Eine "Mathematische Aufgabe" besteht abstrakt aus der Auswertung einer Abbildung

$$\phi : X \to Y \quad \text{in einem } x \in X \quad \text{mit geeigneten Räumen } X, Y$$

Beispiele

- Berechnung eines Integrals: $\int_a^b f(x)dx$:

$$\phi_{\int}((a,b), f) : X \times L^1 \to \mathbb{R}$$

- Lösung einer DGL

Objekte und Auswertungen können meist nur näherungsweise dargestellt werden, da z.B. nicht jede reelle Zahl auf dem Computer exakt dargestellt werden kann.

- Durch nicht exakte Darstellung entstehen Rundungsfehler
- Durch vereinfachte Beschreibung komplexer Vorgänge können auch Modellfehler enstehen
- Durch ungenaue Messungen können Datenfehler entstehen

Die Numerik befasst sich unter anderem mit folgenden Fragestellungen:

Algorithmik: Angabe von Algorithmen bzw. Berechnungsverfahren zur näherungsweisen Lösung von math. Aufgaben

Konditionierung und Stabilität: Einfluss von Störungen(Fehlern) auf das Ergebnis der math. Aufgabe oder Berechnung

Konvergenz: Abschätzung des Fehlers zwischen berechneter und exakter Lösung

Komplexität: Aufwand des numerischen Verfahrens

0.1 Zahlendarstellung und Rundungsfehler

Computer können Zahlen nur mit endlich vielen Ziffern darstellen, damit sind nicht alle reellen (komplexen) Zahlen exakt darstellbar. Manche Programme können Ganzzahlen mit beliebig vielen Stellen oder Gleitkommazahlen mit beliebig vielen Stellen darstellen (endlich viele, auch begrenzt durch Speicherplatz). Rechnungen damit werden dann jedoch sehr langsam. Meist ist die Anzahl der Stellen also begrenzt, weil nur eine gewisse Anzahl an Bits/Bytes für die Darstellung einer Zahl reserviert ist. 1 Byte = 8 Bits, kann Ganzzahlen zwischen 0 und 255, bzw zwischen -128 bis +127, darstellen.

$\underbrace{\pm}_{1-\text{Bit}} m * b^e$, $b = 2$ Basis $m = 1.\underbrace{m_1 \dots m_{52}}_{52-\text{Bits}}$, $e = \underbrace{c}_{11-\text{Bits}} - 1023$ (double-precision)
Menge aller Gleitkommazahlen =: A (endliche Menge)
$D := [x_{min}, x_{max}] \cup 0 \cup [x_{posmin}, x_{max}$ ist der "darstellbare Zahlenbereich"
Rundung bildet D auf A ab $rd : D \to A$, sodass $|x - rd(x)| = \min_{y \in A} |x - y|$
Rundung zur nächstliegenden Zahl.

IEEE : bei gleichweit entfernten Gleitkommazahlen nehme die, wo $m_{52} = 0$ ist.
Für eine Zahl $x = \pm m * 2^e$ mit $m \in [1, 2)$ ist der absolute Rundungsfehler:

$$|x - rd(x)| \leq \frac{1}{2} * 2^{-52} * 2^e$$

der relative Rundungsfehler

$$\frac{|x - rd(x)|}{|x|} \leq \frac{1}{2} * 2^{-52}$$

ist unabhabgig von der Größe von x.
Mit der "Maschinengenauigkeit" einer Gleitkommadarstellung bezeichnet man den Abstand zwischen 1 und der nächst größeren Gleitkommazahl. bei doppelt genauer Darstellung :

$$\text{"Epsilon" "Eps", "eps"} := 2^{-52} \approx 2,22 * 10^{16}$$

Es gilt immer : $rd(x) = rd(x(1 + \varepsilon))$ für alle $|\varepsilon| \leq \frac{eps}{2}$

Wichtig wird das Runden insbesondere auch bei den arithmetischen Operationen $+, -, *, \div$
Diese werden in Computern duch Maschinenoperationen ersetzt $(\oplus \ominus \otimes \oslash)$ bei denen das Ergebnis wieder eine Maschinenzahl ist.
Für jede Operation $* \in \{+, -, *, \div\}$ und $y, x \in A$
gilt :

$$x \otimes y \in A,\ x \otimes y = (x * y)(1 + \varepsilon) \text{ mit } |\varepsilon| \leq \frac{eps}{2}$$

