

**Inhaltsverzeichnis**

<b>I Mathematik 4</b>	<b>1</b>	<b>9 Funktionentheorie</b>	<b>8</b>
<b>1 Wiederholung</b>	<b>1</b>	9.1 Komplexe Differenzierbarkeit . . . . .	8
1.1 Matrixnormen . . . . .	1	9.1.1 Cauchy-Riemannsche DGL . . . . .	8
1.2 Wichtige Reihen . . . . .	1	9.1.2 Folgerungen . . . . .	8
<b>2 Grundbegriffe der Numerik</b>	<b>2</b>	9.2 Komplexe Integration . . . . .	8
2.1 Gleitkomma-Arithmetik . . . . .	2	9.2.1 Kurvenintegrale . . . . .	8
2.2 Kondition . . . . .	2	9.2.2 Cauchy-Integralformel . . . . .	8
2.3 Stabilität . . . . .	3	9.3 Potenzreihendarstellung analytischer Funktionen . . . . .	8
<b>3 Numerische lineare Algebra</b>	<b>3</b>	9.4 Laurent-Reihen und Singularitäten . . . . .	8
3.1 Dreiecksmatrizen . . . . .	3	9.5 Residuenkalkül . . . . .	9
3.1.1 Vorwärts- / Rückwärtssubstitution . . . . .	3		
3.2 Nachiteration . . . . .	3		
3.3 LR/LU-Zerlegung . . . . .	3		
3.4 LRP/LUP-Zerlegung . . . . .	4		
3.5 Cholesky-Zerlegung . . . . .	4		
3.5.1 Rekursiver Algorithmus zur Berechnung von $R$ . . . . .	4		
3.6 Householder-Spiegelung . . . . .	4		
3.7 QR-Zerlegung . . . . .	4		
3.7.1 QR-Algorithmus . . . . .	4		
<b>4 Ausgleichsprobleme</b>	<b>5</b>		
4.1 Lineares Ausgleichsproblem . . . . .	5		
4.2 Lösen mit Normalengleichung . . . . .	5		
4.3 Lösen mit $QR$ -Zerlegung . . . . .	5		
<b>5 Fixpunktiteration</b>	<b>5</b>		
5.1 Kontraktion . . . . .	5		
5.2 Banach'scher Fixpunktsatz . . . . .	6		
5.3 Konvergenzgeschwindigkeit . . . . .	6		
<b>6 Iterative Löser für LGS</b>	<b>6</b>		
6.1 Stationäre Fixpunktiteration . . . . .	6		
6.1.1 Dämpfung . . . . .	6		
6.2 Jacobi-Verfahren . . . . .	6		
6.2.1 Konvergenz . . . . .	6		
6.3 Gauß-Seidel-Verfahren . . . . .	6		
6.3.1 Konvergenz . . . . .	6		
<b>7 Nichtlineare Gleichungen</b>	<b>6</b>		
7.1 Lösbarkeit . . . . .	6		
7.2 Kondition . . . . .	6		
7.3 Bisektionsverfahren . . . . .	6		
7.3.1 Algorithmus . . . . .	6		
7.4 Newton-Verfahren . . . . .	7		
7.4.1 Mehrdimensional . . . . .	7		
<b>8 Optimierung</b>	<b>7</b>		
8.1 Optimalitätsbedingungen . . . . .	7		
8.2 Newton-Verfahren . . . . .	7		
8.3 Abstiegsverfahren . . . . .	7		
8.3.1 Algorithmus . . . . .	7		
8.3.2 Gradientenverfahren . . . . .	7		
8.4 Globalisiertes Newton-Verfahren . . . . .	7		

# Teil I Mathematik 4

## 1 Wiederholung

### 1.1 Matrixnormen

Sei  $\|\cdot\|$  eine Norm auf  $\mathbb{R}^n, n \geq 1$ , dann definiert

$$\| \|A\| \| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|$$

für  $A \in \mathbb{R}^{n \times n}$  eine Norm.  
Außerdem gilt:

1) Submultiplikativität

$$\| \|AB\| \| \leq \| \|A\| \| \cdot \| \|B\| \|$$

2) Verträglichkeit

$$\| \|Ax\| \| \leq \| \|A\| \| \cdot \| \|x\| \|$$

Die von der **Maximumsnorm**  $\|\cdot\|_\infty$  induzierte Matrizen-norm ist die "maximale Zeilensumme".

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

Die von der  **$l_1$ -Norm**  $\|\cdot\|_1$  induzierte Matrizen-norm ist die "maximale Spaltensumme":

$$\| \|A\| \|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$$

Sei  $A$  eine symmetrische Matrix, dann gilt:

$$\| \|A\| \|_2 = |\lambda_{max}| = \{ |\lambda| \mid \lambda \in \mathbb{R} \text{ ist Eigenwert von } A \}$$

Für beliebige  $\mathbb{R}^{n \times n}$ -Matrizen gilt:

$$\| \|A\| \|_2 = \sqrt{\lambda_{max}(A^T A)} = \rho(A^T A)^{\frac{1}{2}} \quad (\text{Spektralradius})$$

Die Eigenwerte von  $A$  und  $A^{-1}$  sind zueinander invers.

$$\Rightarrow \| \|A^{-1}\| \|_2 = \frac{1}{\sqrt{\lambda_{min}(A^T A)}}$$

### 1.2 Wichtige Reihen

$$\ln(1+x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k+1} x^{k+1} = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots, \quad x \in (-1, 1]$$

$$\sin(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$

$$\cos(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots$$

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

Für alternierende Reihen  $\sum_{k=0}^{\infty} (-1)^k a_k$  mit  $a_k \leq 0$  und monoton fallend (z.B.  $\ln, \sin, \cos$ ) gilt nach dem **Leibniz-kriterium**:

$$\underbrace{\left| \sum_{k=0}^{\infty} (-1)^k a_k - \sum_{k=0}^N (-1)^k a_k \right|}_{\text{Restglied}} \leq a_{N+1}$$

Für die Exponentialreihe gilt:

$$\underbrace{\left| e^x - \sum_{k=0}^N \frac{x^k}{k!} \right|}_{\text{Restglied}} \leq 2 \cdot \frac{|x|^{N+1}}{(N+1)!} \quad \text{für } |x| \leq 1$$

## 2 Grundbegriffe der Numerik

### 2.1 Gleitkomma-Arithmetik

Es werden Zahlen der Form

$$a = (-1)^s \cdot m \cdot b^e$$

betrachtet, mit  $b \in \mathbb{N}, b \geq 2, s \in \{0, 1\}$  und

$$m = \sum_{i=0}^{t-1} m_i b^{-i}; \quad m_i \in \{0, \dots, b-1\}$$

$$e = \sum_{i=0}^{r-1} e_i b^i - \underbrace{(b^{r-1} - 1)}_{\text{bias}}; \quad e_i \in \{0, \dots, b-1\}$$

wobei  $t \geq 2$  und  $r \geq 2$  die Genauigkeiten darstellen.

