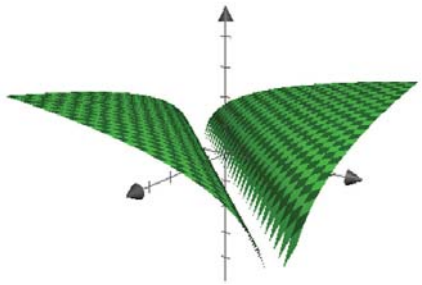
# Proseminar
# Hierarchische Matrizen

## Einführendes Beispiel



Mirko Bächtiger, 15. April 2004

### 0. Randelemtproblem

Verfahren, zur Lösung von Integralgleichungen
zu lösen ist die DGL:

$$\begin{aligned} -\triangle\Phi &= f \\ \Phi &= g_D \end{aligned}$$

Die Funktion

$$N(x) = \int_\Omega G(x-y)f(y)dy$$

mit Kern $G = \log|x-y|$ löst die DGL, erfüllt aber im Allgemeinen nicht die Randbedingungen.
Lösungen der DGL sind in der Form: $\Phi = N + \Phi_o$ (*inhomogen+homogen*)
$\Phi_o$ erfüllt:

$$\begin{aligned} \triangle\Phi_o &= 0 && ,\text{in } \Omega^+ \\ \Phi_o &= \mathcal{F}\ (= g_D - N|_\Gamma) && ,\text{auf } \Gamma \quad (\mathcal{F}: \textit{neue Randbedingungen}) \end{aligned}$$

Ansatz für $x \in \Omega$:

$$\Phi_o(x) = \int_\Gamma G(x-y)\mathcal{U}(y)d\Gamma_y, \qquad V(\mathcal{U}) = \Phi_o(x)|_\Gamma = \mathcal{F}$$

Die Funktion $\mathcal{U}: \Gamma \to \mathbf{C}$, $\mathcal{U} \in \mathbf{C}^o(\Gamma)$ ist unbestimmt.

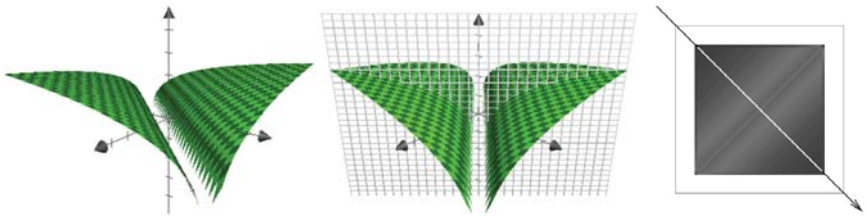
### 1. Modellproblem

Wir betrachten die Integralgleichung:

$$\int_0^1 \log|x-y|\mathcal{U}(y)dy = \mathcal{F}(x) \qquad ,x \in [0,1] \tag{1}$$

Für ein $\mathcal{F}: [0,1] \to \mathbf{R}$ und suchen die Lösung $\mathcal{U}: [0,1] \to \mathbf{R}$

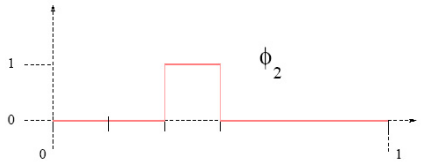
Der Kern $g(x,y) = \log|x-y|$ hat in $[0,1]^2$ eine Singularität in der Diagonalen $(x=y)$.



### Galerkin's Methode:

Lösen der Integralgleichung auf dem Raum $V_n := \text{span}\{\varphi_o, \ldots, \varphi_{n-1}\}$, wobei die Basisfunktionen wie folgt definiert sind:

$$\varphi_i(x) = \begin{cases} 1 & : \quad \frac{i}{n} \le x \le \frac{i+1}{n} \\ 0 & : \quad sonst \end{cases}$$




$$\int_0^1 \int_0^1 \varphi_i(x)\log|x-y|\mathcal{U}(x)dydx = \int_0^1 \varphi_i(x)\mathcal{F}(x)dx \qquad 0 \le i < n$$

Wir suchen diskrete Lösung $\mathcal{U}_n = \sum_{j=0}^{n-1} u_j\varphi_j$, so dass $u$ die Lösung des linearen Systems

$$Gu = f, \quad G_{ij} := \int_0^1\int_0^1 \varphi_i(x)\log|x-y|\varphi_j(y)dydx, \quad f_i := \int_0^1 \varphi_i(x)\mathcal{F}(x)dx$$

ist.

