Sketchy Decisions

Joel A. Tropp

Computing + Mathematical Sciences

California Institute of Technology

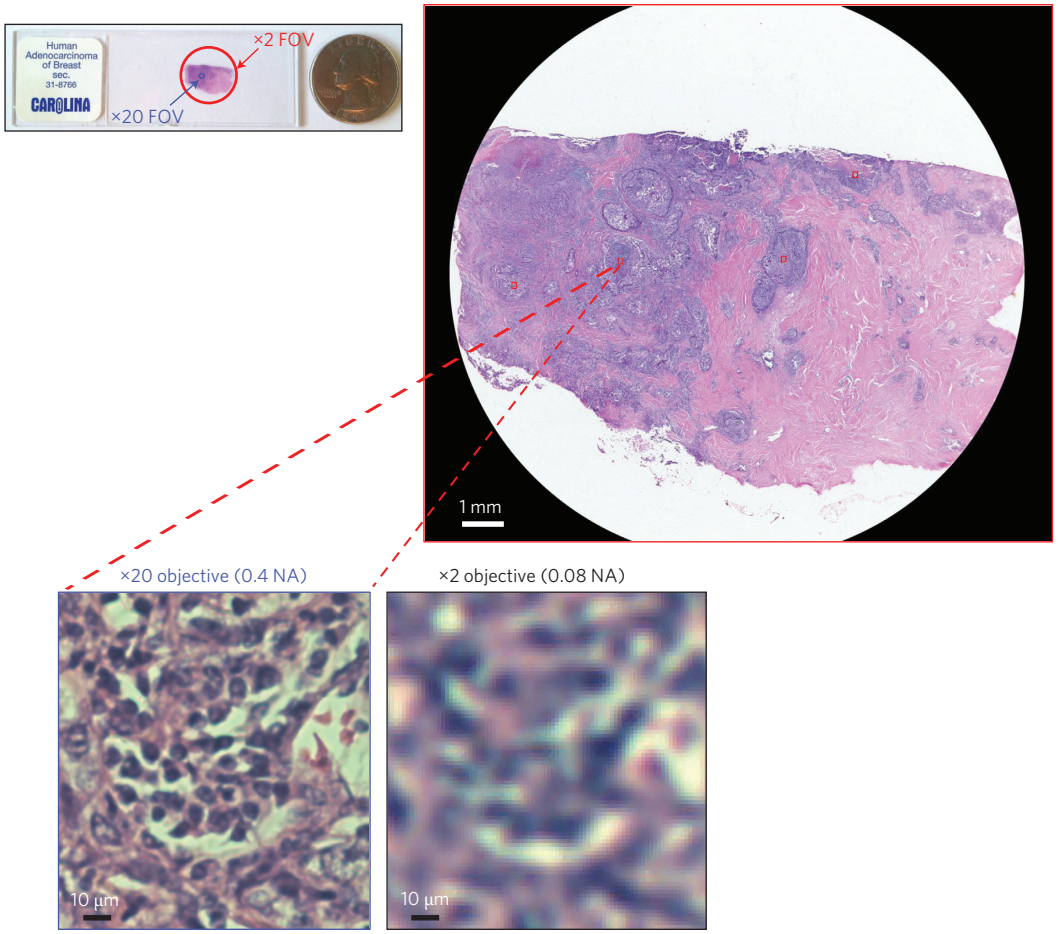
jtropp@cms.caltech.edu

Collaborators: Volkan Cevher (EPFL), Roarke Horstmeyer (Duke),
Quoc Tran-Dinh (UNC), Madeleine Udell (Cornell), Alp Yurtsever (EPFL)

