Lecture 3:
Minimizing Large Sums

Peter Richtárik

Graduate School in Systems, Optimization, Control and Networks
Belgium 2015

Motivation: Machine Learning & Empirical Risk Minimization

Training Linear Predictors

Statistical Nature of Data

Label

$$(A_i, y_i) \sim Distribution$$

Data
(e.g., image, text, measurements, ...)

$$A_i \in \mathbb{R}^{d \times m}$$
$$y_i \in \mathbb{R}^m$$

Prediction of Labels from Data

Find $w \in \mathbb{R}^d$ Linear predictor

Such that when (data, label) pair is drawn from the distribution

$$(A_i, y_i) \sim Distribution$$

Then

$$A_i^\top w \approx y_i$$

Predicted label

True label

Measure of Success

$$loss(a, b)$$

data

label

We want the expected loss (=risk) to be small:

$$\mathbf{E}\left[loss(A_i^\top w, y_i)\right]$$

$$(A_i, y_i) \sim Distribution$$

Finding a Linear Predictor via Empirical Risk Minimization (ERM)

Draw i.i.d. data (samples) from the distribution

$$(A_1, y_1), (A_2, y_2), \ldots, (A_n, y_n) \sim Distribution$$

Output predictor which minimizes the empirical risk:

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} loss(A_i^\top w, y_i)$$

Primal and Dual Problems

Primal Problem: ERM

$\phi_i : \mathbb{R}^m \mapsto \mathbb{R}$

$\frac{1}{\gamma}$-smooth and convex

regularization parameter

$$\min_{w \in \mathbb{R}^d} \left[P(w) \equiv \frac{1}{n} \sum_{i=1}^{n} \phi_i(A_i^\top w) + \lambda g(w) \right]$$

d = # features (parameters)

n = # samples

$A_i \in \mathbb{R}^{d \times m}$

1 - strongly convex function (regularizer)

Is the difficulty in *n* or *d*?

- **Big n**
  - Work in the **primal**
  - Process one loss function (= one example) at a time
  - Type of methods: stochastic gradient descent (modern variants: SAG, SVRG, S2GD, mS2GD, SAGA, S2CD, MISO, FINITO, …)
- **Big d**
  - Work in the **primal**
  - Process one primal variable at a time
  - Type of methods: randomized coordinate descent (e.g., Hydra, Hydra2)
- **Big n**
  - Work in the **dual**
  - Process one dual variable (=one example) at a time
  - Type of methods: randomized coordinate descent (modern variants: RCDM, PCDM, Shotgun, SDCA, APPROX, Quartz, ALPHA, SDNA, SPDC, ASDCA, …)
  - E.g. SDCA = run coordinate descent on the dual problem

Dual Problem

$$D(\alpha) \equiv -\lambda g^* \underbrace{\left(\frac{1}{\lambda n} \sum_{i=1}^{n} A_i \alpha_i \right)}_{\in \mathbb{R}^d} - \frac{1}{n} \sum_{i=1}^{n} \phi_i^* (\overbrace{-\alpha_i}^{\in \mathbb{R}^m})$$

1 – smooth & convex

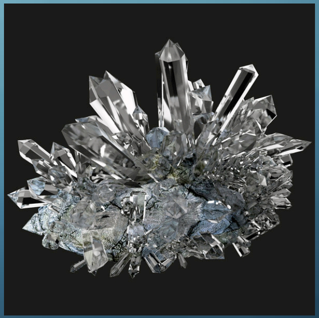
γ - strongly convex

$$g^*(w') = \max_{w \in \mathbb{R}^d} \left\{ (w')^\top w - g(w) \right\}$$

$$\phi_i^*(a') = \max_{a \in \mathbb{R}^m} \left\{ (a')^\top a - \phi_i(a) \right\}$$

$$\max_{\alpha = (\underbrace{\alpha_1}_{\in \mathbb{R}^m}, \ldots, \underbrace{\alpha_n}_{\in \mathbb{R}^m}) \in \mathbb{R}^N = \mathbb{R}^{nm}} D(\alpha)$$

An Efficient Dual Method



Zheng Qu, P.R. and Tong Zhang
Randomized dual coordinate ascent with arbitrary sampling
In NIPS 2015 (arXiv:1411.5873)

Empirical Risk Minimization

Primal Problem: ERM

$1/\gamma$ - smooth & convex

regularization parameter

$$\min_{w \in \mathbb{R}^d} \left[P(w) \equiv \frac{1}{n} \sum_{i=1}^{n} \phi_i(A_i^\top w) + \lambda g(w) \right]$$

d = # features (parameters)

n = # samples

1 - strongly convex function (regularizer)

Assumption 1

The loss functions $\phi_i : \mathbb{R}^m \mapsto \mathbb{R}$ are $\frac{1}{\gamma}$-smooth:

$$\|\nabla \phi_i(a) - \nabla \phi_i(a')\| \leq \frac{1}{\gamma} \|a - a'\|, \quad a, a' \in \mathbb{R}^m$$

Lipschitz constant of the gradient of the function

Assumption 2

Regularizer is 1-strongly convex

$$g(w) \geq g(w') + \langle \nabla g(w'), w - w' \rangle + \frac{1}{2}\|w - w'\|^2, \quad w, w' \in \mathbb{R}^d$$

subgradient

Dual Problem

$$D(\alpha) \equiv -\lambda g^*\Bigg(\underbrace{\frac{1}{\lambda n}\sum_{i=1}^{n} A_i\alpha_i}_{\in\,\mathbb{R}^d}\Bigg) - \frac{1}{n}\sum_{i=1}^{n}\phi_i^*(\overbrace{-\alpha_i}^{\in\,\mathbb{R}^m})$$

1 – smooth & convex

γ - strongly convex

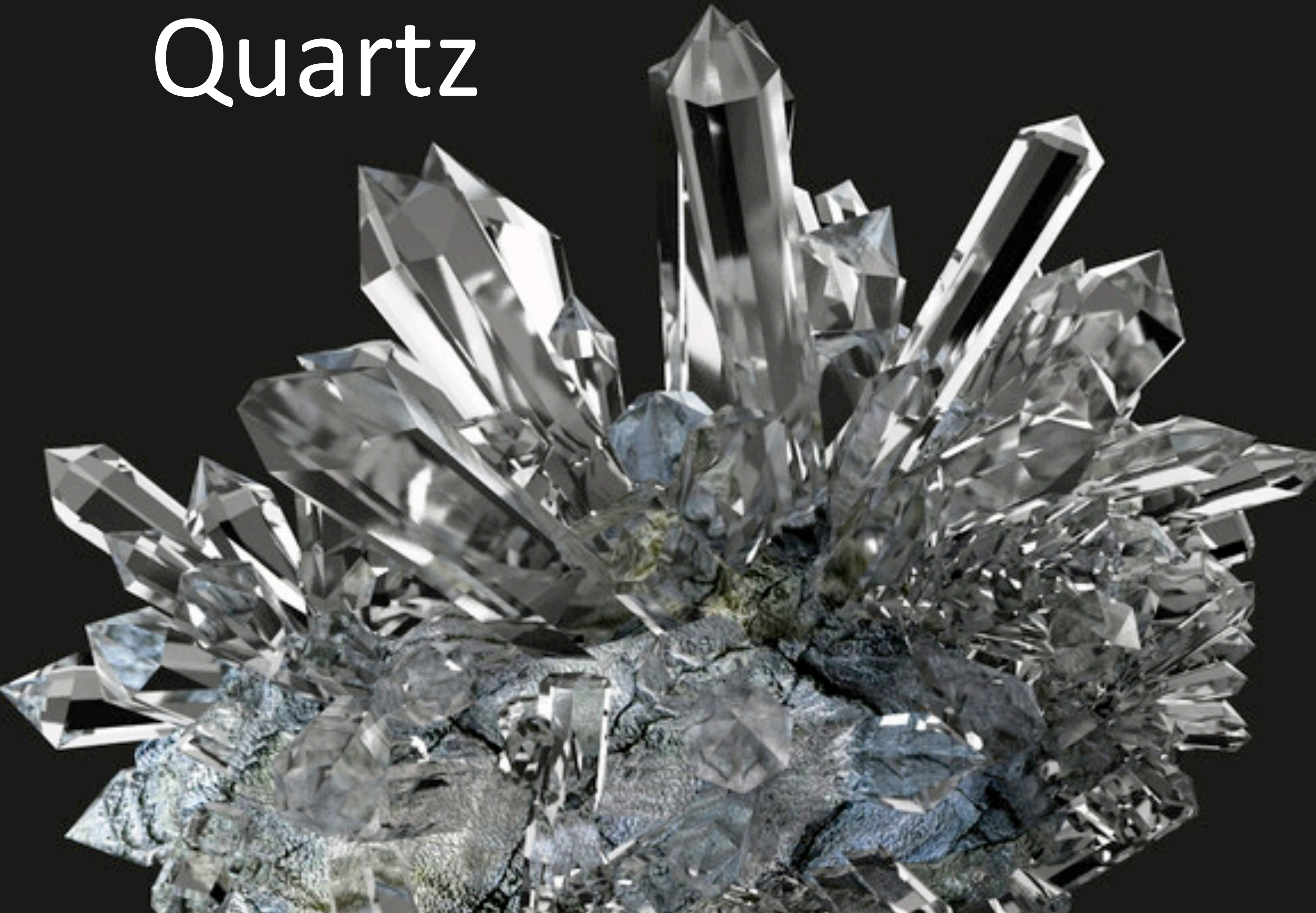
$$g^*(w') = \max_{w\in\mathbb{R}^d}\left\{(w')^\top w - g(w)\right\}$$

