Hybrid compression of boundary element matrices for high-frequency Helmholtz problems

Steffen Börm
with C. Börst and J. M. Melenk

Christian-Albrechts-Universität zu Kiel

Numerical Analysis of Complex PDE Models in the Sciences
Erwin-Schrödinger-Institut, Wien, 2018

Overview

Overview

Helmholtz equation

Goal: Solve the Helmholtz equation

$$\begin{aligned} \Delta u(x) + \kappa^2 u(x) &= 0 && \text{for all } x \in \Omega \subseteq \mathbb{R}^3, \\ u(x) &= f(x) && \text{for all } x \in \partial\Omega. \end{aligned}$$

Approach: Boundary integral formulation

$$u(x) = \int_{\partial\Omega} g(x,y) \frac{\partial u}{\partial n}(y)\, dy - \int_{\partial\Omega} \frac{\partial g}{\partial n(y)}(x,y) u(y)\, dy \quad \text{for all } x \in \Omega$$

with fundamental solution

$$g(x,y) = \frac{\exp(\iota\kappa \|x-y\|)}{4\pi \|x-y\|}.$$

Discretization

Galerkin's method with finite-element bases $(\varphi_i)_{i=1}^n$ and $(\psi_j)_{j=1}^n$ leads to matrix $G \in \mathbb{C}^{n \times n}$ with entries

$$g_{ij} = \int_{\partial\Omega} \varphi_i(x) \int_{\partial\Omega} g(x,y)\psi_j(y)\,dy\,dx, \qquad \text{for all } i,j \in [1:n].$$

Challenges:

- G is not sparse.

Discretization

Galerkin's method with finite-element bases $(\varphi_i)_{i=1}^n$ and $(\psi_j)_{j=1}^n$ leads to matrix $G \in \mathbb{C}^{n \times n}$ with entries

$$g_{ij} = \int_{\partial\Omega} \varphi_i(x) \int_{\partial\Omega} g(x,y)\psi_j(y)\,dy\,dx, \qquad \text{for all } i,j \in [1:n].$$

Challenges:

- G is not sparse.
- We are interested in the high-frequency case $\kappa^2 \sim n$, so the kernel function g oscillates rapidly.

Overview

Low-rank approximation

Standard approach: Approximate g by degenerate kernel,

$$g(x,y) \approx \sum_{\nu=1}^{k} v_\nu(x) \sum_{\mu=1}^{k} s_{\nu\mu} \overline{w_\mu(y)}$$

Low-rank approximation

Standard approach: Approximate g by degenerate kernel,

$$g(x,y) \approx \sum_{\nu=1}^{k} v_\nu(x) \sum_{\mu=1}^{k} s_{\nu\mu} \overline{w_\mu(y)} = \sum_{\nu,\mu=1}^{k} s_{\nu\mu}\, v_\nu \otimes \overline{w_\mu}(x,y).$$

Low-rank approximation

Standard approach: Approximate g by degenerate kernel,

$$g(x,y) \approx \sum_{\nu=1}^{k} v_\nu(x) \sum_{\mu=1}^{k} s_{\nu\mu} \overline{w_\mu(y)} = \sum_{\nu,\mu=1}^{k} s_{\nu\mu}\; v_\nu \otimes \overline{w_\mu}(x,y).$$

Discretization yields rank-k matrix factorization

$$G \approx VSW^*, \quad V_{i\nu} = \int_{\partial\Omega} \varphi_i(x) v_\nu(x)\,dx, \quad W_{j\mu} = \int_{\partial\Omega} \psi_j(y) w_\mu(y)\,dy.$$

Low-rank approximation

Standard approach: Approximate g by degenerate kernel,

$$g(x,y) \approx \sum_{\nu=1}^{k} v_\nu(x) \sum_{\mu=1}^{k} s_{\nu\mu} \overline{w_\mu(y)} = \sum_{\nu,\mu=1}^{k} s_{\nu\mu}\, v_\nu \otimes \overline{w_\mu}(x,y).$$

Discretization yields rank-k matrix factorization

$$G \approx VSW^*, \quad V_{i\nu} = \int_{\partial\Omega} \varphi_i(x) v_\nu(x)\,dx, \quad W_{j\mu} = \int_{\partial\Omega} \psi_j(y) w_\mu(y)\,dy.$$

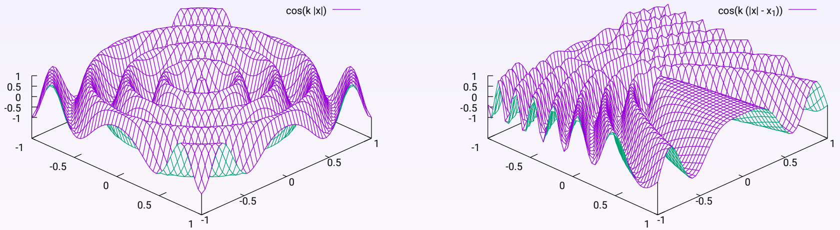
Challenges:

- Degenerate functions converge only locally.
- In standard methods, oscillations lead to large rank k.

Directional approximation

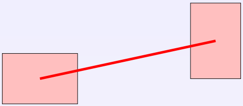
Idea: Divide spherical wave by plane wave in direction c, $\|c\| = 1$:

$$g(x,y) = \exp(\iota\kappa\langle x-y, c\rangle) \underbrace{\frac{\exp(\iota\kappa(\|x-y\| - \langle x-y, c\rangle))}{4\pi\|x-y\|}}_{=g_c(x,y)}.$$



✎ Brandt (1991), Engquist/Ying (2007), Messner/Schanz/Darve (2012), Bebendorf/Kuske/Venn (2015)

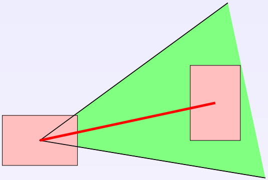
Admissibility conditions



Local approximation: Since approximation works only locally, we have to identify suitable subsets of $\mathbb{R}^3 \times \mathbb{R}^3$.
Let $B_t, B_s \in\subseteq \mathbb{R}^3$ be axis-parallel boxes with centers m_t, m_s.
g_c is smooth in $B_t \times B_s$ if

$$\kappa \left\| \frac{m_t - m_s}{\|m_t - m_s\|} - c \right\| \lesssim \frac{1}{\max\{\mathrm{diam}(B_t), \mathrm{diam}(B_s)\}},$$

