt 4 werden die Vektoren q_k' für die Orthonormalbasis noch normiert. □

Satz 11.3. Es sei $A \in \mathbb{R}^{m \times n}$ und $\text{Rang}(A) = n$. Dann existiert die eindeutige *schmale QR-Zerlegung*

$$A = QR \tag{11.2}$$

mit $Q \in \mathbb{R}^{m \times n}$, $R \in \mathbb{R}^{n \times n}$ mit $Q^T Q = I$ und R obere Dreiecksmatrix. Darüber hinaus ist $R^T R = A^T A$ gerade die Choleskyzerlegung von $A^T A$.

Beweis. Zunächst zeigen wir die Existenz konstruktiv. Mit den Koeffizienten aus dem Gram-Schmidt-Verfahren von oben gilt

$$\begin{aligned} a_1 &= \|q_1'\|_2 \; q_1, && k = 1, \\ a_k &= \|q_k'\|_2 \; q_k + \sum_{j=1}^{k-1} \frac{(a_k, q_j')_2}{(q_j', q_j')_2} q_j' = \|q_k'\|_2 \; q_k + \sum_{j=1}^{k-1} \frac{(a_k, q_j')_2}{\|q_j'\|_2} q_j, && k > 1. \end{aligned}$$

Mit den Matrizen $Q = [q_1, \ldots, q_n] \in \mathbb{R}^{m \times n}$ und $R \in \mathbb{R}^{n \times n}$,

$$r_{1,1} = \|a_1\|_2, \qquad r_{k,k} = \|q_k'\|_2, \qquad r_{i,k} = \frac{(a_k, q_i')_2}{\|q_i'\|_2},$$

gilt gerade $A = QR$. Wegen $q_k \neq 0$ ist R regulär.

Nun zur Eindeutigkeit. Es gilt $A^T A = (QR)^T QR = R^T Q^T QR = R^T R$. A ist symmetrisch und positiv definit und hat die eindeutige Cholesky Zerlegung $A^T A = LL^T$ somit muss $L = R^T$ gelten. Da R regulär, gilt $AR^{-1} = Q$ und damit ist Q auch eindeutig bestimmt. □

Man kann sich die LU-Zerlegung als Gaußelimination mit Buchführung vorstellen. In diesem Sinn ist die schmale QR-Zerlegung eine Gram-Schmidt-Orthogonalisierung mit Buchführung. Allerdings benötigt man zusätzlichen Speicherplatz für diese Buchführung in R, selbst wenn man A durch Q überschreibt.

Mit der (schmalen) QR-Zerlegung kann man die Normalengleichung nun wie folgt lösen:

$$A^T Ax = A^T b \quad \Leftrightarrow \quad R^T Q^T QRx = A^T b \quad \Leftrightarrow \quad R^T Rx = A^T b,$$

d.h. man löst erst $R^T y = A^T b$ durch vorwärts einsetzen und dann $Rx = y$ durch rückwärts einsetzen. $A^T A$ wurde dabei *nicht* explizit aufgestellt.

Allerdings besitzt die Methode wie vorgestellt noch einen entscheidenden Nachteil: Im Gram-Schmidt Verfahren pflanzen sich Rundungsfehler sehr stark fort und es ist nicht numerisch stabil. Allerdings gibt es eine Modifikation welche deutlich weniger anfällig für Rundungsfehler ist:

Algorithmus 11.4 (Modifizierter Gram-Schmidt)**.** Das modifizierte Verfahren von *Gram-Schmidt* berechnet eine Orthonormalbasis $q_1, \ldots, q_n$ von Bild(A). In exakter Arithmetik produziert es die selben Vektoren wie das klassische Gram-Schmidt Verfahren:

1) Für $j = 1, \ldots, n$ setze $q'_j = a_j$ und setze $k = 1$.

2) Ist $k = n$ gehe zu 4)

3) Für $j = k+1, \ldots, n$ führe folgendes update durch

$$q'_j = q'_j - \frac{(q'_j, q'_k)_2}{(q'_k, q'_k)_2} q'_k.$$

   Setze $k = k + 1$ und gehe zu 2).

4) Für $j = 1, \ldots, n$ setze $q_j = \frac{q'_j}{\|q'_j\|_2}$.

Im klassischen Gram-Schmidt Verfahren von oben berechnet man q'_k aus allen vorherigen Vektoren mittels

$$q'_k = a_k - \sum_{j=1}^{k-1} \frac{(a_k, q'_j)_2}{(q'_j, q'_j)_2} q'_j,$$

dabei ist $\frac{(a_k, q'_j)_2}{(q'_j, q'_j)_2} q'_j$ die orthogonale Projektion von a_k in Richtung q'_j, welche man aus a_k „heraus nimmt“. Stattdessen berechnet die modifizierte Gram-Schmidt Orthogonalisierung

$$\begin{aligned}
q'_{k,0} &= a_k \\
q'_{k,1} &= q'_{k,0} - \frac{(q'_{k,0}, q'_1)_2}{(q'_{1,0}, q'_{1,0})_2} q'_{1,0} \\
q'_{k,2} &= q'_{k,1} - \frac{(q'_{k,1}, q'_{2,1})_2}{(q'_{2,1}, q'_{2,1})_2} q'_{2,1} \\
&\vdots \\
q'_{k,k-1} &= q'_{k,k-2} - \frac{(q'_{k,k-1}, q'_{k-1,k-2})_2}{(q'_{k-1,k-2}, q'_{k-1,k-2})_2} q'_{k-1,k-2}
\end{aligned}$$

Man projiziert also immer die aktuell beste Version von q'_k in Richtung des berechneten q'_j und subtrahiert diesen Anteil. In exakter Arithmetik wäre stets

$$\frac{(q'_{k,i}, q'_{j,j-1})_2}{(q'_{j,j-1}, q'_{j,j-1})_2} = \frac{(a_k, q'_{j,j-1})_2}{(q'_{j,j-1}, q'_{j,j-1})_2}$$

da $q'_{k,i} \in \text{span}\{a_k, q'_{1,0}, q'_{2,1}, \ldots, q'_{i,i-1}\}$. □

Beispiel 11.5. Folgende Tabelle zeigt den Unterschied zwischen dem klassischen und modifizierten Gram-Schmidt Verfahren. Wir benutzen wieder die Matrix aus dem Beispiel über Integralgleichungen und orthogonalisieren der Spalten. Dann berechnen wir $\|Q^TQ - I\|_\infty$:

n	γ	$\text{cond}_\infty(A)$	cgs	mgs
128	0.5	29	2.78057e-14	2.76229e-14
128	1e-2	12145	0.737637	1.55591e-11
128	1e-4	1.26e+06	0.791452	1.61103e-09

Für kleine Konditionszahlen besteht kein Unterschied. Bei großer Kondition sieht man jedoch deutlich die Verbesserung durch den modifizierten Algorithmus. Die Berechnung erfolgte in Genauigkeit `double`. □

Vorlesung 12

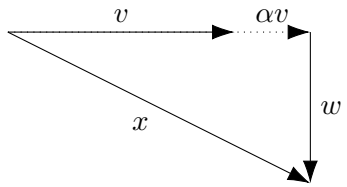
QR-Zerlegung

Wir lernen in diesem Kapitel eine weitere QR-Zerlegung kennen, die noch kompakter abgespeichert werden kann. Aber wie ist das überhaupt möglich? Wir haben doch bewiesen, dass die QR-Zerlegung eindeutig ist? Des Rätsels Lösung liegt in der Dimension der Matrizen. Ist A eine $m \times n$ Matrix mit $m \geq n$ dann hat bei der schmalen QR-Zerlegung Q die Dimension $m \times n$ und R die Dimension $n \times n$. Diese Zerlegung ist eindeutig. Bei der in diesem Kapitel besprochenen vollen QR-Zerlegung wird Q die Dimension $m \times m$ und R die Dimension $m \times n$ haben. Diese Zerlegung ist nicht eindeutig. Die Beziehung zur schmalen QR-Zerlegung wird im Laufe des Kapitels dargelegt.

12.1 Householder Transformation

Es gibt mehrere Möglichkeiten eine volle QR-Zerlegung zu berechnen. Wir betrachten hier nur die Householder-Transformation, welche wir erst mal vorbereiten wollen.

Wir betrachten die Projektion eines Vektors x in Richtung eines Vektors v wie sie bereits im Gram-Schmidt Verfahren auftrat.



Sind x, v, $v \neq 0$ gegeben, so wollen wir $x = \alpha v + w$ darstellen mit $(w, v)_2 = 0$. Aus der Orthogonalitätsbedingung und dem Ansatz erhalten wir

$$0 = (w, v)_2 = (x - \alpha v, v)_2 = (x, v)_2 - \alpha (v, v)_2$$

woraus wir

$$\alpha = \frac{(x, v)_2}{(v, v)_2}$$

berechnen können. Wir beobachten

$$\alpha v = \frac{(x, v)_2}{(v, v)_2} v = \frac{x^T v}{v^T v} v = \frac{v(v^T x)}{v^T v} = \frac{v v^T}{v^T v} x$$

mit der Rang-1-Matrix vv^T. Damit erhalten wir die Darstellungen

$$w = x - \alpha v = \left(I - \frac{v v^T}{v^T v} \right) x, \qquad x = \alpha v + w = \frac{v v^T}{v^T v} x + \left(I - \frac{v v^T}{v^T v} \right) x.$$

Mit diesen Vorarbeiten können wir nun die Householder-Transformation einführen.

Definition 12.1. Sei $v \in \mathbb{R}^n$, $v \neq 0$ gegeben. Dann heißt die Matrix

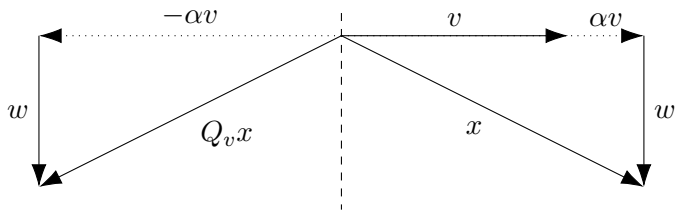
$$Q_v = I - 2\frac{vv^T}{v^T v}$$

Householder-Matrix bzw. *Householder*-Transformation oder auch *Householder*-Reflektion. Der Vektor v heißt *Householder*-Vektor. □

Mit der orthogonalen Zerlegung $x = \alpha v + w$ von oben erkennen wir sofort

$$Q_v x = \left(I - 2\frac{vv^T}{v^T v}\right) x = -\frac{vv^T}{v^T v}x + \left(I - \frac{vv^T}{v^T v}\right) x = -\alpha v + w,$$

also geometrisch:



In mehr als zwei Dimensionen wird x gerade an der Ebene senkrecht zu v gespiegelt. Außerdem gilt:

Lemma 12.2. Sei $v \in \mathbb{R}^n$, $v \neq 0$. Dann ist Q_v ist orthogonal und symmetrisch, d.h. $Q_v^T Q_v = I$ und $Q_v^T = Q_v$.

Beweis. Die Symmetrie $Q_v^T = Q_v$ ist offensichtlich, da eine Rang-1-Matrix vv^T symmetrisch ist. Dann rechnen wir

$$\begin{aligned} Q_v^T Q_v = Q_v Q_v &= \left(I - 2\frac{vv^T}{v^T v}\right)\left(I - 2\frac{vv^T}{v^T v}\right) \\ &= I - 4\frac{vv^T}{v^T v} + 4\frac{vv^T vv^T}{v^T vv^T v} = I - 4\frac{vv^T}{v^T v} + 4\frac{v(v^T v)v^T}{(v^T v)(v^T v)} \\ &= I - 4\frac{vv^T}{v^T v} + 4\frac{vv^T}{v^T v} = I. \end{aligned}$$

Alternativ kann man auch geometrisch argumentieren. Für $x = \alpha v + w$ gilt $Q_v x = -\alpha v + w$ und somit $Q_v^2 x = Q_v(-\alpha v + w) = \alpha v + w = x$. Zweimalige Spiegelung an der Ebene senkrecht zu v liefert also wieder x. □

12.2 Herleitung der Zerlegung

Nun betrachten wir das folgende Problem: Gegeben seien $x, z \in \mathbb{R}^n$, $z \neq 0$. Gesucht sei ein Householder-Vektor v so dass gilt

$$Q_v x = \beta z, \qquad \beta \neq 0.$$

Der Skalar β sei beliebig, wichtig ist nur, dass $Q_v x$ in Richtung von z zeigt.

Wegen

$$Q_v x = x - 2\frac{vv^T}{v^T v}x = x - 2\frac{v^T x}{v^T v}v \in \operatorname{span}\{x, v\}$$

und z i. allg. linear unabhängig von x muss $v \in \operatorname{span}\{x, z\}$ gelten. Wir machen also den Ansatz $v = c_1 x + c_2 z$ und erhalten damit

$$\begin{aligned} Q_v x &= \left(I - 2\frac{vv^T}{v^T v} \right) x = x - 2\frac{v^T x}{v^T v} v = x - 2\frac{v^T x}{v^T v}(c_1 x + c_2 z) \\ &= \left(1 - 2\frac{v^T x}{v^T v} c_1 \right) x - 2\frac{v^T x}{v^T v} c_2 z = \beta z. \end{aligned}$$

Nun steckt das unbekannte $v = c_1 x + c_2 z$ ja noch in beiden Linearfaktoren. Da $\beta \neq 0$ beliebig ist kann auch c_2 beliebig sein, d.h. wir müssen nur erreichen, dass der erste Linearfaktor Null wird. Es zeigt sich, dass dazu nur ein Parameter notwendig ist und wir setzen deshalb $c_1 = 0$ (!) und fahren mit dem Ansatz $v = x + c_2 z$ fort:

$$\begin{aligned} 0 &= 1 - 2\frac{v^T x}{v^T v} = 1 - 2\frac{(x + c_2 z)^T x}{(x + c_2 z)^T (x + c_2 z)} \\ &= 1 - 2\frac{x^T x + c_2 z^T x}{x^T x + 2c_2 z^T x + c_2^2 z^T z} \\ &= \frac{x^T x + 2c_2 z^T x + c_2^2 z^T z - 2(x^T x + c_2 z^T x)}{x^T x + 2c_2 z^T x + c_2^2 z^T z} \\ &= \frac{c_2^2 z^T z - x^T x}{x^T x + 2c_2 z^T x + c_2^2 z^T z} \;\Leftrightarrow\; c_2^2 \|z\|_2^2 = \|x\|_2^2 \;\Rightarrow\; c_2 = \pm \frac{\|x\|_2}{\|z\|_2}. \end{aligned}$$

Man erhält also $v = x \pm \frac{\|x\|_2}{\|z\|_2} z$. In der Praxis wählt man das Vorzeichen so, dass $\|v\|_2$ möglichst groß wird, also

$$\|v\|_2^2 = \left(x \pm \frac{\|x\|_2}{\|z\|_2} z, x \pm \frac{\|x\|_2}{\|z\|_2} z \right)_2 = 2\|x\|_2^2 \pm 2\frac{\|x\|_2}{\|z\|_2}(x, z)_2 \to \max,$$

also positiv, wenn $(x, z)_2 \geq 0$ und negativ wenn $(x, z)_2 < 0$ und damit gilt

$$c_2 = \operatorname{sgn}((x, z)_2)\frac{\|x\|_2}{\|z\|_2}, \qquad v = x + \operatorname{sgn}((x, z)_2)\frac{\|x\|_2}{\|z\|_2} z. \tag{12.1}$$

Für β rechnen wir mit der Abkürzung $\gamma = \operatorname{sgn}((x, z)_2)$ nach:

$$\begin{aligned} \beta &= -2\frac{v^T x}{v^T v} c_2 \\ &= -2\frac{\left(x + \gamma\frac{\|x\|_2}{\|z\|_2} z \right)^T x}{\left(x + \gamma\frac{\|x\|_2}{\|z\|_2} z \right)^T \left(x + \gamma\frac{\|x\|_2}{\|z\|_2} z \right)} \gamma\frac{\|x\|_2}{\|z\|_2} \\ &= -\frac{2\left(\|x\|_2^2 + \gamma\frac{\|x\|_2}{\|z\|_2}(z, x)_2 \right)}{\|x\|_2^2 + 2\gamma\frac{\|x\|_2}{\|z\|_2}(z, x)_2 + \|x\|_2^2} \gamma\frac{\|x\|_2}{\|z\|_2} = -\gamma\frac{\|x\|_2}{\|z\|_2} \end{aligned} \tag{12.2}$$

Damit kommen wir nun zur Berechnung der vollen QR-Zerlegung.

Satz 12.3. Sei $A \in \mathbb{R}^{m \times n}$ mit $m \geq n$ und $\operatorname{Rang}(A) = n$. Dann gelten folgende Aussagen:

a) Es existiert eine orthogonale Matrix $Q \in \mathbb{R}^{m \times m}$ sowie eine obere Dreiecksmatrix $R \in \mathbb{R}^{m \times n}$ so dass

$$A = QR.$$

b) Die ersten n Spalten von Q bilden eine Orthonormalbasis von Bild(A).

c) Die ersten n Spalten von Q und das obere Dreieck von R stimmen mit der schmalen QR-Zerlegung aus Satz 11.3 überein und sind somit eindeutig bestimmt.

Beweis. Der Beweis für die Zerlegung ist konstruktiv.

Es sei $A = [a_1, \ldots, a_n]$ die Spaltendarstellung von A. Im ersten Schritt $k = 1$ suchen wir einen Vektor v_1 so dass

$$Q_{v_1} A = \begin{bmatrix} \beta_1 & * & \ldots & * \\ 0 & * & \ldots & * \\ \vdots & & & \\ 0 & * & \ldots & * \end{bmatrix},$$

also $Q_{v_1} a_1 = \beta_1 e_1$ mit dem Koordinatenenheitsvektor $e_1 = (1, 0, \ldots, 0)^T$. Dies erreichen wir nach Gleichung (12.1) mit $v_1 = a_1 + \operatorname{sgn}((a_1, e_1)_2)\|a_1\|e_1$ und dann gilt $\beta_1 = -\operatorname{sgn}((a_1, e_1)_2)\|a_1\|_2$ wegen Formel (12.2).

Zu Beginn von Schritt $k > 1$ nehmen wir an, dass schon gilt

$$Q_{v_{k-1}} \cdots Q_{v_1} A = \begin{bmatrix} R^{(k-1)} & B^{(k-1)} \\ 0 & C^{(k-1)} \end{bmatrix}$$

mit einer oberen Dreiecksmatrix $R^{(k-1)}$ der Größe $k-1 \times k-1$ und $C^{(k-1)} = [c_1, \ldots, c_{n-k+1}]$ einer $m-k+1 \times n-k+1$ Matrix. Die Spalte c_1 kann nicht Null sein sonst wäre der Rang von A nicht n. Es sei nun $\tilde{v}_k \in \mathbb{R}^{m-k+1}$ so gewählt, dass $Q_{\tilde{v}_k} c_1 = \beta_k e_1$ mit dem Koordinateneinheitsvektor e_1 der Länge $m-k+1$. Dies erreicht man wie oben mit $\tilde{v}_k = c_1 + \operatorname{sgn}((c_1, e_1)_2)\|c_1\|e_1$ und $\beta_k = -\operatorname{sgn}((c_1, e_1)_2)\|c_1\|_2$. Mit dem durch $k-1$ Nullen ergänzten Vektor

$$(v_k)_{i-k+1} = \begin{cases} 0 & i < k \\ (\tilde{v}_k)_i & k \le i \le m \end{cases}$$

ist dann

$$Q_{v_k} = \begin{bmatrix} I & 0 \\ 0 & Q_{\tilde{v}_k} \end{bmatrix}$$

ebenfalls Householdermatrix und es gilt

$$Q_{v_k} Q_{v_{k-1}} \cdots Q_{v_1} A = \begin{bmatrix} I & 0 \\ 0 & Q_{\tilde{v}_k} \end{bmatrix} \begin{bmatrix} R^{(k-1)} & B^{(k-1)} \\ 0 & C^{(k-1)} \end{bmatrix} = \begin{bmatrix} R^{(k-1)} & B^{(k-1)} \\ 0 & \tilde{C}^{(k-1)} \end{bmatrix}$$

und

$$\begin{aligned} \tilde{C}^{(k-1)} = Q_{\tilde{v}_k} C^{(k-1)} &= \left(I - 2\frac{\tilde{v}_k \tilde{v}_k^T}{\tilde{v}_k^T \tilde{v}_k} \right) C^{(k-1)} \\ &= C^{(k-1)} - \frac{2}{\|\tilde{v}_k\|_2^2} \tilde{v}_k (\tilde{v}_k^T C^{(k-1)}). \end{aligned} \tag{12.3}$$

$\tilde{C}^{(k-1)}$ wird somit wie in der LU-Zerlegung durch ein Rang-1-Update berechnet. Da die erste Spalte von $\tilde{C}^{(k-1)}$ nach Konstruktion die Gestalt $\beta_k e_1$ hat gilt nun

$$\begin{bmatrix} R^{(k-1)} & B^{(k-1)} \\ 0 & \tilde{C}^{(k-1)} \end{bmatrix} = \begin{bmatrix} R^{(k)} & B^{(k)} \\ 0 & C^{(k)} \end{bmatrix}$$

mit einer nun $k \times k$ oberen Dreiecksmatrix $R^{(k)}$.

Nach n Schritten endet das Verfahren mit einer oberen Dreiecksmatrix R der geforderten Dimension. Für die Hauptdiagonalelemente von R gilt $r_{k,k} = \beta_k$. Jede der Matrizen Q_{v_k} ist Householder-Matrix, also orthogonal und symmetrisch. Mit

$$Q = Q_{v_1} \cdots Q_{v_n}$$

erhalten wir die Zerlegung $A = QR$.

Für b) betrachten wir Q und R in Blockform und zwar

$$Q = \begin{bmatrix} \tilde{Q} & \hat{Q} \end{bmatrix}, \qquad R = \begin{bmatrix} \tilde{R} \\ 0 \end{bmatrix},$$

wobei $\tilde{Q}$ die ersten n Spalten und $\hat{Q}$ die folgenden $m - n$ Spalten von Q sind. Entsprechend enthält $\tilde{R}$ die ersten n Zeilen von R. Wegen der oberen Dreiecksgestalt von R sind die folgenden $m - n$ Zeilen Null. Mit der Blockzerlegung gilt

$$A = QR = \begin{bmatrix} \tilde{Q} & \hat{Q} \end{bmatrix} \begin{bmatrix} \tilde{R} \\ 0 \end{bmatrix} = \tilde{Q}\tilde{R}.$$

Damit bilden, wie behauptet, die ersten n Spalten von Q, also $\tilde{Q}$ eine Basis von Bild(A). Wegen der Orthogonalität von Q gilt

$$Q^T Q = I \quad \Leftrightarrow \quad \begin{bmatrix} \tilde{Q}^T \\ \hat{Q}^T \end{bmatrix} \begin{bmatrix} \tilde{Q} & \hat{Q} \end{bmatrix} = \begin{bmatrix} \tilde{Q}^T\tilde{Q} & \tilde{Q}^T\hat{Q} \\ \hat{Q}^T\tilde{Q} & \hat{Q}^T\hat{Q} \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ 0 & I_{m-n} \end{bmatrix}.$$

Die Spalten von $\tilde{Q}$ bilden somit eine Orthonormalbasis von Bild(A). Die Spalten von $\hat{Q}$ sind eine Orthonormalbasis von Bild$(A)^\perp$.

Zu c). Die Dimensionen von $\tilde{Q}$ und $\tilde{R}$ stimmen mit denen aus der schmalen QR-Zerlegung aus Satz 11.3 überein. Dort wurde gezeigt dass diese Zerlegung eindeutig ist. Somit stimmen $\tilde{Q}$ und $\tilde{R}$ mit der schmalen QR-Zerlegung überein. □

12.3 Bemerkungen zur Implementierung

In der Praxis wird die Matrix Q nie explizit aufgestellt. Stattdessen merkt man sich nur die Householdervektoren $V = [v_1, \ldots, v_n]$ und kann damit die Anwendung von Q jederzeit realisieren. Dies kostet nicht nur weniger Speicher sondern ist auch schneller, da die Anwendung einer Matrix Q_v nur Skalarprodukte und ein Vektorupdate erfordert. Im Laufe des Algorithmus überschreibt man die Matrix A in der Regel. V ist untere Dreiecksmatrix und kann, wie in der LU-Zerlegung, in den entsprechenden Einträgen von A gespeichert werden die zu Null werden. Das strikte obere Dreieck enthält am Ende Einträge von R. Im Gegensatz zur LU-Zerlegung enthalten jedoch sowohl V als auch R Einträge auf der Diagonalen. Daher speichert man die Diagonale von R, also die Faktoren β_k, in einem zusätzlichen Vektor ab.

Die Multiplikation mit Q bzw. Q^T wird mittels Multiplikation mit den einzelnen Faktoren Q_{v_k} realisiert. Man beachte dabei, dass

$$Q^T = (Q_{v_1} \cdots Q_{v_n})^T = Q_{v_n}^T \cdots Q_{v_1}^T = Q_{v_n} \cdots Q_{v_1},$$

es dreht sich also gerade die Reihenfolge der Anwendung der Householdermatrizen um. Multiplikation mit Q_{v_k} wird durch explizite Ausnutzung der Rang-1-Gestalt realisiert:

$$Q_{v_k}x = \left(I - 2\frac{vv^T}{v^Tv}\right)x = x - 2\frac{v^Tx}{v^Tv}v$$

und benötigt also nur Skalarprodukte und Vektoroperationen und *keine* Matrix-Vektor-Multiplikation!

Bemerkung 12.4. Der Rechenaufwand für die QR-Zerlegung einer $n \times n$ Matrix ist asymptotisch doppelt so hoch wie für die LU-Zerlegung, also

$$N_{QR}(n) = \frac{4}{3}n^3 + O(n^2).$$

Denn der Aufwand wird bestimmt durch das Rang-1-Update aus Gleichung (12.3). Dort bildet man zuerst den Zeilenvektor $w_k^T = \tilde{v}_k^T C^{(k-1)}$ mit Aufwand $2(n-k)^2$, skaliert mit $\frac{2}{\|\tilde{v}_k\|_2^2}$ mit Aufwand $O(n-k)$ und führt dann das Rang-1-Update $v_k w_k^T$ mit Aufwand $2(n-k)^2$ durch.

Für die Speicherung der QR-Zerlegung wird bei Überschreiben von A nur zusätzlicher Speicher für einen Vektor der Länge n benötigt. □

Die QR-Zerlegung hat mehrere Anwendungen:

a) Lösung der Normalengleichung. Berechne die QR-Zerlegung. Extrahiere daraus die schmale $\tilde{Q}\tilde{R}$-Zerlegung nutze $A^TAx = A^Tb \Rightarrow \tilde{R}^T\tilde{R}x = \tilde{Q}^T\tilde{R}^Tb$. Dies zeigt, dass man dieselbe rechte Seite wie bei der schmalen QR-Zerlegung erhält. Allerdings ist $\tilde{Q}$ *nicht* verfügbar! Praktisch berechnen wird man R^TQ^Tb !

b) Lösung eines (regulären) linearen Gleichungssystemes. Berechne die QR-Zerlegung. $Ax = b \Rightarrow QRx = b \Rightarrow Rx = Q^Tb$, d.h. man muss Q^T anwenden und rückwärts einsetzen.

c) In der Berechnung von Eigenwerten einer Matrix mittels sogenannter QR-Iteration.

Vorlesung 13

Iterative Lösung linearer Gleichungssysteme

Bisher haben wir Methoden der linearen Algebra betrachtet, insbesondere die Lösung linearer Gleichungssysteme. Für den Rest der Vorlesung wollen wir uns Methoden zuwenden die mehr dem Bereich der Analysis zuzurechnen sind. Insbesondere spielt dabei oft die Annäherung von Größen mittels Folgen eine Rolle.

Zum Übergang in den Bereich der Analysis wollen wir die Lösung von linearen Gleichungssystemen $Ax = b$ mittels iterativer Verfahren betrachten. Diese Verfahren konstruieren, ausgehend von einem Startwert x^0, eine Folge

$$x^0, x^1, x^2, \dots$$

so dass gilt

$$\lim_{k\to\infty} x^k = x.$$

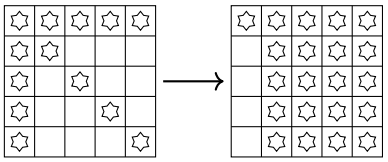
Beachte: hier ist k ein Iterationsindex und keine Potenz (welche für Vektoren auch keinen Sinn macht). Die Elemente der Folge heißen auch Iterierte.

13.1 Motivation

Zunächst ist die Frage naheliegend warum man Iterationsverfahren überhaupt in Betracht ziehen sollte. Schließlich haben wir oben eine Reihe guter Verfahren kennen gelernt welche uns die Lösung linearer Gleichungssysteme $Ax = b$ ermöglichen.

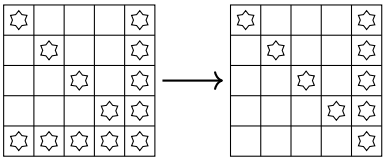
Hauptmotivation für Iterationsverfahren zur Lösung von linearen Gleichungssystemen liegt im Umgang mit Matrizen welche viele Nullen enthalten. Betrachte dazu die Matrix A in der Zeichnung unten. A sei regulär und werde ohen Pivotisierung LU-zerlegt. Die Sterne bedeuten dabei Elemente ungleich Null, alle anderen Elemente der Matrix sind Null.

Die Elimination der ersten Spalte in A führt dazu, dass alle Elemente in den Spalten $2, \dots, n$ potentiell ungleich Null werden, Ein Phänomen welches man „fill-in“ nennt. Dabei sollte man sich die Matrix $n \times n$-Matrix A beliebig groß mit der angedeuteten Pfeilform vorstellen. Dann enthält A zu Beginn $3n - 2$ Nichtnullelemente. Nach einem Schritt hat A dann n^2-n+1 Nichtnullelemente.

Matrix A 

Betrachte nun die Matrix B welche aus A durch vertauschen der ersten und letzten Zeile, sowie der ersten und letzten Spalte hervorgeht. Wir erkennen, dass in diesem Fall keinerlei fill-in entsteht. Dieses Beispiel ist sicher extrem gewählt. Für viele in praktischen Anwendungen auftretende Matrizen entsteht

jedoch in der Regel so viel fill-in, dass der dadurch entstehende Speicher- und Rechenaufwand sehr hoch ist und iterative Verfahren ab einer gewissen Größe n besser geeignet sind.

Matrix B 

Wie lässt sich nun die Nullstruktur in einer Matrix mittels iterativer Verfahren ausnutzen? Wir werden im folgenden sehen, dass iterative Verfahren im wesentlichen mit Matrix-Vektor-Produkten $y = Ax$ arbeiten. Dabei muss mit Nullen gar nicht erst gerechnet werden.

Definition 13.1 (Graph einer Matrix)**.** Es sei $A \in \mathbb{K}^{n\times n}$ eine Matrix. Dann ist $G(A) = (I(A), E(A))$ der *Graph von* A wenn gilt

$$I(A) = \{1, \ldots, n\}, \qquad E(A) = \{(i,j) \in I \times I \,:\, a_{i,j} \neq 0\}.$$

Mit $E_i(A) = \{(\alpha, \beta) \in E(A) \,:\, \alpha = i\}$ bezeichnen wir die Indizes der Nichtnullelemente der i-ten Zeile von A.

Definition 13.2 (Dünnbesetzte Matrix)**.** Sei $A \in \mathbb{K}^{n\times n}$ Element einer Menge von Matrizen beliebiger Größe n. Dann nennt man A dünnbesetzt falls

$$|E(A)| = O(n)$$

statt n^2. In der Praxis haben dünnbesetzte Matrizen oft eine konstante maximale Zahl von Einträgen in einer Zeile, d.h. $|E_i(A)| \leq C$. □

Im Matrix-Vektor-Produkt nutzt man dann aus:

$$y_i = (Ax)_i = \sum_{j=1}^{n} a_{i,j}x_j = \sum_{(\alpha,\beta)\in E_i(A)} a_{\alpha,\beta}x_\beta,$$

d.h. der Gesamtaufwand beträgt $2|E(A)|$ Rechenoperationen statt $2n^2$ Rechenoperationen. Zur Speicherung dünnbesetzter Matrizen werden spezielle Datenstrukturen verwendet, welche in geeigneter Weise $E(A)$ sowie nur die Nichtnullelemente von A abspeichern.

13.2 Lineare Iterationsverfahren

Eine große Klasse von Iterationsverfahren fällt in die Kategorie linearer Iterationsverfahren, welche durch eine lineare Fehlerfortpflanzung charakterisiert sind.

Sei $Ax = b$ das zu lösende, reguläre Gleichungssystem mit der Lösung x. Zu einer Iterierten x^k definieren wir den Fehler e^k als

$$e^k = x - x^k. \tag{13.1}$$

Für den Fehler gilt folgende Identität:

$$Ae^k = A(x - x^k) = Ax - Ax^k = b - Ax^k =: d^k. \tag{13.2}$$

Die Größe $d^k = b - Ax^k$ bezeichnet man als *Defekt* und $Ae^k = d^k$ als *Defektgleichung*. Die Defektgleichung ist natürlich genauso schwierig zu lösen wie die ursprüngliche Gleichung. Die Idee linearer Iterationsverfahren besteht nun darin die Defektgleichung *näherungsweise* zu lösen in dem man die Matrix A durch eine einfacher zu lösende Matrix W ersetzt:

$$Wv^k = d^k \qquad \Rightarrow \qquad v^k = W^{-1}d^k.$$

Damit erhält man natürlich nicht den Fehler e^k sondern eine Nährung des Fehlers v^k. Aus Gleichung (13.1) entnehmen wir $x = x^k + e^k \approx x^k + v^k$ und somit das Iterationsverfahren

$$x^{k+1} = x^k + v^k = x^k + W^{-1}d^k = x^k + W^{-1}(b - Ax^k). \tag{13.3}$$

Wir zerlegen die zu lösende Matrix in $A = L + D + U$, wobei L eine untere linke Dreiecksmatrix, D die Diagonale und U eine obere rechte Dreiecksmatrix ist. Konkrete Beispiele zur Wahl von W sind in dieser Notation

$$\begin{aligned} W_R &= \frac{1}{\omega}I,\ \omega \in \mathbb{R} && \text{(Richardson-Verfahren)},\\ W_{Jac} &= D && \text{(Jacobi-Verfahren)},\\ W_{GS} &= L + D && \text{(Gauß-Seidel Verfahren)}. \end{aligned}$$

Alternativ können wir statt W^{-1} auch eine Matrix B einsetzen und so die Iteration

$$x^{k+1} = x^k + B(b - Ax^k) \tag{13.4}$$

definieren. In diesem Zusammenhang heißt B oft auch *Vorkonditionierer*. Insbesondere erlaubt diese Darstellung auch, dass B singulär sein kann. Man sieht auch unmittelbar, dass die Lösung x einen *Fixpunkt* der Iteration darstellt, denn es gilt $x + B(b - Ax) = x$.

Die Konvergenzanalyse basiert auf der Fehlerfortpflanzung

$$\begin{aligned} e^{k+1} &= x - x^{k+1} = x - x^k - B(b - Ax^k) = e^k - B(Ax - Ax^k)\\ &= e^k - BAe^k = (I - BA)e^k. \end{aligned}$$

Die Matrix

$$S = I - BA$$

nennt man *Iterationsmatrix* und es gilt für den Fehler nach k Schritten:

$$e^k = Se^{k-1} = S^k e^0.$$

Aus der Fehlerfortpflanzung erkennen wir, dass sowohl A als auch B regulär sein müssen, sonst kann das Iterationsverfahren nicht für beliebigen Startwert konvergieren. Sei etwa A singulär. Dann dann existiert $0 \neq z \in \text{Kern}(A)$, d.h. $Az = 0$ und damit $Sz = (I - BA)z = z$. Somit kann der Fehler z nicht korrigiert werden, das Verfahren stagniert bei diesem Fehler. Analog sei B singulär und A regulär, dann gibt es $0 \neq z = Ay \in \text{Kern}(B)$ so dass $Sy = (I - BA)y = y - Bz = y$.

Schließlich können wir das lineare Iterationsverfahren auch noch in einer alternativen Form schreiben:

$$\begin{aligned} x^{k+1} &= x^k + B(b - Ax^k) = x^k + Bb - BAx^k\\ &= Sx^k + Bb. \end{aligned} \tag{13.5}$$

13.3 Konvergenz linearer Iterationsverfahren

Die Konvergenz linearer Iterationsverfahren hängt entscheidend mit den Eigenwerten der Iterationsmatrix S zusammen. Für diagonalsierbare Matrizen sieht man das unmittelbar ein: Sei $S = T^{-1}DT$ mit der Diagonalmatrix $D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$. Dann gilt $S^k = (T^{-1}DT)^k = T^{-1}D^kT$ und $D^k = \operatorname{diag}(\lambda_1^k, \ldots, \lambda_n^k)$ und dass Verfahren konvergiert falls $|\lambda_i| < 1$ für alle $1 \leq i \leq n$, also $\varrho(S) < 1$. Dass dieser Zusammenhang auch für nicht diagonalisierbare Matrizen gilt benötigt etwas Vorbereitung.

Satz 13.3 (Schur-Normalform)**.** Es sei $A \in \mathbb{K}^{n\times n}$ mit $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$. Dann gibt es eine unitäre Matrix Q sowie eine obere Dreiecksmatrix R so dass gilt

$$A = QR\bar{Q}^T.$$

Beweis. Durch Induktion über n. Für $n = 1$ gilt die Aussage mit $Q = 1$ und $A = R$.

Die Aussage sei bis $n-1$ bewiesen und es sei $A \in \mathbb{K}^{n\times n}$. Dann Wähle einen Eigenwert λ mit einem zugehörigen Eigenvektor e. So ein Eigenpaar existiert immer, dann nach dem Fundamentalsatz der Algebra hat das charakteristische Polynom von A genau n Nullstellen und zu einer solchen Nullstelle λ ist $A-\lambda I$ singulär und hat einen nichttrivialen Kern.

Nun setze $x_1 = e/\|e\|_2$, ergänze x_1 mit $x_2, \ldots, x_n$ zu einer Orthonormalbasis des $\mathbb{K}^n$ und setze $X = [x_1, \ldots, x_n]$. Mit der Matrix X erhalten wir die 2×2 Blockzerlegung

$$\bar{X}^T AX = \bar{X}^T A[x_1, \ldots, x_n] = \bar{X}^T[\lambda x_1, Ax_2, \ldots Ax_n] = \begin{bmatrix} \lambda & a^T \\ 0 & B \end{bmatrix},$$

mit dem $n-1$-Vektor a und der $n-1 \times n-1$-Matrix B. Nach Induktionsvoraussetzung gibt es nun eine unitäre Matrix P, die B auf obere Dreiecksgestalt transformiert: $\bar{P}^T BP = U$. Damit gilt nun

$$\begin{bmatrix} 1 & 0 \\ 0 & \bar{P}^T \end{bmatrix} \begin{bmatrix} \lambda & a^T \\ 0 & B \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & P \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \bar{P}^T \end{bmatrix} \begin{bmatrix} \lambda & a^T P \\ 0 & BP \end{bmatrix} = \begin{bmatrix} \lambda & a^T P \\ 0 & \bar{P}^T BP \end{bmatrix} = \begin{bmatrix} \lambda & a^T P \\ 0 & \bar{U} \end{bmatrix}.$$

Somit haben wir gezeigt dass

$$\begin{bmatrix} 1 & 0 \\ 0 & \bar{P}^T \end{bmatrix} \bar{X}^T AX \begin{bmatrix} 1 & 0 \\ 0 & P \end{bmatrix} = \begin{bmatrix} \lambda & a^T P \\ 0 & \bar{U} \end{bmatrix},$$

also

$$A = \underbrace{X \begin{bmatrix} 1 & 0 \\ 0 & P \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} \lambda & a^T P \\ 0 & \bar{U} \end{bmatrix}}_{R} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & \bar{P}^T \end{bmatrix} \bar{X}^T}_{\bar{Q}^T}.$$

Hier haben wir ausgenutzt dass ein Produkt unitärer Matrizen stets wieder unitär ist. □

Lemma 13.4. Es sei $R \in \mathbb{K}^{n\times n}$ eine obere Dreiecksmatrix mit $|r_{i,i}| \leq \varrho < 1$. Dann gilt für die Einträge der m-ten Potenz:

$$\left|(R^m)_{i,j}\right| = O\left(\varrho^{\max(m-j+i,0)}\right), \qquad j \geq i.$$

Beweis. Wir definieren die Notation $|a|_+ = \max(a, 0)$ und beweisen mittels Induktion über den Exponenten m. Für $m = 1$ gilt nach Voraussetzung $|r_{i,i}| \leq \varrho < 1$, d.h. $\left|\left(R^1\right)_{i,i}\right| = O(\varrho)$ für $i = j$ und $\left|\left(R^1\right)_{i,j}\right| = O(1)$ sonst.

Sei nun die Aussage für R^{m-1} bewiesen. Dann ist $R^m = RR^{m-1} = RU$ mit der Abkürzung $U = R^{m-1}$ Dann rechnen wir (mit etwas missbrauchter Notation):

$$\begin{aligned}\left|(R^m)_{i,j}\right| &= \sum_{k=1}^{n} r_{i,k}u_{k,j} = \sum_{k=i}^{j} r_{i,k}u_{k,j} = r_{i,i}u_{i,j} + \sum_{k=i+1}^{j} r_{i,k}u_{k,j} \\ &= O(\varrho)O(\varrho^{|m-1-j+i|_+}) + \sum_{k=i+1}^{j} O(1)O(\varrho^{|m-1-j+k|_+}) \\ &= O(\varrho^{|m-j+i|_+}) + O(\varrho^{|m-j+i|_+}) = O(\varrho^{|m-j+i|_+}).\end{aligned}$$

□

Korollar 13.5. Es sei $R \in \mathbb{K}^{n\times n}$ eine obere Dreiecksmatrix mit $|r_{i,i}| \leq \varrho < 1$, dann gilt

$$\lim_{m\to\infty} R^m = 0.$$

Ist R strikte obere Dreiecksmatrix dann gilt $R^m = 0$ für alle $m \geq n$.

Beweis. Die erste Aussage folgt aus Lemma 13.4 da $\lim_{m\to\infty} \varrho^{m-j+i} = 0$ für $|\varrho| < 1$. Im Fall $\varrho = 0$ ergibt der Beweis von Lemma 13.4, dass $(R^m)_{i,j} = 0$ wenn $m - j + i > 0$, also $m > j - i$. Da $j - i$ maximal $n - 1$ sein kann gilt $R^m = 0$ für $m \geq n$. □

Satz 13.6. $A, B \in \mathbb{R}^{n\times n}$ seien regulär. Das lineare Iterationsverfahren $x^{k+1} = x^k + B(b - Ax^k)$ konvergiert für jeden Startwert x^0 gegen die Lösung von $Ax = b$ genau dann wenn $\varrho(S) = \varrho(I - BA) < 1$. $\varrho(S)$ heißt *Konvergenzrate*.

Beweis. Für den Fehler $e^k = x - x^k$ in Schritt k gilt wie oben gezeigt $e^k = Se^{k-1} = S^k e^0$ mit dem Startfehler $e^0 = x - x^0$. Konvergenz der Folge $\{x^k\}$ gegen die Lösung x ist also äquivalent dazu, dass $\{e^k\}$ Nullfolge ist.

a) Wir zeigen zunächst, dass aus der Konvergenz der Folge der Iterierten gegen die Lösung x für jeden Startwert folgt, dass $\varrho(S) < 1$ gelten muss. Nach Voraussetzung gilt $0 = \lim_{k\to\infty} e^k = \lim_{k\to\infty} S^k e^0$ für jedes e^0. Falls nun $\varrho(S) \geq 1$ wäre, könnte man einen Eigenvektor z zu $|\lambda| \geq 1$ wählen und damit wäre $\lim_{k\to\infty} S^k z = \lim_{k\to\infty} \lambda^k z \neq 0$ wegen $|\lambda| \geq 1$ im Widerspruch zur Voraussetzung.

b) Nun zeigen wir, dass aus $\varrho(S) < 1$ Konvergenz der Folge der Iterierten folgt. Nach Satz 13.3 gibt es eine unitäre Matrix Q und eine obere Dreiecksmatrix R so dass gilt $S = QR\bar{Q}^T$ und $r_{i,i}$ sind die Eigenwerte der Matrix S, also gilt nach Vorraussetzung $|r_{i,i}| \leq \varrho(S) < 1$. Damit haben wir

$$\lim_{k\to\infty} S^k = \lim_{k\to\infty} \left(QR\bar{Q}^T\right)^k = QR^k\bar{Q}^T = 0$$

nach Lemma 13.4 und Korollar 13.5.

□

Satz 13.6 gibt zwar eine vollständige Charakterisierung der Konvergenz linearer Iterationsverfahren, ist aber in der Praxis schwierig einsetzbar da die Eigenwerte der Iterationsmatrix nicht einfach zugänglich sind.

Vorlesung 14

Iterative Lösung linearer Gleichungssysteme II

14.1 Symmetrisch positiv definite Matrizen

In Satz 13.6 haben wir ein allgemeines Konvergenzresultat für lineare Iterationsverfahren bewiesen. Der Spektralradius der Iterationsmatrix S ist im allgmeinen jedoch schwer zugänglich, außer im symmetrisch positiv definiten Fall wie folgender Satz zeigt.

Satz 14.1. $A, B \in \mathbb{R}^{n\times n}$ seien symmetrische und positive definite Matrizen. Dann konvergiert die Iteration

$$x^{k+1} = x^k + \frac{1}{\lambda_{\max}(BA)} B(b - Ax^k)$$

mit der Konvergenzrate

$$\varrho = 1 - \frac{1}{\kappa_2(BA)} = 1 - \frac{\lambda_{min}(BA)}{\lambda_{\max}(BA)}.$$

Beweis. A ist symmetrisch und positiv definit, somit existiert eine orthogonale Transformationsmatrix welche A auf Diagonalgestalt transformiert: $Q^T AQ = D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ mit $0 < \lambda_1 \leq \ldots \leq \lambda_n$. Wir definieren die Wurzeln $D^{\frac{1}{2}} = \operatorname{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_n})$ und $A^{\frac{1}{2}} = QD^{\frac{1}{2}}Q^T$. Dann haben die Matrizen BA und $T = A^{\frac{1}{2}}BAA^{-\frac{1}{2}} = A^{\frac{1}{2}}BA^{\frac{1}{2}}$ dieselben Eigenwerte (Ähnlichkeitstransformation). $T = A^{\frac{1}{2}}BA^{\frac{1}{2}}$ ist symmetrisch und positiv definit und daher diagonalisierbar mit den reellen, positiven Eigenwerten $0 < \mu_1 \leq \ldots \leq \mu_n$ (die auch die Eigenwerte von BA sind).