Im allgemeinen gelten die typischen Geseze nicht, also:

i) $(x \otimes y) \oplus z$ ist nicht assoziativ

ii) $(x \otimes y) \otimes z$ ist nicht distributiv

iii) $x \oplus y = x$ falls $|y| \leq \frac{|x|}{2} eps$

mit iii) kann man eps durch ausprobieren berechnen. (noch was von foto abschreiben)

0.2 Kondition und Stabilität

Einfluss von Störungen oder Fehlern auf das Ergebnis einer mathematischen Aufgabe oder eines Berechnungsverfahrens.

Beispiel 0.2.1. Kleine Unterschiede von Werten können evtl. auf dem Rechner/ in der gewählten Zahlendarstellung gar nicht unterschieden werden. (Berechnungsverfahren)

Beispiel 0.2.2. Mathematische Aufgabe: Beispiel für lineares Gleichungssystems: $x: \ Ax = b$ mit $A = \begin{pmatrix} 1,2969 & 0,8648 \\ 0,2161 & 0,1441 \end{pmatrix}$ für $b = \begin{pmatrix} 0,8642 \\ 0,1220 \end{pmatrix}$ ist die Lösung $x = \begin{pmatrix} -2 \\ 2 \end{pmatrix}$. Für $b = \begin{pmatrix} 0,86419999 \\ 0,12200001 \end{pmatrix}$ ist die Lösung $x = \begin{pmatrix} 0,9911 \\ -0,487 \end{pmatrix}$

Definition 0.2.3. Eine mathematische Aufgabe heißt "schlecht konditioniert" wenn kleine Änderungen in den Daten große relative Fehler verursachen. Andernfalls heißt die Aufgabe "gut konditioniert"

Bemerkung 0.2.4. Eine gute Konditionierung deiner math. Aufgabe ist notwendig, um das Problem numerisch sinnvoll lösen zu können, da Rundungsfehler sonst große Fehler verursachen können.

Sei $\phi : X \to Y$ eine mathematische Aufgabe

Definition 0.2.5. Ein Verfahren oder Algorithmus zur (näherungsweisen) Lösung der math. Aufgabe ϕ ist eine Abbildung $\tilde{\phi} : X \to Y$,
die durch Hintereinanderschaltung endlich vieler (oder abzählbar unendlich vieler) elemetarer, möglicherweise rundungsfehlerbehafteter, Rechenoperation

$$\phi^{(k)}, \ k = 1, 2, 3, \ldots$$

definiert ist, also

$$\tilde{\phi} = \cdots \circ \phi^{(3)} \circ \phi^{(2)} \circ \phi^{(2)} \circ \phi^{(1)}$$

Bemerkung 0.2.6. typicherweise gibt es verschiedene Algorithmen für die gleiche math. Aufgabe ϕ. Von einem "guten" Algorithmus erwartet man, dass die im Verlauf des Algorithmen akkumulierten Fehler den durch die Kondition der math. Aufgabe unvermeidbaren Fahler nicht wesentlich übersteigen.

Definition 0.2.7. Ein Algorithmus $\tilde{\phi}$ heißt "instabil" , wenn es eine Störung $\tilde{x}$ von x gibt so dass der durch den Rundungsfehler und Störungen verursachte relative Fehler erheblich größer ist als der nur durch die Störung verursachte Fehler, d.h. falls $\phi(x) \neq =$ und $\dfrac{|\tilde{\phi}(\tilde{x}) - \phi(x)|}{|\phi(x)|} \gg \dfrac{|\phi(\tilde{x}) - \phi(x)|}{|\phi(x)|}$. Der Algorithmus heißt stabil, falls er nicht instabil ist. (ggf ales "bei x" oder "für kleine $|x|$", große $|x|$, o.ä.)