$$\phi_i^*(a') = \max_{a\in\mathbb{R}^m}\left\{(a')^\top a - \phi_i(a)\right\}$$

$$\max_{\alpha=(\underbrace{\alpha_1}_{\in\,\mathbb{R}^m},\ldots,\underbrace{\alpha_n}_{\in\,\mathbb{R}^m})\in\mathbb{R}^N=\mathbb{R}^{nm}} D(\alpha)$$

The Algorithm

Quartz



Fenchel Duality

$$\bar{\alpha} = \frac{1}{\lambda n} \sum_{i=1}^{n} A_i \alpha_i$$

$$P(w) - D(\alpha) \quad = \quad \lambda \left(g(w) + g^* \left(\bar{\alpha}\right)\right) + \frac{1}{n} \sum_{i=1}^{n} \phi_i(A_i^\top w) + \phi_i^*(-\alpha_i) =$$

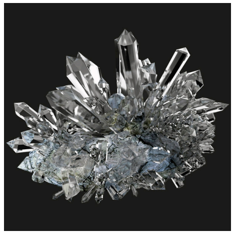
$$\underbrace{\lambda(g(w) + g^* \left(\bar{\alpha}\right) - \langle w, \bar{\alpha} \rangle)}_{\geq 0} + \frac{1}{n} \sum_{i=1}^{n} \underbrace{\phi_i(A_i^\top w) + \phi_i^*(-\alpha_i) + \left\langle A_i^\top w, \alpha_i \right\rangle}_{\geq 0}$$

Weak duality

Optimality conditions

$$w = \nabla g^*(\bar{\alpha}) \qquad\qquad \alpha_i = -\nabla \phi_i(A_i^\top w)$$

The Algorithm



$$(\alpha^t, w^t) \quad \Rightarrow \quad (\alpha^{t+1}, w^{t+1})$$

Quartz: Bird's Eye View

STEP 1: PRIMAL UPDATE

$$w^{t+1} \leftarrow (1-\theta)w^t + \theta \nabla g^*(\bar{\alpha}^t)$$

STEP 2: DUAL UPDATE

Choose a random set S_t of dual variables

For $i \in S_t$ do

$$p_i = \mathbf{P}(i \in S_t)$$

$$\alpha_i^{t+1} \leftarrow \left(1 - \frac{\theta}{p_i}\right)\alpha_i^t + \frac{\theta}{p_i}\left(-\nabla \phi_i(A_i^\top w^{t+1})\right)$$

Algorithm 1 Quartz

Parameters: proper random sampling $\hat{S}$ and a positive vector $v \in \mathbb{R}^n$

Initialization: Choose $\alpha^0 \in \mathbb{R}^N$ and $w^0 \in \mathbb{R}^d$

Set $p_i = \mathbb{P}(i \in \hat{S})$, $\theta = \min_i \frac{p_i \lambda \gamma n}{v_i + \lambda \gamma n}$ and $\bar{\alpha}^0 = \frac{1}{\lambda n} \sum_{i=1}^n A_i \alpha_i^0$

Convex combination constant

for $t \geq 1$ **do**

$\boxed{w^t = (1-\theta) w^{t-1} + \theta \nabla g^*(\bar{\alpha}^{t-1})}$ **STEP 1**

$\alpha^t = \alpha^{t-1}$

Generate a random set $S_t \subseteq [n]$, following the distribution of $\hat{S}$

for $i \in S_t$ **do**

Calculate $\Delta \alpha_i^t$ using one of the following options:

~~**Option I** :~~

$\cancel{\Delta \alpha_i^t = \arg\max_{\Delta \in \mathbb{R}^m} \left[-\phi_i^*(-(\alpha_i^{t-1} + \Delta)) - \nabla g^*(\bar{\alpha}^{t-1})^\top A_i \Delta - \frac{v_i \|\Delta\|^2}{2\lambda n} \right]}$

Option II :

$\Delta \alpha_i^t = -\theta p_i^{-1} \alpha_i^{t-1} - \theta p_i^{-1} \nabla \phi_i(A_i^\top w^t)$

$\alpha_i^t = \alpha_i^{t-1} + \Delta \alpha_i^t$

end for

STEP 2

$\bar{\alpha}^t = \bar{\alpha}^{t-1} + (\lambda n)^{-1} \sum_{i \in S_t} A_i \Delta \alpha_i^t$

Just maintaining $\bar{\alpha}$

end for

Output: w^t, α^t

Other Stochastic Dual Methods for ERM

Randomized Dual Coordinate Ascent Methods for ERM

Algorithm	1-nice	1-optimal	τ-nice	arbitrary	additional speedup	direct p-d analysis	acceleration
SDCA	●						
mSDCA	●		●		●		
ASDCA	●		●				●
AccProx-SDCA	●						●
DisDCA	●		●				
Iprox-SDCA	●	●					
APCG	●						●
SPDC	●	●	●			●	●
Quartz	●	●	●	●	●	●	

SDCA:	SS Shwartz & T Zhang, 09/2012
mSDCA	M Takac, A Bijral, P R & N Srebro, 03/2013
ASDCA:	SS Shwartz & T Zhang, 05/2013
AccProx-SDCA:	SS Shwartz & T Zhang, 10/2013
DisDCA:	T Yang, 2013
Iprox-SDCA:	P Zhao & T Zhang, 01/2014
APCG:	Q Lin, Z Lu & L Xiao, 07/2014
SPDC:	Y Zhang & L Xiao, 09/2014
Quartz:	Z Qu, P R & T Zhang, 11/2014

Complexity

Assumption 3
(Expected Separable Overapproximation)

Parameters $v_1, \dots, v_n$ satisfy:

$$\mathbf{E} \left\| \sum_{i \in S_t} A_i \alpha_i \right\|^2 \leq \sum_{i=1}^n p_i v_i \|\alpha_i\|^2$$

inequality must hold for all

$\alpha_1, \dots, \alpha_n \in \mathbb{R}^m$

$p_i = \mathbf{P}(i \in S_t)$

Complexity

$$\theta = \min_i \frac{p_i \lambda \gamma n}{v_i + \lambda \gamma n}$$

Theorem [Qu, R & Zhang 14]

$$\mathbf{E}[P(w^t) - D(\alpha^t)] \leq (1-\theta)^t (P(w^0) - D(\alpha^0))$$

$$t \;\geq\; \max_i \left(\frac{1}{p_i} + \frac{v_i}{p_i \lambda \gamma n} \right) \log \left(\frac{P(w^0) - D(\alpha^0)}{\epsilon} \right)$$

