

Vorlesung 401-1652-10L : Numerische Mathematik I

Frühlingssemester 2014 : Prof. Ralf Hiptmair (SAM, D-MATH)

[Vorlesungsniederschrift, Rohformat, URL: <http://www.math.ethz.ch/education/bachelor/lectures/fs2014/math/nm>]

Numerik $\hat{=}$ Entwicklung und Analyse von Algorithmen zur (approximativen) Lösung von Problemen in $\mathbb{R}/\mathbb{C}$

Computer = endlicher Automat $\longleftrightarrow$ $\mathbb{R}$ unendlich/kontinuierlich
 $\hookrightarrow$ neue Aspekte : Numerik $\neq$ Analysis

Kapitel 1 : Computerarithmetik

Bsp.: Berechnung von e : $e = \lim_{n \rightarrow \infty} (1 + \frac{1}{n})^n$: $e_n := (1 + \frac{1}{n})^n = \exp(n \log(1 + \frac{1}{n})) = \exp(n (\frac{1}{n} + O(\frac{1}{n^2})))$
 $= \exp(1 + O(\frac{1}{n}))$ für $n \rightarrow \infty$
 $= e \cdot e^{O(\frac{1}{n})} = e (1 + O(\frac{1}{n}))$ ← relativer Fehler

$\Rightarrow$ Für grosses n liefert e_n eine gute Näherung für e

Notation: Landau-Symbol : $g(x) = O(f(x))$ für $x \rightarrow a$ $\iff \limsup_{x \rightarrow a} \frac{|g(x)|}{|f(x)|} < \infty$ " $|g(x)| \leq C |f(x)|$ für $x \approx a$ "

Begriff: Relativer Fehler: $\delta = \frac{|x - \hat{x}|}{|x|}$ $x \triangleq$ exakter Wert; $\hat{x} =$ Näherung; $\tilde{x} = x(1 + \delta)$, $|\delta| = \delta$

```

Editor - /Volumes/Scratch/Users/hiptmair/NOSAVE/numpde/numcourses/nummathl/skript_EN/skript/MATLAB/elimat.m
File Edit Text Go Cell Tools Debug Desktop Window Help
x [Icons] 1.0 + 1.1 x [Icons]
1 % Example 1.1: nummathI: Approximation of e
2 for i=1:15
3     n = 10^i; r = (1+1/n)^n;
4     fprintf('10^%2d %20.15f %20.15f\n', i, r, r-exp(1));
5 end

```

n	Näherung e_n	Fehler $e_n - e$
10^1	2.593742460100002	-0.124539368359044
10^2	2.704813829421529	-0.013467999037517
10^3	2.716923932235520	-0.001357896223525
10^4	2.718145926824356	-0.000135901634689
10^5	2.718268237197528	-0.000013591261517
10^6	2.718280469156428	-0.000001359302618
10^7	2.718281693980372	-0.000000134478673
10^8	2.718281786395798	-0.000000042063248
10^9	2.718282030814509	0.000000202355464
10^{10}	2.718282053234788	0.000000224775742
10^{11}	2.718282053357110	0.000000224898065
10^{12}	2.718523496037238	0.000241667578192
10^{13}	2.716110034086901	-0.002171794372145
10^{14}	2.716110034087023	-0.002171794372023
10^{15}	3.035035206549262	0.316753378090216

$O(\frac{1}{n})$

?

Bsp. 1.2:

(Abgeschnittene) Exponentialreihe

$$e^x = \sum_{k=0}^{\infty} \frac{1}{k!} x^k \approx \sum_{k=0}^n \frac{1}{k!} x^k =: S_n \quad \text{Taylor-Pol.}$$

Aufgabe: Algorithmus zur Auswertung von e^x mit rel. Fehler $\tau > 0$

Restgliedabschätzung: $|e^x - S_n(x)| \leq \left[f^{(n+1)}(\xi) \frac{|x-x_0|^{n+1}}{(n+1)!} \right] e^{\xi} \frac{|x|^{n+1}}{(n+1)!}$, $\min(x, 0) \leq \xi \leq \max(x, 0)$

$$\frac{|e^x - S_n(x)|}{e^x} \leq \begin{cases} \frac{|x|^{n+1}}{(n+1)!} & \text{für } x \geq 0 \\ \frac{|x|^{n+1}}{(n+1)!} e^{-x} & \text{für } x < 0 \end{cases} \quad [e^{\xi} \leq e^x !] \triangleright \text{Abbruchkriterium}$$