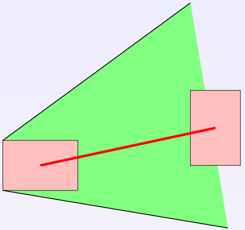
Admissibility conditions



Local approximation: Since approximation works only locally, we have to identify suitable subsets of $\mathbb{R}^3 \times \mathbb{R}^3$.
Let $B_t, B_s \in\subseteq \mathbb{R}^3$ be axis-parallel boxes with centers m_t, m_s.
g_c is smooth in $B_t \times B_s$ if

$$\kappa \left\| \frac{m_t - m_s}{\|m_t - m_s\|} - c \right\| \lesssim \frac{1}{\max\{\operatorname{diam}(B_t), \operatorname{diam}(B_s)\}},$$

Admissibility conditions

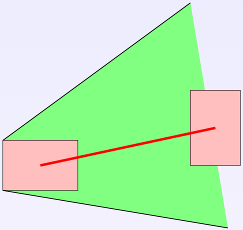


Local approximation: Since approximation works only locally, we have to identify suitable subsets of $\mathbb{R}^3 \times \mathbb{R}^3$.
Let $B_t, B_s \in\subseteq \mathbb{R}^3$ be axis-parallel boxes with centers m_t, m_s.
g_c is smooth in $B_t \times B_s$ if

$$\kappa \left\| \frac{m_t - m_s}{\|m_t - m_s\|} - c \right\| \lesssim \frac{1}{\max\{\operatorname{diam}(B_t), \operatorname{diam}(B_s)\}},$$

$$\kappa \max\{\operatorname{diam}^2(B_t), \operatorname{diam}^2(B_s)\} \lesssim \operatorname{dist}(B_t, B_s),$$

Admissibility conditions



Local approximation: Since approximation works only locally, we have to identify suitable subsets of $\mathbb{R}^3 \times \mathbb{R}^3$.
Let $B_t, B_s \in\subseteq \mathbb{R}^3$ be axis-parallel boxes with centers m_t, m_s.
g_c is smooth in $B_t \times B_s$ if

$$\kappa \left\| \frac{m_t - m_s}{\|m_t - m_s\|} - c \right\| \lesssim \frac{1}{\max\{\mathrm{diam}(B_t), \mathrm{diam}(B_s)\}},$$

$$\kappa \max\{\mathrm{diam}^2(B_t), \mathrm{diam}^2(B_s)\} \lesssim \mathrm{dist}(B_t, B_s),$$

$$\max\{\mathrm{diam}(B_t), \mathrm{diam}(B_s)\} \lesssim \mathrm{dist}(B_t, B_s).$$

Directional interpolation

Idea: Choose admissible boxes $B_t, B_s \subseteq \mathbb{R}^3$ and interpolate g_c.

$$g_c(x, y) \approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \mathcal{L}_{t,\nu}(x) g_c(\xi_{t,\nu}, \xi_{s,\mu}) \mathcal{L}_{s,\mu}(y),$$

$$g(x, y) = \exp(\iota\kappa\langle x - y, c\rangle) g_c(x, y)$$

Directional interpolation

Idea: Choose admissible boxes $B_t, B_s \subseteq \mathbb{R}^3$ and interpolate g_c.

$$g_c(x,y) \approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \mathcal{L}_{t,\nu}(x) g_c(\xi_{t,\nu}, \xi_{s,\mu}) \mathcal{L}_{s,\mu}(y),$$

$$g(x,y) = \exp(\iota\kappa\langle x-y, c\rangle) g_c(x,y)$$

$$\approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \underbrace{\exp(\langle x, c\rangle)\mathcal{L}_{t,\nu}(x)}_{=:v_{tc,\nu}(x)} \underbrace{g_c(\xi_{t,\nu}, \xi_{s,\mu})}_{=:s_{ts,\nu\mu}} \underbrace{\overline{\exp(\langle y, c\rangle)\mathcal{L}_{s,\mu}(y)}}_{=:\overline{v_{sc,\mu}(y)}}$$

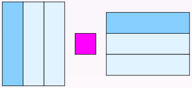
Directional interpolation

Idea: Choose admissible boxes $B_t, B_s \subseteq \mathbb{R}^3$ and interpolate g_c.

$$\begin{aligned}
g_c(x,y) &\approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \mathcal{L}_{t,\nu}(x) g_c(\xi_{t,\nu}, \xi_{s,\mu}) \mathcal{L}_{s,\mu}(y), \\
g(x,y) &= \exp(\iota\kappa\langle x-y, c\rangle) g_c(x,y) \\
&\approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \underbrace{\exp(\langle x,c\rangle)\mathcal{L}_{t,\nu}(x)}_{=:v_{tc,\nu}(x)} \underbrace{g_c(\xi_{t,\nu}, \xi_{s,\mu})}_{=:s_{ts,\nu\mu}} \underbrace{\overline{\exp(\langle y,c\rangle)\mathcal{L}_{s,\mu}(y)}}_{=:\overline{v_{sc,\mu}(y)}}
\end{aligned}$$

$\mathcal{DH}^2$-matrix: If B_t, B_s correspond to clusters $t, s \subseteq [1:n]$, we obtain a low-rank approximation of a submatrix

$$G|_{t\times s} \approx V_{tc} S_{ts} V_{sc}^*,$$



coupling matrix S_{ts} requires only $\mathcal{O}(k^2)$ coefficients.

Directional interpolation

Idea: Choose admissible boxes $B_t, B_s \subseteq \mathbb{R}^3$ and interpolate g_c.