Nun betrachten wir die Iterationsmatrix $S = I - \omega BA$, zunächst mit einem frei wählbaren Dämpfungsfaktor $\omega > 0$. Nun gilt $A^{\frac{1}{2}}SA^{-\frac{1}{2}} = I - \omega T$ und daher entspricht jeder Eigenwert μ_i von T einem Eigenwert $1 - \omega\mu_i$ von S. Ordnet man diese der Größe nach an gilt:

$$\lambda_{\min}(S) = 1 - \omega\mu_n \leq \ldots \leq 1 - \omega\mu_1 = \lambda_{\max}(S) < 1.$$

Der größte Eigenwert ist in jedem Fall kleiner als 1 da $\mu_i > 0$ und $\omega > 0$. Mit der Wahl

$$\omega = \frac{1}{\mu_n} = \frac{1}{\lambda_{\max}(T)} = \frac{1}{\lambda_{\max}(BA)}$$

gilt $\lambda_{\min}(S) = 0$ und

$$\varrho(S) = 1 - \frac{\mu_1}{\mu_n} = \frac{\lambda_{\min}(BA)}{\lambda_{\max}(BA)} = 1 - \frac{1}{\kappa_2(BA)}.$$

Man könnte ω auch so wählen, dass $1 - \omega\mu_n - (-1) = 1 - (1 - \omega\mu_1)$, also $\omega = \frac{2}{\mu_n + \mu_1}$ und erhält damit die etwas bessere Konvergenzrate $\varrho = 1 - \frac{2\mu_1}{\mu_n + \mu_1}$. □

Der Satz zeigt, dass die Konvergenzrate mit steigender Konditionszahl schlechter wird. Die Aufgabe der Matrix B ist es die Kondition von BA so klein wie möglich zu machen, daher der Name „Vorkonditionierer“. Allerdings soll auch der Aufwand zur Anwendung von B möglichst gering sein. Daher scheidet $B = A^{-1}$ mit $\kappa_2(BA) = \kappa_2(I) = 1$ aus.

Der Satz benötigt immer noch eine Abschätzung des größten Eigenwertes einer Matrix. Dazu dient der folgende Satz.

Satz 14.2 (Gerschgorin Kreise)**.** Sei $A \in \mathbb{R}^{n\times n}$ und $\bar{B}(z,r) = \{\xi \in \mathbb{C} : |z-\xi| \le r\}$. Dann gilt für alle Eigenwerte λ von A:

$$\lambda \in \bigcup_{k\in I_n=\{1,\dots,n\}} \bar{B}\left(a_{k,k}, r_k\right), \qquad r_k = \sum_{j\in I_n, j\neq k} |a_{k,j}|.$$

Beweis. Es sei λ ein Eigenwert von A mit zughörigem, normiertem Eigenvektor e, $\|e\|_\infty = 1$. Dann gibt es ein $k \in I_n = \{1,\dots,n\}$ mit $|e_k| = 1$ und es gilt in Zeile k:

$$(Ae)_k = \sum_{j\in I_n} a_{k,j}e_j = \lambda e_k \quad \Leftrightarrow \quad (\lambda - a_{k,k})e_k = \sum_{j\in I_n, j\neq i} a_{k,j}e_j.$$

Betrag bilden liefert

$$|\lambda - a_{k,k}||e_k| = |\lambda - a_{k,k}| = \left|\sum_{j\in I_n, j\neq i} a_{k,j}e_j\right| \le \sum_{j\in I_n, j\neq k} |a_{k,j}||e_j| = \sum_{j\in I_n, j\neq k} |a_{k,j}|.$$

Da der Eigenwert λ beliebig gewählt war gilt diese Aussage für alle Eigenwerte simultan und damit folgt die Behauptung. □

Mit dem letzten Satz erhält man eine Abschätzung des Spektralradius einer Matrix als

$$\varrho(A) \le \max\left\{\left|a_{k,k} \pm \sum_{1\le j\le n, j\neq k} |a_{k,j}|\right| \;:\; 1 \le k \le n\right\}.$$

Diese ist insbesondere für dünnbesetzte Matrizen relativ gut. Aus dem Beweis von Satz 14.1 ersieht man ebenfalls, dass die Iteration $x^{k+1} = x^k + \omega B(b - Ax^k)$ für alle $\omega \le \lambda_{\max}^{-1}(BA)$ konvergiert. Man kann kann den größten Eigenwert also auch überschätzen (natürlich wird dann die Konvergenzrate schlechter).

Damit haben wir zumindest mit dem Richardson- und dem Jacobi-Verfahren im symmetrisch positiv definiten Fall zwei praktikable Verfahren. Im Gauß-Seidel Verfahren ist B unsymmetrisch und Satz 14.1 ist nicht anwendbar und wir müssen einen anderen Weg wählen.

Neben einer direkten Abschätzung des Spektralradius der Iterationsmatrix ist die Abschätzung einer Norm der Iterationsmatrix ein anderer Ansatz für einen Konvergenzbeweis:

$$e^k = Se^{k-1} \qquad \Rightarrow \qquad \|e^k\| \le \|S\|\|e^{k-1}\| \le \|S\|^k\|e^{k-1}\|.$$

Ziel ist es nun $\|S\| < 1$ und damit Konvergenz zu zeigen. Da jede Norm den Spektralradius überschätzt, d.h. $\varrho(S) \le \|S\|$, ist die Bedingung $\|S\| < 1$ zwar hinreichend aber nicht notwendig für die Konvergenz. Insbesondere ist nun entscheidend in *welcher Norm* man die Untersuchung durchführt.

Im Fall symmetrisch positiv definiter (reeller) Matrizen ist mittels

$$(x, y)_A = (x, Ay)_2 \tag{14.1}$$

ein Skalarprodukt erklärt welches die sogennante *Energienorm* impliziert:

$$\|x\|_A = \sqrt{(x, x)_A} = \sqrt{(x, Ax)_2}. \tag{14.2}$$

Mit Hilfe dieser Vektornorm ist schließlich die zugeordnete Matrixnorm erklärt:

$$\|S\|_A = \sup_{x \neq 0} \frac{\|Sx\|_A}{\|x\|_A}. \tag{14.3}$$

Der nun folgende Satz schätzt die Norm der Iterationsmatrix des Gauß-Seidel-Verfahrens in der Energienorm ab.

Satz 14.3. Es sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit. Dann konvergiert das Gauß-Seidel Verfahren.

Beweis. Für das Gauß-Seidel-Verfahren zerlegen wir $A = L + D + U$ in striktes unteres Dreieck, Diagonale und striktes oberes Dreieck. Wegen der Symmetrie von A gilt $U = L^T$ und wegen der positven Definitheit von A ist dessen Diagonale D auch positiv definit (hat positive Einträge). Für die Matrix $W = L + D$ gilt

$$\begin{aligned}(x, (W + W^T)x)_2 = (x, (L + D + L^T + D)x)_2 = (x, Ax)_2 + (x, D, x)_2 \\ > (x, Ax)_2 > 0 \quad \forall x \neq 0.\end{aligned}$$

Betrachte das lineare Iterationsverfahren $x^{k+1} = x^k + W^{-1}(b - Ax^k)$. Wir zeigen nun: Aus $(x, (W + W^T)x)_2 > (x, Ax)_2 > 0$ für alle $x \neq 0$ folgt $\|S\|_A = \|I - W^{-1}A\|_A < 1$.

$S = I - W^{-1}A$ ist ähnlich zu $\hat{S} = A^{\frac{1}{2}} S A^{-\frac{1}{2}} = A^{\frac{1}{2}}(I - W^{-1}A)A^{-\frac{1}{2}} = I - A^{\frac{1}{2}} W^{-1} A^{\frac{1}{2}}$. Nun zeigen wir

$$\begin{aligned}\|S\|_A &= \sup_{x \neq 0} \frac{\|Sx\|_A}{\|x\|_A} = \sup_{x \neq 0} \left(\frac{(Sx, ASx)_2}{(x, Ax)_2} \right)^{\frac{1}{2}} = \sup_{x \neq 0} \left(\frac{(A^{\frac{1}{2}} S A^{-\frac{1}{2}} A^{\frac{1}{2}} x, A^{\frac{1}{2}} S A^{-\frac{1}{2}} A^{\frac{1}{2}} x)_2}{(A^{\frac{1}{2}} x, A^{\frac{1}{2}} x)_2} \right)^{\frac{1}{2}} \\ &= \sup_{y = A^{\frac{1}{2}} x \neq 0} \left(\frac{(\hat{S}y, \hat{S}y)_2}{(y, y)_2} \right)^{\frac{1}{2}} = \sup_{y \neq 0} \frac{\|\hat{S}y\|_2}{\|y\|_2} = \|\hat{S}\|_2.\end{aligned}$$

Nun rechnen wir

$$\begin{aligned}\|\hat{S}x\|_2^2 &= \left((I - A^{\frac{1}{2}} W^{-1} A^{\frac{1}{2}})x, (I - A^{\frac{1}{2}} W^{-1} A^{\frac{1}{2}})x \right)_2 = \left(x, (I - A^{\frac{1}{2}} W^{-T} A^{\frac{1}{2}})^T (I - A^{\frac{1}{2}} W^{-1} A^{\frac{1}{2}})x \right)_2 \\ &= \left(x, (I - A^{\frac{1}{2}}(W^{-T} + W^{-1})A^{\frac{1}{2}} + A^{\frac{1}{2}} W^{-T} A W^{-1} A^{\frac{1}{2}})x \right)_2 \\ &= (x, x)_2 - \left(x, A^{\frac{1}{2}} W^{-T}(W + W^T) W^{-1} A^{\frac{1}{2}} x \right)_2 + \left(W^{-1} A^{\frac{1}{2}} x, A W^{-1} A^{\frac{1}{2}} x \right)_2 \\ &= (x, x)_2 - \left(W^{-1} A^{\frac{1}{2}} x, (W + W^T) W^{-1} A^{\frac{1}{2}} x \right)_2 + \left(W^{-1} A^{\frac{1}{2}} x, A W^{-1} A^{\frac{1}{2}} x \right)_2 \\ &< (x, x)_2 - \left(W^{-1} A^{\frac{1}{2}} x, (W + W^T) W^{-1} A^{\frac{1}{2}} x \right)_2 + \left(W^{-1} A^{\frac{1}{2}} x, (W + W^T) W^{-1} A^{\frac{1}{2}} x \right)_2 \\ &= \|x\|_2^2.\end{aligned}$$

Mit $\|\hat{S}x\|_2 < \|x\|_2$ erhalten wir $\|S\|_A = \|\hat{S}\|_2 < 1$. □

14.2 Diagonaldominante Matrizen

Es zeigt sich, dass die Maximumnorm $\|.\|_\infty$ für diagonaldominante Matrizen gut geeignet ist. In Definition 10.3 wurden schon diagonaldominante Matrizen eingeführt. Allerdings erlaubte diese Definition auch singuläre Matrizen, weshalb sie für die Untersuchung konvergenter Iterationsverfahren nicht geeignet ist. Wir benötigen also erst einmal eine besser geeignete Definition.

Definition 14.4. Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt *reduzibel* wenn es eine Zerlegung der Indexmenge $I = \{1, \ldots, n\}$ in zwei Teile gibt $I = I_1 \cup I_2$, $I_1 \cap I_2 = \emptyset$ so dass $a_{i,j} = 0$ für alle $(i, j) \in (I_1 \times I_2)$. Ist dies nicht der Fall, dann heißt die Matrix *irreduzibel*. □

Eine reduzible Matrix A kann durch Zeilen- und Spaltenvertauschungen so transformiert werden, dass gilt

$$P^T AP = \begin{pmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{pmatrix}.$$

Dabei entspricht die erste Blockzeile gerade den Zeilen mit Indizes in I_1 und die zweite Zeile I_2.

Definition 14.5. Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt *irreduzibel diagonaldominant* falls (alle) folgende Bedingungen gelten:

i) A ist irreduzibel.

ii) Für alle $1 \le i \le n$ gilt $\sum_{1\le j\le n, j\neq i} |a_{i,j}| \le |a_{i,i}|$, und

iii) für mindestens einen Index i gilt $\sum_{1\le j\le n, j\neq i} |a_{i,j}| < |a_{i,i}|$. □

Satz 14.6. Es sei $A \in \mathbb{R}^{n\times n}$ irreduzibel diagonaldominant. Dann konvergieren sowohl das Jacobi- als auch das Gauß-Seidel-Verfahren.

Beweis. Betrachten wir zunächst das Gauß-Seidel-Verfahren. Wir zeigen $\|S\|_\infty < 1$. Dazu zerlege erst einmal $A = L + D + U$ in unteres Dreieck, Diagonale und oberes Dreieck. Dann schreibt sich die Iterationsmatrix des Gauß-Seidel Verfahrens

$$\begin{aligned} S = I - (L + D)^{-1}A &\Leftrightarrow (L + D)S = (L + D) - A = -U \\ &\Leftrightarrow DS = -(U + LS) \\ &\Leftrightarrow S = -D^{-1}(U + LS). \end{aligned}$$

In Komponenten gilt

$$(Sx)_i = (-D^{-1}(U + LS)x)_i = -\frac{1}{a_{i,i}} \left(\sum_{j>i} a_{i,j}x_j + \sum_{j<i} a_{i,j}(Sx)_j \right)$$

und mit Beträgen und Dreiecksungleichung dann

$$|(Sx)_i| \le \frac{1}{|a_{i,i}|} \left(\sum_{j>i} |a_{i,j}||x_j| + \sum_{j<i} |a_{i,j}||(Sx)_j| \right). \tag{14.4}$$

Per Induktion über die Indexmenge zeigen wir erst $(Sx)_i \leq \|x\|_\infty$ und dann führen wir den Fall $\|S\|_\infty = 1$ zum Widerspruch. Sei nun also $i = 1$. Dann ist die zweite Summe in (14.4) leer, also

$$|(Sx)_1| \leq \frac{1}{|a_{1,1}|} \sum_{j>1} |a_{1,j}||x_j| \leq \frac{1}{|a_{1,1}|} \sum_{j>1} |a_{1,j}|\|x\|_\infty \leq \frac{\|x\|_\infty}{|a_{1,1}|} \sum_{j\neq 1} |a_{1,j}| \leq \|x\|_\infty.$$

Nun sei $(Sx)_i \leq \|x\|_\infty$ bis zum Index $i-1$ bewiesen. Dann gilt für Index i:

$$\begin{aligned} |(Sx)_i| &\leq \frac{1}{|a_{i,i}|} \left(\sum_{j>i} |a_{i,j}||x_j| + \sum_{j<i} |a_{i,j}||(Sx)_j| \right) \\ &\leq \frac{1}{|a_{i,i}|} \left(\sum_{j>i} |a_{i,j}|\|x\|_\infty + \sum_{j<i} |a_{i,j}|\|x\|_\infty \right) \\ &= \frac{\|x\|_\infty}{|a_{i,i}|} \sum_{j\neq 1} |a_{i,j}| \leq \|x\|_\infty. \end{aligned} \tag{14.5}$$

Damit gilt nun erst einmal

$$\|Sx\|_\infty = \max_{1\leq i\leq n} |(Sx)_i| \leq \|x\|_\infty$$

und damit dann

$$\|S\|_\infty = \sup_{x\neq 0} \frac{\|Sx\|_\infty}{\|x\|_\infty} \leq 1.$$

Nun nehmen wir an es sei $\|S\|_\infty = 1$, d.h. es gibt einen Vektor x so dass $\|x\|_\infty = 1$ und $\|Sx\|_\infty = 1$ sowie einen Index i so dass wegen (14.5) gilt

$$1 = |(Sx)_i| \leq \frac{1}{|a_{i,i}|} \left(\sum_{j>i} |a_{i,j}||x_j| + \sum_{j<i} |a_{i,j}||(Sx)_j| \right) \leq \|x\|_\infty = 1.$$

In beiden $\leq$ Relationen muss also jeweils die Gleichheit gelten, also gilt

$$\sum_{j>i} |a_{i,j}||x_j| + \sum_{j<i} |a_{i,j}||(Sx)_j| = |a_{i,i}|. \tag{14.6}$$

Dies ist nur möglich, wenn folgende Bedingungen alle erfüllt sind

i) $|x_j| = 1$ wenn $a_{i,j} \neq 0$ und $j > i$,

ii) $|(Sx)_j| = 1$ wenn $a_{i,j} \neq 0$ und $j < i$, sowie

iii) $\sum_{j\neq i} |a_{i,j}| = |a_{i,i}|$.

Warum? Angenommen es wäre $\sum_{j\neq i} |a_{i,j}| < |a_{i,i}|$. Da $\|x\|_\infty = 1$ und $\|Sx\|_\infty = 1$ kann somit (14.6) nicht erfüllt werden. Also muss $\sum_{j\neq i} |a_{i,j}| = |a_{i,i}|$ gelten. Angenommen es sei nun i) oder ii) nicht erfüllt. dann wäre $\sum_{j>i} |a_{i,j}||x_j| + \sum_{j<i} |a_{i,j}||(Sx)_j| < |a_{i,i}|$, also (14.6) nicht erfüllt.

Wegen der Bedingung (14.6) bzw. den dazu äquivalenten Bedingungen erhalten wir zu einem Index i mit $(Sx)_i = 1$ weitere Indizes $j \in E_i(A)$ für die gilt $(Sx)_j = 1$. Auf diese Indizes kann man das Argument wieder anwenden. Da A irreduzibel ist kann man so schlussendlich die ganze Indexmenge $I = \{1, \ldots, n\}$

erreichen und es gilt $\sum_{j\neq i}|a_{i,j}| = |a_{i,i}|$ für alle $i \in I$. Dies ist jedoch ein Widerspruch zur irreduziblen Diagonaldominanz, die $\sum_{j\neq i}|a_{i,j}| < |a_{i,i}|$ für mindestens einen Index i fordert.

Nun zum einfacheren Fall des Jacobi-Verfahrens. Für die Iterationsmatrix gilt dann

$$S = I - D^{-1}A = I - D^{-1}(L + D + U) = -D^{-1}(L + U)$$

und damit

$$|(Sx)_i| = \left|\frac{1}{a_{i,i}}\sum_{j\neq i} a_{i,j}x_j\right| \leq \frac{\|x\|_\infty}{|a_{i,i}|}\sum_{j\neq i}|a_{i,j}| \leq \|x\|_\infty.$$

Damit gilt wieder $\|S\|_\infty \leq 1$.

Den Fall $\|S\|_\infty = 1$ führt man mit den selben Argumenten wie oben zum Widerspruch, nur wird (14.6) durch die Beziehung

$$\sum_{j\neq i}|a_{i,j}||x_j| = |a_{i,i}|$$

ersetzt. □

14.3 Praktische Aspekte

Nun haben wir Konvergenz linearer Iterationsverfahren in einigen Fällen bewiesen. Allerdings sind einige der Konvergenzresultat keine Auskunft über die Konvrgenzgschwindigkeit. Für die praktische Durchführung von Iterationsverfahren fehlt nun noch ein Abbruchkriterium.

Ein naheliegender Ansatz wäre k so zu wählen, dass

$$\|e^k\| \leq \epsilon\|e^0\|$$

für eine gegebene Totalreduktion ϵ. Man fordert also eine Reduktion des Anfangsfehlers um einen gewissen Faktor.

Nun ist der Fehler in der Regel jedoch nicht einfach berechenbar und wir haben auch die Norm noch nicht weiter spezifiziert. Eine möglich Wahl ist

$$\|e^k\|^2_{A^TA} = (e^k, A^TAe^k)_2 = (Ae^k, Ae^k)_2 = (d^k, d^k)_2 = \|d^k\|_2^2 = \|b - Ax^k\|_2^2$$

womit wir eine praktisch berechenbare Größe erhalten. Dies resultiert dann in folgendem Algorithmus zur Durchführung eines linearen Iterationsverfahrens.

Algorithmus 14.7. Es seien gegeben $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, Startwert $x^0 \in \mathbb{R}^n$ sowie ein Reduktionsfaktor ϵ und eine Verfahrensmatrix $W \in \mathbb{R}^{n\times n}$. Dann kann das lineare Iterationsverfahren in folgender Weise praktisch durchgeführt werden.

1) Setze $x = x^0$, berechne $d = b - Ax$ und $n = n^0 = \|d\|_2$.

2) Falls gilt $n \leq \epsilon n^0$ beende die Iteration, sonst mache den nächsten Schritt:

3) Löse $Wv = d$

4) Setze $x = x + v$

5) Setze $d = d - Av$, (denn $d^{k+1} = b - Ax^{k+1} = b - A(x^k + v) = d^k - Av$).

6) Setze $n = \|d\|_2$ und gehe zu 2)

Der Hauptaufwand besteht zum einen in der Berechnung von v im Schritt 3 sowie der Matrix-Vektor-Multiplikation in Schritt 5.

Wir wollen das oben eingeführte Abbruchkriterium noch etwas genauer untersuchen. Welche Reduktion des Anfangsfehlers kann man in der Euklidschen Norm erwarten? Dazu betrachten wir den Fall einer symmetrisch positiv definiten Matrix A mit der Orthonormalbasis bestehend aus Eigenvektoren z_i:

$$\|e\|_{A^TA}^2 = (Ae, Ae)_2 = \left(A\sum_{i=1}^n \alpha_i z_i, A\sum_{i=1}^n \alpha_i z_i \right)_2 = \sum_{i=1}^n \alpha_i^2 \lambda_i^2.$$

Damit erhalten wir

$$\lambda_{\min}(A)\|e\|_2 \leq \|e\|_{A^TA} \leq \lambda_{\max}(A)\|e\|_2$$

und

$$\lambda_{\min}(A)\|e^k\|_2 \leq \|e^k\|_{A^TA} = \|d^k\|_2 \leq \epsilon\|d^0\|_2 = \epsilon\|e^0\|_{A^TA} \leq \lambda_{\max}(A)\|e^0\|_2.$$

Damit gilt also in der Euklidschen Norm die Abschätzung

$$\|e^k\|_2 \leq \epsilon\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}\|e^0\|_2 = \epsilon\kappa_2(A)\|e^0\|_2.$$

Für schlecht konditionierte Matrizen kann das Konvergenzkriterium $\|d^k\|_2 \leq \epsilon\|d^0\|_2$ zu einer deutlich kleineren Reduktion der Fehler in der Euklidschen Norm führen.

Dieser Effekt hängt jedoch auch mit dem Verfahren selbst zusammen. Reduziert der Vorkonditionierer alle Fehler in etwa gleich gut dann tritt der Effekt nicht auf (Betrachte z.B. $B = \omega A^{-1}$ mit $\omega < 1$). Für die Richardson-Iteration hingegen kann obiger worst-case mit einer Verringerung der Reduktion um den Faktor $\kappa_2(A)$ tatsächlich auftreten.

Wenden wir uns nun dem Gesamtaufwand zur Lösung des linearen Gleichungssystemes zu. Dieser setzt sich zusammen aus dem Aufwand pro Iteration und der Anzahl der benötigten Iteration. Diese erhalten wir aus dem Konvergenzfaktor ϱ und der geforderten Reduktion ϵ über

$$\varrho^k = \epsilon \qquad \Rightarrow \qquad k(\epsilon) = \frac{\ln\epsilon}{\ln\varrho} = \frac{|\ln\epsilon|}{|\ln\varrho|}.$$

Die letzte Identität ergibt sich aus der Tatsache dass $\epsilon < 1$ und $\varrho < 1$.

Mit dem Resultat $\varrho = 1 - \frac{1}{\kappa_2(BA)}$ aus Satz 14.1 der Taylorentwicklung

$$\ln\varrho = \ln\left(1 - \frac{1}{\kappa_2(BA)}\right) = -\frac{1}{\kappa_2(BA)} + O\left(\kappa_2^2(BA)\right)$$

erhalten wir

$$\begin{aligned} k(\epsilon) &= \frac{|\ln\epsilon|}{\frac{1}{\kappa_2(BA)} + O\left(\kappa_2^{-2}(BA)\right)} = \frac{\kappa_2(BA)|\ln\epsilon|}{1 + O\left(\kappa_2^{-1}(BA)\right)} \\ &= \kappa_2(BA)|\ln\epsilon|\left(1 - O\left(\kappa_2^{-1}(BA)\right) + O\left(\kappa_2^{-2}(BA)\right)\right) \doteq \kappa_2(BA)|\ln\epsilon| \end{aligned}$$

Unter der Annahme, dass der Aufwand für eine Iteration bei dünnbesetzter Matrix Cn beträgt ist der Gesamtaufwand zur Lösung eines linearen Gleichungssystems somit

$$A_{IT}(n) = C\kappa_2(BA)|\ln\epsilon|n.$$

Für die besten Verfahren gilt tatsächlich in praktisch relevanten Fällen $\kappa(BA) = const$ und somit kann so ein System mit Aufwand linear in n gelöst werden!

Vorlesung 15

Lösung nichtlinearer algebraischer Gleichungen

15.1 Motivation

Wir sind nun interessiert an folgendem Problem: Gegeben eine Funktion

$$f : D \to \mathbb{R}^n, \qquad D \subset \mathbb{R}^n,$$

finde ein $x \in D$ so dass

$$f(x) = 0.$$

So ein x nennen wir Lösung der Gleichung „$f(x) = 0$“. Wir können die Gleichung auch in Komponenten aufschreiben:

$$f_i(x_1, \ldots, x_n) = 0, \qquad i = 1, \ldots, n.$$

Für Probleme dieser Art gibt es keine allgemeinen, direkten Verfahren welche nach einer a-priori bekannten Zahl von Rechenoperationen eine Lösung liefern, sondern es kommen iterative Lösungsverfahren zum Einsatz. Wir können also an die Methoden aus den letzten beiden Vorlesungen anknüpfen.

Beispiel 15.1. a) In einer Variable betrachte die Gleichung

$$f(x) = x \exp(x) - 3 = 0$$

mit einer Lösung $x \in [1, 1.2]$, siehe Abbildung 15.1.

b) In zwei Variablen betrachte

$$f_1(x_1, x_2) = x_1^2 + x_2^2 - 4 = 0,$$
$$f_2(x_1, x_2) = \frac{x_1^2}{9} + x_2^2 - 1 = 0.$$

Die erste Gleichung für sich betrachtet beschreibt die Punkte auf einem Kreis mit Radius 2 um den Ursprung. Die zweite Gleichung beschreibt Punkte auf einer Ellipse mit großer Halbachse 3 und kleiner Halbachse 1 um den Ursprung, siehe Abbildung 15.1. Demnach gibt es vier Punkte an denen beide Gleichungen gleichzeitig erfüllt sind.

c) Für $A \in \mathbb{R}^{n \times n}$ regulär und $b \in \mathbb{R}^n$ betrachte $f(x) = Ax - b$ mit der einzigen Lösung $A^{-1}b$. Das allgemeine Problem $f(x) = 0$ umfasst also auch lineare Gleichungssysteme.

15.2 Grundlagen der Analysis

Für die Betrachtungen zur Konvergenz von Iterationsverfahren benötigen wir einige weitere Grundlagen aus der Analysis. Wir knüpfen an die Grundlagen der Linearen Algebra in Vorlesung 5 an.

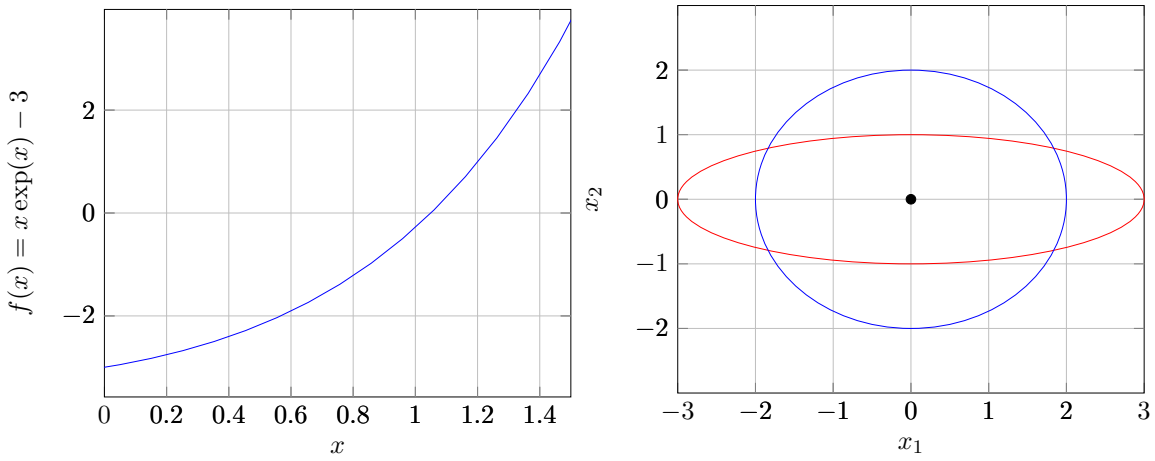


Abbildung 15.1: Links: Die Funktion $f(x) = x\exp(x) - 3$ in einer Variablen. Rechts: Die Menge der Punkte mit $f_1(x_1, x_2) = x_1^2 + x_2^2 = 4$ in blau sowie $f_2(x_1, x_2) = x_1^2/9 + x_2^2 = 1$ in rot.

Lineare Gleichungssysteme haben entweder keine, genau eine oder unendlich viele Lösungen. Die obigen Beispiele legen nahe, dass nichtlineare, algebraische Gleichungen auch genau m, $m \in \mathbb{N}$ Lösungen haben können. Um bestimmte Lösungen zu finden schränkt man den Bereich der Suche auf $D \subset \mathbb{R}^n$ ein. In diesem Zusammenhang spielt der Begriff der offenen bzw. abgeschlossenen Menge eine Rolle.

Definition 15.2 (Offene Kugel)**.** Die Menge der Punkte

$$B_\epsilon(z) = \{x \in \mathbb{R}^n \,:\, \|x - z\| < \epsilon\}$$

heißt offene Kugel mit Radius ϵ um z bezüglich der Norm $\|.\|$. □

Im folgenden sei $\|.\|$ eine beliebige, aber fest gewählte Norm solange nichts anderes angegeben ist. Mit Hilfe der offenen Kugel können wir den Begriff der offenen Menge definieren.

Definition 15.3 (Offene Menge)**.** Eine Menge $D \subset \mathbb{R}^n$ heißt offen, wenn es zu jedem $x \in D$ ein $\epsilon = \epsilon(x) > 0$ gibt so dass gilt $B_\epsilon(x) \subset D$. □

Definition 15.4 (Umgebung)**.** Zu einem Punkt $x \in \mathbb{R}^n$ heißt jede offene Menge $N \subset \mathbb{R}^n$ mit $x \in N$ eine Umgebung von x. □

Definition 15.5 (Abgeschlossene Menge)**.** Eine Menge $D \subset \mathbb{R}^n$ heißt abgeschlossen wenn $\mathbb{R}^n \setminus D$ (das Komplement bezüglich $\mathbb{R}^n$) offen ist. □

Als Beispiel für eine abgeschlossene Menge nennen wir die abgeschlossene Kugel

$$\bar{B}_\epsilon(z) = \{x \in \mathbb{R}^n \,:\, \|x - z\| \leq \epsilon\},$$

denn die Menge $\mathbb{R}^n \setminus \bar{B}_\epsilon(z) = \{x \in \mathbb{R}^n \,:\, \|x - z\| > \epsilon\}$ ist offen. $\mathbb{R}^n$ selbst ist sowohl abgeschlossen als auch offen.

Damit wenden wir uns nun wieder dem Konvergenzbegriff für Folgen zu. Folgen $\left\{x^{(k)}\right\}_{k=1}^{\infty}$ mit Elementen $x^{(k)} \in \mathbb{R}^n$ notieren wir auch kurz als $\left\{x^k\right\} \subset \mathbb{R}^n$.

Definition 15.6 (Konvergenz)**.** Eine Folge $\{x^k\} \subset \mathbb{R}^n$ konvergiert gegen $x \in \mathbb{R}^n$, falls gilt

$$\forall \epsilon > 0 \; \exists N(\epsilon) \in \mathbb{N} \; \forall k \geq N(\epsilon) \; : \; \|x - x^k\| < \epsilon.$$

□

Wir hatten auch schon den Begriff der Cauchy-Folge eingeführt:

Definition 15.7 (Cauchy-Folge)**.** Eine Folge $\{x^k\} \subset \mathbb{R}^n$ heißt Cauchy-Folge falls gilt

$$\forall \epsilon > 0 \; \exists N \in \mathbb{N} \; \forall k, l \geq N \; : \; \|x^k - x^l\| < \epsilon.$$

□

Die reellen Zahlen sind nun gerade so definiert, dass in ihnen alle Cauchy-Folgen konvergieren. Die Konvergenz bezüglich irgendeiner Norm impliziert die Konvergenz in der Maximumnorm und diese impliziert wiederum die komponentenweise Konvergenz von $x_i^k \to x_i$. Dies wiederum liefert die Konvergenz aller Cauchy-Folgen im $\mathbb{R}^n$. Man sagt $\mathbb{R}^n$ ist ein vollständiger Vektorraum.

Folgendes Lemma betrachtet die Konvergenz von Cauchy-Folgen in Teilmengen $D \subset \mathbb{R}^n$.

Lemma 15.8. Es sei $D \subset \mathbb{R}^n$ abgeschlossen und nicht leer. Für eine Cauchy-Folge $\{x^k\} \subset D$ (alle Elemente sind in D) gilt dann $x = \lim_{k\to\infty} x^k \in D$.

Beweis. Nach Voraussetzung ist $\{x^k\}$ eine Cauchy-Folge. Somit exisitiert ein Grenzwert $x = \lim_{k\to\infty} x^k \in \mathbb{R}^n$. Zu zeigen ist, dass $x \in D$. Angenommen es sei $x \notin D$, also $x \in \mathbb{R}^n \setminus D$. D ist nach Voraussetzung abgeschlossen, also ist $\mathbb{R}^n \setminus D$ offen. Somit gibt es ein $\epsilon > 0$ und $B_\epsilon(x) \subset \mathbb{R}^n \setminus D$. Da aber $x^k \in D$ für alle $k \in \mathbb{N}$ muss somit $\|x - x^k\| \geq \epsilon$ gelten. Dies ist ein Widerspruch zur Konvergenz der x^k gegen x. Also muss $x \in D$ sein. □

Für offene Mengen D kann sehr wohl $\{x^k\} \subset D$, aber $x = \lim_{k\to\infty} \notin D$ gelten. x liegt dann „am Rand“ von D. Durch systematische Hinzunahme von Grenzwerten kann man eine offene Menge abschließen.

Definition 15.9. Sei $D \subset \mathbb{R}^n$ eine offene Menge. Dann heißt $\bar{D}$ Abschluss von D, wenn für jedes $x \in \bar{D}$ gilt: $x \in D$ oder x ist Grenzwert einer Cauchy-Folge $\{x^k\} \subset D$. □

Verwandt dazu ist der Prozess der Vervollständigung. Dort fügt man zu einer gegebenen Menge die Grenzwerte aller Cauchy-Folgen hinzu. So erhält man die reellen Zahlen als Vervollständigung der rationalen Zahlen.

Alle unten zu besprechenden Verfahren fordern gewisse Eigenschaften der Funktion f. Wir beginnen mit der Stetigkeit.

Definition 15.10. Es sei $D \subset \mathbb{R}^n$. Eine Funktion $f : D \to \mathbb{R}^n$ heißt stetig im Punkt $x \in D$ falls gilt:

$$\forall \epsilon > 0 \; \exists \delta = \delta(x, \epsilon) \; \forall y \in B_\delta(x) \cap D : \quad \|f(x) - f(y)\| < \epsilon.$$

Ist f stetig in allen Punkten $x \in D$ so heißt f stetig auf D. □

Für stetige Funktionen auf D gilt folgende wichtige Aussage.

Lemma 15.11. Es sei $D \subset \mathbb{R}^n$ und $f : D \to \mathbb{R}^n$ stetig auf D. Die Folge $\{x^k\} \subset D$ konvergiere gegen $x \in D$. Dann gilt

$$\lim_{k\to\infty} f\left(x^k\right) = f(x).$$

Beweis. Zu zeigen ist die Konvergenz der Folge $\{f^k = f(x^k)\} \subset \mathbb{R}^n$. Zu jedem ϵ brauchen wir also ein $N(\epsilon)$ so dass $\|f(x^k) - f(x)\| < \epsilon$ für alle $k \geq N(\epsilon)$. Da f stetig, gilt $\|f(y) - f(x)\| < \epsilon$ für alle $\|y - x\| < \delta(x, \epsilon)$ und dies erreichen wir für alle $y = x^k$ mit $k \geq N_2(\delta(x, \epsilon))$ da $x^k \to x$. □

15.3 Fixpunktiteration

Statt der Nullstellensuche

$$f(x) = 0$$

betrachten wir im folgenden Gleichungen in Fixpunktform

$$g(x) = x$$

mit der Idee, dass man die Funktion g so wählt, dass gilt

$$f(x) = 0 \quad \Leftrightarrow \quad g(x) = x.$$

Bei dieser Umformung ist zu beachten dass f im allgemeinen nur auf einer Teilmenge $D \to \mathbb{R}^n$ definiert ist. Dann fordern wir, dass $g : D \to D$.

Ein allgemeiner Ansatz zur Umformung ist die Relaxation

$$g(x) = x - \sigma f(x)$$

mit einem $0 \neq \sigma \in \mathbb{R}$. Offensichtlich folgt aus $f(x) = 0$, dass $g(x) = x$ und umgekehrt folgt aus $g(x) = x - \sigma f(x) = x$, dass $f(x) = 0$.

Es sind aber auch andere Umformungen möglich, die aber von der genauen Gestalt von f abhängen. Zum Beispiel gilt $f(x) = x\exp(x) - 3 = 0 \Leftrightarrow x = 3/\exp(x)$, oder $f(x) = x\exp(x) - 3 = 0 \Leftrightarrow x = \ln(3/x) = \ln 3 - \ln x$.

Definition 15.12 (Fixpunkt)**.** Es sei $g : D \to D$ eine Abbildung. $\xi \in D$ heißt Fixpunkt der Abbildung g wenn gilt $g(\xi) = \xi$. Fixpunkte sind also Lösungen der Gleichung in Fixpunktform. □

Der Vorteil der Fixpunktform ist, dass wir nun in generischer Weise eine Iterationsvorschrift angeben können.

Definition 15.13 (Fixpunktiteration)**.** Die Iteration

$$x^{k+1} = g\left(x^k\right) \tag{15.1}$$

bezeichnen wir als Fixpunktiteration. Ausgehend von einem Startwert x^0 wird so eine Folge $\{x^k\}_{k=0}^{\infty}$ generiert. □

Nun stellt sich natürlich die Frage unter welchen Umständen die Fixpunktiteration gegen einen Fixpunkt der Abbildung g konvergiert.

Definition 15.14 (Lipschitzstetigkeit, Kontraktion)**.** Es sei $D \subset \mathbb{R}^n$ nicht leer und abgeschlossen. Die Abbildung $g : D \to D$ erfüllt eine (globale) Lipschitzbedingung, falls gilt

$$\|g(x) - g(y)\| \leq L\|x - y\| \qquad \forall x, y \in D.$$

Gilt zusätzlich $L < 1$, nennt man g eine Kontraktion und L den Kontraktionsfaktor. □

Bisher haben wir die Norm $\|.\|$ nicht weiter spezifiziert. Die Lipschitzkonstante hängt in der Regel jedoch von der Norm ab und es kann daher vorkommen, dass g bezüglich einer Norm eine Kontraktion ist, bezüglich einer anderen Norm jedoch nicht. Dies ist völlig vergleichbar zu der Situation bei Iterationsverfahren bei linearen Gleichungssystemen (siehe nächstes Beispiel). Auch dort ist bei Normabschätzungen der Iterationsmatrix eine geschickte Wahl der Norm zu treffen.

Beispiel 15.15. Wir betrachten den Fall linearer Gleichungssysteme, d.h. $f(x) = Ax - b$. In der alternativen Form (13.5) schreiben wir das lineare Iterationsverfahren als $x^{k+1} = Sx^k + Bb$ und identifizieren $g(x) = Sx + Bb$ mit der Iterationsmatrix $S = I - BA$. Wir rechnen nach, dass die Lösung $\xi = A^{-1}b$ des linearen Gleichungssystemes ein Fixpunkt ist:

$$g(\xi) = (I - BA)A^{-1}b + Bb = A^{-1}b - Bb + Bb = A^{-1}b = \xi.$$

Außerdem gilt

$$\|g(x) - g(y)\| = \|Sx + Bb - (Sy + Bb)\| = \|S(x - y)\| \leq \|S\|\|x - y\|$$

und somit $L = \|S\|$. Für $\|S\| < 1$ konvergiert das Iterationsverfahren.

Beobachtung 15.16. Es sei $D \subset \bar{B}_M(0) \subset \mathbb{R}^n$ nicht leer und abgeschlossen. Die Abbildung $g : D \to D$ sei eine Kontraktion bezüglich der Norm $\|.\|$, d.h. es gilt

$$\|g(x) - g(y)\| \leq q\|x - y\| \qquad \forall x, y \in D$$

mit $q < 1$. Dann gilt $g(D) \subset D$.

Beweis. Angenommen es sei $g(D) = D$, also g injektiv. D.h. zu jedem $x \in D$ gibt es ein $x' \in D$ mit $g(x') = x$. Nun seien x, y so gewählt, dass

$$\|x - y\| = \sup_{v,w \in D} \|v - w\| = \operatorname{diam}(D)$$

und $g(x') = x$ und $g(y') = y$. Dann gilt

$$\frac{\|x - y\|}{\|x' - y'\|} = \frac{\|g(x') - g(y')\|}{\|x' - y'\|} = \frac{\operatorname{diam}(D)}{\|x' - y'\|} \geq \frac{\operatorname{diam}(D)}{\operatorname{diam}(D)} = 1.$$

Dies ist ein Widerspruch zur Voraussetzung, somit muss $g(D) \subset D$ sein. □

Satz 15.17 (Banachscher Fixpunktsatz)**.** Es sei $D \subset \mathbb{R}^n$ nicht leer und abgeschlossen. Die Abbildung $g : D \to D$ sei eine Kontraktion bezüglich der Norm $\|.\|$, d.h. es gilt

$$\|g(x) - g(y)\| \leq q\|x - y\| \qquad \forall x, y \in D$$

mit $q < 1$. Dann besitzt g einen eindeutigen Fixpunkt ξ und die durch die Fixpunktiteration (15.1) definierte Folge $\{x^k\}$ konvergiert für jeden Startwert $x^0 \in D$ gegen diesen Fixpunkt. Es gelten die Fehlerabschätzungen

$$\|x^k - \xi\| \leq \frac{q}{1-q}\|x^k - x^{k-1}\| \leq \frac{q^k}{1-q}\|x^1 - x^0\|.$$

Die erste Abschätzung bezeichnet man als *a-posteriori* Abschätzung und die zweite als *a-priori* Abschätzung.

Beweis. Zunächst bemerken wir, dass die Folge der $\{x^k\} \subset D$ wegen $g : D \to D$ in D wohldefiniert ist. Wir zeigen nun, dass $\{x^k\}$ eine Cauchy-Folge ist. Zunächst beobachten wir

$$\|x^{k+1} - x^k\| = \|g(x^k) - g(x^{k-1}\| \leq q\|x^k - x^{k-1}\| \leq q^k\|x^1 - x^0\|.$$

Es sei $t \in \mathbb{N} \cup \{0\}$, $\mathbb{N} \ni m \geq 1$. Dann gilt

$$\begin{aligned}
\|x^{k+m} - x^k\| &= \|x^{k+m} - x^{k+m-1} + x^{k+m-1} - x^{k+m-2} + x^{k+m-2} - \ldots + x^{k+1} - x^k\| \\
&\leq \|x^{k+m} - x^{k+m-1}\| + \|x^{k+m-1} - x^{k+m-2}\| + \ldots + \|x^{k+1} - x^k\| \\
&\leq q^{k+m-1}\|x^1 - x^0\| + q^{k+m-2}\|x^1 - x^0\| + \ldots + q^k\|x^1 - x^0\| \\
&= \left(q^{k+m-1} + q^{k+m-2} + q^k\right)\|x^1 - x^0\| \\
&= q^k\left(\sum_{i=0}^{m-1} q^i\right)\|x^1 - x^0\| = q^k\frac{1-q^m}{1-q}\|x^1 - x^0\| \\
&\leq q^k\frac{1}{1-q}\|x^1 - x^0\|.
\end{aligned}$$

Zu jedem $\epsilon > 0$ können wir also $N = N(\epsilon)$ finden so dass $\|x^{k+m} - x^k\| < \epsilon$ für alle $k \geq N$ und $m \in \mathbb{N}$. Damit ist $\{x^k\}$ eine Cauchy-Folge. Nach Lemma 15.8 konvergiert die Cauchy-Folge $\{x^k\}$ in einer abgeschlossenen Menge D gegen $\xi = \lim_{k\to\infty} x^k \in D$. Wir zeigen, dass ξ Fixpunkt von g ist. Für jedes $t \in \mathbb{N}$ gilt

$$\|\xi - g(\xi)\| = \|\xi - x^k + x^k - g(\xi)\| \leq \|\xi - x^k\| + \|g(x^{k-1}) - g(\xi)\| \leq \|\xi - x^k\| + q\|x^{k-1} - \xi\|$$

und somit $\|\xi - g(\xi)\| = 0$ da $x^k \to \xi$.

Die Eindeutigkeit des Fixpunktes ergibt sich mittels Beweis durch Widerspruch. Angenommen es gibt zwei verschieden Fixpunkte $\xi_1 \neq \xi_2$. Dann gilt

$$\|\xi_1 - \xi_2\| = \|g(\xi_1) - g(\xi_2)\| \leq q\|\xi_1 - \xi_2\| \quad \Leftrightarrow \quad (1-q)\|\xi_1 - \xi_2\| = 0.$$

Nun ist $1 - q > 0$ nach Voraussetzung und es muss damit $\|\xi_1 - \xi_2\| = 0$ gelten um diese Gleichung zu erfüllen. Da $\|.\|$ eine Norm ist muss $\xi_1 - \xi_2 = 0$, also $\xi_1 = \xi_2$ gelten. Dies ist ein Widerspruch zur Annahme $\xi_1 \neq \xi_2$. Somit ist der Fixpunkt eindeutig.