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

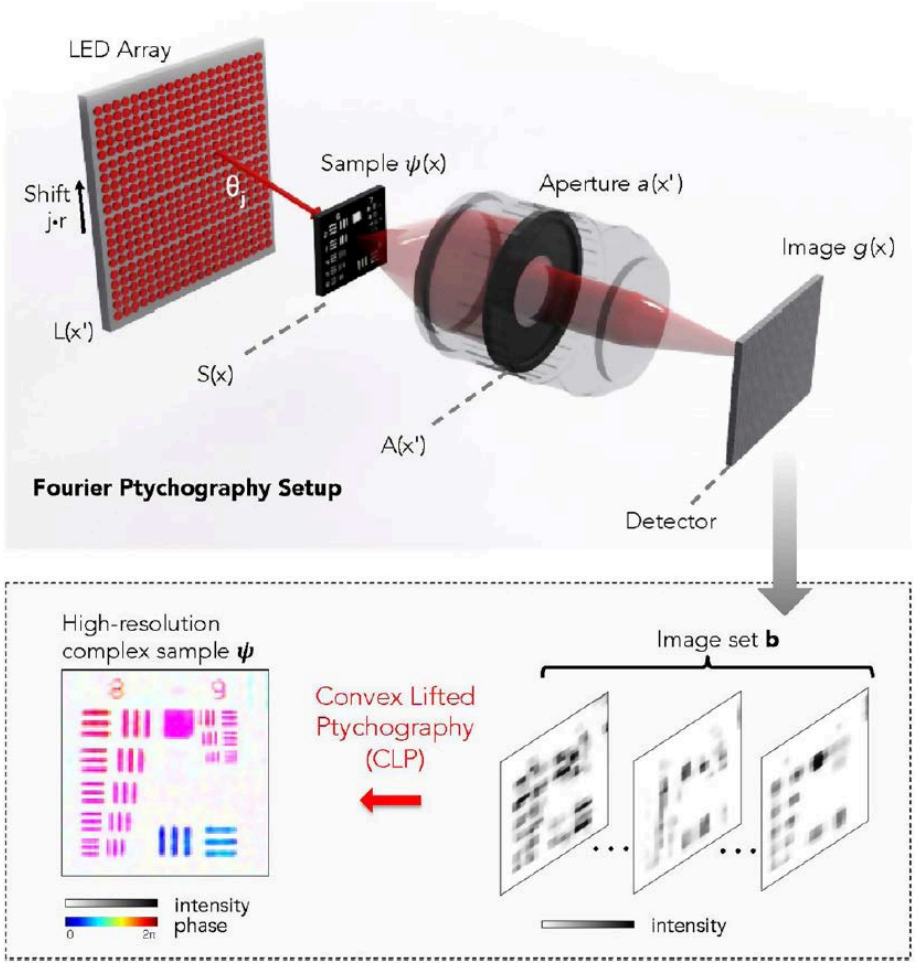
Fourier Ptychography

Microscopy: Field of View / Resolution



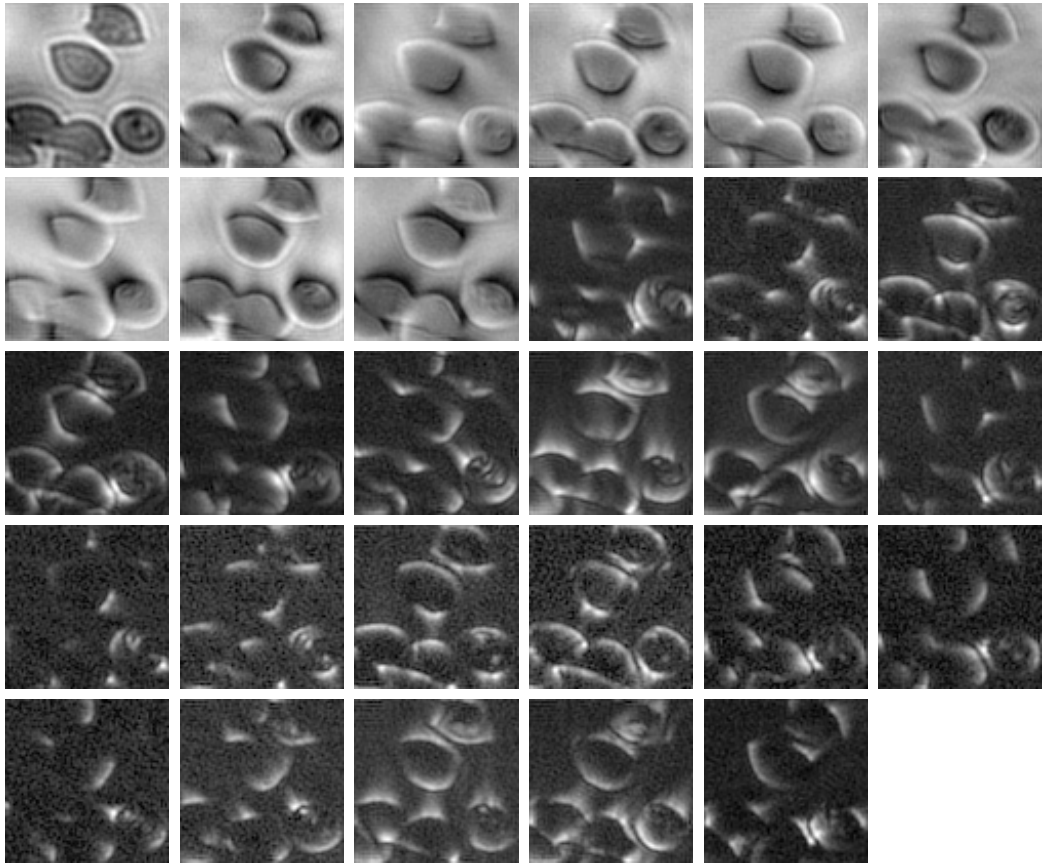
Source: Adapted from Zhang et al. 2013.

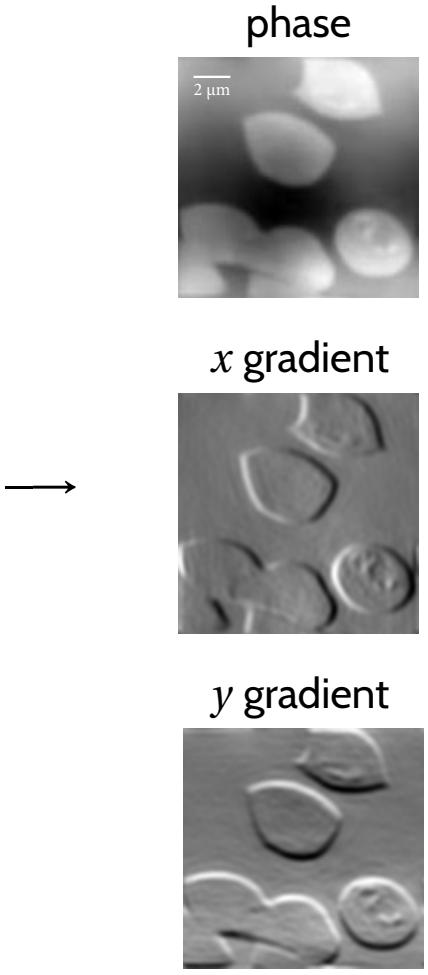
Fourier Ptychography: Field of View + Resolution