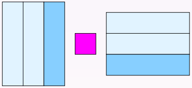
$$g_c(x,y) \approx \sum_{\nu=1}^{k}\sum_{\mu=1}^{k} \mathcal{L}_{t,\nu}(x) g_c(\xi_{t,\nu},\xi_{s,\mu}) \mathcal{L}_{s,\mu}(y),$$

$$g(x,y) = \exp(\iota\kappa\langle x-y,c\rangle) g_c(x,y)$$

$$\approx \sum_{\nu=1}^{k}\sum_{\mu=1}^{k} \underbrace{\exp(\langle x,c\rangle)\mathcal{L}_{t,\nu}(x)}_{=:v_{tc,\nu}(x)} \underbrace{g_c(\xi_{t,\nu},\xi_{s,\mu})}_{=:s_{ts,\nu\mu}} \underbrace{\overline{\exp(\langle y,c\rangle)\mathcal{L}_{s,\mu}(y)}}_{=:\overline{v_{sc,\mu}(y)}}$$

$\mathcal{DH}^2$-matrix: If B_t, B_s correspond to clusters $t, s \subseteq [1:n]$, we obtain a low-rank approximation of a submatrix

$$G|_{t\times s} \approx V_{tc} S_{ts} V_{sc}^{*},$$



coupling matrix S_{ts} requires only $\mathcal{O}(k^2)$ coefficients.

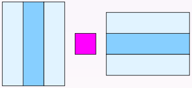
Directional interpolation

Idea: Choose admissible boxes $B_t, B_s \subseteq \mathbb{R}^3$ and interpolate g_c.

$$
\begin{aligned}
g_c(x,y) &\approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \mathcal{L}_{t,\nu}(x) g_c(\xi_{t,\nu}, \xi_{s,\mu}) \mathcal{L}_{s,\mu}(y), \\
g(x,y) &= \exp(\iota\kappa\langle x-y, c\rangle) g_c(x,y) \\
&\approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \underbrace{\exp(\langle x, c\rangle)\mathcal{L}_{t,\nu}(x)}_{=:v_{tc,\nu}(x)} \underbrace{g_c(\xi_{t,\nu}, \xi_{s,\mu})}_{=:s_{ts,\nu\mu}} \underbrace{\overline{\exp(\langle y, c\rangle)\mathcal{L}_{s,\mu}(y)}}_{=:\overline{v_{sc,\mu}(y)}}
\end{aligned}
$$

$\mathcal{DH}^2$-matrix: If B_t, B_s correspond to clusters $t, s \subseteq [1:n]$, we obtain a low-rank approximation of a submatrix

$$
G|_{t\times s} \approx V_{tc} S_{ts} V_{sc}^*,
$$



coupling matrix S_{ts} requires only $\mathcal{O}(k^2)$ coefficients.

Nested directional bases

Problem: To satisfy the first admissibility condition, we require $\sim \kappa^2 \operatorname{diam}^2(B_t)$ directions for each cluster t.
$\to$ Storing (V_{tc}) directly requires $\mathcal{O}(nk\kappa^2 \log n)$ units of storage.
$\to$ $\mathcal{O}(n^2 k \log n)$ for high frequencies, $\kappa^2 \sim n$.

Nested directional bases

Problem: To satisfy the first admissibility condition, we require $\sim \kappa^2 \operatorname{diam}^2(B_t)$ directions for each cluster t.
$\rightarrow$ Storing (V_{tc}) directly requires $\mathcal{O}(nk\kappa^2 \log n)$ units of storage.
$\rightarrow$ $\mathcal{O}(n^2 k \log n)$ for high frequencies, $\kappa^2 \sim n$.

Idea: Express V_{tc} in terms of $V_{t'c'}$ for sons t' of t.

$$v_{tc,\nu}(x) = \exp(\langle x, c \rangle)\mathcal{L}_{t,\nu}(x)$$

Nested directional bases

Problem: To satisfy the first admissibility condition, we require $\sim \kappa^2 \operatorname{diam}^2(B_t)$ directions for each cluster t.
$\rightarrow$ Storing (V_{tc}) directly requires $\mathcal{O}(nk\kappa^2 \log n)$ units of storage.
$\rightarrow$ $\mathcal{O}(n^2 k \log n)$ for high frequencies, $\kappa^2 \sim n$.

Idea: Express V_{tc} in terms of $V_{t'c'}$ for sons t' of t.

$$v_{tc,\nu}(x) = \exp(\langle x, c' \rangle) \exp(\langle x, c - c' \rangle) \, \mathcal{L}_{t,\nu}(x)$$

Nested directional bases

Problem: To satisfy the first admissibility condition, we require $\sim \kappa^2 \operatorname{diam}^2(B_t)$ directions for each cluster t.
$\to$ Storing (V_{tc}) directly requires $\mathcal{O}(nk\kappa^2 \log n)$ units of storage.
$\to$ $\mathcal{O}(n^2 k \log n)$ for high frequencies, $\kappa^2 \sim n$.

Idea: Express V_{tc} in terms of $V_{t'c'}$ for sons t' of t.

$$\begin{aligned} v_{tc,\nu}(x) &= \exp(\langle x, c' \rangle) \exp(\langle x, c - c' \rangle)\, \mathcal{L}_{t,\nu}(x) \\ &\approx \exp(\langle x, c' \rangle) \sum_{\nu'=1}^{k} \exp(\langle \xi_{t',\nu'}, c - c' \rangle) \mathcal{L}_{t,\nu}(\xi_{t',\nu'})\, \mathcal{L}_{t',\nu'}(x) \end{aligned}$$

Nested directional bases

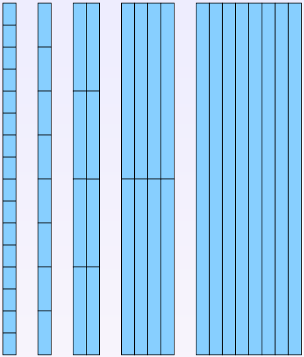
Problem: To satisfy the first admissibility condition, we require $\sim \kappa^2 \operatorname{diam}^2(B_t)$ directions for each cluster t.
$\to$ Storing (V_{tc}) directly requires $\mathcal{O}(nk\kappa^2 \log n)$ units of storage.
$\to$ $\mathcal{O}(n^2 k \log n)$ for high frequencies, $\kappa^2 \sim n$.

Idea: Express V_{tc} in terms of $V_{t'c'}$ for sons t' of t.