Nun noch zur Fehlerabschätzung. Wir verfahren ähnlich wie oben mit der Teleskopsumme:

$$\begin{aligned}
\|x^{k+m} - x^k\| &= \|x^{k+m} - x^{k+m-1} + x^{k+m-1} - x^{k+m-2} + x^{k+m-2} - \ldots + x^{k+1} - x^k\| \\
&\leq \|x^{k+m} - x^{k+m-1}\| + \|x^{k+m-1} - x^{k+m-2}\| + \ldots + \|x^{k+1} - x^k\| \\
&\leq q^m\|x^k - x^{k-1}\| + q^{m-1}\|x^k - x^{k-1}\| + \ldots + q\|x^k - x^{k-1}\| \\
&= \left(q^m + q^{m-1} + q\right)\|x^k - x^{k-1}\| \\
&\leq \frac{1}{1-q}\|x^k - x^{k-1}\|.
\end{aligned}$$

Diese Abschätzung gilt für alle $m \in \mathbb{N}$ also auch für den Grenzwert (Fixpunkt):

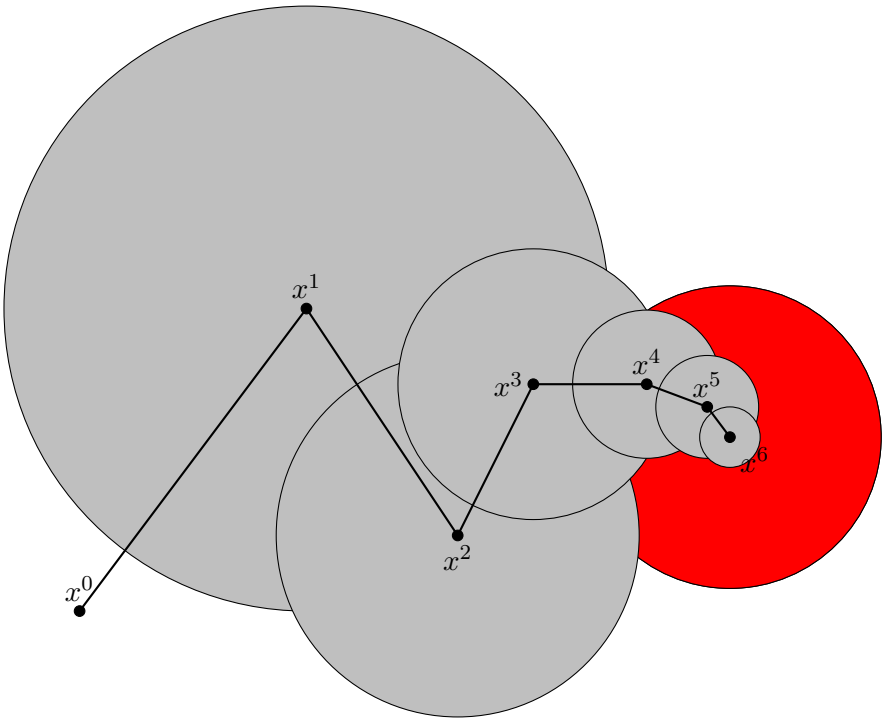
$$\|\xi - x^k\| \leq \frac{1}{1-q}\|x^k - x^{k-1}\|$$

womit die a-posteriori Abschätzung bewiesen ist. Wegen $\|x^{k+1} - x^k\| \leq q^k\|x^1 - x^0\|$ gilt auch die a-priori Abschätzung. □

Die Konvergenz einer Fixpunktiteration im $\mathbb{R}^2$ mit $q = 0.8$ ist in der Zeichnung unten illustriert. Aus der Kontraktionseigenschaft erhalten wir

$$\|x^{k+1} - x^k\| \leq q\|x^k - x^{k-1}\|.$$

Der graue Kreis um ein x^k gibt jeweils an in welchem Bereich sich das nächste x^{k+1} befinden kann. Der rote Kreis gibt mittels der a-posteriori Abschätzung an in welchem Bereich um x^6 sich der Fixpunkt ξ befindet. Der rote Kreis hat einen Radius von 2.0. Der Radius für die a-priori Abschätzung würde 6.55 betragen und wäre deutlich größer.



Vorlesung 16

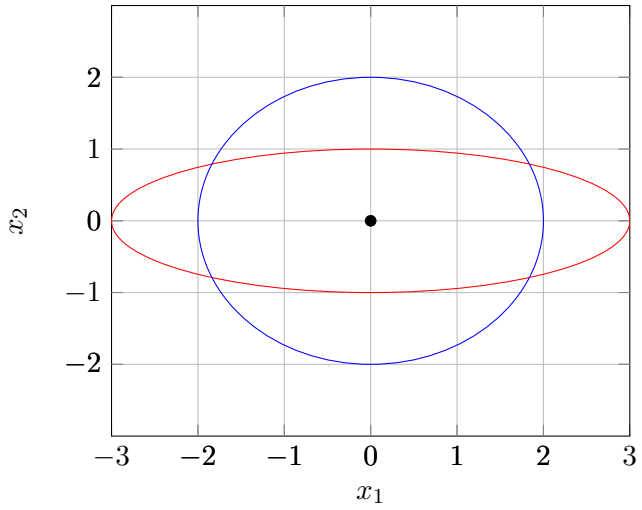
Newton-Verfahren

Wir starten mit einem Beispiel zur Konvergenz der Fixpunktiteration.

Beispiel 16.1. Wir untersuchen die Nullstellen der Funktion aus Beispiel 15.1 b):

$$f_1(x_1, x_2) = x_1^2 + x_2^2 - 4 = 0,$$
$$f_2(x_1, x_2) = \frac{x_1^2}{9} + x_2^2 - 1 = 0.$$

Die Lösungen der separaten Gleichungen sind hier noch einmal gezeichnet:



Demnach gibt es vier Lösungen in der Nähe der Punkte $\{-2, 2\} \times \{-1, 1\}$.

Wir untersuchen nun die Fixpunktiteration

$$x^{k+1} = g(x^k) = x - \sigma f(x)$$

für verschiedene Startwerte und Relaxationsparameter σ.

Für den Startwert $x^0 = (2, 1)^T$ und $\sigma = 0.1$ ergibt sich folgendes Konvergenzverhalten bei einer geforderten Reduktion von 10^{-10}:

```
Banach  norm=1.0943e+00
      1 norm=6.1020e-01 red=5.5760e-01
      2 norm=3.5535e-01 red=5.8235e-01
      3 norm=2.2022e-01 red=6.1972e-01
      4 norm=1.5079e-01 red=6.8473e-01
      5 norm=1.1587e-01 red=7.6841e-01
      6 norm=9.6811e-02 red=8.3552e-01
      7 norm=8.4127e-02 red=8.6899e-01
      8 norm=7.3974e-02 red=8.7930e-01
      9 norm=6.5071e-02 red=8.7965e-01
     10 norm=5.7066e-02 red=8.7698e-01
...
    150 norm=1.6684e-10 red=8.6917e-01
```

```
    151 norm=1.4501e-10 red=8.6917e-01
    152 norm=1.2604e-10 red=8.6917e-01
    153 norm=1.0955e-10 red=8.6917e-01
    154 norm=9.5215e-11 red=8.6917e-01
Banach converged in 154 steps reduction=8.7009e-11
Ergebnis: 1.837117307046946e+00 7.905694151026036e-01
```

Der Reduktionsfaktor beträgt etwa $q = 0.8692$.

Für verschiedene Relaxationsparameter ergibt sich folgendes Bild:

σ	Reduktion	Schritte
0.6	1.0	∞
0.51	1.0	∞
0.505	0.9933	3307
0.5	0.9735	849
0.4	0.5789	43
0.3	0.6075	43
0.2	0.7383	71
0.1	0.8692	154
0.05	0.9346	318
0.01	0.9869	1631

Für $\sigma > 0.505$ liegt keine Konvergenz vor, im Bereich $0.3 \ldots 0.4$ ist die Konvergenzrate am besten, bei kleineren Werten wird sie wieder schlechter.

Nun untersuchen wir die Konvergenz in der Nähe der anderen Nullstellen und es ergibt sich folgendes Bild:

- Die Konvergenz in der Nähe der Nullstelle $(-1.83711730704694, -7.90569415102603)$ im dritten Quadranten ist vollkommen analog, nur dass man den Relaxationsparameter negativ wählen muss.
- In der Nähe der Nullstellen im zweiten und vierten Quadranten erhält man keine Konvergenz, egal wie man den Relaxationsparameter wählt und wie nahe an der Nullstelle man startet.

In dieser Vorlesung wollen wir vor allem zwei Fragestellungen nachgehen:

1. Wie können wir nachweisen dass die Fixpunktiteration konvergiert, also g eine Kontraktion ist.
2. Wie können wir die Konvergenzeigenschaften des Relaxationsverfahrens verbessern.

16.1 Nachweis der Kontraktionseigenschaft

In der Regel ist es schwierig die Kontraktionseigenschaft nachzuweisen. Ein möglicher Zugang basiert auf der Einbeziehung von Ableitungsinformation. Sei $g : [a, b] \to \mathbb{R}$ stetig differenzierbar, dann folgt mit dem Hauptsatz der Differential- und Integralrechnung sowie der Transformation $s(t) = y + s(x - y)$ für zwei beliebige $x, y \in [a, b]$:

$$g(x) - g(y) = \int_y^x g'(s)ds = \int_0^1 g'(y + s(x-y))(x-y)ds = \left(\int_0^1 g'(y + s(x-y))ds \right) (x-y).$$

Durch Betragsbildung erhalten wir

$$|g(x) - g(y)| \leq \left| \int_0^1 g'(y + s(x-y))ds \right| |x - y)|$$

und somit die Lipschitzkonstante als

$$L = \sup_{x,y \in [a,b]} \left| \int_0^1 g'(y + s(x-y))ds \right|.$$

Es zeigt sich zudem, dass die Einbeziehung von Ableitungsinformation auch die zweite Frage oben nach der Verbesserung der Konvergenzeigenschaften beantwortet. Dazu müssen wir aber zunächst auf Ableitungen vektorwertiger Funktionen eingehen.

Definition 16.2 (Jacobimatrix)**.** Sei $g : \mathbb{R}^n \to \mathbb{R}^n$ eine Funktion mit den Komponenten $g(x) = (g_1(x), \ldots, g_n(x))^T$. g sei stetig und stetig partiell differenzierbar mindestens in einer Umgebung $N(\xi)$ des Punktes ξ. Dann heißt die $n \times n$ Matrix mit den Einträgen

$$(J_g(\xi))_{i,j} = \frac{\partial g_i}{\partial x_j}(\xi)$$

Jacobimatrix von g am Punkt ξ. □

Aus dem (Spezialfall des) Satz von Taylor für Funktionen in mehreren Variablen, Satz 3.4, folgt dann bei komponentenweiser Anwendung:

$$g(\xi + \eta) = g(\xi) + J_g(\xi)\eta + O(\|\eta\|^2).$$

Zuerst benötigen wir noch zwei vorbereitende Resultate Lemma.

Lemma 16.3 (Mittelwertsatz im $\mathbb{R}^n$)**.** Es sei $g : D \to \mathbb{R}^n$ eine stetige Funktion auf der nichtleeren, abgeschlossenen Menge D mit stetigen partiellen Ableitungen auf D. Für zwei Punkte $x, y \in D$ gilt dann die Darstellung

$$g(x) - g(y) = \int_0^1 J_g(sx + (1-s)y)ds \; (x - y).$$

Beweis. Für gegebene $x, y \in D$ definiere die Funktion $\varphi : [0, 1] \to \mathbb{R}^n$ als

$$\varphi(s) = g(sx + (1-s)y).$$

Es gilt $\varphi(0) = g(y)$, $\varphi(1) = g(x)$. φ ist stetig und auch stetig differenzierbar auf $[0, 1]$. Mit dem Hauptsatz der Differential- und Integralrechnung erhalten wir

$$g(x) - g(y) = \varphi(1) - \varphi(0) = \int_0^1 \varphi'(s)ds.$$

Die Ableitung $\varphi'(s)$ können wir mit der Kettenregel ausrechnen:

$$\varphi_i'(s) = \sum_{j=1}^n \frac{\partial g_i}{\partial x_j}(sx + (1-s)y)(x_j - y_j) = \left(J_g(sx + (1-s)y)(x - y)\right)_i.$$

So erhalten wir

$$g(x) - g(y) = \int_0^1 \varphi'(s)ds = \int_0^1 J_g(sx + (1-s)y)ds(x-y)$$

was behauptet wurde. □

Lemma 16.4. Es sei $f : [a, b] \to \mathbb{R}^n$ (komponentenweise) integrierbar und $\|.\|$ eine beliebige Vektornorm. Dann gilt

$$\left\| \int_a^b f(s)ds \right\| \leq \int_a^b \|f(s)\|ds.$$

Beweis.

$$\left\| \int_a^b f(s)ds \right\| = \lim_{n\to\infty} \left\| \sum_{i=1}^n f(s_i)\Delta s_i \right\| \leq \lim_{n\to\infty} \sum_{i=1}^n \|f(s_i)\|\Delta s_i = \int_a^b \|f(s)\|ds.$$

□

Der folgende Satz verbindet die Kontraktionseigenschaft mit der Jacobimatrix.

Satz 16.5. Es sei $g : D \to \mathbb{R}^n$, $D \subset \mathbb{R}^n$ nicht leer und abgeschlossen sowie g auf D stetig. g habe den Fixpunkt $\xi \in D$ und die Jacobimatrix J_g existiert in einer Umgebung $N(\xi) \subset D$ des Fixpunktes ξ und es gilt

$$\|J_g(\xi)\| < 1.$$

Dann gibt es ein $\epsilon > 0$ so dass $g(\bar{B}_\epsilon(\xi)) \subset \bar{B}_\epsilon(\xi)$ und die Fixpunktiteration (15.1) konvergiert für alle Startwerte in $\bar{B}_\epsilon(\xi)$.

Beweis. Wir setzen zur Abkürzung $K = \|J_g(\xi)\|$ und wählen eine Zahl zwischen K und 1 auf folgende Weise:

$$2K < K + 1 < 2 \quad \Rightarrow \quad K < \frac{1}{2}(K+1) < 1.$$

Die partiellen Ableitungen von g sind nach Voraussetzung stetig in einer Umgebung $N(\xi)$ des Fixpunktes. Daher gibt es ein $\epsilon > 0$ und $\bar{B}_\epsilon(\xi) \subset N(\xi) \subset D$ so dass gilt

$$\|J_g(z)\| \leq \frac{1}{2}(K+1) < 1 \qquad \forall z \in \bar{B}_\epsilon(\xi).$$

Mit den beiden Resultaten von oben schätzen wir nun ab:

$$\begin{aligned}
\|g(x) - g(y)\| &= \left\| \int_0^1 J_g(sx + (1-s)y)ds(x-y) \right\| \\
&\leq \int_0^1 \|J_g(sx + (1-s)y)\|ds\|(x-y)\| \\
&\leq \|x-y\| \int_0^1 \frac{1}{2}(K+1)ds = \frac{1}{2}(K+1)\|x-y\|
\end{aligned}$$

Damit haben wir die Lipschitzbedingung mit $L = \frac{1}{2}(K+1) < 1$ für alle Punkte $x, y \in \bar{B}_\epsilon(\xi)$ nachgewiesen. Mit dem Fixpunktsatz 15.17 konvergiert daher die Fixpunktiteration (15.1) für alle Startwerte in $\bar{B}_\epsilon(\xi)$. □

16.2 Relaxation

Eine Möglichkeit die Nullstellensuche $f(x) = 0$ in Fixpunktform zu transformieren ist die Relaxation $g(x) = x - \sigma f(x)$, welche wir diesem Abschnitt näher untersuchen wollen. Mittels der Taylorentwicklung gilt

$$\begin{aligned} g(x) - g(y) &= x - \sigma f(x) - (y - \sigma f(y)) = x - y - \sigma(f(x) - f(y)) \\ &= x - y - \sigma\left(f(\xi + x - \xi) - f(\xi + y - \xi)\right) \\ &= x - y - \sigma\big(f(\xi) + J_f(\xi)(x - \xi) + O(\|x - \xi\|^2) \\ &\qquad - (f(\xi) + J_f(\xi)(y - \xi) + O(\|y - \xi\|^2))\big) \\ &= x - y - \sigma\left(J_f(\xi)(x - y) + O(\|x - \xi\|^2 + \|y - \xi\|^2)\right) \\ &= (I - \sigma J_f(\xi))(x - y) + O(\|x - \xi\|^2 + \|y - \xi\|^2). \end{aligned}$$

Wir beobachten:

1) Sind x, y nahe am Fixpunkt so ist der $O()$ Term sehr klein und die Konvergenz wird von dem ersten Term bestimmt.

2) Der Vergleich mit linearen Iterationsverfahren legt nahe dass es sich bei der Relaxation um eine Art Richardson-Iteration handelt. Wäre $J_j(\xi)$ etwa symmetrisch und positiv definit, dann könnte man für genügend kleines $\sigma > 0$ Konvergenz in einer Umgebung des Fixpunktes erwarten. Diese Eigenschaft ist aber bei nichtlinearen Problemen eher unwahrscheinlich.

3) Ein anderer Zugang wäre etwa über Diagonaldominanz gegeben, da man dann keine Symmetrie benötigt.

Der Vergleich mit linearen Iterationsverfahren legt auch nahe, dass man statt $B = \sigma I$ einen besseren Vorkonditionierer einsetzen könnte. Ideal wäre z.B. $B = J_f^{-1}(\xi)$. Dann wäre sogar $g(x) - g(y) = O(\|x - \xi\|^2 + \|y - \xi\|^2)$. Allerdings kennen wir ξ nicht und können daher $J_f(\xi)$ auch nicht auswerten. Eine berechenbare alternative wäre aber $J_f(x^k)$! Damit beschäftigen wir uns im nächsten Kapitel.

16.3 Newton-Verfahren

Die Betrachtungen im letzten Kapitel legen die folgende Iteration nahe:

$$x^{k+1} = g(x^k) = x^k - J_f^{-1}(x^k) f(x^k). \tag{16.1}$$

Diese Iteration nennt man *Newton*-Verfahren. Dieses konvergiert sehr schnell, so dass wir gar eine neue Definition für Konvergenzordnung benötigen.

Definition 16.6 (Konvergenzordnung)**.** Sei $\{x^k\} \subset \mathbb{R}^n$ ein konvergente Folge mit dem Grenzwert $\xi \in \mathbb{R}^n$.

i) $\{x^k\}$ konvergiert linear mit Konvergenzfaktor ϱ gegen ξ wenn ein $N \in \mathbb{N}$ existiert so dass

$$\|x^{k+1} - \xi\| \leq \varrho \|x^k - \xi\| \qquad \forall k \geq N.$$

ii) $\{x^k\}$ konvergiert mit Konvergenzordnung $p > 1$ gegen ξ falls es eine Konstante $C > 0$ und $N \in \mathbb{N}$ gibt so dass

$$\|x^{k+1} - \xi\| \leq C \|x^k - \xi\|^p \qquad \forall k \geq N.$$

Für $p = 2$ spricht man von quadratischer Konvergenz. □

Für das Newton-Verfahren gilt:

$$\begin{aligned} g(x) - g(y) &= x - J_f^{-1}(x)f(x) - (y - J_f^{-1}(y)f(y)) \\ &= x - y - (J_f^{-1}(x)f(x) - J_f^{-1}(y)f(y) \end{aligned}$$

In der Nähe des Fixpunktes ξ sollten sich die Inversen der Jacobimatrix in den Punkten x, y ähnlich wie die Inversen am Fixpunkt verhalten. Dies zeigt das folgende Lemma.

Lemma 16.7. Die Funktion $f : \mathbb{R}^n \to \mathbb{R}^n$ sei in einer Umgebung $U(\xi)$ stetig differenzierbar und es gelte für alle $x, y \in \bar{B}_R(\xi)$ die Lipschitzbedingung für die Jacobimatrix:

$$\|J_f(x) - J_f(y)\| \leq L\|x - y\|.$$

Weiter sei $J_f(\xi)$ regulär. Dann gilt für jedes $0 < q < 1$ mit $r = \min\left(\frac{q}{L\|J_f^{-1}(\xi)\|}, R\right)$ dass $J_f(z)$ ebenfalls regulär für alle $z \in \bar{B}_r(\xi)$ und es gilt die Abschätzung

$$\|J_f(z)^{-1}\| \leq \frac{1}{1-q}\|J_f^{-1}(\xi)\|.$$

Beweis. Wir setzen an:

$$J_f(z) = J_f(\xi)+J_f(z)-J_f(\xi) = J_f(\xi)(I+J_f^{-1}(\xi)(J_f(z)-J_f(\xi))) = J_f(\xi)(I+B).$$

Nun ist $J_f(z)$ regulär falls $J_f(\xi)$ regulär und $I+B$ regulär, mit $B = J_f^{-1}(\xi)(J_f(z)-J_f(\xi))$. $J_f(\xi)$ ist regulär nach Voraussetzung. Mit der Bedingung

$$\|z - \xi\| \leq r = \min\left(\frac{q}{L\|J_f^{-1}(\xi)\|}, R\right)$$

gilt

$$\begin{aligned} \|B\| &= \|J_f^{-1}(\xi)(J_f(z) - J_f(\xi))\| \leq \|J_f^{-1}(\xi)\|\|J_f(z) - J_f(\xi)\| \\ &\leq \|J_f^{-1}(\xi)\|L\|z - \xi\| \leq q \end{aligned}$$

und daraus folgt mit Hilfssatz 6.4 :

$$\|(I + B)^{-1}\| \leq \frac{1}{1 - \|B\|} \leq \frac{1}{1-q}.$$

Damit gilt dann schließlich

$$\|J_f^{-1}(z)\| = \|(I + B)^{-1}J_f^{-1}(\xi)\| \leq \|(I + B)^{-1}\|\|J_f^{-1}(\xi)\| \leq \frac{1}{1-q}\|J_f^{-1}(\xi)\|.$$

□

Damit sind wir nun in der Lage die Konvergenz des Newton-Verfahrens zu zeigen.

Satz 16.8. Die Funktion $f : \mathbb{R}^n \to \mathbb{R}^n$ habe eine Nullstelle $f(\xi) = 0$ und sei in einer Umgebung $U(\xi)$ der Nullstelle stetig differenzierbar. Weiter gelte für alle $x, y \in \bar{B}_R(\xi) \subset U(\xi)$ die Lipschitzbedingung für die Jacobimatrix:

$$\|J_f(x) - J_f(y)\| \leq L\|x - y\|.$$

Die Jacobimatrix $J_f(\xi)$ an der Nullstelle sei regulär. Dann konvergiert das Newton-Verfahren (16.1) für alle Startwerte $x^0 \in \bar{B}_r(\xi)$ mit $r = \min\left(\frac{1}{2L\|J_f^{-1}(\xi)\|}, R\right)$ quadratisch gegen die Nullstelle.

Beweis. Aus der Definition des Newton-Verfahrens erhalten wir die Identitäten:

$$\begin{aligned} J_f(x^k)x^{k+1} &= J_f(x^k)x^k - f(x^k), \\ J_f(x^k)\xi &= J_f(x^k)\xi - f(\xi). \end{aligned}$$

Subtraktion liefert

$$J_f(x^k)(x^{k+1} - \xi) = J_f(x^k)(x^k - \xi) - (f(x^k) - f(\xi)).$$

Mit dem Mittelwertsatz 16.3 gilt

$$f(x^k) - f(\xi) = \int_0^1 J_f(sx^k + (1-s)\xi)ds\ (x^k - \xi)$$

und damit

$$\begin{aligned} J_f(x^k)(x^{k+1} - \xi) &= J_f(x^k)(x^k - \xi) - \int_0^1 J_f(sx^k + (1-s)\xi)ds\ (x^k - \xi) \\ &= \int_0^1 J_f(x^k)ds\ (x^k - \xi) - \int_0^1 J_f(sx^k + (1-s)\xi)ds\ (x^k - \xi) \\ &= \int_0^1 J_f(x^k) - J_f(sx^k + (1-s)\xi)ds\ (x^k - \xi). \end{aligned}$$

Nun schätzen wir ab (nutze $q = \frac{1}{2}$ in Lemma 16.7) :

$$\begin{aligned} \|x^{k+1} - \xi\| &= \left\| J_f^{-1}(x^k) \int_0^1 J_f(x^k) - J_f(sx^k + (1-s)\xi)ds\ (x^k - \xi) \right\| \\ &\leq \left\| J_f^{-1}(x^k) \right\| \int_0^1 \left\| J_f(x^k) - J_f(sx^k + (1-s)\xi) \right\| \left\| x^k - \xi \right\| ds \\ &\leq \left\| J_f^{-1}(x^k) \right\| L \int_0^1 \left\| (1-s)(x^k - \xi) \right\| ds\ \left\| x^k - \xi \right\| \\ &\leq 2 \left\| J_f^{-1}(\xi) \right\| L \int_0^1 1 - sds\ \left\| x^k - \xi \right\|^2 = L \left\| J_f^{-1}(\xi) \right\| \left\| x^k - \xi \right\|^2. \end{aligned} \tag{16.2}$$

Per Induktion zeigen wir nun dass $x^k \in \bar{B}_r(\xi)$. Für $k = 0$ gilt nach Voraussetzung $\|x^0 - \xi\| \leq r \leq \frac{1}{2L\|J_f^{-1}(\xi)\|}$. Unter der Annahme $\|x^k - \xi\| \leq r$ gilt

$$\|x^{k+1} - \xi\| \leq L \left\|J_f^{-1}(\xi)\right\| \left\|x^k - \xi\right\| \left\|x^k - \xi\right\| \leq \frac{1}{2} \left\|x^k - \xi\right\|$$

und somit auch $\|x^{k+1} - \xi\| \leq r$. Wegen

$$\|x^k - \xi\| \leq \left(\frac{1}{2}\right)^k \|x^0 - \xi\|$$

konvergiert die Newton-Iteration gegen die Nullstelle. Wegen (16.2) ist die Konvergenz quadratisch. □

Bemerkung 16.9. 1) Der Beweis für das Newton-Verfahren setzt die Existenz der Nullstelle voraus. Wollte man die Existenz dieser Nullstelle auch beweisen müsste man zuerst noch die Kontraktionseigenschaft zeigen.

2) Es gibt Beweise mit leicht schwächeren Voraussetzungen. Dies bezieht sich auf die hier geforderte Lipschitzbedingung für die Jacobimatrix. Diese impliziert die Existenz (beschränkter) zweiter Ableitungen von f.

3) Das Newton-Verfahren konvergiert lokal, d.h. der Startwert muss geenügend nahe an der Nullstelle sein. Dies kann in der Praxis ein Problem sein. Mittels sogenannter Globalisierungsstrategien wird versucht den Konvergenzradius zu vergrößern. Eine dieser Strategien ist die Liniensuche. Die Konvergenz ist dann in der Regel zu Beginn höchstens linear.

4) In der Praxis führt man das Newton-Verfahren folgendermaßen durch: 1) Berechne $d = f(x^k)$ sowie $J = J_f(x^k)$. 2) Löse das lineare Gleichungssystem $Jv = d$, schließlich setze 3) $x^{k+1} = x^k - v$. Der wesentliche Aufwand bei der Durchführung des Newton-Verfahrens besteht in der Aufstellung der Jacobimatrix J sowie der Lösung des linearen Gleichungssystemes $Jv = d$. In Anwendungen, z.B. bei partiellen Differentialgleichungen, kann die Jacobimatrix wieder dünn besetzt sein und es werden iterative Verfahren eingesetzt. Um den Aufwand zu reduzieren betrachtet man inexakte Newton-Verfahren bei denen das System $Jv = d$ nur näherungsweise gelöst wird, z.B. in dem iterativ mit geringer Genauigkeit gelöst oder die Jacobimatrix nur näherungsweise berechnet wird.

Beispiel 16.10. Wir kehren zurück zum Beispiel 16.1 vom Anfang der Vorlesung:

$$f_1(x_1, x_2) = x_1^2 + x_2^2 - 4 = 0,$$
$$f_2(x_1, x_2) = \frac{x_1^2}{9} + x_2^2 - 1 = 0.$$

Dort haben wir die Konvergenz der Relaxation untersucht. Nun lösen wir das gleiche Problem mit dem Newton-Verfahren Für den Startwert $x^0 = (2, 1)^T$ und $\sigma = 0.1$ ergibt sich folgendes Konvergenzverhalten bei einer geforderten Reduktion von 10^{-10}:

```
Newton    norm=1.0943e+00
  step  1 norm=7.0588e-02 red=6.4504e-02
  step  2 norm=6.9692e-04 red=9.8730e-03
```

```
  step  3 norm=1.2395e-07 red=1.7785e-04
  step  4 norm=4.7207e-15 red=3.8086e-08
Newton converged in 4 steps reduction=4.3139e-15
Ergebnis: 1.837117307087384e+00 7.905694150420968e-01
```

Das Verfahren konvergiert beeindruckend schnell. Statt 154 Iterationen werden nur 4 benötigt. Man sieht auch gut die quadratische Konvergenz: der Reduktionsfaktor quadriert sich jeweils.

Für den Startwert $x^0 = (200, 1)^T$ in größerer Entfernung ergibt sich:

```
Newton    norm=4.0243e+04
  step  1 norm=1.0060e+04 red=2.4998e-01
  step  2 norm=2.5141e+03 red=2.4991e-01
  step  3 norm=6.2768e+02 red=2.4966e-01
  step  4 norm=1.5608e+02 red=2.4865e-01
  step  5 norm=3.8188e+01 red=2.4468e-01
  step  6 norm=8.7674e+00 red=2.2958e-01
  step  7 norm=1.5799e+00 red=1.8020e-01
  step  8 norm=1.2542e-01 red=7.9382e-02
  step  9 norm=1.1168e-03 red=8.9044e-03
  step 10 norm=9.1788e-08 red=8.2191e-05
Newton converged in 10 steps reduction=2.2808e-12
Ergebnis: 1.837117331916086e+00 7.905694150420948e-01
```

Die quadratische Konvergenz zeigt sich erst wenn man genügend nahe an der Nullstelle ist. Vorher ist die Konvergenz linear (was wir nicht bewiesen haben).

Schließlich untersuchen wir die Konvergenz gegen die Nullstellen im zweiten und vierten Quadranten. Für den Startwert $x^0 = (-2, 1)^T$ erhalten wir

```
Newton    norm=1.0943e+00
  step  1 norm=7.0588e-02 red=6.4504e-02
  step  2 norm=6.9692e-04 red=9.8730e-03
  step  3 norm=1.2395e-07 red=1.7785e-04
  step  4 norm=4.7207e-15 red=3.8086e-08
Newton converged in 4 steps reduction=4.3139e-15
Ergebnis: -1.837117307087384e+00 7.905694150420968e-01
```

also Konvergenz gegen die Lösung im zweiten Quadranten. Das Verfahren konvergiert, im Unterschied zur Relaxation, sehr gut für alle vier Nullstellen.

Vorlesung 17

Polynominterpolation

17.1 Einführung

Im folgenden beschäftigen wir uns mit der Darstellung von Funktionen $f : \mathbb{R} \to \mathbb{R}$ welche durch „Daten“ gegeben sind. Dieses Thema haben wir in Abschnitt 11.1 schon kurz behandelt ohne dabei auf verschiedene Methoden einzugehen.

Wozu kann man das brauchen? Hier einige Beispiele:

1) Rekonstruktion eines funktionalen Zusammenhangs aus gemessenen Funktionswerten. Dies erlaubt dann auch die Bestimmung von Funktionswerten an anderen Stellen als den Gemessenen.

2) „Teuer“ auszuwertende Funktionen mit weniger Aufwand auswerten. Früher waren Werte der Logarithmusfunktion tabelliert, da sie sehr aufwändig zu berechnen waren. Werte an nicht tabellierten Stellen wurden „interpoliert“.

3) Darstellung von „Fonts“, also Punktmengen im $\mathbb{R}^2$.

4) Darstellung von dreidimensionalen Körpern im Rechner, nennt man auch „computer aided design“ (CAD). Dies ist der Ausgangspunkt für numerische Simulationen oder Computergrafik.

5) Lösung von Differential- und Integralgleichungen. Dort ist das Ziel der Berechnung eine Funktion, die geeignet im Rechner dargestellt werden muss.

6) Datenkompression: Sind viele Werte der Funktion „ähnlich“ können diese Werte gegebenenfalls durch einen einfachen Verlauf (konstant, linear) repräsentiert werden.

Alle diese Beispiele legen natürlich nahe, dass der dabei entstehende Fehler in der Darstellung kontrolliert werden muss.

Im folgenden werden wir uns weitgehend mit Funktionen $f : \mathbb{R} \to \mathbb{R}$ in einer Variablen beschäftigen. In der Praxis kann man die meisten Methoden aber auf den mehrdimensionalen Fall erweitern. Dies ist technisch oft sehr viel aufwändiger und soll daher nicht behandelt werden.

Angenommen es soll eine m-mal stetig differenzierbare Funktion $f \in C^m([a, b])$ im Rechner dargestellt werden, dann ist dies in der Regel nicht exakt möglich, da $C^m([a, b])$ als Vektorraum unendliche Dimension hat. Im Rechner können wir nur endlich viel Information darstellen und betrachten daher Näherungsfunktionen mit endlich vielen Parametern, hier einige Beispiele:

a) $p(x) = a_0 + a_1 x + \ldots + a_n x^n$ (Polynome).

b) $r(x) = \frac{a_0 + a_1 x + \ldots + a_n x^n}{b_0 + b_1 x + \ldots + b_m x^m}$ (Rationale Funktionen).

c) $t(x) = \frac{1}{2} a_0 + \sum_{k=1}^{n} (a_k \cos(kx) + b_k \sin(kx))$ (trigonometrische Polynome, endliche Fourier-Reihe).

d) $e(x) = \sum_{k=1}^{n} a_k e^{b_k x}$ (Exponentialsumme)

Häufig bedient man sich einer linearen Struktur, d.h. die Näherungsfunktion ist eine Linearkombination von endlich vielen Grundfunktionen, wie etwa in den Fällen a) und c). Dann bilden die Funktionen einen endlichdimensionalen Vektorraum.

Die Grundaufgabe der Approximation lautet dann: Gegeben eine Menge von Funktionen P, sowie eine zu approximierende Funktion f (die in der Regel nicht in P ist). Dann finde eine Funktion $g \in P$ so dass die Differenz $f - g$ in geeigneter Weise minimiert wird. Beispiele wären:

a) $\int (f(x) - g(x))^2 dx \to \min$ (L_2-Fehler).

b) $\max_{a \le x \le b} |f(x) - g(x)| \to \min$ (Maximumnorm Fehler)

c) $\max_{i=0,\dots,n} |f(x_i) - g(x_i)| \to \min$ für bestimmte Werte $a \le x_i \le b$.

Man spricht von Interpolation falls g durch die Bedingungen

$$g(x_i) = y_i = f(x_i), \qquad i = 0, \dots, n$$

festgelegt wird. Dies ist ein Spezialfall der allgemeinen Approximationsaufgabe.

Wir beschäftigen uns nun mit der Interpolation mit Polynomen! Die Menge der Polynome vom Grad kleiner gleich $n \in \mathbb{N} \cup \{0\}$ ist definiert als

$$P_n = \left\{ p(x) = \sum_{i=0}^{n} a_i x^i \ : \ a_i \in \mathbb{R} \right\}.$$

Die Funktionen x^i für $i \in \mathbb{N} \cup \{0\}$ heißen Monome. P_n ist ein $n+1$-dimensionaler Vektorraum und die Monome $1, \dots, x^n$ bilden eine Basis von P_n wie wir später zeigen werden.

Die Interpolationsaufgabe mit Polynomen lautet dann wie folgt:

Definition 17.1. Zu $n+1$ gegebenen und paarweise verschiedenen Stützstellen $x_0, \dots, x_n$ und beliebigen Werten $y_0, \dots, y_n \in \mathbb{R}$ finde

$$p \in P_n : \qquad p(x_i) = y_i, \qquad i = 0, \dots, n.$$

□

Ein naive Lösung der Interpolationsaufgabe gelingt mittels dem Ansatz

$$p(x_i) = \sum_{j=0}^{n} a_j x_i^j = y_i, \qquad i = 0, \dots, n$$

welcher dann auf ein lineares Gleichungssystem der Größe $n + 1$ für die Koeffizienten führt:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & & & \dots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \quad \Leftrightarrow \quad V[x_0, \dots, x_n]a = y. \tag{17.1}$$

Die dabei auftretende Matrix $V[x_0, \dots, x_n]$ nennt man Vandermonde-Matrix. Man erkennt sofort, dass die Stützstellen x_i paarweise verschieden sein müssen, sonst ist die Matrix nicht regulär. Dass $V[x_0, \dots, x_n]$ in diesem Fall dann

regulär ist, wird sich aus der Eindeutigkeit der Polynominterpolation ergeben welche wir unten zeigen. Die Vandermondematrix ist schlecht konditioniert schon für relativ kleine n, da die Kondition exponentiell mit n ansteigt.

Die schlechte Kondition von $V[x_0, \ldots, x_n]$ sowie der hohe Rechenaufwand $O(n^3)$ zur Bestimmung der Koeffizienten $a_0, \ldots, a_n$ führen zur Frage nach geschickteren Lösungen.

17.2 Lagrange-Interpolation

Der Schlüssel zur geschickteren Lösung der Interpolationsaufgabe liegt in der Wahl einer anderen Basis.

Definition 17.2 (Lagrange-Polynome)**.** Zu den $n+1$ paarweise verschiedenen Stützstellen x_i, $i = 0, \ldots, n$ definiere die Polynome

$$L_i^{(n)}(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}, \qquad i = 0, \ldots, n.$$

$L_i^{(n)}(x)$ heißt i-tes Lagrange-Polynom vom Grad n. □

Die Lagrange-Polynome haben folgende Eigenschaften:

Lemma 17.3. a) $L_i^{(n)}(x)$ hat Grad n.

b) Es gilt $L_i^{(n)}(x_k) = \delta_{i,k},\ i, k = 0, \ldots, n.$

c) Die $L_i^{(n)}(x)$, $i = 0, \ldots, n$ bilden eine Basis von P_n.

Beweis. a) $L_i^{(n)}(x)$ besteht aus einem Produkt von n Faktoren der Form $(x - x_j)$.

b) Für $k \neq i$ kommt im Produkt $\prod_{j=0, j\neq i}^{n} \frac{x-x_j}{x_i-x_j}$ der Index $k = j$ vor, also der Faktor $x_k - x_k = 0$ im Zähler und damit ist das Produkt Null. Ist $k = i$ dann kommt der Index $j = k$ im Produkt nicht vor und es gilt $\prod_{j=0, j\neq i}^{n} \frac{x_k-x_j}{x_k-x_j} = 1$.

c) Wegen der letzten Eigenschaft sind die $L_i^{(n)}(x)$ linear unabhängig. Dies sieht man wie folgt: $L_k^{(n)}(x)$ hat an der Stelle x_k den Wert 1. Für alle anderen $L_i^{(n)}(x)$, $i \neq k$ gilt $L_k^{(n)}(x_k) = 0$, also kann keine Linearkombination der $L_i^{(n)}(x)$, $i \neq k$, das $L_k^{(n)}(x)$ ergeben. Damit haben wir $n + 1$ linear unabhängige Polynome vom Grad maximal n. Da die Dimension von P_n $n + 1$ ist bilden die $L_k^{(n)}(x)$ eine Basis.

□

Abbildung 17.1 zeigt die Lagrange-Polynome vom Grad 6 zu äquidistanten Stützstellen.

Wegen der Eigenschaft b) ist die Interpolationsaufgabe in der Basis der Lagrange-Polynome besonders einfach zu lösen.

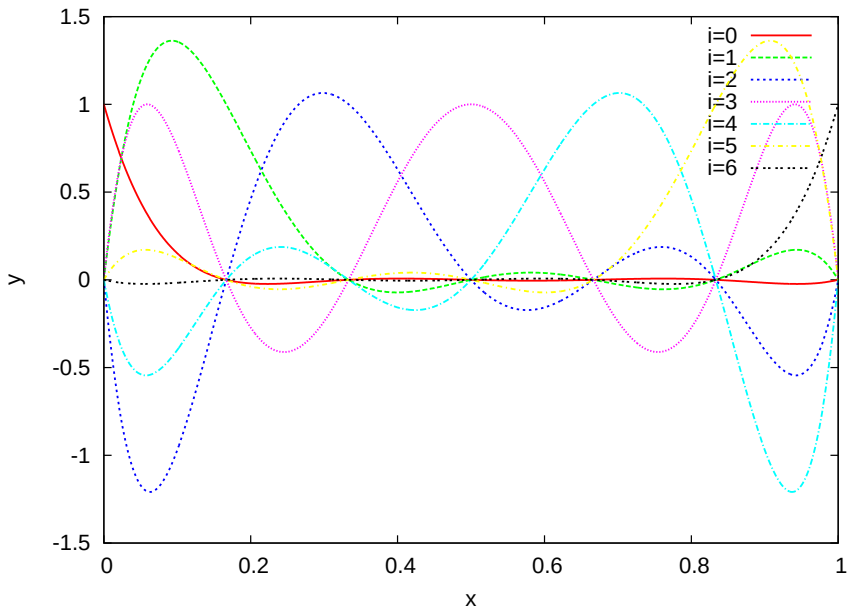


Abbildung 17.1: Lagrange-Polynome vom Grad 6 mit äquidistanten Stützstellen im Intervall $[0, 1]$.

Satz 17.4. Es seien $n+1$ paarweise verschiedene Stützstellen x_i sowie Werte $y_i \in \mathbb{R}$ für $i = 0, \ldots, n$ gegeben. Dann interpoliert

$$p(x) = \sum_{i=0}^{n} y_i L_i^{(n)}(x)$$

die gegebenen Werte an den Stützstellen.

Beweis. Es gilt $p(x_k) = \sum_{i=0}^{n} y_i L_i^{(n)}(x_k) = \sum_{i=0}^{n} y_i \delta_{i,k} = y_k$. □

Formal erhalten wir also den Koeffizientenvektor a durch „Lösen“ dees linearen Gleichungssystemes $Ia = y$.

Schließlich beweisen wir noch den Satz.

Satz 17.5. Zu gegebenen $n+1$ paarweise verschiedenen Stützstellen x_i sowie Werten $y_i \in \mathbb{R}$ für $i = 0, \ldots, n$ gibt es genau ein Polynom $p \in P_n$ welches die Werte y_i an den Stützstellen x_i interpoliert.

Beweis. $p(x) = \sum_{i=0}^{n} y_i L_i^{(n)}(x)$ interpoliert die gegebenen Stützstellen. Die $L_i^{(n)}$ bilden eine Basis von P_n, also ist die Darstellung eindeutig. □

Beispiel 17.6. Zur Illustration der Polynominterpolation mit Lagrange-Polynomen betrachten wir die Tabelle

x_i	y_i
0	1.0000
2	0.4546
7	0.0938
10	−0.0544

Abbildung 17.2 zeigt das Interpolationspolynom in rot sowie die vier Lagrange-Polynome zu den Datenpunkten. □

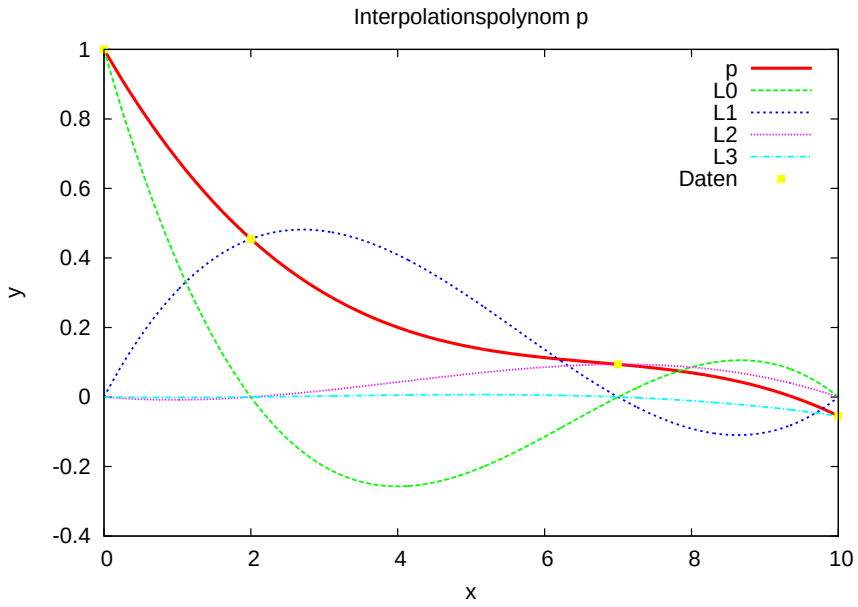


Abbildung 17.2: Interpolation von vier Datenpunkten.

17.3 Newton-Interpolation

Die Langrange-Polynome erlauben eine (sehr) einfache Lösung der Polynominterpolationsaufgabe. Allerdings haben sie folgenden Nachteil: Jedes Basispolynom hängt von allen Stützstellen ab. Dies ist von Nachteil, wenn man die Zahl der Stützstellen dynamisch erweitern möchte. Also man beginnt etwa mit drei Stützstellen, erstellt ein Interpolationspolynom und möchte dann eine weitere Stützstelle hinzufügen. In diesem Fall müssen alle Lagrange-Polynome neu berechnet werden. Nicht so in der Basis der Newton-Polynome. Diese sind wie folgt definiert.

Definition 17.7 (Newton-Polynome)**.** Zu den $n+1$ paarweise verschiedenen Stützstellen x_i, $i = 0, \ldots, n$ definiere die Polynome

$$N_0(x) = 1, \qquad N_i(x) = \prod_{j=0}^{i-1}(x - x_j), \quad i = 1, \ldots, n.$$

$N_i(x)$ heißt i-tes Newton-Polynom. □

Die Newton-Polynome haben folgende Eigenschaften:

Lemma 17.8. a) $N_i(x)$ hat Grad i.

b) Es gilt $N_i(x_k) = 0$ für alle $k < i$.

c) Die $N_i(x)$, $i = 0, \ldots, n$ bilden eine Basis von P_n.