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

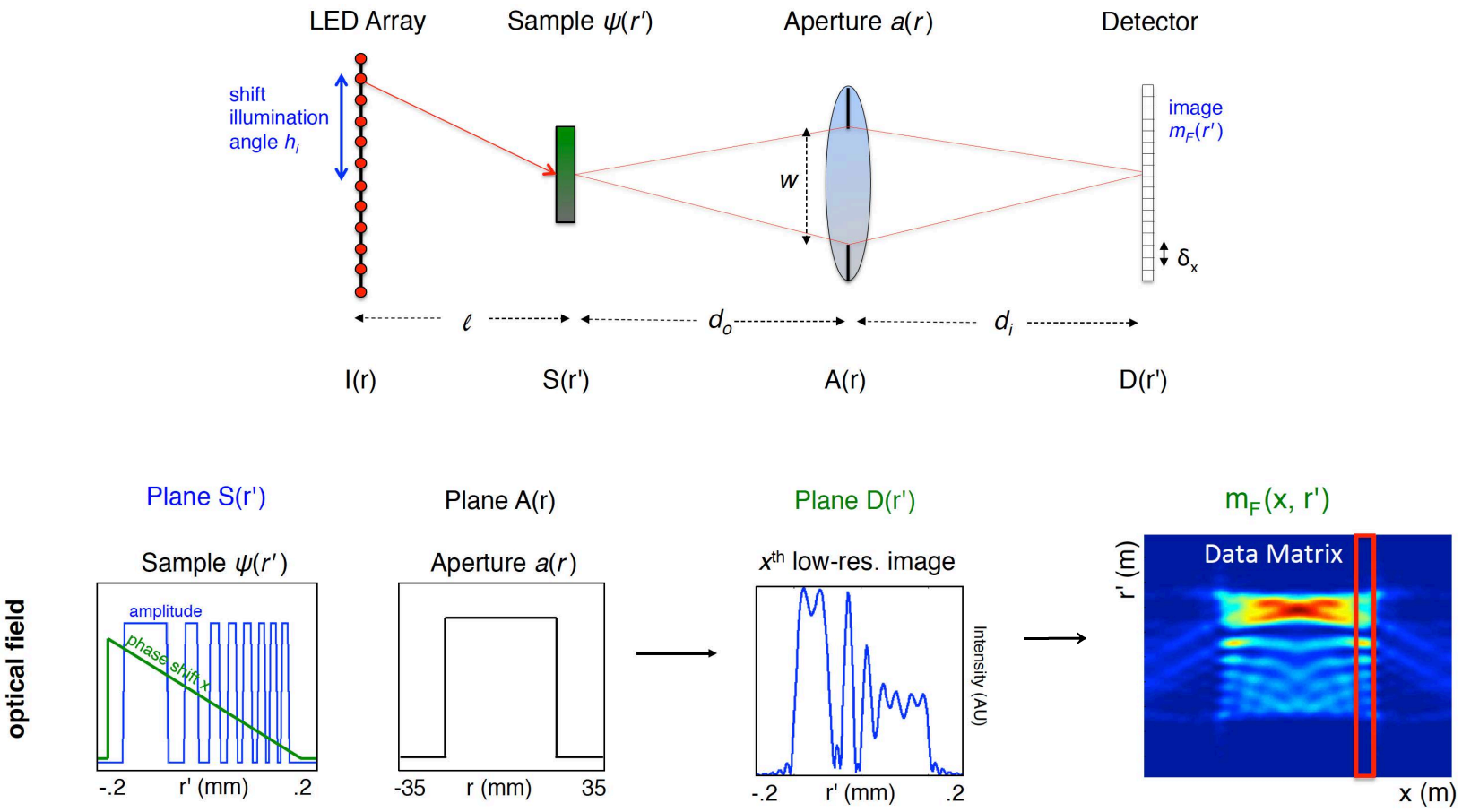
Fourier Ptychography: Malaria Example





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 optimization:

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

- Assume $\text{loss}(\cdot;\ b)$ 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:

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

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

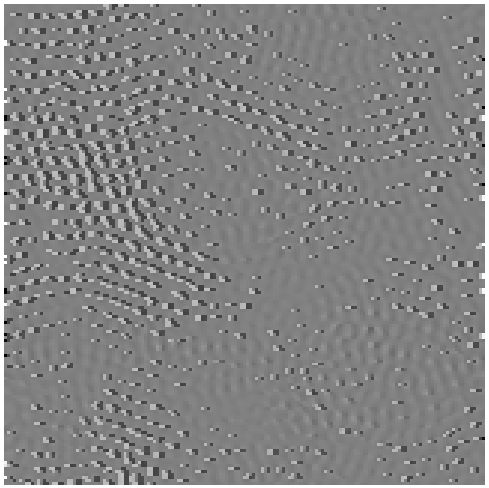
$$\underset{\boldsymbol{X}\in\mathbb{C}^{n\times n}}{\text{minimize}} \quad \sum_{i=1}^{d} \text{loss}\big(\boldsymbol{a}_i^*\boldsymbol{X}\boldsymbol{a}_i;\ b_i\big) \quad \text{subject to} \quad \text{trace}(\boldsymbol{X}) = \alpha; \quad \boldsymbol{X} \text{ psd}$$

- Return maximum eigenvector $\boldsymbol{x}_\star$ of a solution $\boldsymbol{X}_\star$

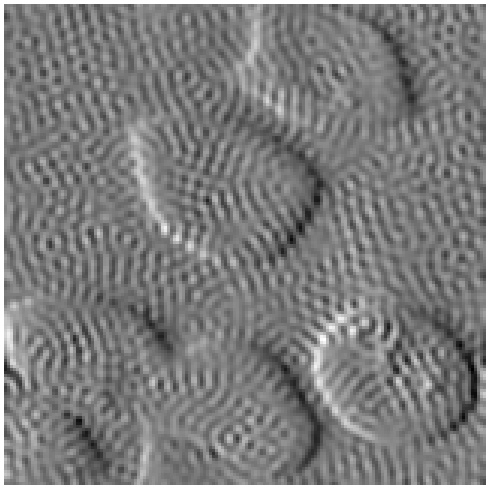
- **Malaria example:** Matrix $\boldsymbol{X}$ has $n^2 = 6.55 \cdot 10^8$ real dof

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

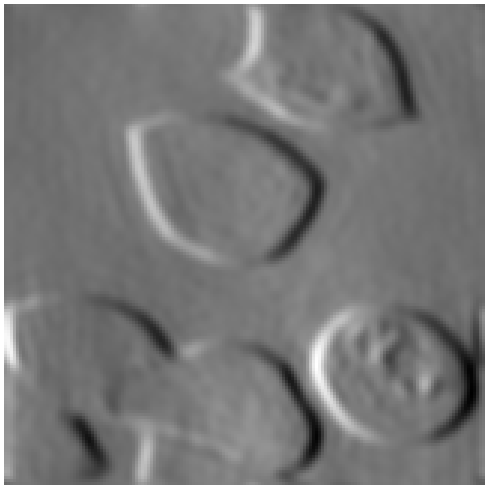
Convexity: Why Bother?



Wirtinger Flow
(not convex)



Burer–Monteiro
(sort of convex)



???
(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.

Optimization with Optimal Storage

Optimization with Optimal Storage

Can we develop algorithms
that reliably solve an optimization problem
using storage that does not exceed
the size of the problem data
or the size of the solution?

Convex Low-Rank Matrix Optimization

$$\underset{\boldsymbol{X} \in \mathbb{H}_n}{\text{minimize}} \quad f(\mathcal{A}\boldsymbol{X}) \quad \text{subject to} \quad \operatorname{trace}(\boldsymbol{X}) = \alpha; \quad \boldsymbol{X} \text{ psd}$$

Details:

- $\mathcal{A} : \mathbb{H}_n \to \mathbb{R}^d$ is a real-linear map on $n \times n$ Hermitian matrices
- $f : \mathbb{R}^d \to \mathbb{R}$ is convex and differentiable

- 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)

Optimal Storage

What kind of storage bounds can we hope for?

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

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

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

- Need $\Theta(n + d)$ storage for output of black-box operations

- Need $\Theta(r\,n)$ storage for rank-r approximate solution of model problem

> `Definition.` An algorithm for the model problem has optimal storage if its working storage is $\Theta(d + r\,n)$ 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 $\Theta(n^2)$
- 2008–Present: **Storage-efficient convex first-order methods**
  - Conditional gradient method (CGM) and extensions
  - Store matrix in low-rank form $O(tn)$; no storage guarantees
- 2009–Present: **Nonconvex heuristics**
  - Burer–Monteiro factorization idea + various nonlinear programming methods
  - Store low-rank matrix factors $\Theta(rn)$
  - For guaranteed solution, need 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; ... **Heuristics:** Burer & Monteiro 2003; Keshavan et al. 2009; Jain et al. 2012; Candès et al. 2014; Bhojanapalli et al. 2015; Boumal et al. 2016; ....