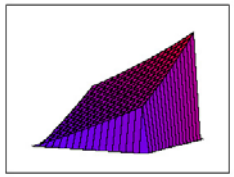
**Grundidee:**

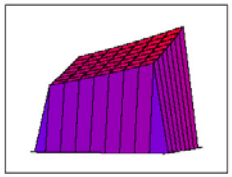
Problem: Matrix $G$ ist dicht besetzt
Ziel: durch Platz sparende Matrix $\tilde{G}$ ersetzen
Idee: den Kern $g(x,y) = \log|x-y|$ durch degenerierten Kern

$$\tilde{g}(x,y) = \sum_{\nu=0}^{k-1} g_\nu(x) h_\nu(y)$$

ersetzen (Grund: Integration bez. $x$ ist separiert von jener bez. $y$).

Der Kern $g(x,y) = \log|x-y|$ kann nicht auf ganz $[0,1]\times[0,1]$ durch degenerierten Kern $\tilde{g}$ ersetzt werden (Singularität).
$\Rightarrow$ Approximation von $g$ durch $\tilde{g}$ dort, wo $g$ glatt ist





**2. Taylorentwicklung des Kerns**

Sei $\tau := [a,b]$, $\sigma := [c,d]$, $\tau\times\sigma \subset [0,1]\times[0,1]$ eine Unterteilung mit der Eigenschaft $b < c$, so dass $\tau\cap\sigma = \emptyset$. Kern auf $\tau\cap\sigma = \emptyset$ nicht singulär.

**Lemma 1:** Ableitungen von $g(x,y) = \log|x-y|$ für $x \neq y$ und $\nu \in \mathbf{N}$ lauten:

$$\partial_x^\nu g(x,y) = (-1)^{\nu-1}(\nu-1)!(x-y)^{-\nu}$$
$$\partial_y^\nu g(x,y) = (\nu-1)!(x-y)^{-\nu}$$

**Lemma 2:** Taylorentwicklung von $g(x,y) = \log|x-y|$
Sei $x_o := (a+b)/2$. Für jedes $k \in \mathbf{N}$ approximiert die Funktion

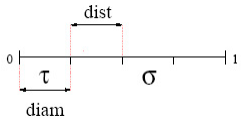
$$\tilde{g}(x,y) := \sum_{\nu=0}^{k-1} \frac{1}{\nu!}\partial_x^\nu g(x_o,y)(x-x_o)^\nu$$

den Kern $g(x,y) = \log|x-y|$ mit dem Fehler

$$|g(x,y) - \tilde{g}(x,y)| \leq \left(1 + \frac{|x_o-a|}{|c-b|}\right)\left(1 + \frac{|c-b|}{|x_o-a|}\right)^{-k}$$

Problem:
Wenn $c \to b$, dann geht die Schätzung des Fehlers gegen unendlich.
$\Rightarrow$ Bedingung $c < b$ ersetzen durch eine strengere Bedingung:

$$\text{diam}(\tau) \leq \text{dist}(\tau,\sigma)$$




Dann kann der approximierte Fehler geschrieben werden als:

$$|g(x,y) - \tilde{g}(x,y)| \leq \left(1+\frac{1}{2}\right)\left(1+\frac{2}{1}\right)^{-k} \leq 3^{-k+1}$$

(da $2|x_o-a| \leq |c-b|$)

$\Rightarrow$ gleichförmige Schranke für den approximierten Fehler, unabhängig vom Intervall, solange die zulässige Bedingung erfüllt ist.
Fehler nimmt exponential ab mit der Ordnung von $k$.