Nachfolgend sei  $b = 2$ .

IEEE "single":

32 Bit ( $s$ : 1 Bit,  $e$ : 8 Bit,  $m$ : 23 Bit)

IEEE "double":

64 Bit ( $s$ : 1 Bit,  $e$ : 11 Bit,  $m$ : 52 Bit)

Definitionen:

- $a \in \mathbb{R}$  heißt **normierte Gleitkommazahl** ( $a \in \mathbb{M}_{t,r}^N$ ), falls eine Darstellung existiert mit  $m_0 = 1$  und  $e_{\min} < e < e_{\max}$ , wobei gelte:

$$e_{\min} = -b^{r-1} + 1$$

$$e_{\max} = b^{r-1}$$

- $a \in \mathbb{R}$  heißt **subnormale Gleitkommazahl** ( $a \in \mathbb{M}_{t,r}^S$ ), falls  $e = e_{\min}$  und  $m_0 = 0$ , wobei nicht alle  $m_i$  Null sind.

$$\Rightarrow a = (-1)^s \left( \sum_{i=1}^{t-1} m_i 2^{-i} \right) 2^{e_{\min}+1}$$

- Sonderfälle

- "0":  $\forall i: m_i = 0, e_i = 0$

- " $\infty$ ":  $\forall i: m_i = 0, e_i = 1$

- "NaN":  $\forall i: e_i = 1$ , wobei nicht alle  $m_i$  Null sind.

- $\mathbb{M}_{t,r} = \mathbb{M}_{t,r}^N \cup \mathbb{M}_{t,r}^S \cup \{\pm\infty\} \cup \{NaN\}$

- Der Abstand von 1 zur nächstgrößeren Zahl in  $\mathbb{M}_{t,r}$  ist:

$$\text{eps} = 2^{-(t-1)} \quad (\text{Maschinengenauigkeit})$$

- Rundung:  $\text{rd}_{t,r} : \mathbb{R} \rightarrow \mathbb{M}_{t,r}^N$

Für die IEEE-754-Standardrundung gilt:

$$\frac{|x - \text{rd}_{t,r}|}{|x|} \leq \frac{\text{eps}_t}{2} \quad \text{für } x \neq 0$$

Für alle  $x, y \in \mathbb{M}_{t,r}^N$  und  $*$   $\in \{+, -, \cdot, /$  gilt:

$$x \otimes y = \text{rd}_{t,r}(x * y)$$

## 2.2 Kondition

Seien  $x, y$  normierte Vektorräume (mit Norm  $\|\cdot\|$ ).

Ein Problem ist eine Funktion  $f : x \rightarrow y$ .

Gesucht ist  $\delta f = f(x + \delta x) - f(x)$  im Vergleich zu  $\delta x$ .

Die **absolute Kondition** von  $f$  im Punkt  $x \in X$  ist

$$\kappa_{\text{abs}}^f(x) = \lim_{\delta \rightarrow 0} \sup_{\|\delta x\| < \delta} \frac{\|\delta f\|}{\|\delta x\|}$$

Sei  $f : \mathbb{R} \rightarrow \mathbb{R}$  differenzierbar (mit  $\|\cdot\| = |\cdot|$ ), dann gilt:

$$\begin{aligned} \kappa_{\text{abs}}(x) &= |f'(x)| \\ \Rightarrow \|\delta f\| &\leq \kappa_{\text{abs}} \cdot \|\delta x\| \end{aligned}$$

Für die **relative Kondition** gilt:

$$\kappa_{\text{rel}}(x) = \lim_{\delta \rightarrow 0} \sup_{\|\delta x\| < \delta} \frac{\frac{\|\delta f\|}{\|f(x)\|}}{\frac{\|\delta x\|}{\|x\|}} = \lim_{\delta \rightarrow 0} \sup_{\|\delta x\| < \delta} \frac{\|\delta f\| \cdot \|x\|}{\|f(x)\| \cdot \|\delta x\|}$$

Sei  $f : \mathbb{R} \rightarrow \mathbb{R}$  differenzierbar, dann gilt:

$$\begin{aligned} \kappa_{\text{rel}} &= \frac{|f'(x)|}{\frac{|f(x)|}{|x|}} = \frac{|f'(x)| \cdot |x|}{|f(x)|} \\ \Rightarrow \frac{\|\delta f\|}{\|f\|} &\leq \kappa_{\text{rel}} \cdot \frac{\|\delta x\|}{\|x\|} \end{aligned}$$

Sei  $A \in \mathbb{R}^{n \times n}$ ,  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ ,  $b \rightarrow f(b) = A^{-1}b = x$ , dann gilt:

$$\begin{aligned} \kappa_{\text{abs}}(b) &= \|f'(b)\| = \|A^{-1}\| \\ \kappa_{\text{rel}}(b) &= \frac{\|A^{-1}\|}{\frac{\|A^{-1}b\|}{\|b\|}} = \|A^{-1}\| \frac{\|Ax\|}{\|x\|} \leq \|A^{-1}\| \cdot \|A\| \end{aligned}$$

Für die Kondition bzgl.  $\|\cdot\|_2$  gilt außerdem:

$$\kappa_{\text{rel}}(A) = \sqrt{\frac{\lambda_{\max}(A^T A)}{\lambda_{\min}(A^T A)}}$$

$\mathcal{N.B.}$ :

$$\frac{\|f(\tilde{x}) - f(x)\|}{\|f(x)\|} \leq \kappa_{\text{rel}} \cdot \frac{\|\tilde{x} - x\|}{\|x\|} + O\left(\frac{\|\tilde{x} - x\|^2}{\|x\|^2}\right)$$

## 2.3 Stabilität

Definitionen:

- Ein **Algorithmus** ist eine Abb.  $\tilde{f} : x \rightarrow y$  mit  $\tilde{f} = \tilde{f}_k \circ \dots \circ \tilde{f}_1$ , wobei jeder Teilschritt  $\tilde{f}_i, i = 1, \dots, k$  nur Operationen aus  $\{\oplus, \ominus, \odot, \oslash, \text{rd}\}$  enthält.
- Ein Algorithmus  $\tilde{f}$  für ein Problem  $f$  heißt **rückwärtsstabil**, falls  $\forall x \in X \exists \tilde{x} \in X$  mit

$$\underbrace{\tilde{f}(x)}_{(1+\epsilon)f(x)} = f(\tilde{x}) \quad \text{und} \quad \frac{\|x - \tilde{x}\|}{\|x\|} = O(\epsilon) \quad \text{für } \epsilon \rightarrow 0$$

wobei  $\epsilon$  den relativen Fehler darstellt.