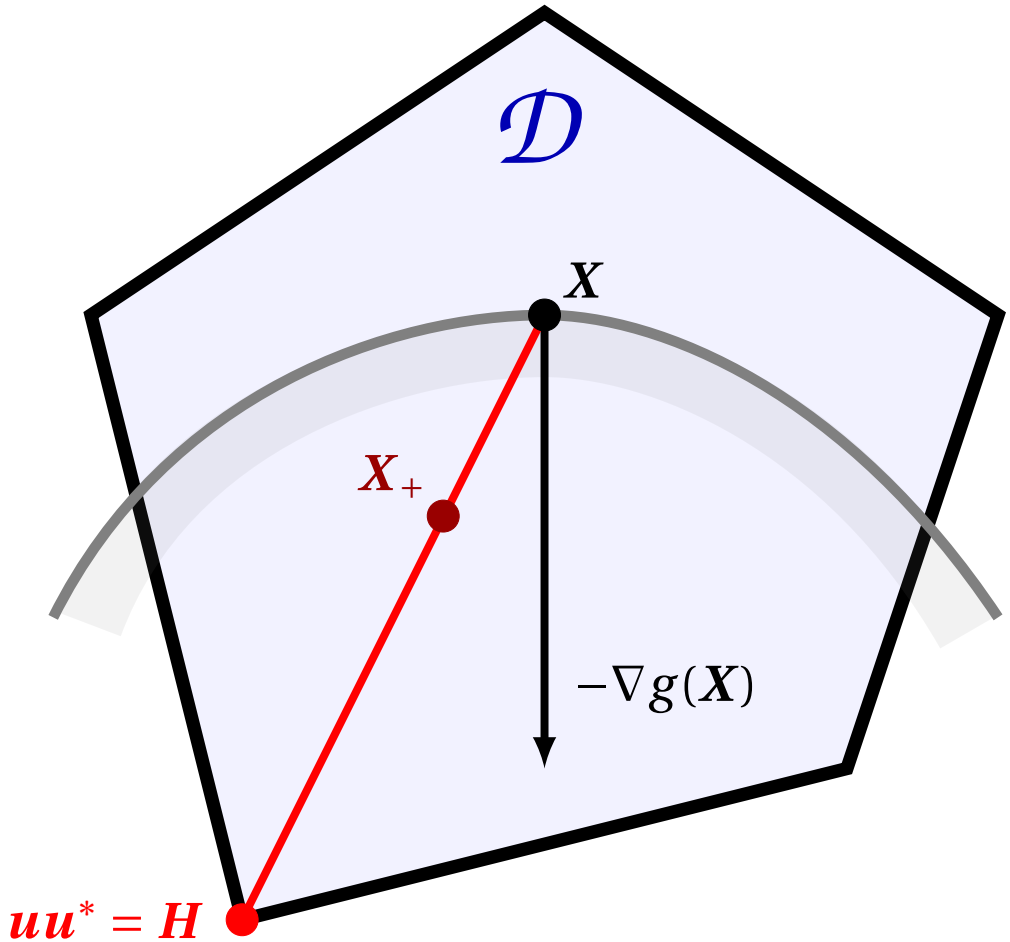
The Challenge

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

Is there an algorithm
that provably computes
a low-rank approximation
to a solution of the model problem
+ has optimal storage guarantees?

Conditional Gradient Method

Geometry of CGM



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

$$X_+ = (1-\eta)X + \eta H$$

$$\{Y : g(Y) \le g(X)\}$$

$$\min_{X \in \mathcal{D}} g(X)$$

$$\mathcal{D} = \{Y \text{ psd} : \operatorname{trace}(Y) = 1\}$$

CGM for the Model Problem

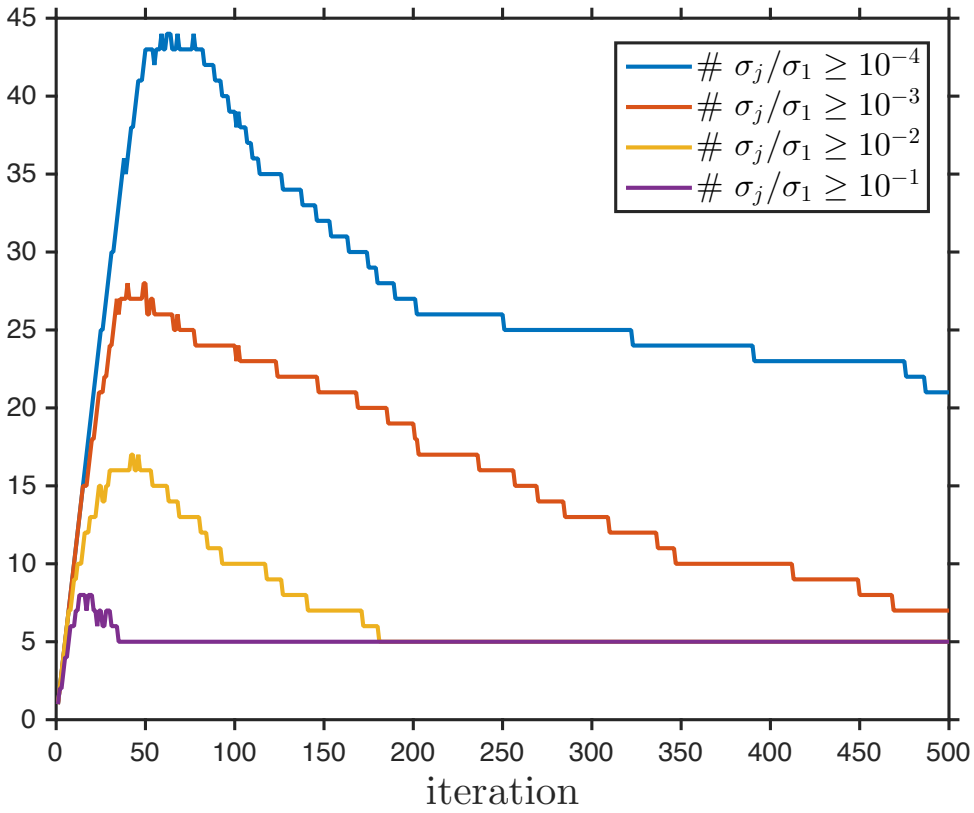
Input: Problem data; suboptimality ε
Output: Approximate solution $\boldsymbol{X}_{\mathrm{cgm}}$

1 **function** CGM
2 $\quad \boldsymbol{X} \leftarrow \mathbf{0}$ ▷ Initialize variable
3 $\quad$ **for** $t \leftarrow 0, 1, 2, 3, \ldots$ **do**
4 $\quad\quad \boldsymbol{u} \leftarrow \mathtt{MinEigVec}(\mathcal{A}^*(\nabla f(\mathcal{A}\boldsymbol{X})))$ ▷ Lanczos!
5 $\quad\quad \boldsymbol{H} \leftarrow -\alpha \boldsymbol{u}\boldsymbol{u}^*$ ▷ Form update direction
6 $\quad\quad$ **if** $\left\langle \boldsymbol{X} - \boldsymbol{H},\ \mathcal{A}^*(\nabla f(\mathcal{A}\boldsymbol{X})) \right\rangle \leq \varepsilon$
7 $\quad\quad\quad$ **then break for** ▷ Stop when ε-suboptimal
8 $\quad\quad \eta \leftarrow 2/(t+2)$ ▷ Update learning rate
9 $\quad\quad \boldsymbol{X} \leftarrow (1-\eta)\boldsymbol{X} + \eta \boldsymbol{H}$ ▷ Update variable
10 $\quad$ **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; Hazan 2008; Clarkson 2010; Jaggi 2013.

