CROQuant: Complex Rank-One Quantization Algorithm, with Application to Butterfly Factorizations

Maël Chaumette, Rémi Gribonval, Elisa Riccietti

OCKHAM, Inria Lyon, LIP, ENS Lyon

Motivation: Reduce memory footprint by lowering numerical precision in numerical algebra (e.g. FFT) by exploiting **rescaling invariances.**

Rank-One Quantization problem

$$\min_{\hat{\boldsymbol{x}}\in\mathbb{CF}_t^m,\hat{\boldsymbol{y}}\in\mathbb{CF}_t^n} \left\|\boldsymbol{x}\boldsymbol{y}^H - \hat{\boldsymbol{x}}\hat{\boldsymbol{y}}^H\right\|_F^2$$

unquantized $\boldsymbol{x}\boldsymbol{y}^H$ — quantized $\hat{\boldsymbol{x}}\hat{\boldsymbol{y}}^H$

Q: how to efficiently solve the problem with quantized unknowns?

Optimally solved in the real case [1] by exploiting the rescaling invariance property $\boldsymbol{x}\boldsymbol{y}^H = (\lambda\boldsymbol{x})\left(\frac{1}{\overline{\lambda}}\boldsymbol{y}\right)^H$

Round-To-Nearest (RTN)

$$\hat{\boldsymbol{x}} = \mathrm{round}(\boldsymbol{x})$$
$$\hat{\boldsymbol{y}} = \mathrm{round}(\boldsymbol{y})$$

Separately maps each coefficient to their nearest neighbor in $\mathbb{CF}_t$

Fast but potentially not **optimal**

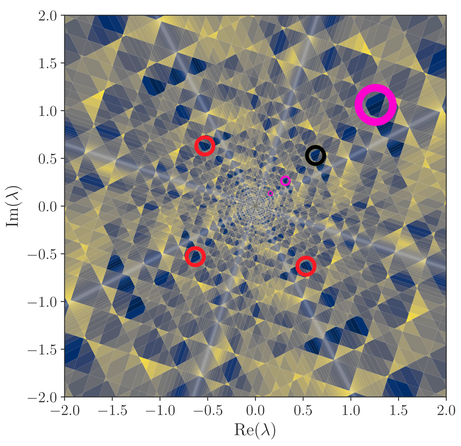
Key characterization

$$\hat{\boldsymbol{x}}^* = \mathrm{round}(\lambda^*\boldsymbol{x})$$
$$\hat{\boldsymbol{y}}^* = \mathrm{round}(\mu(\lambda^*)\boldsymbol{y})$$
$$\text{where } \lambda^* \in \arg\inf_{\lambda\in\mathbb{C}} f(\lambda)$$

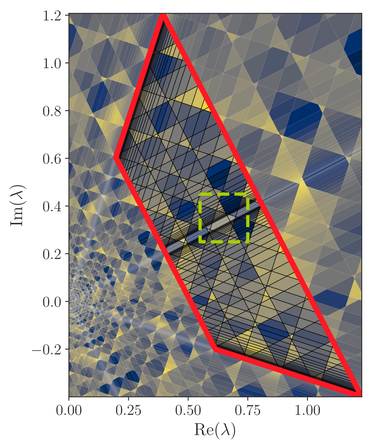
Reduction of a problem with $2(m+n)$ quantized variables to a one scalar problem

Geometrical properties of f

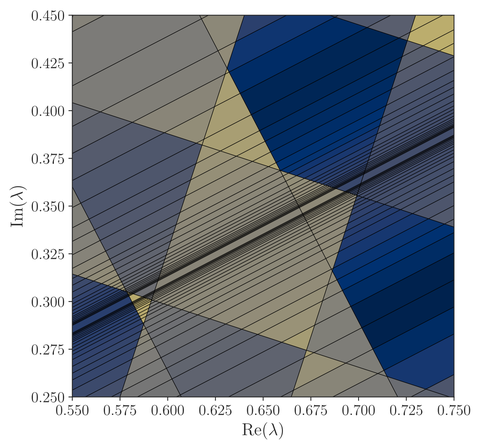
1. Invariance by multiplication by 2 and by i



