

# Effiziente Verfahren für vollbesetzte Matrizen

Steffen Börm

Christian-Albrechts-Universität, Kiel

Norddeutsches Kolloquium 2012

# Übersicht

- 1 Einleitung
- 2  $\mathcal{H}^2$ -Matrizen
- 3 Aktuelle Projekte
- 4 Zusammenfassung

# Übersicht

- 1 Einleitung
- 2  $\mathcal{H}^2$ -Matrizen
- 3 Aktuelle Projekte
- 4 Zusammenfassung

# Modellproblem

**Ziel:** Gegeben die Kernfunktion  $g(x, y) = -\log|x - y|$ , löse die Integralgleichung

$$\int_0^1 g(x, y)u(y) dy = f(x) \quad \text{für } x \in [0, 1].$$

**Diskretisierung** führt zu Gleichungssystem  $Gu = b$  mit einer Matrix  $G \in \mathbb{R}^{n \times n}$ , gegeben durch

$$G_{ij} = \int_{(i-1)/n}^{i/n} \int_{(j-1)/n}^{j/n} g(x, y) dy dx \quad \text{für } 1 \leq i, j \leq n.$$

**Problem:** Hohe Genauigkeit erfordert  $n \gg 1$ , Speicherbedarf  $\sim n^2$ .

**Idee:** Approximiere  $G$  durch effizient darstellbare Matrix.

# Entartete Approximation

**Analysis:** Sei  $\tau \subseteq [0, 1]$  ein Intervall und  $\sigma \subseteq [0, 1]$  eine Menge sowie

$$\tilde{g}_k(x, y) := \sum_{\nu=1}^k \mathcal{L}_{\tau, \nu}(x) g(\xi_{\tau, \nu}, y) \quad \text{für } x \in \tau, y \in \sigma$$

das Interpolationspolynom in Punkten  $\xi_{\tau, 1}, \dots, \xi_{\tau, k} \in \tau$ . Dann gilt

$$\|g - \tilde{g}_k\|_{\infty, \tau \times \sigma} \leq 2 \left( \frac{\text{diam}(\tau)}{4 \text{dist}(\tau, \sigma)} \right)^k$$

# Entartete Approximation

**Analysis:** Sei  $\tau \subseteq [0, 1]$  ein Intervall und  $\sigma \subseteq [0, 1]$  eine Menge sowie

$$\tilde{g}_k(x, y) := \sum_{\nu=1}^k \mathcal{L}_{\tau, \nu}(x) g(\xi_{\tau, \nu}, y) \quad \text{für } x \in \tau, y \in \sigma$$

das Interpolationspolynom in Punkten  $\xi_{\tau, 1}, \dots, \xi_{\tau, k} \in \tau$ . Dann gilt

$$\|g - \tilde{g}_k\|_{\infty, \tau \times \sigma} \leq 2 \left( \frac{\text{diam}(\tau)}{4 \text{dist}(\tau, \sigma)} \right)^k \lesssim 4^{-k}$$

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

**Ergebnis:**  $g$  kann durch entartete Kernfunktion approximiert werden, Approximation konvergiert exponentiell mit wachsendem  $k$ .

# Approximative Faktorisierung

Idee: Konstruiere Approximation von  $G$  aus Approximation von  $g$ .

$$G_{ij} = \int_{(i-1)/n}^{i/n} \int_{(j-1)/n}^{j/n} g(x, y) dy dx$$

# Approximative Faktorisierung

Idee: Konstruiere Approximation von  $G$  aus Approximation von  $g$ .

$$G_{ij} \approx \int_{(i-1)/n}^{i/n} \int_{(j-1)/n}^{j/n} \sum_{\nu=1}^k \mathcal{L}_{\tau,\nu}(x) g(\xi_{\tau,\nu}, y) dy dx$$



# Approximative Faktorisierung

Idee: Konstruiere Approximation von  $G$  aus Approximation von  $g$ .

$$G_{ij} \approx \sum_{\nu=1}^k \int_{(i-1)/n}^{i/n} \mathcal{L}_{\tau,\nu}(x) dx \int_{(j-1)/n}^{j/n} g(\xi_{\tau,\nu}, y) dy$$

# Approximative Faktorisierung

Idee: Konstruiere Approximation von  $G$  aus Approximation von  $g$ .

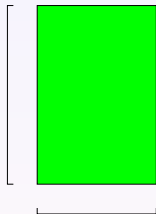
$$G_{ij} \approx \sum_{\nu=1}^k \underbrace{\int_{(i-1)/n}^{i/n} \mathcal{L}_{\tau,\nu}(x) dx}_{=:A_{i\nu}} \underbrace{\int_{(j-1)/n}^{j/n} g(\xi_{\tau,\nu}, y) dy}_{=:B_{j\nu}} = (AB^*)_{ij}.$$

Blockapproximation: Wähle Mengen  $\hat{\tau}, \hat{\sigma} \subseteq \{1, \dots, n\}$  mit

$$\begin{aligned} [(i-1)/n, i/n] &\subseteq \tau, & \text{für alle } i \in \hat{\tau}, \\ [(j-1)/n, j/n] &\subseteq \sigma & \text{für alle } j \in \hat{\sigma}, \end{aligned}$$

dann haben wir  $A \in \mathbb{R}^{\hat{\tau} \times k}$ ,  $B \in \mathbb{R}^{\hat{\sigma} \times k}$  gefunden mit

$$\|G|_{\hat{\tau} \times \hat{\sigma}} - AB^*\|_2 \lesssim 4^{-k}.$$



# Approximative Faktorisierung

Idee: Konstruiere Approximation von  $G$  aus Approximation von  $g$ .