$$\Rightarrow \quad \mathbf{E}\left[P(w^t) - D(\alpha^t)\right] \leq \epsilon$$

Example

Data: $n = 7 \times 10^5$

$\gamma = \frac{1}{4} \quad v_i \equiv \lambda_{\max}(A_i^\top A_i) \leq 1$

Method: $|S_t| \equiv 1 \quad p_i = \frac{1}{n} \quad \lambda = \frac{1}{n}$

$$(1-\theta)^n = 0.8187$$

$$(1-\theta)^{12n} = 0.0907 < \frac{1}{10}$$

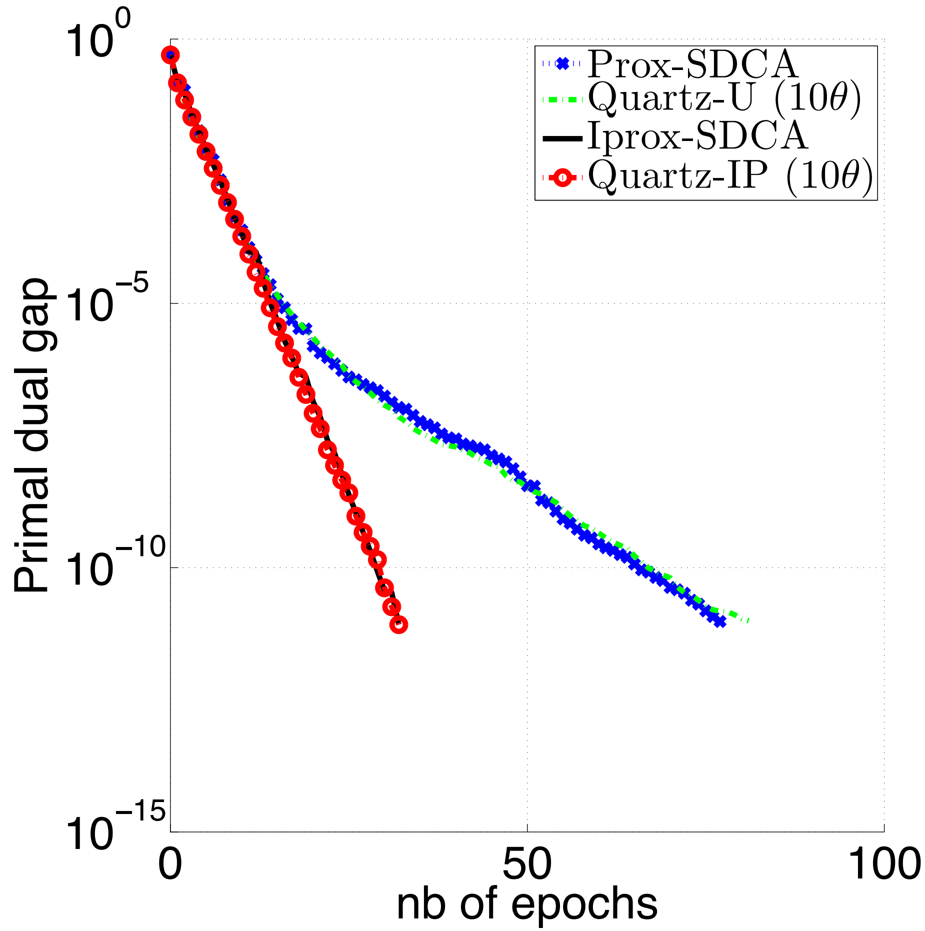
Updating One Dual Variable at a Time

Complexity of Quartz specialized to serial sampling

Optimal sampling	$n + \frac{\frac{1}{n}\sum_{i=1}^{n} L_i}{\lambda\gamma}$
Uniform sampling	$n + \frac{\max_i L_i}{\lambda\gamma}$

$$L_i \equiv \lambda_{\max}\left(A_i^\top A_i\right)$$

Experiment: Quartz vs SDCA, uniform vs optimal sampling



$$\text{Data} = \texttt{cov1}, \quad n = 522,911, \quad \lambda = 10^{-6}$$

An Efficient Primal Method

S. Shalev-Shwartz
SDCA without Duality, *NIPS 2015* (arXiv:1502.06177)

Dominik Csiba and P.R.
Primal method for ERM with flexible mini-batching schemes and non-convex losses, *arXiv:1506.02227,* 2015

Empirical Risk Minimization

Primal Problem: ERM

$\phi_i : \mathbb{R}^m \mapsto \mathbb{R}$

$\frac{1}{\gamma}$-smooth and convex

regularization parameter

$$\min_{w \in \mathbb{R}^d} \left[P(w) \equiv \frac{1}{n} \sum_{i=1}^{n} \phi_i(A_i^\top w) + \frac{\lambda}{2} \|w\|_2^2 \right]$$

d = # features (parameters)

n = # samples

$A_i \in \mathbb{R}^{d \times m}$

We had a general 1-strongly convex function *g* here before

Assumption

The loss functions $\phi_i : \mathbb{R}^m \mapsto \mathbb{R}$ are $\frac{1}{\gamma}$-smooth:

$$\|\nabla \phi_i(a) - \nabla \phi_i(a')\| \leq \frac{1}{\gamma} \|a - a'\|, \quad a, a' \in \mathbb{R}^m$$

Lipschitz constant of the gradient of the function

Dual Problem

$\in \mathbb{R}^m$

$$D(\alpha) \equiv -\lambda g^*\left(\frac{1}{\lambda n}\sum^{n} \ldots\right) - \frac{1}{\ldots}$$

1 – smooth
& convex

$$g^*(w') = \max_{w \in \mathbb{R}^d} \{(w')^\top w \ldots$$

$$\alpha = (\underbrace{\alpha_1}_{\in \mathbb{R}^m}, \underbrace{\ldots}_{\in \mathbb{R}^m}$$

Goal: An efficient algorithm which naturally operates in the primal space (i.e., on the primal problem) only

The method will have the "same" theoretical guarantee as Quartz

The computer lab will be based on this

6.2
The Algorithm

Motivation I

w^* is optimal

$$0 = \nabla P(w^*) = \left(\frac{1}{n} \sum_{i=1}^{n} A_i \nabla \phi_i (A_i^\top w^*) \right) + \lambda w^*$$

$$w^* = \frac{1}{\lambda n} \sum_{i=1}^{n} A_i \alpha_i^*$$

$$\alpha_i^* := -\nabla \phi_i (A_i^\top w^*)$$

Motivation II

Algorithmic Ideas:

1. Simultaneously search for both w^* and $\alpha_1^*, \ldots, \alpha_n^*$

2. Try to do "something like"

$$\alpha_i^{t+1} \leftarrow -\nabla \phi_i(A_i^\top w^t)$$

3. Maintain the relationship

$$w^t = \frac{1}{\lambda n} \sum_{i=1}^{n} A_i \alpha_i^t$$

Does not quite work: too "greedy"

The Algorithm: dfSDCA

Initialize the relationship

STEP 0: INITIALIZE

Choose $\alpha_1^0, \ldots, \alpha_n^0 \in \mathbb{R}^m$

$$w^0 = \frac{1}{\lambda n} \sum_{i=1}^{n} A_i \alpha_i^0$$

STEP 1: "DUAL" UPDATE

Choose a random set S_t of "dual variables"

For $i \in S_t$ do

Controlling "greed" by taking a convex combination

$$\theta = \min_i \frac{p_i \lambda \gamma n}{v_i + \lambda \gamma n}$$

$$\alpha_i^{t+1} \leftarrow \left(1 - \frac{\theta}{p_i}\right) \alpha_i^t + \frac{\theta}{p_i} \left(-\nabla \phi_i (A_i^\top w^t)\right)$$

$$p_i = \mathbf{P}(i \in S_t)$$

STEP 2: PRIMAL UPDATE

$$w^{t+1} \leftarrow w^t - \sum_{i \in S_t} \frac{\theta}{n \lambda p_i} A_i \left(\nabla \phi_i (A_i^\top w^t) + \alpha_i^t\right)$$

This is just maintaining the relationship

Complexity

ESO Assumption (same as before!)

Parameters $v_1, \ldots, v_n$ satisfy:

$$\mathbf{E} \left\| \sum_{i \in S_t} A_i \alpha_i \right\|^2 \leq \sum_{i=1}^{n} p_i v_i \|\alpha_i\|^2$$

inequality must hold for all

$\alpha_1, \ldots, \alpha_n \in \mathbb{R}^m$

$p_i = \mathbf{P}(i \in S_t)$

Complexity

A constant depending on

$P, w^0, \alpha_i^0, w^*, \alpha_i^*$

Theorem [Csiba & R '15]

$$t \geq \max_i \left(\frac{1}{p_i} + \frac{v_i}{p_i \lambda \gamma n} \right) \log \left(\frac{C}{\epsilon} \right)$$

$p_i = \mathbf{P}(i \in S_t)$

$$\mathbf{E}\left[P(w^t) - P(w^*)\right] \leq \epsilon$$

Experiments

Some More Efficient Primal Methods for ERM: SAG, SVRG and S2GD

SAG: Stochastic Average Gradient

N. Le Roux, M. Schmidt, and F. Bach. **A stochastic gradient method with an exponential convergence rate for finite training sets**. *NIPS*, 2012