**3. Niedrigrang-Approximation von Matrixblöcken**

Sei $\tau := \bigcup_{i\in t}\text{supp}\varphi_i$ und $\sigma := \bigcup_{i\in s}\text{supp}\varphi_i$ die zugehörigen Bereiche zu $t$ und $s$ aus $\mathcal{I} := \{0,\ldots,n-1\}$.
Wenn $\tau\times\sigma$ zulässig ist gemäss $\text{diam}(\tau) \leq \text{dist}(\tau,\sigma)$, dann approximiere den Kern $g$ in diesem Bereich durch Taylorentwicklung $\tilde{g}$:

$$G_{ij} := \int_0^1\int_0^1 \varphi_i(x) g(x,y)\varphi_j(y)dydx$$

$$\tilde{G}_{ij} := \int_0^1\int_0^1 \varphi_i(x) \tilde{g}(x,y)\varphi_j(y)dydx$$

wobei $\tilde{g} = \sum_{\nu=0}^{k-1} g_\nu(x)h_\nu(y)$, der durch die Taylorapproximation konstruierte degenerierte Kern ist und $(i,j) \in t\times s$

Zwei Vorteile:

- Doppelintegral kann man aufsplitten:

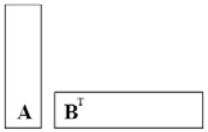
$$\begin{aligned}\tilde{G}_{ij} &= \int_0^1 \int_0^1 \varphi_i(x) \sum_{\nu=0}^{k-1} g_\nu(x) h_\nu(y) \varphi_i(y) dy dx \\ &= \sum_{\nu=0}^{k-1} \left( \int_0^1 \varphi_i(x) g_\nu(x) dx \right) \left( \int_0^1 \varphi_i(y) h_\nu(y) dy \right)\end{aligned}$$

- Untermatrix $G|_{t\times s}$ kann faktorisiert werden:

$$G|_{t\times s} = AB^T, \quad A \in \mathbf{R}^{t\times\{0,\ldots,k-1\}}, \quad B \in \mathbf{R}^{s\times\{0,\ldots,k-1\}}$$

wobei

$$A_{i\nu} := \int_0^1 \varphi_i(x) g_\nu(x) dx, \quad B_{j\nu} := \int_0^1 \varphi_j(y) h_\nu(y) dy$$




**Definition:** Zulässige Blocks
Ein Block $t \times s \subset \mathcal{I} \times \mathcal{I}$ von Indizes heisst zulässig, falls der zugehörige Bereich $\tau \times \nu$ zulässig ist gemäss $\text{diam}(\tau) \leq \text{dist}(\tau, \sigma)$.

**Lemma 3:** elementenweise Matrix Approximationsfehler
Die Matrixelemente sind in den zulässigen Blocks $t\times s$ beschränkt durch:

$$|G_{ij} - \tilde{G}_{ij}| \leq n^{-2} 3^{-k+1}$$

**Lemma 4:** Matrix Approximationsfehler
Der Approximationsfehler $||G - \tilde{G}||_F$ für die Matrix $\tilde{G}$ mit dem degenerierten Kern $\tilde{g}$ in den zulässigen Blocks $t_\nu \times s_\nu$ und $g$ in den unzulässigen Blocks ist beschränkt durch:

$$||G - \tilde{G}||_F \leq n^{-2} 3^{-k+1} \qquad wobei \quad ||M||_F^2 := \sum M_{ij}^2 \quad \text{(Frobenius-Norm)}$$

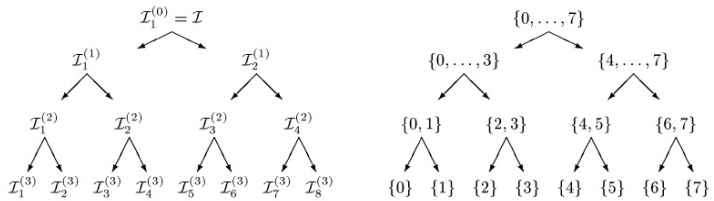
Frage: Wie wollen wir $G$ unterteilen?