Beispiel 0.2.8. math. Aufgabe:

$$\phi(x) = \frac{1}{x(x+1)} \ , x \in \mathbb{R}\backslash 0, -1$$

Es gilt:

$$\frac{1}{x(x+1)} = \frac{1}{x} - \frac{1}{x+1}$$

Zwei mögliche verfahren:

$$\tilde{\phi}_1(x) = \frac{1}{x(x+1)}$$

ist stabil für $x \gg 1$

$$\tilde{\phi}_2(x) = \left(\frac{1}{x}\right) - \left(\frac{1}{(x+1)}\right)$$

ist instabil für $x \gg 1$ wegen Auslöschung der Differenzbildung.

0.3 Landau-Symbole, Genauigkeit und Komplexität

Beispiel 0.3.1.

(a) $A \in \mathcal{M}_n(\mathbb{R})$, n-Vektor $x \in \mathbb{R}^n$, $Ax = b \in \mathbb{R}^n$, $b_i = \sum_{j=1}^n A_{ij}x_j$
Rechnung von Ax : n^2 Matrixmultiplikationen, $n(n-1)$ Additionen nötig.
$\rightsquigarrow$ Rechenaufwand etwa quadratisch in der Dimension des Gleichungssystems. (bei voll besetzter Matrix)

(b) Genauigkeit der Differenzenquotienten zur Approximation der Ableitung:

$$\left| n'(x) \frac{n(x+h) - n(x)}{h} \right| \leq h \frac{1}{2} \underset{[x,x+h]}{max} |n''|$$

Fehler gleich Größenordnung wie Abstand h.

Definition 0.3.2. Seien $D \subset \mathbb{R}^n$, $f, g : D \to \mathbb{R}$, und $x, x_0 \in D$
Man sagt:

i) Die Funktion f wächst für $x \to x_0$ langsamer als g", geschrieben als: $f = \sigma(g)$ "f ist klein-o von g"

ii) Die Funktion "f wächst für $x \to x_0$ nicht wesentlich schneller als g", geschrieben als $f = \mathcal{O}(g)$, "f ist groß-$\mathcal{O}$ von g" wenn $\exists c > 0 \ \exists \varepsilon > 0$: $|f(x)| \leq c|g(x)| \ \forall x \in B_\varepsilon(x_0) |x - x_0| \leq \varepsilon$

iii) analoge Definition für $x \to \pm\infty$

Bemerkung 0.3.3. "Konditionierung schlecht" bzw. "Konditionierung instabil" ist nicht genau definiert.
Was einfacher ist: Aufgabenstellung bzw. Verfahren: Falls Fehlerverstärkung bei verfahren kleiner als bei anderen, dann ist das erste Verfahren "stabiler", bzw. eine aufgabe "besser konditioniert"

0.4 Differentielle Fehleranalyse:

Math. Aufgaben $\phi : X \to Y$. Ist ϕ (unendlich) differenzierbar, dann kann die Kondition auch mit Hilfe der Ableitungen von ϕ bestimmt/berechnet werden: Sei $\phi : \mathbb{R} \to \mathbb{R}$ eine Abbildung, $x \in \mathbb{R}^m,\ x = (x_1, ..., x_m)^T$

$$\phi(x) = (\phi_1(x_1, ...x_m), ..., \phi_m(x_1, ...x_m))^T$$

Die ϕ_i seinen alle zweimal stetig differenzierbar (partiell).
Damit gilt dann:

$$\begin{aligned}\Delta y_i &= \phi_i(x + \Delta x) - \phi(x) \\ &= \sum_{j=1}^{m} \frac{\partial \phi_i(x)}{\partial x_i} \Delta x_i + R_i(x, \Delta x) \text{ (Taylor)} \\ &= \sum_{j=1}^{m} \frac{\partial \phi_i(x)}{\partial x_i} \Delta x_i + R_i(x, \Delta x) + \mathcal{O}(|\Delta x|^2)\end{aligned}$$

Dann folgt für den relativen Fehler: $\frac{\Delta y}{|y|} = \frac{\Delta y}{|\phi(x)|}$

$\frac{\Delta y}{y_i} = \sum_{j=1}^{m} \frac{\partial \phi_j(x)}{\partial x_,} \frac{x_j}{\phi_i(x)} \frac{\Delta x_j}{x_j}$ Für $x_j \neq 0, y_i \neq 0$
$\frac{\Delta y}{y_i}$ Ist der relative Aufgabenfahler
$\frac{\partial \phi_j(x)}{\partial x_,} =: K_{ij}(x)$
$\frac{\Delta x_j}{x_j}$ ist der relative Datenfehler

Definition 0.4.1. Die $K_{ij}(x), i = 1, ..., n, j = 1, ..., m$ heißen "relative Konditionszahlen" von ϕ in x sie sind ein Maß dafür, wie sich kleine relative Fehler in den Eingangsdaten im ergebnis auswirken.