SVRG: Stochastic Variance Reduced Gradient

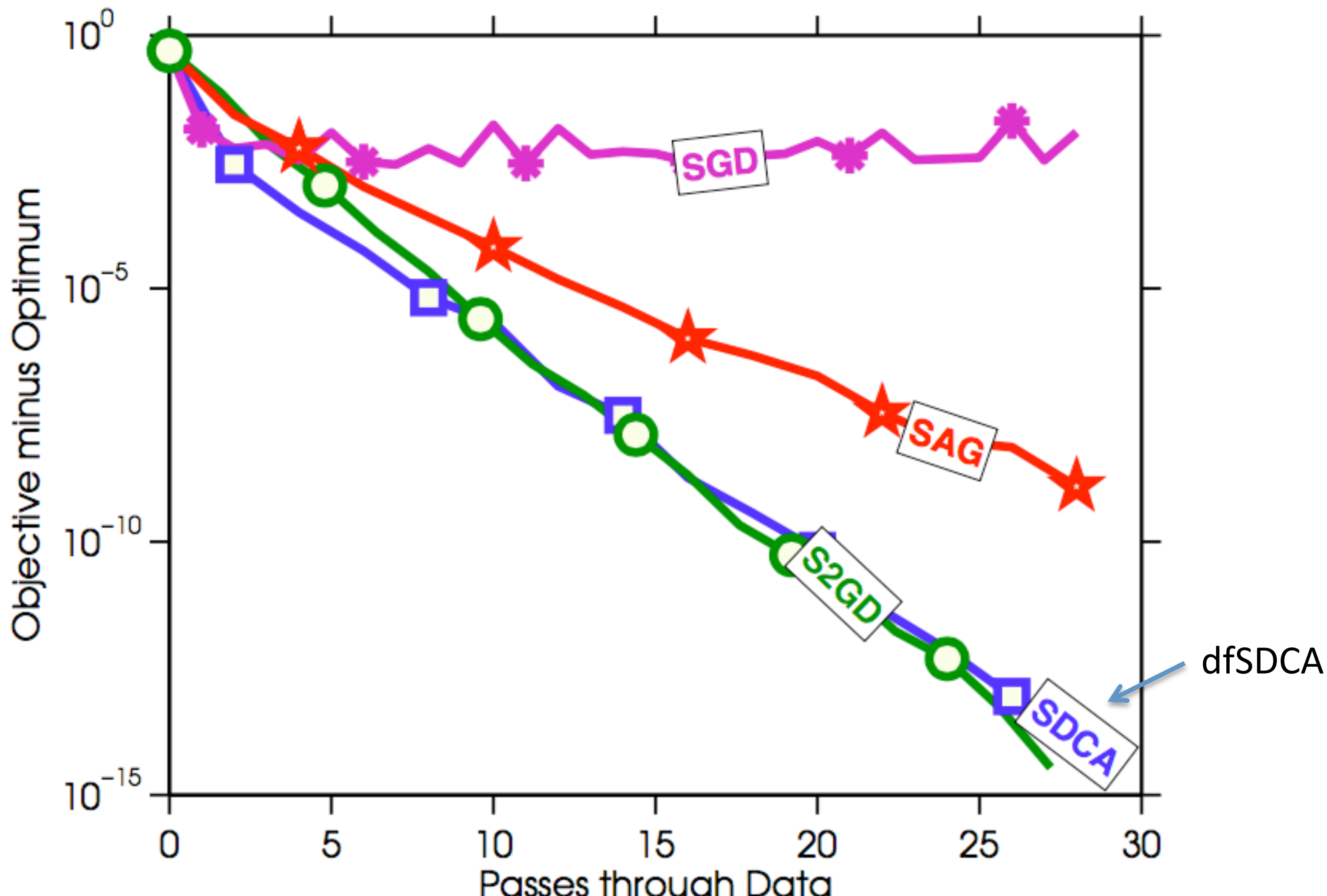
Rie Johnson and Tong Zhang. **Accelerating stochastic gradient descent using predictive variance reduction.** *NIPS*, 2013.

S2GD: Semi-Stochastic Gradient Descent

J. Konečný and P. R. **Semi-stochastic gradient descent methods.** *arXiv:1312.1666*, 2013

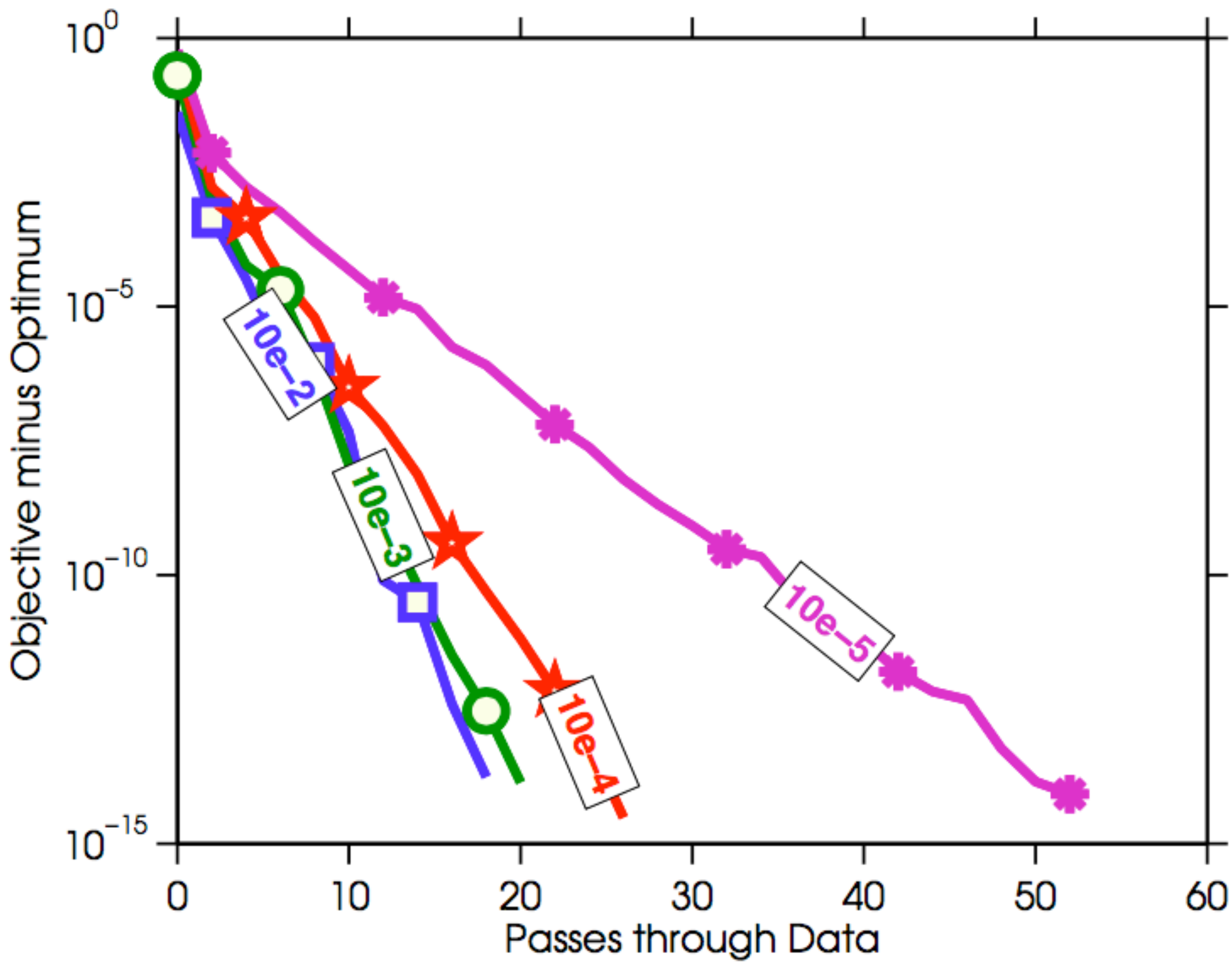
Modern Methods for ERM vs SGD

Dataset: rcv1 (n = 20,241 ; d = 47,232)