**Ziel:**

Finde zu geg.  $f$  ein  $\tilde{f}$ , sodass der Fehler  $\|\tilde{f}(x) - f(x)\|$  bzw.  $\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|}$  möglichst "klein" ist.

Sei  $\tilde{f}$  ein rückwärtsstabiler Algorithmus für  $f$ , dann gilt:

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} \leq O(\kappa_{\text{rel}}(x) \cdot \text{eps}) \quad \text{für } \text{eps} \rightarrow 0$$

## 3 Numerische lineare Algebra

### 3.1 Dreiecksmatrizen

**Satz:**

Falls  $A = LR$  mit  $L \in \mathbb{R}^{n \times n}$  untere und  $R \in \mathbb{R}^{n \times n}$  obere Dreiecksmatrix,  $\det(A) \neq 0$ , so ist das LGS  $Ax = b$  für alle  $b \in \mathbb{R}^n$  in  $2n^2$  Operationen lösbar.

$\mathcal{N.B.}$

$$\sum_{k=1}^n (ak - b) = \frac{1}{2}n(an + a - 2b)$$

#### 3.1.1 Vorwärts- / Rückwärtssubstitution

Falls  $A$  obere oder untere Dreiecksmatrix, so kann das LGS  $Ax = b$  durch Vorwärts- bzw. Rückwärtssubstitution gelöst werden.

Matlab-Code für Vorwärtssubstitution:

```
for j = 1 : n
    x(j) = (b(j) - A(j, 1 : j - 1) * x(1 : j - 1)) / A(j, j);
end
```

Matlab-Code für Rückwärtssubstitution:

```
for j = n : -1 : 1
    x(j) = (b(j) - A(j, j + 1 : n) * x(j + 1 : n)) / A(j, j);
end
```

Aufwand: je  $n^2$  Flops

### 3.2 Nachiteration

Mit Hilfe der Nachiteration kann eine Näherungslösung  $\tilde{x}$  eines LGS  $Ax = b$  genauer approximiert werden.

**Vorgehensweise:**

1) Berechne  $\tilde{x}$ :

$$A\tilde{x} = b$$

2) Berechne  $r$ :

$$r = b - A\tilde{x}$$

3) Berechne  $z$ :

$$Az = r$$

4) Die genauere Näherungslösung  $\hat{x}$  lautet:

$$\hat{x} = \tilde{x} + z$$

### 3.3 LR/LU-Zerlegung

Sei  $A \in \mathbb{R}^{n \times n}, \det(A) \neq 0, a_{ii} \neq 0, L \in \mathbb{R}^{n \times n}$  untere und  $R \in \mathbb{R}^{n \times n}$  obere Dreiecksmatrix.

Dann kann das LGS  $Ax = b$  wie folgt gelöst werden:

1) Berechne  $L$  und  $R$ , sodass  $A = LR$

2) Berechne  $y$ , sodass  $Ly = b$  (für LRP-Zerlegung:  $Ly = Pb$ )

3) Berechne  $x$ , sodass  $Rx = y$

Für die Berechnung von  $L$  und  $R$  wird folgender Algorithmus angewendet:

1) Setze  $l_{i1} = \frac{a_{i1}}{a_{11}}$  für  $i \geq 1$ .

2) Setze  $a_{ij} = a_{ij} - l_{i1} \cdot a_{1j}$  für alle  $j$  und  $i > 1$ :

$$\text{Zeile } i = \text{Zeile } i - l_{i1} \cdot \text{Zeile } 1, \quad i > 1$$

3) Gehe zu Schritt 1 und ersetze alle 1er durch 2,3,4,...

Beispiel:

2	3	1	5
6	13	5	19
2	19	10	23
4	10	11	31

2	3	1	5
3	4	2	4
1	16	9	18
2	4	9	21

2	3	1	5
3	4	2	4
1	4	1	2
2	1	7	17

2	3	1	5
3	4	2	4
1	4	1	2
2	1	7	3

1. Iteration      2. Iteration      3. Iteration

$$\Rightarrow \underbrace{\begin{pmatrix} 2 & 3 & 1 & 5 \\ 6 & 13 & 5 & 19 \\ 2 & 19 & 10 & 23 \\ 4 & 10 & 11 & 31 \end{pmatrix}}_A = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 2 & 1 & 7 & 1 \end{pmatrix}}_L \underbrace{\begin{pmatrix} 2 & 3 & 1 & 5 \\ 0 & 4 & 2 & 4 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 \end{pmatrix}}_R$$

Aufwand für Zerlegung:  $\frac{2}{3}n^3 - \frac{1}{2}n^2 - \frac{1}{6}n$  Flops

Das oben genannte Verfahren ist ein direktes/exaktes Lösungsverfahren (keine Verfahrensfehler; nur Rundungsfehler)

Matlab-Code:

```
[n,~] = size(A);
for k = 1:n
    I = k+1:n;
    A(I,k) = A(I,k)/A(k,k);
    A(I,I) = A(I,I) - A(I,k)*A(k,I);
end
```

### 3.4 LRP/LUP-Zerlegung

Die LRP-Zerlegung ist eine LU-Zerlegung mit Pivoting-Strategie:

Suche  $j \geq k$ , sodass  $|a_{jk}^{(k)}| \geq |a_{ik}^{(k)}| \forall i \geq k$  und vertausche die  $j$ -te mit der  $k$ -ten Zeile in  $A^{(k)}$ .

Für  $\det(A) \neq 0$  existiert eine Permutationsmatrix  $P \in \mathbb{R}^{n \times n}$ , sodass gilt:

$$PA = LR$$

**Definition:**

Eine Matrix heißt **Permutationsmatrix**, falls sie in jeder Zeile und jeder Spalte jeweils eine 1 enthält und sonst nur 0.

