Sketchy Decisions

Joel A. Tropp
Steele Family Professor of
Applied & Computational Mathematics

Computing + Mathematical Sciences
California Institute of Technology
jtropp@cms.caltech.edu

Collaborators: Volkan Cevher (EPFL), Roarke Horstmeyer (Duke),
Olivier Fercoq (Télécom Paris), Madeleine Udell (Cornell), **Alp Yurtsever** (MIT)

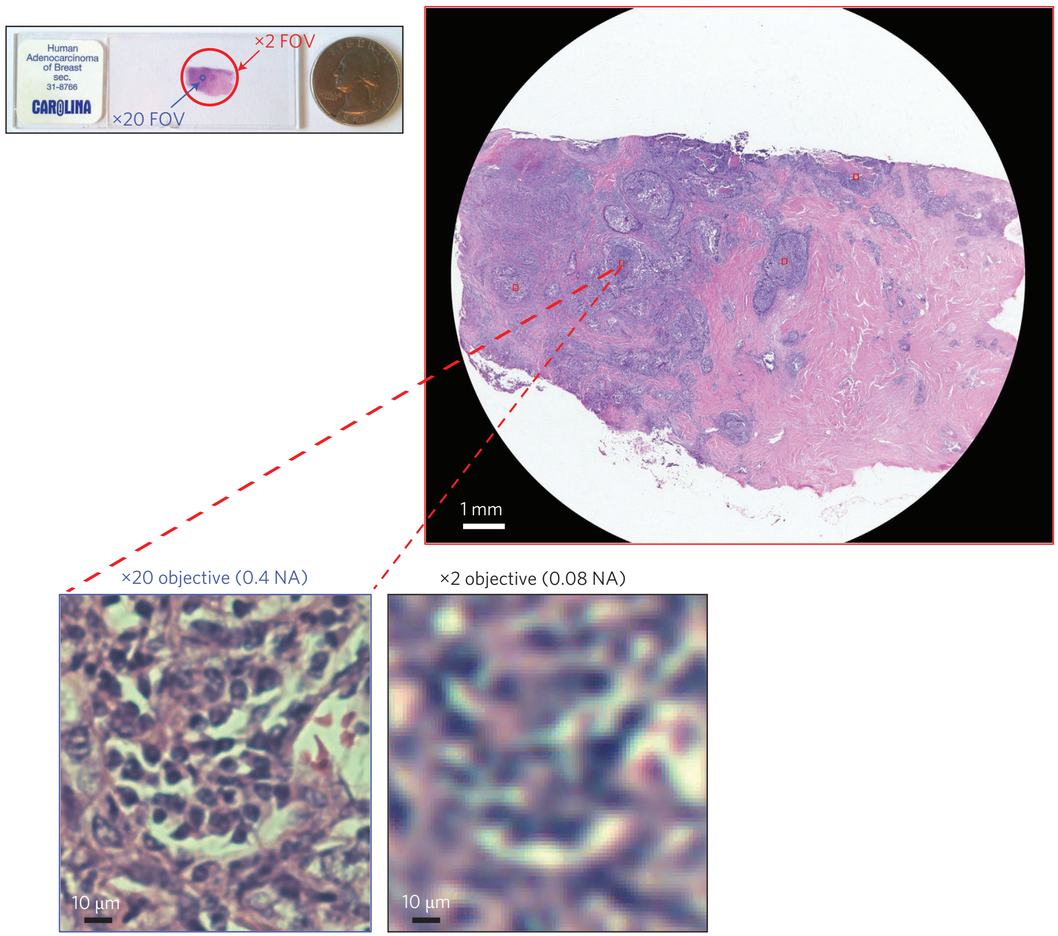
Research supported by ONR, AFOSR, NSF, DARPA, ERC, SNF, Sloan, and Moore.

Outline

- 1:00–1:30 pm. Fourier ptychography and scalable SDP algorithms
- 1:35–2:20 pm. Nonlinear SDPs via SketchyCGM
- 2:30–3:15 pm. Standard-form SDPs via SketchyCGAL

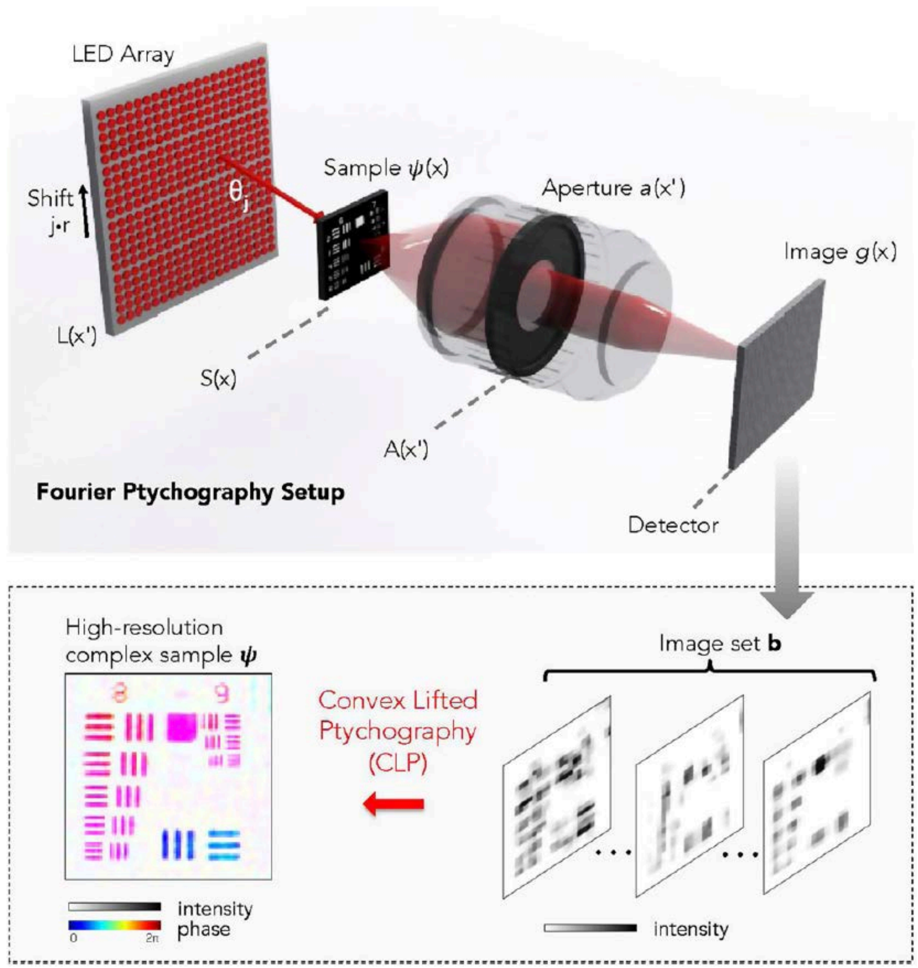
Fourier
Ptychography

Microscopy: Field of View / Resolution



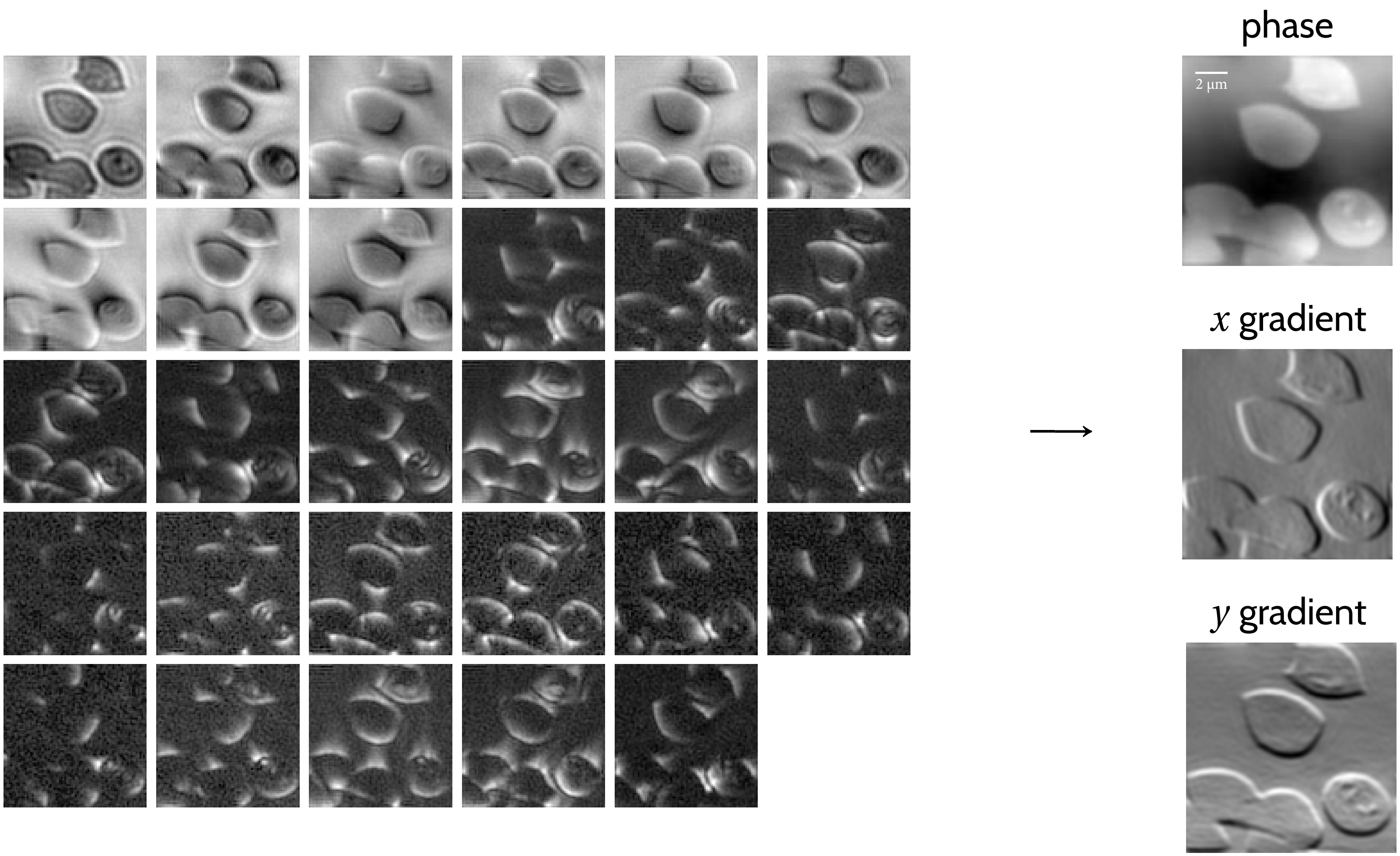
Source: Adapted from Zhang et al. 2013.