Behavior of dfSDCA for various λ

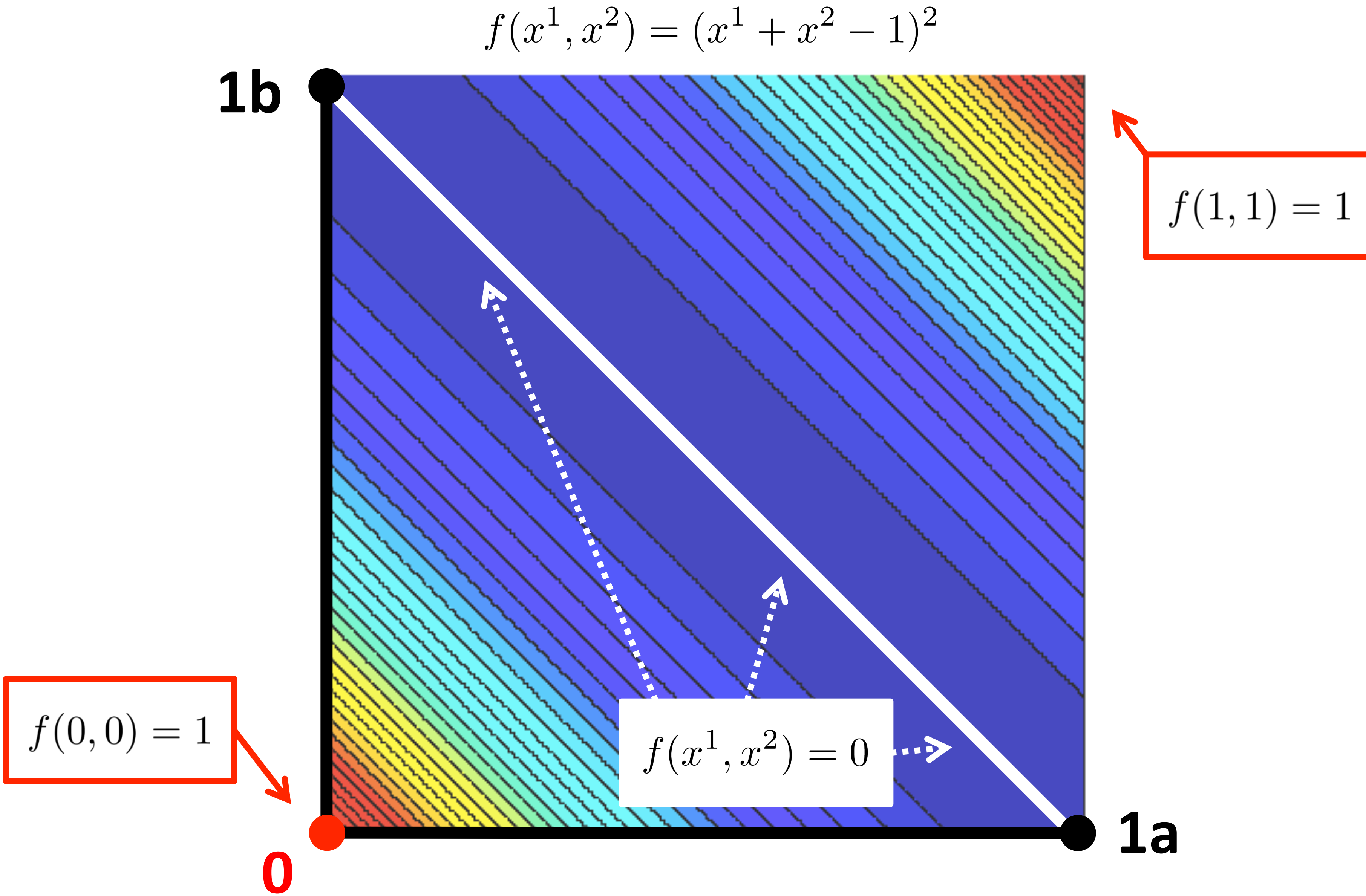
Dataset: rcv1 (n = 20,241 ; d = 47,232)



Parallelization (Minibatching)