$$G_{ij} \approx \sum_{\nu=1}^k \underbrace{\int_{(i-1)/n}^{i/n} \mathcal{L}_{\tau,\nu}(x) dx}_{=:A_{i\nu}} \underbrace{\int_{(j-1)/n}^{j/n} g(\xi_{\tau,\nu}, y) dy}_{=:B_{j\nu}} = (AB^*)_{ij}.$$

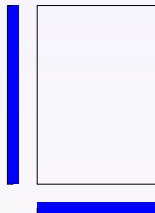
Blockapproximation: Wähle Mengen  $\hat{\tau}, \hat{\sigma} \subseteq \{1, \dots, n\}$  mit

$$[(i-1)/n, i/n] \subseteq \tau, \quad \text{für alle } i \in \hat{\tau},$$

$$[(j-1)/n, j/n] \subseteq \sigma \quad \text{für alle } j \in \hat{\sigma},$$

dann haben wir  $A \in \mathbb{R}^{\hat{\tau} \times k}$ ,  $B \in \mathbb{R}^{\hat{\sigma} \times k}$  gefunden mit

$$\|G|_{\hat{\tau} \times \hat{\sigma}} - AB^*\|_2 \lesssim 4^{-k}.$$



# Approximative Faktorisierung

Idee: Konstruiere Approximation von  $G$  aus Approximation von  $g$ .

$$G_{ij} \approx \sum_{\nu=1}^k \underbrace{\int_{(i-1)/n}^{i/n} \mathcal{L}_{\tau,\nu}(x) dx}_{=: A_{i\nu}} \underbrace{\int_{(j-1)/n}^{j/n} g(\xi_{\tau,\nu}, y) dy}_{=: B_{j\nu}} = (AB^*)_{ij}.$$

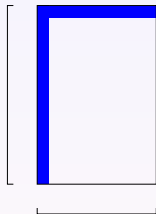
Blockapproximation: Wähle Mengen  $\hat{\tau}, \hat{\sigma} \subseteq \{1, \dots, n\}$  mit

$$[(i-1)/n, i/n] \subseteq \tau, \quad \text{für alle } i \in \hat{\tau},$$

$$[(j-1)/n, j/n] \subseteq \sigma \quad \text{für alle } j \in \hat{\sigma},$$

dann haben wir  $A \in \mathbb{R}^{\hat{\tau} \times k}$ ,  $B \in \mathbb{R}^{\hat{\sigma} \times k}$  gefunden mit

$$\|G|_{\hat{\tau} \times \hat{\sigma}} - AB^*\|_2 \lesssim 4^{-k}.$$



# Hierarchische Matrix

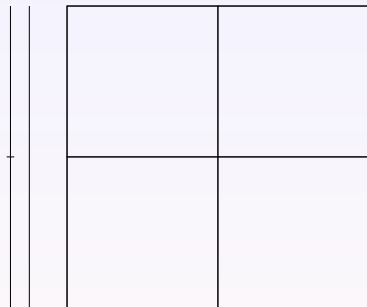
Idee: Zerlege  $G$  in Teilmatrizen, die approximiert werden können, ...



Approximation auf  $\tau \times \sigma$  zulässig,  
falls  $\text{diam}(\tau) \leq \text{dist}(\tau, \sigma)$  gilt.

# Hierarchische Matrix

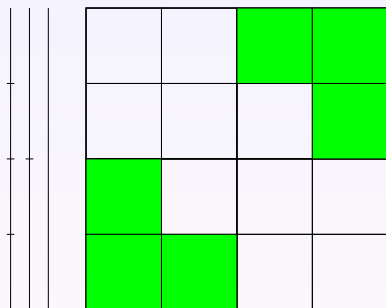
Idee: Zerlege  $G$  in Teilmatrizen, die approximiert werden können, ...



Approximation auf  $\tau \times \sigma$  zulässig,  
falls  $\text{diam}(\tau) \leq \text{dist}(\tau, \sigma)$  gilt.

# Hierarchische Matrix

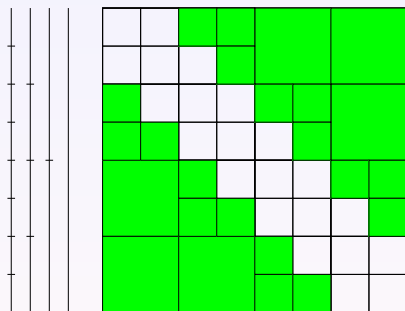
Idee: Zerlege  $G$  in Teilmatrizen, die approximiert werden können, ...



Approximation auf  $\tau \times \sigma$  zulässig,  
falls  $\text{diam}(\tau) \leq \text{dist}(\tau, \sigma)$  gilt.

# Hierarchische Matrix

Idee: Zerlege  $G$  in Teilmatrizen, die approximiert werden können, ...

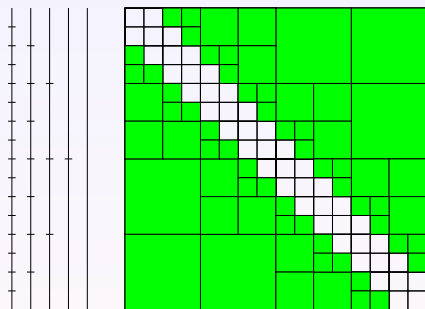


Approximation auf  $\tau \times \sigma$  zulässig,  
falls  $\text{diam}(\tau) \leq \text{dist}(\tau, \sigma)$  gilt.