Evolution of ε-Rank of CGM Iterates



Comments: Malaria data, quadratic loss, $\alpha = 1{,}400$.

SketchyCGM

Crisis / Opportunity

Crisis:

- CGM needs many iterations to converge to a near-low-rank solution
- The ε-rank of the CGM iterates can increase without bound
- CGM requires high + unpredictable storage
- Typically involves dynamic memory allocation

Opportunity:

- Modify CGM to work with optimal storage!
- Drive the CGM iteration with small "dual" variable $\boldsymbol{z} = \mathcal{A}\boldsymbol{X}$
- 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.

CGM, Redux

Input: Problem data; suboptimality ε
Output: Approximate solution $\boldsymbol{X}_{\mathrm{cgm}}$

1 **function** CGM
2 $\quad \boldsymbol{X} \leftarrow \mathbf{0}_{n \times n}$
3 $\quad$ **for** $t \leftarrow 0, 1, 2, 3, \ldots$ **do**
4 $\quad\quad \boldsymbol{u} \leftarrow \texttt{MinEigVec}(\mathcal{A}^*(\nabla f(\mathcal{A}\boldsymbol{X})))$
5 $\quad\quad \boldsymbol{H} \leftarrow -\alpha \boldsymbol{u}\boldsymbol{u}^*$
6 $\quad\quad$ **if** $\left\langle \mathcal{A}(\boldsymbol{X} - \boldsymbol{H}),\ \nabla f(\mathcal{A}\boldsymbol{X}) \right\rangle \leq \varepsilon$
7 $\quad\quad\quad$ **then break for**
8 $\quad\quad \eta \leftarrow 2/(t+2)$
9 $\quad\quad \boldsymbol{X} \leftarrow (1-\eta)\boldsymbol{X} + \eta \boldsymbol{H}$
10 $\quad$ **return** $\boldsymbol{X}$

Idea: Apply $\mathcal{A}$ to all expressions involving $\boldsymbol{X}$ and $\boldsymbol{H}$...

A Dual Formulation of CGM

Input: Problem data; suboptimality ε

Output: Approximate dual solution $\mathcal{A}\boldsymbol{X}_{\text{cgm}}$

```
1   function DUALCGM
2       AX ← A 0_{n×n}
3       for t ← 0, 1, 2, 3, ... do
4           u ← MinEigVec(A*(∇f(AX)))
5           AH ← A(−α u u*)
6           if ⟨A(X − H), ∇f(AX)⟩ ≤ ε
7               then break for
8           η ← 2/(t + 2)
9           AX ← (1 − η) AX + η AH
10      return AX
```

Idea: Change variables $\boldsymbol{z} = \mathcal{A}\boldsymbol{X}$ and $\boldsymbol{h} = \mathcal{A}\boldsymbol{H}$...

A Dual Formulation of CGM

Input: Problem data; suboptimality ε

Output: Approximate dual solution $\boldsymbol{z}_{\mathrm{cgm}}$

1 **function** DUALCGM
2 $\quad \boldsymbol{z} \leftarrow \mathbf{0}_d$
3 $\quad$ **for** $t \leftarrow 0, 1, 2, 3, \ldots$ **do**
4 $\quad\quad \boldsymbol{u} \leftarrow \texttt{MinEigVec}(\mathcal{A}^*(\nabla f(\boldsymbol{z})))$
5 $\quad\quad \boldsymbol{h} \leftarrow \mathcal{A}(-\alpha \boldsymbol{u}\boldsymbol{u}^*)$
6 $\quad\quad$ **if** $\langle \boldsymbol{z} - \boldsymbol{h},\ \nabla f(\boldsymbol{z}) \rangle \leq \varepsilon$
7 $\quad\quad\quad$ **then break for**
8 $\quad\quad \eta \leftarrow 2/(t+2)$
9 $\quad\quad \boldsymbol{z} \leftarrow (1-\eta)\boldsymbol{z} + \eta \boldsymbol{h}$
10 $\quad$ **return** $\boldsymbol{z}$

A Dual Formulation of CGM

Input: Problem data; suboptimality ε
Output: Approximate dual solution $\boldsymbol{z}_{\mathrm{cgm}}$

function DUALCGM
  $\boldsymbol{z} \leftarrow \mathbf{0}_d$ ▷ Initialize dual variable
  for $t \leftarrow 0, 1, 2, 3, \ldots$ **do**
    $\boldsymbol{u} \leftarrow \texttt{MinEigVec}(\mathcal{A}^*(\nabla f(\boldsymbol{z})))$ ▷ Lanczos!
    $\boldsymbol{h} \leftarrow \mathcal{A}(-\alpha \boldsymbol{u}\boldsymbol{u}^*)$ ▷ Form dual update direction
    if $\langle \boldsymbol{z} - \boldsymbol{h}, \nabla f(\boldsymbol{z}) \rangle \leq \varepsilon$
      then break for
    $\eta \leftarrow 2/(t+2)$
    $\boldsymbol{z} \leftarrow (1-\eta)\boldsymbol{z} + \eta \boldsymbol{h}$ ▷ Update dual variable
  return $\boldsymbol{z}$

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
- 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
- Can compute provably good rank-r approximation $\hat{\boldsymbol{X}}$ from sketch
- Only needs additional storage $\Theta(r\,n)$!

Sources: Woolfe et al. 2008; Clarkson & Woodruff 2009; Halko et al. 2009; Gittens 2011, 2013; Woodruff 2014; Cohen et al. 2015; Boutsidis et al. 2015; Tropp et al. 2016, 2017; ....

SketchyCGM for the Model Problem

Input: Problem data; suboptimality ε; target rank r
Output: Rank-r approximate solution $\hat{\boldsymbol{X}} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^*$ in factored form

1 **function** SketchyCGM
2 SKETCH.INIT(n, r) ▷ Initialize sketch to zero
3 $\boldsymbol{z} \leftarrow \boldsymbol{0}$
4 **for** $t \leftarrow 0, 1, 2, 3, \ldots$ **do**
5 $\boldsymbol{u} \leftarrow \texttt{MinEigVec}(\mathcal{A}^*(\nabla f(\boldsymbol{z})))$
6 $\boldsymbol{h} \leftarrow \mathcal{A}(-\alpha \boldsymbol{u}\boldsymbol{u}^*)$
7 **if** $\langle \boldsymbol{z} - \boldsymbol{h},\ \nabla f(\boldsymbol{z}) \rangle \leq \varepsilon$ **then break for**
8 $\eta \leftarrow 2/(t+2)$
9 $\boldsymbol{z} \leftarrow (1-\eta)\boldsymbol{z} + \eta \boldsymbol{h}$
10 SKETCH.CGMUPDATE$(-\sqrt{\alpha}\boldsymbol{u}, \eta)$ ▷ Update sketch of $\boldsymbol{X}$
11 $(\boldsymbol{V}, \boldsymbol{\Lambda}) \leftarrow$ SKETCH.RECONSTRUCT$()$ ▷ Approx. eigendecomp of $\boldsymbol{X}$
12 **return** $(\boldsymbol{V}, \boldsymbol{\Lambda})$

Source: Yurtsever et al. 2017.