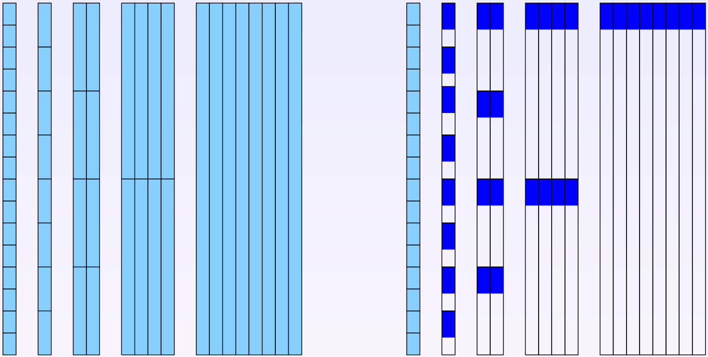
$$\begin{aligned} v_{tc,\nu}(x) &= \exp(\langle x, c' \rangle) \exp(\langle x, c - c' \rangle)\, \mathcal{L}_{t,\nu}(x) \\ &\approx \exp(\langle x, c' \rangle) \sum_{\nu'=1}^{k} \exp(\langle \xi_{t',\nu'}, c - c' \rangle) \mathcal{L}_{t,\nu}(\xi_{t',\nu'})\, \mathcal{L}_{t',\nu'}(x) \\ &= \sum_{\nu'=1}^{k} e_{t'c,\nu'\nu} v_{t'c',\nu'}(x) \end{aligned}$$

Approach: Store only $k \times k$ transfer matrices $E_{tc'}$ for non-leaf clusters.

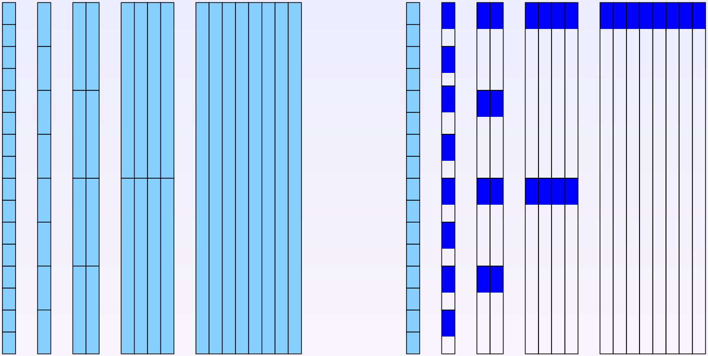
Complexity



Complexity



Complexity

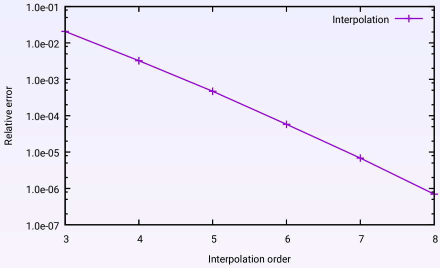


Result: Complexity $\mathcal{O}(nk + \kappa^2 k^2 \log n)$,
$\mathcal{O}(nk^2 \log n)$ in the high-frequency case with $\kappa^2 \sim n$

✐ Messner/Schanz/Darve (2012), B. (2017)

Directional interpolation: Convergence

Experiment: Compress Helmholtz Galerkin matrix corresponding to the unit sphere approximated by with $n = 32\,768$ triangles, $\kappa h \approx 0.6$.

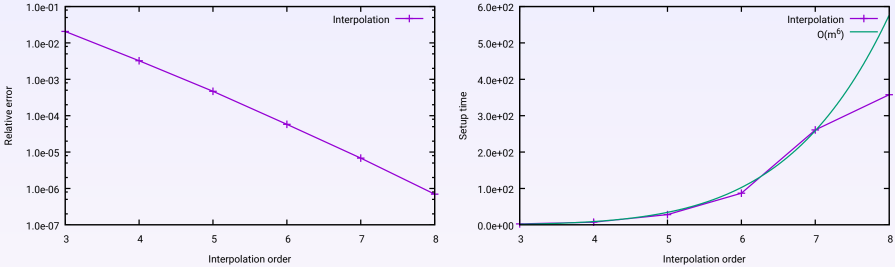


Observations:

- Exponential convergence.

Directional interpolation: Convergence

Experiment: Compress Helmholtz Galerkin matrix corresponding to the unit sphere approximated by with $n = 32\,768$ triangles, $\kappa h \approx 0.6$.

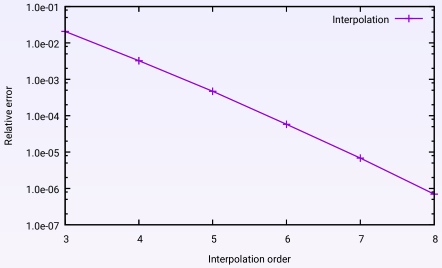


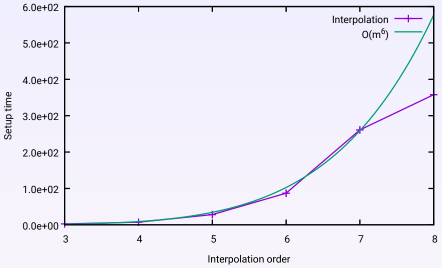
Observations:

- Exponential convergence.
- Setup time proportional to k^2, in the range of a few minutes.

Directional interpolation: Convergence

Experiment: Compress Helmholtz Galerkin matrix corresponding to the unit sphere approximated by with $n = 32\,768$ triangles, $\kappa h \approx 0.6$.





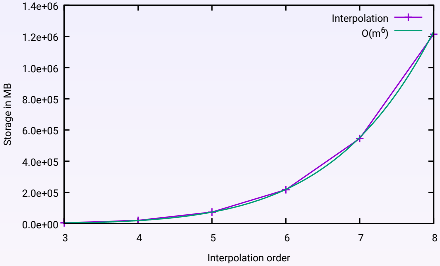
Observations:

- Exponential convergence.
- Setup time proportional to k^2, in the range of a few minutes.

✎ Messner/Schanz/Darve (2012), B./Melenk (2017)

Directional interpolation: Storage

Experiment: Compress Helmholtz Galerkin matrix corresponding to the unit sphere approximated by with $n = 32\,768$ triangles.