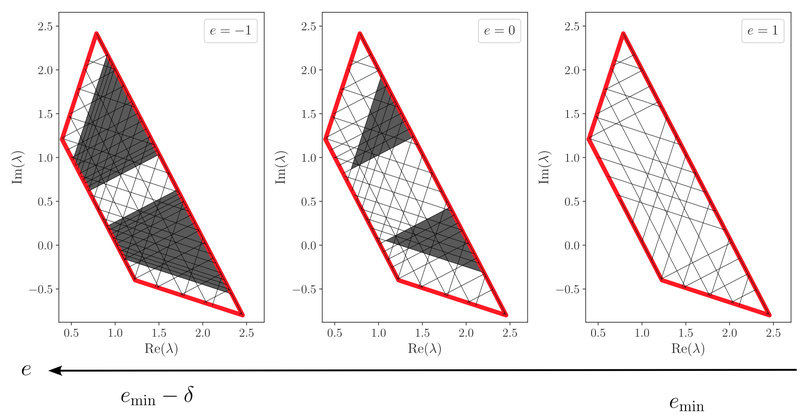
2. Piecewise constant with polygonal pieces



3. But with an *accumulation phenomenon* that generates an *infinite number of pieces*



Manage the accumulation phenomenon



Introduction of a parameter δ that controls the number of polygonal pieces and their distance to the accumulation lines

Construction of *CROQuant*

Algorithm 1 CROQuant: Complex Rank-One Quantization Algorithm

Input: $\boldsymbol{x} \in \mathbb{R}^m$, $\boldsymbol{y} \in \mathbb{R}^n$, $t \geq 1$, $\delta \in \mathbb{N}$

Initialize $\hat{\lambda} \leftarrow 1$
Build the tiling domain Ω
Build $\mathcal{R}_\delta$, the set of polygonal pieces inside Ω associated with δ
Build Λ_δ, the set of centroids from the polygonal pieces $\mathcal{R}_\delta$
for $\lambda \in \Lambda_\delta$ **do**
    if $f(\lambda) < f(\hat{\lambda})$ **then** $\hat{\lambda} \leftarrow \lambda$
return $\mathrm{round}(\hat{\lambda}\boldsymbol{x})$, $\mathrm{round}(\mu(\hat{\lambda})\boldsymbol{y})$

Application to butterfly Factorizations

$$\boldsymbol{Z} = \boldsymbol{B}_1 \cdot \boldsymbol{B}_2 \cdot \ldots \cdot \boldsymbol{B}_L$$

Butterfly Quantization problem

$$\min_{\hat{\boldsymbol{B}}_1,\ldots,\hat{\boldsymbol{B}}_L\in\mathbb{CF}_t^{n\times n}} \left\|\boldsymbol{B}_1\cdots\boldsymbol{B}_L - \hat{\boldsymbol{B}}_1\cdots\hat{\boldsymbol{B}}_L\right\|_F^2$$

Key property

$$\boldsymbol{B}_1\cdots\boldsymbol{B}_j\underbrace{\boldsymbol{B}_{j+1}\cdots\boldsymbol{B}_k}_{\boldsymbol{X}}\underbrace{\boldsymbol{B}_{k+1}\cdots\boldsymbol{B}_l}_{\boldsymbol{Y}^H}\boldsymbol{B}_{l+1}\cdots\boldsymbol{B}_L$$

$\boldsymbol{X}\boldsymbol{Y}^H = \sum_{i=1}^{n} \boldsymbol{x}_i\boldsymbol{y}_i^H$ where $\boldsymbol{x}_i\boldsymbol{y}_i^H$ have *disjoint supports* [2]

→ use n times CROQuant

For $L > 2$: need *heuristics* to decide how to **order/group** the factors

Left-To-Right (LTR): writing $\boldsymbol{B}_1(\boldsymbol{B}_2(\cdots(\boldsymbol{B}_{L-1}\boldsymbol{B}_L)\cdots))$
Pairwise : writing $(\boldsymbol{B}_1\boldsymbol{B}_2)(\boldsymbol{B}_3\boldsymbol{B}_4)\cdots(\boldsymbol{B}_{L-1}\boldsymbol{B}_L)$