Fourier Ptychography: Field of View + Resolution



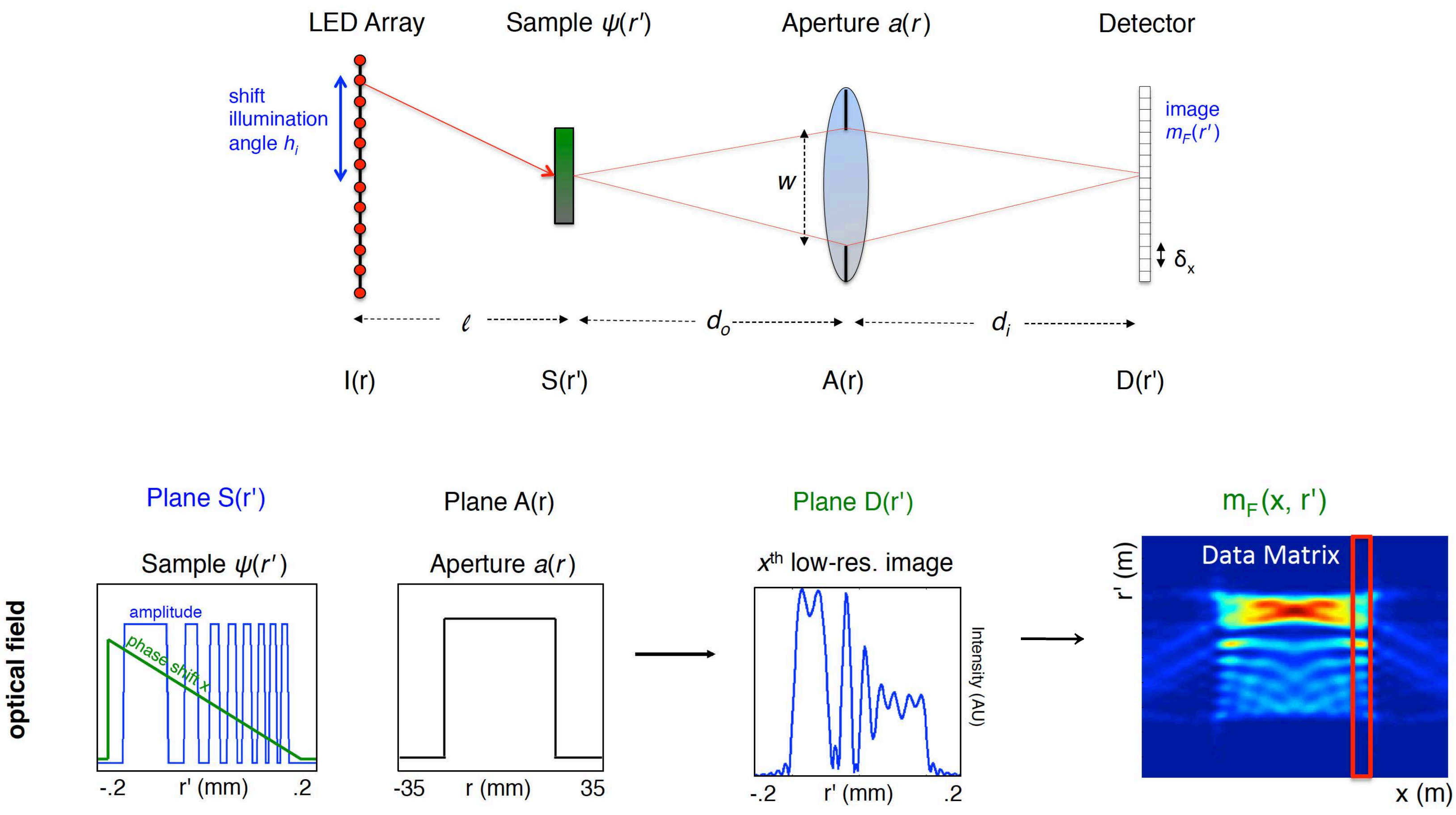
Sources: Zhang et al. 2013; Horstmeyer & Yang 2014; Ou et al. 2014; Horstmeyer et al. 2015.

Fourier Ptychography: Malaria Example



Source: Yurtsever et al. 2017.

Fourier Ptychography: Schematic



Source: Adapted from Horstmeyer & Yang 2014.

Fourier Ptychography: Reconstruction

- Acquire a family of noisy measurements:

$$b_i = |\langle \boldsymbol{a}_i, \boldsymbol{\psi} \rangle|^2 + \xi_i \quad \text{for } i = 1, \dots, d$$

  - $\boldsymbol{a}_i \in \mathbb{C}^n$ are known measurement vectors that model FP system
  - $\boldsymbol{\psi} \in \mathbb{C}^n$ is the unknown sample transmission function
  - $\xi_i \in \mathbb{R}$ is unknown noise

- Reconstruction via unconstrained optimization:

$$\underset{\boldsymbol{x} \in \mathbb{C}^n}{\text{minimize}} \quad \sum_{i=1}^{d} \operatorname{loss}\big(|\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle|^2;\ b_i\big)$$

- Assume $\operatorname{loss}(\cdot;\ b)$ is a convex function
- **Malaria example:** $n = 25\,600$ and $d = 185\,600$

Sources: Zhang et al. 2013; Horstmeyer & Yang 2014; Horstmeyer et al. 2015.

Fourier Ptychography: Convex Reconstruction

- **Observe:** $|\langle \boldsymbol{a}, \boldsymbol{x} \rangle|^2 = \boldsymbol{a}^* (\boldsymbol{x}\boldsymbol{x}^*) \boldsymbol{a} = \boldsymbol{a}^* \boldsymbol{X} \boldsymbol{a}$ where $\boldsymbol{X}$ is rank-one, psd
- Lift to matrix optimization problem:

$$\underset{\boldsymbol{X} \in \mathbb{H}_n}{\text{minimize}} \quad \sum_{i=1}^{d} \operatorname{loss}\big(\boldsymbol{a}_i^* \boldsymbol{X} \boldsymbol{a}_i;\ b_i\big) \quad \text{subject to} \quad \operatorname{rank}(\boldsymbol{X}) = 1; \quad \boldsymbol{X} \text{ psd}$$

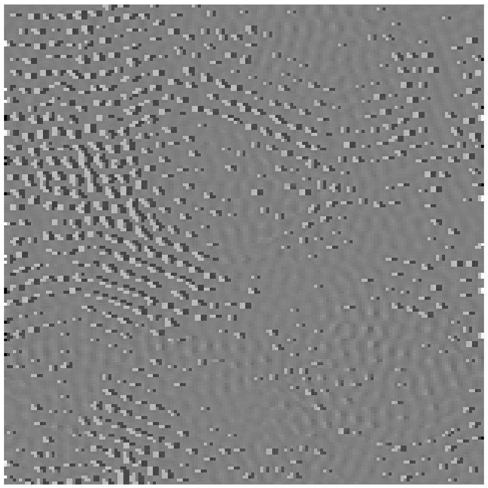
- Replace rank constraint with trace constraint to obtain convex problem:

$$\underset{\boldsymbol{X} \in \mathbb{H}_n}{\text{minimize}} \quad \sum_{i=1}^{d} \operatorname{loss}\big(\boldsymbol{a}_i^* \boldsymbol{X} \boldsymbol{a}_i;\ b_i\big) \quad \text{subject to} \quad \operatorname{trace}(\boldsymbol{X}) = \alpha; \quad \boldsymbol{X} \text{ psd}$$

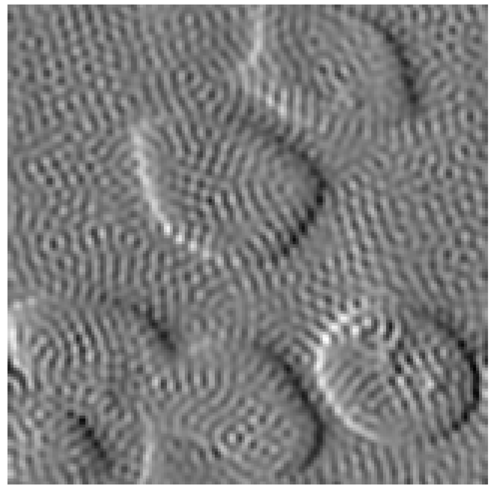
- Return maximum eigenvector $\boldsymbol{x}_\star$ of a solution $\boldsymbol{X}_\star$
- **Malaria example:** Matrix $\boldsymbol{X}$ has $n^2 = 6.55 \cdot 10^8$ real dof

Sources: AIM Frames Workshop 2008; Edidin et al. 2009; Chai et al. 2011; Candès et al. 2013; Horstmeyer et al. 2015.

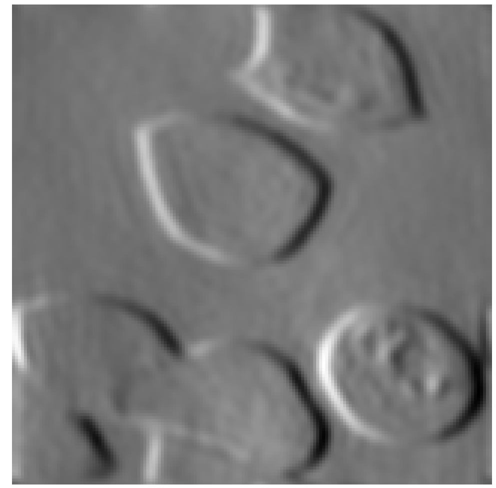
Convexity: Why Bother?



Wirtinger Flow
(not convex)



Burer–Monteiro
(sort of convex)
$X = YY^*$



???
(convex)

images of x phase gradient

Challenge: How to solve the convex ptychography problem at scale?

Sources: Burer & Monteiro 2003; Candès et al. 2014; Horstmeyer et al. 2015; Yurtsever et al. 2017.

Nonlinear SDP with Optimal Storage

Convex Low-Rank Matrix Optimization

$$\text{minimize} \quad f(\mathcal{A}\boldsymbol{X}) \quad \text{subject to} \quad \boldsymbol{X} \in \alpha\boldsymbol{\Delta}_n \qquad \text{(SDP-nl)}$$

- $\mathcal{A} : \mathbb{H}_n \to \mathbb{R}^d$ is a real-linear map
- $f : \mathbb{R}^d \to \mathbb{R}$ is convex and continuously differentiable
- $\boldsymbol{\Delta}_n := \{\boldsymbol{X} \in \mathbb{H}_n^+ : \operatorname{trace} \boldsymbol{X} = 1\}$ = density matrices

- In many applications,
  - $\mathcal{A}$ extracts d linear measurements of $n \times n$ matrix
  - $f = \operatorname{loss}(\,\cdot\,; \boldsymbol{b})$ for data $\boldsymbol{b} \in \mathbb{R}^d$
  - $d \ll n^2$
  - α modulates rank of solution

- Models problems in signal processing, statistics, and machine learning (e.g., convex ptychography)

$\mathbb{H}_n$ = $n \times n$ Hermitian matrices; $\mathbb{H}_n^+$ = $n \times n$ psd matrices

Approximate Solutions

- Let $X_\star$ be an optimal point of (SDP-nl)
- Algorithms produce a feasible point X that is *ε-suboptimal*:

$$f(\mathcal{A}X) - f(\mathcal{A}X_\star) \le \varepsilon$$

- **Smoothness:** Distance to optimal point controls suboptimality:

$$f(\mathcal{A}X) - f(\mathcal{A}X_\star) \le L\,\|X - X_\star\|_*$$

- **Stability:** Suboptimality controls distance to (unique) optimum:

$$f(\mathcal{A}X) - f(\mathcal{A}X_\star) \ge \kappa\,\|X - X_\star\|_*$$

- **Smoothness + Stability:** Suboptimality comparable with distance to optimum
- **Assume** smoothness + stability to simplify guarantees