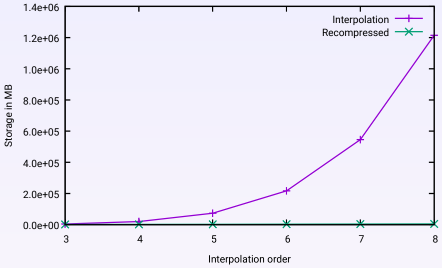


Observation: 1.2 TB for 32 768 degrees of freedom is inacceptable.

Overview

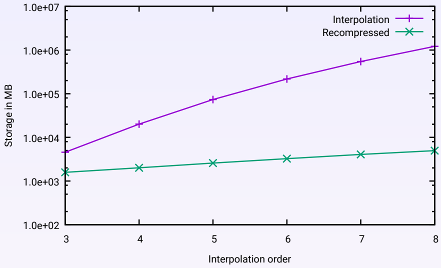
Algebraic recompression

Experiment: Apply algebraic recompression to the approximation constructed by directional interpolation.



Algebraic recompression

Experiment: Apply algebraic recompression to the approximation constructed by directional interpolation.

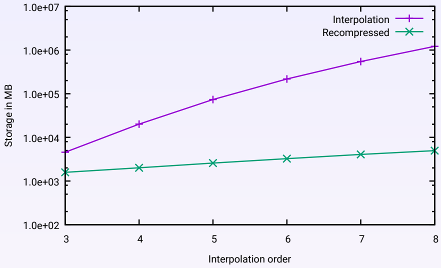


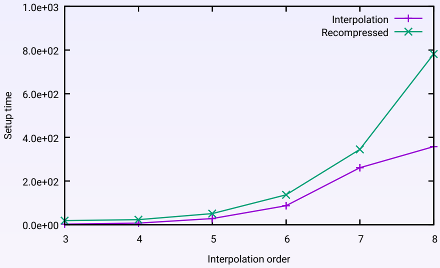
Observations:

- Storage reduced by more than a factor of more than 100.

Algebraic recompression

Experiment: Apply algebraic recompression to the approximation constructed by directional interpolation.



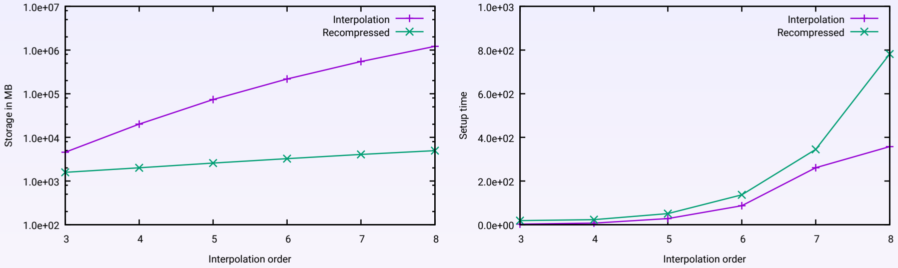


Observations:

- Storage reduced by more than a factor of more than 100.
- Setup time approximately doubled.

Algebraic recompression

Experiment: Apply algebraic recompression to the approximation constructed by directional interpolation.



Observations:

- Storage reduced by more than a factor of more than 100.
- Setup time approximately doubled.

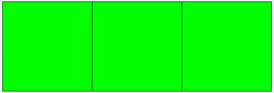
✎ B./Börst (in preparation)

Uniform approximation

Goal: Replace V_{tc} with a basis Q_{tc} of lower rank.

Since V_{tc} is required by multiple blocks, we have to approximate

$$\begin{pmatrix} V_{tc} S_{ts_1} V_{s_1 c}^* & \cdots & V_{tc} S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix}$$

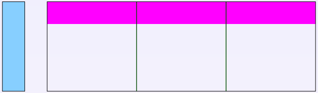


Uniform approximation

Goal: Replace V_{tc} with a basis Q_{tc} of lower rank.

Since V_{tc} is required by multiple blocks, we have to approximate

$$V_{tc} \underbrace{\begin{pmatrix} S_{ts_1} V_{s_1 c}^* & \cdots & S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix}}_{=:B_{tc}^*}.$$



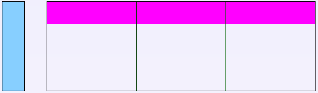
The matrix B_{tc} has only k columns. $\to$ Low-rank factorization.

Uniform approximation

Goal: Replace V_{tc} with a basis Q_{tc} of lower rank.

Since V_{tc} is required by multiple blocks, we have to approximate

$$V_{tc} \underbrace{\begin{pmatrix} S_{ts_1} V_{s_1 c}^* & \cdots & S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix}}_{=:B_{tc}^*}.$$



The matrix B_{tc} has only k columns. $\to$ Low-rank factorization.

Compression: Construct isometric matrix Q_{tc} of low rank such that

$$Q_{tc} Q_{tc}^* V_{tc} B_{tc}^* \approx V_{tc} B_{tc}^*,$$

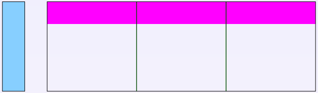
i.e., by SVD or rank-revealing QR.

Uniform approximation

Goal: Replace V_{tc} with a basis Q_{tc} of lower rank.

Since V_{tc} is required by multiple blocks, we have to approximate

$$V_{tc} \underbrace{\begin{pmatrix} S_{ts_1} V^*_{s_1 c} & \cdots & S_{ts_\ell} V^*_{s_\ell c} \end{pmatrix}}_{=:B^*_{tc}}.$$



The matrix B_{tc} has only k columns. $\to$ Low-rank factorization.

Compression: Construct isometric matrix Q_{tc} of low rank such that

$$Q_{tc} Q^*_{tc} V_{tc} B^*_{tc} \approx V_{tc} B^*_{tc},$$

i.e., by SVD or rank-revealing QR.

Challenge: B_{tc} has a large number of rows.

Basis weights

Goal: $Q_{tc} Q_{tc}^* V_{tc} B_{tc}^* \approx V_{tc} B_{tc}^*$.

Idea: Applying orthogonal transformations from the right does not change the approximation quality.

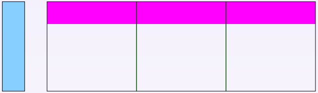
Basis weights

Goal: $Q_{tc} Q_{tc}^* V_{tc} B_{tc}^* \approx V_{tc} B_{tc}^*$.