Beispiel:

1	2	0	2	0,6
2	3	3	4	-2
3	5	5	4	2
4	-1	-2	3,4	-1

3	5	5	4	2
2	3	3	4	-2
1	2	0	2	0,6
4	-1	-2	3,4	-1

3	5	5	4	2
2	0,6	0	1,6	-3,2
1	0,4	-2	0,4	-0,2
4	-0,2	-1	4,2	-0,6

1. Iteration

3	5	5	4	2
2	0,6	0	1,6	-3,2
1	0,4	-2	0,4	-0,2
4	-0,2	-1	4,2	-0,6

3	5	5	4	2
1	0,4	-2	0,4	-0,2
2	0,6	0	1,6	-3,2
4	-0,2	-1	4,2	-0,6

3	5	5	4	2
1	0,4	-2	0,4	-0,2
2	0,6	0	1,6	-3,2
4	-0,2	0,5	4	-0,5

2. Iteration

3	5	5	4	2
1	0,4	-2	0,4	-0,2
2	0,6	0	1,6	-3,2
4	-0,2	0,5	4	-0,5

3	5	5	4	2
1	0,4	-2	0,4	-0,2
4	-0,2	0,5	4	-0,5
2	0,6	0	1,6	-3,2

3	5	5	4	2
1	0,4	-2	0,4	-0,2
4	-0,2	0,5	4	-0,5
2	0,6	0	0,4	-3

3. Iteration

$$\Rightarrow \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}}_P \underbrace{\begin{pmatrix} 2 & 0 & 2 & 0,6 \\ 3 & 3 & 4 & -2 \\ 5 & 5 & 4 & 2 \\ -1 & -2 & 3,4 & -1 \end{pmatrix}}_A = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0,4 & 1 & 0 & 0 \\ -0,2 & 0,5 & 1 & 0 \\ 0,6 & 0 & 0,4 & 1 \end{pmatrix}}_L \underbrace{\begin{pmatrix} 5 & 5 & 4 & 2 \\ 0 & -2 & 0,4 & -0,2 \\ 0 & 0 & 4 & -0,5 \\ 0 & 0 & 0 & -3 \end{pmatrix}}_R$$

Matlab-Code:

```
[n,~] = size(A);
p = 1:n;
for k = 1:n
    [~,j] = max(abs(A(p(k:n),k)));
    j = j + (k-1);
    p([k,j]) = p([j,k]);
    I = k+1:n;
    A(p(I),k) = A(p(I),k)/A(p(k),k);
    A(p(I),I) = A(p(I),I) - A(p(I),k)*A(p(k),I);
end
```

### 3.5 Cholesky-Zerlegung

Sei  $A \in \mathbb{R}^{n \times n}$  symmetrisch und positiv definit, dann existiert eine rechte obere Dreiecksmatrix  $R$  mit

$$A = R^T R$$

**3.5.1 Rekursiver Algorithmus zur Berechnung von R**

- 1) Ersetze  $a_{11}$  durch  $\sqrt{a_{11}}$
- 2) Ersetze alle Einträge unter  $a_{11}$  durch 0
- 3) Ersetze alle Einträge rechts von  $a_{11}$  durch  $\frac{a_{1i}}{a_{11}}$
- 4) Beende die Funktion, falls  $n == 1$
- 5) Um den inneren Teil der Matrix zu berechnen ( $A(2 : n, 2 : n)$ ), rufe die Funktion mit folgender Matrix auf:

$$A(2 : n, 2 : n) - A(1, 2 : n)^T \cdot A(1, 2 : n)$$

Die entstandene Matrix stellt dann  $R$  dar.

Aufwand für Zerlegung:  $\frac{1}{3}n^3$  Flops

**3.6 Householder-Spiegelung**

Die Spiegelung  $\tilde{x}$  von  $x \in \mathbb{R}^k$  an der Hyperebene  $E_V$  senkrecht zu  $v \in \mathbb{R}^k$  heißt Householder-Spiegelung. Es gilt:

$$\tilde{x} = \underbrace{\left( I_n - \frac{2vv^T}{\|v\|^2} \right)}_{H_v \in \mathbb{R}^{m \times m}} \cdot x$$

Sei  $v \neq 0$ , dann gilt:

- $H_v^T = H_v, H_v H_v = I \Rightarrow H_v$  orthogonal
- $H_{\alpha v} = H_v \forall \alpha \in \mathbb{R} \setminus \{0\}$
- $H_v a = \mp \|a\| e_1$ , falls  $v = a \pm \|a\| e_1$

**3.7 QR-Zerlegung**

Sei  $A \in \mathbb{R}^{m \times n}, m \geq n, \text{Rang}(A) = n, Q \in \mathbb{R}^{m \times m}$  orthogonale Matrix,  $R \in \mathbb{R}^{m \times n}$  rechte obere Dreiecksmatrix mit

$$A = QR$$

**3.7.1 QR-Algorithmus**

Achtung: Falls ein LGS  $Ax = b$  gelöst werden soll, dann muss  $b$  genauso wie  $A$  transformiert werden.

Vorab: Setze  $A^{(1)} = A, p = \min\{m - 1, n\}$  und  $k = 1$ .

- 1) Setze  $\tilde{A} = A^{(k)}(k : m, k : n)$  und  $a = 1$ . Spalte von  $\tilde{A}$
- 2) Berechne  $v = a + \text{sign}(a_1) \cdot \|a\| \cdot e_1 \in \mathbb{R}^{m-k+1}$
- 3) Berechne Update von  $\tilde{A}$ :

$$\tilde{A} \leftarrow \underbrace{\left( I - \frac{2vv^T}{\|v\|^2} \right)}_{H_v} \cdot \tilde{A} = \tilde{A} - \underbrace{\frac{2v}{\|v\|^2} \cdot (v^T \cdot \tilde{A})}_{\text{so in Matlab}}$$

- 4) Setze  $Q_k = \begin{pmatrix} I_{k-1} & 0 \\ 0 & H_v \end{pmatrix}$  (falls  $Q$  explizit berechnet werden soll)
- 5) Setze  $A^{(k+1)} = Q_k \cdot A^{(k)}$  (ersetze  $\tilde{A}$  in  $A^{(k)}$  mit Update aus 3.)
- 6) Erhöhe  $k$  und gehe zu Schritt 1

$$\Rightarrow R = A^{(p+1)} = Q^T A$$

$$Q = Q_1 \cdot \dots \cdot Q_p, Q^T = Q_p \cdot \dots \cdot Q_1, \text{ da } Q_k = Q_k^T$$

Falls  $m = n$ , so ist das LGS  $Ax = b$  wie folgt lösbar:

$$\underbrace{Q^T A}_{R=A^{(p+1)}} \cdot x = \underbrace{Q^T b}_{b^{(p+1)}}$$

$$\Rightarrow Rx = b^{(p+1)}$$

Aufwand für Zerlegung:

$$\frac{4}{3}n^3 \text{ Flops, falls } m = n$$

$$2n^2 \left( m - \frac{1}{3}n \right) \text{ Flops, falls } m \gg n$$

Matlab-Code (QR\_householder):

```
function A = QR_householder(A)
[m,n] = size(A);
for k = 1:min(m-1,n)
    v = A(k:end,k);
    na = norm(v);
    if v(1) >= 0
        s = 1;
    else
        s = -1;
    end
    v(1) = v(1) + s*na;
    v = [1; v(2:end)/v(1)];
    A(k:end,k+1:end) = A(k:end,k+1:end)
    -(2/(v'*v)*v)*(v'*A(k:end,k+1:end));
    A(k,k) = -s*na;
    A(k+1:end,k) = v(2:end);
end
```