$\|\cdot\|_*$ = Schatten 1-norm = dual of ℓ_2 operator norm = trace norm

Low-Rank Approximation of a Solution

- **NP-hard** to solve (SDP-nl) + $\operatorname{rank} \boldsymbol{X} \le r$
- **Legerdemain:** Find a rank-r approximation $\hat{\boldsymbol{X}}$ of a solution to (SDP-nl):

$$\|\hat{\boldsymbol{X}} - \boldsymbol{X}_\star\|_* \le \text{const} \cdot \|\boldsymbol{X}_\star - [\![\boldsymbol{X}_\star]\!]_r\|_*$$

- In particular, if $\operatorname{rank}(\boldsymbol{X}_\star) \le r$, then $\hat{\boldsymbol{X}} = \boldsymbol{X}_\star$

- **Goal:** Compute a rank-r approximation $\hat{\boldsymbol{X}}$ to an ε-suboptimal point $\boldsymbol{X}_\varepsilon$:

$$\|\hat{\boldsymbol{X}} - \boldsymbol{X}_\varepsilon\|_* \le \text{const} \cdot \|\boldsymbol{X}_\varepsilon - [\![\boldsymbol{X}_\varepsilon]\!]_r\|_*$$

- **Assume** smoothness + stability + $\operatorname{rank}(\boldsymbol{X}_\star) \le r$
- **Conclude**

$$\|\hat{\boldsymbol{X}} - \boldsymbol{X}_\star\|_* \quad \asymp \quad f(\mathcal{A}\boldsymbol{X}) - f(\mathcal{A}\boldsymbol{X}_\star) \quad \asymp \quad \varepsilon$$

$[\![\cdot]\!]_r$ = a best rank-r approximation with respect to $\|\cdot\|_*$

Optimal Storage

What kind of storage bounds can we hope for?

- **Assume** black-box implementation of operations with linear map:

$$\boldsymbol{u} \mapsto \mathcal{A}(\boldsymbol{u}\boldsymbol{u}^*) \qquad\qquad (\boldsymbol{u}, \boldsymbol{z}) \mapsto (\mathcal{A}^*\boldsymbol{z})\boldsymbol{u}$$

$$\mathbb{C}^n \to \mathbb{R}^d \qquad\qquad \mathbb{C}^n \times \mathbb{R}^d \to \mathbb{C}^n$$

- Need $\Theta(n + d)$ storage for output of black-box operations
- Need $\Theta(rn)$ storage for rank-r approximation to a solution

> **Definition.** An algorithm for the nonlinear SDP (SDP-nl) has optimal storage if its working storage is $\Theta(d + rn)$ rather than $\Theta(n^2)$.

Source: Yurtsever et al. 2017; Cevher et al. 2017.

So Many Algorithms...

- 1990s: **Interior-point methods**
  - Storage cost $\Theta(n^4)$ for Hessian
- 2000s: **Convex first-order methods**
  - (Accelerated) proximal gradient, spectral bundle methods, and others
  - Store matrix variable $\Theta(n^2)$; projection onto constraint set via SVD
- 2008–Present: **Storage-efficient convex first-order methods**
  - Conditional gradient method (CGM), entropic mirror descent (EMD), and extensions
  - Store matrix with rank $O(tn)$; no storage guarantees
- 2009–Present: **Nonconvex heuristics**
  - Burer–Monteiro factorization idea + various nonlinear programming methods
  - Store low-rank matrix factors $\Theta(rn)$
  - For Burer–Monteiro, necessary that rank $r = \Omega(\sqrt{d})$ + extra assumptions
  - Other nonconvex methods frame unrealistic + unverifiable statistical assumptions

Sources: Interior-point: Nemirovski & Nesterov 1994; ... **First-order:** Rockafellar 1976; Helmberg & Rendl 1997; Auslender & Teboulle 2006; ... **CGM:** Frank & Wolfe 1956; Levitin & Poljak 1967; Jaggi 2013; Baes et al. 2013; ... **Heuristics:** Homer & Peinado 1997; Burer & Monteiro 2003; Keshavan et al. 2009; Jain et al. 2012; Candès et al. 2014; Bhojanapalli et al. 2015; Boumal et al. 2016; Cifuentes & Moitra 2019; ....

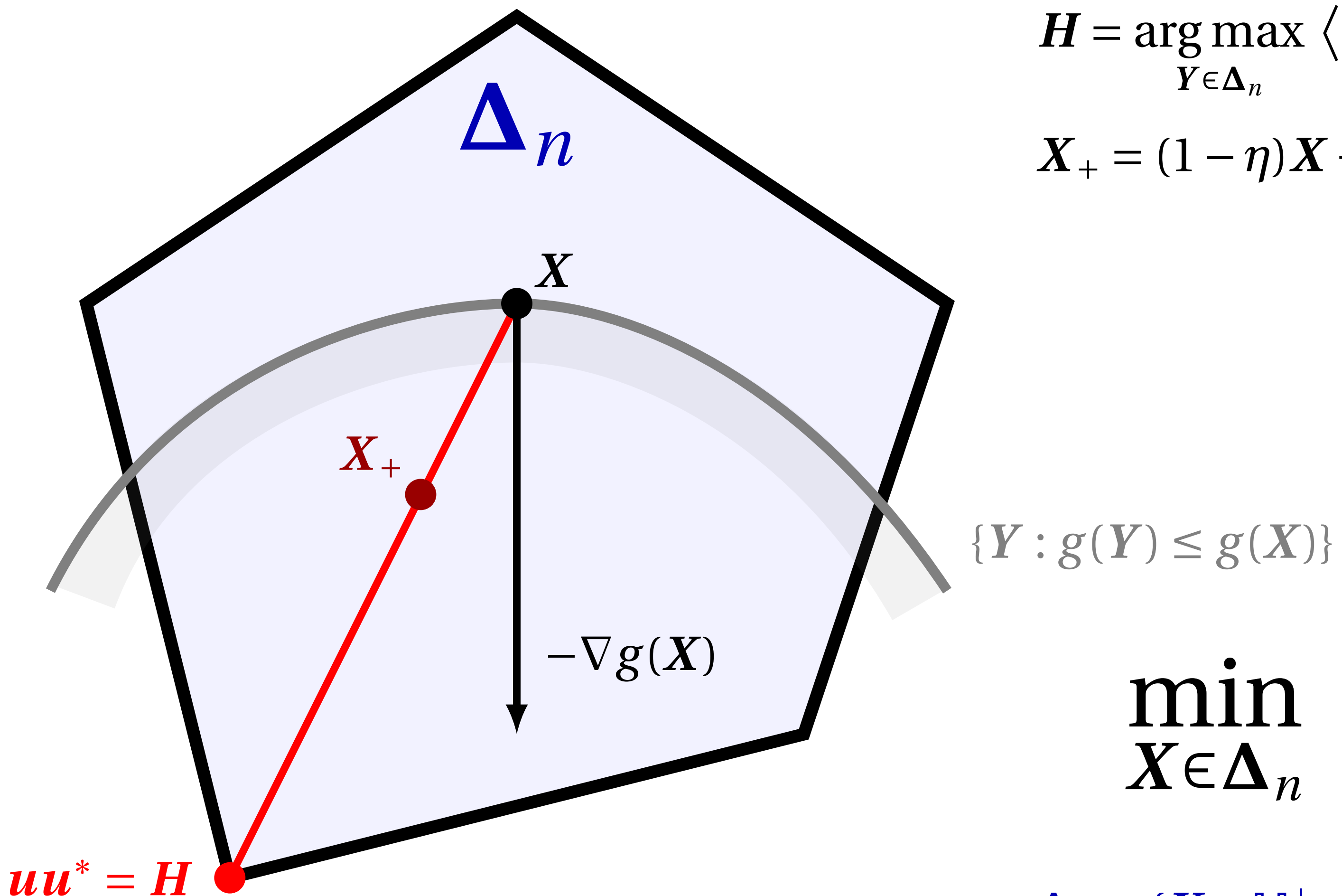
The Challenge

- Some algorithms provably solve the model problem...
- Some algorithms have optimal storage guarantees...

Is there a practical algorithm
that provably computes
a low-rank approximation
to a solution of the nonlinear SDP
+ has optimal storage guarantees?

SketchyCGM

Geometry of Conditional Gradient



$$\boldsymbol{H} = \arg\max_{\boldsymbol{Y} \in \Delta_n} \langle \boldsymbol{Y}, -\nabla g(\boldsymbol{X}) \rangle$$

$$\boldsymbol{X}_+ = (1-\eta)\boldsymbol{X} + \eta \boldsymbol{H}$$

$$\min_{\boldsymbol{X} \in \Delta_n} g(\boldsymbol{X})$$

$$\Delta_n = \{\boldsymbol{X} \in \mathbb{H}_n^+ : \operatorname{trace} \boldsymbol{X} = 1\}$$

CGM for the Nonlinear SDP

Input: Problem data
Output: Approximate solution $\boldsymbol{X}_{\mathrm{cgm}} \in \alpha \boldsymbol{\Delta}_n$

function CGM
  $\boldsymbol{X} \leftarrow \mathbf{0}_{n \times n}$ ▷ Initialize matrix variable
  for $t \leftarrow 0, 1, 2, 3, \dots$ **do**
    $\eta \leftarrow 2/(t+2)$ ▷ Step size schedule
    $\boldsymbol{u} \leftarrow \texttt{MinEvec}(\mathcal{A}^*(\nabla f(\mathcal{A}\boldsymbol{X})))$ ▷ Eigenvector computation
    $\boldsymbol{H} \leftarrow -\alpha \boldsymbol{u}\boldsymbol{u}^*$ ▷ Form update direction
    $\boldsymbol{X} \leftarrow (1-\eta)\boldsymbol{X} + \eta \boldsymbol{H}$ ▷ Linear update to matrix variable
  return $\boldsymbol{X}$

Comment: In notation of last slide, $g = f \circ \mathcal{A}$. The gradient $\nabla g = \mathcal{A}^* \circ \nabla f \circ \mathcal{A}$.

Sources: Frank & Wolfe 1956; Levitin & Poljak 1967; Jones 1992; DeVore & Temlyakov 1996; Hazan 2008; Clarkson 2010; Jaggi 2013.

Convergence of CGM

Fact 1 (CGM: Convergence Rate). *Let $\boldsymbol{X}_\star$ be an arbitrary solution to the nonlinear SDP* (SDP-nl)*. For each iteration $t \geq 0$, the matrix $\boldsymbol{X}_t$ constructed by CGM satisfies*

$$f(\mathcal{A}\boldsymbol{X}_t) - f(\mathcal{A}\boldsymbol{X}_\star) \leq \frac{2C}{2+t}.$$

The constant C reflects the curvature of the objective and size of the domain.

- CGM behavior depends on curvature of objective
- Objective values converge at rate $O(1/t)$!
- **Extension:** Computable stopping criterion (omitted)
- **Extension:** Works with very approximate eigenvalue calculations

Source: Frank & Wolfe 1956; Levitin & Poljak 1967; Hazan 2008; Clarkson 2010; Jaggi 2013; ....