Beweis. a) N_i ist ein Produkt von i Faktoren der Form $(x - x_j)$.

b) Für $k < i$ kommt der Faktor $(x - x_k)$ in N_i vor, also gilt $N_i(x_k) = \ldots (x_k - x_k) \ldots = 0$.

c) Per Induktion. Betrachte zunächst $i = 0$. Da $N_0(x_0) = 1$ und $N_j(x_0) = 0$ für alle $j > 0$ ist N_0 linear unabhängig von allen N_j mit $j = 1, \ldots, n$. Nun seien $N_0, \ldots, N_{i-1}$ linear unabhängig von allen N_j, $j \geq i$. Nun gilt $N_i(x_i) \neq 0$, aber $N_j(x_i) = 0$ für alle $j > i$, also ist auch N_i linear unabhängig von allen N_j, $j > i$. Somit sind die N_i linear unabhängig. Da die Dimension von P_n $n + 1$ ist bilden diese eine Basis.

□

Die Interpolationsaufgabe löst man nun wie folgt unter Ausnutzung der Eigenschaft b):

$$p(x_k) = \sum_{i=0}^{n} a_i N_i(x_k) = \sum_{i=0}^{k} a_i N_i(x_k) = y_i, \qquad k = 0, \ldots, n.$$

Also

$$a_0 = y_0, \qquad a_k = \left(y_k - \sum_{i=0}^{k-1} a_i N_i(x_k) \right) / N_k(x_k), \quad k > 0.$$

Wir erhalten also den Koeffizientenvektor a durch Lösen des linearen Gleichungssystemes $La = y$ mit unterer Dreiecksmatrix $l_{k,i} = N_i(x_k)$.

Fügt man nun eine Stützstelle x_{n+1} dazu, ändern sich die Koeffizienten $a_0, \ldots, a_n$ nicht, sondern man kann direkt $a_{n+1} = (y_{n+1} - \sum_{i=0}^{n} a_i N_i(x_{n+1})) / N_{n+1}(x_{n+1})$ bestimmen.

Wir geben noch zwei weitere Sätze zur Newton-Interpolation ohne Beweis an. Diese haben früher eine gewisse Rolle gespielt und man sollte sie zumindest kennen.

Satz 17.9 (Dividierte Differenzen)**.** Es seien $n + 1$ paarweise verschiedene Stützstellen x_i sowie Werte $y_i \in \mathbb{R}$ für $i = 0, \ldots, n$ gegeben. Dann lässt sich das Interpolationspolynom in der Newton-Basis berechnen als

$$p(x) = \sum_{i=0}^{n} y[x_0, \ldots, x_i] N_i(x).$$

Dabei sind die $y[x_0, \ldots, x_i]$ die sogenannten „dividierten Differenzen“, die Rekursiv definiert sind durch:

$$\begin{aligned} i = 0, \ldots, n: \qquad & y[x_i] = y_i, \\ k = 1, \ldots, n - i: \qquad & y[x_i, \ldots x_{i+k}] = \frac{y[x_{i+1}, \ldots x_{i+k}] - y[x_i, \ldots x_{i+k-1}]}{x_{i+k} - x_i}. \end{aligned}$$

Beweis. Siehe [Rannacher, 2017, Satz 2.2]. □

Die Berechnung der Koeffizienten der Newton-Darstellung mittels dividierter Diffierenzen ist numerisch stabiler als die naive Variante bei gleichem Rechenaufwand. Es ergibt sich folgendes Rechenschema:

$$\begin{array}{llll} y_0 = y[x_0] & y[x_0, x_1] & y[x_0, x_1, x_2] & y[x_0, x_1, x_2, x_3] \\ y_1 = y[x_1] & y[x_1, x_2] & y[x_1, x_2, x_3] & \\ y_2 = y[x_2] & y[x_2, x_3] & & \\ y_3 = y[x_3] & & & \end{array}$$

Dabei wird (ausser in der ersten Spalte) jeder Eintrage aus den beiden Wert links und diagonal links unten berechnet. Fügt man einen neuen Punkt hinzu, so sind Werte entlang der Diagonalen entsprchend einzufügen.

Schließlich erlaubt das sogenannte Neville-Schema eine effiziente Auswertung an einzelnen Punkten, d.h. es kombiniert das Aufstellen des Interpolationspolynoms (in der Newton-Basis) und die Auswertung an einem Punkt.

Satz 17.10 (Neville Schema)**.** Es seien $n+1$ paarweise verschiedene Stützstellen x_i sowie Werte $y_i \in \mathbb{R}$ für $i = 0, \ldots, n$ gegeben. $p(x)$ sei das Interpolationspolynom zu diesen Punkten. Für einen beliebigen Punkt x gilt dann

$$p(x) = p_{0,n}(x)$$

mit der Rekursion

$$\begin{aligned} i = 0, \ldots, n: &\quad p_{i,i}(x) = y_i, \\ k = 1, \ldots, n-i: &\quad p_{i,i+k}(x) = p_{i,i+k-1}(x) + (x - x_i)\frac{p_{i+1,i+k}(x) - p_{i,i+k-1}(x)}{x_{i+k} - x_i} \end{aligned}$$

Beweis. Siehe [Rannacher, 2017, Definition 2.4 und Satz 2.2]. □

Auch hier ergibt sich ein ähnliches Berechnungsschema wie oben:

$$\begin{array}{llllll} x_0 & y_0 = p_{0,0}(x) & p_{0,1}(x) & p_{0,2}(x) & p_{0,3}(x) \\ x_1 & y_1 = p_{1,1}(x) & p_{1,2}(x) & p_{1,3}(x) & \\ x_2 & y_2 = p_{2,2}(x) & p_{2,3}(x) & & \\ x_3 & y_3 = p_{3,3}(x) & & & \end{array}$$

Vorlesung 18

Interpolationsfehler und numerische Differentiation

Es sei $y_i = f(x_i)$, $i = 0, \ldots, n$ die Auswertung einer Funktion f an $n+1$ paarweise verschiedenen Stützstellen und $p(x)$ sei das Interpolationspolynom vom Grad n welches die Punkte (x_i, y_i), $i = 0, \ldots, n$ interpoliert. Für die Differenz $f(x) - p(x)$ gilt dann

$$f(x) - p(x) = \begin{cases} 0 & x = x_i \\ ? & x \neq x_i \end{cases}$$

Nun stellen wir uns die Frage wie groß die Differenz zwischen den Stützstellen maximal werden kann.

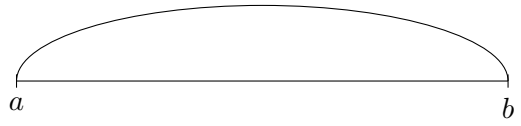
18.1 Interpolationsfehler

Wir benutzen den folgenden fundamentalen Satz der Analysis

Satz 18.1 (Satz von Rolle)**.** Sei $u(x)$ auf $[a, b]$ stetig und auf (a, b) differenzierbar sowie $u(a) = u(b) = 0$. Dann existiert mindestens ein $x \in (a, b)$ mit $u'(x) = 0$.

Beweis. Spezialfall des Mittelwertsatzes der Differentialrechnung. □

Folgende Zeichnung illustriert den Satz von Rolle:



Satz 18.2. Die Funktion f sei $n+1$-mal stetig differenzierbar auf dem Intervall $[a, b]$. Weiter sei $a \leq x_0 < x_1 < \cdots < x_n \leq b$ eine Menge paarweise verschiedener Stützstellen in $[a, b]$ (lässt auch Extrapolation zu!). Dann gibt es zu jedem $x \in [a, b]$ ein $\xi_x \in \overline{(x_0, \ldots, x_n, x)}$ (=kleinstes abgeschlossenes Intervall, welches alle Punkte enthält), so dass

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{j=0}^{n} (x - x_j).$$

Beweis. Im Spezialfall $x \in \{x_0, \ldots, x_n\}$ liefert $\prod_{j=0}^{n} (x - x_j) = 0$, somit ist ξ_x beliebig wählbar. Sei also nun $x \in [a, b] \setminus \{x_0, \ldots, x_n\}$. Zu diesem x definiere die Funktion

$$F_x(t) = f(t) - p(t) - \underbrace{\frac{f(x) - p(x)}{l(x)}}_{\substack{\text{eine Zahl} \\ \text{abhängig von } x}} l(t) \text{ mit } l(t) = \prod_{j=0}^{n} (t - x_j)$$

$F_x(t)$ hat die $n+2$ Nullstellen $\{\underbrace{x_0, \ldots, x_n}_{n+1}, x\}$, denn

$$i = 0, \ldots, n: \qquad F_x(x_i) = \underbrace{f(x_i) - p(x_i)}_{=0} - \frac{f(x) - p(x)}{l(x)} \underbrace{l(x_i)}_{=0} = 0$$

$$x: \qquad F_x(x) = f(x) - p(x) - \frac{f(x) - p(x)}{\cancel{l(x)}} \cancel{l(x)} = 0$$

Auf die Funktion $F_x(t)$ wenden wir nun den Satz von Rolle an. Da $F_x(t)$ wie oben gezeigt, mindestens $n+2$ Nullstellen hat, hat

- $F_x^{(1)}(t)$ (die erste Ableitung) mindestens $n+1$ Nullstellen und
- $F_x^{(2)}(t)$ (die zweite Ableitung) mindestens n Nullstelle und schliesslich
- $F_x^{(n+1)}(t)$ ($n+1$-te Ableitung) noch mindestens 1 Nullstelle.

Zusätzlich gilt: Diese Nullstellen sind in $\overline{(x_0, \ldots, x_n, x)}$, da eine Nullstelle der Ableitung immer zwischen zwei Nullstellen der Funktion liegt.

Nun nenne die Nullstelle von $F_x^{(n+1)}(t)$ $\xi_x \in \overline{(x_0, \ldots, x_n, x)}$ (bzw. falls es mehrere gibt, nehme eine davon). Dann gilt für die Nullstelle:

$$F_x^{(n+1)}(\xi_x) = f^{(n+1)}(\xi_x) - \underbrace{p^{(n+1)}(\xi_x)}_{\substack{=0, \text{ da } p \\ \text{vom Grad } n}} - \frac{f(x) - p(x)}{l(x)} \underbrace{l^{(n+1)}(\xi_x)}_{\substack{l(t)=t^{n+1}+\ldots \\ n+1 \text{ mal differenzieren} \\ l^{(n+1)}(t)=(n+1)!}}$$

$$= f^{(n+1)}(\xi_x) - \frac{f(x) - p(x)}{l(x)}(n+1)! \stackrel{!}{=} 0$$

Auflösen nach $f(x) - p(x)$ liefert die Behauptung. □

Diskussion des Interpolationsfehlers. Durch bilden des Betrages erhalten wir:

$$|f(x) - p(x)| \leq \frac{|f^{(n+1)}(\xi_x)|}{(n+1)!} \prod_{j=0}^{n} |x - x_j| \quad \text{für ein } \xi_x.$$

Wir betrachten nun das Intervall $[a, b]$ mit äquidistanter Wahl der Stützstellen $x_i = a + hi$, $h = (b-a)/n$, also $|x_i - x_{i+1}| = h$, sowie $x \in [a, b]$ und $x \neq x_i$. Für ein solches x sei nun l die Anzahl der Stützpunkte links von x auf dem Zahlenstrahl und r die Anzahl der Stützpunkte rechts von x und natürlich $l + r = n + 1$. Mit dieser Notation erhalten wir

$$\prod_{j=0}^{n} |x - x_j| \leq \underbrace{lh \cdots 2h \cdot 1h}_{\text{links von } x} \cdot \underbrace{1h \cdot 2h \cdots rh}_{\text{rechts von } x} = l!\ r!\ h^{n+1}$$

also insgesamt

$$|f(x) - p(x)| = |f^{(n+1)}(\xi_x)| \underbrace{\frac{l!\ r!}{(n+1)!}}_{\substack{\leq 1 \text{ da} \\ l+r=n+1}} h^{n+1}.$$

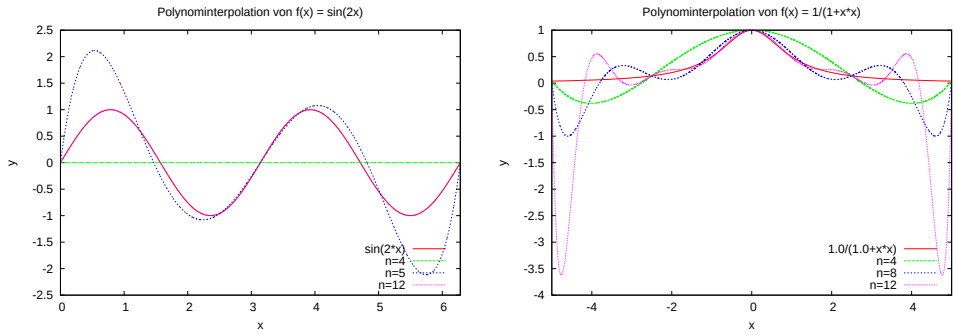


Abbildung 18.1: Interpolation der Funktionen $\sin(2x)$ (oben) und $\frac{1}{1+x^2}$ (unten) mit äquidistanten Stützstellen und verschiedenen Polynomgraden.

Für beschränkte Ableitung $|f^{(n+1)}|$ können wir den Interpolationsfehler durch Hinzunahme von Stützpunkten, also $n \to \infty$, beliebig klein machen. Insbesondere ist die Konvergenz exponentiell in n.

Allerdings sind die höheren Ableitungen auch einfacherer Funktionen für $n \to \infty$ oft nicht beschränkt, sondern wachsen sehr schnell. So gilt etwa für Runges Beispiel [Rannacher, 2017, S. 32]:

$$f(x) = \frac{1}{1+x^2} \qquad |f^{(n+1)}(x)| \approx 2^n\, n!\, O\left(\frac{1}{|x|^{n+2}}\right)$$

(siehe Auch Abbildung 18.1 unten).

Bemerkung 18.3. 1) Dies ist kein Widerspruch zum Approximationssatz von Weierstrass. Dieser besagt: Jede Funktion $f \in C[a,b]$ kann beliebig gut gleichmäßig auf $[a,b]$ mit Polynomen approximiert werden, d.h. dort ist von Approximation und nicht von Interpolation die Rede.

2) Allgemein gilt bei Verwendung von Polynomen hohen Grades, dass entsprechend hohe Differenzierbarkeit von f vorliegen muss. □

Beispiel 18.4. Wir betrachten ein Beispiel. Bei der Funktion $\sin(2x)$ auf dem Intervall $[0, 2\pi]$ wächst die n+1-te Ableitung genügend langsam und die Funktion lässt sich auch auf äquidistanten Stützstellen sehr gut mit Polynomen interpolieren, siehe Abbildung 18.1 (links). Ganz anders ist dies bei der rationalen Funktion $\frac{1}{1+x^2}$, in Abbildung 18.1, rechts. Hier oszilliert das Interpolationspolynom insbesondere zum Rand des Intervalles hin sehr stark und insbesondere wird der maximale Fehler mit steigendem Polynomgrad immer größer. Dieses Verhalten kann durch die Wahl nichtäquidistanter Stützstellen vermieden werden. □

Horner Schema Der Vollständigkeit halber sei noch die numerisch stabile *Auswertung* von Polynomen erwähnt. Für $n = 3$ könnte man den Wert eines Polynoms an der Stelle x in der Monomdarstellung folgendermaßen ausrechnen:

$$\begin{aligned} p(x) &= a_3x^3 + a_2x^2 + a_1x + a_0 \\ &= (a_3x + a_2)x^2 + a_1x + a_0 \\ &= ((a_3x + a_2)x + a_1)x + a_0 \quad . \end{aligned}$$

Allgemein erhalten wir die folgende Rekursion zur Bestimmung von $p(x)$

$$\begin{aligned} b_n &= a_n \\ b_k &= a_k + x b_{k+1} \qquad k = n-1, \dots, 0 \\ p(x) &= b_0. \end{aligned}$$

Dies nennt man das *Horner-Schema*.

18.2 Anwendungen der Polynominterpolation: Numerische Differentation

Wir betrachten die folgende Aufgabe: Berechne die Ableitung der Ordnung k einer tabellarisch gegebenen Funktion oder einer im Rechner als Programm gegebenen Funktion (d.h. diese ist nur an Punkten auswertbar). Idee zur Lösung der Aufgabe: Erstelle erst ein Interpolationspolynom zu bestimmten Stützstellen, leite dieses ab und werte es aus.

Wir wollen nun diese Idee verfolgen wobei wir zunächst als Ableitungsordnung genau den Polynomgrad wählen (bzw. umgekehrt den Polynomgrad gleich der Ableitungsordnung wählen).

Das i-te Lagrange-Polynom zu gewählten Stützstellen lautet:

$$L_i^{(n)}(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)} = \underbrace{\left(\prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{1}{(x_i - x_j)} \right)}_{=:\lambda_i \in \mathbb{R}} x^n + \alpha_{n-1} x^{n-1} + \cdots + \alpha_0.$$

Nun liefert n-maliges differenzieren:

$$\frac{\mathrm{d}^n}{\mathrm{d}x^n} L_i^{(n)}(x) = \lambda_i n! \qquad \text{(konstant, unabhängig von } x)$$

Damit gilt für die n-te Ableitung eines Interpolationspolynoms vom Grad n:

$$\frac{\mathrm{d}^n}{\mathrm{d}x^n} \underbrace{\left(\sum_{i=0}^{n} y_i L_i^{(n)}(x) \right)}_{p(x)} = n! \sum_{i=0}^{n} y_i \lambda_i, \qquad \text{(unabhängig von } x).$$

Eine Aussage über den Fehler liefert:

Satz 18.5. Sei $f \in C^n[a,b]$ und $a = x_0 < x_1 < \cdots < x_n = b$. Dann gibt es ein $\xi \in (a,b)$, so dass für die durch Interpolation mit einem Polynom vom Grad n an den Stützstellen x_i bestimmte n-te Ableitung mit den Koeffizienten $\lambda_i = \prod_{\substack{j=0 \\ j\neq i}}^{n} (x_i - x_j)^{-1}$ gilt:

$$f^{(n)}(\xi) = n! \sum_{i=0}^{n} y_i \lambda_i$$

Beweis. Führe die Differenzfunktion $g(x) = f(x) - p(x)$ ein. Da p das Interpolationspolynom für f an $n+1$ Stützstellen ist, hat g mindestens $n+1$ Nullstellen. Mit dem Satz von Rolle gilt wieder: $g'(x) = g^{(1)}(x)$ hat mindestens n Nullstellen, $g''(x) = g^{(2)}(x)$ hat mindestens $n-1$ Nullstellen und $g^{(n)}(x)$ hat mindestens $n+1-n = 1$, also mindestens eine, Nullstelle. Eine davon nennen wir ξ und es gilt $0 = g^{(n)}(\xi) = f^{(n)}(\xi) - n! \sum_{i=0}^{n} y_i \lambda_i$. □

Wir erhalten also mit unserem Ansatz den exakten Wert der n-ten Ableitung von f an einer Stelle im Intervall welches durch die Stützstellen aufgespannt wird.
Um einfachere Formeln zu bekommen verwenden wir nun äquidistante Stützstellen, d.h.

$$x_i = x_0 + ih, 0 \leq i \leq n.$$

Damit ist

$$\begin{aligned}\lambda_i &= \frac{1}{\underbrace{\underbrace{(x_i - x_0)}_{ih} \dots \underbrace{(x_i - x_{i-1})}_{h}}_{i \text{ Faktoren positiv}} \underbrace{\underbrace{(x_i - x_{i+1})}_{-h} \dots \underbrace{(x_i - x_n)}_{x_0+ih-x_0-nh=-(n-i)h}}_{n-i \text{ Faktoren negativ}}} \\ &= \frac{1}{h^n (-1)^{n-i} i! (n-i)!} = \frac{(-1)^{n-i}}{h^n n!} \underbrace{\binom{n}{i}}_{\substack{\text{Binominal-} \\ \text{koeffizient)}}}\end{aligned}$$

und damit

$$f^{(n)}(x) \approx \frac{d^n}{dx^n} \left(\sum_{i=0}^{n} y_i L_i^{(n)} \right) = \frac{1}{h^n} \sum_{i=0}^{n} (-1)^{n-i} \binom{n}{i} y_i.$$

Insbesondere erhält man damit

$$\begin{aligned} f^{(1)}(x) &\approx \frac{y_1 - y_0}{h} \\ f^{(2)}(x) &\approx \frac{y_2 - 2y_1 + y_0}{h^2} \\ f^{(3)}(x) &\approx \frac{y_3 - 3y_2 + 3y_1 - y_0}{h^3} \end{aligned}$$

Bisher haben wir die n-te Ableitung aus einem Polynom vom Grad n bestimmt, d.h. Ableitungsordnung und Polynomgrad waren gekoppelt. Im allgemeinen kann man die m-te Ableitung aus einem Polynom vom Grad $n > m$ bestimmen. Der Näherungswert hängt dann von der Auswertestelle x ab. Zum Beispiel liefern $m = 1$, $n = 2$ sowie äquidistante Stützstellen $x_0, x_1 = x_0 + h, x_2 = x_0 + 2h$:

$$p'(x_1) = \frac{y_2 - y_0}{2h} \qquad \text{„zentraler Differenzenquotient“}$$

Mittels Taylorreihenentwicklung zeigt man die Fehlerabschätzungen

$$\begin{aligned} f'(x) &= \frac{f(x+h) - f(x-h)}{2h} + O(h^2) && \text{für } f \in C^3, \\ f''(x) &= \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2) && \text{für } f \in C^4. \end{aligned}$$

Beispiel 18.6. Wir wollen die zweite Ableitung von $f(x) = \sinh(x)$ für $x = 0.6$ mit dem zweiten Differenzenquotient praktisch ermitteln:

$$\frac{d^2}{dx^2} \sinh(x) \approx \frac{\sinh(x+h) - 2\sinh(x) + \sinh(x-h)}{h^2}$$

Den exakten Wert der 2. Ableitung können wir ebenfalls ermitteln:

$$\begin{aligned}
&\sinh(x) = \frac{1}{2}(e^x - e^{-x}),\\
&\frac{d}{dx}\sinh = \cosh = \frac{1}{2}(e^x + e^{-x}),\\
&\frac{d^2}{dx^2}\sinh(x) = \sinh(x).
\end{aligned}$$

Bei Auswertung mit doppelter Genauigkeit erhält man den Wert

$$\sinh(0.6) = 6,366535821482 \cdot 10^{-1}.$$

Dagegen liefert die numerische Differentiation die folgende Tabelle (es sind die jeweils korrekten Ziffern unterstrichen):

h	Differenzenquotient		
$1 \cdot 10^{-1}$	6.3371	$\cdot 10^{-1}$	
$1 \cdot 10^{-2}$	6.3665888	$\cdot 10^{-1}$	
$1 \cdot 10^{-3}$	6.366536352	$\cdot 10^{-1}$	
$1 \cdot 10^{-4}$	6.3665358540	$\cdot 10^{-1}$	Auslöschung
$1 \cdot 10^{-5}$	6.3665017	$\cdot 10^{-1}$	Auslöschung
$1 \cdot 10^{-6}$	6.3671	$\cdot 10^{-1}$	Auslöschung
$\vdots$			
$1 \cdot 10^{-10}$	1.1102	$\cdot 10^{4}$	völlig falsch !

Wir stellen fest:

- Numerische Differentation ist sehr anfällig gegenüber Rundungsfehlern. Da der Zähler zwangsläufig im Limit Differenzen gleichgroßer Zahlen berechnet, findet Auslöschung statt.
- Eine gute Schrittweite sollte man proportional zur Wurzel der Maschinengenauigkeit ϵ wählen, z.B. $h = \sqrt{\epsilon}(|x| + 1)$.
- Eine Abhilfe für dieses Problem bietet die „Extrapolation zum Limes“ im nächsten Kapitel.

Vorlesung 19

Extrapolation zum Limes

19.1 Motivation

Wir betrachten folgende Aufgabe: Eine Größe $a(h) \in \mathbb{R}$ sei im Rechner für $h > 0$ berechenbar, nicht jedoch für $h = 0$. Man möchte nun

$$a(0) = \lim_{h \to 0} a(h)$$

mit guter Genauigkeit berechnen.

Beispiel 19.1. Die Gründe, weshalb $a(0)$ nicht auswertbar ist können dabei vielfältig sein.

a) Berechne

$$a(0) = \lim_{x \to 0} \frac{\cos(x) - 1}{\sin(x)} \qquad (= 0).$$

Hier kann man nicht direkt das Argument $x = 0$ einsetzen, da dann der Nenner Null wird.

b) Numerische Differentiation: Berechne

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}.$$

Hier tritt für $h = 0$ ebenfalls der Nenner Null auf und für kleine h tritt Auslöschung auf.

c) Numerische Integration (werden wir später genauer behandeln):

$$\begin{aligned} \int_a^b f(x)dx &= \lim_{N \to \infty} \sum_{i=1}^{N} \frac{b-a}{N} \, f\left(a + (i - 1/2)\frac{b-a}{N}\right) \\ &= \lim_{N \to \infty} \sum_{i=1}^{N} h \, f\left(a + \frac{2i-1}{2} \, h\right) \quad \text{mit } h := \frac{b-a}{N}. \end{aligned}$$

Für $h \to 0$ steigt der Rechenaufwand immer weiter an, so dass nur bis zu einem (rechnerabhängigen) h_{min} gerechnet werden kann.

d) Numerische Lösung eines Anfangswertproblems (aus der Numerik):

$$y'(t) = f(t, y(t)), \quad y(0) = y_0$$

$$\text{berechne } y(T) \approx y_N \quad \text{mit} \quad y_n = y_{n+1} + hf(t, y_{n-1}) \quad \text{mit } h := \frac{T}{N}$$

Auch hier steigt für $h \to 0$ der Rechenaufwand linear mit $N = T/h$ an. □

19.2 Extrapolationsmethode

Diese basiert auf der folgenden Idee: Zu den Parametern $h_0 > h_1 > \cdots > h_n > 0$ bestimme das Interpolationspolynom $p(h)$ vom Grad n so dass

$$p(h_i) = a(h_i) \qquad i = 0, \ldots, n,$$

und setze $a(0) \approx p(0)$. Man spricht von „Extrapolation“, da $0 \notin [h_0, h_n]$.

Beispiel 19.2. Wir betrachten den Ausdruck aus Nummer a) von oben:

$$a(h) = \frac{\cos(h) - 1}{\sin(h)}$$

und erhalten die folgenden Werte für drei Werte von h:

$$\left.\begin{array}{ll} h_0 = \frac{1}{8} & a(h_0) = -6.258151 \cdot 10^{-2} \\ h_1 = \frac{1}{16} & a(h_1) = -3.126018 \cdot 10^{-2} \\ h_2 = \frac{1}{32} & a(h_2) = -1.562627 \cdot 10^{-2} \end{array}\right\} \begin{array}{c}\text{halbiert sich jeweils,} \\ \text{da } h \text{ halbiert} \\ \text{wird}\end{array}$$

und bei Extrapolation mit dem zugehörigen Polynom vom Grad 2:

$$a(0) \approx p_2(0) = -1.02 \cdot 10^{-5} \qquad \text{(exakter Wert 0)}$$

also sehr viel besser! □

Nun stellt sich die Frage, warum das so gut funktioniert.

Die Funktion $a(x)$ sei $n+1$-mal stetig differenzierbar in einer genügend großen Umgebung von $x = 0$. Dann gibt es zu jedem $h > 0$ (in dieser Umgebung) ein $\xi_h \in [0, h]$, so dass: (Taylorreihe):

$$\underbrace{a(h)}_{\substack{\text{Für } h>0 \\ \text{numerisch} \\ \text{berechenbar}}} = \underbrace{a(0) + a'(0)h + \frac{a''(0)}{2}h^2 + \cdots + \frac{a^{(n)}(0)}{n!}h^n}_{\substack{\text{Polynom in h vom Grad n} \\ \text{Koeffizienten hängen NICHT von h ab!}}} + \underbrace{\frac{a^{(n+1)}(\xi_h)}{(n+1)!}h^{n+1}}_{\substack{\text{abhängig} \\ \text{von } h!}}$$

Nun bilde eine Linearkombination von Auswertungen für paarweise verschiedene h_i (dabei ist $a_j = a^{(j)}(0)/j!$):

$$\begin{aligned} \sum_{i=0}^{n} c_i a(h_i) &= \sum_{i=0}^{n} c_i \left(\sum_{j=0}^{n} a_j h_i^j \right) + \sum_{i=0}^{n} c_i \frac{a^{(n+1)}(\xi_{h_i})}{(n+1)!} h_i^{n+1} \\ &= \sum_{j=0}^{n} a_j \left(\sum_{i=0}^{n} c_i h_i^j \right) + \sum_{i=0}^{n} c_i \frac{a^{(n+1)}(\xi_{h_i})}{(n+1)!} h_i^{n+1} \\ &= a(0) + \sum_{i=0}^{n} c_i \frac{a^{(n+1)}(\xi_{h_i})}{(n+1)!} h_i^{n+1} \end{aligned}$$

wenn wir die c_i so wählen, dass

$$\sum_{i=0}^{n} c_i h_i^j = \begin{cases} 1 & j = 0, \\ 0 & \text{sonst.} \end{cases} \tag{19.1}$$

Wir sehen, dass der Fehler die Ordnung $O(h^{n+1})$ hat, mit $h = \max\{h_0, \ldots, h_n\}$. Die Bestimmungsgleichungen (19.1) können wir auch kompakt schreiben als

$$\begin{aligned} & V^T c = e_0 \\ \text{mit} \quad & e_0 = (1, 0, \ldots, 0)^T \\ \text{und} \quad & V = V[h_0, \ldots, h_n] \text{ der Vandermonde-Matrix (17.1)} \end{aligned}$$

Die Auswertung der Linearkombination mit diesen Koeffizienten ergibt dann:

$$A := \sum_{i=0}^{n} c_i \underbrace{a(h_i)}_{=y_i} = \big(\underbrace{V^{-T} e_0}_{=c}\big)^T y = e_0^T V^{-1} y.$$

Nun betrachte alternativ das Interpolationspolynom p gegeben durch

$$p(h_i) = \sum_{j=0}^{n} b_j h_i^j \stackrel{!}{=} a(h_i) = y_i, \qquad i = 0, \ldots, n$$

und somit $Vb = y$ für dessen Koeffizienten, wieder mit der Vandermondematrix $V = V[h_0, \ldots, h_n]$. Auswerten dieses I-Polynoms an der Stelle Null bedeutet

$$p(0) = b_0 = e_0^T b = e_0^T V^{-1} y = A \text{ von oben!}$$

Die oben beschriebene Elimination der Fehlerterme in der Taylorentwicklung entspricht also genau der Extrapolationsmethode!
Für den Fehler ergibt sich dann unter Einbeziehung der Koeffizienten c:

$$\begin{aligned} |A - a(0)| &= \left| \sum_{i=0}^{n} c_i \frac{a^{(n+1)}(\xi_i)}{(n+1)!} h_i^{n+1} \right| \\ &\underbrace{\leq}_{\substack{\text{wähle} \\ h_i := h r^i \\ \text{z.B. } r = 1/2}} \sum_{i=0}^{n} \left\| V^{-T} e_0 \right\|_\infty \frac{|a^{(n+1)}(\xi_i)|}{(n+1)!} h^{n+1} r^{i(n+1)} \\ &\leq \left\| V^{-T} \right\|_\infty \underbrace{\|e_0\|_\infty}_{=1} \max_{0 \leq i \leq n} |a^{(n+1)}(\xi_i)| \cdot \frac{h^{n+1}}{(n+1)!} \underbrace{\sum_{i=0}^{n} r^{i(n+1)}}_{\substack{= r^{n+1}+1 \\ \text{(geom. Reihe)}}} \\ &\leq \|V^{-T}\|_\infty \max_{0 \leq i \leq n} \underbrace{|a^{(n+1)}(\xi_i)|}_{\text{beschränkt}} \frac{h^{n+1}}{(n+1)!} \underbrace{(1 + r^{n+1})}_{\leq \frac{5}{4} \text{für } r \leq \frac{1}{2} \wedge n \geq 1} \end{aligned}$$

Beachte: diese hohe Ordnung ist nur bei entsprechender Differenzierbarkeit von a möglich.

19.3 Extrapolation für gerade Entwicklungen

Es geht sogar noch besser: Betrachte die Näherung von $f'(x)$ mit zentralem Differenzenquotienten:

$$a(h) := \frac{f(x+h) - f(x-h)}{2h}$$

Mittels Taylorentwicklung von f erhalten wir

$$f(x+h) = f(x) + hf^{(1)}(x) + \frac{h^2}{2}f^{(2)}(x) + \frac{h^3}{3!}f^{(3)}(x) + \cdots + \frac{h^{2n+1}}{(2n+1)!}f^{(2n+1)}(x) + \frac{h^{2n+2}}{(2n+2)!}f^{(2n+2)}(\xi_+)$$
$$f(x-h) = f(x) - hf^{(1)}(x) + \frac{h^2}{2}f^{(2)}(x) - \frac{h^3}{3!}f^{(3)}(x) + \cdots - \frac{h^{2n+1}}{(2n+1)!}f^{(2n+1)}(x) + \frac{h^{2n+2}}{(2n+2)!}f^{(2n+2)}(\xi_-)$$

und damit nach Subtraktion und teilen durch $2h$:

$$\frac{f(x+h)-f(x-h)}{2h} = f'(x) + \frac{h^2}{3!}f^{(3)}(x) + \cdots + \frac{h^{2n}}{(2n+1)!}f^{(2n+1)}(x) + \frac{h^{2n+1}}{(2n+2)!}\left[f^{(2n+2)}(\xi_+) + f^{(2n+2)})(\xi_-)\right].$$

Der Fehler $a(h) - f'(x)$ besitzt somit eine Entwicklung in gerade Potenzen von h (bis zur Ordnung $2n$):

$$a(h) - f'(x) = +\frac{h^2}{3!}f^{(3)}(x) + \cdots + \frac{h^{2n}}{(2n+1)!}f^{(2n+1)}(x) + \frac{h^{2n+1}}{(2n+2)!}\left[f^{(2n+2)}(\xi_+) + f^{(2n+2)}(\xi_-)\right]$$

Da alle ungeraden Terme in der Fehlerdarstellung bei der Subtraktion „von selbst“ wegfallen, kann man mit der gleichen Anzahl von Auswertungen doppelt so viele Fehlerterme eliminieren! Dazu wählt man ein Interpolationspolynom in den Stellen h_i^2.

Natürlich muss die Funktion f entsprechend oft differenzierbar sein um diese hohe Konvergenzordnung zu erreichen.

Vorlesung 20

Trigonometrische Interpolation

20.1 Trigonometrische Summen und Polynome

In dieser Vorlesung betrachten wir die Interpolation *periodischer* Funktionen, d.h. es gebe ein $\omega \in \mathbb{R}, \omega > 0$, die Periode, sodass für eine Funktion $f : \mathbb{R} \to \mathbb{R}$ gilt

$$f(x + \omega) = f(x) \quad \forall x \in \mathbb{R}.$$

Eine Option wäre die Verwendung von Polynomen auf dem Intervall $[0, \omega]$. Dann müsste man entsprechende Forderungen an die Differenzierbarkeit an den Stellen 0 und ω stellen. Dies führt auf die Hermite-Interpolation welche wir hier nicht besprochen haben.

Ein alternativer Ansatz, den wir hier verfolgen wollen, nutzt periodische Grundfunktionen als Ausgangspunkt. Statt Polynomen verwenden wir also trigonometrische Funktionen. Wir setzen als Interpolationsfunktion an:

$$t_n(x) = \frac{a_0}{2} + \sum_{k=1}^{m} \left(a_k \cos(kx) + b_k \sin(kx) \right),$$

die sogenannte „trigonometrische Summe“. $t_n(x)$ ist 2π-periodisch. Ist $f(x)$ eine ω-periodische Funktion, so können wir durch die Transformation $\hat{f}(\hat{x}) = f(\frac{2\pi}{\omega}\hat{x})$ eine 2π-periodische Funktion erhalten und diese interpolieren. Wir beschäftigen uns im folgenden also nur mit 2π-periodischen Funktionen.

$t_n(x)$ hat $2m+1$ Parameter. Wir setzen als Abkürzung $n \coloneqq 2m$ und wählen wieder äquidistante Stützstellen:

$$x_k = \frac{2\pi k}{n+1}, \qquad k = 0, \ldots, n.$$

Beachte: Es gilt $x_0 = 0$ und $x_n < 2\pi$. Der Wert 2π wird erst für $k = n + 1$ erreicht.

Es zeigt sich: Die Interpolationsaufgabe $t_n(x_k) = y_k \coloneqq f(x_k)$, $k = 0, \ldots, n$, ist einfacher in den komplexen Zahlen zu lösen! Dazu betrachten wir das spezielle *komplexwertige* trigonometrische Polynom

$$t_n^*(x) = \sum_{k=0}^{n} c_k \, e^{ikx} = \sum_{k=0}^{n} c_k \, \left(e^{ix}\right)^k.$$

mit der imaginären Einheit $i = \sqrt{-1}$ und Koeffizienten $c_k \in \mathbb{C}$. Die Beziehung zur trigonometrischen Summe ergibt sich mit der Eulerschen Formel

$$e^{ikx} = \cos(kx) + i \sin(kx).$$

Wir untersuchen jedoch zunächst die Eigenschaften der komplexen Exponentialfunktion:

Lemma 20.1 (Komplexe Einheitswurzeln)**.** Setze

$$w_k := e^{ix_k} = e^{i\frac{2\pi k}{n+1}}$$

für alle $k \in \mathbb{Z}$ und gegebenes $n \in \mathbb{N}$. $w_k \in \mathbb{C}$ heißt „k-te Einheitswurzel“ und hat folgende Eigenschaften:

a) $w_k^{n+1} = 1 \quad \forall\, k \in \mathbb{Z}$,
d.h. die w_k sind Lösungen der Gleichung $w^{n+1} - 1 = 0$ in $\mathbb{C}$.
Beweis:

$$w_k^{n+1} = \left(e^{i\frac{2\pi k}{n+1}}\right)^{n+1} = e^{i2\pi k} = \underbrace{\cos(2\pi k)}_{=1} + i\underbrace{\sin(2\pi k)}_{=0} = 1.$$

b) $w_k^j = w_j^k \quad \forall\, j,k \in \mathbb{Z}$
Beweis:

$$w_k^j = \left(e^{i\frac{2\pi k}{n+1}}\right)^j = e^{i\frac{2\pi jk}{n+1}} = \left(e^{i\frac{2\pi j}{n+1}}\right)^k = w_j^k$$

c) $w_k^{-j} = w_j^{-k} \quad \forall\, j,k \in \mathbb{Z}$
Beweis wie b)

d) $w_k^j = w_k^{j \mod (n+1)} = w_{k \mod (n+1)}^j = w_{k \mod (n+1)}^{j \mod (n+1)} \quad \forall j,k \in \mathbb{Z}$
zeige nur die erste Identität Rest analog:
Sei $j = r(n+1) + s$ mit $0 \le s \le n$:

$$w_k^j = e^{i\frac{2\pi kj}{n+1}} = e^{i\frac{2\pi k[r(n+1)+s]}{n+1}} = \underbrace{e^{i\frac{2\pi kr\cancel{(n+1)}}{\cancel{n+1}}}}_{1} \cdot e^{i\frac{2\pi ks}{n+1}} = w_k^{j \mod (n+1)}$$

e) $\sum_{j=0}^n w_k^j = \begin{cases} n+1 & k \mod (n+1) = 0 \\ 0 & \text{sonst} \end{cases}$

O.B.d.A. sei $k < n+1$ (sonst setze $k = k \mod (n+1)$, siehe d)).
Sei $k = 0:\quad w_0^j = e^{ij\cdot 0} = 1 \quad \forall\, j$, also $\sum_{j=0}^n 1 = n+1$.

$k \neq 0$: w_k ist nach a) Lösung von

$$0 = w^{n+1} - 1 = (w-1)\left(w^n + w^{n-1} + \cdots + w + 1\right)$$

Für $k \neq 0$ ist $w_k \neq 1$ also muss der zweite Faktor $\sum_{j=0}^n w_k^j = 0$ sein. Dies ist gerade die Behauptung.

20.2 Diskrete Fourier-Transformation (DFT)

Wir betrachten nun zunächst die Interpolationsaufgabe für das komplexwertige Polynom $t_n^*(x)$ mit $x \in \mathbb{R}$. Wegen $e^{ikx} = \cos(kx) + i\sin(kx)$ ist $t_n^* : \mathbb{R} \to \mathbb{C}$ eine 2π-periodische Funktion. Da $\mathbb{R} \subset \mathbb{C}$ können damit auch reellwertige, periodische Funktionen dargeestellt werden, welche genau mit der trigonometrischen Summe übereinstimmen.

Satz 20.2 (Komplexe trigonometrische Interpolation)**.** Zu gegebenen Zahlen $y_0, \ldots, y_n \in \mathbb{C}$ gibt es genau eine Funktion der Gestalt

$$t_n^*(x) = \sum_{k=0}^{n} c_k e^{ikx}$$

die den Interpolationsbedingungen

$$t_n^*(x_j) = y_j, \qquad j = 0, \ldots, n, \quad x_j = \frac{2\pi j}{n+1},$$

genügt. Die komplexen Koeffizienten sind bestimmt durch

$$c_k = \frac{1}{n+1} \sum_{j=0}^{n} y_j e^{-ijx_k} \qquad \forall\, k = 0, \ldots, n. \tag{20.1}$$

Beweis. Mit der Abkürzung $w = e^{ix}$ gilt

$$t_n^*(x) = \sum_{k=0}^{n} c_k e^{ikx} = \sum_{k=0}^{n} c_k (\underbrace{e^{ix}}_{:=w})^k = \sum_{k=0}^{n} c_k w^k = p_n(w).$$

Da die Transformation $w = e^{ix}$ auf $[0, 2\pi[$ eindeutig ist, entspricht jedem t_n^* also ein komplexes Polynom vom Grad n mit den Interpolationsbedingungen

$$t_n^*(x_j) = p_n(e^{ix_j}) = p_n(w_j) = y_j, \qquad j = 0, \ldots, n.$$

Da die Polynominterpolation zu paarweise verschiedenen Stützstellen (auch im Komplexen) eindeutig ist, gibt es genau ein solches Polynom p_n und damit t_n^*.

Die Koeffizienten c_k ergeben sich (wie bei reellen Polynomen) durch das Lösen des linearen Gleichungssystems

$$p_n(e^{ix_j}) = \sum_{l=0}^{n} c_l \cdot (e^{ix_j})^l = \sum_{l=0}^{n} c_l e^{i\frac{2\pi lj}{n+1}} = \sum_{l=0}^{n} c_l w_j^l \overset{!}{=} y_j, \qquad j = 0, \ldots, n.$$

mit $n+1$ (komplexen) Gleichungen für die $n+1$ (komplexen) Unbekannten.

In diesem Fall kann man das Gleichungssystem aufgrund der Eigenschaften der komplexen Einheitswurzeln explizit auflösen. Für ein $k \in \{0, \ldots, n\}$ gilt:

$$\sum_{j=0}^{n} w_k^{-j} y_j = \sum_{j=0}^{n} w_k^{-j} \underbrace{\left(\sum_{l=0}^{n} c_l w_j^l \right)}_{=y_j} = \sum_{l=0}^{n} c_l \underbrace{\left(\sum_{j=0}^{n} w_j^{l-k} \right)}_{\sum_{j=0}^{n} w_{l-k}^{j} = \begin{cases} n+1 & l=k \\ 0 & l \neq k \end{cases}} = c_k(n+1)$$

und damit

$$c_k(n+1) = \sum_{j=0}^{n} y_j w_k^{-j} \iff c_k = \frac{1}{n+1} \sum_{j=0}^{n} y_j e^{-ijx_k}$$

wie behauptet. □

Im Fall $y_0, \ldots, y_n \in \mathbb{R}$ liefert der Satz ein reellwertiges Interpolationspolynom welches mit der trigonometrischen Summe übereinstimmt. Damit lassen sich die reellen Koeffizienten a_k, b_k aus den komplexen Koeffizienten c_k berechnen wie der folgende Satz zeigt.