Beachte: $\triangleq$ Summenkern


```

1 function y = expeval(x,tol)
2 % nummath I: Ch. 1, Ex. 1.2
3 % Evaluation of exponential series
4 s=1; term=1; k=1; % Initialization
5 while (abs(term)>tol*abs(min(s,1))) %Termination
6     term = term*x/k;
7     s = s + term; % Summation
8     k = k+1;
9 end
10 y = s;

```

Handwritten notes in MATLAB editor:
 - Line 5: min(s,1) is underlined in orange.
 - Line 5: **%Termination** is written in green.
 - Line 6: $\rightarrow \text{term} = \frac{x^{k-1}}{(k-1)!}$ is written in blue.
 - A pink asterisk is next to line 5.

x	Approximation $\widetilde{\exp}(x)$	$\exp(x)$	$\frac{ \exp(x) - \widetilde{\exp}(x) }{\exp(x)}$
-20	5.6218844674e-09	2.0611536224e-09	1.727542676201181
-18	1.5385415977e-08	1.5229979745e-08	0.010205938187564
-16	1.1254180496e-07	1.1253517472e-07	0.000058917020257
-14	8.3152907681e-07	8.3152871910e-07	0.000000430176956
-12	6.1442133148e-06	6.1442123533e-06	0.000000156480737
-10	4.5399929556e-05	4.5399929762e-05	0.000000004544414
-8	3.3546262817e-04	3.3546262790e-04	0.000000000788902
-6	2.4787521758e-03	2.4787521767e-03	0.000000000333306
-4	1.8315638879e-02	1.8315638889e-02	0.000000000530694
-2	1.3533528320e-01	1.3533528324e-01	0.000000000273603
0	1.0000000000e+00	1.0000000000e+00	0.000000000000000
2	7.3890560954e+00	7.3890560989e+00	0.000000000479969
4	5.4598149928e+01	5.4598150033e+01	0.000000001923058
6	4.0342879295e+02	4.0342879349e+02	0.000000001344248
8	2.9809579808e+03	2.9809579870e+03	0.000000002102584
10	2.2026465748e+04	2.2026465795e+04	0.000000002143800
12	1.6275479114e+05	1.6275479142e+05	0.000000001723845
14	1.2026042798e+06	1.2026042842e+06	0.000000003634135
16	8.8861105010e+06	8.8861105205e+06	0.000000002197990
18	6.5659968911e+07	6.5659969137e+07	0.000000003450972
20	4.8516519307e+08	4.8516519541e+08	0.000000004828738

$\text{tol} = 10^{-8}$ 8 Stellen

Bsp 1.2.a : Summation der harmonischen Reihe

MATLAB-CODE: Summation vorwärts

```

s = 0;
for i = 1:1e6 → aufwärts
    s = s+1/i;
end

```

MATLAB-CODE: Summation rückwärts

```

s = 0;
for i = 1e6:-1:1 → abwärts
    s = s+1/i;
end

```

← ? →

„+“ assoziativ !?

➤ $s = 14.39272672286498889$

➤ $s = 14.39272672286577226$

[„Exakte“ Auswertung durch MAPLE-Code: ➤ $s = 14.39272672286572363138112749\dots$]

1.1. Gleitpunktzahlen ("wissenschaftliche Notation")

$$\hookrightarrow \underline{1.75} \cdot 10^6 \quad (\beta = 10)$$

Satz 1.3. [Wissenschaftliche Notation]

Jedes $x \in \mathbb{R} \setminus \{0\}$ lässt sich in der Form

$$x = \pm \beta^e \left(\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \frac{d_3}{\beta^3} + \dots \right)$$

darstellen; mit der **Basis** $2 \leq \beta \in \mathbb{N}$, den **Ziffern** (engl. "digits") $d_1, d_2, \dots, \in \{0, 1, 2, \dots, \beta - 1\}$, wobei $d_1 \neq 0$, und dem **Exponenten** $e \in \mathbb{Z}$.

1.2. Maschinenzahlen

Def. 1.5. [Maschinenzahlen $\mathbb{F}(\beta, t, e_{\min}, e_{\max})$]

Sei $\beta \in \mathbb{N}$, $\beta \geq 2$ (**Basis**) und $t \in \mathbb{N}$ (**Mantissenlänge**), sowie $e_{\min} < 0 < e_{\max}$ mit $e_{\min}, e_{\max} \in \mathbb{Z}$ (**Exponentenschranken**). Dann ist die Menge $\mathbb{F} = \mathbb{F}(\beta, t, e_{\min}, e_{\max})$ wie folgt definiert:

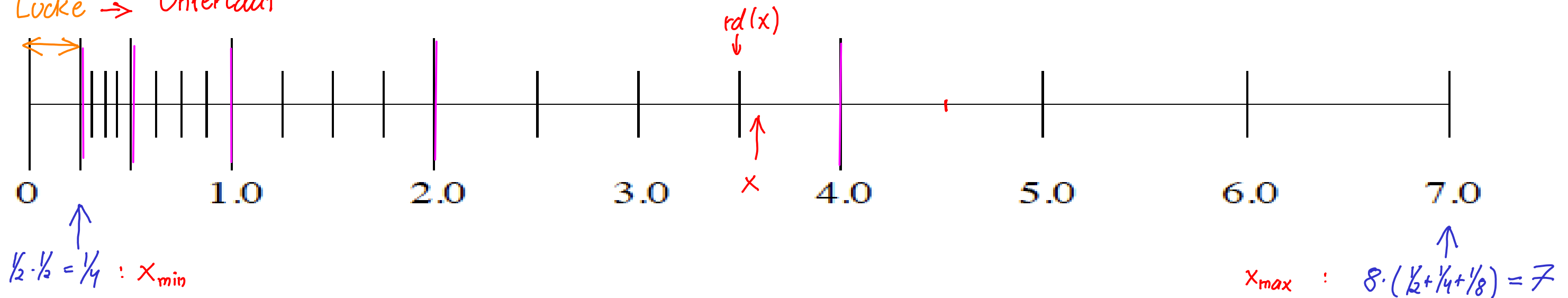
$$\mathbb{F} := \left\{ \pm \beta^e \left(\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \dots + \frac{d_t}{\beta^t} \right) : \begin{array}{l} d_1, \dots, d_t \in \{0, \dots, \beta - 1\}, \\ d_1 \neq 0, \\ e \in \mathbb{Z}, e_{\min} \leq e \leq e_{\max}. \end{array} \right\} \cup \{0\}.$$

↑
Mantisse $\hat{=}$ endliche Summe

▷ $\mathbb{F}$ ist eine **endliche** und **diskrete** Teilmenge von $\mathbb{R}$

Bsp: $\beta = 2, t = 3, e_{\min} = -1, e_{\max} = 3$

Lücke $\rightarrow$ Unterlauf



Lemma 1.6. [Kleinste und grösste Maschinenzahl]

In $\mathbb{F} = \mathbb{F}(\beta, t, e_{\min}, e_{\max})$ gilt

$$x_{\min}(\mathbb{F}) := \min\{x \in \mathbb{F} : x > 0\} = \beta^{e_{\min}-1},$$

$$x_{\max}(\mathbb{F}) := \max\{x \in \mathbb{F}\} = \beta^{e_{\max}}(1 - \beta^{-t}).$$

Lemma 1.7. [Relativer Abstand zwischen Maschinenzahlen]

Sei $x \in \mathbb{F} := \mathbb{F}(\beta, t, e_{\min}, e_{\max}) \setminus \{0\}$ und $y \in \mathbb{F}$ eine benachbarte Maschinenzahl. Dann gilt

$$\beta^{-1}\epsilon_M \leq \frac{|x - y|}{|x|} \leq \epsilon_M,$$

mit $\epsilon_M = \beta^{1-t}$.

$\hookrightarrow$ relativer Abstand: nur abhängig von β, t

IEEE - Standard : $\beta = 2$ (Dualsystem)

single precision (float) : $t = 23$, $e_{\min} = -127$, $e_{\max} = 128$: 4 Byte

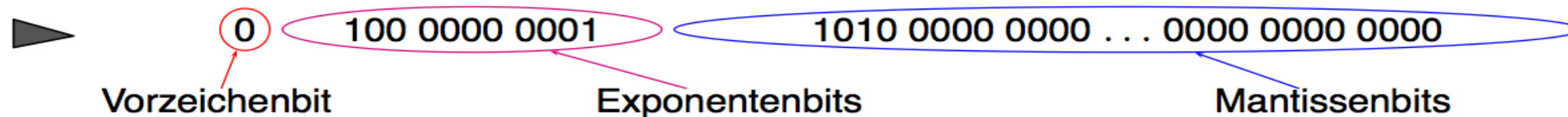
double precision : $t = 52$, $e_{\min} = -1023$, $e_{\max} = 1024$: 8 Byte [MATLAB]

+ Vorzeichenbit

```
>> format hex
```

```
>> 6.5
```

```
ans = 401a000000000000
```



$$x = (-1)^0 \cdot \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{16}\right) \times 2^{1026-1023} = 6.5.$$

```
>> format hex; realmin, format long; realmin
```

```
ans = 0010000000000000
```

```
ans = 2.225073858507201e-308
```

```
>> format hex; realmax, format long; realmax
```

```
ans = 7fefffffffffffffff
```

```
ans = 1.797693134862316e+308
```

```
>> x = exp(1000), y = 3/x, z = x*sin(pi), w = x*log(1)
```

```
x = Inf → Überlauf
```

```
y = 0
```

```
z = Inf
```

```
w = NaN → Abbruch
```