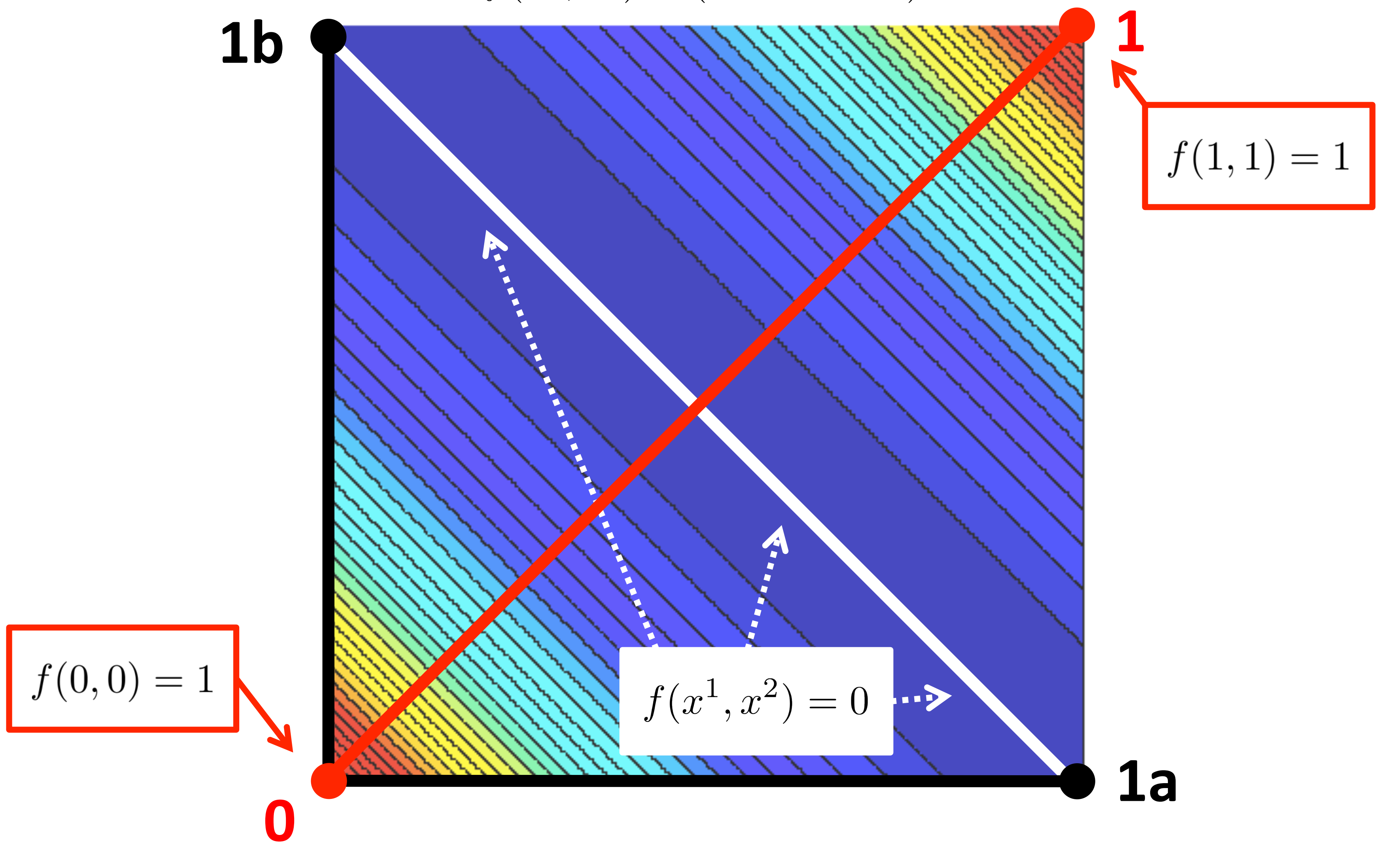
NAIVE APPROACH

Failure of naive parallelization

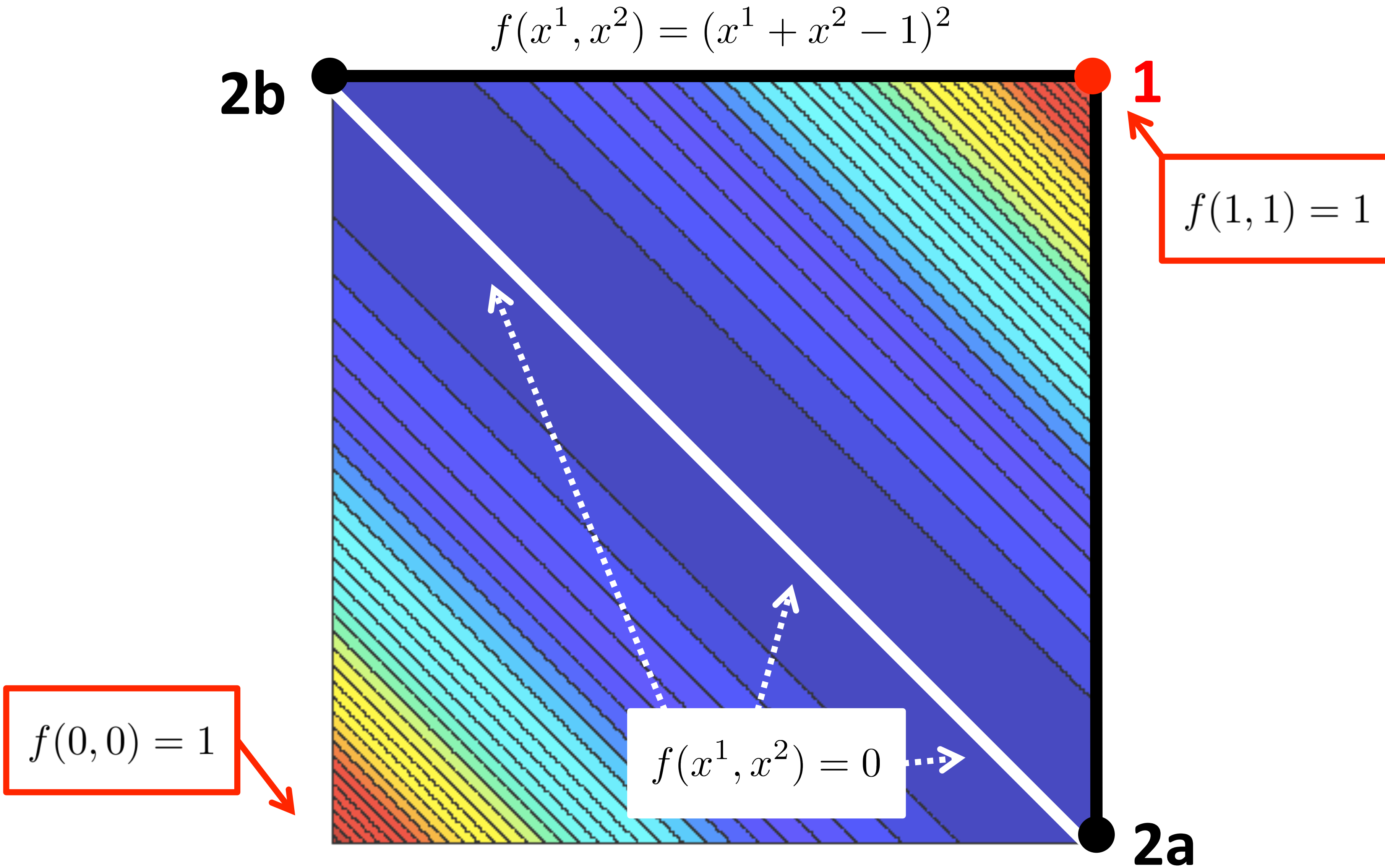


Failure of naive parallelization

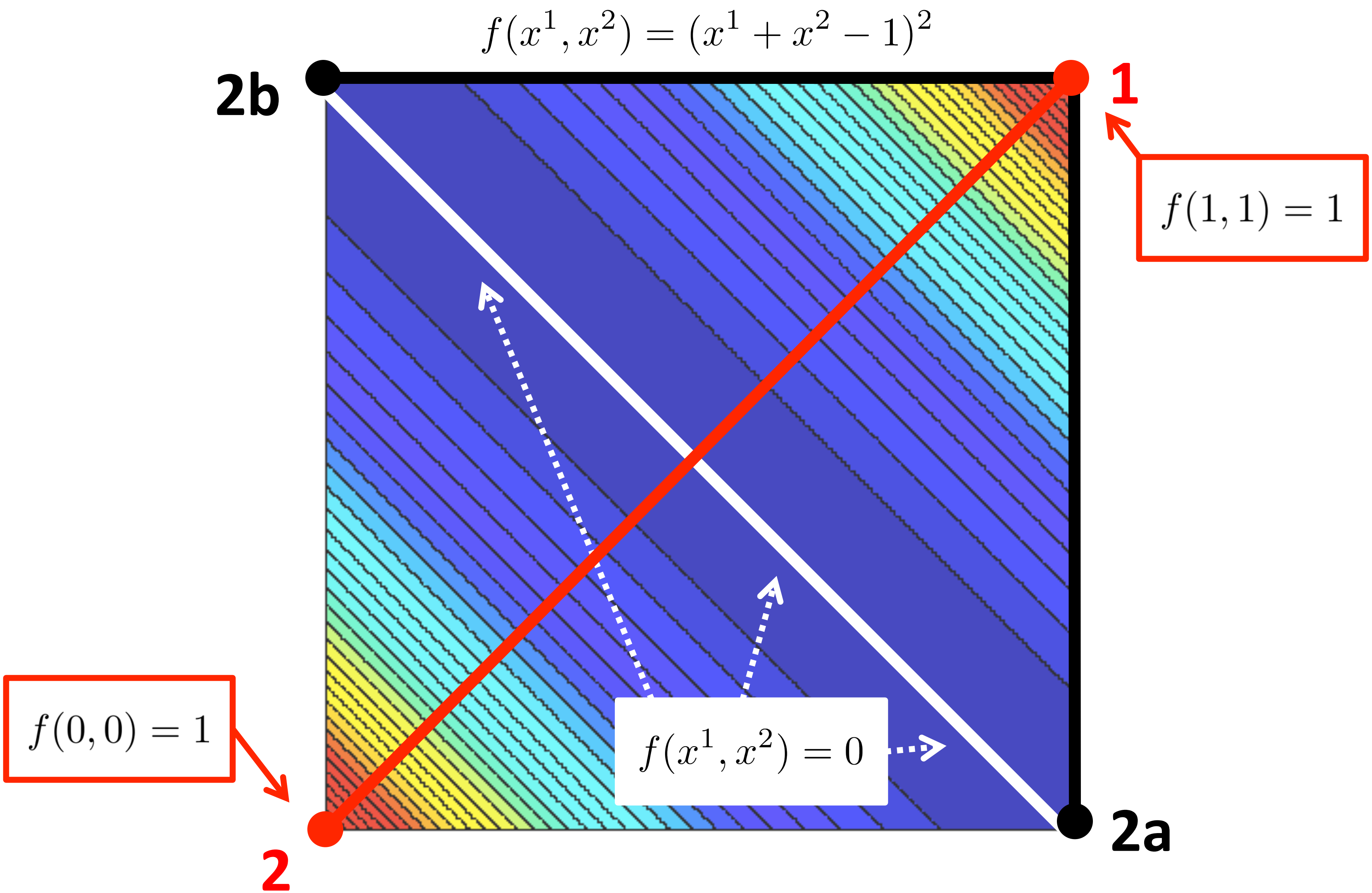
$$f(x^1, x^2) = (x^1 + x^2 - 1)^2$$



Failure of naive parallelization

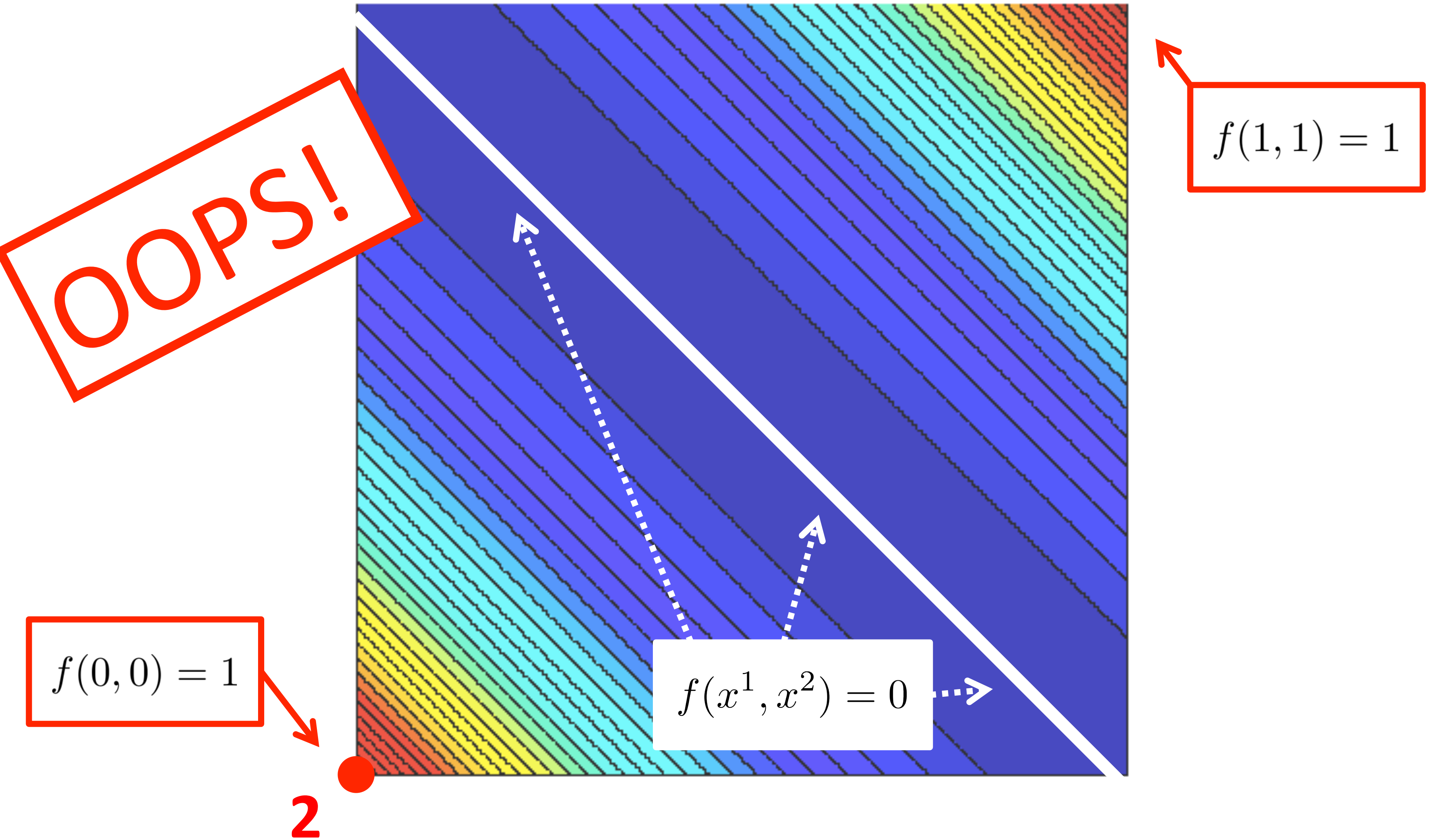


Failure of naive parallelization

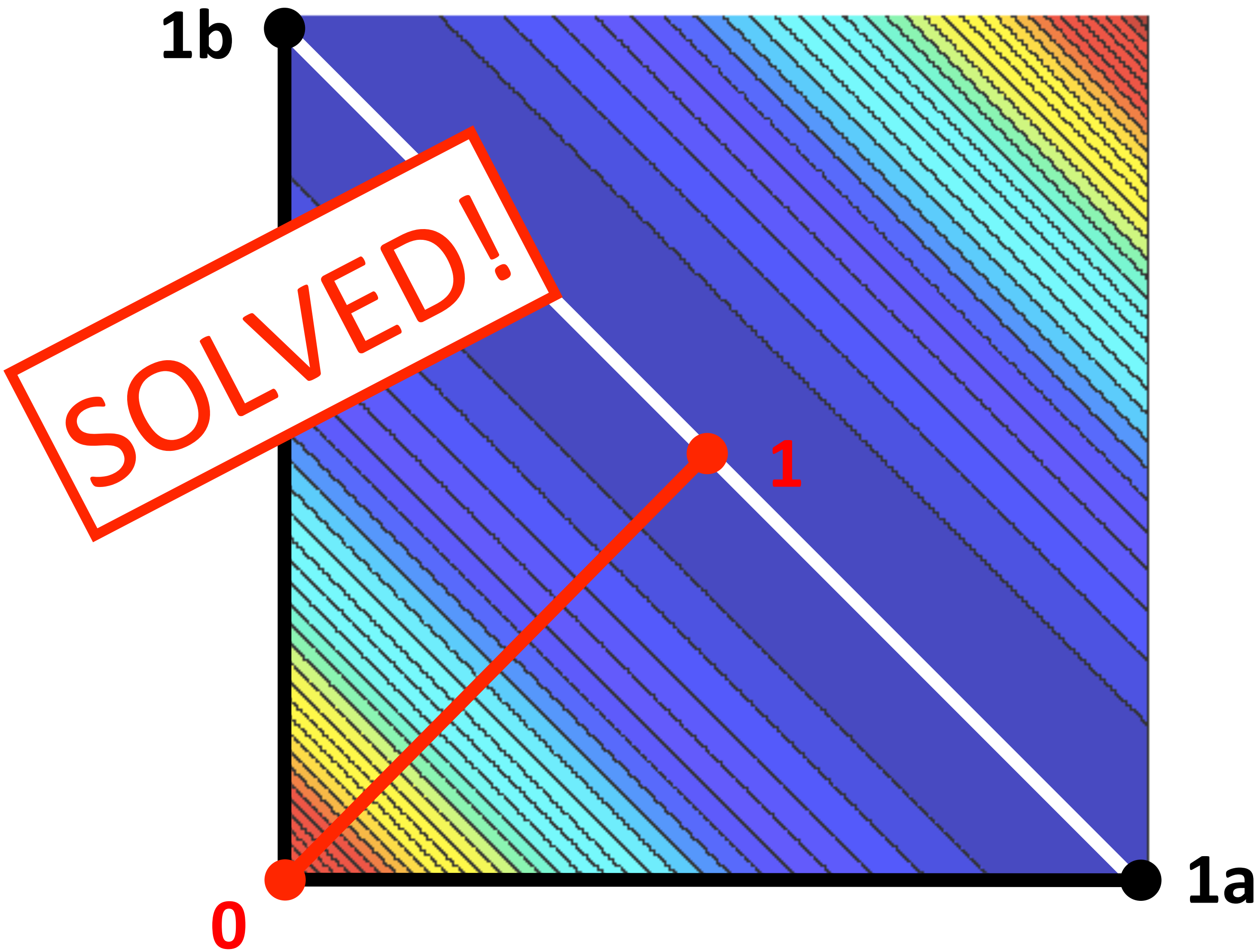