Satz 20.3 (Reelle Diskrete Fouriertransformation)**.** Für $n \in \mathbb{N}_0$ gibt es zu gegebenen reellen Zahlen $y_0, \ldots, y_n$ genau eine trigonometrische Summe der Form

$$t_n(x) = \frac{a_0}{2} + \sum_{k=1}^{m} (a_k \cos(kx) + b_k \sin(kx)) + \frac{\theta}{2} a_{m+1} \cos((m+1)x)$$

mit $t_n(x_j) = y_j,\ j = 0, \ldots, n,\ x_j = \frac{2\pi j}{n+1}$, sowie

$$\begin{aligned}
&n \text{ gerade:} && \theta = 0,\ m = \frac{n}{2} && \to \quad a_0, \ldots, a_m, b_1, \ldots, b_m,\\
&n \text{ ungerade:} && \theta = 1,\ m = \frac{n-1}{2} && \to \quad a_0, \ldots, a_{m+1}, b_1, \ldots, b_m.
\end{aligned}$$

Es gibt $n+1$ Koeffizienten, also n gerade: $2 \cdot \frac{n}{2} + 1 = n + 1$ und n ungerade: $2 \cdot \frac{(n-1)}{2} + 2 = n + 1$. Die Koeffizienten sind bestimmt durch:

$$a_k = \frac{2}{n+1} \sum_{j=0}^{n} y_j \cos(jx_k), \quad b_k = \frac{2}{n+1} \sum_{j=0}^{n} y_j \sin(jx_k)$$

Beweis. Die c_k seien die Koeffizienten des komplexen trigonometrischen Polynoms zu den reellen Daten $y_j \in \mathbb{R}$. Setze

$$\begin{aligned}
&a_0 = 2c_0 && \\
&a_k = c_k + c_{n+1-k} && k = 1, \ldots, m\\
&b_k = i(c_k - c_{n+1-k}) && k = 1, \ldots, m\\
&a_{m+1} = 2c_{m+1} && n = 2m+1 (n \text{ ungerade})
\end{aligned}$$

Nun rechnen wir nach, dass mit dieser Wahl gilt $t_n(x_l) = y_l$. Wir beschränken uns auf den Fall $n = 2m$ gerade. Aus der Eulerschen Formel folgen die Darstellungen

$$\cos x = \frac{e^{ix} + e^{-ix}}{2}, \qquad \sin x = \frac{e^{ix} - e^{-ix}}{2i}.$$

Nun setzen wir ein

$$\begin{aligned}
t_n(x_l) &= \frac{a_0}{2} + \sum_{k=1}^{m} (a_k \cos(kx_l) + b_k \sin(kx_l))\\
&= \frac{1}{n+1} \sum_{j=0}^{n} y_j + \sum_{k=1}^{m} \left[(c_k + c_{n+1-k}) \frac{e^{ikx_l} + e^{-ikx_l}}{2} + (i(c_k - c_{n+1-k})) \frac{e^{ikx_l} - e^{-ikx_l}}{2i} \right]\\
&= \frac{1}{n+1} \sum_{j=0}^{n} y_j + \sum_{k=1}^{m} \frac{1}{n+1} \sum_{j=0}^{n} y_j \left[e^{-ijx_k} + e^{ijx_k} \right] \frac{e^{ikx_l} + e^{-ikx_l}}{2}\\
&\quad + \sum_{k=1}^{m} \frac{i}{n+1} \sum_{j=0}^{n} y_j \left[e^{-ijx_k} - e^{ijx_k} \right] \frac{e^{ikx_l} - e^{-ikx_l}}{2i}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{n+1}\sum_{j=0}^{n} y_j + \sum_{j=0}^{n}\frac{y_j}{2(n+1)}\sum_{k=1}^{m}\Big(\\
&\quad e^{-ijx_k}e^{ikx_l} + e^{-ijx_k}e^{-ikx_l} + e^{ijx_k}e^{ikx_l} + e^{ijx_k}e^{-ikx_l} + \\
&\quad e^{-ijx_k}e^{ikx_l} - e^{-ijx_k}e^{-ikx_l} - e^{ijx_k}e^{ikx_l} + e^{ijx_k}e^{-ikx_l}\Big) \\
&= \frac{1}{n+1}\sum_{j=0}^{n} y_j + \sum_{j=0}^{n}\frac{y_j}{2(n+1)}\sum_{k=1}^{m}\Big(2e^{-ijx_k}e^{ikx_l} + 2e^{ijx_k}e^{-ikx_l}\Big) \\
&= \sum_{j=0}^{n}\frac{y_j}{(n+1)}\left(1 + \sum_{k=1}^{m}\left(e^{i\frac{2\pi(n+1-k)(j-l)}{n+1}} + e^{i\frac{2\pi k(j-l)}{n+1}}\right)\right) \\
&= \sum_{j=0}^{n}\frac{y_j}{(n+1)}\left(\sum_{k=0}^{n} e^{i\frac{2\pi k(j-l)}{n+1}}\right) = \sum_{j=0}^{n}\frac{y_j}{(n+1)}\left(\sum_{k=0}^{n} w_{j-l}^{k}\right) = y_l.
\end{aligned}
$$

Hier haben wir wieder die Eigenschaften der komplexen Einheitswurzeln benutzt. Nun rechnen wir noch die Formeln für die Koeffizienten nach. Für a_0 gilt mit dieser Wahl

$$a_0 = 2c_0 = \frac{2}{n+1}\sum_{j=0}^{n} y_j e^{-ijx_0} = \frac{2}{n+1}\sum_{j=0}^{n} y_j.$$

Nun a_k für $k > 0$

$$
\begin{aligned}
a_k = c_k + c_{n+1-k} &= \frac{1}{n+1}\sum_{j=0}^{n} y_j\Big(e^{-ijx_k} + \underbrace{e^{-ijx_{n+1-k}}}_{e^{-i\frac{2\pi j((n+1)-k)}{n+1}} = e^{i\frac{2\pi jk}{n+1}} = e^{ijx_k}}\Big) \\
&= \frac{1}{n+1}\sum_{j=0}^{n} y_j\left(e^{-ijx_k} + e^{ijx_k}\right) \\
&= \frac{1}{n+1}\sum_{j=0}^{n} y_j\Big(\underbrace{\cos(-jx_k)}_{=\cos(jx_k)} + \underbrace{i\sin(-jx_k)}_{-\sin(jx_k)} + \cos(jx_k) + i\sin(jx_k)\Big) \\
&= \frac{2}{n+1}\sum_{j=0}^{n} y_j\cos(jx_k)
\end{aligned}
$$

und b_k:

$$
\begin{aligned}
b_k = i(c_k - c_{n+1-k}) &= \frac{i}{n+1}\sum_{j=0}^{n} y_j\Big(e^{-ijx_k} - \underbrace{e^{-ijx_{n+1-k}}}_{e^{-i\frac{2\pi j((n+1)-k)}{n+1}} = e^{i\frac{2\pi jk}{n+1}} = e^{ijx_k}}\Big) \\
&= \frac{i}{n+1}\sum_{j=0}^{n} y_j\left(e^{-ijx_k} - e^{ijx_k}\right) \\
&= \frac{i}{n+1}\sum_{j=0}^{n} y_j\Big(\underbrace{\cos(-jx_k)}_{=\cos(jx_k)} + \underbrace{i\sin(-jx_k)}_{-\sin(jx_k)} - \cos(jx_k) - i\sin(jx_k)\Big) \\
&= \frac{1}{n+1}\sum_{j=0}^{n} y_j(-2i^2\sin(jx_k)) = \frac{2}{n+1}\sum_{j=0}^{n} y_j\sin(jx_k)
\end{aligned}
$$

□

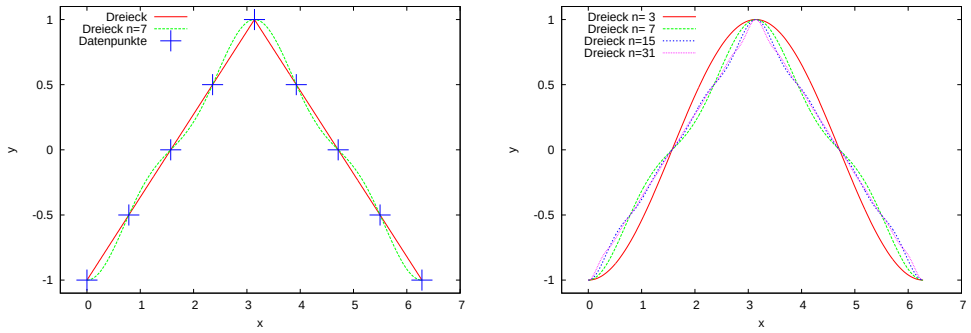


Abbildung 20.1: Interpolation eines Dreieckssignals.

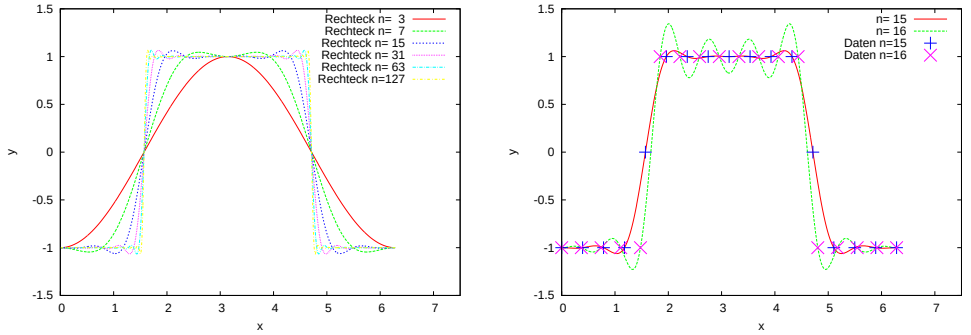


Abbildung 20.2: Interpolation eines Rechteckssignals. Links ist die Sprungstelle Interpolationspunkt mit vorgeschriebenem Mittelwert und rechts ist die Sprungstelle kein Interpolationspunkt.

Beispiel 20.4. Wir betrachten nun die Interpolation verschiedener periodischer Funktionen mittels trigonometrischer Summe. Abbildung **??** zeigt die Interpolation einer dreiecksförmigen Funktion

$$f_\Delta(x) = \begin{cases} \frac{2}{\pi}\left(x - \frac{\pi}{2}\right) & 0 \le x < \pi \\ \frac{2}{\pi}\left(\frac{3\pi}{2} - x\right) & \pi \le x < 2\pi \end{cases}$$

bei Interpolation mit einer steigenden Zahl von Stützstellen. Die Dreiecksfunktion ist stetig aber nicht stetig differenzierbar.

Als zweites Beispiel betrachten wir die unstetige Rechteckfunktion, die wir folgendermaßen definieren:

$$f_R(x) = \begin{cases} -1 & 0 \le x < \pi/2 \\ 0 & 0 \le x = \pi/2 \\ 1 & \pi/2 < x < 3\pi/2 \\ 0 & x = 3\pi/2 \\ -1 & 3\pi/2 < x < 2\pi \end{cases}$$

Wählt man $n = 2^k - 1$ dann lauten die Stützstellen $x_k = \frac{2\pi x}{n+1}$ und die Stellen $\frac{\pi}{2}$ und $\frac{3\pi}{2}$ sind gerade Stützstellen. Dort haben wir den Funktionswert als

$$\frac{1}{2}\left(\lim_{x\to\zeta^+} f(x) + \lim_{x\to\zeta^-} f(x)\right)$$

für $\zeta \in \{\frac{\pi}{2}, \frac{3\pi}{2}\}$ definiert. Damit ergeben sich bei steigenden Stützstellen die Ergebnisse in Abbildung 20.2 links. Wählt man jedoch $n = 2^k$ dann ergibt sich das Bild rechts. In diesem Fall sind die Unstetigkeitsstellen keine Stützpunkte und es ergeben sich sich sehr viel größere sogenannte Über- und Unterschwinger. Diese nennt man auch „Gibbssches Phänomen“. Auch bei weiterer Erhöhung der Zahl der Stützstellen geht der Fehler in der Maximumnorm direkt an der Unstetigkeitsstelle nicht gegen Null. □

Bemerkung 20.5 (Bezug zur Fourierreihe)**.** Die trigonometrische Interpolation kann man als Näherung der Fourierreihe einer 2π-periodischen Funktion $f : \mathbb{R} \to \mathbb{R}$ betrachten. Unter gewissen Voraussetzungen (f ist eine stetig differenzierbare, 2π-periodische Funktion oder f ist eine quadratintegrable Funktion auf dem Intervall $[-\pi, \pi]$) konvergiert die unendliche Reihe

$$g(x) = \sum_{k=-\infty}^{\infty} c_k e^{ikx}$$

mit den Koeffizienten

$$c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-ikx}\, dx$$

gleichmäßig gegen die Funktion f für alle $x \in [-\pi, \pi]$. Die Koeffizienten aus Satz 20.2 erhält man, wenn man die Integrale numerisch mit der summierten Trapezregel berechnet.

Andererseits kann man die trigonometrische Interpolation auch als eine Diskretisierung der Fouriertransformation sehen, man spricht deshalb auch von diskreter Fourier-Transformation (DFT). Für eine Funktion $f : \mathbb{R} \to \mathbb{C}$ berechnet sich die Fourier-Transformierte $\mathcal{F}f$ als

$$(\mathcal{F}f)(\omega) = \int_{-\infty}^{\infty} e^{-i\omega x} \cdot f(x)\, dx, \quad \omega \in \mathbb{R}.$$

Die Fourier-Transformierte ist nicht mehr eine Funktion des Ortes (oder der Zeit, je nach Bedeutung von x) sondern eine Funktion der Kreisfrequenz ω (bzw. der Frequenz). Man spricht daher auch von einer Transformation vom Orts- in den Frequenzraum. Die Fourier-Transformierte gibt Auskunft über das (kontinuierliche) Spektrum einer Funktion (also den Anteil von bestimmten Frequenzen in einem Signal) man spricht deshalb auch von der Spektralfunktion. Bei der DFT übernimmt k die Rolle von ω und man spricht von einem diskreten Spektrum.

20.3 Schnelle Fourier-Transformation (FFT)

In diesem Abschnitt behandeln wir die Frage wie man die DFT effizient berechnen kann. Hierbei handelt es sich um einen der berühmtesten Algorithmen der angewandten Mathematik bzw. Informatik, der von James Cooley und John Tukey im Jahr 1965 publiziert wurde.

Die DFT entspricht den Korrespondenzen

$$\begin{aligned} c_k &= \frac{1}{N}\sum_{j=0}^{N-1} y_j e^{-i\frac{2\pi jk}{N}} \quad \text{für } k = 0,\dots,N-1 \text{ (Hintransformation)} \\ y_j &= \sum_{j=0}^{N-1} c_k e^{i\frac{2\pi jk}{N}} \quad \text{für } j = 0,\dots,N-1 \text{ (Rücktransformation)} \end{aligned} \tag{20.2}$$

wobei wir $N = n+1$ gesetzt haben.

- Die Hintransformation entspricht der Berechnung der Koeffizienten des trigonometrischen Polynoms aus den Daten y_i aus Satz 20.2.
- Die Rücktransformation entspricht der Auswertung des trigonometrischen Polynoms an den Stellen x_j wobei man die y_j erhält.
- Der Rechenaufwand für die Hin- oder Rücktransformation aller N Werte ist $O(N^2)$ (N-maliges Auswerten von von N Summentermen).

Wir zeigen nun wie man diesen Rechenaufwand signifikan reduzieren kann. Dazu setze zur Abkürzung $\tilde{c}_k = Nc_k$ und es sei angenommen, dass $N = 2n$ gerade ist. Dann lassen sich die Summen für die Koeffizienten in zwei Teilsummen aufspalten:

$$\tilde{c}_k = \sum_{j=0}^{2n-1} y_j e^{\frac{-2\pi i}{2n}jk} = \underbrace{\sum_{j=0}^{n-1} y_{2j} e^{\frac{-2\pi i}{2n}2jk}}_{\text{gerader Teil}} + \underbrace{\sum_{j=0}^{n-1} y_{2j+1} e^{\frac{-2\pi i}{2n}(2j+1)k}}_{\text{ungerader Teil}}$$

mit $y_j^g = y_{2j}$, $\tilde{c}_k^g = \tilde{c}_{2k}$, $y_j^u = y_{2j+1}$, $\tilde{c}_k^u = \tilde{c}_{2k+1}$ ergibt sich nach ausklammern:

$$\tilde{c}_k = \underbrace{\sum_{j=0}^{n-1} y_j^g \cdot e^{-\frac{2\pi i}{n}jk}}_{:=\tilde{c}_k^g} + e^{-\frac{2\pi i}{2n}k} \underbrace{\sum_{j=0}^{n-1} y_j^u e^{-\frac{2\pi i}{n}jk}}_{:=\tilde{c}_k^u}$$

Wegen der $n = \frac{N}{2}$-Periodizität der Einheitswurzeln $e^{-\frac{2\pi i}{n}k}$ gilt:

$$\tilde{c}_{k+\frac{N}{2}}^g = \tilde{c}_k^g \qquad \text{und} \qquad \tilde{c}_{k+\frac{N}{2}}^u = \tilde{c}_k^u, \qquad k = 0,\dots,\frac{N}{2}-1,$$

d.h. das Problem lässt sich auf die Berechnung von zwei Fourier Transformierten der Länge $N/2$ reduzieren.

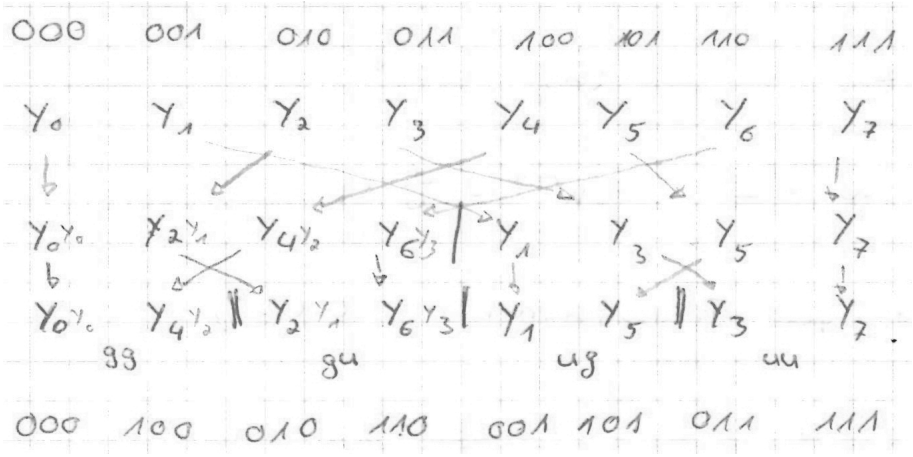
- $\tilde{c}_k^g$ und $\tilde{c}_k^u$ für $k = 0,\dots,\frac{N}{2}-1$ berechnen sich jeweils durch eine DFT der Länge $n = \frac{N}{2}$.
- Daraus berechnet man die ursprünglich gesuchten Koeffizienten mittels

$$\begin{aligned} \tilde{c}_k &= \tilde{c}_k^g + e^{-i\frac{2\pi k}{N}}\tilde{c}_k^u && 0 \le k < \frac{N}{2} \\ \tilde{c}_k &= \tilde{c}_{k-\frac{N}{2}}^g + e^{-i\frac{2\pi k}{N}}\tilde{c}_{k-\frac{N}{2}}^u && \frac{N}{2} \le k < N \end{aligned} \tag{20.3}$$

- Falls $\frac{N}{2}$ wieder gerade ist, kann man das Prinzip rekursiv fortsetzen.

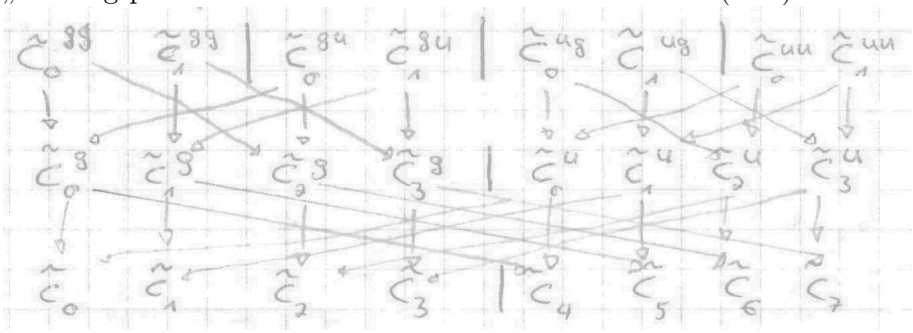
- Ist $N = 2^d$ eine Zweierpotenz, erhält man schließlich eine DFT der Länge 1, die gleich dem Eingangswert ist.

Beispiel 20.6. „Abstiegsphase“ der Rekursion: Umsortieren der Eingabedaten. Am besten schreibt man die Indizes zur Basis 2.



Die dabei auftretende Permutation der Indizes $(b_{d-1}, \ldots, b_0) \to (b_0, b_1, \ldots, b_{d-1})_2$ heißt „bit-reversal“.

„Aufstiegsphase“: Rekombination der Koeffizienten nach (20.3).



Dieses Verknüfpungsmuster nennt man „perfect shuffle“.

Rechenaufwand der FFT

für $N = 2^d$ (Zweierpotenz $\Rightarrow d = \log_2 N$)

$$
\begin{aligned}
A(N) = & \underbrace{2A\left(\frac{N}{2}\right)}_{\substack{\text{2 Transf. der} \\ \text{Länge } N/2 \text{ ber.}}} + \underbrace{c \cdot N}_{\text{Aufwand für (20.3)}} \quad) \\
& = 2\,[2 \cdot A\left(\frac{N}{4}\right) + c\frac{N}{2}] + c \cdot N \\
& = 4 \cdot A\left(\frac{N}{4}\right) + c \cdot N + c \cdot N \text{ (d-mal}\ldots\text{)} \\
& = \underbrace{2^d}_{=N}\underbrace{A(1)}_{\leq c} + \underbrace{cN + \cdots + cN}_{d-1 \text{ Summanden}} \\
& \leq d \cdot c \cdot N = O(N \log_2 N)
\end{aligned}
$$

Statt $O(N^2)$ erhält man also den Aufwand $O(N \log_2 N)$.

20.4 Spektralanalyse

Mit der Diskreten Fourier-Analyse kann man ein periodisches Signal durch Sinus und Kosinusschwingungen unterschiedlicher Frequenzen darstellen. Die Koeffizienten a_k, b_k bilden das Spektrum des Signals. Man spricht auch von Zeit- und Frequenzbereich.

Dies hat unzählige Anwendungen in der sog. Signalverarbeitung. So arbeitet das JPEG Verfahren zur Bildkompression mit einer diskreten Kosinustransformation (eine Variante der diskreten Fourier-Analyse, die nur mit reellen Werten rechnet) und Abschneiden im Frequenzbereich.

Einen Spektrumanalysator hat vielleicht auch schon jeder mal an einem Verstärker oder MP3-Player gesehen.

Beispiel 20.7. Wir wollen uns nun die Spektren zu einigen einfachen Signalen sowie ihre trigonometrische Interpolation anschauen.

Abbildung 20.3 zeigt einige Beispiele für Spektren. Die Konstante im Zeitbereich hat einen Puls als Spektrum. Umgedreht hat ein Puls im Ortsbereich ein konstantes Spektrum (sogenanntes „weisses Rauschen“). Schließlich wird noch das Spektrum eines Dreiecks- bzw. Rechtecksignals gezeigt. Im Frequenzbereich ist $\sqrt{a_k^2 + b_k^2}$ als Wert geplottet. □

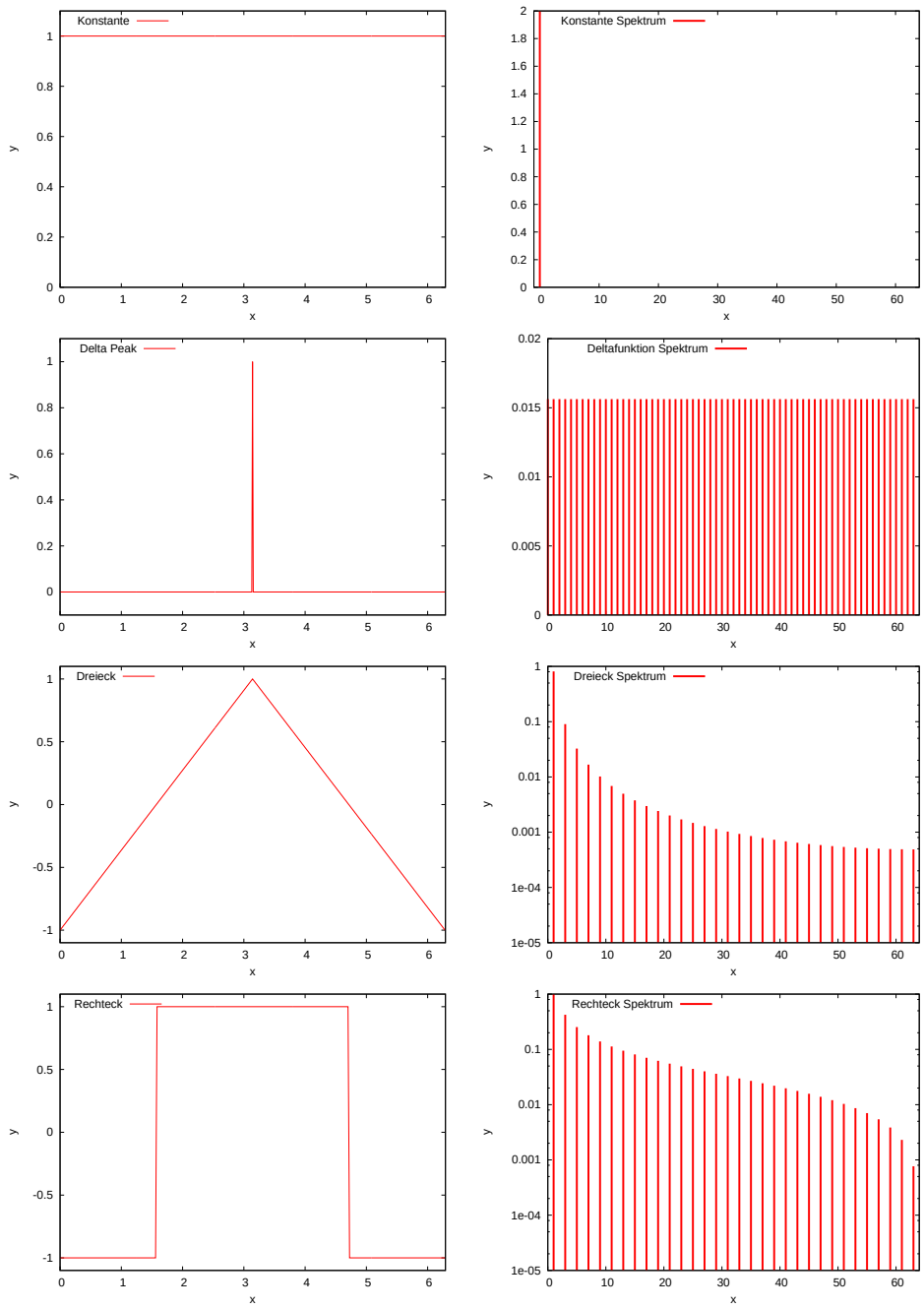


Abbildung 20.3: Signal (links) und Frequenzspektrum (rechts) zu einer konstanten Funktion (oben), einem Delta-Peak (2. Zeile von oben), einem Dreieck (3. Zeile von oben) und einem Rechteckssignal (unten). Man beachte die logarithmische Darstellung im Frequenzbereich.

Vorlesung 21

Interpolation mit Splines

Bis jetzt war die Anzahl der Stützstellen direkt an den Polynomgrad gekoppelt. Bei der Lagrangeinterpolation haben wir festgestellt, dass insbesondere die Wahl äquidistanter Stützstellen zu starken Ausschlägen des Interpolationspolynoms zum Rand des Intervalles hin führen kann. Eine Möglichkeit dies zu beheben ist die stückweise Interpolation mit Polynomen niedrigen Grades. Dann kann die Anzahl Stützstellen vom Polynomgrad entkoppelt werden.

21.1 Spline-Räume

Definition 21.1. Sei $X = (x_0, x_1, \ldots, x_n)$ mit $a = x_0 < x_1 < \ldots < x_n = b$ eine Zerlegung des Intervalles $[a, b]$ und sei $m \in \mathbb{N}$. Die Menge von Funktionen

$$S^m(X) = \left\{ s \in C^{m-1}[a,b] : \quad s|_{[x_i, x_{i+1}]} \in P_m,\ 0 \leq i < n \right\}$$

heißt Spline-Raum vom Grad m über der Zerlegung X. □

$S^m(X)$ ist ein endlichdimensionaler Vektorraum.

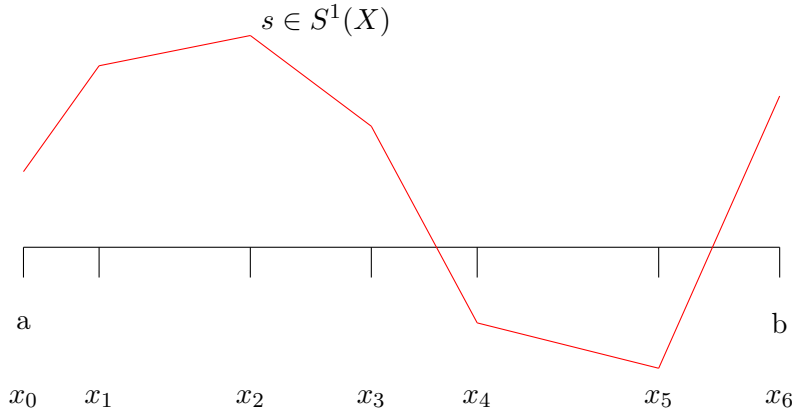
Beispiel 21.2. Als Beispiel betrachte $S^1(X)$. Dies bedeutet:

i) $S^1(X) \subset C^0[a,b]$, also $s \in S^1(X)$ stetig.

ii) $s \in S^1(X)$ ist Polynom vom Grad 1 auf jedem Teilintervall $[x_i, x_{i+1}]$, also stückweise linear.

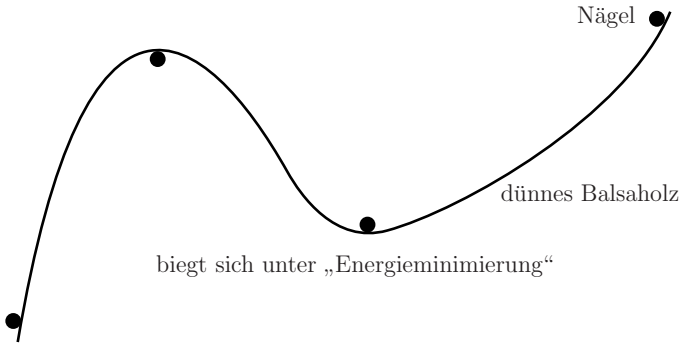
iii) Ein $s \in S^1(X)$ ist eindeutig durch Werte in den Stützstellen beschrieben, d.h. für die Dimension gilt $\dim S^1(X) = \#X$.

Hier eine graphische Darstellung:



□

In der Praxis ist $S^3(X)$ sehr beliebt. $S^3(X)$ heißt Raum der kubischen Splines. Historisch hat man im Schiffs- und Flugzeugbau sogenannte Straklatten eingesetzt um glatte Kurven zu erhalten. Dazu fixierte man dünnes, biegsames Holz durch Nägel wie in folgender Abbildung gezeigt:



Im Ruhezustand wird dabei die Energie, bzw. die Krümmung, minimiert:

$$\underbrace{\int_a^b \frac{|y''(t)|^2}{1+|y'(t)|^2}\,\mathrm{d}t}_{\text{totale Krümmung}} \overset{|y'(t)|\ll 1}{\approx} \int_a^b |y''(t)|^2\,\mathrm{d}t \to \min.$$

21.2 Konstruktion kubischer Splines

Nun betrachten wir die Aufstellung eines kubischen Splines $s \in S^3(X)$. Die Funktion setzt sich stückweise aus n Polynomen zusammen:

$$s(x) = \begin{cases} p_i(x) & x \in [x_{i-1}, x_i), \quad i \in \{1,\dots,n\}, \\ p_n(x) & x = x_n. \end{cases}$$

Für die Polynome p_i gelten folgende Bedingungen:

a) Interpolationsbedingung (Stetigkeit):

$$i = 1,\dots,n\colon \quad \left.\begin{matrix} p_i(x_{i-1}) = y_{i-1} \\ p_i(x_i) = y_i \end{matrix}\right\} 2n \text{ Bedingungen}$$

b) Stetigkeit der ersten und zweiten Ableitung an den inneren Punkten:

$$i = 1,\dots,n-1\colon \left.\begin{matrix} p_i'(x_i) = p_{i+1}'(x_i) \\ p_i''(x_i) = p_{i+1}''(x_i) \end{matrix}\right\} 2(n-1) = 2n-2 \text{ Bedingungen}$$

ergibt zusammen $4n-2$ Bedingungen.

Pro Polynom p_i (vom Grad 3) hat man 4, also insgesamt $4n$ Freiheitsgrade (Parameter) zu bestimmen.

Die fehlenden zwei Bedingungen erhält man durch *Randbedingungen* an den Stellen x_0 und x_n.

c) Randbedingungen. Wähle eine der folgenden Varianten:

(i)	Natürliche Randbedingungen:	$p_1''(x_0) = 0,$	$p_n''(x_n) = 0.$
(ii)	Hermite-Randbedingungen:	$p_1'(x_0) = f'(x_0),$	$p_n'(x_n) = f'(x_n).$
(iii)	Periodische Randbedingungen:	$p_1'(x_0) = p_n'(x_n),$	$p_1''(x_0) = p_n''(x_n).$

Wir behandeln im folgenden nur die natürlichen Randbedingungen.

Satz 21.3 (Berechnung kubischer Splines)**.** Wir schreiben die Teilpolynome des Splines für $x \in [x_{i-1}, x_i)$ in der Form

$$p_i(x) = a_0^{(i)} + a_1^{(i)}(x - x_i) + a_2^{(i)}(x - x_i)^2 + a_3^{(i)}(x - x_i)^3 \qquad i = 1, \ldots, n.$$

Die $a_2^{(i)}$ sind dann die Lösung des linearen Gleichungssystems der Dimension $n - 1$:

$$h_i a_2^{(i-1)} + 2(h_i + h_{i+1})a_2^{(i)} + h_{i+1}a_2^{(i+1)} = 3\left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}\right), \quad i = 1, \ldots, n-1, \tag{21.1}$$

wobei $a_2^{(0)} = a_2^{(n)} = 0$ (natürliche Randbedingung!) und $h_i \coloneqq x_i - x_{i-1}$. Der Wert $a_2^{(0)}$ ist dabei fiktiv, da es kein Polynom p_0 gibt. Die restlichen Koeffizienten ergeben sich zu:

$$a_0^{(i)} = y_i, \tag{21.2}$$

$$a_1^{(i)} = \frac{y_i - y_{i-1}}{h_i} + \frac{h_i}{3}\left(2a_2^{(i)} + a_2^{(i-1)}\right), \tag{21.3}$$

$$a_3^{(i)} = \frac{a_2^{(i)} - a_2^{(i-1)}}{3h_i}. \tag{21.4}$$

Beweis. i) Berechne Ableitungen der Teilpolynome

$$p_i'(x) = a_1^{(i)} + 2a_2^{(i)}(x - x_i) + 3a_3^{(i)}(x - x_i)^2 \tag{21.5}$$

$$p_i''(x) = 2a_2^{(i)} + 6a_3^{(i)}(x - x_i) \tag{21.6}$$

(ii) Interpolationsbedingung nutzen: Einsetzen von x_i, y_i:

$$y_i = p_i(x_i) = a_0^{(i)} \Rightarrow \boxed{a_0^{(i)} = y_i} \quad i = 1, \ldots, n. \tag{21.7}$$

Das ist (21.2).

Einsetzen von x_{i-1}:

$$y_{i-1} = p_i(x_{i-1}) = \underbrace{a_0^{(i)}}_{=y_i} + a_1^{(i)}(-h_i) + a_2^{(i)}h_i^2 + a_3^{(i)}(-h_i^3) \quad i = 1, \ldots, n$$

$$\Leftrightarrow y_{i-1} - y_i = -h_i a_1^{(i)} + h_i^2 a_2^{(i)} - h_i^3 a_3^{(i)} \tag{21.8}$$

(iii) Randbedingungen einsetzen. Wir behandeln nur natürliche:

$$0 = p_1''(x_0) = 2a_2^{(1)} - 6a_3^{(1)}h_1 \tag{21.9}$$

$$0 = p_n''(x_n) = 2a_2^{(n)} \Rightarrow \boxed{a_2^{(n)} = 0} \tag{21.10}$$

(iv) Stetigkeit der ersten Ableitung

$$p_i'(x_i) = p_{i+1}'(x_i) \quad i = 1, \ldots, n-1$$

$$\Leftrightarrow a_1^{(i)} = a_1^{(i+1)} - 2a_2^{(i+1)}h_{i+1} + 3a_3^{(i+1)}h_{i+1}^2 \tag{21.11}$$

(v) Stetigkeit der zweiten Ableitung

$$\begin{aligned} & p_i''(x_i) = p_{i+1}''(x_i) \quad i = 1, \ldots, n-1 \\ \Leftrightarrow\ & 2a_2^{(i)} = 2a_2^{(i+1)} - 6a_3^{(i+1)} h_{i+1} \end{aligned} \tag{21.12}$$

(vi) Drücke $a_3^{(i)}$ durch $a_2^{(i)}$ aus. Löse (21.12) nach $a_3^{(i+1)}$ auf:

$$a_3^{(i+1)} = \frac{a_2^{(i+1)} - a_2^{(i)}}{3h_{i+1}} \quad i = 1, \ldots, n-1.$$

Umnummerieren liefert:

$$\Leftrightarrow \boxed{a_3^{(i)} = \frac{a_2^{(i)} - a_2^{(i-1)}}{3h_i}} \quad i = 2, \ldots, n \tag{21.13}$$

Für $i = 1$ erhält man das Gleiche aus der Randbedingung (21.9) wenn man die Vereinbarung $a_2^{(0)} = 0$ einführt. Damit gilt diese Formel auch für $i = 1, \ldots, n$. Das ist (21.4).

(vii) $a_1^{(i)}$ durch $a_2^{(i)}$ ausdrücken. Dazu löse (21.8) nach $a_1^{(i)}$ auf und setze (21.13) ein:

$$\begin{aligned} a_1^{(i)} &= \frac{y_i - y_{i-1}}{h_i} + h_i a_2^{(i)} - h_i^2 a_3^{(i)} \qquad i = 1, \ldots, n \\ &\overset{(21.13)}{=} \frac{y_i - y_{i-1}}{h_i} + h_i a_2^{(i)} - h_i^2 \left(\frac{a_2^{(i)} - a_2^{(i-1)}}{3h_i} \right) \\ &\boxed{a_1^{(i)} = \frac{y_i - y_{i-1}}{h_i} + \frac{h_i}{3} \left(2a_2^{(i)} + a_2^{(i-1)} \right)} \quad i = 1, \ldots, n \end{aligned}$$

Das ist (21.3).

(viii) Nun setze $a_1^{(i)}$ und $a_3^{(i)}$ in die verbleibende Gleichung (21.11) ein:

$$\begin{aligned} & \frac{y_i - y_{i-1}}{h_i} + \frac{h_i}{3} \left(2a_2^{(i)} + a_2^{(i-1)} \right) \\ & \qquad = \frac{y_{i+1} - y_i}{h_{i+1}} + \frac{h_{i+1}}{3} \left(2a_2^{(i+1)} + a_2^{(i)} \right) - 2a_2^{(i+1)} h_{i+1} + 3h_{i+1}^2 \left(\frac{a_2^{(i+1)} - a_2^{(i)}}{3h_{i+1}} \right) \quad i \\ & \Longleftrightarrow h_i a_2^{(i-1)} + a_2^{(i)} \left(2h_i - h_{i+1} + 3h_{i+1} \right) + a_2^{(i+1)} \left(-2h_{i+1} + 6h_{i+1} - 3h_{i+1} \right) \\ & \qquad = 3 \left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i} \right) \qquad i = 1, \ldots, n-1 \end{aligned}$$

$$\Longleftrightarrow \boxed{h_i a_2^{(i+1)} + 2 \left(h_i + h_{i+1} \right) a_2^{(i)} + h_{i+1} a_2^{(i+1)} = 3 \left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i} \right)}.$$

Und das ist (21.1). Beachte, dass in (iii) und (vi) $a_2^{(0)} = a_2^{(n)} = 0$ gesetzt wurde (natürliche Randbedingungen).

□

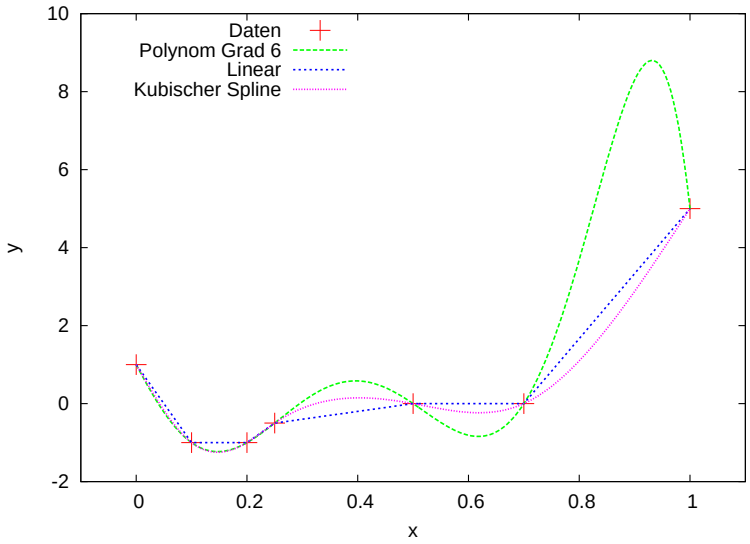


Abbildung 21.1: Vergleich der Interpolation mit Polynomen, stückweise linearen Funktionen und kubischen Splines.

Bemerkung 21.4 (Zur Lösung des Gleichungssystems)**.** Das bei kubischen Splines (auf einem Intervall) entstehende Gleichungssystem ist symmetrisch und strikt diagonaldominant. Damit ist eine stabile LR-Zerlegung ohne Pivotisierung möglich. Weiter ist das System tridiagonal und es entsteht bei der Zerlegung kein fill-in. Der Rechenaufwand ist linear in der Anzahl der Stützstellen. □

Beispiel 21.5. Es sei folgende Wertetabelle zu interpolieren:

x	0	0.1	0.2	0.25	0.5	0.7	1.0
y	1	-1	-1	-0.5	0	0	5

Abbildung 21.1 zeigt die Interpolation mit einem Polynom vom Grad 6, linearen Splines und kubischen Splines (mit natürlichen Randbedigungen) anhand eines Beispiels. Deutlich zu erkennen ist der starke „Überschwinger“ bei der Polynominterpolation. □

Der Vollständigkeit halber zitieren wir noch ein Resultat zur Fehlerabschätzung für kubische Splines.

Satz 21.6 (Fehlerabschätzung)**.** Sei $f \in C^4[a,b]$. Erfüllt der kubische Spline Hermite-Randbedingungen so gilt

$$\max_{a\le x\le b} |f(x) - s(x)| \le h^4 \max_{a\le x\le b} |f^{(4)}(x)|$$

für

$$h := \max_{1\le i\le n} |x_i - x_{i-1}|$$

Beweis. [Schaback and Wendland, 2005, Satz 11.13]. □

Selbst unter (wesentlich) schwächeren Voraussetzungen konvergiert die Spline-Interpolation gleichmäßig gegen f.

21.3 Praktisches Beispiel zur Spline-Interpolation

Beispiel 21.7 (Zur Konvergenzordnung stückweiser Polynome)**.** Wir betrachten die Interpolation der folgenden drei Funktionen, die in Abbildung 21.2 dargestellt sind:

$$f_1(x) = \exp(-x^2) \qquad \text{in } [-10, 10], \tag{21.14}$$

$$f_2(x) = \begin{cases} \cos^2(x) & |x| < \frac{\pi}{2} \\ 0 & |x| \geq \frac{\pi}{2} \end{cases} \qquad \text{in } [-\pi, \pi], \tag{21.15}$$

$$f_3(x) = \begin{cases} -1 & x < \frac{1}{2} \\ +1 & x \geq \frac{1}{2} \end{cases} \qquad \text{in } [0, 1], \tag{21.16}$$

mittels Polynomen, $S_h^{1,0}$ und $S_h^{3,2}$ (mit natürlichen Randbedingungen, alle Funktionen f_i erfüllen (näherungsweise) $f_i'' = 0$ an den Randpunkten).

Die Ergebnisse sind in den Abbildungen 21.3, 21.4 und 21.5 dargestellt. Für alle Funktionen beobachten wir das folgende qualitative Verhalten:

a) Die Interpolation mit Polynomen steigenden Grades an äquidistanten Stützstellen schlägt in allen Fällen fehl, d. h. der Interpolationsfehler (in der Maximumnorm) steigt mit dem Grad an.

b) Kubische Splines konvergieren und liefern einen glatten Verlauf. Allerdings kommt es zu möglicherweise „unphysikalischen“ Unter- bzw. Überschwingern, ähnlich wie bei der trigonometrischen Interpolation. Diese sind aber selbst im Fall von $f_3(x)$ um die Sprungstelle lokalisiert.

c) Stückweise lineare Funktionen haben diesen Defekt nicht.

Wir bestimmen nun den Interpolationsfehler in der Maximumnorm noch experimentell durch Auswertung an vielen Zwischenstellen. Bei der Funktion $f_1(x) = e^{-x^2}$ erhalten wir die Tabelle

n	$S_h^{1,0}$	$S_h^{3,2}$	p_n
4	$6.045 \cdot 10^{-1}$	$7.420 \cdot 10^{-1}$	$8.038 \cdot 10^{-1}$
6	$4.447 \cdot 10^{-1}$	$5.612 \cdot 10^{-1}$	$9.999 \cdot 10^{-1}$
8	$3.002 \cdot 10^{-1}$	$3.918 \cdot 10^{-1}$	2.311
10	$1.774 \cdot 10^{-1}$	$2.464 \cdot 10^{-1}$	5.949
16	$1.060 \cdot 10^{-1}$	$2.753 \cdot 10^{-2}$	
32	$6.946 \cdot 10^{-2}$	$7.083 \cdot 10^{-3}$	
64	$2.241 \cdot 10^{-2}$	$3.316 \cdot 10^{-4}$	
128	$5.974 \cdot 10^{-3}$	$1.918 \cdot 10^{-5}$	
256	$1.517 \cdot 10^{-3}$	$1.173 \cdot 10^{-6}$	
512	$3.809 \cdot 10^{-4}$	$7.289 \cdot 10^{-8}$	
1024	$9.533 \cdot 10^{-5}$	$4.549 \cdot 10^{-9}$	
2048	$2.383 \cdot 10^{-5}$	$2.842 \cdot 10^{-10}$	

Polynome konvergieren offensichtlich nicht. Der stückweise lineare Spline konvergiert mit der Ordnung h^2 (d. h. $e_{2n}/e_n = (1/2)^2$), kubische Splines konvergieren mit der Ordnung h^4 (d. h. $e_{2n}/e_n = (1/2)^4$). In beiden Fällen gilt dies nur, wenn n genügend groß, man spricht von „asymptotischer“ Konvergenz.