**4. Cluster Baum** $T_\mathcal{I}$

Die Kandidaten für $t, s \subset \mathcal{I}$ für die Konstruktion der Partition werden in einem Cluster Baum gespeichert. ($n = 2^p$)

Wurzel des Baumes: $\mathcal{I}_1^{(0)} := \{0, \ldots, n-1\}$

Nachfahre von $\mathcal{I}_1^{(0)}$: $\mathcal{I}_1^{(1)} := \{0, \ldots, \frac{n}{2} - 1\}$ und $\mathcal{I}_2^{(1)} := \{\frac{n}{2}, \ldots, n-1\}$
Nachfahre von $\mathcal{I}_1^{(1)}$: $\mathcal{I}_1^{(2)} := \{0, \ldots, \frac{n}{4} - 1\}$ und $\mathcal{I}_2^{(2)} := \{\frac{n}{4}, \ldots, \frac{n}{2} - 1\}$
Nachfahre von $\mathcal{I}_2^{(1)}$: $\mathcal{I}_3^{(2)} := \{\frac{n}{2}, \ldots, \frac{3n}{4} - 1\}$ und $\mathcal{I}_4^{(2)} := \{\frac{3n}{4}, \ldots, n-1\}$
usw.

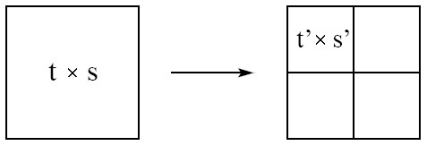



**5. Block Cluster Baum** $T_{\mathcal{I}\times\mathcal{I}}$

Anzahl der möglichen Blocks $t \times s$ vom Baum ist
$(\#T_\mathcal{I})^2 = (2n-1)^2 = \mathcal{O}(n^2)$
$\Rightarrow$ dauert zu lange, um alle zu testen.

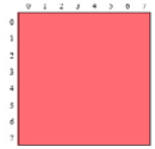
Eine Methode: Block Cluster Baum Algorithmus

1. (**Start**) $root(T_{\mathcal{I}\times\mathcal{I}}) := \mathcal{I} \times \mathcal{I}$
2. (**Iterate**) Für jedes unzulässige Blatt $t \times s$ definiere Nachfahre von $t \times s$ als $\quad S(t \times s) := \{t' \times s' \mid t' \in S(t) \ \ und \ \ s' \in S(s)\}$
3. (**Stop**) alle Blätter sind zulässig oder $S(t) = \emptyset$ oder $S(s) = \emptyset$



**Beispiel:** Block cluster Baum, $p = 3$

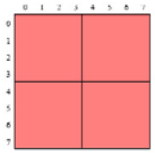
$\{0,\ldots,7\} \times \{0,\ldots,7\}$



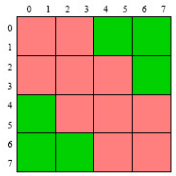
Die Wurzel ist nicht zulässig, da der zugehörige Bereich zum Index $\{0,\ldots,7\}$ das Intervall $[0,1]$ ist und

$$\text{diam}([0,1]) = 1 \not\leq 0 = \text{dist}([0,1],[0,1])$$

$\{0,1,2,3\} \times \{0,1,2,3\}$, $\{0,1,2,3\} \times \{4,5,6,7\}$,
$\{4,5,6,7\} \times \{0,1,2,3\}$, $\{4,5,6,7\} \times \{4,5,6,7\}$.

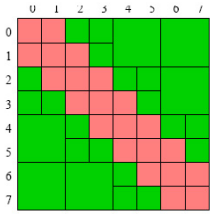


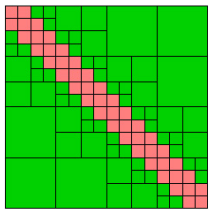
$\{0,1\} \times \{0,1\}$, $\{0,1\} \times \{2,3\}$, $\{0,1\} \times \{4,5\}$, $\{0,1\} \times \{6,7\}$,
$\{2,3\} \times \{0,1\}$, $\{2,3\} \times \{2,3\}$, $\{2,3\} \times \{4,5\}$, $\{2,3\} \times \{6,7\}$,
$\{4,5\} \times \{0,1\}$, $\{4,5\} \times \{2,3\}$, $\{4,5\} \times \{4,5\}$, $\{4,5\} \times \{6,7\}$,
$\{6,7\} \times \{0,1\}$, $\{6,7\} \times \{2,3\}$, $\{6,7\} \times \{4,5\}$, $\{6,7\} \times \{6,7\}$.