1.3. Runden

$$rd : \text{Dom}(\mathbb{F}) := \{x \in \mathbb{R} : x_{\min} \leq |x| \leq x_{\max}\} \cup \{0\} \longrightarrow \mathbb{F}$$

$$rd(x) \in \operatorname{argmin} \{ |y-x| : y \in \mathbb{F} \}$$

[$\# \operatorname{argmin} \{ \} = 2 \Rightarrow rd(x)$ so dass d_t gerade, $d_t \neq$ letzte Mantissenstelle]

Satz 1.10. [Schranke für relativen Rundungsfehler]

Für alle $x \in \text{Dom}(\mathbb{F})$ existiert (ein von x abhängiges) $\delta \in \mathbb{R}$ derart, so dass

$$rd(x) = x(1 + \delta), \quad |\delta| \leq u,$$

wobei $u = u(\mathbb{F}) := \frac{1}{2}\beta^{1-t}$ die **Rundungseinheit** (engl. "unit roundoff") des Maschinenzahlensystems $\mathbb{F}$ bezeichnet.

$\delta \leftrightarrow$ relativer Rundungsfehler

Notation: $\lceil \cdot \rceil, \lfloor \cdot \rfloor \triangleq$ "nächst grössere/kleinere ganze Zahl"

IEEE-Std: $u \approx 2 \cdot 10^{-16}$ ['eps' in MATLAB]

Beweis: O.B.d.A: $x > 0$

$$x = p\beta^{e-t}, \beta^{t-1} \leq p \leq \beta^t, p \in \mathbb{R}$$

Nächstgelegene Maschinenzahlen

$$\begin{aligned} y_- &:= \lfloor p \rfloor \beta^{e-t} \\ y_+ &:= \lceil p \rceil \beta^{e-t} \end{aligned} \in \mathbb{F}$$

$$\text{Lemma 1.7: } |y_+ - y_-| \leq \epsilon_m |y_-|$$

$$\underbrace{|rd(x) - x|}_{\in \{y_-, y_+\}} \leq \frac{1}{2} |y_+ - y_-| \leq \frac{1}{2} \beta^{1-t} \underbrace{|y_-|}_{\leq |x|} \quad \square$$

Maschinenoperationen: " $\circ \in \{+, -, *, /\}$ " : $\circ : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \cup \{\infty\}$ "Analysis"

aber $\circ : \mathbb{F} \times \mathbb{F} \not\rightarrow \mathbb{F}$

$\Rightarrow$ Notwendig: Rechner ersetzt $\circ$ durch $\tilde{\circ} : \mathbb{F} \times \mathbb{F} \rightarrow \mathbb{F}$: **Modell** : $x \tilde{\circ} y := rd(x \circ y)$

Unter- und Überlauf, wenn $x \circ y \notin \text{Dom}(\mathbb{F})$

Annahme 1.13. [“Axiom” der Computerarithmetik]

Für $\circ \in \{+, -, *, /\}$ und für alle $x, y \in \mathbb{F}$, die $x \circ y \in \text{Dom}(\mathbb{F})$ erfüllen, gilt

$$\exists \delta \in \mathbb{R}, \quad |\delta| \leq u(\mathbb{F}) : \quad x \tilde{\circ} y = (x \circ y)(1 + \delta) .$$

→ 1.13. impliziert nicht Assoziativität / Kommutativität

Annahme (1.8) [Zusatz zum “Axiom” der Computerarithmetik]

Für die Computerimplementierung $\tilde{f}$ von “elementaren Funktionen”

$f \in \{\sin, \cos, \exp, \cosh, \tan, \dots\}$ und für alle $x \in \mathbb{F}$, die $f(x) \in \text{Dom}(\mathbb{F})$ erfüllen, gilt

$$\exists \delta \in \mathbb{R}, \quad |\delta| \leq u(\mathbb{F}) : \quad \tilde{f}(x) = f(x)(1 + \delta) .$$

1.4. Rundungsfehleranalyse

Zu Bsp 1.1.:

$$e_n := (1 + \frac{1}{n})^n \approx e$$

↑ Fokus auf diesem Rundungsfehler

Mit Annahme 1.13 für '+' (& Potenzierung)