# Hierarchische Matrix

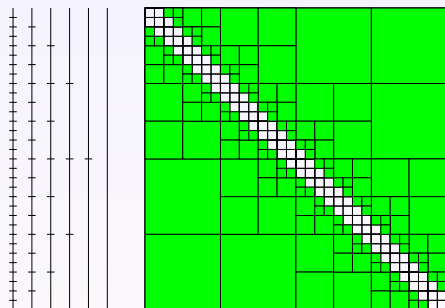
Idee: Zerlege  $G$  in Teilmatrizen, die approximiert werden können, ...



Approximation auf  $\tau \times \sigma$  zulässig,  
falls  $\text{diam}(\tau) \leq \text{dist}(\tau, \sigma)$  gilt.

# Hierarchische Matrix

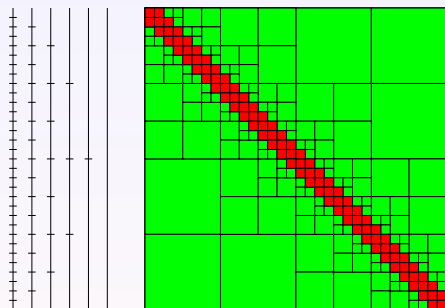
Idee: Zerlege  $G$  in Teilmatrizen, die approximiert werden können, ...



Approximation auf  $\tau \times \sigma$  zulässig,  
falls  $\text{diam}(\tau) \leq \text{dist}(\tau, \sigma)$  gilt.

# Hierarchische Matrix

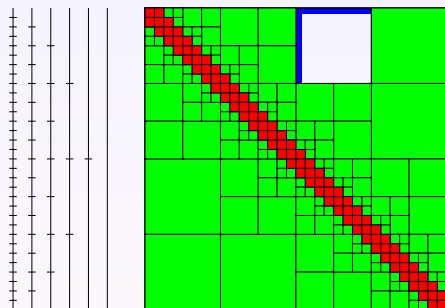
**Idee:** Zerlege  $G$  in Teilmatrizen, die approximiert werden können, und einen Rest, der eine schwachbesetzte Matrix ist.



Matrix zerlegt in **Blöcke**  $b = (\tau, \sigma)$ ,  
konstruiert aus **Clustern**  $\tau, \sigma$ .

# Hierarchische Matrix

**Idee:** Zerlege  $G$  in Teilmatrizen, die approximiert werden können, und einen Rest, der eine schwachbesetzte Matrix ist.



Matrix zerlegt in Blöcke  $b = (\tau, \sigma)$ ,  
konstruiert aus Clustern  $\tau, \sigma$ .

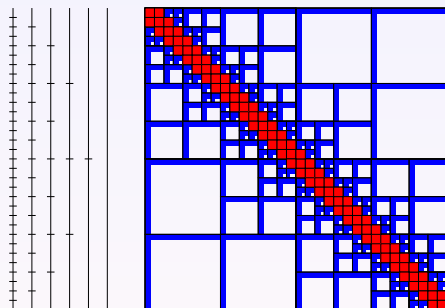
**Zulässige** Blöcke gespeichert  
in faktorisierter Form

$$G|_{\hat{\tau} \times \hat{\sigma}} \approx A_b B_b^*$$

mit  $A_b \in \mathbb{R}^{\hat{\tau} \times k}$ ,  $B_b \in \mathbb{R}^{\hat{\sigma} \times k}$ .

# Hierarchische Matrix

**Idee:** Zerlege  $G$  in Teilmatrizen, die approximiert werden können, und einen Rest, der eine schwachbesetzte Matrix ist.



Matrix zerlegt in Blöcke  $b = (\tau, \sigma)$ ,  
konstruiert aus Clustern  $\tau, \sigma$ .

Zulässige Blöcke gespeichert  
in faktorisierter Form

$$G|_{\hat{\tau} \times \hat{\sigma}} \approx A_b B_b^*$$

mit  $A_b \in \mathbb{R}^{\hat{\tau} \times k}$ ,  $B_b \in \mathbb{R}^{\hat{\sigma} \times k}$ .

**Ergebnis:** Hierarchische Matrix, Speicherbedarf  $\sim nk \log_2 n$ .

# Übersicht

- 1 Einleitung
- 2  $\mathcal{H}^2$ -Matrizen**
- 3 Aktuelle Projekte
- 4 Zusammenfassung

# Uniforme $\mathcal{H}$ -Matrix

Idee: Interpolation in **beiden** Variablen für  $b = (\tau, \sigma)$

$$\tilde{g}_k(x, y) \approx \sum_{\nu=1}^k \sum_{\mu=1}^k \mathcal{L}_{\tau, \nu}(x) g(\xi_{\tau, \nu}, \xi_{\sigma, \mu}) \mathcal{L}_{\sigma, \mu}(y)$$

# Uniforme $\mathcal{H}$ -Matrix

Idee: Interpolation in beiden Variablen für  $b = (\tau, \sigma)$

$$G_{ij} \approx \sum_{\nu=1}^k \sum_{\mu=1}^k \underbrace{\int_{(i-1)/n}^{i/n} \mathcal{L}_{\tau,\nu}(x) dx}_{=: V_{\tau,i\nu}} \underbrace{g(\xi_{\tau,\nu}, \xi_{\sigma,\mu})}_{=: S_{b,\nu\mu}} \underbrace{\int_{(j-1)/n}^{j/n} \mathcal{L}_{\sigma\mu}(y) dy}_{=: W_{\sigma,j\mu}}$$