Randomized Lanczos

- Lanczos efficiently minimizes the Rayleigh quotient of $\boldsymbol{M} \in \mathbb{H}_n$ over

$$\operatorname{span}\{\boldsymbol{\omega},\ \boldsymbol{M}\boldsymbol{\omega},\ \boldsymbol{M}^2\boldsymbol{\omega},\ \dots,\ \boldsymbol{M}^q\boldsymbol{\omega}\}$$

- Uses q matrix–vector products with $\boldsymbol{M}$
- Can be implemented with storage $\Theta(n)$!
- **Randomization:** Draw test vector $\boldsymbol{\omega} \sim \textsc{normal}(\mathbf{0}, \mathbf{I}_n)$

Fact 2 (Randomized Lanczos). *Fix* $\boldsymbol{M} \in \mathbb{H}_n$. *For* $\varepsilon \in (0,1]$ *and* $\delta \in (0, 0.5]$, *randomized Lanczos returns a unit vector* $\boldsymbol{u} \in \mathbb{C}^n$ *with*

$$\boldsymbol{u}^* \boldsymbol{M} \boldsymbol{u} \leq \lambda_{\min}(\boldsymbol{M}) + \tfrac{1}{8}\varepsilon \, \|\boldsymbol{M}\| \quad \text{with probability} \geq 1 - 2\delta$$

whenever $q \geq \frac{1}{2} + \varepsilon^{-1/2} \log(n/\delta^2)$.

- **Outcome:** Implement CGM via RandLanczos with $q_t = O(t^{1/2} \log n)$

Sources: Kuczyński & Woźniakowski 1992; Arora et al. 2005; Tropp 2017–2021; Jaggi 2013; Yurtsever et al. 2017–2021; ....

Crisis / Opportunity

Crisis:

- CGM needs many iterations to converge to a near-low-rank solution
- The numerical rank of the CGM iterates can increase without bound
- CGM requires high + unpredictable storage

Opportunity:

- Modify CGM to work with optimal storage!
- Drive the CGM iteration with small "state" variable $\boldsymbol{z} = \mathcal{A}\boldsymbol{X}$
- Use primitives to access linear map $\mathcal{A}$
- Maintain small randomized sketch of primal matrix variable $\boldsymbol{X}$
- After iteration terminates, reconstruct matrix variable $\boldsymbol{X}$ from sketch

Source: Yurtsever et al. 2017–2021.

CGM for the Nonlinear SDP

Input: Problem data

Output: Approximate solution $\boldsymbol{X}_{\mathrm{cgm}} \in \alpha \boldsymbol{\Delta}_n$

function CGM
- $\boldsymbol{X} \leftarrow \mathbf{0}_{n \times n}$ ▷ Initialize matrix variable
- **for** $t \leftarrow 0, 1, 2, 3, \dots$ **do**
  - $\eta \leftarrow 2/(t+2)$ ▷ Step size schedule
  - $\boldsymbol{u} \leftarrow \texttt{MinEvec}(\mathcal{A}^*(\nabla f(\mathcal{A}\boldsymbol{X})))$ ▷ Eigenvector computation
  - $\boldsymbol{H} \leftarrow -\alpha \boldsymbol{u}\boldsymbol{u}^*$ ▷ Form update direction
  - $\boldsymbol{X} \leftarrow (1-\eta)\boldsymbol{X} + \eta \boldsymbol{H}$ ▷ Linear update to matrix variable
- **return** $\boldsymbol{X}$

Sources: Frank & Wolfe 1956; Levitin & Poljak 1967; Jones 1992; DeVore & Temlyakov 1996; Hazan 2008; Clarkson 2010; Jaggi 2013.

State Formulation of CGM

Input: Problem data

Output: Approximate solution state $\boldsymbol{z}_{\mathrm{cgm}} = \mathcal{A}\boldsymbol{X}_{\mathrm{cgm}} \in \mathbb{R}^d$

```
function StateCGM
    z ← 0_d                                   ▷ Initialize state variable
    for t ← 0, 1, 2, 3, ... do
        η ← 2/(t + 2)
        u ← AppMinEvec(𝒜*(∇f(z)))             ▷ RandLanczos via 𝒜* primitive
        h ← 𝒜(−α u u*)                         ▷ State update via 𝒜 primitive
        z ← (1 − η) z + η h                    ▷ Linear update to state variable
    return z
```

Benefit: Only uses storage $\Theta(n+d)$!

Problem: Where do we get $\boldsymbol{X}_{\mathrm{cgm}}$?

Sketching the Decision Variable

- **Idea:** Maintain small sketch of primal variable $\boldsymbol{X}$!
- Fix target rank r of solution, and draw Gaussian dimension reduction map

$$\boldsymbol{\Omega} \in \mathbb{C}^{n \times k} \quad \text{where } k = 2r$$

- Sketch takes the form

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\Omega} \in \mathbb{C}^{n \times k}$$

- Can perform linear update $\boldsymbol{X} \leftarrow (1-\eta)\boldsymbol{X} + \eta \boldsymbol{H}$ by operating on sketch:

$$\boldsymbol{Y} \leftarrow (1-\eta)\boldsymbol{Y} + \eta \boldsymbol{H}\boldsymbol{\Omega} \qquad \text{(Recall: } \boldsymbol{H} = \boldsymbol{u}\boldsymbol{u}^*\text{)}$$

- Can compute provably good rank-r approximation $\hat{\boldsymbol{X}}$ from sketch:

$$\hat{\boldsymbol{X}} = [\![\boldsymbol{Y}(\boldsymbol{\Omega}^*\boldsymbol{Y})^\dagger \boldsymbol{Y}^*]\!]_r \qquad \text{(truncated Nyström)}$$

- Sketch uses additional storage $\Theta(rn)$!

Sources: Nyström 1930; Williams & Seeger 2001; Drineas & Mahoney 2005; Woolfe et al. 2008; Clarkson & Woodruff 2009; Halko et al. 2009; Gittens 2011, 2013; Tropp et al. 2017–2021; ....

Guarantees for Reconstruction

Theorem 3 (Nyström Sketch)**.** *The Nyström Sketch has reconstruction guarantee*

$$\mathbb{E}\,\|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_* \leq 2\,\|\boldsymbol{X} - [\![\boldsymbol{X}]\!]_r\|_*$$

- If the sketch contains a matrix $\boldsymbol{X}$ with a good low-rank approximation, then the reconstruction $\hat{\boldsymbol{X}}$ is also a good low-rank approximation of $\boldsymbol{X}$
- Similar bounds hold with high probability
- Larger sketches reduce error $(k = \zeta r)$
- Improvements when $\boldsymbol{X}$ has spectral decay
- **Extension:** Shift $\hat{\boldsymbol{X}}$ so $\operatorname{trace} \hat{\boldsymbol{X}} = \alpha$

$\|\cdot\|_*$ = Schatten 1-norm; $[\![\cdot]\!]_r$ = best rank-r approximation

Sources: Nyström 1930; Williams & Seeger 2001; Drineas & Mahoney 2005; Woolfe et al. 2008; Clarkson & Woodruff 2009; Halko et al. 2009; Gittens 2011, 2013; Tropp et al. 2017–2021; Kueng 2018; ....

SketchyCGM for the Model Problem

Input: Problem data; target rank r

Output: Rank-r approximate solution $\hat{\boldsymbol{X}} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^* \in \alpha\boldsymbol{\Delta}_n$ in factored form

function SketchyCGM
    Sketch.Init(n, r) ▷ Initialize sketch to zero
    $\boldsymbol{z} \leftarrow \mathbf{0}_d$
    for $t \leftarrow 0, 1, 2, 3, \dots$ **do**
        $\eta \leftarrow 2/(t+2)$
        $\boldsymbol{u} \leftarrow \texttt{AppMinEvec}(\mathcal{A}^*(\nabla f(\boldsymbol{z})))$
        $\boldsymbol{h} \leftarrow \mathcal{A}(-\alpha \boldsymbol{u}\boldsymbol{u}^*)$
        $\boldsymbol{z} \leftarrow (1-\eta)\boldsymbol{z} + \eta \boldsymbol{h}$
        Sketch.CGMUpdate$(-\sqrt{\alpha}\boldsymbol{u}, \eta)$ ▷ Update sketch of $\boldsymbol{X}$
    $(\boldsymbol{V}, \boldsymbol{\Lambda}) \leftarrow$ Sketch.Reconstruct$()$ ▷ Approx. eigendecomp of $\boldsymbol{X}$
    $\boldsymbol{\Lambda} \leftarrow \boldsymbol{\Lambda} + r^{-1}(\alpha - \operatorname{trace} \boldsymbol{\Lambda})\,\mathbf{I}_r$ ▷ Shift $\hat{\boldsymbol{X}}$ to fix trace
    return $(\boldsymbol{V}, \boldsymbol{\Lambda})$

Source: Yurtsever et al. 2017.

Methods for SKETCH Object

function SKETCH.INIT(n, r) ▷ Rank-r approx of $n \times n$ psd matrix
    $k \leftarrow 2r$
    $\boldsymbol{\Omega} \leftarrow \texttt{randn}(\mathbb{C}, n, k)$
    $\boldsymbol{Y} \leftarrow \texttt{zeros}(n, k)$

function SKETCH.CGMUPDATE($\boldsymbol{u}$, η)
    $\boldsymbol{Y} \leftarrow (1-\eta)\boldsymbol{Y} + \eta \boldsymbol{u}(\boldsymbol{u}^* \boldsymbol{\Omega})$ ▷ Average $\boldsymbol{u}\boldsymbol{u}^*$ into sketch

function SKETCH.RECONSTRUCT( )
    $\boldsymbol{C} \leftarrow \texttt{chol}(\boldsymbol{\Omega}^* \boldsymbol{Y})$ ▷ Cholesky decomposition
    $\boldsymbol{Z} \leftarrow \boldsymbol{Y} / \boldsymbol{C}$ ▷ Solve least-squares problems
    $(\boldsymbol{U}, \boldsymbol{\Sigma}, \sim) \leftarrow \texttt{svds}(\boldsymbol{Z}, r)$ ▷ Compute r-truncated SVD
    return $(\boldsymbol{U},\ \boldsymbol{\Sigma}^2)$ ▷ Return eigenvalue factorization

Comment: Modifications required for numerical stability

Sources: Yurtsever et al. 2017; Tropp et al. 2017.