Berechnete Wert: $\tilde{e}_n = [(1 + \frac{1}{n})(1 + \delta)]^n (1 + \delta')$ mit $|\delta|, |\delta'| \leq \mu := \mu(\mathbb{F})$

$$= \exp(n(\log(1 + \frac{1}{n}) + \log(1 + \delta)))$$

$$= \exp(n(\frac{1}{n} + O(\frac{1}{n^2}) + \delta + O(\delta^2))) \quad \text{für } n \rightarrow \infty, \delta \rightarrow 0$$

$$= e \cdot \exp(O(\frac{1}{n}) + n \cdot \delta + n \cdot O(\delta^2))$$

↑
Approximationsfehler

↑
Fortgepflanzte Rundungsfehler

$$\log(1+x) = x + O(x^2) \quad \text{für } x \rightarrow 0$$

Wenn $n \approx \delta^{-1} \Rightarrow$ keine gültigen Stellen

$$|\log(1+x) - x| \leq C|x|^2 \quad \text{für } |x| \text{ hinreichend klein}$$

n	Näherung e_n	Fehler $e_n - e$
10^1	2.593742460100002	-0.124539368359044
10^2	2.704813829421529	-0.013467999037517
10^3	2.716923932235520	-0.001357896223525
10^4	2.718145926824356	-0.000135901634689
10^5	2.718268237197528	-0.000013591261517
10^6	2.718280469156428	-0.000001359302618
10^7	2.718281693980372	-0.000000134478673
10^8	2.718281786395798	-0.000000042063248
10^9	2.718282030814509	0.000000202355464
10^{10}	2.718282053234788	0.000000224775742
10^{11}	2.718282053357110	0.000000224898065
10^{12}	2.718523496037238	0.000241667578192
10^{13}	2.716110034086901	-0.002171794372145
10^{14}	2.716110034087023	-0.002171794372023
10^{15}	3.035035206549262	0.316753378090216

← 7 Stellen

Beispiel: Euklidisches Skalarprodukt $x^T y$

Algorithmus:

```
function s = mydot(x,y)
% Euclidean inner product
s = 0; for j=1:length(x), s = s+x(j)*y(j); end
```

[Warnung! Schlechter MATLAB-Stil!]

Notation ∇
 $\delta \triangleq$ "generischer relativer Rundungsfehler"
 $\hookrightarrow$ verschiedene Werte
 erfüllen (vgl. Annahme 1.13) $|\delta| \leq u(\mathbb{F})$

$\triangleright$
 $(x, y \in \mathbb{C}^n)$

$$\tilde{S}_n = x_1 y_1 \prod_{i=1}^n (1 + \delta_i) + x_2 y_2 \prod_{i=1}^n (1 + \delta_i) + x_3 y_3 \prod_{i=1}^{n-1} (1 + \delta_i) + \dots + x_n y_n (1 + \delta_n)$$

Lemma 1.15. [Abschätzung für Produkte von Fehlertermen]

Für $|\delta_i| \leq u(\mathbb{F})$, $i = 1, \dots, n$ und $n < 1/u(\mathbb{F})$ existiert ein $\theta_n \in \mathbb{R}$ derart, dass

$$\prod_{i=1}^n (1 + \delta_i) = 1 + \theta_n, \quad \text{wobei} \quad |\theta_n| \leq \frac{nu(\mathbb{F})}{1 - nu(\mathbb{F})} =: \gamma_n(\mathbb{F}).$$

$$\triangleright \tilde{S}_n = x_1 y_1 (1 + \theta_n) + x_2 y_2 (1 + \theta'_n) + x_3 y_3 (1 + \theta_{n-1}) + \dots + x_n y_n (1 + \theta_2)$$

mit $|\theta_i|, |\theta'_i| \leq \gamma_n(\mathbb{F})$

$$\triangleright \tilde{S}_n = [x_1 (1 + \theta_n)] y_1 + [x_2 (1 + \theta'_n)] y_2 + \dots + [x_n (1 + \theta_2)] y_n =: \tilde{x}^T y \quad (*)$$

$$S_i := \sum_{j=1}^i x_j y_j$$

$$\tilde{S}_1 = x_1 y_1 (1 + \delta)$$

$$\begin{aligned} \tilde{S}_2 &= (\tilde{S}_1 + x_2 y_2 (1 + \delta)) (1 + \delta) \\ &= x_1 y_1 (1 + \delta)(1 + \delta) + x_2 y_2 (1 + \delta)(1 + \delta) \end{aligned}$$

$$\tilde{S}_3 = x_1 y_1 (1 + \delta)(1 + \delta)(1 + \delta) + x_2 y_2 (1 + \delta)(1 + \delta)(1 + \delta) + x_3 y_3 (1 + \delta)(1 + \delta)$$