Idea: Applying orthogonal transformations from the right does not change the approximation quality.

Basis weights: Construct QR factorizations $V_{sc} = P_{sc} R_{sc}$, $R_{sc} \in \mathbb{C}^{k \times k}$.

$$B_{tc}^* = \begin{pmatrix} S_{ts_1} V_{s_1 c}^* & \cdots & S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix}$$



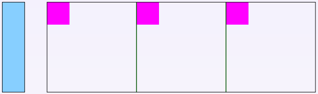
Basis weights

Goal: $Q_{tc} Q_{tc}^* V_{tc} B_{tc}^* \approx V_{tc} B_{tc}^*$.

Idea: Applying orthogonal transformations from the right does not change the approximation quality.

Basis weights: Construct QR factorizations $V_{sc} = P_{sc} R_{sc}$, $R_{sc} \in \mathbb{C}^{k \times k}$.

$$\begin{aligned} B_{tc}^* &= \begin{pmatrix} S_{ts_1} V_{s_1 c}^* & \cdots & S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix} \\ \to \widehat{B}_{tc}^* &= \begin{pmatrix} S_{ts_1} R_{s_1 c}^* & \cdots & S_{ts_\ell} R_{s_\ell c}^* \end{pmatrix}. \end{aligned}$$



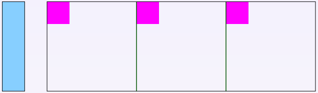
Basis weights

Goal: $Q_{tc} Q_{tc}^* V_{tc} B_{tc}^* \approx V_{tc} B_{tc}^*$.

Idea: Applying orthogonal transformations from the right does not change the approximation quality.

Basis weights: Construct QR factorizations $V_{sc} = P_{sc} R_{sc}$, $R_{sc} \in \mathbb{C}^{k \times k}$.

$$\begin{aligned} B_{tc}^* &= \begin{pmatrix} S_{ts_1} V_{s_1 c}^* & \cdots & S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix} \\ \to \widehat{B}_{tc}^* &= \begin{pmatrix} S_{ts_1} R_{s_1 c}^* & \cdots & S_{ts_\ell} R_{s_\ell c}^* \end{pmatrix}. \end{aligned}$$



Total weights: Construct thin QR factorization $\widehat{B}_{tc} = \widehat{P}_{tc} Z_{tc}$.

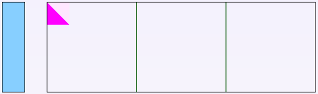
Basis weights

Goal: $Q_{tc}Q_{tc}^* V_{tc} B_{tc}^* \approx V_{tc} B_{tc}^*$.

Idea: Applying orthogonal transformations from the right does not change the approximation quality.

Basis weights: Construct QR factorizations $V_{sc} = P_{sc} R_{sc}$, $R_{sc} \in \mathbb{C}^{k \times k}$.

$$\begin{aligned} B_{tc}^* &= \begin{pmatrix} S_{ts_1} V_{s_1 c}^* & \cdots & S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix} \\ \to \widehat{B}_{tc}^* &= \begin{pmatrix} S_{ts_1} R_{s_1 c}^* & \cdots & S_{ts_\ell} R_{s_\ell c}^* \end{pmatrix}. \end{aligned}$$



Total weights: Construct thin QR factorization $\widehat{B}_{tc} = \widehat{P}_{tc} Z_{tc}$.

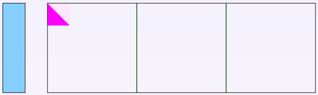
Basis weights

Goal: $Q_{tc} Q_{tc}^* V_{tc} B_{tc}^* \approx V_{tc} B_{tc}^*$.

Idea: Applying orthogonal transformations from the right does not change the approximation quality.

Basis weights: Construct QR factorizations $V_{sc} = P_{sc} R_{sc}$, $R_{sc} \in \mathbb{C}^{k \times k}$.

$$\begin{aligned} B_{tc}^* &= \begin{pmatrix} S_{ts_1} V_{s_1 c}^* & \cdots & S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix} \\ \to \widehat{B}_{tc}^* &= \begin{pmatrix} S_{ts_1} R_{s_1 c}^* & \cdots & S_{ts_\ell} R_{s_\ell c}^* \end{pmatrix}. \end{aligned}$$



Total weights: Construct thin QR factorization $\widehat{B}_{tc} = \widehat{P}_{tc} Z_{tc}$.

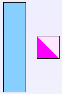
Compression: Find low-rank Q_{tc} with $Q_{tc} Q_{tc}^* V_{tc} Z_{tc}^* \approx V_{tc} Z_{tc}^*$, e.g., by singular value decomposition.

Error control achieved by weighting the matrices S_{ts}.

Nested bases

Goal: Obtain nested bases, i.e., transfer matrices $F_{t'c}$ with

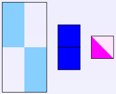
$$Q_{tc} = \begin{pmatrix} Q_{t_1 c_1} F_{t_1 c} \\ Q_{t_2 c_2} F_{t_2 c} \end{pmatrix}$$



Nested bases

Goal: Obtain nested bases, i.e., transfer matrices $F_{t'c}$ with

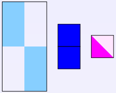
$$Q_{tc} = \begin{pmatrix} Q_{t_1 c_1} F_{t_1 c} \\ Q_{t_2 c_2} F_{t_2 c} \end{pmatrix} = \begin{pmatrix} Q_{t_1 c_1} & \\ & Q_{t_2 c_2} \end{pmatrix} \begin{pmatrix} F_{t_1 c} \\ F_{t_2 c} \end{pmatrix} = U_{tc} \widehat{Q}_{tc}$$



Nested bases

Goal: Obtain nested bases, i.e., transfer matrices $F_{t'c}$ with

$$Q_{tc} = \begin{pmatrix} Q_{t_1 c_1} F_{t_1 c} \\ Q_{t_2 c_2} F_{t_2 c} \end{pmatrix} = \begin{pmatrix} Q_{t_1 c_1} & \\ & Q_{t_2 c_2} \end{pmatrix} \begin{pmatrix} F_{t_1 c} \\ F_{t_2 c} \end{pmatrix} = U_{tc} \widehat{Q}_{tc}$$



Compression now takes the form

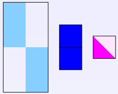
$$Q_{tc} Q_{tc}^* V_{tc} Z_{tc}^* \approx V_{tc} Z_{tc}^*,$$
$$\widehat{Q}_{tc} \widehat{Q}_{tc}^* U_{tc}^* V_{tc} Z_{tc}^* \approx U_{tc}^* V_{tc} Z_{tc}^*,$$
$$\widehat{Q}_{tc} \widehat{Q}_{tc}^* \widehat{V}_{tc} Z_{tc}^* \approx \widehat{V}_{tc} Z_{tc}^*$$

with the significantly smaller matrix $\widehat{V}_{tc} := U_{tc}^* V_{tc} \in \mathbb{C}^{(2\tilde{k}) \times k}$.