# Uniforme $\mathcal{H}$ -Matrix

Idee: Interpolation in beiden Variablen für  $b = (\tau, \sigma)$

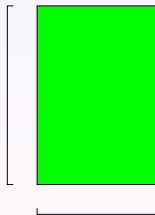
$$G_{ij} \approx \sum_{\nu=1}^k \sum_{\mu=1}^k \underbrace{\int_{(i-1)/n}^{i/n} \mathcal{L}_{\tau,\nu}(x) dx}_{=: V_{\tau,i\nu}} \underbrace{g(\xi_{\tau,\nu}, \xi_{\sigma,\mu})}_{=: S_{b,\nu\mu}} \underbrace{\int_{(j-1)/n}^{j/n} \mathcal{L}_{\sigma\mu}(y) dy}_{=: W_{\sigma,j\mu}}$$

Blockapproximation durch Faktorisierung in drei Terme

$$G|_{\hat{\tau} \times \hat{\sigma}} \approx V_{\tau} S_b W_{\sigma}^*$$

Clusterbasen  $(V_{\tau})_{\tau}$  und  $(W_{\sigma})_{\sigma}$  hängen jeweils nur von einem Cluster ab.

Kopplungsmatrizen  $(S_b)_b$  sind  $k \times k$ -Matrizen.



# Uniforme $\mathcal{H}$ -Matrix

Idee: Interpolation in beiden Variablen für  $b = (\tau, \sigma)$

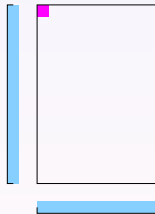
$$G_{ij} \approx \sum_{\nu=1}^k \sum_{\mu=1}^k \underbrace{\int_{(i-1)/n}^{i/n} \mathcal{L}_{\tau,\nu}(x) dx}_{=: V_{\tau,i\nu}} \underbrace{g(\xi_{\tau,\nu}, \xi_{\sigma,\mu})}_{=: S_{b,\nu\mu}} \underbrace{\int_{(j-1)/n}^{j/n} \mathcal{L}_{\sigma\mu}(y) dy}_{=: W_{\sigma,j\mu}}$$

Blockapproximation durch Faktorisierung in drei Terme

$$G|_{\hat{\tau} \times \hat{\sigma}} \approx V_{\tau} S_b W_{\sigma}^*$$

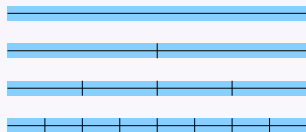
Clusterbasen  $(V_{\tau})_{\tau}$  und  $(W_{\sigma})_{\sigma}$  hängen jeweils nur von einem Cluster ab.

Kopplungsmatrizen  $(S_b)_b$  sind  $k \times k$ -Matrizen.



# Clusterbasen

Unschön: Speicherbedarf der Clusterbasis  $(V_\tau)_\tau$  wächst wie  $n \log n$ .

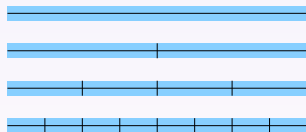


# Clusterbasen

**Unschön:** Speicherbedarf der Clusterbasis  $(V_T)_T$  wächst wie  $n \log n$ .

**Hierarchie:** Dank der konstanten Interpolationsordnung gilt

$$\mathcal{L}_{T,\nu}(x) = \sum_{\kappa=1}^k \mathcal{L}_{T',\kappa}(x) \mathcal{L}_{T,\nu}(\xi_{T',\kappa})$$

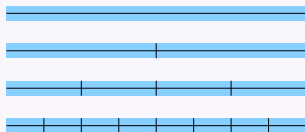


# Clusterbasen

**Unschön:** Speicherbedarf der Clusterbasis  $(V_\tau)_\tau$  wächst wie  $n \log n$ .

**Hierarchie:** Dank der konstanten Interpolationsordnung gilt

$$V_{\tau, i\nu} = \int_{(i-1)/n}^{i/n} \mathcal{L}_{\tau, \nu}(x) dx = \sum_{\kappa=1}^k \underbrace{\int_{(i-1)/n}^{i/n} \mathcal{L}_{\tau', \kappa}(x) dx}_{=V_{\tau', i\kappa}} \underbrace{\mathcal{L}_{\tau, \nu}(\xi_{\tau', \kappa})}_{=:E_{\tau', \kappa\nu}}$$



# Clusterbasen

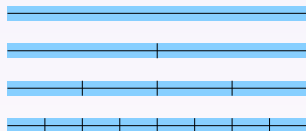
**Unschön:** Speicherbedarf der Clusterbasis  $(V_\tau)_\tau$  wächst wie  $n \log n$ .