Der Fehler bei Interpolation der Funktion

$$f_2(x) = \begin{cases} \cos^2(x) & x < \pi/2 \\ 0 & x \geq \pi/2 \end{cases}$$

ist in folgender Tabelle angegeben:

n	$S_h^{1,0}$	$S_h^{3,2}$
4	$1.052 \cdot 10^{-1}$	$1.649 \cdot 10^{-1}$
8	$1.052 \cdot 10^{-1}$	$4.498 \cdot 10^{-2}$
16	$3.518 \cdot 10^{-2}$	$8.434 \cdot 10^{-3}$
32	$9.423 \cdot 10^{-3}$	$1.945 \cdot 10^{-3}$
64	$2.396 \cdot 10^{-3}$	$4.764 \cdot 10^{-4}$
128	$6.015 \cdot 10^{-4}$	$1.184 \cdot 10^{-4}$
256	$1.505 \cdot 10^{-4}$	$2.958 \cdot 10^{-5}$
512	$3.764 \cdot 10^{-5}$	$7.394 \cdot 10^{-6}$
1024	$9.412 \cdot 10^{-6}$	$1.848 \cdot 10^{-6}$

In diesem Fall konvergiert der maximale Fehler auch im Falle kubischer Splines nur mit h^2. Dies liegt daran, dass $f_2''(x)$ unstetig am Punkt $x = \pi/2$ ist (springt von 2 auf 0). Die dritte Ableitung ist dann unstetig an der Stelle $x = \pi/2$. □

Bemerkung 21.8. Für die Interpolation mit Splinefunktionen merken wir uns:

- Je höher der Grad des Splineraumes umso schneller konvergiert das Verfahren. Im Optimalfall erhält man $O(h^{k+1})$ Konvergenz für Grad k.
- Dies gilt allerdings nur dann, wenn die zu interpolierende Funktion genügend oft differenzierbar ist. Ist dies nicht der Fall so lohnt also auch die Verwendung von hohem Grad nicht. □

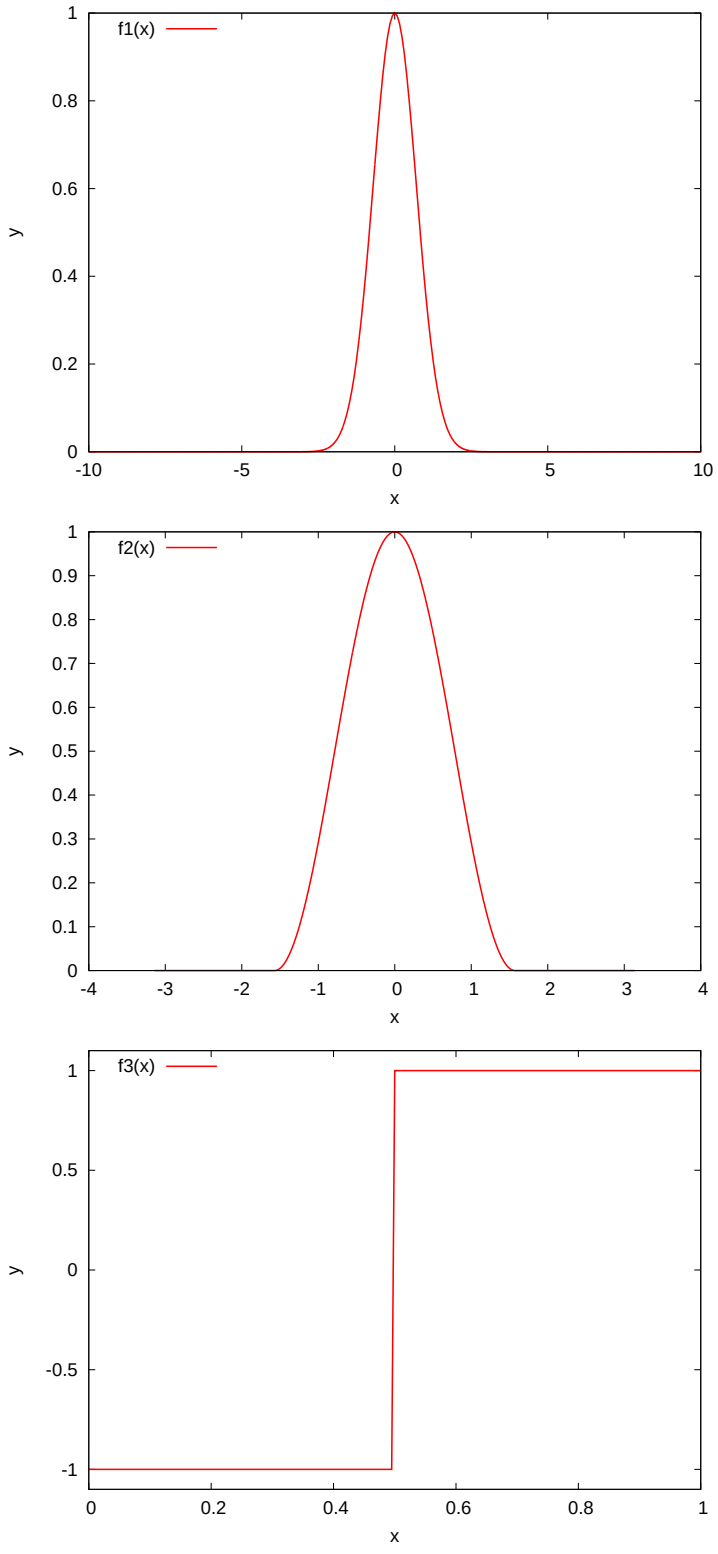


Abbildung 21.2: Die Funktionen $f_1(x)$ (links), $f_2(x)$ (Mitte) und $f_3(x)$ (rechts)

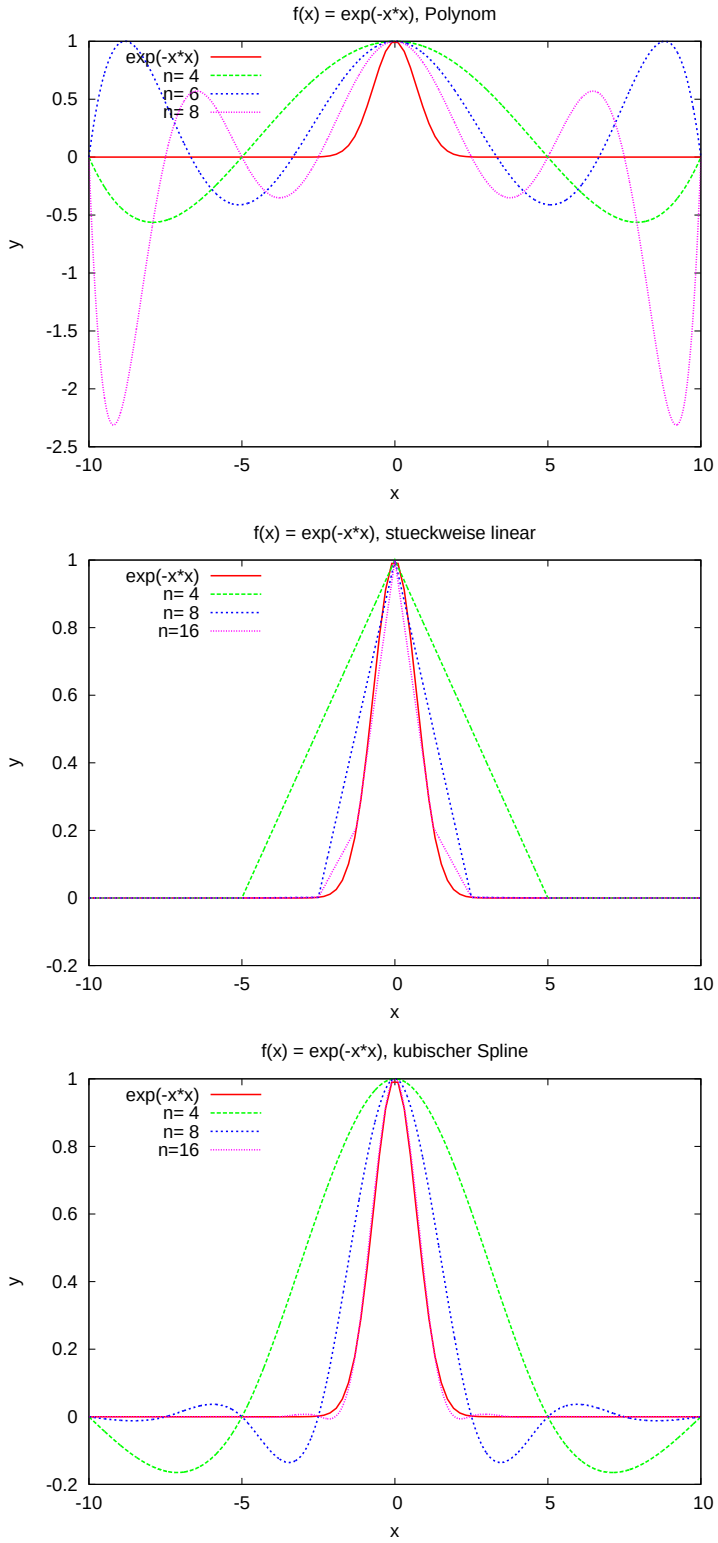


Abbildung 21.3: Interpolation der Funktion $f_1(x)$ mit Lagrange-Polynomen, stückweise linearen Funktionen und kubischen Splines.

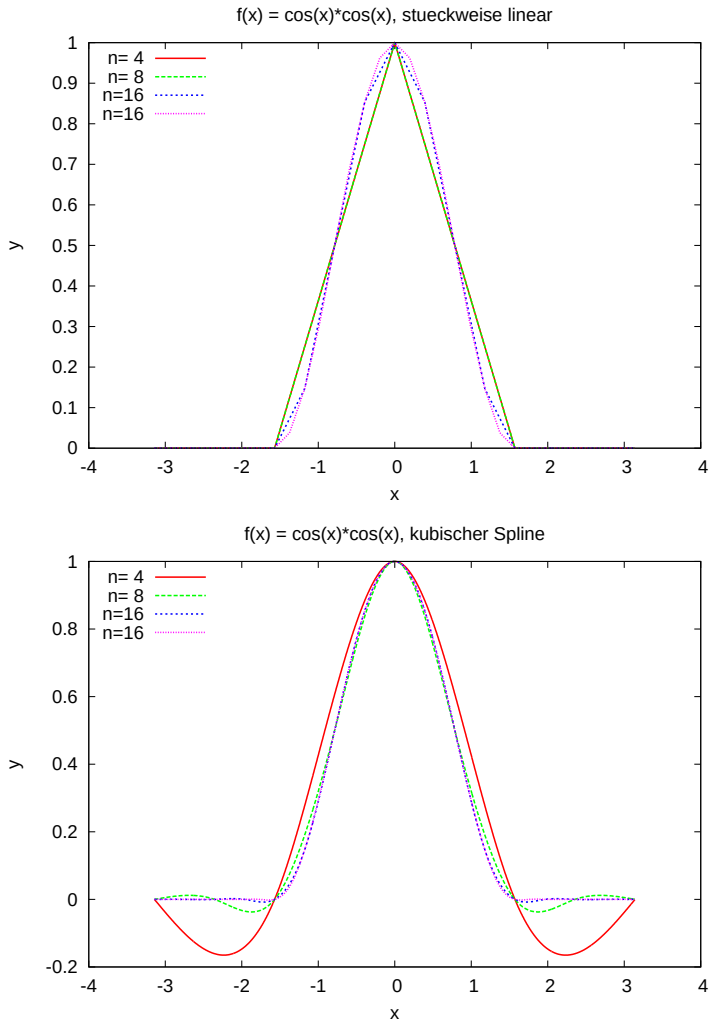


Abbildung 21.4: Interpolation der Funktion $f_2(x)$ mit Lagrange-Polynomen, stückweise linearen Funktionen und kubischen Splines.

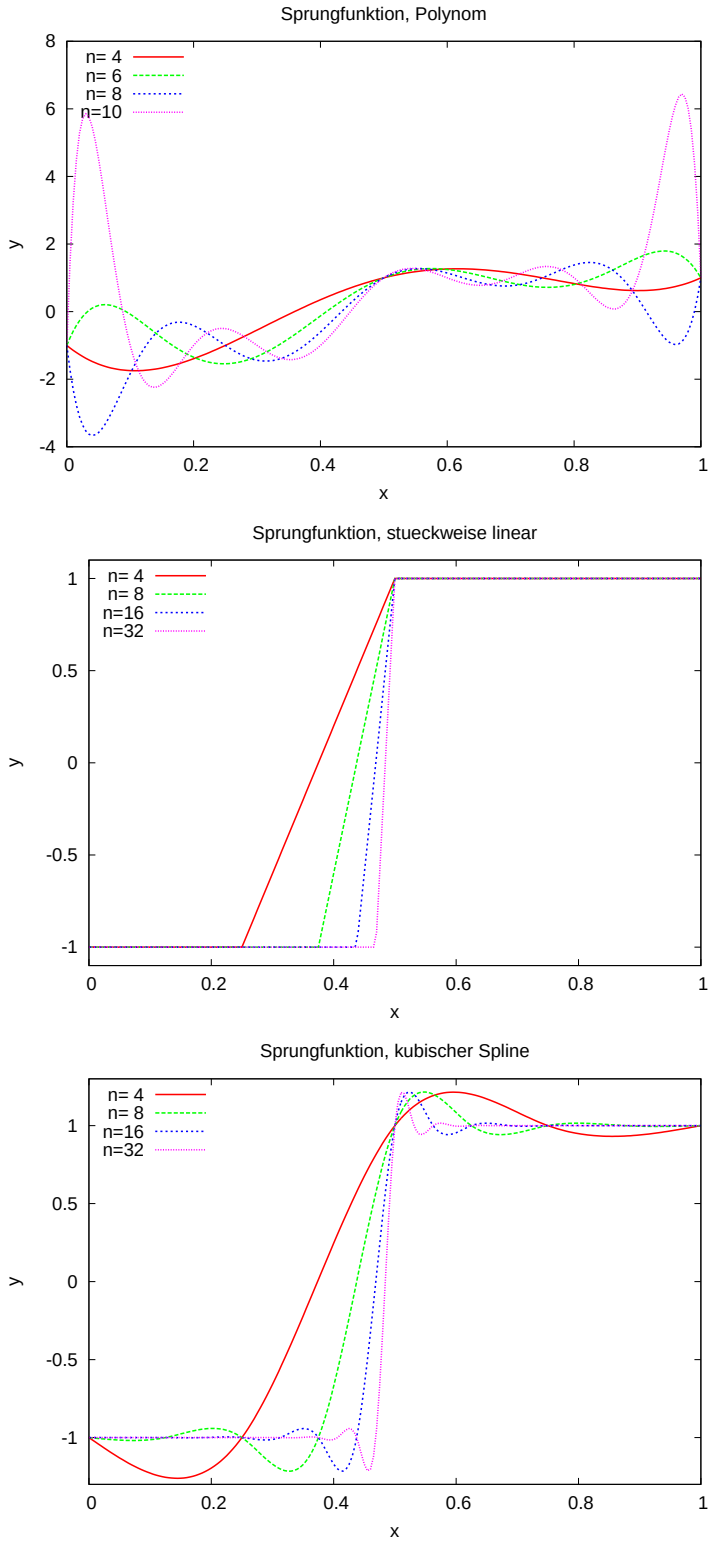


Abbildung 21.5: Interpolation der Funktion $f_3(x)$ mit Lagrange-Polynomen, stückweise linearen Funktionen und kubischen Splines.

Vorlesung 22

Kurvendarstellung mit Bernstein-Polynomen

Kurven sind Abbildungen $u(t) : [a, b] \to \mathbb{R}^d$ für $d = 2, 3$. Für die Repräsentation von Kurven können für jede Komponente (stückweise) Polynome, also die bisher behandelten Techniken, eingesetzt werden. In Zeichenprogrammen bzw. zur Darstellung von Fonts haben sich jedoch Bernsteinpolynome bewährt, die wir in dieser Vorlesung kennenlernen. Bernsteinpolynome interpolieren jedoch nicht sondern approximieren.

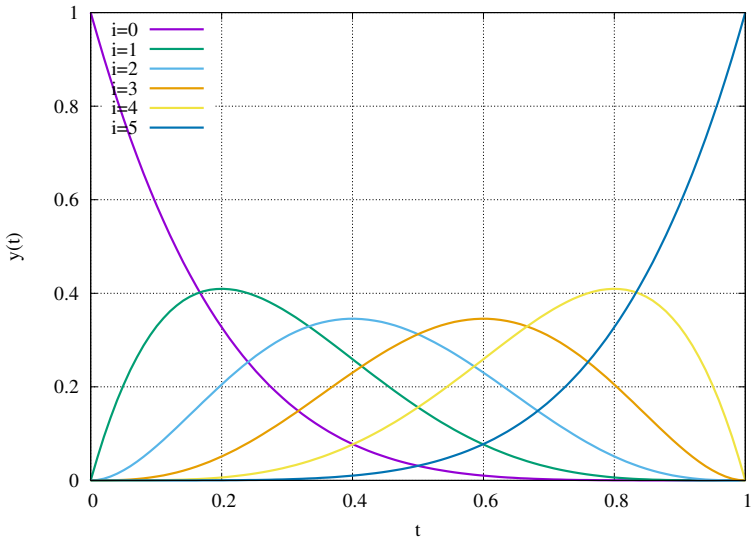
22.1 Bernsteinpolynome

Definition 22.1 (Bernstein-Polynome)**.** Die $n + 1$ Polynome

$$B_i^{(n)}(t) = \binom{n}{i}(1-t)^{n-i}\, t^i, \qquad i = 0, \ldots, n,$$

vom Grad n heißen Bernstein-Polynome auf $[0, 1]$. □

Hier sind die Bernsteinpolynome für $n = 5$ dargestellt:



Mittels der Transformation $\varphi : [a, b] \to [0, 1],\ \varphi(u) = \frac{u-a}{b-a}$, definiert man die Bernstein-Polynome auf einem allgemeinen Intervall $[a, b]$:

$$\begin{aligned} B_{i,[a,b]}^{(n)} &= B_i^{(n)}(\varphi(u)) \\ &= \binom{n}{i}\left(1 - \frac{u-a}{b-a}\right)^{n-i}\left(\frac{u-a}{b-a}\right)^i \\ &= \binom{n}{i}\frac{1}{(b-a)^n}(b-u)^{n-i}(n-a)^i. \end{aligned}$$

Satz 22.2 (Eigenschaften der Bernstein-Polynome)**.** Wir fassen einige wichtige Eigenschaften der Bernsteinpolynome zusammen.

1) Die Bernsteinpolynome bilden eine Partition der Eins:

$$\sum_{i=0}^{n} B_i^{(n)}(t) = 1.$$

2) $t = 0$ ist i-fache Nullstelle von $B_i^{(n)}$ und $t = 1$ ist $n - i$-fache Nullstelle von $B_i^{(n)}$ mithin sind $t = 0, t = 1$ die einzigen Nullstellen der Bernsteinpolynome und unterscheiden sich nur durch ihre Vielfachheiten.

3) Es gilt folgende Symmetrieeigenschaft: $B_i^{(n)}(t) = B_{n-i}^{(n)}(1-t)$

4) Es gilt folgende Positivitätseigenschaft:

$$0 \leq B_i^{(n)}(t) \leq 1 \quad (0 \leq t \leq 1), \qquad B_i^{(n)}(t) > 0 \quad (0 < t < 1).$$

5) $B_i^{(n)}$ hat in $[0, 1]$ genau ein Maximum im Punkt $\frac{i}{n}$.

6) Die $\left\{B_i^{(n)} : 0 \leq i < n\right\}$ sind linear unabhängig und bilden eine Basis von P_n.

7) Die Bernsteinpolynome erlauben folgende rekursive Darstellung über den Grad n:

$$B_i^{(n)}(t) = \begin{cases} (1-t)B_0^{(n-1)}(t) & i = 0 \\ tB_{i-1}^{(n-1)}(t) + (1-t)B_i^{(n-1)}(t) & 0 < i < n \\ tB_{n-1}^{(n-1)}(t) & i = n \end{cases}$$

8) Für die erste Ableitung gilt entsprechend die rekursive Darstellung:

$$\frac{\mathrm{d}}{\mathrm{d}t} B_i^{(n)}(t) = \begin{cases} -nB_0^{(n-1)}(t) & i = 0 \\ n\left[B_{i-1}^{(n-1)}(t) - B_i^{(n-1)}(t)\right] & 0 < i < n \\ nB_{n-1}^{(n-1)}(t) & i = n \end{cases}$$

Beweis. 1) Mit dem bionomischen Lehrsatz ergibt sich

$$1 = ((1-t)+t)^n = \sum_{i=0}^{n} \underbrace{\binom{n}{i}(1-t)^{n-i}\, t^i}_{=B_i^{(n)}(t)}.$$

2) Ergibt sich aus der Definition und dem Fundamentalsatz der Algebra: Ein Polynom vom Grad n hat genau n Nullstellen.

3) Durch Einsetzen in die Definition.

4) Positivität. Ergibt sich unmittelbar aus der Tatsache, dass $0 \leq t, 1-t \leq 1$ für $t \in [0, 1]$ bzw. $0 < t < 1$ für $t \in (0, 1)$.

5) Wir rechnen für die Ableitungen nach:

$$\begin{aligned}\frac{d}{dt}B_i^{(n)}(t) &= \binom{n}{i}\left[-(n-i)(1-t)^{n-i-1}\,t^i + i(1-t)^{n-i}\,t^{i-1}\right]\\ &= \binom{n}{i}(1-t)^{n-i-1}\,t^{i-1}\Big[\underbrace{(1-t)\,i-(n-i)\,t}_{i-it-nt+it}\Big]\\ &= \binom{n}{i}\underbrace{(1-t)^{n-i-1}}_{\substack{n-i-1\text{-fache}\\ \text{Nullstelle}\\ \text{bei } t=1}}\ \underbrace{t^{i-1}}_{\substack{i-1\text{-fache}\\ \text{Nullstelle}\\ \text{bei } t=0}}\ \underbrace{(i-nt)}_{\substack{\text{Nullstelle}\\ \text{bei } t=i/n}}.\end{aligned}$$

Wir zählen $n-1$ Nullstellen. Da die Ableitung ein Polynom vom Grad $n-1$ ist, sind dies alle Nullstellen. Für $0 < i < n$ hat $B_i^{(n)}$ ist $t = i/n$ einfache Nullstelle. Alle anderen Nullstellen sind bei $t = 0$ und $t = 1$ und dazwischen ist $B_i^{(n)}(t) > 0$, also muss ein Maximum vorliegen. Für $i = 0$, bzw. $i = n$ stimmt die Formel ebenfalls aber dann tritt das Maximum am Rand des Intervalles $[0, 1]$ auf (und ist nicht an einer Nullstelle der Ableitung).

6) Lineare Unabhängigkeit liegt vor wenn:

$$\sum_{i=0}^{n}\alpha_i B_i^{(n)}(t) = 0\ \forall\, t \quad \Rightarrow \quad \alpha_i = 0 \text{ für } i = 0,\dots,n.$$

Dazu betrachte die j-te Ableitung:

$$\frac{\mathrm{d}^j}{\mathrm{d}t^j}\sum_{i=0}^{n}\alpha_i B_i^{(n)}(t) = \sum_{i=0}^{n}\alpha_i\frac{\mathrm{d}^j}{\mathrm{d}t^j}B_i^{(n)}(t) \overset{!}{=} 0\ \forall\, t\in\mathbb{R}$$

Für $j = 0$ (keine Ableitung) und $t = 0$ ist nur $B_0^{(n)}(0) \neq 0$, alle anderen $B_i^{(n)}$ haben dort eine Nullstelle $\Rightarrow \alpha_0 = 0$. Nun betrachte $j = 1$ an der Stelle $t = 0$. Wieder ist nur $\frac{\mathrm{d}}{\mathrm{d}t}B_1^{(n)}(0) \neq 0$, also $\alpha_1 = 0$. Dieses Argument kann man fortsetzen bis zur n-ten Ableitung von $B_n^{(n)}$ an der Stelle $t = 0$. Dort gilt $\frac{\mathrm{d}^n}{\mathrm{d}t^n}B_n^{(n)} = \frac{\mathrm{d}^n}{\mathrm{d}t^n}t^n = n!$, also muss $\alpha_n = 0$ sein.

7) Die Randfälle $i = \{0, n\}$ sieht man durch einsetzen. Für $0 < i < n$ rechnet man nach

$$\begin{aligned}&tB_{i-1}^{(n-1)}(t) + (1-t)B_i^{(n-1)}(t)\\ &\quad= t\binom{n-1}{i-1}(1-t)^{n-1-i-1}t^{(i-1)} + (1-t)\binom{n-1}{i}(1-t)^{n-1-i}t^i\\ &\quad= \binom{n-1}{i-1}(1-t)^{n-i}t^i + \binom{n-1}{i}(1-t)^{n-1}t^i\\ &\quad= \Big[\underbrace{\binom{n-1}{i-1}+\binom{n-1}{i}}_{=\binom{n}{i}}\Big](1-t)^{n-i}t^i\end{aligned}$$

8) Die Randfälle $i = \{0, n\}$ folgen direkt aus der Formel für die Ableitung. Die Fälle $0 < i < n$ rechnet man durch einsetzen nach.

□

Es gilt der folgende bemerkenswerte Satz.

Satz 22.3. Für eine auf dem Intervall $[0,1]$ stetige Funktion $f\colon [0,1] \to \mathbb{R}$ konvergiert die Folge der Funktionen

$$f_n(t) = \sum_{i=0}^{n} f\left(\frac{i}{n}\right) B_i^{(n)}(t)$$

gleichmäßig gegen f.

Beweis. Folgt aus dem Beweis des Approximationssatzes von Weierstraß mit Bernsteinpolynomen, [Estep, 2005, S. 559]. □

Bei den $f_n(t)$ handelt es sich offensichtlich um Polynome vom Grad n. Die starken Ausschläge der Lagrange-Polynome auf äquidistanten Stützstellen treten hier nicht auf. Allerdings ist $f_n(t)$ auch kein Interpolationspolynom.

22.2 Bezier-Kurven

Definition 22.4 (Bezier-Polynom)**.** Ein Polynom $p(t)$ vom Grad n in der Darstellung

$$p(t) = \sum_{i=0}^{n} \beta_i B_i^{(n)}(t), \quad \beta_i \in \mathbb{R}$$

heißt Bezier-Polynom oder Bezier-Darstellung von p. □

Die Definition der Bezier-Polynome lässt sich auf Koeffizienten im $\mathbb{R}^d$ erweitern:

Definition 22.5 (Bezier-Kurve)**.** Für gegebene Punkte $b_0, \ldots, b_n \in \mathbb{R}^d$ heißt das vektorwertige Polynom

$$x(t) = \sum_{i=0}^{n} b_i B_i^{(n)}(t)$$

Bezier-Kurve. □

Definition 22.6 (Konvexe Hülle einer Punktmenge)**.** Gegeben seien $n+1$ Punkte $b_0, \ldots, b_n \in \mathbb{R}^d$. Für reelle Zahlen

$$0 \le q_0, \ldots, q_n \le 1, \qquad \sum_{i=0}^{n} q_i = 1,$$

nennt man $\sum_{i=0}^{n} q_i b_i$ eine Konvexkombination. Die Menge aller Konvexkombinationen der Punkte $b_0, \ldots, b_n$ nennt man ihre konvexe Hülle. □

Die konvexe Hülle zweier Punkte b_i, b_{i+1} ist die Verbindungsgerade von b_i nach b_{i+1}. Der Rand der konvexen Hülle der Punkte $b_0, \ldots, b_n$ bildet einen Polytop im $\mathbb{R}^d$. Im Fall $d = 2$ nennt man den Rand der konvexen Hülle in unserem Zusammenhang auch Bezier-Polygon.

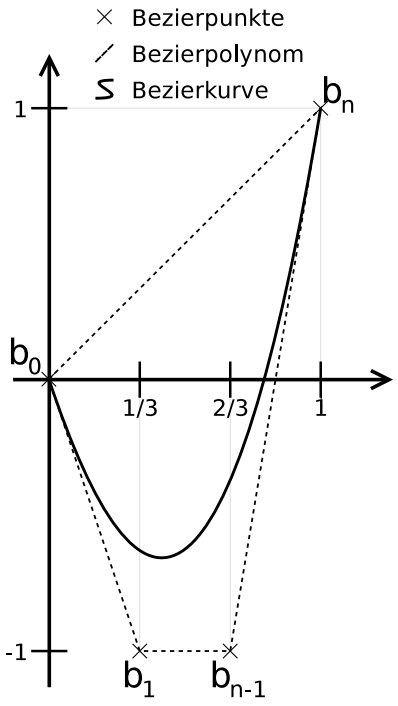


Abbildung 22.1: Bezier-Kurve, konvexe Hülle und Bezier-Polygon.

Beispiel 22.7. Wir betrachten die folgenden Punkte im $\mathbb{R}^2$:

$$b_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, b_1 = \begin{pmatrix} \frac{1}{3} \\ -1 \end{pmatrix}, b_2 = \begin{pmatrix} \frac{2}{3} \\ -1 \end{pmatrix}, b_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Die zugehörige Bezier-Kurve ermittelt man als:

$$x(t) = \begin{pmatrix} t \\ t^3 + 3t^2 - 3t \end{pmatrix}$$

Abbildung 22.1 zeigt diese Bezier-Kurve und das zugehörige Bezier-Polygon bzw. die konvexe Hülle der Punkte.

In diesem Fall ist wegen $x_1(t) = t$ das Polynom $p(t) = x_2(t)$ das Bezier-Polynom zu den Koeffizienten $\beta_0 = 0$, $\beta_1 = -1$, $\beta_2 = -1$ und $\beta_3 = 1$. □

Satz 22.8. Gegeben seien $n+1$ Punkte $b_0, \ldots, b_n \in \mathbb{R}^d$. Dann hat die Bezier-Kurve $x(t) = \sum_{i=0}^{n} b_i B_i^{(n)}(t)$ folgende Eigenschaften:

i) $x(t)$ liegt in der konvexen Hülle der Punkte $b_0, \ldots, b_n$.

ii) Es ist $x(0) = b_0$ und $x(1) = b_n$.

iii) Die Ableitung $x'(t)$, d.h. die Tangente an die Kurve, hat in den Endpunkte die Werte

$$x'(0) = n(b_1 - b_0), \qquad x'(1) = n(b_n - b_{n-1}).$$

Beweis. Wir nutzen die Eigenschaften der Bernstein-Polynome.

i) Wegen $0 \leq B_i^{(n)}(t) \leq 1$ und $\sum_{i=0}^{n} B_i^{(n)}(t) = 1$ ist $x(t)$ eine Konvexkombination und damit ist $x(t)$ ein Element der konvexen Hülle der Punkte.

ii) Gilt wegen $B_0^{(n)}(0) = 1$ und $B_i^{(n)}(0) = 0$, $i > 0$ bzw. $B_n^{(n)}(1) = 1$ und $B_i^{(n)}(1) = 0$, $i < n$.

iii) Hier nutzt man die rekursive Darstellung der Ableitung der Bernsteinpolynome:

$$\begin{aligned} x'(t) &= \frac{\mathrm{d}}{\mathrm{d}t} \sum_{i=0}^{n} b_i B_i^{(n)}(t) = \sum_{i=0}^{n} b_i \frac{\mathrm{d}}{\mathrm{d}t} B_i^{(n)}(t) \\ &= b_0 \left[-n B_0^{(n-1)}(t) \right] + \sum_{i=1}^{n-1} b_i n \left[B_{i-1}^{(n-1)}(t) - B_i^{(n-1)}(t) \right] + b_n \left[n B_{n-1}^{(n-1)}(t) \right]. \end{aligned}$$

Für $t = 0$ ist nur $B_0^{(n-1)}(0) = 1$, alle anderen sind 0, also

$$x'(0) = -nb_0 + nb_1 = n(b_1 - b_0).$$

Ebenso ist für $t = 1$ nur $B_{n-1}^{(n-1)}(0) = 1$ und alle anderen sind 0, also

$$x'(1) = -nb_{n-1} + nb_n = n(b_n - b_{n-1}).$$

□

Die präzise Kontrolle über die Endpunkte und die Ableitungen an den Endpunkten wird gerne benutzt um glatte Kurven interaktiv zu zeichnen, z.B. in OfficeDraw und anderen Grafikprogrammen.

22.3 Algorithmus von de Casteljau

Eine effiziente und numerisch stabile Auswertung einer Bezier-Kurve gelingt mit dem Algorithmus von de Casteljau. Mittels der Rekursionsformel erhalten wir

$$\begin{aligned} x(t) &= \sum_{i=0}^{n} b_i B_i^{(n)}(t) \\ &= b_0(1-t)B_0^{(n-1)}(t) + \sum_{i=1}^{n-1} b_i \left[tB_{i-1}^{(n-1)}(t) + (1-t)B_i^{(n-1)}(t) \right] + b_n t B_{n-1}^{(n-1)}(t) \\ &= \sum_{i=0}^{n-1} \left[(1-t)b_i + tb_{i+1} \right] B_i^{(n-1)}(t) = \sum_{i=0}^{n-1} b_i^{(1)} B_i^{(n-1)}(t). \end{aligned}$$

Mit der rekursiven Definition

$$\begin{aligned} b_i^{(0)} &= b_i && 0 \le i \le n, \\ b_i^{(k)} &= (1-t)b_i^{(k-1)} + tb_{i+1}^{(k-1)} && 0 < k \le n,\ 0 \le i \le n-k, \end{aligned}$$

gilt dann

$$x(t) = b_0^n$$

und es gibt sich ein Auswertungsschema genau wie im Neville-Schema:

$k=0$	$k=1$	$k=2$	$k=3$	$k=4$
$b_0 = b_0^{(0)}$	$b_0^{(1)}$	$b_0^{(2)}$	$b_0^{(3)}$	$b_0^{(4)} = x(t)$
$b_1 = b_1^{(0)}$	$b_1^{(1)}$	$b_1^{(2)}$	$b_1^{(3)}$	
$b_2 = b_2^{(0)}$	$b_2^{(1)}$	$b_2^{(2)}$		
$b_3 = b_3^{(0)}$	$b_3^{(1)}$			
$b_4 = b_4^{(0)}$				

Vorlesung 23

Approximation in Skalarprodukträumen

In dieser Vorlesung betrachten wir die Approximationsaufgabe im Detail. Es zeigt sich, dass dafür das Skalarproduktes sehr wichtig ist.

23.1 Gauß-Approximation

Definition 23.1. Ein Vektorraum von Funktionen über $\mathbb{R}$ oder $\mathbb{C}$, auf dem ein Skalarprodukt $\langle \cdot, \cdot \rangle$ definiert ist, heißt Skalarproduktraum oder auch Prähilbertraum. □

Jeder Prähilbertraum ist auch normiert, da durch

$$\|f\| = \sqrt{\langle f, f \rangle}$$

stets eine Norm definiert wird. Ein Prähilbertraum wird zum Hilbertraum wenn er zusätzlich noch vollständig ist.

Beispiel 23.2. Wir beschränken uns auf reelle Vektorräume von Funktionen über dem Intervall $[a, b]$ und führen das Skalarprodukt

$$\langle f, g \rangle = \int_a^b f(x)g(x)\,dx$$

ein.

1) Der Raum der stetigen Funktionen $C([a, b])$ mit dem Skalarprodukt oben bildet einen Prähilbertraum.

2) Der Raum der quadratintegrierbaren Funktionen $L^2((a, b))$, d.h. Funktionen f für die $\int_a^b |f(x)|^2 dx = \|f\|^2 < \infty$ gilt, ist ebenfalls ein Prähilbertraum. In diesem Fall ist $\int_a^b dx$ das sogenannte „Lebesgue-Integral“ (siehe Vorlesung Funktionalanalysis).

   Im Gegensatz zu $C([a, b])$ ist $L^2[a, b]$ vollständig bezüglich der induzierten Norm $\|.\|$ und damit ein Hilbertraum. $L^2([a, b])$ ist das Analogon des $\mathbb{R}^n$ für Funktionenräume.

3) Gegeben sei eine endliche Menge von Funktionen $\Psi = \{\psi_1, \ldots, \psi_N\}$ mit $\psi_i \in L^2([a, b])$ oder $\psi_i \in C([a, b])$ dann ist

$$S = \text{span}\Psi = \left\{ f : f = \sum_{i=1}^{N} c_i \psi_i \right\}$$

   ein endlich-dimensionaler Prähilbertraum. Da S ismorph zum $\mathbb{R}^N$ ist, ist S auch ein Hilbertraum. □

Im folgenden werden wir uns auf endlich-dimensionale Teilräume S eines Prähilbertraumes H beschränken und folgende Approximationsaufgabe betrachten:
Zu $f \in H$ finde $g \in S$, sodass

$$\|f - g\| \to \min \tag{23.1}$$

wobei $\|f\| = \sqrt{\langle f, f\rangle}$ stets die durch das Skalarprodukt induzierte Norm ist.

Satz 23.3 (Allgemeine Gauß-Approximation)**.** Sei H Prähilbertraum und $S \subset H$ ein endlichdimensionaler Teilraum. Es gibt genau eine Funktion $g \in S \subset H$ die $\|f - g\|$ für gegebenes $f \in H$ minimiert und g erfüllt die Bedingung

$$\langle g, \varphi\rangle = \langle f, \varphi\rangle \quad \forall\, \varphi \in S. \tag{23.2}$$

Beweis. Wir zeigen dies in mehreren Schritten.

1. Zuerst zeigen wir: Aus g minimiert $\|f - g\|$ folgt $\langle f - g, \varphi\rangle = 0$ für alle $\varphi \in S$. Für jede sogenannte Testfunktion φ definieren wir die Funktion

   $$F_\varphi(t) \coloneqq \|f - (g + t\varphi)\|^2, \qquad t \in \mathbb{R}.$$

   Da nach Voraussetzung $\|f - g\|$ minimal ist, muss die Funktion $F_\varphi(t)$ für jedes $\varphi \in S$ ihr Minimum bei $t = 0$ annehmen.

   Notwendige Bedingung für ein Minimum ist dass die Ableitung bei $t = 0$ verschwindet:

   $$\begin{aligned}
   \frac{\mathrm{d}}{\mathrm{d}t} F_\varphi(t)\big|_{t=0} &= \frac{\mathrm{d}}{\mathrm{d}t}\langle f - g - t\varphi, f - g - t\varphi\rangle\big|_{t=0} \\
   &= \frac{\mathrm{d}}{\mathrm{d}t}\left[\langle f - g, f - g\rangle - 2t\langle f - g, \varphi\rangle + t^2\langle \varphi, \varphi\rangle\right]\big|_{t=0} \\
   &= [-2\langle f - g, \varphi\rangle + 2t\langle \varphi, \varphi\rangle]|_{t=0} \\
   &= -2\langle f - g, \varphi\rangle \overset{!}{=} 0
   \end{aligned}$$

   Also $\langle f - g, \varphi\rangle = 0$ für alle $\varphi \in S$. Geometrisch bedeuted dies: Der „Fehler“ $f - g$ ist orthogonal zu allen $\varphi \in S$.

2. Nun die Gegenrichtung. Wir nehmen an $\langle f - g, \varphi\rangle = 0$ für alle $\varphi \in S$. Zu zeigen ist nun, dass g die Funktion $\|f - g\|$ minimiert. Für ein beliebiges $g' \in S$ gilt:

   $$\begin{aligned}
   \|f - g'\|^2 = \|f - g + \underbrace{g - g'}_{=:\varphi}\|^2 = \|f - g + \varphi\|^2 &= \langle f - g + \varphi, f - g + \varphi\rangle \\
   &= \langle f - g, f - g\rangle + 2\underbrace{\langle f - g, \varphi\rangle}_{=0 \text{ nach Vor.}} + \langle \varphi, \varphi\rangle \\
   &= \|f - g\|^2 + \|\varphi\|^2 \geq \|f - g\|^2 \text{ also ist } g \text{ Minimum.}
   \end{aligned}$$

   Damit ist die Äquivalenz gezeigt:

   $$\|f - g\| \to \min \iff \langle f - g, \varphi\rangle = 0 \iff \langle g, \varphi\rangle = \langle f, \varphi\rangle \quad \forall\, \varphi \in S$$

3. Eindeutigkeit des Minimums:

   Angenommen es gäbe zwei Minima g_1 und g_2 mit $g_1 \neq g_2$. Dann ist

$$\begin{aligned}\|f-g_1\|^2 &= \|f-g_2+\underbrace{g_2-g_1}_{=:\varphi}\|^2 \\ &= \|f-g_2\|^2 + \underbrace{\|g_2-g_1\|^2}_{>0 \text{ da } g_1\neq g_2} > \|f-g_2\| \;\lightning \text{ zu } g_1 \text{ Minimum}\end{aligned}$$

4. Existenz des Minimums: Dies zeigen wir konstruktiv und erhalten damit auch einen Algorithmus zur Berechnung von g.

   Da S endlich-dimensional ist, gibt es eine Basis $\Psi = \{\psi_1, \ldots, \psi_N\}$ mit $N = \dim S$. Das gesuchte Element g hat die Darstellung

$$g = \sum_{j=1}^{N} c_j \psi_j.$$

   Einsetzen der Basisdarstellung in (23.2) liefert

$$\begin{aligned}&\langle g, \varphi\rangle = \langle f, \varphi\rangle \quad \forall\, \varphi \in S \\ &\iff \left\langle \sum_{j=1}^{N} c_j \psi_, \psi_i \right\rangle = \langle f, \psi_i\rangle \quad i = 1, \ldots, N \\ &\iff \sum_{j=1}^{N} c_j \langle \psi_j, \psi_i\rangle = \langle f, \psi_i\rangle \quad i = 1, \ldots, N \\ &\iff Ac = b \\ &\text{mit } (A)_{ij} = \langle \psi_j, \psi_i\rangle \qquad \text{(Massenmatrix oder Gramsche Matrix)} \\ &\quad (b)_i = \langle f, \psi_i\rangle.\end{aligned}$$

   Man erhält g dargestellt in der Basis Ψ durch lösen eines linearen Gleichungssystemes. Nun ist zu zeigen, dass dieses Gleichungssystem regulär ist. Dies ist der Fall, da A symmetrisch und positiv definit ist, denn

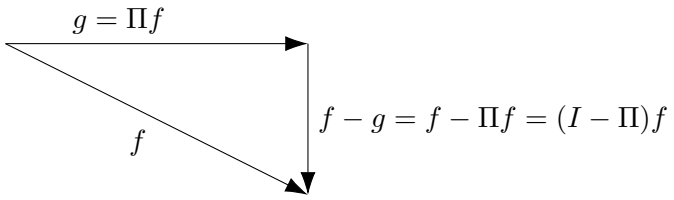
   a) $(A)_{ij} = \langle \psi_j, \psi_i\rangle = \langle \psi_i, \psi_j\rangle = (A)_{ji}$,

   b) und für beliebiges $0 \neq x \in \mathbb{R}^N$ gilt

$$\begin{aligned}x^T A x &= \sum_{i=1}^{N} x_i \left(\sum_{j=1}^{N} (A)_{ij}\; x_j \right) = \sum_{i=1}^{N} x_i \left(\sum_{j=1}^{N} \langle \psi_j, \psi_i\rangle\; x_j \right) \\ &= \sum_{i=1}^{N} x_i \left\langle \sum_{j=1}^{N} x_j \psi_j, \psi_i \right\rangle = \left\langle \underbrace{\sum_{j=1}^{N} x_j \psi_j}_{=:u\in S}, \underbrace{\sum_{i=1}^{N} x_i \psi_i}_{=:u} \right\rangle \\ &= \langle u, u\rangle > 0 \text{ da } u \neq 0 \iff x \neq 0.\end{aligned}$$

Damit ist gezeigt, dass g aus (23.2) die eindeutig bestimmte Bestapproximation von f bezüglich der Norm $\|.\|$ in H ist. □

Die Bedingung (23.2) bedeutet, dass der Approximationsfehler $f - g$ senkrecht zu allen Elementen $\varphi \in S$ und somit auch zu g ist. Geometrisch können wir das so veranschaulichen:



Die Abbildung $\Pi : H \to S$ wird im folgenden Satz näher charakterisiert.

Satz 23.4. Die mittels (23.2) definierte Abbildung $\Pi : H \to S$ mit

$$\langle \Pi f, \varphi \rangle = \langle f, \varphi \rangle, \qquad \forall \varphi \in S$$

ist eine orthogonale Projektion und es gilt

i) $\Pi^2 = \Pi$,

ii) $\langle \Pi f, (I - \Pi) f \rangle = 0$ für jedes $f \in H$.

iii) $\|f\|^2 = \|\Pi f\|^2 + \|(I - \Pi) f\|^2$ (Verallgemeinerung des Satz von Pythagoras) und damit $\|\Pi f\| \leq \|f\|$ und $\|(I - \Pi) f\| \leq \|f\|$ (Stabilität der Projektion).

Beweis. i) Wegen $\Pi f \in S$ wird $\langle g, \varphi \rangle = \langle \Pi f, \varphi \rangle \ \forall \phi \in S$ durch $g = \Pi f$ erfüllt und es gilt $\|f - g\| = 0$. Nach Satz 23.3 ist g eindeutig bestimmt. Somit ist $\Pi(\Pi f) = \Pi f$.

ii) Wegen $\Pi f \in S$ gilt $\langle \Pi f, \Pi f \rangle = \langle f, \Pi f \rangle$ und damit

$$\langle \Pi f, (I - \Pi) f \rangle = \langle \Pi f, f \rangle - \langle \Pi f, \Pi f \rangle = \langle \Pi f, f \rangle - \langle f, \Pi f \rangle = 0.$$

iii) Schließlich

$$\begin{aligned} \|f\|^2 &= \|\Pi f + f - \Pi f\|^2 \\ &= \langle \Pi f + (I - \Pi) f, \Pi f + (I - \Pi) f \rangle \\ &= \langle \Pi f, \Pi f \rangle + 2 \langle \Pi f, (I - \Pi) f \rangle + \langle (I - \Pi) f, (I - \Pi) f \rangle \\ &= \|\Pi f\|^2 + \|(I - \Pi) f\|^2. \end{aligned}$$

Wegen $\|\Pi f\|^2 = \|f\|^2 - \|(I - \Pi) f\|^2 \leq \|f\|^2$ und analog für $\|(I - \Pi) f\|^2$. □

23.2 Gauß-Approximation mit Orthonormalbasen

Besonders einfach wird die Lösung der Approximationsaufgabe, wenn Ψ eine Orthonormalbasis ist, d.h. $\langle \psi_i, \psi_j \rangle = \delta_{ij}$. Denn dann gilt $A = I$, also

$$\sum_{j=1}^{N} c_j \underbrace{\langle \psi_j, \psi_i \rangle}_{=\delta_{ij}} = c_i = \langle f, \psi_i \rangle \quad i = 1, \ldots, N$$

und somit

$$g = \sum_{j=1}^{N} c_j \psi_j = \sum_{j=1}^{N} \langle f, \psi_j \rangle \ \psi_j \tag{23.3}$$

Satz 23.5 (Parseval-Identität)**.** Es sei $S \subset H$ ein Unterraum eines Prähilbertraumes und $\{\psi_i : 1 \leq i \leq N\}$ eine Orthonormalbasis von S. Dann gilt für jedes $v \in S$ die Gleichung

$$\|v\|^2 = \sum_{i=1}^{N} \langle v, \psi_i \rangle^2,$$

die auch Parseval-Identät heißt.