Less Filling / Great Taste

Theorem 4 (SketchyCGM). *SketchyCGM has the following properties (whp):*

- *SketchyCGM computes a rank-r approximation of a solution of* (SDP-nl)
- *SketchyCGM has optimal storage* $\Theta(d + rn)$
- ***Assume*** (SDP-nl) *has smoothness + stability +* $\operatorname{rank} \boldsymbol{X}_\star \le r$
- ***Then*** *SketchyCGM produces rank-r iterates $\hat{\boldsymbol{X}}_t$ that satisfy*

$$\mathbb{E}_{\boldsymbol{\Omega}} \|\hat{\boldsymbol{X}}_t - \boldsymbol{X}_\star\|_* \quad \asymp \quad \mathbb{E}_{\boldsymbol{\Omega}} f(\mathcal{A}\hat{\boldsymbol{X}}_t) - f(\mathcal{A}\boldsymbol{X}_\star) \quad = \quad O(t^{-1})$$

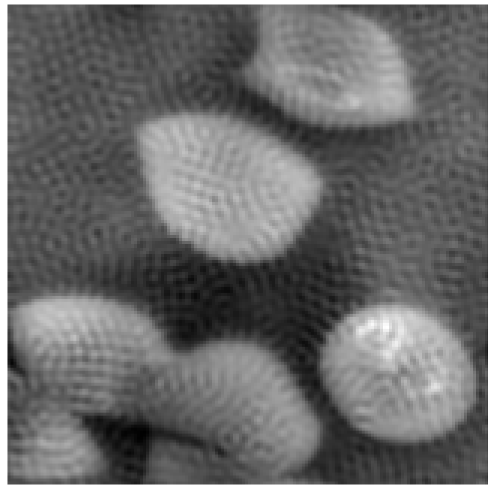
- *To achieve ε-suboptimal solution, SketchyCGM has arithmetic costs*
  1. $O(r^2 n + \varepsilon^{-1}(d + rn) + \varepsilon^{-3/2} n \log n)$ *flops*
  2. $O(\varepsilon^{-3/2} \log n)$ *applications of the $\mathcal{A}^*$ and ∇f primitives*
  3. $O(\varepsilon^{-1})$ *applications of the $\mathcal{A}$ primitive*

Source: Yurtsever et al. 2017.

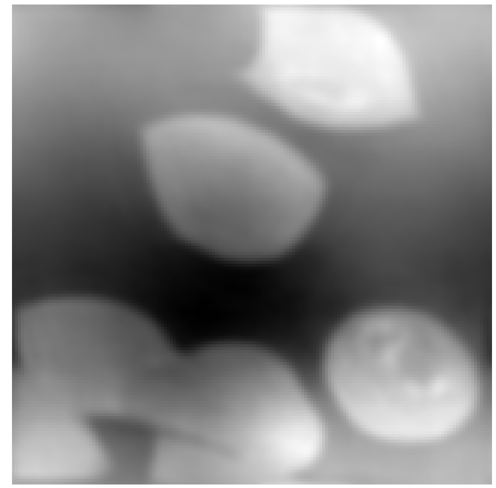
Performance of SketchyCGM

Fourier Ptychography, Redux

Wirtinger Flow



Burer–Monteiro



SketchyCGM

29 illuminations; 80^2 pixels each; $d = 1.86 \cdot 10^5$ measurements

image size $n = 160^2$ pixels; matrix variable $n^2 = 6.55 \cdot 10^8$

SketchyCGM storage (rank $r = 1$): $6.53 \cdot 10^5$

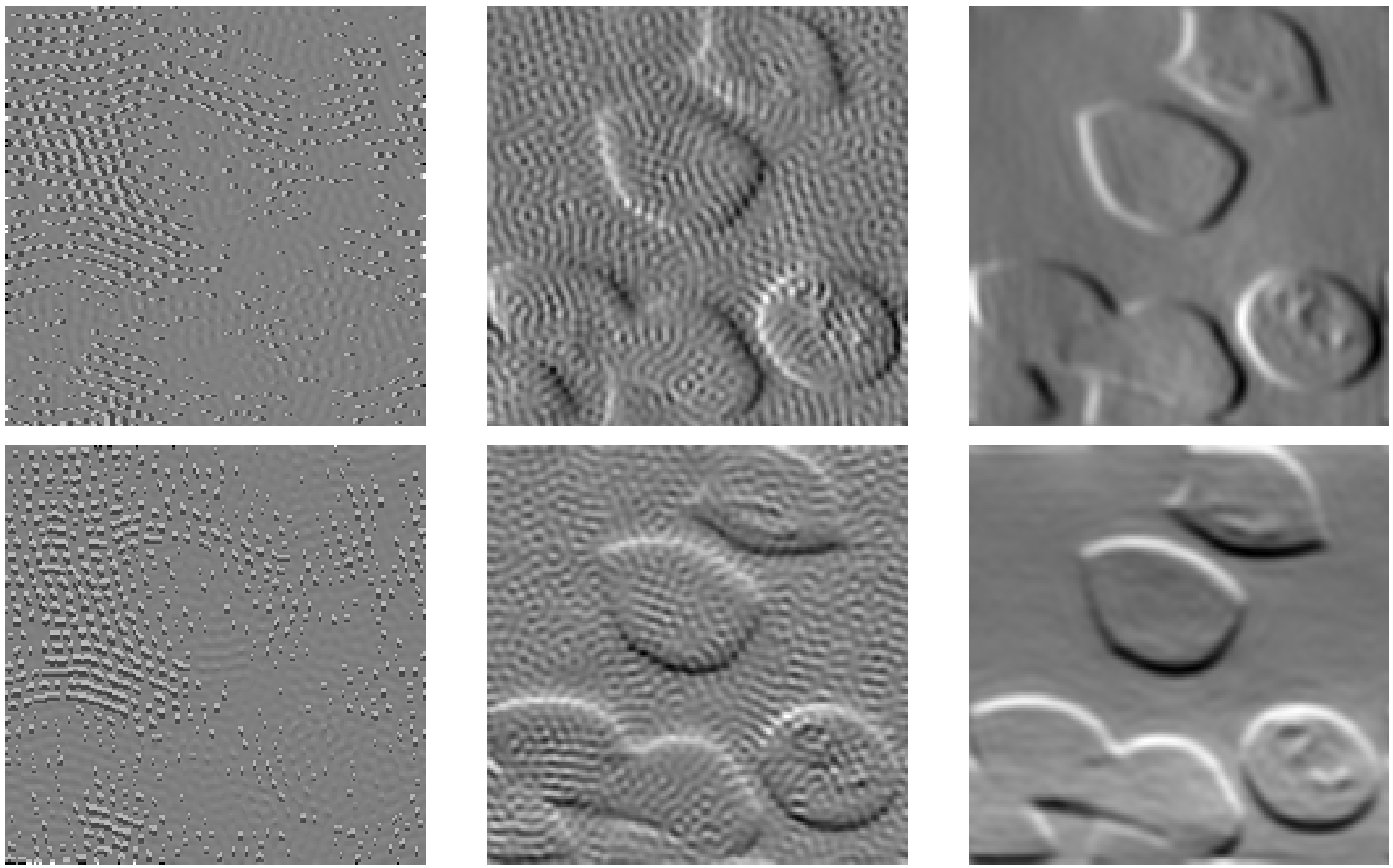
quadratic loss

Sources: Burer & Monteiro 2003; Edidin et al. 2009; Chai et al. 2011; Candès et al. 2014; Horstmeyer et al. 2015; Yurtsever et al. 2017.

Fourier Ptychography: Malaria Phase Gradients

Δ_x

Δ_y



Wirtinger Flow | Burer–Monteiro | SketchyCGM

Linear SDP:
MaxCut

The Most Unkindest Cut of All

- Let $\boldsymbol{L} \in \mathbb{H}_n$ be the (psd) Laplacian of a graph with n vertices and m edges
- Calculate the *maximum cut* via a mathematical program:

$$\text{maximize} \quad \boldsymbol{x}^{\mathrm{T}} \boldsymbol{L} \boldsymbol{x} \quad \text{subject to} \quad \boldsymbol{x} \in \{\pm 1\}^n \qquad \text{(MAXCUT)}$$

- **NP-hard**, so relax to an SDP via the map $\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}} \mapsto \boldsymbol{X}$:

$$\begin{aligned} &\text{maximize} && \operatorname{trace}(\boldsymbol{L}\boldsymbol{X}) \\ &\text{subject to} && \operatorname{diag}(\boldsymbol{X}) = \mathbf{1}, \quad \boldsymbol{X} \text{ psd} \end{aligned} \qquad \text{(MAXCUT SDP)}$$

- Report signum of maximum eigenvector of solution (or randomly round)
- Provably good idea, but...
  - Laplacian $\boldsymbol{L}$ of a graph with m edges has $\Theta(m)$ nonzeros
  - SDP decision variable $\boldsymbol{X}$ has $\Theta(n^2)$ degrees of freedom
  - Storage! Communication! Computation!

Sources: Delorme & Poljak 1993; Goemans & Williamson 1996.

SDP with Optimal Storage

Model Problem: Trace-Constrained SDP

$$\begin{aligned} \text{minimize} \quad & \operatorname{trace}(\boldsymbol{C}\boldsymbol{X}) \\ \text{subject to} \quad & \mathcal{A}\boldsymbol{X} = \boldsymbol{b} \qquad \text{(SDP)} \\ & \boldsymbol{X} \in \alpha\boldsymbol{\Delta}_n \end{aligned}$$

- $\boldsymbol{C} \in \mathbb{H}_n$ and $\boldsymbol{b} \in \mathbb{R}^d$
- $\mathcal{A} : \mathbb{H}_n \to \mathbb{R}^d$ is a real-linear map
- $\boldsymbol{\Delta}_n := \{\boldsymbol{X} \in \mathbb{H}_n^+ : \operatorname{trace} \boldsymbol{X} = 1\}$
- $\alpha > 0$ controls trace (and sometimes modulates rank)

- In many applications, $d \ll n^2$ and all solutions have low rank
- **Goal:** Produce a rank-r approximation to a solution of (SDP)

$\mathbb{H}_n$ = $n \times n$ Hermitian matrices; $\mathbb{H}_n^+$ = $n \times n$ psd matrices

Approximate Solutions

- Let $\boldsymbol{X}_\star$ be an optimal point of (SDP)
- Algorithms produce a semifeasible matrix $\boldsymbol{X} \in \alpha\boldsymbol{\Delta}_n$ that is *ε-suboptimal*:

$$\|\mathcal{A}\boldsymbol{X} - \boldsymbol{b}\| \le \varepsilon \quad \text{and} \quad \operatorname{trace}(\boldsymbol{C}\boldsymbol{X}) - \operatorname{trace}(\boldsymbol{C}\boldsymbol{X}_\star) \le \varepsilon$$

- **Stability:** Suboptimality controls distance to (unique) optimum:

$$\max\left\{\|\mathcal{A}\boldsymbol{X} - \boldsymbol{b}\|,\ \operatorname{trace}(\boldsymbol{C}\boldsymbol{X}) - \operatorname{trace}(\boldsymbol{C}\boldsymbol{X}_\star)\right\} \quad \ge \quad \kappa\,\|\boldsymbol{X} - \boldsymbol{X}_\star\|_*$$

- **Assume** stability to simplify guarantees

$\|\cdot\| = \ell_2$ norm

$\|\cdot\|_* =$ Schatten 1-norm

Low-Rank Approximation of a Solution

- **NP-hard** to solve (SDP) + $\operatorname{rank} \boldsymbol{X} \le r$
- **Legerdemain:** Find a rank-r approximation $\hat{\boldsymbol{X}}$ of a solution to (SDP):

$$\|\hat{\boldsymbol{X}} - \boldsymbol{X}_\star\|_* \le \text{const} \cdot \|\boldsymbol{X}_\star - [\![\boldsymbol{X}_\star]\!]_r\|_*$$

- In particular, if $\operatorname{rank}(\boldsymbol{X}_\star) \le r$, then $\hat{\boldsymbol{X}} = \boldsymbol{X}_\star$
- **Goal:** Compute a rank-r approximation $\hat{\boldsymbol{X}}$ to an ε-suboptimal point $\boldsymbol{X}_\varepsilon$:

$$\|\hat{\boldsymbol{X}} - \boldsymbol{X}_\varepsilon\|_* \le \text{const} \cdot \|\boldsymbol{X}_\varepsilon - [\![\boldsymbol{X}_\varepsilon]\!]_r\|_*$$

- **Assume** stability + $\operatorname{rank}(\boldsymbol{X}_\star) \le r$
- **Conclude**

$$\|\hat{\boldsymbol{X}} - \boldsymbol{X}_\star\|_* \quad \asymp \quad \max\left\{ \|\mathcal{A}\hat{\boldsymbol{X}} - \boldsymbol{b}\|,\ \operatorname{trace}(\boldsymbol{C}\hat{\boldsymbol{X}}) - \operatorname{trace}(\boldsymbol{C}\boldsymbol{X}_\star) \right\} \quad \asymp \quad \varepsilon$$

$[\![\cdot]\!]_r$ = a best rank-r approximation with respect to $\|\cdot\|_*$

Optimal Storage

What kind of storage bounds can we hope for?