**Hierarchie:** Dank der konstanten Interpolationsordnung gilt

$$V_{\tau, i\nu} = \int_{(i-1)/n}^{i/n} \mathcal{L}_{\tau, \nu}(x) dx = \sum_{\kappa=1}^k \underbrace{\int_{(i-1)/n}^{i/n} \mathcal{L}_{\tau', \kappa}(x) dx}_{=V_{\tau', i\kappa}} \underbrace{\mathcal{L}_{\tau, \nu}(\xi_{\tau', \kappa})}_{=:E_{\tau', \kappa\nu}}$$

Falls  $\text{sons}(\tau) = \{\tau_1, \tau_2\}$  gilt, erhalten wir

$$V_\tau = \begin{pmatrix} V_{\tau_1} & E_{\tau_1} \\ V_{\tau_2} & E_{\tau_2} \end{pmatrix}.$$



**Idee:** Falls  $\tau$  kein Blattcluster ist, können wir  $V_\tau$  durch die **Transfermatrizen**  $E_{\tau'} \in \mathbb{R}^{k \times k}$  seiner Söhne  $\tau'$  beschreiben.

# Clusterbasen

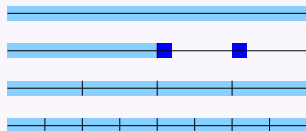
**Unschön:** Speicherbedarf der Clusterbasis  $(V_\tau)_\tau$  wächst wie  $n \log n$ .

**Hierarchie:** Dank der konstanten Interpolationsordnung gilt

$$V_{\tau, i\nu} = \int_{(i-1)/n}^{i/n} \mathcal{L}_{\tau, \nu}(x) dx = \sum_{\kappa=1}^k \underbrace{\int_{(i-1)/n}^{i/n} \mathcal{L}_{\tau', \kappa}(x) dx}_{=V_{\tau', i\kappa}} \underbrace{\mathcal{L}_{\tau, \nu}(\xi_{\tau', \kappa})}_{=:E_{\tau', \kappa\nu}}$$

Falls  $\text{sons}(\tau) = \{\tau_1, \tau_2\}$  gilt, erhalten wir

$$V_\tau = \begin{pmatrix} V_{\tau_1} & E_{\tau_1} \\ V_{\tau_2} & E_{\tau_2} \end{pmatrix}.$$



**Idee:** Falls  $\tau$  kein Blattcluster ist, können wir  $V_\tau$  durch die Transfermatrizen  $E_{\tau'} \in \mathbb{R}^{k \times k}$  seiner Söhne  $\tau'$  beschreiben.

# Clusterbasen

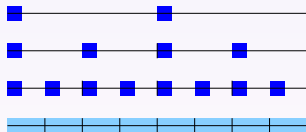
**Unschön:** Speicherbedarf der Clusterbasis  $(V_\tau)_\tau$  wächst wie  $n \log n$ .

**Hierarchie:** Dank der konstanten Interpolationsordnung gilt

$$V_{\tau, i\nu} = \int_{(i-1)/n}^{i/n} \mathcal{L}_{\tau, \nu}(x) dx = \sum_{\kappa=1}^k \underbrace{\int_{(i-1)/n}^{i/n} \mathcal{L}_{\tau', \kappa}(x) dx}_{=V_{\tau', i\kappa}} \underbrace{\mathcal{L}_{\tau, \nu}(\xi_{\tau', \kappa})}_{=:E_{\tau', \kappa\nu}}$$

Falls  $\text{sons}(\tau) = \{\tau_1, \tau_2\}$  gilt, erhalten wir

$$V_\tau = \begin{pmatrix} V_{\tau_1} & E_{\tau_1} \\ V_{\tau_2} & E_{\tau_2} \end{pmatrix}.$$



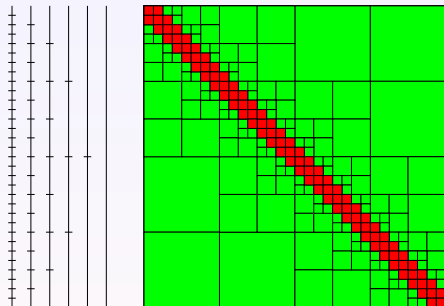
**Idee:** Falls  $\tau$  kein Blattcluster ist, können wir  $V_\tau$  durch die Transfermatrizen  $E_{\tau'} \in \mathbb{R}^{k \times k}$  seiner Söhne  $\tau'$  beschreiben.

**Ergebnis:** Speicherbedarf reduziert auf  $\sim nk$ .



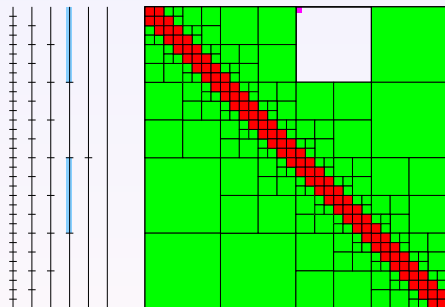
# $\mathcal{H}^2$ -Matrix

Ausgangspunkt ist wieder eine Zerlegung der Matrix in zulässige und (kleine) unzulässige Blöcke.



# $\mathcal{H}^2$ -Matrix

Ausgangspunkt ist wieder eine Zerlegung der Matrix in zulässige und (kleine) unzulässige Blöcke.

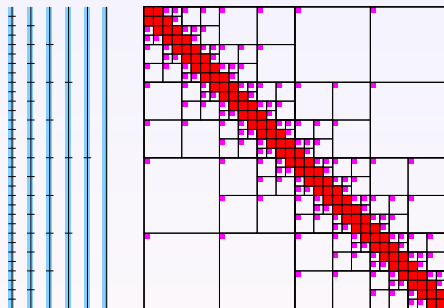


Zulässige Blöcke gespeichert  
in faktorisierter Form

$$G|_{\hat{\tau} \times \hat{\sigma}} \approx V_{\tau} S_b W_{\sigma}^*.$$

# $\mathcal{H}^2$ -Matrix

Ausgangspunkt ist wieder eine Zerlegung der Matrix in zulässige und (kleine) unzulässige Blöcke.