Beweis durch Induktion:

$$\prod_{i=1}^n (1 + \delta_i) = (1 + \theta_{n-1})(1 + \delta_n) =: 1 + \theta_n$$

$$\Rightarrow \theta_n = \delta_n + \theta_{n-1} (1 + \delta_n)$$

$$\begin{aligned} \Rightarrow |\theta_n| &\leq u + \frac{(n-1)u}{1 - (n-1)u} (1 + u) = \\ &= \frac{u - (n-1)u^2 + (n-1)u + (n-1)u^2}{1 - (n-1)u} \\ &\leq \gamma_n(\mathbb{F}) \quad \square \end{aligned}$$

Notation: $\underline{v} \in \mathbb{C}^n$: $|\underline{v}| := (|v_i|)_{i=1}^n \in \mathbb{R}^n$
 (Nicht zu verwechseln mit $\|\underline{v}\| \triangleq \text{Norm}$)

mit $\tilde{\underline{x}} = \underline{x} + \Delta \underline{x}$
 $|\Delta \underline{x}| \leq \gamma_n(\mathbb{F}) |\underline{x}|$
 ↑
 Komponentenweise Ungleichung

Terminologie: $\Delta \underline{x} \triangleq$ Rückwärtsfehler des Skalarprodukts unter Rundungsfehler einfluss

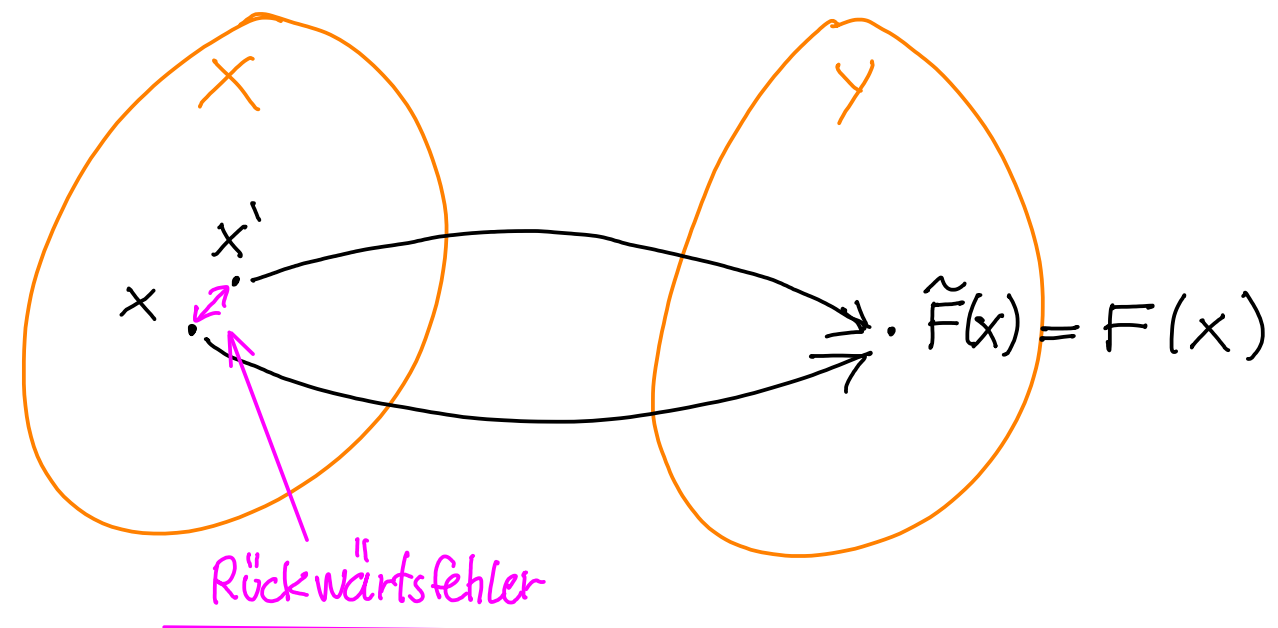
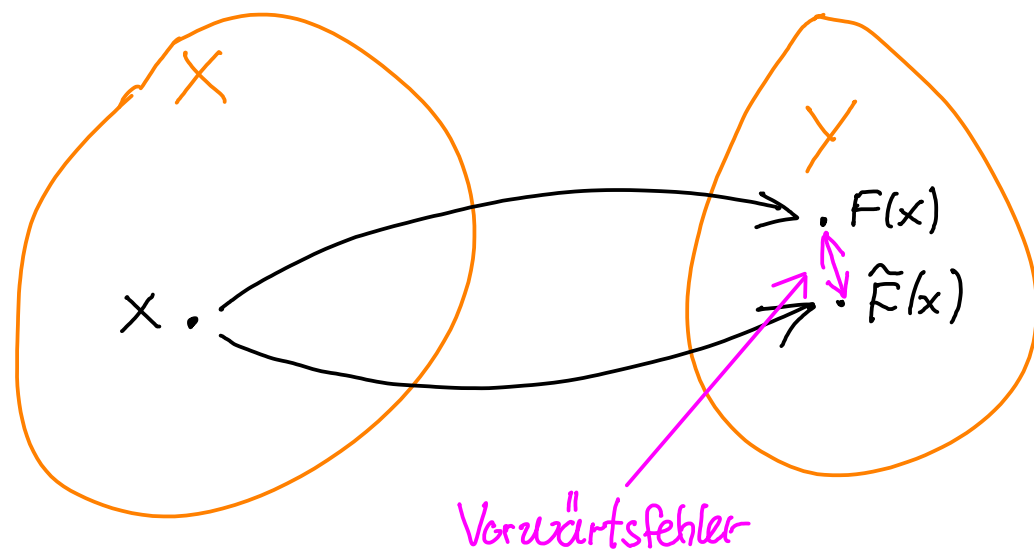
(Absoluter) Vorwärtsfehler: $s_n - \tilde{s}_n$