- **Assume** black-box implementation of operations with objective + constraint:

$$\boldsymbol{u} \mapsto \boldsymbol{C}\boldsymbol{u} \qquad \boldsymbol{u} \mapsto \mathcal{A}(\boldsymbol{u}\boldsymbol{u}^*) \qquad (\boldsymbol{u}, \boldsymbol{z}) \mapsto (\mathcal{A}^*\boldsymbol{z})\boldsymbol{u}$$

$$\mathbb{C}^n \to \mathbb{C}^n \qquad \mathbb{C}^n \to \mathbb{R}^d \qquad \mathbb{C}^n \times \mathbb{R}^d \to \mathbb{C}^n$$

- Need $\Theta(n+d)$ storage for output of black-box operations
- Need $\Theta(rn)$ storage for rank-r approximate solution of model problem

Definition. An algorithm for the trace-constrained SDP (SDP) has optimal storage if its working storage is $\Theta(d + rn)$ rather than $\Theta(n^2)$.

Source: Yurtsever et al. 2017; Cevher et al. 2017.

The Challenge

- Some algorithms provably solve the trace-constrained SDP...
- Some algorithms have optimal storage guarantees...

Is there a practical algorithm
that provably computes
a low-rank approximation
to a solution of the trace-constrained SDP
+ has optimal storage guarantees?

SketchyHCGM

Smoothing + Homotopy

$$\begin{array}{ll} \text{minimize} & f_\beta(\boldsymbol{X}) := \operatorname{trace}(\boldsymbol{C}\boldsymbol{X}) + \frac{\beta}{2}\|\mathcal{A}\boldsymbol{X} - \boldsymbol{b}\|^2 \\ \text{subject to} & \boldsymbol{X} \in \alpha\boldsymbol{\Delta}_n \end{array} \qquad \text{(SDP-}\beta\text{)}$$

- Objective is convex and continuously differentiable
- As $\beta \to \infty$, the solutions of (SDP-β) converge to the solution set of (SDP)

- **Idea:** Solve (SDP-β) with CGM while increasing smoothing parameter β
- Gradient of objective and updates:

$$\nabla f_\beta(\boldsymbol{X}) = \boldsymbol{C} + \beta\mathcal{A}^*(\mathcal{A}\boldsymbol{X} - \boldsymbol{b})$$

$$\boldsymbol{H} = \arg\max_{\boldsymbol{Y} \in \alpha\boldsymbol{\Delta}_n} \langle \boldsymbol{Y}, -\nabla f_\beta(\boldsymbol{X}) \rangle$$

- Parallel with development of SketchyCGM from CGM

Sources: Yurtsever et al. 2018–2021.

HCGM for Trace-Constrained SDP

Input: Problem data

Output: Approximate solution matrix $\boldsymbol{X}_{\mathrm{hcgm}}$

function HCGM
- $\boldsymbol{X} \leftarrow \mathbf{0}_{n \times n}$ ▷ Initialize matrix variable
- **for** $t \leftarrow 0, 1, 2, 3, \dots$ **do**
  - $\beta \leftarrow (t+2)^{1/2}$ and $\eta \leftarrow 2/(t+2)$ ▷ Parameter schedule
  - $\boldsymbol{u} \leftarrow \mathtt{MinEvec}(\boldsymbol{C} + \beta \mathcal{A}^*(\mathcal{A}\boldsymbol{X} - \boldsymbol{b}))$
  - $\boldsymbol{H} \leftarrow -\alpha \boldsymbol{u}\boldsymbol{u}^*$ ▷ Form update direction
  - $\boldsymbol{X} \leftarrow (1-\eta)\boldsymbol{X} + \eta \boldsymbol{H}$ ▷ Linear update to matrix variable
- **return** $\boldsymbol{X}$

State: Track state $\boldsymbol{z} = \mathcal{A}\boldsymbol{X}$

Primitives: Access $\mathcal{A}$ using primitives + approximate eigenvector computation

Sources: Yurtsever et al. 2018–2021.

StateHCGM for Trace-Constrained SDP

Input: Problem data

Output: Approximate solution state $\boldsymbol{z}_{\mathrm{hcgm}}$

```
function StateHCGM
    z ← 0_d                                        ▷ Initialize state variable
    for t ← 0, 1, 2, 3, ... do
        β ← (t+2)^{1/2} and η ← 2/(t+2)
        u ← AppMinEvec(C + βA*(z − b))             ▷ RandLanczos, q = t^{1/4} log n
                                                   ▷ Via C and A* primitives
        h ← A(−α uu*)                              ▷ State update via A primitive
        z ← (1−η)z + ηh                            ▷ Linear update to state variable
    return z
```

Benefit: Only uses storage $\Theta(n+d)$!

Problem: Where do we get $\boldsymbol{X}_{\mathrm{hcgm}}$? Maintain a sketch.

Sources: Yurtsever et al. 2018–2021.

SketchyHCGM for Trace-Constrained SDP

Input: Problem data
Output: Rank-r approximate solution $\hat{\boldsymbol{X}} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^* \in \alpha\boldsymbol{\Delta}_n$ in factored form

```
function SketchyHCGM
    Sketch.Init(n, r)                                   ▷ Initialize sketch to zero
    z ← 0_d
    for t ← 0, 1, 2, 3, ... do
        β ← (t+2)^{1/2} and η ← 2/(t+2)
        u ← AppMinEvec(C + β𝒜*(z − b))
        h ← 𝒜(−α u u*)
        z ← (1 − η) z + η h
        Sketch.CGMUpdate(−√α u, η)                     ▷ Update sketch of X
    (V, Λ) ← Sketch.Reconstruct()                       ▷ Approx. eigendecomp of X
    Λ ← Λ + r^{−1}(α − trace Λ) I_r                     ▷ Shift X̂ to fix trace
    return (V, Λ)
```

Sources: Yurtsever et al. 2018–2021.

Less Filling / Great Taste

Theorem 5 (SketchyHCGM). *SketchyHCGM has the following properties (whp):*

- *SketchyHCGM computes a rank-r approximation of a solution of* (SDP)
- *SketchyHCGM has optimal storage* $\Theta(d + rn)$
- ***Assume*** (SDP) *has stability +* $\operatorname{rank} \boldsymbol{X}_\star \le r$
- ***Then*** *SketchyHCGM produces rank-r iterates $\hat{\boldsymbol{X}}_t$ that satisfy*

$$\mathbb{E}_{\boldsymbol{\Omega}} \|\hat{\boldsymbol{X}}_t - \boldsymbol{X}_\star\|_* \asymp \mathbb{E}_{\boldsymbol{\Omega}} \max\{\|\mathcal{A}\hat{\boldsymbol{X}}_t - \boldsymbol{b}\|, \operatorname{trace}(\boldsymbol{C}\hat{\boldsymbol{X}}_t) - \operatorname{trace}(\boldsymbol{C}\boldsymbol{X}_\star)\} = O(t^{-1/2})$$

- *To achieve an ε-suboptimal solution, SketchyHCGM has arithmetic costs*

1. $O(\varepsilon^{-2}(d + rn) + \varepsilon^{-5/2} n \log n)$ *flops*
2. $O(\varepsilon^{-5/2} \log n)$ *applications of primitives* $\boldsymbol{C}$ *and* $\mathcal{A}^*$
3. $O(\varepsilon^{-2})$ *applications of primitive* $\mathcal{A}$

Source: Yurtsever et al. 2018–2021.

SketchyCGAL for Trace-Constrained SDP

- **Problem:** $O(t^{-1/2})$ convergence is probably optimal, but still impractical
- **Solution:** Augmented Lagrangians!

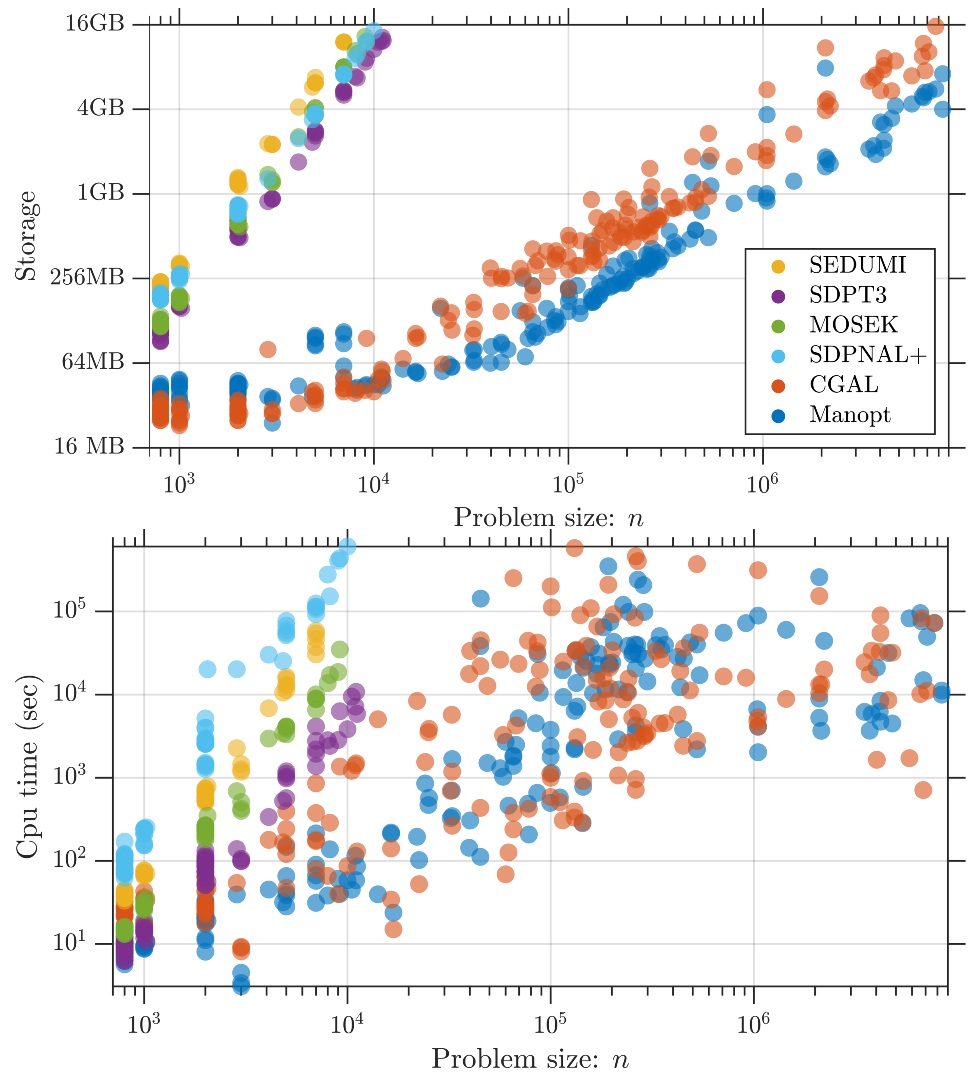
$$\text{maximize}_{\boldsymbol{y}} \quad \text{minimize}_{\boldsymbol{X}} \quad \operatorname{trace}(\boldsymbol{C}\boldsymbol{X}) + \langle \boldsymbol{y},\, \mathcal{A}\boldsymbol{X} - \boldsymbol{b} \rangle + \frac{\beta}{2} \| \mathcal{A}\boldsymbol{X} - \boldsymbol{b} \|^2$$
$$\text{subject to} \quad \boldsymbol{X} \in \alpha \boldsymbol{\Delta}_n, \quad \boldsymbol{y} \in \mathbb{R}^d$$