Matlab-Code (Qmult\_householder):

```
function B = Qmult_householder(A,B)
[m,n] = size(A);
for k = 1:min(m-1,n)
    v = [1; A(k+1:end,k)];
    B(k:end,:) = B(k:end,:) - (2/(v'*v)*v)*(v'*B(k:end,:));
end
```

Matlab-Code (LGS Ax = b lösen):

```
A = QR_householder(A);
x = rsub(triu(A), Qmult_householder(A,b));
```

**4 Ausgleichsprobleme**

Seien Messpunkte  $t_i, b_i \in \mathbb{R} \times \mathbb{R}, i = 1, \dots, m$  gegeben. Ziel: Bestimme die  $n$  Parameter  $x_1$  bis  $x_n$ , sodass gilt:

$$b_i \approx \varphi(t_i, x), \quad i = 1, \dots, m$$

Falls  $\varphi$  linear in  $x$  ist, so ist  $x$  die Lösung eines überbestimmten GLS:

$$\begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} = b \approx Ax = \begin{pmatrix} a(t_1)^T \\ \vdots \\ a(t_m)^T \end{pmatrix} x$$

### 4.1 Lineares Ausgleichsproblem

Bestimme  $x \in \mathbb{R}^n$  zu  $A \in \mathbb{R}^{m \times n}$  und  $b \in \mathbb{R}^m$ , sodass gilt:

$$\|b - Ax\|_2^2 = \min$$

### 4.2 Lösen mit Normalengleichung

$$A^T Ax = A^T b$$

Für ein Polynom  $b = f(t) = x_0 + x_1 t + \dots + x_n t^n$  als Ausgleichskurve gilt:

$$A = \begin{pmatrix} 1 & t_1 & t_1^2 & \dots & t_1^n \\ 1 & t_2 & t_2^2 & \dots & t_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_m & t_m^2 & \dots & t_m^n \end{pmatrix}$$

### 4.3 Lösen mit QR-Zerlegung

Sei eine QR-Zerlegung gegeben mit

$$Q^T b = b^{(p+1)} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \quad R = Q^T A = A^{(p+1)} = \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix}$$

mit  $\tilde{R} \in \mathbb{R}^{n \times n}$ ,  $b_1 \in \mathbb{R}^n$ ,  $b_2 \in \mathbb{R}^{m-n}$ .

Dann gilt für das Minimierungsproblem:

$$\begin{aligned} \|b - Ax\|^2 &= \|Q^T(b - Ax)\|^2 = \|Q^T b - Q^T \underbrace{QR}_A x\|^2 \\ &= \left\| \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} - \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix} x \right\|^2 = \|b_1 - \tilde{R}x\|^2 + \|b_2\|^2 \end{aligned}$$

Somit folgt

$$\|b_1 - \tilde{R}x\|^2 + \|b_2\|^2 = \min \Leftrightarrow \tilde{R}x = b_1$$

## 5 Fixpunktiteration

Ziel: Gesucht wird ein Fixpunkt  $x^* \in X$  der Abbildung  $\varphi : X \rightarrow X$ :

$$\varphi(x^*) = x^*$$

- Ein FP  $x^*$  heißt stabil, falls  $\|\varphi'(x^*)\| < 1$
- Ein FP  $x^*$  heißt instabil/abstoßend, falls  $\|\varphi'(x^*)\| > 1$

### 5.1 Kontraktion

Eine (Selbst-)Abbildung  $\varphi : X \rightarrow X$  heißt Kontraktion, falls es ein festes  $0 \leq L < 1$  gibt mit:

$$\|\varphi(x) - \varphi(y)\| \leq L \cdot \|x - y\|, \quad \forall x, y \in X \quad (\text{Lipschitz-stetig})$$

Ist  $\varphi : X \rightarrow X$  stetig differenzierbar, so gilt:

$$L = \sup_{x \in X} \|\varphi'(x)\| = \sup_{x \in X} \|J_\varphi(x)\|$$

### 5.2 Banach'scher Fixpunktsatz

Sei  $X$  abgeschlossen und  $\varphi : X \rightarrow X$  (Selbstabbildung) eine Kontraktion. Dann besitzt  $\varphi$  einen eindeutigen Fixpunkt  $x^* \in X$  und die Iteration

$$x_{k+1} = \varphi(x_k)$$

konvergiert für jeden Startwert  $x_0 \in X$  gegen  $x^*$ .

Außerdem gelten folgende Abschätzungen:

$$\underbrace{|x_n - x^*| \leq \frac{L^n}{1-L} |x_1 - x_0|}_{\text{A-Priori}} \quad \text{und} \quad \underbrace{|x_n - x^*| \leq \frac{L}{1-L} |x_n - x_{n-1}|}_{\text{A-Posteriori}}$$

### 5.3 Konvergenzgeschwindigkeit

Die Konvergenz der Folge  $(x_k)$  ist von der Ordnung  $p$ , wenn gilt:

$$\|x_{k+1} - x^*\| \leq C \|x_k - x^*\|^p \quad \forall k = 0, 1, \dots$$

Falls  $\varphi : X \rightarrow X$  zweimal stetig diffbar ist mit FP  $x^*$  und  $\varphi'(x^*) = 0$ , dann konvergiert die Iteration  $x_{k+1} = \varphi(x_k)$  mindestens lokal quadratisch.

## 6 Iterative Löser für LGS

### 6.1 Stationäre Fixpunktiteration

Idee: Zerlege  $A$  additiv in  $A = B - C$  mit  $\det(B) \neq 0$ , dann gilt:

$$\begin{aligned} Ax = b &\Leftrightarrow Bx = b + Cx \\ &\Leftrightarrow x = B^{-1}b + B^{-1}Cx = \varphi(x) \\ \Rightarrow x^{(k+1)} &= \varphi(x^{(k)}) = B^{-1}b + B^{-1}Cx^{(k)} \\ &= x^{(k)} + B^{-1}(b - Ax^{(k)}) \\ \varphi(x) &= Nb + Mx^{(k)} \end{aligned}$$

Sei  $M \in \mathbb{R}^{n \times n}$ . Dann ist  $\|M\| < 1$  für eine Operatornorm  $\|\cdot\|$  hinreichend für die Konvergenz des Iteration  $\varphi(x) = Nb + Mx$ .