Methods for SKETCH Object

1 **function** SKETCH.INIT(n, r) ▷ Rank-r approx of $n \times n$ psd matrix
2 $k \leftarrow 2r$
3 $\boldsymbol{\Omega} \leftarrow \texttt{randn}(\mathbb{C}, n, k)$
4 $\boldsymbol{Y} \leftarrow \texttt{zeros}(n, k)$

5 **function** SKETCH.CGMUPDATE($\boldsymbol{s}$, θ)
6 $\boldsymbol{Y} \leftarrow (1-\theta)\boldsymbol{Y} + \theta \boldsymbol{s}(\boldsymbol{s}^*\boldsymbol{\Omega})$ ▷ Average $\boldsymbol{s}\boldsymbol{s}^*$ into sketch

7 **function** SKETCH.RECONSTRUCT( )
8 $\boldsymbol{C} \leftarrow \texttt{chol}(\boldsymbol{\Omega}^*\boldsymbol{Y})$ ▷ Cholesky decomposition
9 $\boldsymbol{Z} \leftarrow \boldsymbol{Y}/\boldsymbol{C}$ ▷ Solve least-squares problems
10 $(\boldsymbol{U}, \boldsymbol{\Sigma}, \sim) \leftarrow \texttt{svds}(\boldsymbol{Z}, r)$ ▷ Compute r-truncated SVD
11 **return** $(\boldsymbol{U}, \boldsymbol{\Sigma}^2)$ ▷ Return eigenvalue factorization

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

Less Filling / Great Taste

Theorem 1 (YUTC 2016). *SketchyCGM has the following properties:*

- *SketchyCGM has optimal storage guarantee* $\Theta(d + r\,n)$
- *SketchyCGM produces an* ε*-suboptimal objective value after* $O(\varepsilon^{-1})$ *iterations*
- *Suppose CGM produces iterates* $\boldsymbol{X}_t$ *that converge to* $\boldsymbol{X}_{\mathrm{cgm}}$. *Then SketchyCGM produces rank-*r *iterates* $\hat{\boldsymbol{X}}_t$ *that satisfy*

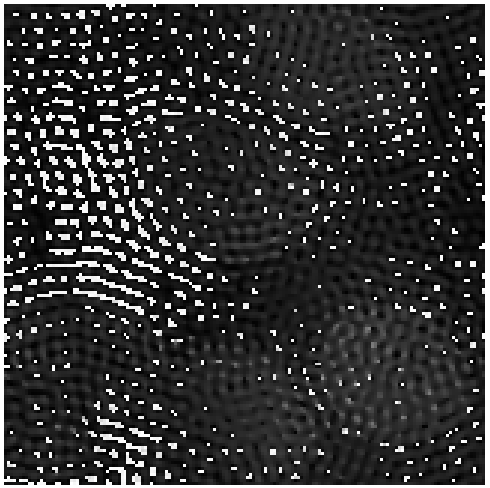
$$\limsup_{t\to\infty} \mathbb{E}\left\|\hat{\boldsymbol{X}}_t - \boldsymbol{X}_{\mathrm{cgm}}\right\|_{S_1} \quad \leq \quad \mathrm{const}\cdot\left\|\boldsymbol{X}_{\mathrm{cgm}} - [\boldsymbol{X}_{\mathrm{cgm}}]_r\right\|_{S_1}$$

In particular, if $\mathrm{rank}(\boldsymbol{X}_{\mathrm{cgm}}) \leq r$, *then* $\mathbb{E}\left\|\hat{\boldsymbol{X}}_t - \boldsymbol{X}_{\mathrm{cgm}}\right\|_{S_1} \to 0$

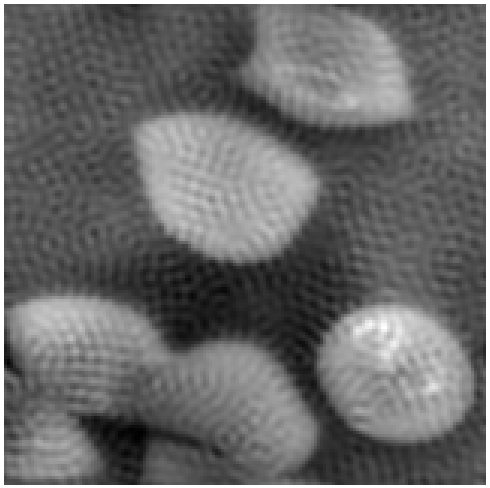
Source: "Everything you always wanted in an algorithm. And less."
https://www.youtube.com/watch?v=0agZEMEpiVI.