- CGAL: Primal update via CGM; dual update via gradient step; homotopy on β
- SketchyCGAL: CGAL + state variable + sketching
- Same theoretical guarantee as SketchyHCGM
- Empirical convergence $O(t^{-1})$
- **Extensions:** No trace constraint; affine cone constraints; other matrix sets; ...

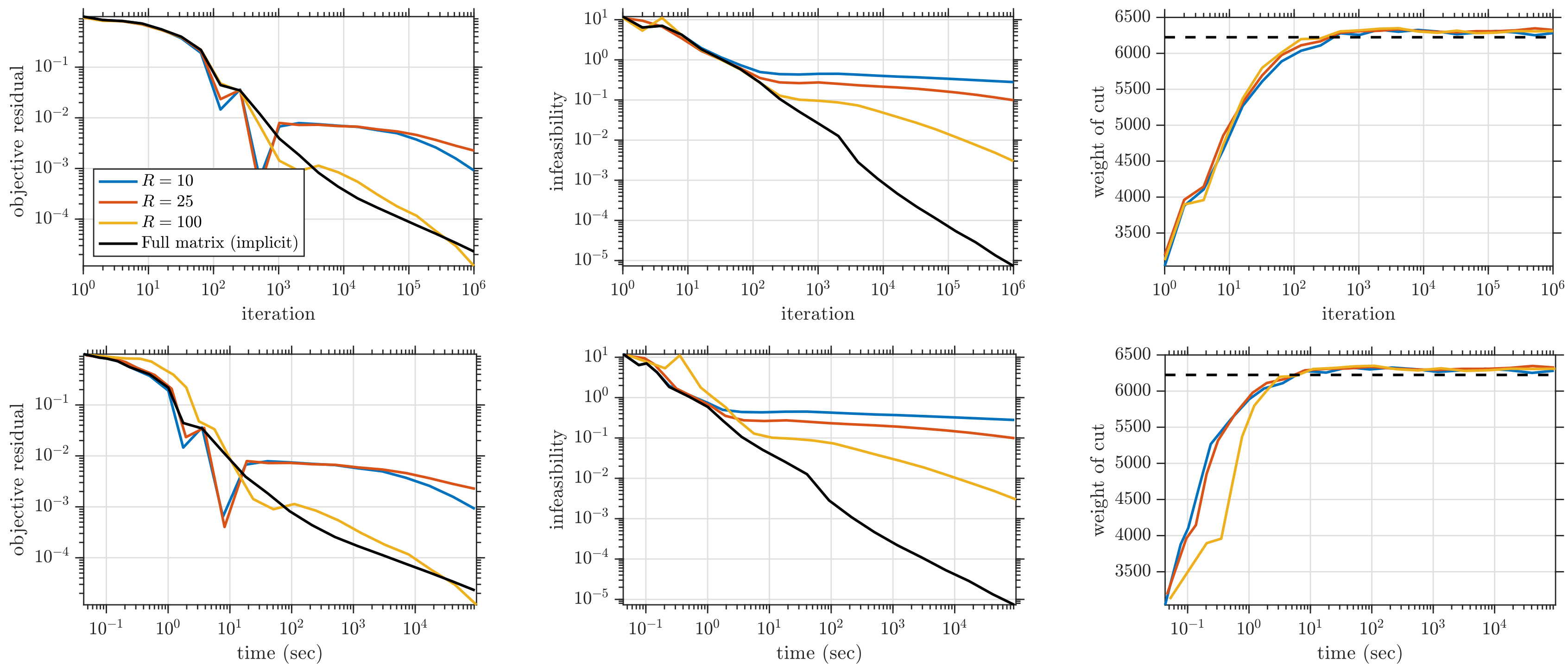
Sources: Yurtsever et al. 2018–2021.


SketchyCGAL...
disneyscreencaps.com

MaxCut for Gset / DIMACS10: Scalability ($R = 10$)



MaxCut for G67: Solution Trajectories

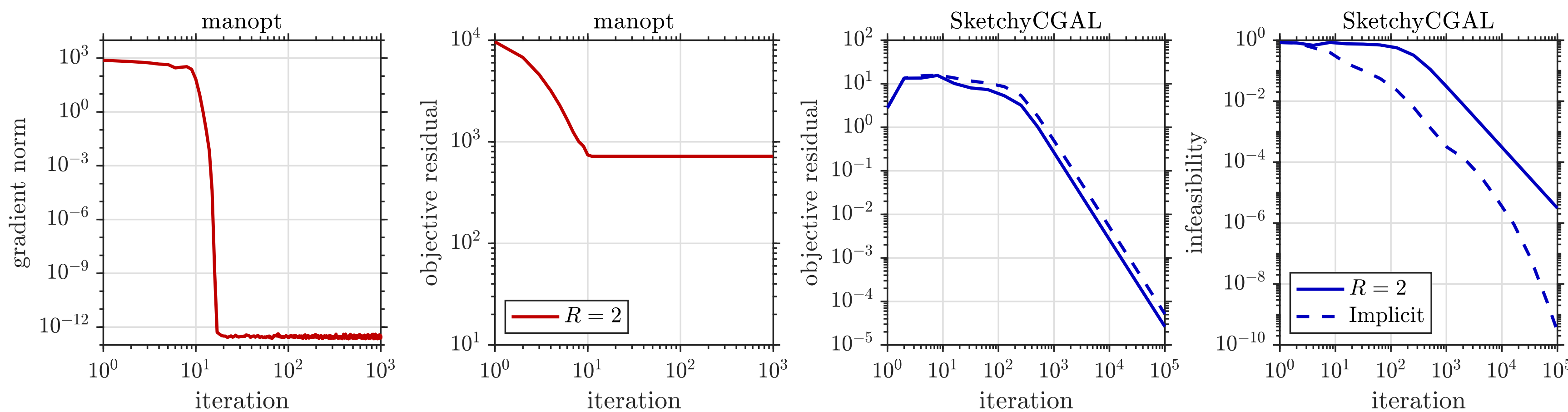


G67 graph = 10 000 vertices; 20 000 edges; dashes = cut from SDPT3

SKETCHYCGAL (rank = sketch size = R); eigenvector rounding

[l] objective residual; [c] infeasibility; [r] cut value *versus* [t] iteration; [b] time

Burer–Monteiro is not Storage-Optimal



MaxCut; dimension $n = 100$; unique solution; solution rank 1

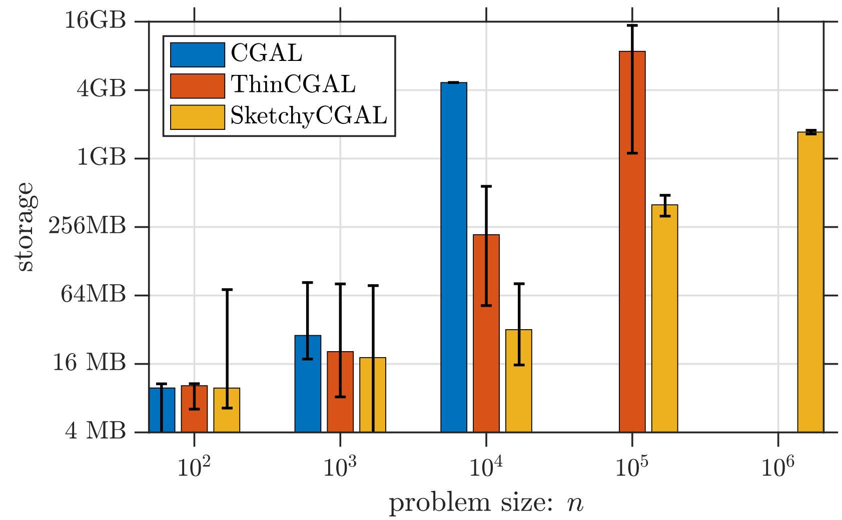
Algorithm trajectories: Typical instance

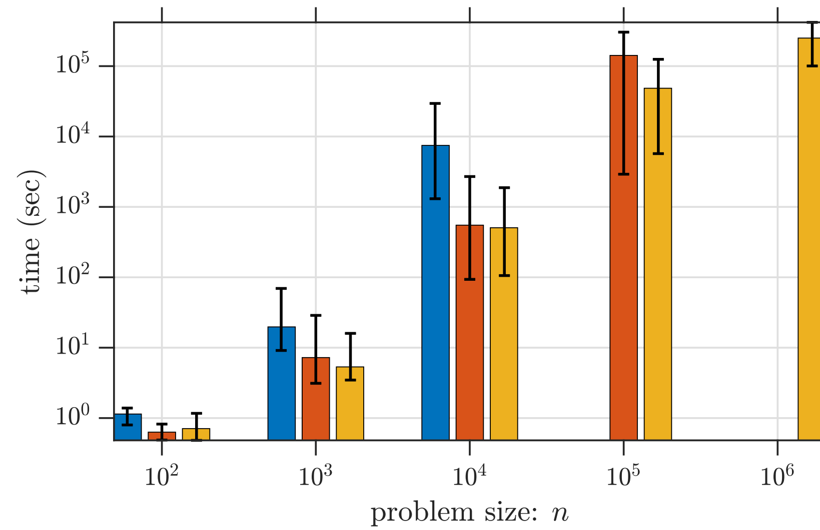
manopt ($R = 2$) fails in 77% to 90% of trials

SketchyCGAL ($R = 2$) solves all instances

Source: Boumal et al. 2014; Waldspurger & Waters 2018.