Die Aufgabe: $y = \phi(x)$ aus x zu berechnen, ist schlecht konditioniert, wenn es ein i, j gibt mit $|K_{ij}(x)| \gg 1$. Ansonsten ist ϕ gut konditioniert.

Beispiel 0.4.2. Grundoperation Addition: $\phi(x_1, x_2) = x_1 + x_2$

$$K_1 = \frac{\partial \phi}{\partial x_1}(x) \frac{x_1}{\phi(x)} = 1 * \frac{x_1}{x_1 + x_2} = \frac{1}{1 + \frac{x_2}{x_1}}$$

$$K_2 = \frac{\partial \phi}{\partial x_2}(x) \frac{x_2}{\phi(x)} = 1 * \frac{x_2}{x_1 + x_2} = \frac{1}{1 + \frac{x_1}{x_2}}$$

Für $\frac{x_1}{x_2} \approx -1$ werden die K_i sehr groß, dort ist die Addition schlecht konditioniert.
Das entspricht $x_1 \approx -x_2$, entspricht Subtraktion von 2 Zahlen, die fest gleich groß sind.
Bei Gleitkommazahlen: Übereinstimmung in den vorderen Mantissenstellen, dadurch Genauigkeit des Resultats geringer als der Daten.

Definition 0.4.3. Unter "Auslöschung" versteht man den Verlust an wesentlichen Dezimalstellen bei der Subtraktion von Zahlen gleichen Vorzeichens. Dies kann zu relativ großen Fehlern führen, falls eine oder beide Zahlen von operationen gerundet ($\Delta x \neq 0$) werden.

Bemerkung 0.4.4. Doese differenzielle Fehleranalyse kann analog für einen Komplexen Algorithmus $\tilde{\phi} = \phi^{(n-1)} \circ ... \circ \phi^{(1)}$, bestehend aus einfachen Rechenoperationen $\phi^{(i)}$, durchgeführt werden, Kettenregel führt auf Ableitung der Hintereinanderschaltungen. Für Komplexe Algorithmen aber nicht sehr Sinnvoll durchzuführen. Man kann stattdessen versuchen statistische Methoden anzuwenden, in denen z.B. Rundungsfehler durch zufalllsvariablen modelliert werden, um damit Wechselwirkungen abschätzen zu können.

Beispiel 0.4.5 (Rekursive Berechnung von Integralen)**.**
Aufgabe: Es sollen die folgenden Untegrale berechnet werden:

$$I_1 := \frac{1}{e} \int_0^1 x^n e^x dx, \ n = 0, 1, 2, ...$$

Berechnung mit integrationsformeln/numerische Integration: Später in Vorlesung.
mit partieller integration sieht man, dass

$$I_n = 1 - nI_{n-1}$$

eine Lösung ist. Für

$$n = 0 \Rightarrow I_0 = \frac{e-1}{e} \approx 0,632...$$

Numerische Berechnung: $I_0 = 0,632...$
$I_5 = 0,1455...$
$I_{10} = 0,0838...$
$I_{15} = 0,059...$
$I_{20} = -30, ...$
$I_{21} = 635,04$
$I_{22} = -13970, ...$
Man sieht leicht:

$$I_n > 0, \ I_n \leq \int_0^1 x^n dx = \frac{1}{n+1}$$

Warum diese Fehler?
In jedem Schritt der Rekursion wird der Fehler aus dem letzten schritt mit Faktor $-n$ multipliziert. Nach n schritten mit gesamtfaktor $(-n)^n * n!$
Die Fakultät wird schnell groß!