Nested bases

Goal: Obtain nested bases, i.e., transfer matrices $F_{t'c}$ with

$$Q_{tc} = \begin{pmatrix} Q_{t_1 c_1} F_{t_1 c} \\ Q_{t_2 c_2} F_{t_2 c} \end{pmatrix} = \begin{pmatrix} Q_{t_1 c_1} & \\ & Q_{t_2 c_2} \end{pmatrix} \begin{pmatrix} F_{t_1 c} \\ F_{t_2 c} \end{pmatrix} = U_{tc} \widehat{Q}_{tc}$$



Compression now takes the form

$$\begin{aligned} Q_{tc} Q_{tc}^* V_{tc} Z_{tc}^* &\approx V_{tc} Z_{tc}^*, \\ \widehat{Q}_{tc} \widehat{Q}_{tc}^* U_{tc}^* V_{tc} Z_{tc}^* &\approx U_{tc}^* V_{tc} Z_{tc}^*, \\ \widehat{Q}_{tc} \widehat{Q}_{tc}^* \widehat{V}_{tc} Z_{tc}^* &\approx \widehat{V}_{tc} Z_{tc}^* \end{aligned}$$

with the significantly smaller matrix $\widehat{V}_{tc} := U_{tc}^* V_{tc} \in \mathbb{C}^{(2\tilde{k}) \times k}$.

Result: Algebraic recompression in $\mathcal{O}(nk + \kappa^2 k^2 \log n)$ operations, $\mathcal{O}(nk^2 \log n)$ in high-frequency case.

Overview

Cross approximation

Problem: For high accuracies, the coupling matrices S_{ts} are large, e.g., $m = 8$ leads to dimension $8^3 = 512$.
$\rightarrow$ Construction of total weights Z_{tc} fairly slow.

Idea: Since S_{ts} corresponds to the smooth kernel function g_c, we may use cross approximation to obtain

$$S_{ts} \approx C_{ts} D_{ts}^*,$$

where C_{ts}, D_{ts} have a reduced number $\hat{k} \leq k$ of columns.

Cross approximation

Problem: For high accuracies, the coupling matrices S_{ts} are large, e.g., $m = 8$ leads to dimension $8^3 = 512$.
$\rightarrow$ Construction of total weights Z_{tc} fairly slow.

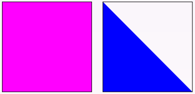
Idea: Since S_{ts} corresponds to the smooth kernel function g_c, we may use cross approximation to obtain

$$S_{ts} \approx C_{ts} D_{ts}^*,$$

where C_{ts}, D_{ts} have a reduced number $\hat{k} \leq k$ of columns.

Condensation: Again apply isometric transformations from the right: QR factorization yields $R_{sc} D_{ts} = P_{ts} \widehat{R}_{ts}$ with $\widehat{R}_{ts} \in \mathbb{C}^{\hat{k} \times \hat{k}}$, P_{ts} isometric.

$S_{ts} R_{sc}^*$



Cross approximation

Problem: For high accuracies, the coupling matrices S_{ts} are large, e.g., $m = 8$ leads to dimension $8^3 = 512$.
$\rightarrow$ Construction of total weights Z_{tc} fairly slow.

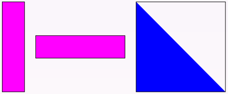
Idea: Since S_{ts} corresponds to the smooth kernel function g_c, we may use cross approximation to obtain

$$S_{ts} \approx C_{ts} D_{ts}^*,$$

where C_{ts}, D_{ts} have a reduced number $\hat{k} \leq k$ of columns.

Condensation: Again apply isometric transformations from the right: QR factorization yields $R_{sc} D_{ts} = P_{ts} \widehat{R}_{ts}$ with $\widehat{R}_{ts} \in \mathbb{C}^{\hat{k} \times \hat{k}}$, P_{ts} isometric.

$S_{ts} R_{sc}^* \approx C_{ts} D_{ts}^* R_{sc}^*$



Cross approximation

Problem: For high accuracies, the coupling matrices S_{ts} are large, e.g., $m = 8$ leads to dimension $8^3 = 512$.
$\rightarrow$ Construction of total weights Z_{tc} fairly slow.

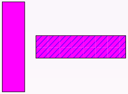
Idea: Since S_{ts} corresponds to the smooth kernel function g_c, we may use cross approximation to obtain

$$S_{ts} \approx C_{ts} D_{ts}^*,$$

where C_{ts}, D_{ts} have a reduced number $\hat{k} \leq k$ of columns.

Condensation: Again apply isometric transformations from the right: QR factorization yields $R_{sc} D_{ts} = P_{ts} \widehat{R}_{ts}$ with $\widehat{R}_{ts} \in \mathbb{C}^{\hat{k} \times \hat{k}}$, P_{ts} isometric.

$$S_{ts} R_{sc}^* \approx C_{ts} D_{ts}^* R_{sc}^* = C_{ts} (R_{sc} D_{ts})^*$$



Cross approximation

Problem: For high accuracies, the coupling matrices S_{ts} are large, e.g., $m = 8$ leads to dimension $8^3 = 512$.
$\rightarrow$ Construction of total weights Z_{tc} fairly slow.

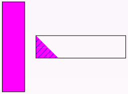
Idea: Since S_{ts} corresponds to the smooth kernel function g_c, we may use cross approximation to obtain

$$S_{ts} \approx C_{ts} D_{ts}^*,$$

where C_{ts}, D_{ts} have a reduced number $\hat{k} \leq k$ of columns.