↳ Abschätzung aus (*) $|s_n - \tilde{s}_n| \leq \gamma_n(\mathbb{F}) \sum_{j=1}^n |x_j| |y_j|$

$$[|s_n - \tilde{s}_n| \leq \|\Delta \underline{x}\|_2 \|\underline{y}\|_2 \leq \gamma_n(\mathbb{F}) \|\underline{x}\|_2 \|\underline{y}\|_2]$$

Bemerkung: Auszuwerten $y = F(x)$, $F: X \longrightarrow Y$
 ↑ ↑
 Datenraum Ergebnisraum (mit Metrik)

$F \longrightarrow \tilde{F} \triangleq$ algorithmische Realisierung von F (unter Rundungsfehler einfluss), $\tilde{F}: X \longrightarrow Y$



"Def.": $\tilde{F}$ heisst **numerisch stabil**, wenn zu jedem $x \in X$ ein $x' \in X$ so existiert, dass

(i) $\hat{F}(x) = F(x')$

(ii) $d_x(x, x') \leq n \cdot v(F)$, wobei $n \triangleq$ Anzahl der elementaren Operationen in $\hat{F}$ bezeichnet

Bsp: Matrix * Vektor: $y = Ax$, $A \in \mathbb{C}^{n,n}$, $x \in \mathbb{C}^n$

```
function y = matvec(A,x)
% Basic matrix-vector multiplication
[m,n] = size(A); y = zeros(m,1);
for i=1:m, y(i) = mydot(A(i,:),x); end
```

▷ Reduktion auf Skalarprodukte:

Aus (*) : $\tilde{y} = (A + \Delta A)x$, $|\Delta A| \leq \gamma_n(F) |A|$
 $\uparrow$ berechnet von 'matvec' $\uparrow$ komponentenweise Ungl.