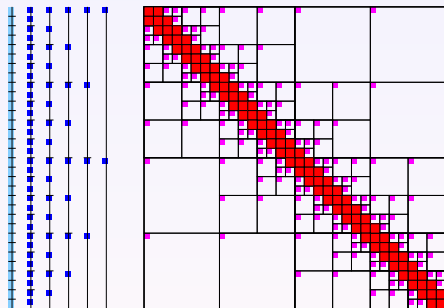


Zulässige Blöcke gespeichert  
in faktorisierter Form

$$G|_{\hat{\tau} \times \hat{\sigma}} \approx V_{\tau} S_b W_{\sigma}^*.$$

# $\mathcal{H}^2$ -Matrix

Ausgangspunkt ist wieder eine Zerlegung der Matrix in zulässige und (kleine) unzulässige Blöcke.



Zulässige Blöcke gespeichert  
in faktorisierte Form

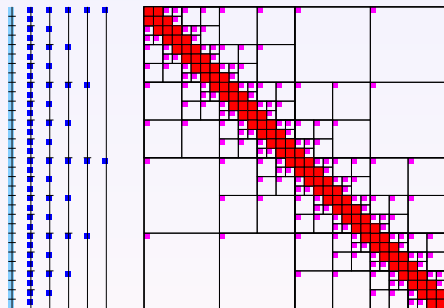
$$G|_{\hat{\tau} \times \hat{\sigma}} \approx V_{\tau} S_b W_{\sigma}^*.$$

Clusterbasen ebenfalls faktorisiert

$$V_{\tau}|_{\hat{\tau}' \times k} = V_{\tau'} E_{\tau'}.$$

# $\mathcal{H}^2$ -Matrix

Ausgangspunkt ist wieder eine Zerlegung der Matrix in zulässige und (kleine) unzulässige Blöcke.



Zulässige Blöcke gespeichert  
in faktorisierter Form

$$G|_{\hat{\tau} \times \hat{\sigma}} \approx V_{\tau} S_b W_{\sigma}^*.$$

Clusterbasen ebenfalls faktorisiert

$$V_{\tau}|_{\hat{\tau}' \times k} = V_{\tau'} E_{\tau'}.$$

Ergebnis:  $\mathcal{H}^2$ -Matrix, Speicherbedarf  $\sim nk$ .

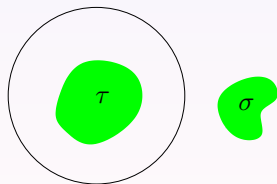
# Übersicht

- 1 Einleitung
- 2  $\mathcal{H}^2$ -Matrizen
- 3 Aktuelle Projekte**
- 4 Zusammenfassung

# Approximation per Green'scher Formel

Randintegralmethode: Kernfunktion  $g$  erfüllt Green'sche Formel

$$g(x, y) = \int_{\partial\omega} g(x, z) \frac{\partial g}{\partial n_z}(z, y) dz - \int_{\partial\omega} \frac{\partial g}{\partial n_z}(x, z) g(z, y) dz.$$



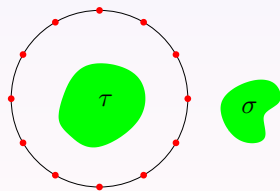
# Approximation per Green'scher Formel

Randintegralmethode: Kernfunktion  $g$  erfüllt Green'sche Formel

$$g(x, y) = \int_{\partial\omega} g(x, z) \frac{\partial g}{\partial n_z}(z, y) dz - \int_{\partial\omega} \frac{\partial g}{\partial n_z}(x, z) g(z, y) dz.$$

Idee: Approximiere Randintegral per Quadraturverfahren.

$$g(x, y) \approx \sum_{\nu=1}^k w_{\nu} g(x, z_{\nu}) \frac{\partial g}{\partial n_z}(z_{\nu}, y) - \sum_{\nu=1}^k w_{\nu} \frac{\partial g}{\partial n_z}(x, z_{\nu}) g(z_{\nu}, y).$$





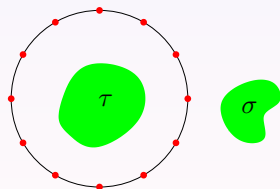
# Approximation per Green'scher Formel

Randintegralmethode: Kernfunktion  $g$  erfüllt Green'sche Formel

$$g(x, y) = \int_{\partial\omega} g(x, z) \frac{\partial g}{\partial n_z}(z, y) dz - \int_{\partial\omega} \frac{\partial g}{\partial n_z}(x, z) g(z, y) dz.$$

Idee: Approximiere Randintegral per Quadraturverfahren.

$$g(x, y) \approx \sum_{\nu=1}^k w_{\nu} g(x, z_{\nu}) \frac{\partial g}{\partial n_z}(z_{\nu}, y) - \sum_{\nu=1}^k w_{\nu} \frac{\partial g}{\partial n_z}(x, z_{\nu}) g(z_{\nu}, y).$$



**Ergebnis:**  $d$ -dimensionales Gebiet reduziert auf  $(d - 1)$ -dimensionalen Rand.  
 $\rightarrow$  Rang  $k \sim m^{d-1}$  statt  $k \sim m^d$

# Matrix-Arithmetik

**Erfolgreich:** Schnelle Arithmetik für hierarchische Matrizen, Multiplikation, Inverse, Faktorisierung in  $\sim nk^2 \log^2 n$  Operationen.