#### 6.1.1 Dämpfung

Durch Dämpfung der FP-Iteration mit  $\omega \in (0, 1]$  können die verschiedenen Verfahren noch beschleunigt werden:

$$\begin{aligned} x^{(k+1)} &= \omega \cdot (Nb + Mx^{(k)}) + (1 - \omega) \cdot x^{(k)} \\ &= \omega \cdot Nb + [\omega \cdot M + (1 - \omega) \cdot I] \cdot x^{(k)} \end{aligned}$$

Falls  $M \in \mathbb{R}^{n \times n}$  nur reelle Eigenwerte  $\lambda_1 \leq \dots \leq \lambda_n < 1$  besitzt, dann gilt

$$\omega_{\text{opt}} = \frac{2}{2 - \lambda_1 - \lambda_n}$$

### 6.2 Jacobi-Verfahren

Zerlege  $A$  additiv in eine Diagonalmatrix  $D$  und obere/untere Dreiecksmatrizen  $R$  und  $L$ :

$$A = D - (L + R)$$

Dann lautet das Jacobi-Verfahren:

$$x^{(k+1)} = \underbrace{D^{-1}}_N b + \underbrace{D^{-1}(L + R)}_M \cdot x^{(k)} = D^{-1} \cdot [b + (L + R) \cdot x^{(k)}]$$

### 6.2.1 Konvergenz

Das Jacobi-Verfahren konvergiert, falls eine der beiden folgenden Bedingungen erfüllt ist:

- $A$  ist strikt diagonaldominant:

$$|a_{ii}| > \sum_{i \neq j} |a_{ij}|, \quad \forall i = 1, \dots, n$$

- Für den Spektralradius gilt:

$$\rho(D^{-1}(L+R)) = \rho(M) = \max\{|\lambda_i(M)| \mid i = 1, \dots, n\} < 1$$

### 6.3 Gauß-Seidel-Verfahren

Zerlege  $A$  additiv in eine Diagonalmatrix  $D$  und obere/untere Dreiecksmatrizen  $R$  und  $L$ :

$$A = (D - L) - R$$

Dann lautet das Gauß-Seidel-Verfahren:

$$\begin{aligned} x^{(k+1)} &= \underbrace{(D - L)^{-1} b}_N + \underbrace{(D - L)^{-1} R}_{M} \cdot x^{(k)} \\ &= (D - L)^{-1} \cdot (b + R x^{(k)}) \end{aligned}$$

Aufwand:  $2n^2$  Flops

#### 6.3.1 Konvergenz

Das Gauß-Seidel-Verfahren konvergiert, falls eine der folgenden Bedingungen erfüllt ist:

- $A$  ist strikt diagonaldominant
- $A$  ist symmetrisch und positiv definit
- Für den Spektralradius gilt:

$$\rho((D - L)^{-1} R) = \rho(M) = \max\{|\lambda_i(M)| \mid i = 1, \dots, n\} < 1$$

## 7 Nichtlineare Gleichungen

### 7.1 Lösbarkeit

Sei  $f : X \rightarrow \mathbb{R}^n$  stetig differenzierbar und  $x^* \in X$  mit  $f(x^*) = 0$ . Falls  $\det(f'(x^*)) \neq 0$ , so ist  $x^*$  lokal eindeutig, d.h.  $\exists$  Umgebung  $U$  von  $x^*$ , sodass  $\tilde{x} \in U, f(\tilde{x}) = 0 \Rightarrow \tilde{x} = x^*$ .

### 7.2 Kondition

Für die absolute Kondition des Problems  $P : f \rightarrow x^*$  bzgl. einer Norm  $\|\cdot\|$  gilt:

$$\kappa = \left\| (f'(x^*))^{-1} \right\|$$

### 7.3 Bisektionsverfahren

#### 7.3.1 Algorithmus

$$x_0 = \frac{1}{2}(a_0 + b_0)$$

Für  $k = 0, 1, \dots$  {

STOP, falls  $f(x_k) = 0$

$a_{k+1} = a_k, b_{k+1} = x_k$ , falls  $f(a_k)f(x_k) < 0$

$a_{k+1} = x_k, b_{k+1} = b_k$ , falls  $f(x_k)f(b_k) < 0$

$x_{k+1} = \frac{1}{2}(a_{k+1} + b_{k+1})$

STOP, falls  $|a_{k+1} - b_{k+1}| < 2 \cdot TOL$

}

Globale Konvergenz:

$$|x_k - x^*| \leq \frac{1}{2} |I_k| = \frac{1}{2^{k+1}} |b_0 - a_0|$$

### 7.4 Newton-Verfahren

Sei  $f : X \rightarrow \mathbb{R}$  zweimal stetig differenzierbar,  $f(x^*) = 0, f'(x^*) \neq 0$ , dann konvergiert das Newton-Verfahren

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

lokal quadratisch.

#### 7.4.1 Mehrdimensional

Sei  $f : X \rightarrow \mathbb{R}^n, X \subset \mathbb{R}^n$  zweimal stetig differenzierbar,  $f(x^*) = 0$ , dann gilt:

$$x_{k+1} = x_k - J_f^{-1}(x_k) f(x_k)$$

## 8 Optimierung

### 8.1 Optimalitätsbedingungen

Definitionen:

- $C^m(x) = \{f : X \rightarrow \mathbb{R} \mid f \text{ } m\text{-mal stetig differenzierbar}\}$
- Sei  $f \in C^1(x)$  und  $\nabla f(x) = 0$  für ein  $x \in X$ . Dann heißt  $x$  **stationärer Punkt**.
- Sei  $X \subset \mathbb{R}^m$  offen,  $f \in C^1(x)$ ,  $x^*$  lokales Minimum. Dann ist  $x^*$  stationärer Punkt.
- Sei  $X \subset \mathbb{R}^n$  offen,  $f \in C^2(x)$ ,  $x^*$  stationärer Punkt mit  $H_f(x^*) = \nabla^2 f(x^*)$  positiv definit, dann ist  $x^*$  ein **lokales Minimum**. (positive Definitheit ist hinreichend, aber nicht unbedingt notwendig)
- Sei  $X \subset \mathbb{R}^n$  offen,  $f \in C^2(x)$ ,  $x^*$  stationärer Punkt mit  $H_f(x^*) = \nabla^2 f(x^*)$  negativ definit, dann ist  $x^*$  ein **lokales Maximum**. (negative Definitheit ist hinreichend, aber nicht unbedingt notwendig)
- Sei  $X \subset \mathbb{R}^n$  offen,  $f \in C^2(x)$ ,  $x^*$  stationärer Punkt mit  $H_f(x^*) = \nabla^2 f(x^*)$  indefinit, dann ist  $x^*$  ein **Sattelpunkt**.