Failure of naive parallelization

$$f(x^1, x^2) = (x^1 + x^2 - 1)^2$$

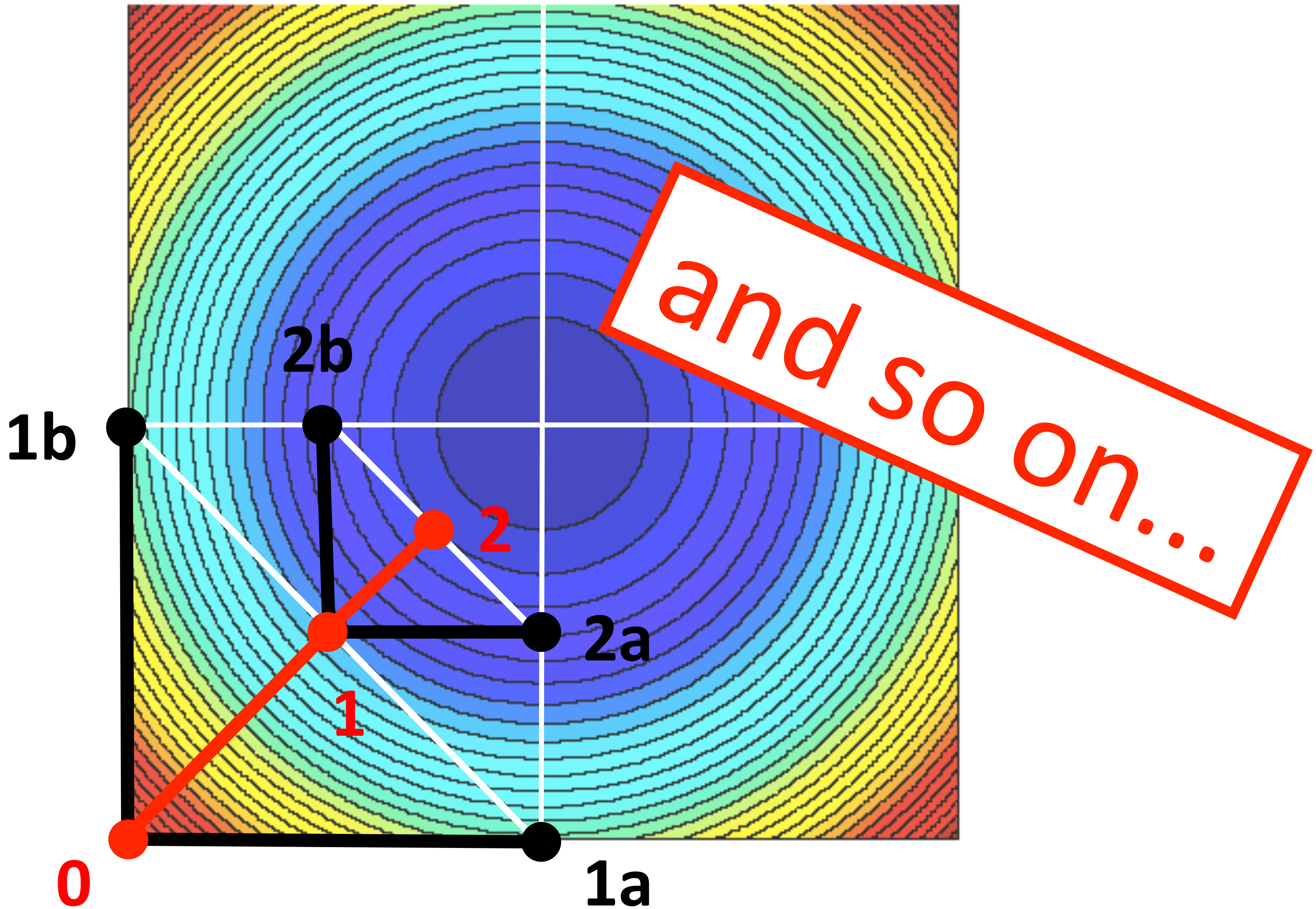


Idea: averaging updates may help



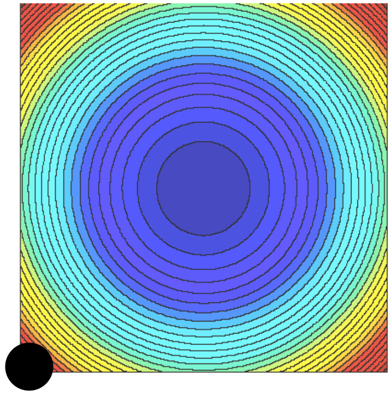
Averaging can be too conservative

$$f(x^1, x^2) = (x^1 - 1)^2 + (x^2 - 1)^2$$



Averaging can be too conservative

$$f(x) = (x^1 - 1)^2 + (x^2 - 1)^2 + \cdots + (x^n - 1)^2$$



$x_0 = 0 \quad f(x_0) = n$

BAD!!!