Performance of SketchyCGM

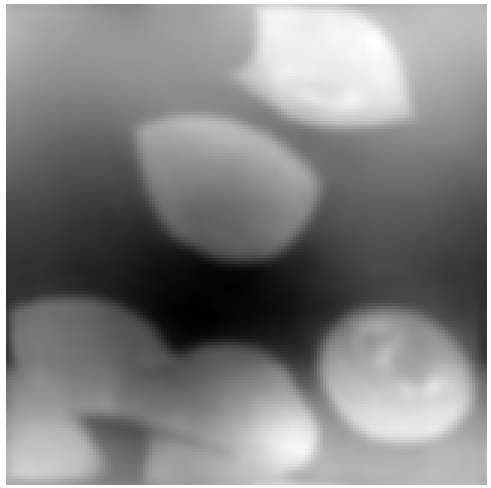
Fourier Ptychography, Redux



Wirtinger Flow



Burer–Monteiro



SketchyCGM

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

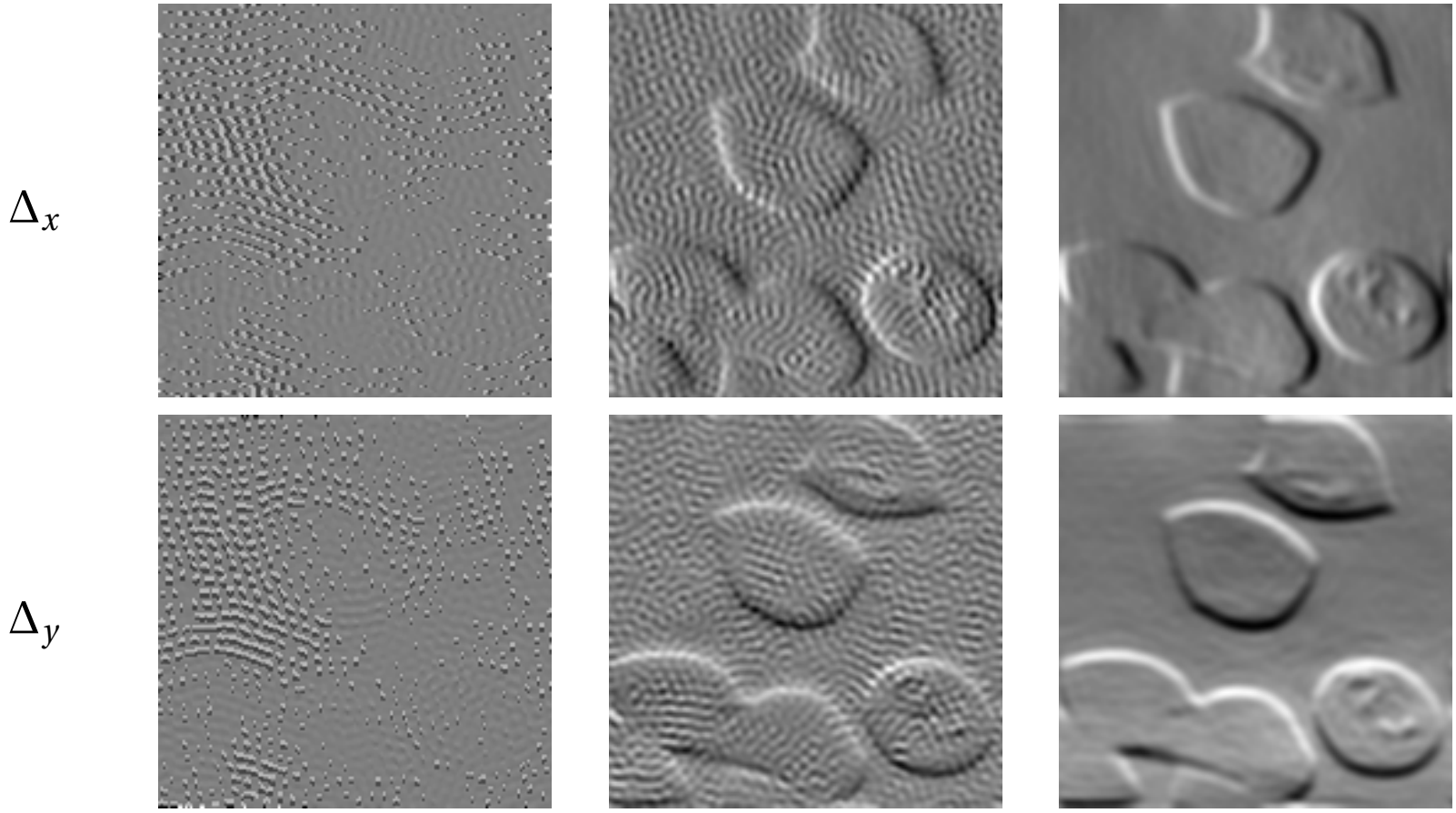
image size $n = 160 \times 160$ pixels; matrix size $n^2 = 6.55 \cdot 10^8$

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

quadratic loss

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

Fourier Ptychography: Malaria Phase Gradients



Convex Low-Rank Matrix Completion

- Suppose $\boldsymbol{X}_\natural \in \mathbb{R}^{m \times n}$ is a rank-r matrix
- Observe a subset of entries + noise:

$$b_{ij} = (\boldsymbol{X}_\natural)_{ij} + \xi_{ij} \quad \text{for } (i,j) \in E$$

- Matrix completion via convex programming:

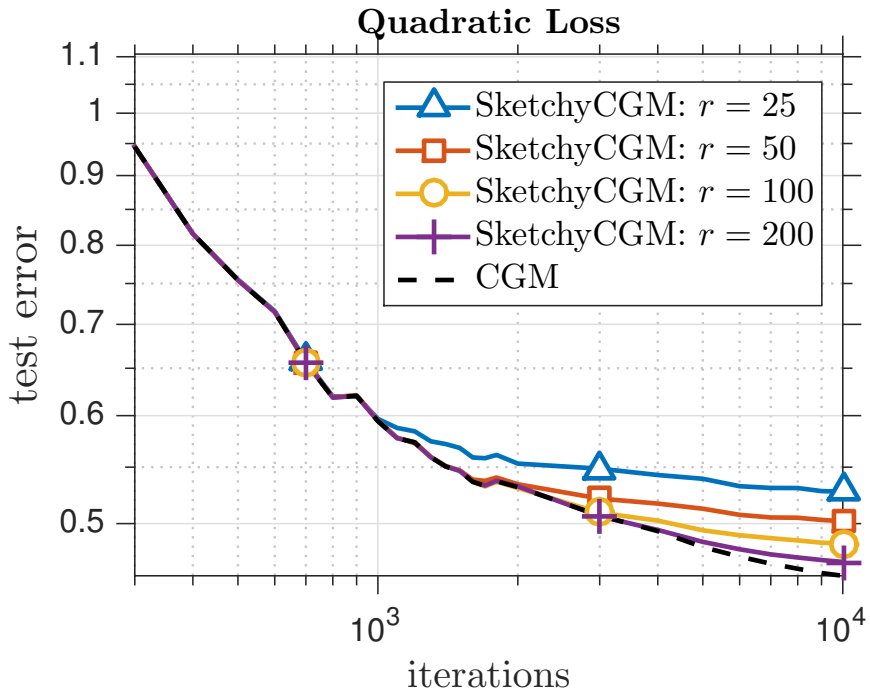
$$\underset{\boldsymbol{X} \in \mathbb{R}^{m \times n}}{\text{minimize}} \quad \sum_{(i,j) \in E} \text{loss}\big(x_{ij};\, b_{ij}\big) \quad \text{subject to} \quad \|\boldsymbol{X}\|_{S_1} \le \alpha$$

- Matrix $\boldsymbol{X}_\natural$ has about $r(m+n)$ degrees of freedom
- Convex method often effective when $\#E = \Theta(r(m+n))$
- But decision variable $\boldsymbol{X}$ has mn degrees of freedom!
- SketchyCGM works with storage $\Theta(\#E + r(m+n))$

Sources: Srebro et al. 2004; Candès & Recht 2010; Yurtsever et al. 2017.