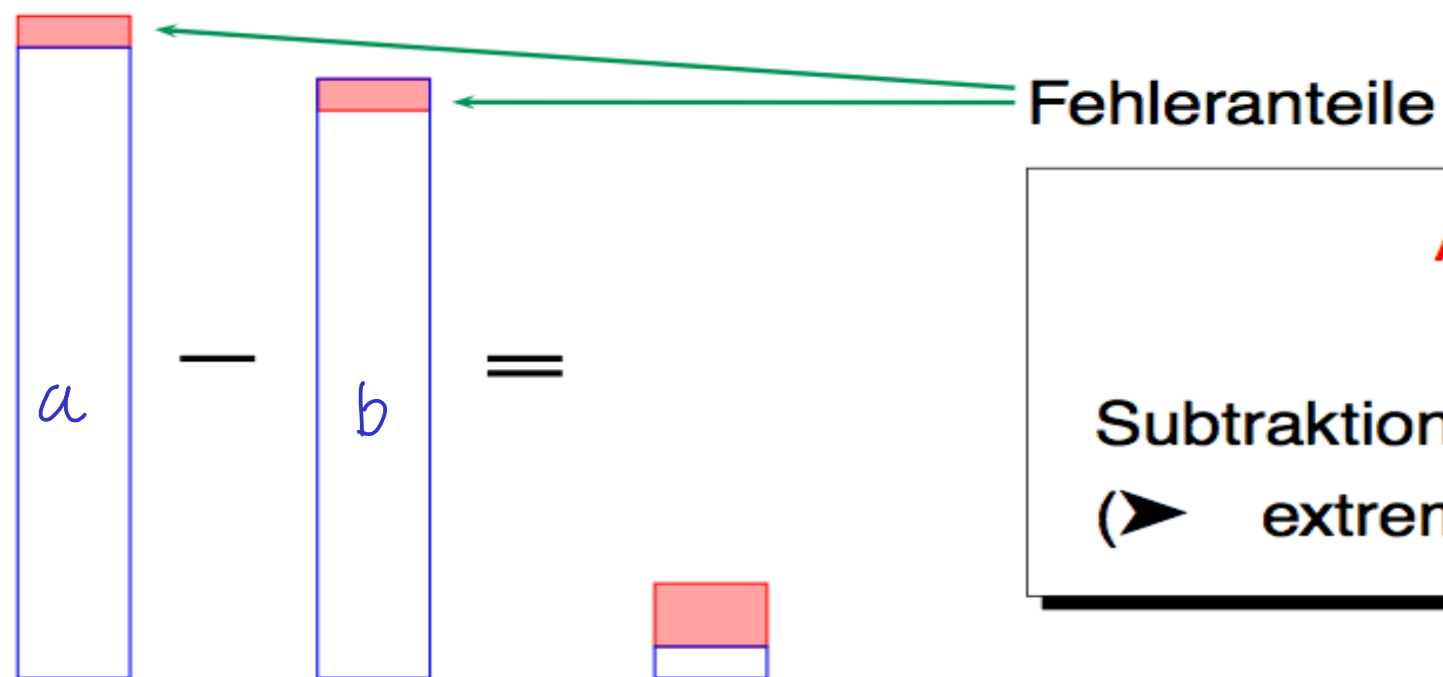
1.5. Auslöschung

$\tilde{a} = a(1 + \delta_a)$, $\tilde{b} = b(1 + \delta_b)$: $\tilde{z} := \tilde{a} - \tilde{b} = a - b + a\delta_a - b\delta_b$; $z := a - b$
 $|\delta_a|, |\delta_b| \leq \mu$

Relativer Fehler: $\frac{|\tilde{z} - z|}{|z|} = \frac{|a\delta_a - b\delta_b|}{|a - b|} \leq \frac{|a| + |b|}{|a - b|} \mu$

↳ sehr gross für $a \approx b$

▷ Gewaltige **Fehlerverstärkung** für $a \approx b$



Auslöschung (engl. *cancellation*)
 $\hat{=}$
 Subtraktion zweier ungefähr gleich grosser Zahlen
 (► extreme Verstärkung relativer Fehler,)

Subtraktionen mit Auslöschung geschehen selbst rundungsfehlerfrei !

Bsp : Vermeidung von Auslöschung $f(a,b) := a^2 - b^2 = (a+b)(a-b)$

Subtraktion von Zahlen mit R.F. In 2-stelliger Dezimalarithmetik, $a=1.3, b=1.2$

Alg. A : $x := a^2, y := b^2, z := x - y$

$\tilde{x} = 1.7, \tilde{y} = 1.4, \hat{z} = 0.3$ falsch !

Alg. B : $x := a+b, y := a-b, z := x * y$

$\tilde{x} = 2.5, \tilde{y} = 0.1, \hat{z} = 0.25$ ✓

Subtraktion von ungestörten Daten

Rundungsfehleranalyse: A : $\tilde{x} = a^2(1+\delta), \tilde{y} = b^2(1+\delta), \hat{z} = (a^2(1+\delta) - b^2(1+\delta))(1+\delta)$

$[z := a^2 - b^2]$ $\Rightarrow \frac{|z - \hat{z}|}{|z|} \leq \underbrace{1 + \left(u + \frac{a^2 + b^2}{a^2 - b^2} u(1+u) \right)}_{\text{rel. Fehler}} = a^2 - b^2 + (a^2\delta - b^2\delta)(1+\delta) + (a^2 - b^2)\delta$
sehr gross für $a^2 \approx b^2$

$$B: \quad \tilde{x} = (a+b)(1+\delta), \quad \tilde{y} = (a-b)(1+\delta) \Rightarrow \tilde{z} = (a^2-b^2) \underbrace{(1+\delta)(1+\delta)(1+\delta)}_{\substack{\text{Lemma 1.15} \\ 1+\theta_3, |\theta_3| \leq \gamma_3(F)}}$$

$$\triangleright \quad \frac{|\tilde{z} - \hat{z}|}{|\tilde{z}|} = 1 + \theta_3$$