Beweis. Mit der Betrachtung von oben kann $v \in S$ mittels $v = \sum_{i=1}^{N} \langle v, \psi_i \rangle \ \psi_i$ in der Orthonormalbasis dargestellt werden. Damit gilt dann

$$\|v\|^2 = \left\langle \sum_{i=1}^{N} \langle v, \psi_i \rangle \ \psi_i, \sum_{j=1}^{N} \langle v, \psi_j \rangle \ \psi_j \right\rangle = \sum_{i=1}^{N} \sum_{j=1}^{N} \langle v, \psi_i \rangle \langle v, \psi_j \rangle \delta_{i,j} = \sum_{i=1}^{N} \langle v, \psi_i \rangle^2.$$

□

Beispiel 23.6 (Fourier-Reihe)**.** Für $N = 2m + 1$, $m \in \mathbb{N}$ ist

$$\Psi_F = \left\{ \frac{1}{\sqrt{2\pi}}, \frac{1}{\sqrt{\pi}} \cos x, \ldots, \frac{1}{\sqrt{\pi}} \cos(mx), \frac{1}{\sqrt{\pi}} \sin x, \ldots, \frac{1}{\sqrt{\pi}} \sin(mx) \right\}$$

eine Orthonormalbasis auf $[-\pi, \pi]$. Dies beweist man durch Nachrechnen unter Nutzung der Additionstheoreme

$$\begin{aligned} \cos x \cos y &= \frac{1}{2} \left[\cos(x + y) + \cos(x - y) \right] \\ \sin x \sin y &= \frac{1}{2} \left[\cos(x - y) - \cos(x + y) \right] \\ \sin x \cos y &= \frac{1}{2} \left[\sin(x + y) + \sin(x - y) \right] \end{aligned}$$

sowie Eigenschaften der Sinus- bzw. Cosinusfunktion an Vielfachen von π. Damit gilt dann

$$\begin{aligned} g(x) &= \frac{a_0}{2} + \sum_{k=1}^{m} \left(a_k \cos(kx) + b_k \sin(kx) \right), \\ a_k &= \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(kx) \, dx \quad k = 0, \ldots, m, \\ b_k &= \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(kx) \, dx \quad k = 1, \ldots, m. \end{aligned}$$

Für endlich viele Glieder $m \in \mathbb{N}$ ist dies die endliche Fourier-Reihe. für unendlich viele Glieder ($m = \infty$) einfach Fourier-Reihe. Diese konvergiert gegen ein Element aus $L^2([-\pi, \pi])$. □

Beispiel 23.7 (Legendre-Polynome)**.** Die Polynome

$$P_n(x) = \frac{1}{2^n \cdot n!} \frac{\mathrm{d}^n}{\mathrm{d}x^n} (x^2 - 1)^n, \qquad n \in \mathbb{N} \cup \{0\}$$

heißen Legendre-Polynome und es gilt

i) P_n ist ein Polynom vom Grad n sowie die

ii) Orthogonalitätseigenschaft

$$\int_{-1}^{1} P_m(x)P_n(x) = \begin{cases} 0 & m \neq n \\ \frac{2}{2n+1} & m = n \end{cases} \qquad m, n \in \mathbb{N} \cup \{0\}.$$

Mit entsprechender Normierung $P_n'(x) = \sqrt{\frac{2n+1}{2}} P_n(x)$ erhält man eine Menge orthogonaler Polynome bezüglich des Skalarproduktes $\int_{-1}^{1} uvdx$. Siehe [Schaback and Wendland, 2005, Kapitel 3]. Die ersten Legendre-Polynome lauten $P_0(x) = 1$, $P_1(x) = x$, $P_2(x) = \frac{1}{2}(3x^2 - 1)$, $P_3(x) = \frac{1}{2}(5x^3 - 3x)$. □

23.3 Fehlerkontrolle

Bisher ist $S \subset H$ fest gewählt, wir berechnen die Bestapproximation $g \in S$ und der Fehler $\|f - g\|$ wird akzeptiert. Nun betrachten wir die folgende Verfeinerung der Approximationsaufgabe:
Finde $S \subset H$ mit dim$S$ möglichst klein, so dass

$$\|f - g\| \leq \text{TOL} \cdot \|f\|.$$

Dabei ist TOL eine vorgegebene Fehlerschranke bzw. Genauigkeit.

Mittels Orthonormalbasen lässt sich der Fehler recht einfach messen. Sei $\{S_N\}$, $S_N \subset H$, $N \in \mathbb{N}$ eine Folge von Approximationsräumen mit dim$S_N = N$. Oft gilt sogar $S_N \subset S_{N+1}$.

Nun sei g_N die Bestapproximation von f in S_N und weiter sei Ψ_N eine Orthonormalbasis von S_N (oft gilt $\Psi_N \subset \Psi_{N+1}$). Dann gilt für den Fehler:

$$\begin{aligned}
0 \leq \|f - g_N\|^2 = \langle f - g_N, f - g_N \rangle &= \langle f, f \rangle - 2\langle f, g_N \rangle - \langle g_N, g_N \rangle \\
&\overset{\text{Basis einsetzen}}{=} \langle f, f \rangle - 2\langle f, \sum_{i=1}^{N} \underbrace{\langle f, \psi_i \rangle}_{=c_i} \psi_i \rangle + \langle \sum_{i=1}^{N} \langle f, \psi_i \rangle \psi_i, \sum_{j=1}^{N} \langle f, \psi_j \rangle \psi_j \rangle \\
&= \langle f, f \rangle - 2 \sum_{i=1}^{N} \langle f, \psi_i \rangle \langle f, \psi_i \rangle + \sum_{i=1}^{N} \sum_{j=1}^{N} \langle f, \psi_i \rangle \langle f, \psi_j \rangle \underbrace{\langle \psi_i, \psi_j \rangle}_{=\delta_{ij}} \\
&= \langle f, f \rangle - 2 \sum_{i=1}^{N} \langle f, \psi_i \rangle^2 + \sum_{i=1}^{N} \langle f, \psi_i \rangle^2 \\
&= \langle f, f \rangle - \sum_{i=1}^{N} \langle f, \psi_i \rangle^2.
\end{aligned}$$

Mit dieser expliziten Darstellung des Fehlers erhalten wir dann

$$\begin{aligned}
\|f - g_N\|^2 = \langle f, f \rangle - \sum_{i=1}^{N} \langle f, \psi_i \rangle^2 &\leq \underbrace{\text{TOL}^2 \cdot \langle f, f \rangle}_{\text{relative Toleranz}} \\
\iff \sum_{i=1}^{N} \langle f, \psi_i \rangle^2 &\geq \left(1 - \text{TOL}^2\right) \langle f, f \rangle. \qquad (23.4)
\end{aligned}$$

Bemerkung 23.8. 1. Hier nehmen wir an, dass $\langle f, f\rangle$ berechenbar ist (zumindest mit ausreichender Genauigkeit).

2. Falls $\Psi_N \subset \Psi_{N+1}$ git (sogenannte hierarchische Basis) sind einfach so viele Basisfunktionen hinzuzufügen bis (23.4) erreicht ist.

3. Für die Fourier-Reihe gilt $\Psi_N \subset \Psi_{N+1}$

4. Aus der obigen Fehlerdarstellung folgt bei $\Psi_N \subset \Psi_{N+1}$ unmittelbar

$$\begin{aligned}\|f - g_{N+1}\|^2 = \langle f, f\rangle - \sum_{i=1}^{N+1} \langle f, \psi_i\rangle^2 &= \underbrace{\langle f, f\rangle - \sum_{i=1}^{N} \langle f, \psi_i\rangle^2}_{\|f-g_N\|^2} - \underbrace{\langle f, \psi_{N+1}\rangle^2}_{\geq 0} \\ &\leq \|f - g_N\|^2\end{aligned}$$

Der Fehler kann also durch Hinzufügen von Basisfunktionen nicht zunehmen. □

Vorlesung 24

Adaptive Approximation mit Haar-Wavelets

24.1 Motivation

Wir knüpfen an an die Approximation von Funktionen und möchten diese nun möglichst effizient gestalten im Sinne von:

- Garantierte Fehlerkontrolle $\|f - g\| \leq \text{TOL} \cdot \|f\|$,
- und möglichst geringe Dimension (und damit Aufwand) des Approximationsraumes S.

Die zu approximierenden Funktionen variieren häufig nur lokal sehr stark, so variiert etwa die Funktion

$$f(x) = \frac{1}{10x^2 + \epsilon}$$

für betragsmäßig sehr kleine ϵ sehr stark in der Nähe von 0. Weiter weg von der Null ist die Funktion deutlich glatter und sollte gut mit wenig Aufwand approximiert werden können. In diesem Abschnitt lernen wir eine Methode kennen die dieses leistet.

Um diese Lokalität besser quantifizieren zu können, zunächst eine Definition.

Definition 24.1. Der Support (oder Träger) einer Funktion ist die kleinste abgeschlossene Teilmenge der Definitionsmenge D einer Funktion $f : D \to \mathbb{R}$, in der alle Punkte liegen, an denen die Funktion Werte ungleich Null annimmt:

$$\text{supp}(f) = \overline{\{x \in D : f(x) \neq 0\}}.$$

□

Für die (orthogonale) Fourier-Basis gilt zum Beispiel:

$$\text{supp}(\psi) = [-\pi, \pi] = D \text{ für } \psi \in \Psi_F.$$

Ebenso gilt für die Legendre-Polynome auf dem Intervall $D = [0, 1]$:

$$\text{supp}(P_n) = [-1, 1] = D \text{ für } n \in \mathbb{N} \cup \{0\}.$$

Man sagt die Basisfunktionen haben einen „globalen“ Träger. Variiert nun eine Funktion f nur an bestimmten Stellen „lokal“ sehr stark dann kann man dieses Verhalten mit globalen Basisfunktionen nicht gut abbilden. Besser wäre es eine Basis mit Lokaltätseigenschaften zu benutzen.

Wir untersuchen in diesem Kapitel insbesondere orthogonale Basisfunktionen mit lokalem Träger. Sei nun $S \subset L_2([a, b])$ mit $S = \text{span}\Psi_N$ und $\Psi_N = \{\psi_i : 1 \leq i \leq N\}$, dann wünschen wir uns folgende Eigenschaften von den ψ_i:

1. Orthonormabasis: $\langle \psi_i, \psi_j \rangle = \int_a^b \psi_i(x)\psi_j(x)\, dx = \delta_{ij}$.

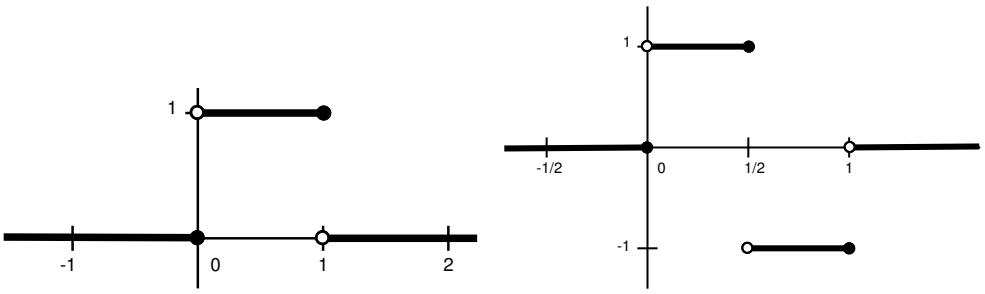


Abbildung 24.1: Die Indikatorfunktion $\chi(x)$ (links) und das daraus abgeleitete Haar-Wavelet der Stufe 0.

2. Hierarchische Basis: $\Psi_N \subset \Psi_{N+1}$. Dies erlaubt adaptive Verfeinerung.

3. Der Träger der Basisfunktionen schrumpft für $i \to \infty$.

Funktionen mit diesen Eigenschaften nennt man Waveletfunktionen. Wir betrachten hier das einfachste Wavelet, das sogennante Haar-Wavelet (benannt nach dem Mathematiker Alfréd Haar, 1909).

24.2 Haar-Wavelets

Sei $\chi(x)$ die charakteristische Funktion bzw. Indikatorfunktion des Intervalls $]0,1]$ (Abbildung 24.1, links):

$$\chi(x) = \begin{cases} 1 & 0 < x \leq 1 \\ 0 & \text{sonst} \end{cases}.$$

Definition 24.2 (Haar-Waveletbasis)**.** Wir definieren zunächst das sogenannte „mother wavelet“

$$\psi(x) = \chi(2x) - \chi\left(2\left(x - \frac{1}{2}\right)\right) = \begin{cases} 1 & 0 < x \leq 1/2 \\ -1 & 1/2 < x \leq 1 \\ 0 & \text{sonst} \end{cases}, \tag{24.1}$$

das in Abbildung 24.1 rechts abgebildet ist.

Die hierarchische Haar-Waveletbasis wird rekursiv definiert. Dabei werden die Basisfunktionen $\psi_i^k(x)$ mit einem zweidimensionalen Index (i,k) durchnummeriert wobei $k \in \mathbb{N} \cup \{0\}$ die Stufe und i mit $0 \leq i < 2^{\max(0,k-1)}$ die Nummer innerhalb einer Stufe k bezeichnet. Auf den Stufe 0 und 1 gibt es je eine Basisfunktion und auf der Stufe $k > 1$ gibt es 2^{k-1} Basisfunktionen. Die Haar-Waveletbasisfunktionen sind definiert als

$$\psi_0^0(x) = \chi(x) \qquad k = 0 \tag{24.2}$$

$$\psi_0^1(x) = \psi(x) \qquad k = 1 \tag{24.3}$$

$$\psi_i^k(x) = 2^{\frac{k-1}{2}} \cdot \psi\left(2^{k-1}\left(x - \frac{i}{2^{k-1}}\right)\right) \qquad k > 1, \quad 0 \leq i < 2^{k-1} \tag{24.4}$$

Der Vorfaktor ergibt sich aus der Normierung.

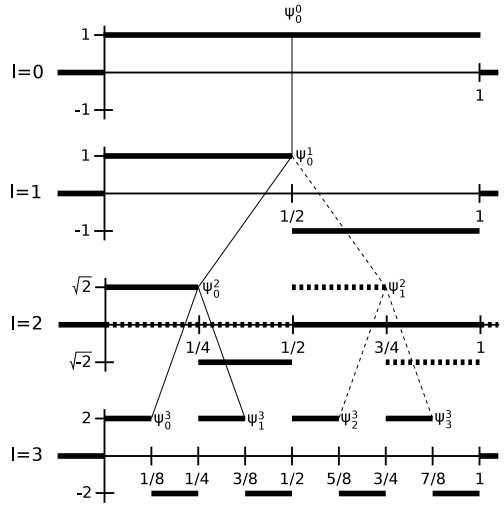


Abbildung 24.2: Baumstruktur von Haar-Wavelets bis zur Stufe 3.

Schließlich besteht die Haar-Waveletbasis der Stufe $k \in \mathbb{N} \cup \{0\}$ aus der Vereinigungsmenge aller Wavelet-Funktionen bis zur Stufe k:

$$\Psi^k = \left\{\psi_0^0\right\} \cup \bigcup_{l=1}^{k} \bigcup_{i=0}^{2^{l-1}-1} \left\{\psi_i^l\right\}. \tag{24.5}$$

Die Basisfunktionen bis zur Stufe $k = 3$ zeigt Abbildung 24.2. □

Lemma 24.3 (Eigenschaften der Haar-Wavelets)**.** Die Funktionen in Ψ^k bilden eine Orthonormalbasis und es gelten folgende Eigenschaften:

i) Auf der Stufe k gibt es $2^{\max(0,k-1)}$ Haar-Wavelets und damit $|\Psi^k| = 2^k$.

ii) $\operatorname{supp}\psi_0^0 = [0,1]$, $\operatorname{supp}\psi_0^1 = [0,1]$, $\operatorname{supp}\psi_i^k = [\frac{i}{2^{k-1}}, \frac{i+1}{2^{k-1}}]$. Das Wavelet ψ_i^k $(k > 0)$ ist ein Haar-Wavelet dessen Träger um den Faktor 2^{k-1} geschrumpf ist, das um $\frac{i}{2^{k-1}}$ nach rechts geschoben ist und dessen Wert um den Faktor $2^{\frac{k-1}{2}}$ skaliert ist.

iii) Die Wavelets bilden eine Baumstruktur bezüglich der Inklusion ihres Trägers. Für $k > 0$ gilt

$$\operatorname{supp}(\psi_i^k) = \operatorname{supp}(\psi_{2i}^{k+1}) \cup \operatorname{supp}(\psi_{2i+1}^{k+1}).$$

iv) Die Funktion ψ_i^k ist unstetig an den Stellen $\frac{i}{2^{k-1}} = \frac{2i}{2^k}, \frac{2i+1}{2^k}$ und $\frac{i+1}{2^{k-1}} = \frac{2i+2}{2^k}$. Sonst ist die Funktion stetig. Alle Unstetigkeitsstellen der Funktionen auf Stufe k liegen also an den Stellen $\frac{i}{2^k}$ für $0 \le i \le 2^k$.

v) Die Wavelets sind zueinander orthonormal, d.h. $\langle \psi_i^k, \psi_j^l \rangle = \delta_{ij} \cdot \delta_{kl}$.

vi) Die Basis ist hierarchisch, d.h. $\Psi^k \subset \Psi^{k+1}$.

Beweis. i) Auf den Stufen $k = 0, 1$ gibt es jeweils ein Haar-Wavelet. Auf der Stufe $k > 1$ gibt es nach Definition 2^{k-1} Haar-Wavelets. Somit ist $\dim \Psi^0 = 1$, $\dim \Psi^1 = 2 = 2^1$. Für $k > 1$ gilt demnach $\dim \Psi^k = \dim \Psi^{k-1} + 2^{k-1} = 2^{k-1} + 2^{k-1} = 2^k$.

ii) Nach Definition gilt $\operatorname{supp}\psi_0^0 = \operatorname{supp}\psi_0^1 = [0,1]$. Nun betrachte die rekursive Definitionsgleichung (24.4). $\left(x - \frac{i}{2^{k-1}}\right)$ bewirkt eine Verschiebung des Mother-Wavelet ψ um $\frac{i}{2^{k-1}}$ nach rechts und die Skalierung im Argument bewirkt eine Kontraktion des Trägers um den Faktor 2^{k-1}. Damit ist ψ_i^k auf einem Intervall der Länge $2^{-(k-1)}$ ungleich Null, welches bei $\frac{i}{2^{k-1}}$ beginnt. Somit ist $\operatorname{supp}\psi_i^k = [\frac{i}{2^{k-1}}, \frac{i+1}{2^{k-1}}]$.

iii) Wir setzen die Träger aus ii) ein:

$$\begin{aligned}\operatorname{supp}(\psi_{2i}^{k+1}) \cup \operatorname{supp}(\psi_{2i+1}^{k+1}) &= \left[\frac{2i}{2^k}, \frac{2i+1}{2^k}\right] \cup \left[\frac{2i+1}{2^k}, \frac{2i+2}{2^k}\right] \\ &= \left[\frac{2i}{2^k}, \frac{2i+2}{2^k}\right] = \left[\frac{i}{2^{k-1}}, \frac{i+1}{2^{k-1}}\right] \\ &= \operatorname{supp}(\psi_i^k).\end{aligned}$$

iv) Da ψ_i^k ein skaliertes und verschobenes Haar-Wavelet ist, ist die Funktion auf der ersten Hälfte des Trägers 1 und auf der zweiten Hälfte -1, d.h. die Funktion hat gerade die drei angegebenen Unstetigkeitsstellen.

v) Zur Orthogonalität. Zunächst sei $k = l > 1 \wedge i \neq j$. Dann sind die Träger disjunkt und somit $\langle \psi_i^k, \psi_j^k \rangle = 0$. Wegen der Symmetrie genügt es nun $k > l$ anzunehmen. Die Unstetigkeitsstellen von Funktionen auf Stufe l liegen an Vielfachen von 2^{-l}, also an den Endpunkten von Trägern von Stufe-k-Funktionen, aber nie im Inneren deren Träger. Somit gilt

$$\langle \psi_i^k, \psi_j^l \rangle = \int_{\frac{i}{2^{k-1}}}^{\frac{i+1}{2^{k-1}}} \psi_i^k(x)\psi_j^l(x)\,dx = c \int_{\frac{i}{2^{k-1}}}^{\frac{i+1}{2^{k-1}}} \psi_i^k(x)\,dx = 0,$$

Dann rechnen wir die Normierung nach:

$$\begin{aligned}&i = j \wedge k = l: \\ &\quad k = 0 : \langle \psi_0^0, \psi_0^0 \rangle = \langle \chi, \chi \rangle = \int_0^1 1\,dx = 1, \\ &\quad k > 0 : \langle \psi_i^k, \psi_i^k \rangle = \int_{2^{-(k-1)}\cdot i}^{2^{-(k-1)}\cdot(i+1)} \left(2^{\frac{k-1}{2}} \underbrace{\psi(2^{k-1}x - i)}_{=\pm 1}\right)^2 dx \\ &\qquad\qquad = \int_{2^{-(k-1)}\cdot i}^{2^{-(k-1)}\cdot(i+1)} 2^{k-1}\,dx = \frac{1}{2^{k-1}} \cdot 2^{k-1} = 1.\end{aligned}$$

vi) Folgt unmittelbar aus der Definition.

□

Mittels Beziehung (23.3) können wir nun eine Funktion f in der Haar-Waveletbasis darstellen:

$$g = \langle f, \psi_0^0 \rangle \psi_0^0 + \sum_{l=1}^{k} \sum_{i=0}^{2^{l-1}-1} \langle f, \psi_i^l \rangle \psi_i^l. \tag{24.6}$$

Aber wie sehen diese Funktionen nun aus? Das nächste Lemma charakterisiert die mittels Haar-Wavelets darstellbaren Funktionen.

Lemma 24.4. Es seien

$$\chi_i^k(x) = \begin{cases} 1 & \frac{i}{2^k} < x \leq \frac{i+1}{2^k} \\ 0 & \text{sonst} \end{cases} \qquad 0 \leq i < 2^k$$

und $\Xi^k = \{\chi_i^k : 0 \leq i < 2^k\}$ die Indikatorfunktionen der Intervalle der Länge 2^{-k}. Dann gilt

$$S^k = \operatorname{span} \Xi^k = \operatorname{span} \Psi^k = \left\{ v : (0,1] \to \mathbb{R} \,\middle|\, v|_{\left(\frac{i}{2^k}, \frac{i+1}{2^k}\right]} = const, 0 \leq i < 2^k \right\}$$

und für die Basisfunktionen gilt die Darstellung

$$\chi_i^k = c_{i,0}^{k,0}\psi_0^0 + \sum_{l=1}^{k} \sum_{j=0}^{2^{l-1}-1} c_{i,j}^{k,l}\psi_j^l.$$

Die Koeffizienten sind rekursiv gegeben durch

$$\begin{aligned} c_{0,0}^{0,0} &= 1 && && (k=0), \\ c_{0,0}^{1,0} &= \frac{1}{2}, && c_{0,0}^{1,1} = \frac{1}{2} && (k=1), \\ c_{1,0}^{1,0} &= \frac{1}{2}, && c_{1,0}^{1,1} = -\frac{1}{2} && (k=1). \end{aligned}$$

sowie

$$c_{2i,0}^{k+1,0} = \frac{1}{2}c_{i,0}^{k,0}, \quad c_{2i,j}^{k+1,l} = \frac{1}{2}c_{i,j}^{k,l} \ (1 \leq l \leq k, 0 \leq j < 2^{l-1}), \quad c_{2i,i}^{k+1,k+1} = 2^{\frac{2}{k}-1},$$
$$c_{2i+1,0}^{k+1,0} = \frac{1}{2}c_{i,0}^{k,0}, \quad c_{2i+1,j}^{k+1,l} = \frac{1}{2}c_{i,j}^{k,l} \ (1 \leq l \leq k, 0 \leq j < 2^{l-1}), \quad c_{2i+1,i}^{k+1,k+1} = -2^{\frac{2}{k}-1}.$$

Beweis. Für $k = 0$ gilt

$$\operatorname{span}\{\psi_0^0\} = \operatorname{span}\{\chi_0^0\} = S^0$$

und für $k = 1$ gilt

$$\begin{aligned} \chi_0^1 &= \frac{1}{2}\psi_0^0 + \frac{1}{2}\psi_0^1, \\ \chi_1^1 &= \frac{1}{2}\psi_0^0 - \frac{1}{2}\psi_0^1. \end{aligned} \tag{24.7}$$

Sei nun also $S^k = \operatorname{span}\Psi^k = \operatorname{span}\Phi^k$. Damit gibt es für jedes $0 \leq i < 2^k$ eine Darstellung

$$\chi_i^k = c_{i,0}^{k,0}\psi_0^0 + \sum_{l=1}^{k} \sum_{j=0}^{2^{l-1}-1} c_{i,j}^{k,l}\psi_j^l.$$

Nun gilt $\Psi^{k+1} = \Psi^k \cup \{\psi_i^{k+1} : 0 \leq i < 2^k\}$ und für $0 \leq i < 2^k$ erhalten wir

$$\chi_{2i}^{k+1} = \frac{1}{2}\chi_i^k + 2^{\frac{2}{k}-1}\psi_i^{k+1} = \frac{1}{2}c_{i,0}^{k,0}\psi_0^0 + \frac{1}{2}\sum_{l=1}^{k} \sum_{j=0}^{2^{l-1}-1} c_{i,j}^{k,l}\psi_j^l + 2^{\frac{2}{k}-1}\psi_i^{k+1},$$

$$\chi_{2i+1}^{k+1} = \frac{1}{2}\chi_i^k - 2^{\frac{2}{k}-1}\psi_i^{k+1} = \frac{1}{2}c_{i,0}^{k,0}\psi_0^0 + \frac{1}{2}\sum_{l=1}^{k} \sum_{j=0}^{2^{l-1}-1} c_{i,j}^{k,l}\psi_j^l - 2^{\frac{2}{k}-1}\psi_i^{k+1}.$$

Die Formeln für die Koeffizienten ergeben sich durch Koeffizientenvergleich. □

Eine Funktion $v \in S^k$ kann man natürlich sowohl in der Haar-Waveletbasis als auch in der Standardbasis darstellen. Die Umwandlung zwischen beiden Basisdarstellungen ist effizient machbar wie das folgende Lemma zeigt.

Lemma 24.5 (Basistransformationen)**.** Es sei $v \in S^k$, $k > 0$, dargestellt in der Haar-Waveletbasis und der Standardbasis:

$$v = \sum_{i=0}^{2^k-1} \alpha_i^k \chi_i^k = \beta_0^0 \psi_0^0 + \sum_{l=1}^{k} \sum_{i=0}^{2^{l-1}-1} \beta_i^l \psi_i^l.$$

Für die Transformation von der Haar-Waveletbasis in die Standardbasis gelten die Rekursionsformeln

$$\begin{aligned} \alpha_0^1 &= \beta_0^0 + \beta_0^1, & \alpha_1^1 &= \beta_0^0 - \beta_0^1, & (l = 1) \\ \alpha_i^l &= \alpha_{i/2}^{l-1} + (-1)^i 2^{\frac{l-1}{2}} \beta_{i/2}^l & 0 \le i < 2^l, & & (1 < l \le k). \end{aligned}$$

Dabei durchläuft man die Stufen $l = 1, \ldots, k$ von unten nach oben.

Für die umgekehrte Richtung gilt:

$$\begin{aligned} \beta_i^l = \langle v, \psi_i^l \rangle &= \frac{\alpha_{2i}^l}{2} - \frac{\alpha_{2i+1}^l}{2}, & & 1 \le l \le k,\ 0 \le i < 2^{l-1}, \\ \alpha_i^{l-1} &= \frac{\alpha_{2i}^l}{2} + \frac{\alpha_{2i+1}^l}{2}, & & 1 \le l \le k,\ 0 \le i < 2^{l-1}, \\ \beta_0^0 &= \alpha_0^0 & & l = 0. \end{aligned}$$

Dabei durchläuft man die Stufen $l = k, \ldots, 0$ von oben nach unten.

Der Rechenaufwand ist für beide Richtungen $O(2^k)$, also linear in der Anzahl der Basisfunktionen.

Beweis. Zunächst die Richtung von der Waveletbasis in die Standardbasis. Für $k = 1$ erhalten wir aus (24.7):

$$\psi_0^0 = \chi_0^0 + \chi_1^1, \qquad \psi_0^1 = \chi_0^1 - \chi_1^1$$

und damit

$$\begin{aligned} v &= \beta_0^0 \psi_0^0 + \beta_0^1 \psi_0^1 = \beta_0^0 (\chi_0^0 + \chi_1^1) + \beta_0^1 (\chi_0^1 - \chi_1^1) \\ &= (\beta_0^0 + \beta_0^1) \chi_0^1 + (\beta_0^0 - \beta_0^1) \chi_1^1, \end{aligned}$$

also

$$\alpha_0^1 = \beta_0^0 + \beta_0^1, \qquad \alpha_1^1 = \beta_0^0 - \beta_0^1.$$

Nun sei bis zur Stufe $k-1$ bereits eine Konversion durchgeführt, d.h. es sei

$$v = \beta_0^0 \psi_0^0 + \sum_{l=1}^{k-1} \sum_{i=0}^{2^{l-1}-1} \beta_i^l \psi_i^l = \sum_{i=0}^{2^{k-1}-1} \alpha_i^{k-1} \chi_i^{k-1},$$

dann erhalten wir die Darstellung auf Stufe k durch:

$$\begin{aligned}
v &= \beta_0^0\psi_0^0 + \sum_{l=1}^{k}\sum_{j=0}^{2^{l-1}-1}\beta_j^l\psi_j^l = \beta_0^0\psi_0^0 + \sum_{l=1}^{k-1}\sum_{j=0}^{2^{l-1}-1}\beta_j^l\psi_j^l + \sum_{i=0}^{2^{k-1}-1}\beta_i^k\psi_i^k \\
&= \sum_{i=0}^{2^{k-1}-1}\alpha_i^{k-1}\chi_i^{k-1} + \sum_{i=0}^{2^{k-1}-1}\beta_i^k\psi_i^k \\
&= \sum_{i=0}^{2^{k-1}-1}\left[\left(\alpha_i^{k-1} + 2^{\frac{k-1}{2}}\beta_i^k\right)\chi_{2i}^k + \left(\alpha_i^{k-1} - 2^{\frac{k-1}{2}}\beta_i^k\right)\chi_{2i+1}^k\right] \\
&= \sum_{i=0}^{2^k-1}\left(\alpha_{i/2}^{k-1} + (-1)^i 2^{\frac{k-1}{2}}\beta_{i/2}^k\right)\chi_i^k,
\end{aligned}$$

und somit die Rekursionsformel

$$\alpha_i^k = \alpha_{i/2}^{k-1} + (-1)^i 2^{\frac{k-1}{2}}\beta_{i/2}^k, \qquad 0 \le i < 2^k.$$

Nun die Richtung von der Standardbasis in die Waveletbasis. Auch hier gehen wir rekursiv über die Stufen vor. Zur Herleitung der Rekursion beobachten wir für $l > 1$ und $0 \le i < 2^{l-1}$:

$$\begin{array}{rcl} \chi_i^{l-1} &=& \chi_{2i}^l + \chi_{2i+1}^l \\ \psi_i^l &=& \chi_{2i}^l - \chi_{2i+1}^l \end{array} \quad \Leftrightarrow \quad \begin{array}{rcl} \chi_{2i}^l &=& \frac{1}{2}\chi_i^{l-1} + \frac{1}{2}\psi_i^l \\ \psi_{2i+1}^l &=& \frac{1}{2}\chi_i^{l-1} - \frac{1}{2}\psi_i^l \end{array}.$$

Damit können wir v darstellen als

$$\begin{aligned}
v &= \sum_{i=0}^{2^k-1}\alpha_i^k\chi_i^k = \sum_{i=0}^{2^{k-1}-1}\left(\alpha_{2i}^k\chi_{2i}^k + \alpha_{2i+1}^k\chi_{2i+1}^k\right) \\
&= \sum_{i=0}^{2^{k-1}-1}\left(\alpha_{2i}^k\left(\frac{1}{2}\chi_i^{k-1} + \frac{1}{2}\psi_i^k\right) + \alpha_{2i+1}^k\left(\frac{1}{2}\chi_i^{k-1} - \frac{1}{2}\psi_i^k\right)\right) \\
&= \sum_{i=0}^{2^{k-1}-1}\left(\frac{\alpha_{2i}^k}{2} + \frac{\alpha_{2i+1}^k}{2}\right)\chi_i^{k-1} + \sum_{i=0}^{2^{k-1}-1}\left(\frac{\alpha_{2i}^k}{2} - \frac{\alpha_{2i+1}^k}{2}\right)\psi_i^k
\end{aligned}$$

Damit berechne:

$$\begin{aligned}
\beta_i^l = \langle v, \psi_i^l\rangle &= \frac{\alpha_{2i}^l}{2} - \frac{\alpha_{2i+1}^l}{2}, && 1 \le l \le k,\ 0 \le i < 2^{l-1}, \\
\alpha_i^{l-1} &= \frac{\alpha_{2i}^k}{2} + \frac{\alpha_{2i+1}^k}{2}, && 1 \le l \le k,\ 0 \le i < 2^{l-1}, \\
\beta_0^0 &= \alpha_0^0 && l = 0.
\end{aligned}$$

Das Rekursionsende folgt aus der Korrespondenz

$$\beta_0^0 = \frac{1}{2}\alpha_0^1 + \frac{1}{2}\alpha_1^1, \qquad \beta_0^1 = \frac{1}{2}\alpha_0^1 - \frac{1}{2}\alpha_1^1, \qquad (k = 1).$$

□

24.3 Datenkompression mit Wavelets

Die Anwendung von Gleichung (23.4) aus Abschnitt 23.3 erlaubt eine Datenkompression mit vorgegebener Toleranz mit Hilfe der Haar-Wavelets.

Durch die Indexmenge $I \subseteq \{(i,k) : k \in \mathbb{N}_0, 0 \leq i < 2^{k-1}\}$ sei eine beliebige Menge von Basisfunktionen ausgewählt.

Algorithmus 24.6.

Gegeben: Funktion $f \in S^k$.
Gesucht: Teilraum $\tilde{S} \subset S^k$ möglichst klein, sodass $\|f - \tilde{f}\| \leq \mathrm{TOL}\langle f, f\rangle$.

Möglicher Algorithmus:

Wähle eine Startmenge I, z.B. $I = \{(0,0)\}$.
Berechne $s = \sum_{(i,k)\in I} \langle f, \psi_i^k\rangle^2$
while $s < (1 - \mathrm{TOL}^2)\langle f, f\rangle$ **do**
  wähle Kind $(j, l+1)$ von $(i,l) \in I$ mit $\langle f, \psi_j^{l+1}\rangle^2$ maximal.
  $I = I \cup \{(j, l+1)\}$ {Füge Basisfunktion mit maximalem Fehler hinzu}
  $s = s + \langle f, \psi_j^{l+1}\rangle^2$
end while

Vorlesung 25

Numerische Integration niedriger Ordnung

Eine Grundaufgabe der Numerik ist die Berechnung von bestimmten Integralen. So erfordert die Bestimmung der Koeffizienten bei der Gauß-Approximation mit Orthonormalbasen die Berechnung des Integrals $\langle f, \psi_i \rangle = \int_a^b f\psi_i \, dx$.

Wir beschränken uns hier auf die numerische Berechnung bestimmter Integrale in einer Raumdimension:

$$I_{[a,b]}(f) = \int_a^b f(x) \, dx.$$

Alle hier behandelten Verfahren führen auf Formeln der Art

$$I_{[a,b]}(f) = (b-a) \sum_{i=0}^{n} w_i f(x_i) + \text{ Fehler}$$

wobei

$$w_i \in \mathbb{R} \text{ die Gewichte}$$
$$\text{und } x_i \in \mathbb{R} \text{ die Stützstellen sind.}$$

Solche Formeln nennt man auch Quadraturformeln.

Mittels einsetzen von $f \equiv 1$ gilt

$$\int_a^b 1 \, dx = b - a = (b-a) \sum_{i=0}^{n} w_i \cdot 1 \iff \sum_{i=0}^{n} w_i = 1.$$

Die Gewichte summieren sich also stets zu 1.

25.1 Newton-Cotes Formeln

Als erstes betrachten wir die sog. Newton-Cotes Formeln, die ein Spezialfall interpolatorischer Quadraturformeln sind.

Diese Formeln basieren auf der folgenden Idee: Zu gegebenem $f(x)$ stelle eine Interpolationspolynom $p(x)$ zu gewissen Stützstellen auf und berechne das Intergral über $p(x)$ exakt.

Dieser Idee folgend, betrachte also das Interpolationspolynom vom Grad n in der Lagrangebasis zu den Stützstellen $(x_i, f(x_i))$, $i = 0, \ldots, n$:

$$p_n(x) = \sum_{i=0}^{n} f(x_i) L_i^{(n)}(x), \qquad L_i^{(n)}(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)}.$$

Dann berechnet sich das Integral wie folgt:

$$I_{[a,b]}(f) \approx I_{[a,b]}^{(n)}(f) = \int_a^b p_n(x) \, dx = \sum_{i=0}^{n} \left(f(x_i) \int_a^b L^{(n)}(x) \, dx \right). \tag{25.1}$$

Die Newton-Cotes-Formeln ergeben sich wenn man *äquidistante* Stützstellen wählt. Dann gibt es noch zwei Varianten:

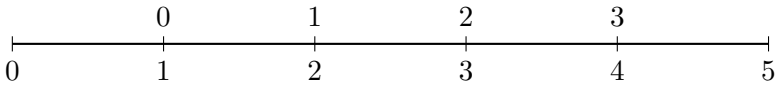
a) Abgeschlossene Formeln: Intervallenden a, b sind Stützstellen

$$x_i = a + i \cdot h \quad i = 0, \ldots, n, \quad h = \frac{b-a}{n}.$$

b) Offene Formeln: Intervallenden a, b sind keine Stützstellen

$$x_i = a + (i+1) \cdot h \quad i = 0, \ldots, n, \quad h = \frac{b-a}{n+2}.$$

Stützpunkte der abgeschlossenen Formeln für $n = 5$ (unten) und der offenen Formeln für $n = 3$ (oben):



Die offenen Formeln benutzen Werte des Interpolationspolynoms außerhalb des Intervalls der Stützstellen, d.h. Extrapolation.

Die Gewichte berechnet man dann bei den abgeschlossenen Formeln wie folgt:

$$I^{(n)}_{[a,b]}(f) \overset{(25.1)}{=} \sum_{i=0}^{n} f(x) \int_a^b L_i^{(n)}(x)\,dx = (b-a) \sum_{i=0}^{n} \underbrace{\left(\frac{1}{b-a} \int_a^b L_i^{(n)}(x)\,dx \right)}_{=:w_i} \cdot f(x_i).$$

Mittels Substitution $x = g(s) = a + sh$, $h = \frac{b-a}{n}$, $g'(s) = h$ ergibt sich:

$$\begin{aligned} w_i &= \frac{1}{b-a} \int_a^b L_i^{(n)}(x)\,dx = \frac{1}{b-a} \int_{g^{-1}(a)}^{g^{-1}(b)} L_i^{(n)}(a+sh) \underbrace{g'(s)}_{=h}\,\mathrm{d}s \\ &= \frac{1}{b-a} \underbrace{\frac{b-a}{n}}_{=h} \int_0^n \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{[a+sh-(a+jh)]}{[a+ih-(a+jh)]}\,\mathrm{d}s = \frac{1}{n} \int_0^n \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{s-j}{i-j}\,\mathrm{d}s. \end{aligned}$$

Berechnet man diese Gewichte, so erhält man:

a) Abgeschlossene Formeln: $n = 1, 2, 3, \quad h = \frac{b-a}{n}$

$$I^{(1)}_{[a,b]}(f) = \frac{b-a}{2} [f(a) + f(b)] \quad \text{Trapezregel}$$

$$I^{(2)}_{[a,b]}(f) = \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \quad \text{Simpsonregel / Keplersche Faßregel}$$

$$I^{(3)}_{[a,b]}(f) = \frac{b-a}{8} [f(a) + 3f(a+h) + 3f(b-h) + f(b)] \quad \frac{3}{8}\text{-Regel}$$

b) Offene Formeln $n = 0, 1, 2, \quad h = \frac{b-a}{n+2}$

$$I^{(0)}_{[a,b]}(f) = (b-a) f\left(\frac{a+b}{2}\right) \quad \text{Mittelpunktsregel, Tangenten-Trapez, Rechteckregel}$$

$$I^{(1)}_{[a,b]}(f) = \frac{b-a}{2} [f(a+h) + f(b-h)]$$

$$I^{(2)}_{[a,b]}(f) = \frac{b-a}{3} \left[2f(a+h) - f\left(\frac{a+b}{2}\right) + 2f(b-h) \right]$$

Bemerkung 25.1. Ab $n = 7$ für abgeschlossene und $n = 2$ für offene Formeln treten negative Gewichte w_i auf. Dies ist ungünstig, weil

- Für $f(x) \geq 0$ garantieren positive w_i, dass auch $I^{(n)}(f) \geq 0$, sonst nicht.
- Erhöhte Gefahr von Auslöschung.
- Die Kondition verschlechtert sich. Sei $\tilde{f}(x_i) = f(x_i) + \Delta y_i$ eine gestörte Auswertung mit $|\Delta y_i| \leq \varepsilon$. Dann gilt:

$$\begin{aligned} I^{(n)}(\tilde{f}) &= (b-a)\sum_{i=0}^{n} \tilde{f}(x_i) w_i = (b-a)\sum_{i=0}^{n} \left(f(x_i) + \Delta y_i\right) w_i \\ &= I^{(n)}(f) + (b-a)\sum_{i=0}^{n} w_i \Delta y_i, \end{aligned}$$

also

$$\left| I^{(n)}(\tilde{f}) - I^{(n)}(f) \right| = \left| (b-a)\sum_{i=1}^{n} w_i \Delta y_i \right| \leq \varepsilon\,(b-a)\sum_{i=0}^{n} |w_i|$$

Sind alle w_i positiv, so gilt $\sum_{i=0}^{n} |w_i| = \sum_{i=1}^{n} w_i = 1$. Gibt es auch Elemente mit umgekehrtem Vorzeichen wird die Kondition schlechter.

Satz 25.2 (Restglieder)**.** Es gelten die folgenden Fehlerdarstellungen:

1. Trapezregel:

$$I(f) - \frac{b-a}{2}\left[f(a) + f(b)\right] = -\frac{(b-a)^3}{12} f''(\xi) \quad f \in C^2([a,b]),\ \xi \in [a,b]$$

2. Simpson-Regel:

$$I(f) - \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right] = -\frac{(b-a)^5}{2880} f^{(4)}(\xi) \quad f \in C^4([a,b]),\ \xi \in [a,b]$$

3. Mittelpunktregel

$$I(f) - (b-a)\, f\left(\frac{a+b}{2}\right) = \frac{(b-a)^3}{24} f''(\xi) \quad f \in C^2([a,b]),\ \xi \in [a,b].$$

Beweis. Mit der Interpolationsfehlerdarstellung aus Satz 18.2 erhalten wir

$$\int_a^b f(x) - p_n(x)\, dx = \int_a^b \frac{f^{(n+1)}(\eta(x))}{(n+1)!} \prod_{j=0}^{n} (x - x_j)\, dx.$$

Zunächst die Trapezregel ($n = 1$). Hier kann man direkt den Mittelwertsatz der Integralrechnung nutzen, da die Funktion $g(x) = (x-a)(x-b)$ keinen Vorzeichenwechsel im Intervall $[a,b]$ hat.

$$\begin{aligned} I(f) - I^{(1)}(f) &= \frac{1}{2}\int_a^b f''(\eta(x)) \underbrace{(x-a)(x-b)}_{=:g(x)\leq 0}\, dx \\ &\overset{\substack{\text{Mittelwert-}\\ \text{satz der}\\ \text{Integral-}\\ \text{rechnung}}}{=} \frac{1}{2} f''(\xi) \int_a^b (x-a)(x-b)\, dx = -\frac{(b-a)^3}{12} f''(\xi),\ \xi \in [a,b] \end{aligned}$$

Bei den anderen Formeln ist ein zusätzlicher Trick notwendig, da die Funktion $\prod_{j=0}^{n}(x-x_j)$ dann einen Vorzeichenwechsel hat. Wir betrachten nur noch die Mittelpunktregel. Es gilt die Taylorentwicklung

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

und damit

$$\begin{aligned} I(f) - (b-a)\, f\left(\frac{a+b}{2}\right) &= \int_a^b f(x) - f\left(\frac{a+b}{2}\right) dx \\ &= \int_a^b \frac{f''(\eta(x))}{2}\left(x - \frac{a+b}{2}\right)^2 dx + f'\left(\frac{a+b}{2}\right) \underbrace{\int_a^b \left(x - \frac{a+b}{2}\right) dx}_{=0} \\ &= \int_a^b \frac{f''(\eta(x))}{2}\left(x - \frac{a+b}{2}\right)^2 dx = \frac{f''(\xi)}{2} \int_a^b \left(x - \frac{a+b}{2}\right)^2 dx \\ &= \frac{f''(\xi)}{2} \int_{-\frac{1}{2}(b-a)}^{\frac{1}{2}(b-a)} z^2\, dz = f''(\xi)\frac{(b-a)^3}{24}. \end{aligned}$$

Für den Rest siehe [Rannacher, 2017, Abschnitt 3.1]. □

Wir beobachten:

- Die Fehlerabschätzung für die Mittelpunktsregel liefert den halben Fehler der Trapezregel bei nur einer Auswertung von f.
- Die Restglieder haben immer die typische Form: $c\,(b-a)^{m+1}\, f^{(m)}(\xi)$.

Die Potenz m ist eine charakteristische Eigenschaft einer Quadraturformel. Diese drückt sich aus in der folgenden Definition.

Definition 25.3 (Ordnung einer Quadratur)**.** Eine Quadraturformel $I^{(n)}(f)$ hat mindestens die Ordnung m, wenn sie Polynome vom Grad $m-1$ exakt integriert. Die Ordnung ist genau m, wenn Polynome vom Grad $m-1$ exakt integriert werden, solche vom Grad m aber nicht.

Bemerkung 25.4. Für die obigen Newton-Cotes Formeln gilt demnach:

- Bei $n+1$ äquidistanten Stützstellen wird ein Polynom vom Grad n exakt integriert, d.h. die Quadraturformel hat mindestens Ordnung $n+1$.
- Wenn ein Restglied die Form $Cf^{(m)}(\xi)$ hat, dann ist die Ordnung mindestens m. Denn ist f ein Polynom vom Grad $m-1$ dann ist die Ableitung der Ordnung m gleich Null und damit auch der Fehler. Somit wird ein Polynom vom Grad $m-1$ exakt integriert und die Ordnung ist mindestens m. Die Simpsonregel interpoliert Polynome vom Grad 3 exakt und hat demnach mindestens die Ordnung 4.
- Bei geschickter Wahl der Stützstellen und Gewichte ist maximal die Ordnung $2n+2$ bei $n+1$ Stützstellen erreichbar wie wir sehen werden.

25.2 Summierte Quadraturformeln

Erhöhen des Polynomgrades ist bei äquidistanten Stützstellen wenig sinnvoll, da

- negative Gewichte auftreten,
- die Lagrange-Interpolation nicht gleichmäßig konvergiert und
- entsprechende Differenzierbarkeit von f gegeben sein muss.