Linear Phase Retrieval: Scalability





minimize $\operatorname{trace} \boldsymbol{X}$ subject to $\mathcal{A}\boldsymbol{X} = \boldsymbol{b}$, $\operatorname{trace} \boldsymbol{X} \leq \alpha$, $\boldsymbol{X}$ psd

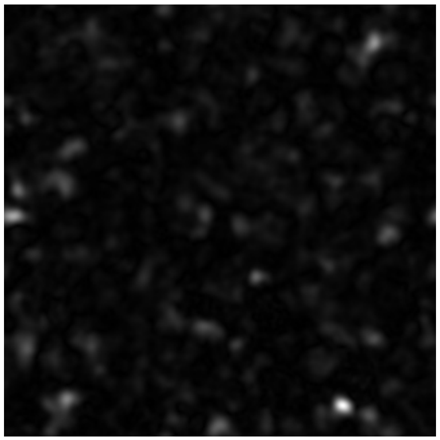
Random instances (rank = 1), measurements $d = 12n$, bound $\alpha = 3n$

CGAL = no sketching; THINCGAL = CGAL + thin SVD update

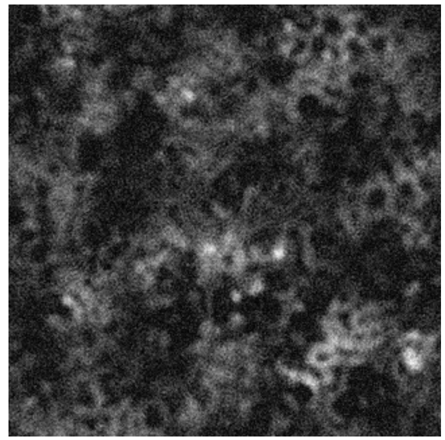
SKETCHYCGAL (sketch size $R = 5$); relative errors = 10^{-2}

Fourier Ptychography (Simulated)

$t = 10$ (69 sec)



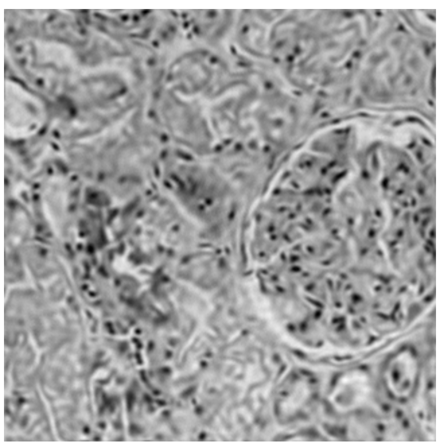
$t = 100$ (1063 sec)



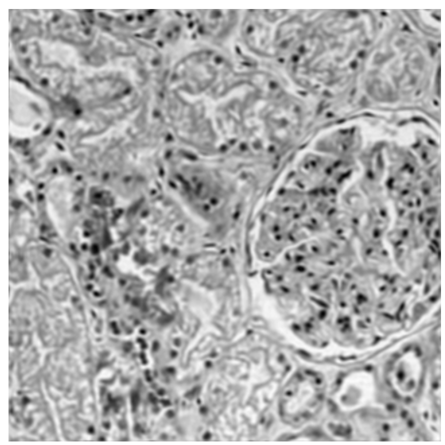
$n = 320^2$ pixels
225 illuminations
64^2 pixels each

$n^2 = 1.05 \times 10^{10}$ vars
$d = 921\,600$ eqns

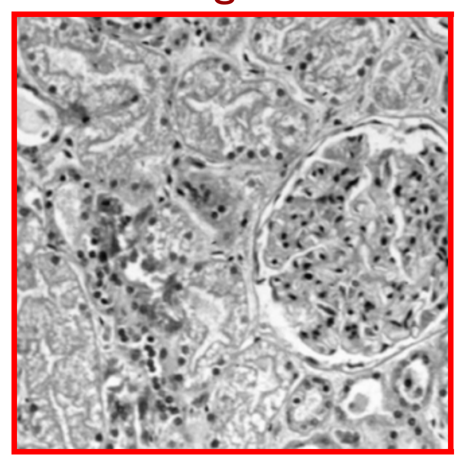
$t = 1\,000$ (18 398 sec)



$t = 10\,000$ (209 879 sec)



original



SDP Relaxations of QAP

$$\text{minimize} \quad \operatorname{trace}(\boldsymbol{A\Pi B\Pi}^*) \qquad \text{(QAP)}$$

$$\text{subject to} \quad \boldsymbol{\Pi} \text{ is an } \mathtt{n} \times \mathtt{n} \text{ permutation matrix}$$

$$\text{minimize} \quad \operatorname{trace}[(\boldsymbol{B} \otimes \boldsymbol{A})\mathsf{Y}] \qquad \text{(QAP-SDP)}$$

$$\text{subject to} \quad \operatorname{trace}_1(\mathsf{Y}) = \mathbf{I}, \quad \operatorname{trace}_2(\mathsf{Y}) = \mathbf{I}, \quad \mathcal{G}(\mathsf{Y}) \geq 0$$

$$\operatorname{vec}(\boldsymbol{P}) = \operatorname{diag}(\mathsf{Y}), \quad \boldsymbol{P}\mathbf{1} = \mathbf{1}, \quad \mathbf{1}^*\boldsymbol{P} = \mathbf{1}^*, \quad \boldsymbol{P} \geq \mathbf{0}$$

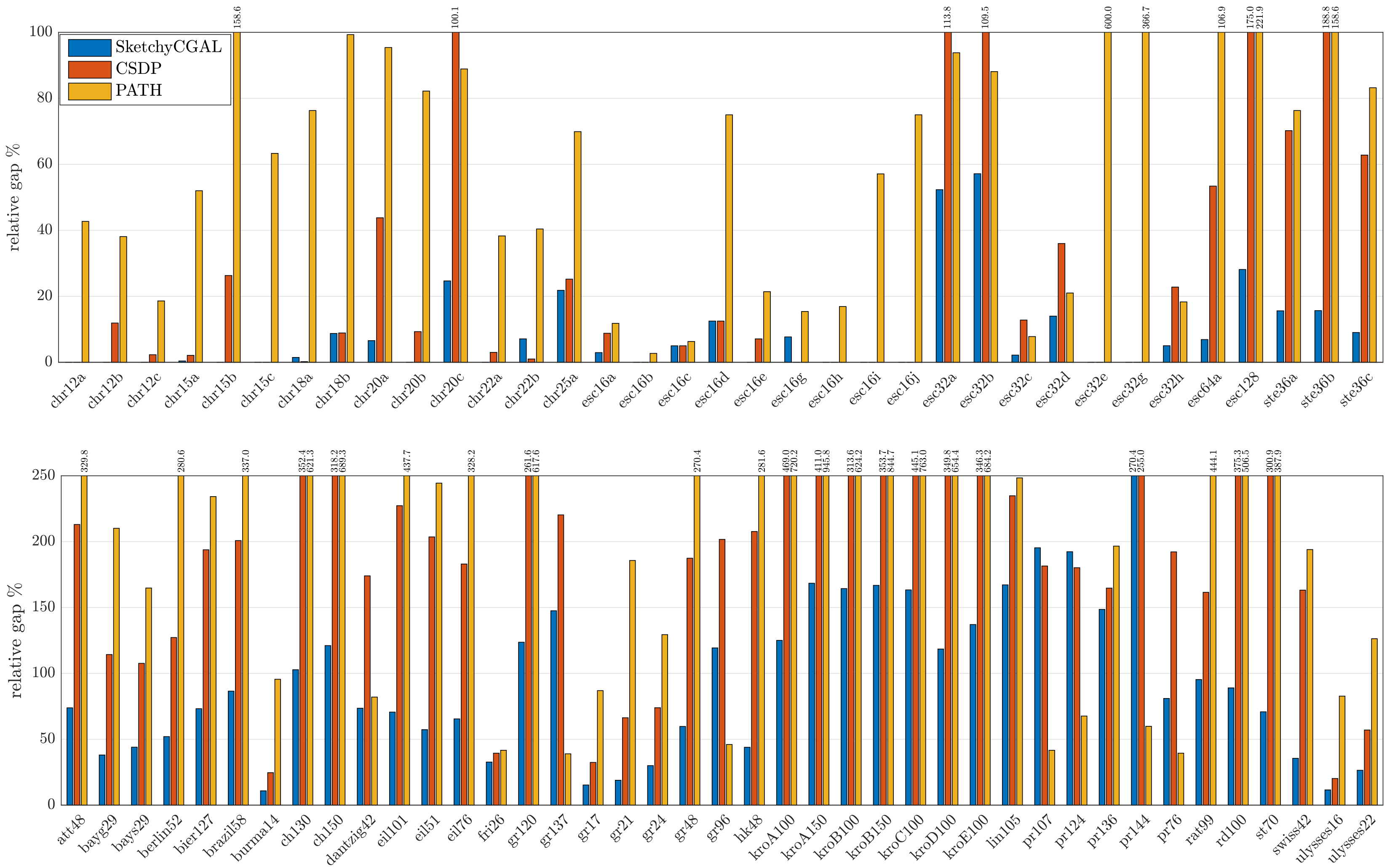
$$\begin{bmatrix} 1 & \operatorname{vec}(\boldsymbol{P})^* \\ \operatorname{vec}(\boldsymbol{P}) & \mathsf{Y} \end{bmatrix} \succcurlyeq \mathbf{0}, \quad \operatorname{trace} \mathsf{Y} = n$$

SDP dimension $n = \mathrm{n}^2$; structure constraints $d = \mathrm{n}^2$

Number of positivity constraints modulated by $\mathcal{G}$

Sources: Zhou et al. 1997; Huang et al. 2014; Bravo-Ferreira et al. 2017; Yurtsever et al. 2018–2021.

QAP Relaxations: Solution Quality



Sources: Zaslavskiy et al. 2009; Bravo-Ferreira et al. 2018; Yurtsever et al. 2018–2021.

To learn more...

E-mail: `jtropp@cms.caltech.edu`

Web: `http://users.cms.caltech.edu/~jtropp`

Related Papers:

- Yurtsever, Tropp, Fercoq, Udell & Cevher, "Scalable semidefinite programming," SIMODS 2021, arXiv arXiv:1912.02949
- Yurtsever, Udell, Tropp & Cevher, "Sketchy decisions: Convex low-rank matrix optimization with optimal storage," AISTATS 2017, arXiv:1702.06838
- Tropp, Yurtsever, Udell & Cevher, "Fixed-rank approximation of a positive-semidefinite matrix from streaming data," NIPS 2017, arXiv:1706.05736
- Tropp, Yurtsever, Udell & Cevher, "Practical sketching algorithms for low-rank matrix approximation," SIMAX 2017, arXiv:1609.00048
- Tropp, Yurtsever, Udell & Cevher, "Streaming low-rank matrix approximation with an application to scientific simulation," SISC 2019, arXiv:1902.08651
- Horstmeyer, Chen, Ou, Ames, Tropp & Yang, "Solving ptychography with a convex relaxation," *New J. Physics*, 2015
- Halko, Martinsson & Tropp, "Finding structure with randomness: Probabilistic algorithms for computing approximate matrix decompositions," *SIREV* 2011, arXiv:0909.4061
- Martinsson & Tropp, "Randomized numerical linear algebra: Foundations and algorithms," *Acta Numerica* 2020, arXiv arXiv:2002.01387