### 8.2 Newton-Verfahren

Sei  $f \in C^2(x)$ . Gesucht sei die Lösung von  $\nabla f(x) = 0$ .

$$\begin{aligned} x_{k+1} &= x_k - H_f(x_k)^{-1} \nabla f(x_k) \\ H_f(x) &= \begin{pmatrix} \partial_{x_1 x_1} f(x) & \dots & \partial_{x_1 x_n} f(x) \\ \vdots & \ddots & \vdots \\ \partial_{x_n x_1} f(x) & \dots & \partial_{x_n x_n} f(x) \end{pmatrix} \end{aligned}$$

Das Iterationsverfahren besitzt lokal quadratische Konvergenz zu einem stationären Punkt  $x^*$  von  $f$ , falls  $H(x^*)$  invertierbar ist.

### 8.3 Abstiegsverfahren

#### Definition:

- $d \in \mathbb{R}^n$  heißt **Abstiegsrichtung** von  $f$  an der Stelle  $x$ , falls  $\exists \delta > 0$ , sodass gilt:

$$f(x + sd) < f(x) \quad \forall s \in (0, \delta]$$

#### 8.3.1 Algorithmus

Wähle  $x^{(0)}$

Für  $k = 0, 1, \dots$  {

STOP, falls  $\nabla f(x^{(k)}) \approx 0$

Bestimme Abstiegsrichtung  $d^{(k)}$  für  $f$  in  $x^{(k)}$

Bestimme Schrittweite  $s_k > 0$  mit  $f(x^{(k)} + s_k d^{(k)}) < f(x^{(k)})$

Setze  $x^{(k+1)} = x^{(k)} + s_k d^{(k)}$

}

#### 8.3.2 Gradientenverfahren

Wähle im obigen Algorithmus:

$$d^{(k)} = -\nabla f(x^{(k)})$$

#### Bestimmung der Schrittweite (2 Möglichkeiten):

1) "Exakte Schrittweite":

$$\min_{\delta > 0} f(x^{(k)} + s d^{(k)})$$

2) "Armijo-Schrittweite":

Wähle Parameter  $\sigma \in (0, \frac{1}{2})$

Setze  $s = 1$

Für  $l = 1, 2, \dots$  {

Falls  $f(x^{(k)} + s d^{(k)}) - f(x^{(k)}) \leq \sigma s \cdot \nabla f(x^{(k)})^T d^{(k)}$ :

Akzeptiere Schrittweite:  $s_k = s$

Sonst:  $s \rightarrow \frac{s}{2}$

}

Dieses Verfahren konvergiert gegen einen stationären Punkt von  $f$  oder es erzeugt eine Folge  $(x_{k \in \mathbb{N}}^{(k)})$ , für die gilt:

- $f(x^{(k+1)}) < f(x^{(k)}) \quad \forall k$  und
- alle Häufungspunkte sind stat. Punkte von  $f$

### 8.4 Globalisiertes Newton-Verfahren

Wähle  $x^{(0)} \in \mathbb{R}^n$  und  $\sigma \in (0, \frac{1}{2})$ ,  $\rho > 0$ .

Für  $k = 0, 1, \dots$  {

STOP, falls  $\nabla f(x^{(k)}) \approx 0$

Löse  $H_f(x^{(k)}) \tilde{d}^{(k)} = -\nabla f(x^{(k)})$ , falls möglich

Falls Newton-Gleichung nicht lösbar oder

$\nabla f(x^{(k)}) \tilde{d}^{(k)} > -\rho \|\nabla f(x^{(k)})\|^2$

$d^{(k)} = -\nabla f(x^{(k)})$

Sonst:  $d^{(k)} = \tilde{d}^{(k)}$

Bestimme Armijo-Schrittweite  $s_k$  (siehe oben)

Setze  $x^{(k+1)} = x^{(k)} + s_k d^{(k)}$

}

## 9 Funktionentheorie

### 9.1 Komplexe Differenzierbarkeit

Eine Funktion  $f : U \rightarrow \mathbb{C}$  heißt in  $z_0$  komplex differenzierbar, falls der Grenzwert

$$\lim_{z \rightarrow z_0} \frac{f(z) - f(z_0)}{z - z_0} = f'(z_0)$$

existiert.

- Ist  $f$  auf ganz  $U$  komplex differenzierbar, so heißt  $f$  analytisch.
- Kompositionen, Summen, Differenzen, Produkte und Quotienten analytischer Funktionen sind wieder analytisch.
- Potenzreihen sind analytisch in ihrem Konvergenzbereich

Wir schreiben  $f(z) = f(x + iy) = F_1(x, y) + i \cdot F_2(x, y)$  und setzen:

$$F(x, y) = \begin{pmatrix} F_1(x, y) \\ F_2(x, y) \end{pmatrix}$$

#### 9.1.1 Cauchy-Riemannsche DGL

$f$  ist genau dann in  $z = x + iy$  komplex differenzierbar, falls  $F$  in  $(x, y)$  differenzierbar ist und folgende DGL erfüllt sind:

$$\partial_x F_1(x, y) = \partial_y F_2(x, y)$$

$$\partial_y F_1(x, y) = -\partial_x F_2(x, y)$$

#### 9.1.2 Folgerungen

Sei  $f : U \rightarrow \mathbb{C}$  analytisch und das zugehörige reelle  $F : U \rightarrow \mathbb{R}^2$  2-mal stetig differenzierbar, dann gilt:

$$\Delta F_1(x) = \partial_1^2 F_1(x) + \partial_2^2 F_1(x) = 0$$

$$\Delta F_2(x) = \partial_1^2 F_2(x) + \partial_2^2 F_2(x) = 0$$

### 9.2 Komplexe Integration

#### 9.2.1 Kurvenintegrale

Sei  $f : U \rightarrow \mathbb{C}$  stetig,  $\gamma \in C^1([a, b], U)$ , dann lautet das komplexe Kurvenintegral von  $f$  entlang von  $\gamma$ :

$$\int_{\gamma} f(z) dz = \int_a^b f(\gamma(t)) \dot{\gamma}(t) dt$$

Mit

$$f(z) = f(x_1 + ix_2) = f_1(z) + i f_2(z); \quad \gamma(t) = \gamma_1(t) + i \gamma_2(t)$$

folgt

$$\begin{aligned} \int_{\gamma} f(z) dz &= \int_a^b \begin{pmatrix} f_1(\gamma(t)) \\ -f_2(\gamma(t)) \end{pmatrix}^T \begin{pmatrix} \dot{\gamma}_1(t) \\ \dot{\gamma}_2(t) \end{pmatrix} dt \\ &+ i \int_a^b \begin{pmatrix} f_2(\gamma(t)) \\ f_1(\gamma(t)) \end{pmatrix}^T \begin{pmatrix} \dot{\gamma}_1(t) \\ \dot{\gamma}_2(t) \end{pmatrix} dt \end{aligned}$$