Ähnlich wie bei den Splines verwendet man die Quadraturformeln stückweise:

- Unterteile das Intervall $[a, b]$ in N Teilintervalle

$$[x_i, x_{i+1}], \quad x_i = a + ih, \quad i = 0, \ldots, N-1, \quad h = \frac{b-a}{N}.$$

- Wende eine der obigen Formeln $I^{(n)}(f)$ in jedem Teilintervall $[x_i, x_{i+1}]$ an.
- Das Ergebnis nennt man dann „summierte Quadraturformel“.

Satz 25.5 (Restglied für summierte Quadraturen)**.** Für die je Teilintervall verwendete Quadraturformel mit $n+1$ Stützstellen gelte

$$I_{[x_i,x_{i+1}]}(f) - I^{(n)}_{[x_i,x_{i+1}]}(f) = \alpha_n h^{m+2} f^{(m+1)}(\xi_i) \quad \xi_i \in [x_i, x_{i+1}], f \in C^{(m+1)}([a,b]).$$

Dann gilt für die summierte Quadraturformel:

$$I(f) - I^{(n)}_h(f) = \alpha_n (b-a) h^{m+1} f^{(m+1)}(\xi) \quad \xi \in [a,b], f \in C^{(m+1)}([a,b]).$$

Beweis. Es sei zunächst der Zwischenwertsatz zitiert: Sei $g(x)$ stetig auf $[\alpha, \beta]$, dann gibt es zu jedem $u \in [\min(g(\alpha), g(\beta)), \max(g(\alpha), g(\beta))]$ ein $\eta \in [\alpha, \beta]$ so dass $g(\eta) = u$.

Aus dem Zwischenwertsatz folgt dann: Sei $\xi_i \in [a,b]$, $i = 0, \ldots, N-1$ und g stetig auf $[a,b]$. Dann nimmt g jeden Wert zwischen $\min_i g(\xi_i)$ und $\max_i(g(\xi_i))$ an. Wegen $\min_i g(\xi_i) \leq \frac{1}{N}\sum_{i=0}^{N-1} g(\xi_i) \leq \max_i g(\xi_i)$ existiert $\xi \in [a,b]$ mit $\sum_{i=0}^{N-1} g(\xi_i) = N g(\xi)$. Also:

$$\begin{aligned} I(f) - I^{(n)}_h(f) &= \sum_{i=0}^{N-1} \alpha_n h^{m+2} f^{(m+1)}(\xi_i) \\ &= \alpha_n h^{m+2} \underbrace{\sum_{i=0}^{N-1} f^{(m+1)}(\xi_i)}_{= N f^{(m+1)}(\xi)} = \alpha_n h^{m+2} N f^{(m+1)}(\xi) \\ &\overset{h=\frac{b-a}{N}}{=} \alpha_n h^{m+2} \frac{b-a}{h} f^{(m+1)}(\xi) = \alpha_n h^{m+1} (b-a) f^{(m+1)}(\xi) \end{aligned}$$

□

Beispiel 25.6. Für die Trapezregel mit $n = 1$ (2 Stützstellen), $m = 1$ ($O(h^3)$ auf einem Teilintervall) erhält man global (auf dem Gesamtintervall) einen Fehler der Form $O(h^2)$. Dies motiviert auch die Definition der Ordnung. Für die Simpsonregel gilt $n = 2, m = 3$, also $O(h^4)$ globale Konvergenzordnung. □

Die Formeln für die summierten Formeln lauten:

1. Summierte Trapezregel

$$I_h^{(1)} = \sum_{i=0}^{N-1} \underbrace{(x_{i+1} - x_i)}_{=h} \frac{1}{2} \left[f(x_i) + f(x_{i+1}) \right] = h \left[\frac{f(a)}{2} + \sum_{i=1}^{N-1} f(a+ih) + \frac{f(b)}{2} \right]$$

$$I(f) - I_h^{(1)}(f) = -\frac{b-a}{12} h^2 f''(\xi) \quad \xi \in [a,b], f \in C^{(2)}([a,b])$$

2. Summierte Simpsonregel

$$\begin{aligned} I_h^{(2)}(f) &= \sum_{i=0}^{N-1} \underbrace{(x_{i+1} - x_i)}_{=h} \frac{1}{6} \left[f(x_i) + 4f\left(\frac{x_i + x_{i+1}}{2} \right) f(x_{i+1}) \right] \\ &= h \left[\frac{f(a)}{6} + \sum_{i=1}^{N-1} \frac{f(a+ih)}{3} + \frac{2}{3} \sum_{i=0}^{N-1} f\left(a + \left(i + \frac{1}{2} \right) h \right) + \frac{f(b)}{6} \right] \end{aligned}$$

$$I(f) - I_h^{(2)}(f) = -\frac{b-a}{2880} h^4 f^{(4)}(\xi_i) \quad \xi \in [a,b], f \in C^{(4)}([a,b])$$

3. Summierte Mittelpunktregel

$$I_h^{(0)}(f) = \sum_{i=0}^{N-1} (x_{i+1} - x_i) \cdot f\left(\frac{x_i + x_{i+1}}{2} \right) = h \sum_{i=0}^{N-1} f\left(a + \left(i + \frac{1}{2} \right) h \right)$$

$$I(f) - I_h^{(0)} = \frac{b-a}{24} h^2 f''(\xi) \quad \xi \in [a,b], f \in C^{(2)}([a,b])$$

Bemerkung 25.7. Die summierte Simpson-Regel lässt sich aus summierter Trapez- und Mittelpunktregel zusammensetzen:

$$I_h^{(2)}(f) = \frac{1}{3} I_h^{(1)}(f) + \frac{2}{3} I_h^{(0)}(f).$$

Die summierte Trapezregel zu dem nächst feineren Gitter erhält man aus summierter Trapez- und Mittelpunktregel des gröberen Gitters:

$$I_{\frac{h}{2}}^{(1)} = \frac{1}{2} I_h^{(1)}(f) + \frac{1}{2} I_h^{(0)}(f)$$

Diese Formeln erlauben eine ökonomischere Auswertung bei fortgesetztem Halbieren durch Wiederverwendung von Funktionswerten. □

Beispiel 25.8 (Beispiele zu Quadraturformeln)**.** Wir betrachten folgende bestimmte Integrale:

(*i*) Eine einfache, unendlich oft differenzierbare Funktion:

$$\int_0^{\pi/2} \sin(x)\, dx = 1.$$

(*ii*) Eine glatte Funktion aber mit großen höheren Ableitungen:

$$\int_{-1}^{1} \frac{1}{10^{-5} + x^2}\, dx = 9.914588332462438 \cdot 10^2.$$

(*iii*) Eine nicht unendlich oft differenzierbare Funktion (Halbkreis):

$$\int_{-1}^{1} \sqrt{1-x^2}\,dx = \pi/2.$$

Summierte Trapezregel für (*i*).

I	Fehler	#Fktausw.
$9.480594489685199 \cdot 10^{-1}$	$5.1941 \cdot 10^{-2}$	3
$9.871158009727754 \cdot 10^{-1}$	$1.2884 \cdot 10^{-2}$	5
$9.967851718861696 \cdot 10^{-1}$	$3.2148 \cdot 10^{-3}$	9
$9.991966804850723 \cdot 10^{-1}$	$8.0332 \cdot 10^{-4}$	17
$9.997991943200188 \cdot 10^{-1}$	$2.0081 \cdot 10^{-4}$	33
$9.999498000921015 \cdot 10^{-1}$	$5.0200 \cdot 10^{-5}$	65
$9.999874501175253 \cdot 10^{-1}$	$1.2550 \cdot 10^{-5}$	129
$9.999968625352869 \cdot 10^{-1}$	$3.1375 \cdot 10^{-6}$	257
$9.999992156341920 \cdot 10^{-1}$	$7.8437 \cdot 10^{-7}$	513
...		
$9.999999999995609 \cdot 10^{-1}$	$4.3909 \cdot 10^{-13}$	524289
1.000000000003847	$3.8467 \cdot 10^{-12}$	1048577

Fehler viertelt sich jeweils, und das von Anfang an. Weniger als 10^{-13} wird mit `double` Genauigkeit nicht erreicht.

Summierte Simpsonregel für (*i*).

I	Fehler	#Fktausw.
1.000134584974194	$1.3458 \cdot 10^{-4}$	5
1.000008295523968	$8.2955 \cdot 10^{-6}$	9
1.000000516684706	$5.1668 \cdot 10^{-7}$	17
1.000000032265001	$3.2265 \cdot 10^{-8}$	33
1.000000002016129	$2.0161 \cdot 10^{-9}$	65
...		
1.000000000000006	$5.7732 \cdot 10^{-15}$	2049
1.000000000000002	$1.7764 \cdot 10^{-15}$	4097

Fehler reduziert sich jeweils um den Faktor $16 = (1/2)^4$, und das fast von Anfang an.

Summierte Trapezregel für (*ii*).

I	Fehler	#Fktausw.
$1.000009999900001 \cdot 10^{5}$	$9.9010 \cdot 10^{4}$	3
$5.000449983500645 \cdot 10^{4}$	$4.9013 \cdot 10^{4}$	5
$2.501113751079469 \cdot 10^{4}$	$2.4020 \cdot 10^{4}$	9
$1.252430268327760 \cdot 10^{4}$	$1.1533 \cdot 10^{4}$	17
$6.300548144658167 \cdot 10^{3}$	$5.3091 \cdot 10^{3}$	33
$3.227572909110977 \cdot 10^{3}$	$2.2361 \cdot 10^{3}$	65
...		
$9.914588332263689 \cdot 10^{2}$	$1.9875 \cdot 10^{-8}$	8193
$9.914588332412698 \cdot 10^{2}$	$4.9740 \cdot 10^{-9}$	16385

Fehlerverhalten am Anfang unklar, erst spät stellt sich h^2 ein.

Summierte Simpsonregel für (*ii*).

I	Fehler	#Fktausw.
$3.333899978334190 \cdot 10^4$	$3.2348 \cdot 10^4$	5
$1.668001673605744 \cdot 10^4$	$1.5689 \cdot 10^4$	9
$8.362024407438566 \cdot 10^3$	$7.3706 \cdot 10^3$	17
$4.225963298451690 \cdot 10^3$	$3.2345 \cdot 10^3$	33
$2.203247830595247 \cdot 10^3$	$1.2118 \cdot 10^3$	65
...		
$9.914588332462367 \cdot 10^2$	$7.0486 \cdot 10^{-12}$	8193
$9.914588332462367 \cdot 10^2$	$7.0486 \cdot 10^{-12}$	16385

Bis 4096 Auswertungen ist Simpson schlechter als Trapez. „Asymptotische Konvergenzrate“ stellt sich erst für genügend kleines h ein.

Summierte Trapezregel für (*iii*).

I	Fehler	#Fktausw.
1.000000000000000	$5.7080 \cdot 10^{-1}$	3
1.366025403784439	$2.0477 \cdot 10^{-1}$	5
1.497854534051220	$7.2942 \cdot 10^{-2}$	9
1.544909572178587	$2.5887 \cdot 10^{-2}$	17
1.561626518913870	$9.1698 \cdot 10^{-3}$	33
1.567551211438566	$3.2451 \cdot 10^{-3}$	65
...		
1.570789982705718	$6.3441 \cdot 10^{-6}$	4097
1.570794083803873	$2.2430 \cdot 10^{-6}$	8193
1.570795533774854	$7.9302 \cdot 10^{-7}$	16385

Die Konvergenzordnung h^2 wird nicht erreicht, sondern nur ein h^α mit $\alpha < 2$ (siehe unten).

Summierte Simpsonregel für (*iii*).

I	Fehler	#Fktausw.
1.488033871712585	$8.2762 \cdot 10^{-2}$	5
1.541797577473481	$2.8999 \cdot 10^{-2}$	9
1.560594584887709	$1.0202 \cdot 10^{-2}$	17
1.567198834492299	$3.5975 \cdot 10^{-3}$	33
1.569526108946797	$1.2702 \cdot 10^{-3}$	65
...		
1.570793849184461	$2.4776 \cdot 10^{-6}$	4097
1.570795450836595	$8.7596 \cdot 10^{-7}$	8193
1.570796017098507	$3.0970 \cdot 10^{-7}$	16385

Die Simpsonregel zeigt *dieselbe* Konvergenzordnung wie die Trapezregel!

Bemerkung 25.9 (Zum Begriff der Konvergenzordnung)**.** Satz 25.5 liefert eine Konvergenzabschätzung der Form

$$|I(f) - I_h^{(n)}(f)| \leq Ch^{m+1}.$$

Für die summierte Trapezregel gilt $m = 1$, man spricht von h^2-Konvergenz, für die summierte Simpsonregel gilt $m = 3$, man hat h^4-Konvergenz.

Wir haben gesehen, dass die Konvergenzordnung kleiner als $m+1$ sein kann, wenn die Funktion nicht genügend oft differenzierbar ist. Experimentell können wir diese folgendermaßen bestimmen.

Mit dem Ansatz $e_h = |I(f) - I_h^{(n)}(f)| = Ch^\alpha$ gilt

$$\frac{e_{h/2}}{e_h} = \frac{C(h/2)^\alpha}{Ch^\alpha} = (1/2)^\alpha$$

und daraus erhalten wir

$$\alpha = \log\left(\frac{e_{h/2}}{e_h}\right) \Big/ \log\left(\frac{1}{2}\right).$$

Das so bestimmte α heißt *experimental order of convergence* (EOC). Im letzten Beispiel oben erhalten wir $\alpha = 3/2$. □

25.3 Fehlerkontrolle

Auch bei der Integration stellt sich die Frage den entstehenden Fehler abzuschätzen und die Schrittweite h so zu steuern, dass der Fehler klein genug ist.

Dazu die Idee: Die Simpson-Summe konvergiert schneller als die Trapezsumme (wenn f glatt genug ist), hat also für genügend kleines h einen kleineren Fehler.

Wir wollen den Fehler in der Trapezsumme für eine Intervallbreite $h/2$ abschätzen. Dazu „schieben“ wir die Auswertung der Simpsonsumme dazwischen:

$$|I(f) - I_{\frac{h}{2}}^{(1)}(f)| = |I(f) - I_h^{(2)}(f) + I_h^{(2)}(f) - I_{\frac{h}{2}}^{(1)}(f)|.$$

Mit der Dreiecksungleichung ergibt sich:

$$|I(f) - I_{\frac{h}{2}}^{(1)}(f)| \leq |I(f) - I_h^{(2)}(f)| + |I_h^{(2)}(f) - I_{\frac{h}{2}}^{(1)}(f)|.$$

Nimmt man nun an, dass die Simpsonsumme genauer ist als die Trapezsumme: $|I(f) - I_h^{(2)}(f)| \leq \omega |I(f) - I_{\frac{h}{2}}^{(1)}(f)|$ mit $0 < \omega < 1$, dann lässt sich abschätzen:

$$|I(f) - I_{\frac{h}{2}}^{(1)}(f)| \leq \omega |I(f) - I_{\frac{h}{2}}^{(1)}(f)| + |I_h^{(2)}(f) - I_{\frac{h}{2}}^{(1)}(f)|.$$

Auflösen nach dem Fehler in der Trapezsumme liefert:

$$|I(f) - I_{\frac{h}{2}}^{(1)}(f)| \leq \frac{1}{1-\omega} |I_h^{(2)}(f) - I_{\frac{h}{2}}^{(1)}(f)|.$$

Besonders ökonomisch lässt sich die Fehlerkontrolle zusammen mit Anmerkung 25.7 umsetzen (deshalb haben wir die Trapezregel zu $h/2$ und die Simpsonsumme zu h verwendet). Die Fehlerkontrolle lässt sich so ohne zusätzlichen Aufwand erledigen.

Algorithmus 25.10 (Integration mit Fehlerschätzung)**.**
Gegeben: $f : D_f \to \mathbb{R}$
Integrationsgrenzen $a, b \in D_f$
Toleranz $\epsilon > 0$
Faktor $\omega \in]0, 1]$

$h = b - a$
$N = 1$
$I^{(1)} = 0.5 \cdot h \cdot (f(a) + f(b))$
while $(h > \varepsilon)$ **do**
  $I^{(0)} = 0$;
  for $(i = 0, i < N, i = i + 1)$ **do** {Mittelpunktsumme}
    $I^{(0)} = I^{(0)} + h \cdot f(a + (i + 0.5) \cdot h)$
  end for
  $I^{(2)} = \frac{1}{3} I^{(1)} + \frac{2}{3} I^{(0)}$ {Simpson-Summe zu h}
  $I^{(1)} = 0.5 \cdot (I^{(1)} + I^{(0)})$ {Trapez-Summe zu $h/2$}
  $h = 0.5 \cdot h$
  $N = 2N$
  if $\left(\frac{1}{1-\omega} \cdot |I^{(2)} - I^{(1)}| \leq TOL \right)$ **then**
    return $I^{(1)}$ {Liefere Trapez-Summe zu h}
  end if
end while

Vorlesung 26

Numerische Integration hoher Ordnung

Bisher haben wir mit den Newton-Cotes Formeln interpolatorische Quadraturformeln mit äquidistanten Stützstellen kennengelernt. Wie gesehen sind diese bis maximal $n = 7$ im offenen Fall und $n = 2$ bei geschlossenen Formeln sinnvoll. Nun wenden wir uns der Frage zu wie man sinnvoll hohe Ordnung erreicht.

26.1 Romberg-Integration

Die nun beschriebene Romberg-Integration kann man als Extrapolationsverfahren auf Basis der summierten Trapezregel interpretieren.

Bei den summierten Formeln ist man an $\lim_{h\to 0} I_h^{(n)}(f)$ interessiert. Man kann also eine Extrapolation zum Limes anwenden. Dabei hilft folgender Satz:

Satz 26.1 (Euler-MacLaurinsche Summenformel)**.** Sei $f \in C^{2m+2}[a,b]$ und einmal integrierbar, dann gilt

$$\begin{aligned} \underbrace{I_h^{(1)}(f)}_{\text{Trapezsumme}} &= \int_a^b f(x)\,dx + \sum_{k=1}^{m} h^{2k}\frac{B_{2k}}{(2k)!}\Big(f^{(2k-1)}(b) - f^{(2k-1)}(a)\Big) \\ &\quad \underbrace{+ h^{2m+2}\frac{B_{2m+2}}{(2m+2)!}(b-a)f^{(2m+2)}(\zeta)}_{\textit{Restglied}} \qquad \zeta \in [a,b] \end{aligned} \tag{26.1}$$

mit $B_0 = 1$, $B_2 = \frac{1}{6}$, $B_4 = -\frac{1}{30}$, ..., den Bernoulli-Zahlen. Diese lassen sich rekursiv berechnen als

$$B_0 = 1,\ B_n = -\frac{1}{n+1}\sum_{k=0}^{n-1}\binom{n+1}{k}B_k.$$

ohne Beweis. □

Satz 26.1 bedeutet, dass für die Trapezsumme folgende Darstellung gilt:

$$I_h^{(1)}(f) = \underbrace{\tau_0}_{\int_a^b f(x)dx} + \underbrace{\tau_1 h^2 + \tau_2 h^4 \ldots + \tau_m h^{2m}}_{\substack{\text{unabhängig von } h! \\ \text{nur abh. von } f,a,b}} + \alpha_{m+1}(h)h^{2m+2} \tag{26.2}$$

wobei die Konstanten τ_i *nicht* von der Schrittweite h abhängen. Diese sog. *asymptotische Entwicklung* stellt ein Polynom in $y = h^2$ dar und wir befinden uns in der Situation von Abschnitt 19.3. Bei der „Extrapolation“ kombiniert man $I_h^{(1)}$ für verschiedene h so, dass möglichst viele Terme außer τ_0 Null werden.

Speziell in ihrer Anwendung auf die Berechnung von Integralen mittels Trapezsummen unterschiedlicher Schrittweiten heißt dieses Verfahren Romberg-Integration. Bei Anwendung auf allgemeine Diskretisierungsverfahren spricht man auch von Richardson-Extrapolation.

Zur praktischen Durchführung, d.h. Auswertung des Interpolationspolynoms zu den Stützwerten h_i^2 an der Stelle 0 eignet sich das Neville-Schema, welches angepasst lautet:

$$p_{i,0}(h_i^2) = I_{h_i}^{(1)}(f) \qquad i = 0, \ldots, n$$

$$p_{i,k}(0) = p_{i+1,k-1}(0) - \frac{p_{i+1,k-1}(0) - p_{i,k-1}(0)}{1 - \left(\frac{h_i}{h_{i+k}}\right)^2} \qquad k = 1, \ldots, n,\ i = 0, ..., n-k.$$

$p_{i,i+k}$ interpoliert Stützstellen h_i^2 bis h_{i+k}^2. Verwendet man als Schrittweite $h_i = h_0 \cdot 2^{-i}$ ergibt sich die Formel:

$$p_{i,k} = p_{i+1,k-1} + \frac{p_{i+1,k-1} - p_{i,k-1}}{2^{2k} - 1} \qquad \text{für } k = 1, \ldots, n,\ i = 0, ..., n-k.$$

Bei Interpolation von $n+1$ Werten $(h_i^2, I_{h_i}^{(1)}(f))$, $i = 0, \ldots, n$ gilt dann $p(0) = I(f) + O(h^{2n+2})$. *Aber*: Anwendung dieser Methode erfordert dann auch $f \in C^{2n+2}[a, b]$.

26.2 Gauß-Integration

Wir untersuchen n un die Frage ob sich die Genauigkeit von Quadraturformeln durch nicht nichtäquidistante Stützstellen verbessern lässt. Wähle also w_i und x_i so, dass Polynome von möglichst hohem Grad exakt integriert werden (Ordnungsmaximierung).

Beispiel 26.2. Finde x_0, x_1 und w_0, w_1 so, dass $p_3(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$ in $I = [-1, 1]$ exakt integriert wird:

$$\int_{-1}^{1} p_3(x)\,dx = [1 - (-1)] \cdot \sum_{i=0}^{1} w_i p_3(x_i) = 2 \cdot [w_0 p_3(x_0) + w_1 p_3(x_1)]$$

$$\Leftrightarrow \quad a_0 \underbrace{\int_{-1}^{1} 1\,dx}_{2} + a_1 \underbrace{\int_{-1}^{1} x\,dx}_{0} + a_2 \underbrace{\int_{-1}^{1} x^2\,dx}_{2/3} + a_3 \underbrace{\int_{-1}^{1} x^3\,dx}_{0} =$$

$$2w_0(a_0 + a_1 x_0 + a_2 x_0^2 + a_3 x_0^3) + 2w_1(a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3)$$

Koeffizientenvergleich (für die a_i) ergibt vier nichtlineare Gleichungen:

$$\left.\begin{array}{rcl} 2a_0 & = & 2a_0(w_0 + w_1) \\ 0a_1 & = & 2a_1(w_0 x_0 + w_1 x_1) \\ \frac{2}{3}a_2 & = & 2a_2(w_0 x_0^2 + w_1 x_1^2) \\ 0a_3 & = & 2a_3(w_0 x_0^3 + w_1 x_1^3) \end{array}\right\} \Rightarrow \text{Lösung: } w_0 = w_1 = \frac{1}{2} \qquad x_0 = \frac{-1}{\sqrt{3}},\ x_1 = \frac{1}{\sqrt{3}}.$$

Damit hat man also mit zwei Stützstellen eine Quadraturformel mit Ordnung 4 erhalten. Zum Vergleich: Die Trapezregel erreicht mit zwei Stützstellen nur Ordnung 2. Newton-Cotes Formeln haben allgemein bei $n+1$ Stützstellen mindestens die Ordnung $n+1$. □

Im folgenden beschränken wir uns auf Quadraturformeln für das Einheitsintervall $[-1, 1]$. Integrale über $[a, b]$ berechnet man per Transformation.

Man kann folgende Aussage zu nichtäquidistanten interpolatorischen Quadraturformeln zeigen.

Satz 26.3. Es gibt keine Quadraturformel mit $n+1$ Stützstellen welche ein Polynom vom Grad $2n+2$ exakt integriert, also die Ordnung $2n+3$ besitzt.

Beweis. Beweis erfolgt durch ein Gegenbeispiel. Angenommen man könnte Polynome vom Grad $2n+2$ bei $n+1$ Stützstellen exakt integrieren (Ordnung $2n+3$). Betrachte das Polynom

$$q(x) = \prod_{i=0}^{n} (x - x_i)^2.$$

Dieses hat folgende Eigenschaften:

- $q(x)$ hat Grad $2n+2$
- $q(x) \geq 0 \quad \forall\, x$ und $q(x) \not\equiv 0$ also ist $\int_{-1}^{1} q(x)\,dx > 0$.
- Andererseits gilt $\sum_{i=0}^{n} \underbrace{q(x_i)}_{=0} w_i = 0$, d.h. q wird nicht exakt integriert.

Damit ist ein Widerspruch $\lightning$ hergeleitet. Dies beweist natürlich nicht dass tatsächlich die Ordnung $2n+2$ ist. □

Die maximal mögliche Ordnung wird mit der Gauß-Quadratur erreicht:

Satz 26.4 (Gauß-Quadratur)**.** Es gibt genau eine interpolatorische Quadraturformel zu $n+1$ paarweise verschiedenen Stützstellen in $[-1,1]$ mit der Ordnung $2n+2$. Ihre Stützstellen sind die Nullstellen $\lambda_0, \ldots, \lambda_n \in (-1,1)$ des $(n+1)$-ten Legendrepolynoms P_{n+1}, siehe auch Beispiel 23.7:

$$P_{n+1}(x) = \frac{1}{2^{n+1} \cdot (n+1)!} \frac{\mathrm{d}^{n+1}}{\mathrm{d}x^{n+1}} (x^2 - 1)^{n+1}.$$

Die Gewichte erhält man mittels

$$w_i = \frac{1}{2} \int_{-1}^{1} \prod_{\substack{j=0 \\ j \neq i}}^{n} \left(\frac{x - \lambda_j}{\lambda_i - \lambda_j} \right)^2 dx$$

Beweis. Siehe [Rannacher, 2017, Satz 3.4]. □

Die Abbildung 26.1 zeigt einige Legendre-Polynome.

Beispiel 26.5. Für $n = 1, 2$ ergibt sich mit $h = \frac{b-a}{2}$, $c = \frac{b+a}{2}$:

$$I^{(1)}(f) = \frac{b-a}{2} \left[f\left(c - \sqrt{1/3}\, h\right) + f\left(c + \sqrt{1/3}\, h\right) \right] \qquad \text{Ordnung: 4,}$$

$$I^{(2)}(f) = \frac{b-a}{18} \left[5f\left(c - \sqrt{3/5}\, h\right) + 8f(c) + 5f\left(c + \sqrt{3/5}\, h\right) \right] \qquad \text{Ordnung: 6.}$$

Hier wurde schon auf das allgemeine Intervall $[a,b]$ transformiert. □

Entsprechend lassen sich auch summierte Formeln aufstellen.

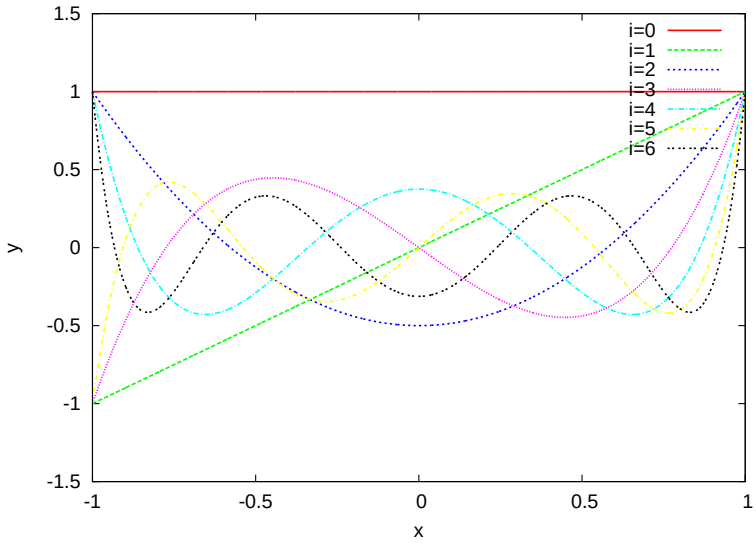
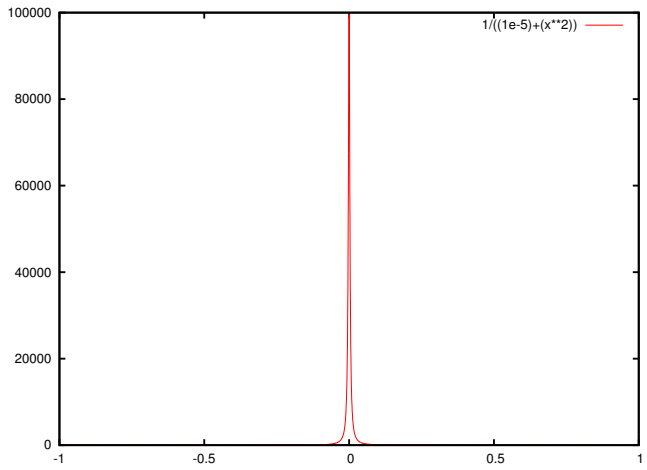


Abbildung 26.1: Die Legendrepolynome bis zum Grad 6.

26.3 Ausblick

Adaptive Quadratur

Quadratur mit konstanter Schrittweite ist bei manchen Integranden ineffizient. Betrachte z.B. die Funktion $f(x) = \frac{1}{10^{-5}+x^2}$.



In so einem Fall möchte man die Schrittweite „adaptiv“, d. h. angepasst an den speziellen Integranden wählen.

Dazu bedient man sich eines lokalen „Fehlerschätzers“ (oder Indikators), der angibt, an welcher Stelle die Schrittweite weiter verkleinert werden muss.

Als Beispiel in dieser Richtung betrachten wir das Prinzip von Archimedes In Abbildung 26.2 können wir das Integral (die Fläche) „hierarchisch“ zerlegen in die Anteile

$$I = I_0 + I_1 + \dots$$

mit

I_0 das Trapez $(a,0),(b,0),(b,f(b)),(a,f(a))$.

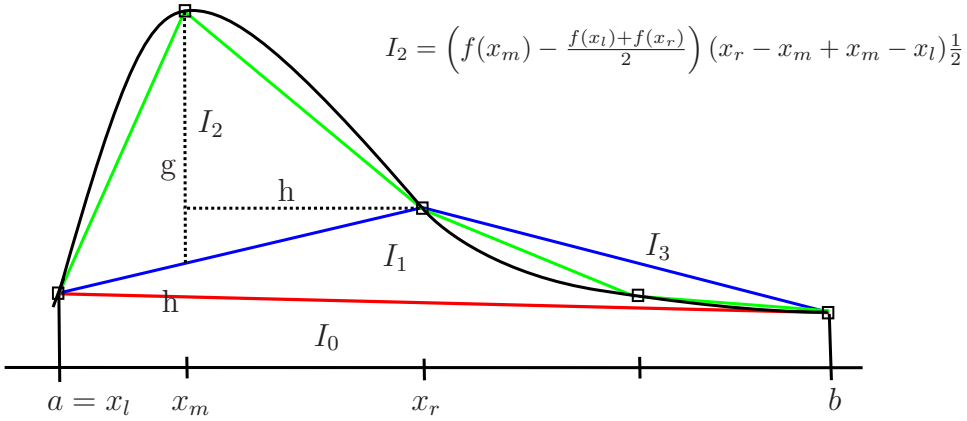


Abbildung 26.2: Zum Prinzip von Archimedes.

I_1 das Dreieck $(a, f(a)), (b, f(b)), ((a+b)/2, f((a+b)/2))$.

I_2 das Dreieck $(a, f(a)), ((a+b)/2, f((a+b)/2)), (a+1/4(b-a), f(a+1/4(b-a)))$.

Die hierarchischen Zuwächse sind jeweils allgemeine Dreiecke und berechnen sich aus der Fläche zweier rechtwinkliger Dreiecke. Seien die Punkte (x_l, f_l) und (x_r, f_r) gegeben, so berechnet sich die Fläche des Dreiecks ($A = g \cdot h/2$) mit $(x_m, f(x_m))$, $x_m = (x_l + x_r)/2$ mittels

$$I_\Delta = \left(f(x_m) - \frac{f(x_l) + f(x_r)}{2}\right) \frac{x_r - x_l}{2}.$$

Der hierarchische Zuwachs I_Δ dient gleichzeitig als lokaler Fehlerindikator. Ist er klein genug, so ist die Funktion dort gut angenähert und das Teilintervall muss *nicht* weiter verfeinert werden.

Algorithmus 26.6 (Adaptive Quadratur nach Archimedes)**.** Es bietet sich die Formulierung als rekursive Funktion an:

Archimedes $(x_l,\ x_r,\ f_l,\ f_r,\ l)$:
$x_m = (x_l + x_r)/2$
$f_m = f(x_m)$
$s = (f_m - (f_l + f_r)/2) \cdot (x_r - x_l)/2$
if $(s \geq \mathrm{TOL} \vee l < l_{\min})$ **then**
  s += Archimedes$(x_l, x_m, f_l, f_m, l+1)$+Archimedes$(x_m, x_r, f_m, f_r, l+1)$
end if
return s

Das Integral berechnet sich dann aus dem ersten Trapez I_0 und den Zuwächsen:

$$I = (b-a)(f(a) + f(b))/2 + \mathrm{Archimedes}(a, b, f(a), f(b), 0).$$

□

Vergleich verschiedener Verfahren höherer Ordnung

Beispiel 26.7 (Beispiel zur numerischen Quadratur hoher Ordnung)**.** Wir betrachten dieselben Funktionen wie in Beispiel 25.8:

(*i*) Eine einfache, unendlich oft differenzierbare Funktion:

$$\int_0^{\pi/2} \sin(x)\,dx = 1.$$

(*ii*) Eine glatte Funktion aber mit großen höheren Ableitungen:

$$\int_{-1}^{1} \frac{1}{10^{-5}+x^2}\,dx = 9.914588332462438 \cdot 10^2.$$

(*iii*) Eine nicht unendlich oft differenzierbare Funktion (Halbkreis):

$$\int_{-1}^{1} \sqrt{1-x^2}\,dx = \pi/2.$$

Vergleich verschiedener Quadraturen für (*i*):

Methode	I	Fehler	#Fktausw.
Gauss4	$9.999101667698898\cdot10^{-1}$	$8.9833\cdot10^{-5}$	4
	$9.999944679581383\cdot10^{-1}$	$5.5320\cdot10^{-6}$	8
	$9.999996555171785\cdot10^{-1}$	$3.4448\cdot10^{-7}$	16
	$9.999999784895880\cdot10^{-1}$	$2.1510\cdot10^{-8}$	32
Gauss6	1.000000118910998	$1.1891\cdot10^{-7}$	6
	1.000000001828737	$1.8287\cdot10^{-9}$	12
	1.000000000028461	$2.8461\cdot10^{-11}$	24
	1.000000000000444	$4.4409\cdot10^{-13}$	48
Archimedes	$9.480594489685199\cdot10^{-1}$	$5.1941\cdot10^{-2}$	3
	$9.871158009727754\cdot10^{-1}$	$1.2884\cdot10^{-2}$	5
	$9.967851718861697\cdot10^{-1}$	$3.2148\cdot10^{-3}$	9
	$9.990131153231707\cdot10^{-1}$	$9.8688\cdot10^{-4}$	15
	$9.997876171856270\cdot10^{-1}$	$2.1238\cdot10^{-4}$	31

Das Verfahren hoher Ordnung zahlt sich aus.

Vergleich verschiedener Quadraturen für (*ii*):

Methode	I	Fehler	#Fktausw.
Trapez	$1.000009999900001 \cdot 10^5$	$9.9010 \cdot 10^4$	3
	$3.227572909110977 \cdot 10^3$	$2.2361 \cdot 10^3$	65
	$1.765586982280199 \cdot 10^3$	$7.7413 \cdot 10^2$	129
	$1.160976493727309 \cdot 10^3$	$1.6952 \cdot 10^2$	257
	$1.003813438906513 \cdot 10^3$	$1.2355 \cdot 10^1$	513
	$9.915347090712996 \cdot 10^2$	$7.5876 \cdot 10^{-2}$	1025
	$9.914588358257512 \cdot 10^2$	$2.5795 \cdot 10^{-6}$	2049
Archimedes	$1.767335925226728 \cdot 10^3$	$7.7588 \cdot 10^2$	25
	$1.004348965298925 \cdot 10^3$	$1.2890 \cdot 10^1$	37
	$9.946212584262852 \cdot 10^2$	3.1624	81
	$9.922788393957054 \cdot 10^2$	$8.2001 \cdot 10^{-1}$	173
	$9.916266302474447 \cdot 10^2$	$1.6780 \cdot 10^{-1}$	361
	$9.914967844457766 \cdot 10^2$	$3.7951 \cdot 10^{-2}$	769
	$9.914672523966888 \cdot 10^2$	$8.4192 \cdot 10^{-3}$	1625
	$9.914606991793892 \cdot 10^2$	$1.8659 \cdot 10^{-3}$	3465
	$9.914592092819358 \cdot 10^2$	$3.7604 \cdot 10^{-4}$	7629

Fehlerreduktion mit Archimedes ist von Anfang an quadratisch, allerdings „überholt“ die Trapezsumme dann kräftig.

Vergleich verschiedener Quadraturen für (*iii*):

Methode	I	Fehler	#Fktausw.
Gauss4	1.592226038754547	$2.1430 \cdot 10^{-2}$	4
	1.570801362699711	$5.0359 \cdot 10^{-6}$	1024
	1.570798107100650	$1.7803 \cdot 10^{-6}$	2048
	1.570796956200537	$6.2941 \cdot 10^{-7}$	4096
	1.570796549318533	$2.2252 \cdot 10^{-7}$	8192
Gauss6	1.578036347519909	$7.2400 \cdot 10^{-3}$	6
	1.570801237513435	$4.9107 \cdot 10^{-6}$	768
	1.570798062869299	$1.7361 \cdot 10^{-6}$	1536
	1.570796940567470	$6.1377 \cdot 10^{-7}$	3072
	1.570796543792309	$2.1700 \cdot 10^{-7}$	6144
Archimedes	1.366025403784439	$2.0477 \cdot 10^{-1}$	5
	1.570774639679624	$2.1687 \cdot 10^{-5}$	365
	1.570791453003758	$4.8738 \cdot 10^{-6}$	765
	1.570795219591928	$1.1072 \cdot 10^{-6}$	1605
	1.570796082320714	$2.4447 \cdot 10^{-7}$	3433

Hohe Ordnung lohnt sich nicht wegen mangelnder Differenzierbarkeit. □

Mehrdimensionale Quadratur

Die Welt ist nicht eindimensional. Für Rechecke ($d = 2$) lassen sich obige Formeln leicht erweitern:

$$\int_c^d \int_a^b f(x,y)\,dx\,dy \approx (b-a)\sum_{i=0}^{n} w_i \int_c^d f(x_i,y)\,dy$$
$$\approx (b-a)(d-c)\sum_{i=0}^{n}\sum_{j=0}^{n} \underbrace{w_i\,w_j}_{=w_{ij}} f(x_i,y_j).$$

Dies lässt sich auf d-dimensionale Würfel verallgemeinern.

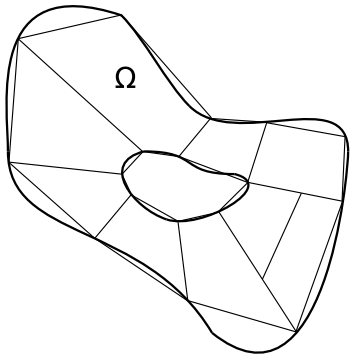


Abbildung 26.3: Triangulierung eines komplexen zweidimensionalen Gebietes mit Dreiecks- und Viereckselementen.

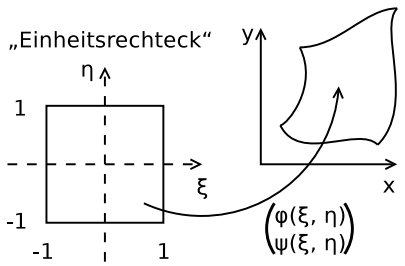
Zur Integration über kompliziertere Gebiete $\Omega \subset \mathbb{R}^d$ geht man wie folgt vor. Man zerlegt man Ω, ggf. näherungsweise, in Teilgebiete Ω_i welche man durch Transformation auf Würfel oder Simplizes abbilden kann, siehe Abbildung 26.3. Dies nennt man Triangulierung oder Gittergenerierung. Dies ist insbesondere in drei Raumdimensionen eine schwierige Aufgabe. Dabei tritt ein zusätzlicher Geometriefehler durch die nicht exakte Zerlegung auf.

Für die einzelnen Teilgebiete nutzt man dann den Transformationssatz für Integrale (hier in zwei Dimensionen):

$$\int_\Omega f(x,y)\,dx\,dy = \int_{-1}^{1}\int_{-1}^{1} f\left(\varphi(\xi,\eta),\psi(\xi,\eta)\right)\left|\frac{\partial(\varphi,\psi)}{\partial(\xi,\eta)}\right| \mathrm{d}\xi\,\mathrm{d}\eta.$$

Die Transformation

$$\begin{pmatrix}\varphi(\xi,\eta)\\ \psi(\xi,\eta)\end{pmatrix} : [-1,1]\times[-1,1] \to \Omega \subset \mathbb{R}^2$$



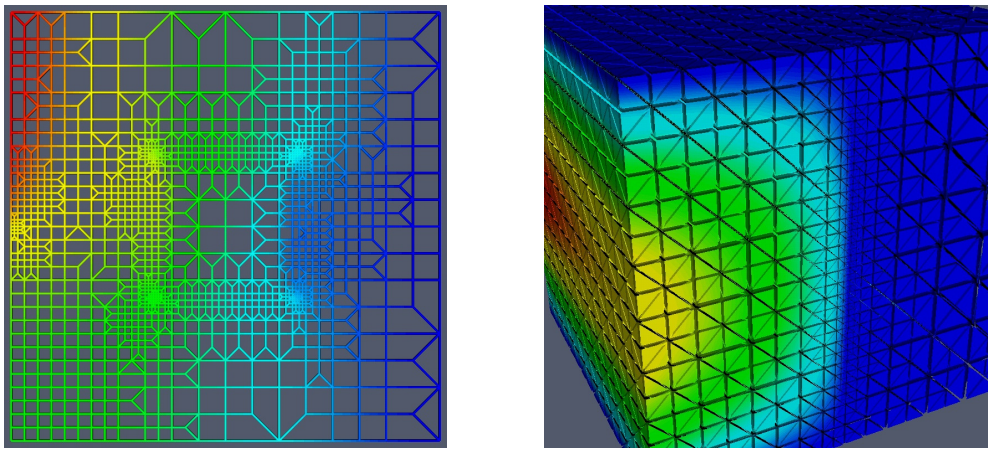


Abbildung 26.4: Links: Adaptives Dreiecks- und Viereckgitter. Rechts: Adaptives Tetraedergitter mit Bisektionsverfeinerung.

bildet das Gebiet $[-1,1]\times[-1,1]$ auf das Teilgebiet Ω_i ab. Durch entsprechende Abbildungen lassen sich auch krummlinig berandete Teilgebiete Ω_i behandeln. In der Formel ist

$$\left|\frac{\partial(\varphi,\psi)}{\partial(\xi,\eta)}\right| = \det \underbrace{\begin{pmatrix} \frac{\partial\varphi}{\partial\xi}(\xi,\eta) & \frac{\partial\psi}{\partial\xi}(\xi,\eta) \\ \frac{\partial\varphi}{\partial\eta}(\xi,\eta) & \frac{\partial\psi}{\partial\eta}(\xi,\eta) \end{pmatrix}}_{\text{(Transponierte Jacobimatrix)}} \neq 0$$

die Determinante der (transponierten) Jacobimatrix. Die Ordnung der Quadraturformel wird dann durch die zu integrierende Funktion f und die Transformation bestimmt.

Auch in mehr als einer Raumdimensionen kann man eine adaptive Verfeinerung der Zerlegung durchführen, siehe Abbildung 26.4.

Fluch der Dimension

Die bisher behandelten Methoden eignen sich gut bis Dimension etwa 5. Für sehr viel höhere Dimensionen d sind die hier behandelten Methoden nicht brauchbar, da der Aufwand exponentiell mit der Dimension ansteigt. Betrachte dazu den d-dimensionalen Würfel $\Omega = [0,1]^d$. Benutzt man in jede Richtung nur zwei Teilintervalle so hat man den d-dimensionalen Würfel in 2^d Teilwürfel zerlegt. Dies bezeichnet man gemeinhin als „Fluch der Dimension“. Eine Möglichkeit ist dann die Monte-Carlo Integration

$$I(f) \approx \frac{C}{N}\sum_{i=1}^{N} f(\xi_i) \quad \text{mit Zufallszahlen } \xi_i \in \Omega.$$

Hier ist die Anzahl der Auswertungen unabhängig von d, allerdings konvergiert das Verfahren nur wie $O(N^{-\frac{1}{2}})$.

Literaturverzeichnis

Martin Braun. *Differentialgleichungen und ihre Anwendungen*. Springer-Verlag Berlin Heidelberg, 1994.

Donald Estep. *Angewandte Analysis in einer Unbekannten*. Springer-Verlag, 2005.

G. Golub and J. M. Ortega. *Scientific Computing*. Teubner, 1996.

Gene Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 4th edition, 2013.

W. Hackbusch. *Iterative Lösung großer schwachbesetzter Gleichungssysteme*. Teubner, 1991.

William Ogilvy Kermack, A. G. McKendrick, and Gilbert Thomas Walker. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 115(772):700–721, 1927. doi: 10.1098/rspa.1927.0118. URL `https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1927.0118`.

Donald E. Knuth. *The Art of Computer Programming. Vol. 2: Seminumerical Algorithms*. Addison-Wesley, 1998.

Rolf Rannacher. *Numerik 0: Einführung in die Numerische Mathematik*. Heidelberg University Publishing, 2017. doi: https://doi.org/10.17885/heiup.206.281.

Rolf Rannacher. *Analysis 1: Differential- und Integralrechnung für Funktionen einer Veränderlichen*. Heidelberg University Publishing, 2018a. doi: https://doi.org/10.17885/heiup.317.431.

Rolf Rannacher. *Analysis 2: Differential- und Integralrechnung für Funktionen mehrerer reeller Veränderlichen*. Heidelberg University Publishing, 2018b. doi: https://doi.org/10.17885/heiup.381.

Robert Schaback and Holger Wendland. *Numerische Mathematik*. Springer-Verlag Berlin Heidelberg, 5. edition, 2005.