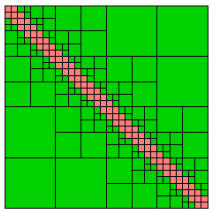


$\{0,1\} \times \{4,5\}$ ist zulässig, da für den zugehörigen Bereich $\left[0,\frac{1}{4}\right] \times \left[\frac{1}{2},\frac{3}{4}\right]$ gilt:

$$\text{diam}\left(\left[0,\frac{1}{4}\right]\right) = \frac{1}{4} = \text{dist}\left(\left[0,\frac{1}{4}\right],\left[\frac{1}{2},\frac{3}{4}\right]\right)$$







## 6. Speicherung von unzulässigen Blättern

unzulässige *(aber kleine)* Blocks $t \times s \subset \mathcal{I} \times \mathcal{I}$, $G$ werden ersetzt durch:

$$\tilde{G}_{ij} := \int_0^1 \int_0^1 \varphi_i(x) \log|x-y| \varphi_j(y) dy dx = \int_{i/n}^{(i+1)/n} \int_{j/n}^{(j+1)/n} \log|x-y| dy dx$$

**Definition:** `fullmatrix` Darstellung
Eine $n \times m$ Matrix $M$ wird in einer `fullmatrix` dargestellt, wenn die Einträge $M_{ij}$ vom Typ `double` sind und in einem Array
$M_{11},\ldots,M_{n1},M_{12},\ldots,M_{n2}, \quad \ldots \quad ,M_{1m},\ldots,M_{nm} \qquad (spaltenweise)$
der Länge $nm$ gespeichert werden.

**Implementierung:** `fullmatrix`

```
typedef struct _fullmatrix fullmatrix;
typedef fullmatrix *pfullmatrix;

struct _fullmatrix {
  int rows;
  int cols;
  double* e;
};
```

## 7. Speicherung von zulässigen Blättern

zulässige Blocks $t \times s \subset \mathcal{I} \times \mathcal{I}$ mit zugehörigen Bereichen $[a,b] \times [c,d]$ und $x_o := (a+b)/2$, Untermatrix wird faktorisiert als:

$$\begin{aligned}
\tilde{G}|_{t\times s} &:= AB^T, \\
A_{i\nu} &:= \int_{i/n}^{(i+1)/n} (x-x_o)^\nu dx, \\
B_{j\nu} &:= \begin{cases} (-1)^{\nu+1}\nu^{-1} \int_{j/n}^{(j+1)/n} (x_o-y)^{-\nu} dy & : \quad \nu > 0 \\ \int_{j/n}^{(j+1)/n} log|x_o-y| dy & : \quad \nu = 0 \end{cases}
\end{aligned}$$

**Definition:** `rkmatrix` Darstellung
Eine $n \times m$ Matrix $M$ mit Rang mind. $k$ wird in `rkmatrix` gespeichert, falls $M = AB^T$, wobei $A \in \mathbf{R}^{n\times k}$ und $B \in \mathbf{R}^{m\times k}$ als Array gespeichert werden (spaltenweise).