**Ziel:** Arithmetik für  $\mathcal{H}^2$ -Matrizen in  $\sim nk^2$  Operationen.

# Matrix-Arithmetik

**Erfolgreich:** Schnelle Arithmetik für hierarchische Matrizen, Multiplikation, Inverse, Faktorisierung in  $\sim nk^2 \log^2 n$  Operationen.

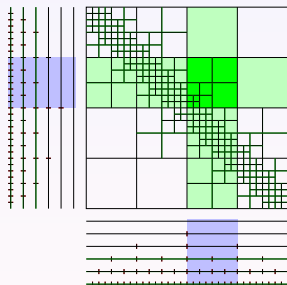
**Ziel:** Arithmetik für  $\mathcal{H}^2$ -Matrizen in  $\sim nk^2 \log n$  Operationen.

# Matrix-Arithmetik

**Erfolgreich:** Schnelle Arithmetik für hierarchische Matrizen, Multiplikation, Inverse, Faktorisierung in  $\sim nk^2 \log^2 n$  Operationen.

**Ziel:** Arithmetik für  $\mathcal{H}^2$ -Matrizen in  $\sim nk^2 \log n$  Operationen.

**Ansatz:** Arithmetische Operationen per Matrix-Vektor-Multiplikation und lokalem Niedrigrang-Update  $G|_{\hat{\tau} \times \hat{\sigma}} + AB^*$ .



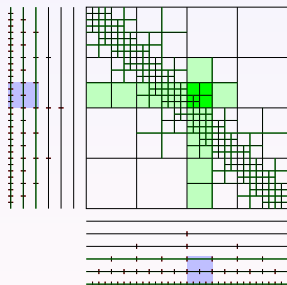
**MVM:**  $\sim k(\#\hat{\tau} + \#\hat{\sigma})$  Operationen genügen.

# Matrix-Arithmetik

**Erfolgreich:** Schnelle Arithmetik für hierarchische Matrizen, Multiplikation, Inverse, Faktorisierung in  $\sim nk^2 \log^2 n$  Operationen.

**Ziel:** Arithmetik für  $\mathcal{H}^2$ -Matrizen in  $\sim nk^2 \log n$  Operationen.

**Ansatz:** Arithmetische Operationen per Matrix-Vektor-Multiplikation und lokalem Niedrigrang-Update  $G|_{\hat{\tau} \times \hat{\sigma}} + AB^*$ .



**MVM:**  $\sim k(\#\hat{\tau} + \#\hat{\sigma})$  Operationen genügen.

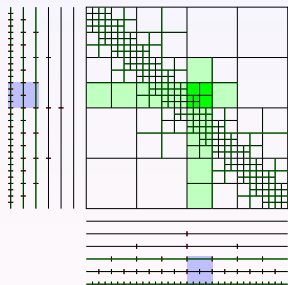
**LLU:** Neuer Algorithmus lässt Aufwand  $\sim k^2(\#\hat{\tau} + \#\hat{\sigma})$  erhoffen.

# Matrix-Arithmetik

**Erfolgreich:** Schnelle Arithmetik für hierarchische Matrizen, Multiplikation, Inverse, Faktorisierung in  $\sim nk^2 \log^2 n$  Operationen.

**Ziel:** Arithmetik für  $\mathcal{H}^2$ -Matrizen in  $\sim nk^2 \log n$  Operationen.

**Ansatz:** Arithmetische Operationen per Matrix-Vektor-Multiplikation und lokalem Niedrigrang-Update  $G|_{\hat{\tau} \times \hat{\sigma}} + AB^*$ .



**MVM:**  $\sim k(\#\hat{\tau} + \#\hat{\sigma})$  Operationen genügen.

**LLU:** Neuer Algorithmus lässt Aufwand  $\sim k^2(\#\hat{\tau} + \#\hat{\sigma})$  erhoffen.

**Potential:** Effiziente Vorkonditionierer für Differentialgleichungen mit un stetigen oder anisotropen Koeffizienten sowie für Integralgleichungen.

# Matrix-Galerkin-Verfahren

**Ziel:** Löse Matrixgleichungen  $AX = B$ ,  $AX + XB = C$ ,  $AXA = B$ .

**Idee:** Suche Näherungslösung  $\tilde{X}$  in Raum der  $\mathcal{H}^2$ -Matrizen,

$$\tilde{X} = \sum_{b=(\tau,\sigma)} V_\tau S_b W_\sigma^*.$$

# Matrix-Galerkin-Verfahren

**Ziel:** Löse Matrixgleichungen  $AX = B$ ,  $AX + XB = C$ ,  $AXA = B$ .

**Idee:** Suche Näherungslösung  $\tilde{X}$  in Raum der  $\mathcal{H}^2$ -Matrizen,

$$\tilde{X} = \sum_{b=(\tau,\sigma)} V_\tau S_b W_\sigma^*.$$

**Umsetzung:** Galerkin-Verfahren im Raum der Matrizen.

$$\langle A\tilde{X}, V_\alpha S_c W_\beta^* \rangle_F = \langle B, V_\alpha S_c W_\beta^* \rangle_F \quad \text{für alle } c = (\alpha, \beta), S_c \in \mathbb{R}^{k \times k}.$$