#### Definitionen:

- $U \subset \mathbb{C}$  heißt **zusammenhängend**, falls es  $\forall z_1, z_2 \in U$  einen Weg  $\gamma : [0, 1] \rightarrow U$  gibt, sodass  $\gamma(0) = z_1, \gamma(1) = z_2$ .
- $U \subset \mathbb{C}$  heißt **einfach zusammenhängend**, falls das Innere jeder ganz in  $U$  verlaufenden geschlossenen Kurve zu  $U$  gehört.

Sei  $U \subset \mathbb{C}$  einfach zusammenhängend,  $f : U \rightarrow \mathbb{C}$  analytisch:

- Dann gilt für jede geschlossene Kurve  $\gamma$ , die ganz in  $U$  verläuft:

$$\int_{\gamma} f(z) dz = 0$$

- Bei nicht geschlossenen Kurven ist das komplexe Kurvenintegral wegunabhängig.
- Dann ist für jedes  $a \in U$

$$F(z) = \int_a^z f(\xi) d\xi$$

eine Stammfunktion von  $f$  und es gilt:

$$\int_{z_0}^{z_1} f(\xi) d\xi = F(z_1) - F(z_0)$$

### 9.2.2 Cauchy-Integralformel

Sei  $U \subset \mathbb{C}, f : U \rightarrow \mathbb{C}$  analytisch,  $\gamma$  geschlossen mit Innerem ganz in  $U$ , dann gilt für jeden Punkt  $z$  im Inneren von  $\gamma$ :

$$f(z) = \frac{1}{2\pi i} \int_{\gamma} \frac{f(\xi)}{\xi - z} d\xi$$

### 9.3 Potenzreihendarstellung analytischer Funktionen

Sei  $f : U \rightarrow \mathbb{C}$  analytisch,  $\{|z - a| \leq r\} \subset U$  für ein  $a \in U, r > 0$ , dann gilt für  $|z - a| < \rho < r$ :

$$f(z) = \sum_{n=0}^{\infty} c_n (z - a)^n$$

mit  $c_n = \frac{1}{2\pi i} \int_{|\xi - a| = r} \frac{f(\xi)}{(\xi - a)^{n+1}} d\xi$

Jede beschränkte analytische Funktion  $f : \mathbb{C} \rightarrow \mathbb{C}$  ist konstant.

### 9.4 Laurent-Reihen und Singularitäten

Idee: Entwickle  $f : \mathbb{C} \setminus \{z_0\} \rightarrow \mathbb{C}$  analytisch in einer Reihe um die Singularität  $z_0$ :

$$f(z) = \sum_{n=-\infty}^{\infty} c_n (z - z_0)^n = \underbrace{\sum_{n=-\infty}^{-1} c_n (z - z_0)^n}_{\text{Hauptteil (HT)}} + \underbrace{\sum_{n=0}^{\infty} c_n (z - z_0)^n}_{\text{Nebenteil (NT)}}$$

#### Definitionen:

- Sei  $U \subset \mathbb{C}, f : U \rightarrow \mathbb{C}$  analytisch. Dann heißt  $z_0 \in U$  **Nullstelle der Ordnung  $m$** , falls  $f(z_0) = f'(z_0) = \dots = f^{(m-1)}(z_0) = 0$  und  $f^{(m)}(z_0) \neq 0$ .
- Sei  $z_0 \in U \subset \mathbb{C}$  offen,  $f : U \setminus \{z_0\} \rightarrow \mathbb{C}$  analytisch. Dann heißt  $z_0$  **isolierte Singularität** von  $f$ .
  - Ist  $f$  auf einer punktierten Umgebung von  $z_0$  beschränkt, so heißt  $z_0$  **hebbare Singularität**.
  - Hat  $(z - z_0)^m \cdot f(z)$  für ein  $m \geq 1$  eine hebbare Singularität in  $z_0$ , dann heißt  $z_0$  **Pol**. Das kleinste solche  $m$  heißt **Ordnung des Pols**.
  - Ansonsten heißt  $z_0$  **wesentliche Singularität**.

#### Eigenschaften:

- 1) LR konvergiert, falls Hauptteil (HT) und Nebenteil (NT) konvergieren.
- 2) NT ist übliche Potenzreihe; sie habe den Konvergenzradius  $R \in [0, \infty)$ .
- 3) HT ist eine Potenzreihe in  $w = \frac{1}{z - z_0}$ ; sie habe den Konvergenzradius  $\frac{1}{r} \in [0, \infty)$ . HT konvergiert somit, falls  $|z - z_0| > r$ .
- 4) Falls  $0 \leq r < R \leq \infty$ , so konvergiert die LR im Kreisring  $\{r < |z - z_0| < R\}$ .
- 5) Eine konvergente LR kann gliedweise differenziert werden.
- 6) Ist  $c_{-1} = 0$ , so besitzt die LR die Stammfunktion

$$\sum_{n=-\infty, n \neq -1}^{\infty} \frac{c_n}{n+1} (z - z_0)^{n+1}$$

die im selben Kreisring konvergiert.

Sei  $K_{r,R}(z_0) = \{z \in \mathbb{C} | 0 \leq r < |z - z_0| < R \leq \infty\}$ ,  $f : K_{r,R}(z_0) \rightarrow \mathbb{C}$  analytisch, dann gilt:

$$f(z) = \sum_{n=-\infty}^{\infty} c_n (z - z_0)^n$$

mit  $c_n = \frac{1}{2\pi i} \int_{|z - z_0| = \rho} \frac{f(z)}{(z - z_0)^{n+1}} dz; \quad r < \rho < R$

### 9.5 Residuenkalkül

Idee: Verwende Laurent-Reihe zur Berechnung von Integralen.

Sei  $f : U \setminus \{z_0\} \rightarrow \mathbb{C}$  analytisch. Dann lautet das Residuum von  $f$  in  $z_0$ :

$$Res_{z_0}(f) = Res(f, z_0) = c_{-1} = \frac{1}{2\pi i} \int_{|z - z_0| = \rho} f(z) dz$$

Lizenz: CC BY-NC-SA 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/de/>