MovieLens 10M

- $m = 71,567$ users, $n = 10,681$ movies, $d = 10^7$ ratings, dim. $mn = 7.64 \cdot 10^8$



Approximate storage costs

Rank (r)	SKETCHYCGM
25	$3.28 \cdot 10^7$
50	$4.51 \cdot 10^7$
100	$6.98 \cdot 10^7$
200	$1.19 \cdot 10^8$

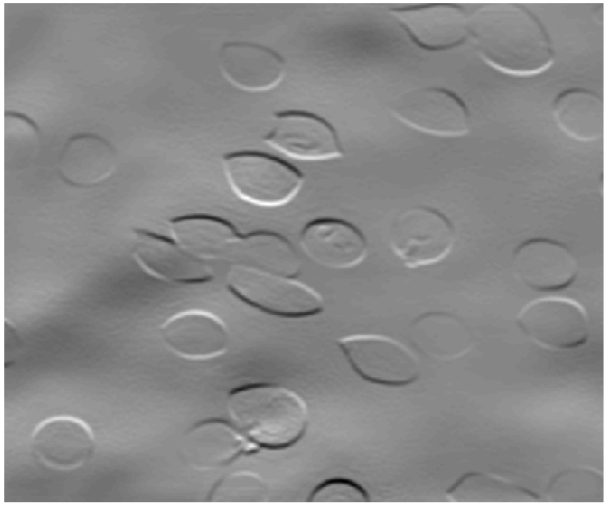
Source: Harper & Konstan 2015; Yurtsever et al. 2017.

Denouement

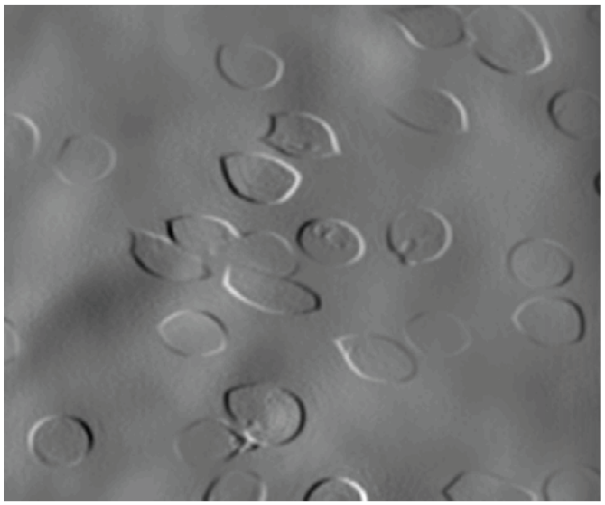
Beyond...

- Other low-rank matrix optimization problems!
- More effective algorithms!
- Many applications!
- Beyond low-rank matrix optimization!

A Large-Scale Problem



x phase gradient



y phase gradient

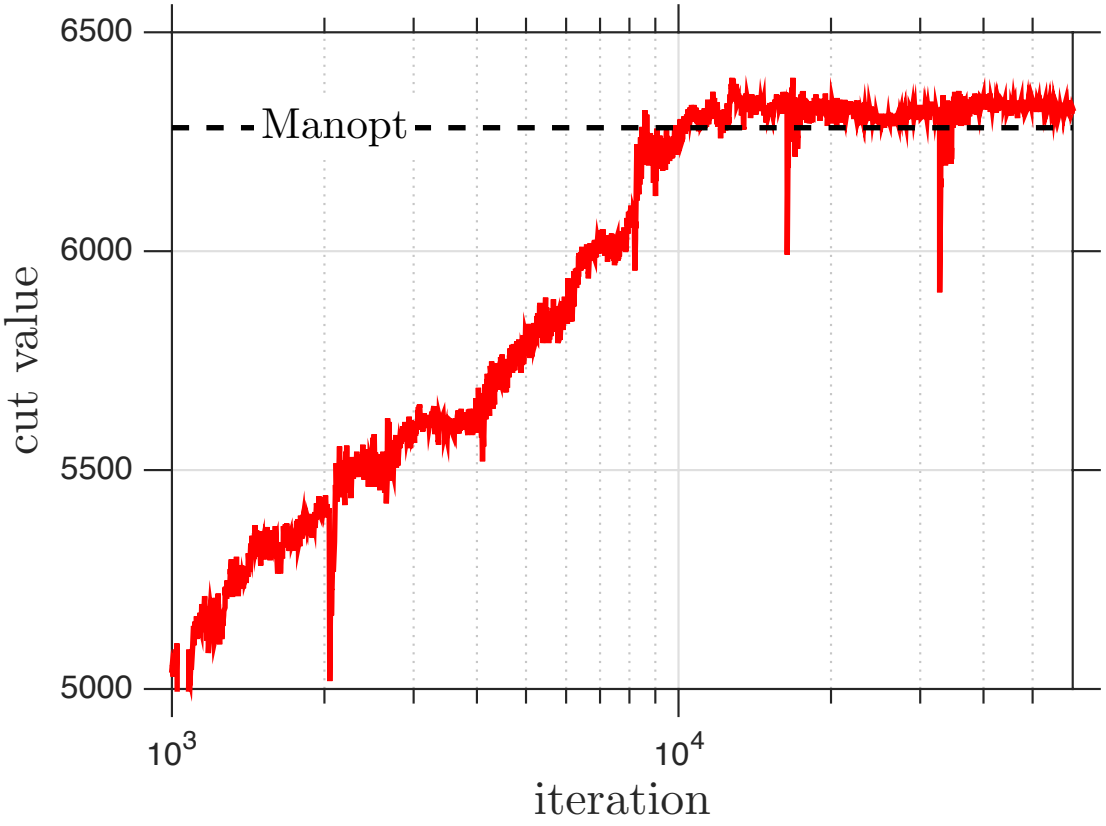
29 illuminations; 250×250 pixels each; $d = 1.81 \cdot 10^6$ measurements

image size $n = 501 \times 501$ pixels; matrix size $n^2 = 6.25 \cdot 10^{10}$

solution via SKETCHYUPD

Sources: Cevher et al. 2017.

The MAXCUT SDP



MAXCUT SDP; sparse graph with 10,000 nodes (G67); via SKETCHYUPD

Sources: Goemans & Williamson 1995; Boumal 2015; Cevher et al. 2017.

To learn more...

E-mail: jtropp@cms.caltech.edu

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

Papers:

- Halko, Martinsson, & Tropp, "Finding structure with randomness: Probabilistic algorithms for computing approximate matrix decompositions," *SIAM Review*, 2011
- Horstmeyer el al. "Solving ptychography with a convex relaxation," *New J. Physics*, 2015
- Tropp, Yurtsever, Udell, & Cevher, "Randomized single-view algorithms for low-rank matrix approximation," Caltech ACM TR 2017-01, arXiv cs.NA 1609.00048, updates coming soon!
- Yurtsever, Udell, Tropp, & Cevher, "Sketchy decisions: Convex low-rank matrix optimization with optimal storage," AISTATS 2017, arXiv ma.OC arXiv:1702.06838
- More to come!