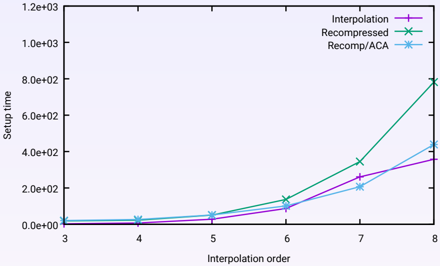
Condensation: Again apply isometric transformations from the right: QR factorization yields $R_{sc} D_{ts} = P_{ts} \widehat{R}_{ts}$ with $\widehat{R}_{ts} \in \mathbb{C}^{\hat{k} \times \hat{k}}$, P_{ts} isometric.

$$S_{ts} R_{sc}^* \approx C_{ts} D_{ts}^* R_{sc}^* = C_{ts} (R_{sc} D_{ts})^* = C_{ts} \widehat{R}_{ts}^* P_{ts}^*.$$



Algebraic compression with cross approximation

Experiment: Algebraic recompression of directional interpolation, coupling matrices approximated by cross approximation.

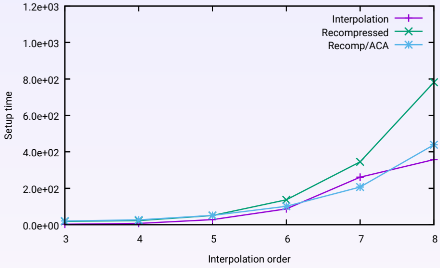


Observations:

- Storage reduction and accuracy as before.

Algebraic compression with cross approximation

Experiment: Algebraic recompression of directional interpolation, coupling matrices approximated by cross approximation.



Observations:

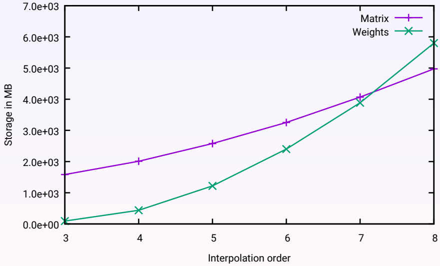
- Storage reduction and accuracy as before.
- Setup time only moderately increased compared to original.

Basis weights revisited

Problem: In order to compute the total weights Z_{tc}, we need a QR factorization of the matrix B_{tc},

$$B_{tc}^* = \begin{pmatrix} S_{ts_1} V_{s_1 c}^* & \cdots & S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix}.$$

The first step is to replace V_{sc} by a QR factorization $V_{sc} = P_{sc} R_{sc}$.
$\rightarrow$ We have to store R_{sc} for all clusters and directions.



Compressed basis weights

Idea: Consider products $S_{ts} V_{sc}^*$, exploit low-rank property of S_{ts}.

Approach: Find isometric P_{sc} of low rank with

$$P_{sc} P_{sc}^* V_{sc} \begin{pmatrix} S_{t_1 s}^* & \cdots & S_{t_\ell s}^* \end{pmatrix} \approx V_{sc} \begin{pmatrix} S_{t_1 s}^* & \cdots & S_{t_\ell s}^* \end{pmatrix}.$$

With $R_{sc} := P_{sc}^* V_{sc}$ we have $S_{ts} V_{sc}^* \approx S_{ts} R_{sc}^* P_{sc}^*$ for all submatrices.

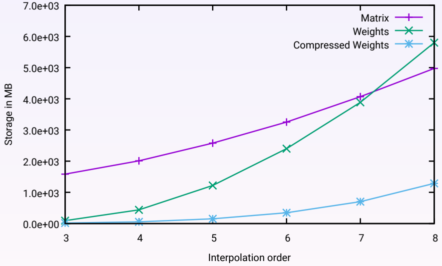
Compressed basis weights

Idea: Consider products $S_{ts} V_{sc}^*$, exploit low-rank property of S_{ts}.

Approach: Find isometric P_{sc} of low rank with

$$P_{sc} P_{sc}^* V_{sc} \begin{pmatrix} S_{t_1 s}^* & \cdots & S_{t_\ell s}^* \end{pmatrix} \approx V_{sc} \begin{pmatrix} S_{t_1 s}^* & \cdots & S_{t_\ell s}^* \end{pmatrix}.$$

With $R_{sc} := P_{sc}^* V_{sc}$ we have $S_{ts} V_{sc}^* \approx S_{ts} R_{sc}^* P_{sc}^*$ for all submatrices.



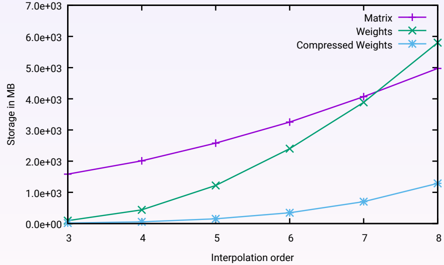
Compressed basis weights

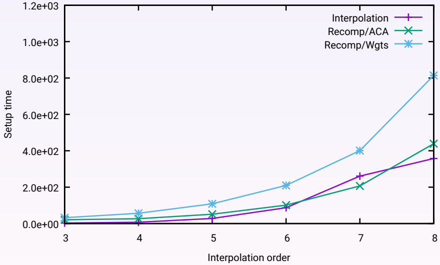
Idea: Consider products $S_{ts} V_{sc}^*$, exploit low-rank property of S_{ts}.

Approach: Find isometric P_{sc} of low rank with

$$P_{sc} P_{sc}^* V_{sc} \begin{pmatrix} S_{t_1 s}^* & \cdots & S_{t_\ell s}^* \end{pmatrix} \approx V_{sc} \begin{pmatrix} S_{t_1 s}^* & \cdots & S_{t_\ell s}^* \end{pmatrix}.$$

With $R_{sc} := P_{sc}^* V_{sc}$ we have $S_{ts} V_{sc}^* \approx S_{ts} R_{sc}^* P_{sc}^*$ for all submatrices.





Summary

Interpolation is fast and reliable, but needs too much storage.

Recompression based on rank-revealing factorizations can significantly reduce the storage requirements.

Modifications like cross approximation and compressed weights improve performance, particularly for high accuracies.