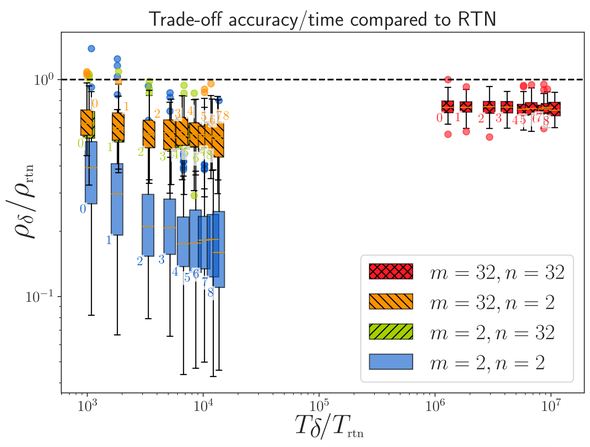
Experiments

On random rank-one matrices

$\mathrm{Re}(\boldsymbol{x}), \mathrm{Im}(\boldsymbol{x}) \sim \mathcal{U}_{[0,1]}$
$\mathrm{Re}(\boldsymbol{y}), \mathrm{Im}(\boldsymbol{y}) \sim \mathcal{U}_{[0,1]}$

$$\rho := \frac{\left\|\boldsymbol{x}\boldsymbol{y} - \hat{\boldsymbol{x}}\hat{\boldsymbol{y}}^H\right\|_F^2}{\left\|\boldsymbol{x}\boldsymbol{y}^H\right\|_F^2}$$

Trade-off accuracy/time compared to RTN

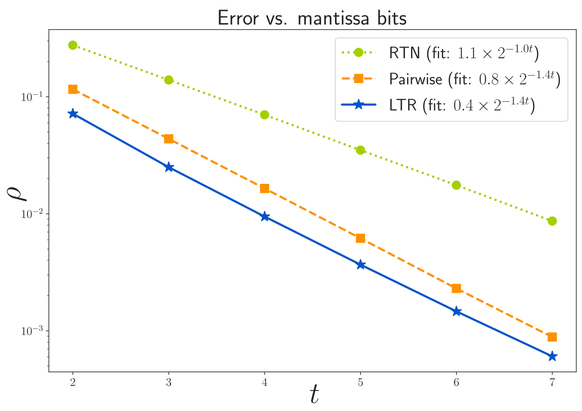


→ 1. CROQuant **improves accuracy** compared to RTN when $\delta \geq 2$
2. As m and n increase, increasing δ is less worthwhile

On random butterfly matrices

$\boldsymbol{B}_1\cdots\boldsymbol{B}_L \sim \mathcal{CN}(\boldsymbol{0}, \boldsymbol{I}_n)$ $\rho := \frac{\left\|\boldsymbol{B}_1\cdots\boldsymbol{B}_L - \hat{\boldsymbol{B}}_1\cdots\hat{\boldsymbol{B}}_L\right\|_F^2}{\left\|\boldsymbol{B}_1\cdots\boldsymbol{B}_L\right\|_F^2}$

Error vs. mantissa bits



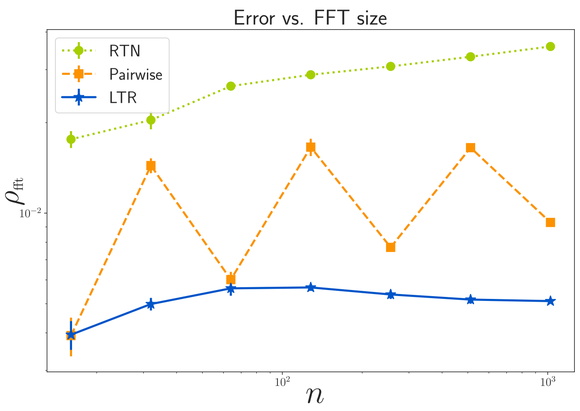
→ Reduction of **30% bits** compared to RTN

On the Fast Fourier transform

$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$
$\boldsymbol{y} = \boldsymbol{F}\boldsymbol{x}$

$$\rho_{\mathrm{fft}} := \frac{\|y - \hat{y}\|_2^2}{\|y\|_2^2}$$

Error vs. FFT size



→ LTR and Pairwise **outperform** RTN on the FFT

[1] Gribonval, R. & Mary, T. & Riccietti E. (2023). *Optimal Quantization of Rank-One Matrices in Floating-Point Arithmetic - with Applications to Butterfly Factorizations.* preprint HAL
[2] Le, Q.-T. & Riccietti E. & Gribonval, R. (2023). *Fast learning of fast transforms, with guarantees.* ICASSP 2022