# Matrix-Galerkin-Verfahren

**Ziel:** Löse Matrixgleichungen  $AX = B$ ,  $AX + XB = C$ ,  $AXA = B$ .

**Idee:** Suche Näherungslösung  $\tilde{X}$  in Raum der  $\mathcal{H}^2$ -Matrizen,

$$\tilde{X} = \sum_{b=(\tau,\sigma)} V_\tau S_b W_\sigma^*.$$

**Umsetzung:** Galerkin-Verfahren im Raum der Matrizen.

$$\langle V_\alpha^* A \tilde{X} W_\beta, S_c \rangle_F = \langle V_\alpha^* B W_\beta, S_c \rangle_F \quad \text{für alle } c = (\alpha, \beta), S_c \in \mathbb{R}^{k \times k}.$$

# Matrix-Galerkin-Verfahren

**Ziel:** Löse Matrixgleichungen  $AX = B$ ,  $AX + XB = C$ ,  $AXA = B$ .

**Idee:** Suche Näherungslösung  $\tilde{X}$  in Raum der  $\mathcal{H}^2$ -Matrizen,

$$\tilde{X} = \sum_{b=(\tau,\sigma)} V_\tau S_b W_\sigma^*.$$

**Umsetzung:** Galerkin-Verfahren im Raum der Matrizen.

$$V_\alpha^* A \tilde{X} W_\beta = V_\alpha^* B W_\beta \quad \text{für alle } c = (\alpha, \beta)$$

# Matrix-Galerkin-Verfahren

**Ziel:** Löse Matrixgleichungen  $AX = B$ ,  $AX + XB = C$ ,  $AXA = B$ .

**Idee:** Suche Näherungslösung  $\tilde{X}$  in Raum der  $\mathcal{H}^2$ -Matrizen,

$$\tilde{X} = \sum_{b=(\tau,\sigma)} V_\tau S_b W_\sigma^*.$$

**Umsetzung:** Galerkin-Verfahren im Raum der Matrizen.

$$\sum_{b=(\tau,\sigma)} \underbrace{V_\alpha^* A V_\tau}_{=: Y_{\alpha\tau}} S_b \underbrace{W_\sigma^* W_\beta}_{=: Z_{\beta\sigma}^*} = \underbrace{V_\alpha^* B W_\beta}_{=: B_{\alpha\beta}} \quad \text{für alle } c = (\alpha, \beta)$$

# Matrix-Galerkin-Verfahren

**Ziel:** Löse Matrixgleichungen  $AX = B$ ,  $AX + XB = C$ ,  $AXA = B$ .

**Idee:** Suche Näherungslösung  $\tilde{X}$  in Raum der  $\mathcal{H}^2$ -Matrizen,

$$\tilde{X} = \sum_{b=(\tau,\sigma)} V_\tau S_b W_\sigma^*.$$

**Umsetzung:** Galerkin-Verfahren im Raum der Matrizen.

$$\sum_{b=(\tau,\sigma)} Y_{\alpha\tau} S_b Z_{\beta\sigma}^* = B_{\alpha\beta} \quad \text{für alle } c = (\alpha, \beta)$$

# Matrix-Galerkin-Verfahren

**Ziel:** Löse Matrixgleichungen  $AX = B$ ,  $AX + XB = C$ ,  $AXA = B$ .

**Idee:** Suche Näherungslösung  $\tilde{X}$  in Raum der  $\mathcal{H}^2$ -Matrizen,

$$\tilde{X} = \sum_{b=(\tau,\sigma)} V_\tau S_b W_\sigma^*.$$

**Umsetzung:** Galerkin-Verfahren im Raum der Matrizen.

$$\sum_{b=(\tau,\sigma)} Y_{\alpha\tau} S_b Z_{\beta\sigma}^* = B_{\alpha\beta} \quad \text{für alle } c = (\alpha, \beta)$$

**Ergebnis:** Gleichungssystem mit Kopplungsmatrizen als Unbekannten. Eigenschaften übertragen sich: Schwachbesetzt, positiv definit, Spektralschranken.  $\rightarrow$  Verwende iterative Lösungsverfahren.

# Übersicht

- 1 Einleitung
- 2  $\mathcal{H}^2$ -Matrizen
- 3 Aktuelle Projekte
- 4 Zusammenfassung**

# Methoden

**Ziel:** Effiziente Handhabung vollbesetzter Matrizen.

**Hierarchische Matrizen:** Blockweise Approximation niedrigen Rangs, Speicherbedarf  $\mathcal{O}(nk \log n)$ .

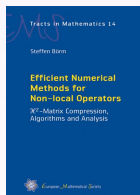
**$\mathcal{H}^2$ -Matrizen:** Blockweise Approximation durch Clusterbasen, Speicherbedarf  $\mathcal{O}(nk)$ .

**Arithmetik:** Multiplikation, Invertierung, Faktorisierung.

**Matrix-Gleichungen:** Galerkin-Verfahren im Raum der  $\mathcal{H}^2$ -Matrizen.

**Software:** HLib und neue H2Lib.

**Literatur:** S. Börm, Efficient Numerical Methods for Non-local Operators (2010)



# Anwendungsgebiete

**Integralgleichungen:** Kompression der Steifigkeitsmatrix, effiziente Vorkonditionierer für Randelementmethode.

**Elliptische Differentialgleichungen:** Vorkonditionierer, insbesondere bei Problemen mit unstetigen und anisotropen Koeffizienten.

**Steuerungstheorie:** Lyapunov-, Sylvester-, Riccati-Gleichungen.