**Implementierung:** rkmatrix

```
typedef struct _rkmatrix rkmatrix;
typedef rkmatrix *prkmatrix;

struct _rkmatrix {
  int k;
  int rows;
  int cols;
  double* a;
  double* b;
};
```

## 8. Hierarchische Matrix Darstellung

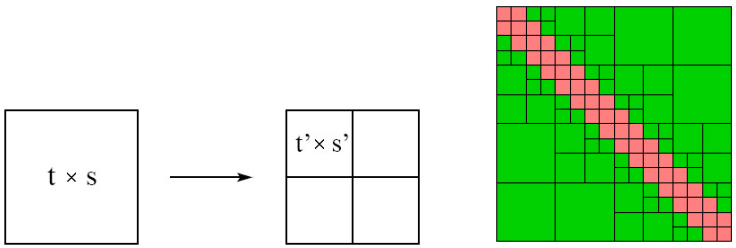
**Definition:** $\mathcal{H}$-Matrix
Sei $T_{\mathcal{I}\times\mathcal{I}}$ ein Block cluster Baum für die Indexmenge $\mathcal{I}$. Dann definieren wir die $\mathcal{H}$-Matrizen als

$$\mathcal{H}(T_{\mathcal{I}\times\mathcal{I}}, k) := \{M \in \mathbf{R}^{\mathcal{I}\times\mathcal{I}} | rank(M|_{t\times s}) \leq k \,\forall \text{ zul. Blätter } t\times s \text{ von } T_{\mathcal{I}\times\mathcal{I}}\}$$

**Definition:** $\mathcal{H}$-Matrix Darstellung
Sei $T_{\mathcal{I}\times\mathcal{I}}$ ein Block cluster Baum für die Indexmenge $\mathcal{I}$. Eine Matrix $M \in \mathcal{H}(T_{\mathcal{I}\times\mathcal{I}})$ ist in einer $\mathcal{H}$-Matrix Darstellung gespeichert, falls die Untermatrizen der unzulässigen Blätter in `fullmatrix` gespeichert und die Untermatrizen der zulässigen Blätter in `rkmatrix` gespeichert werden.

Jeder Block $t \times s$ im Baum $T_{\mathcal{I}\times\mathcal{I}}$ kann sein:

- ein Blatt - dann ist der zugehörige Matrixblock dargestellt als `fullmatrix` oder `rkmatrix`.
- kein Blatt - dann wird der Block $t \times s$ in seine Söhne $t' \times s'$ mit $t' \in S(t)$ und $s' \in S(s)$ aufgeteilt. D.h. dass der Matrixblock eine Supermatrix ist.




**Implementierung:** supermatrix

```
typedef struct _supermatrix supermatrix;
typedef supermatrix *psupermatrix;

struct _supermatrix {
  int rows;
  int cols;
  int block_rows;
  int block_cols;
  prkmatrix r;
  pfullmatrix f;
  psupermatrix* s;
};
```

Eine Supermatrix besteht aus `block_rows` × `block_cols` Untermatrizen. Die Größe der Matrix ist `rows` × `cols`, d.h. $M \in \mathbf{R}^{\texttt{rows}\times\texttt{cols}}$.

Die Matrix $M$ kann sein:

- eine `rkmatrix` - dann $r \neq 0 \times 0$, $f = 0 \times 0$ und $s = 0 \times 0$: $r$ ist die `rkmatrix`-Darstellung von $M$.

- eine `fullmatrix` - dann $f \neq 0 \times 0$, $r = 0 \times 0$ und $s = 0 \times 0$: $f$ ist die `fullmatrix`-Darstellung von $M$.

- eine `supermatrix` - dann $s \neq 0 \times 0$, $f = 0 \times 0$ und $r = 0 \times 0$: der Array $s$ enthält Pointer auf die Untermatrizen $M_{ij}$ von

$$M = \begin{bmatrix} M_{1,1} & \cdots & M_{1,\mathtt{blockcols}} \\ \vdots & \ddots & \vdots \\ M_{\mathtt{blockrows},1} & \cdots & M_{\mathtt{blockrows},\mathtt{blockcols}} \end{bmatrix}$$

  In der Reihenfolge:
  $M_{1,1}, \ldots, M_{\mathtt{blockrows},1}$, $M_{1,2}, \ldots, M_{\mathtt{blockrows},2}$, $\cdots$
  $\cdots$ , $M_{1,\mathtt{blockcols}}, \ldots, M_{\mathtt{blockrows},\mathtt{blockcols}}$