$$\boxed{k \geq \frac{n}{2} \log\left(\frac{n}{\epsilon}\right)}$$

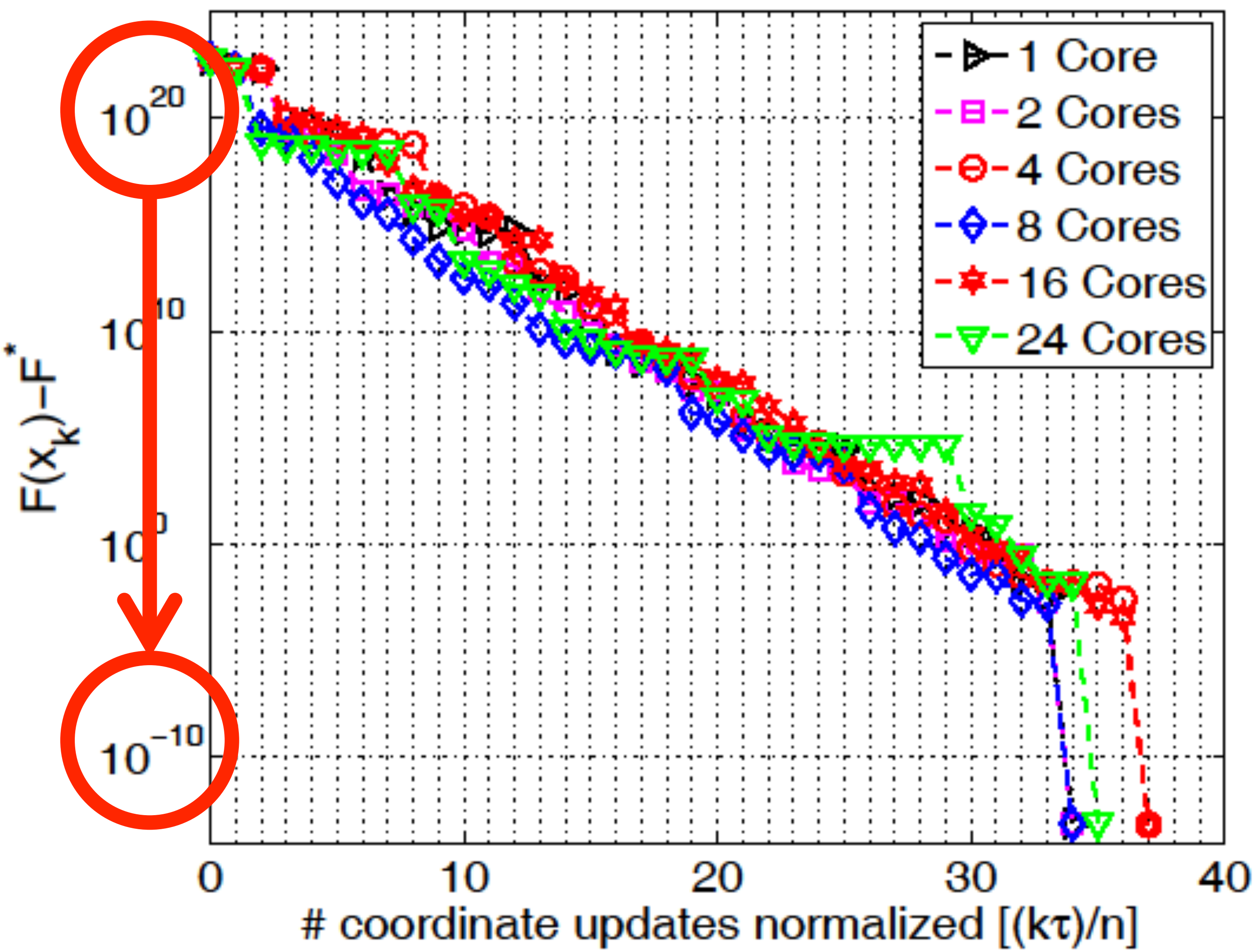
$$f(x_k) = n\left(1 - \frac{1}{n}\right)^{2k} \leq \epsilon$$

WANT

Experiment with a 1 billion-by-2 billion LASSO problem

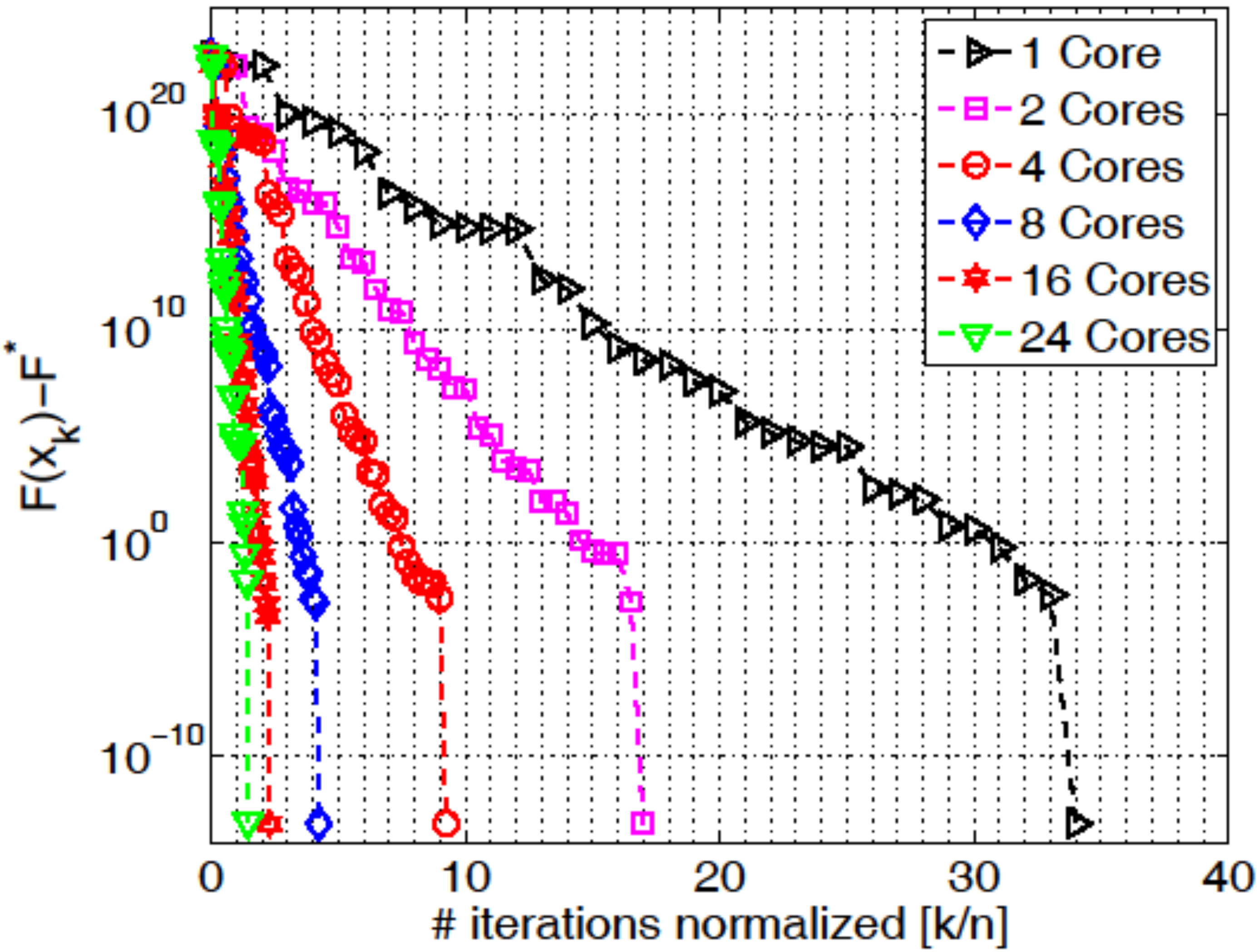
P.R. and Martin Takáč
Parallel coordinate descent methods for big data optimization
Mathematical Programming, 2015 (arXiv:1212.0873)

Coordinate Updates



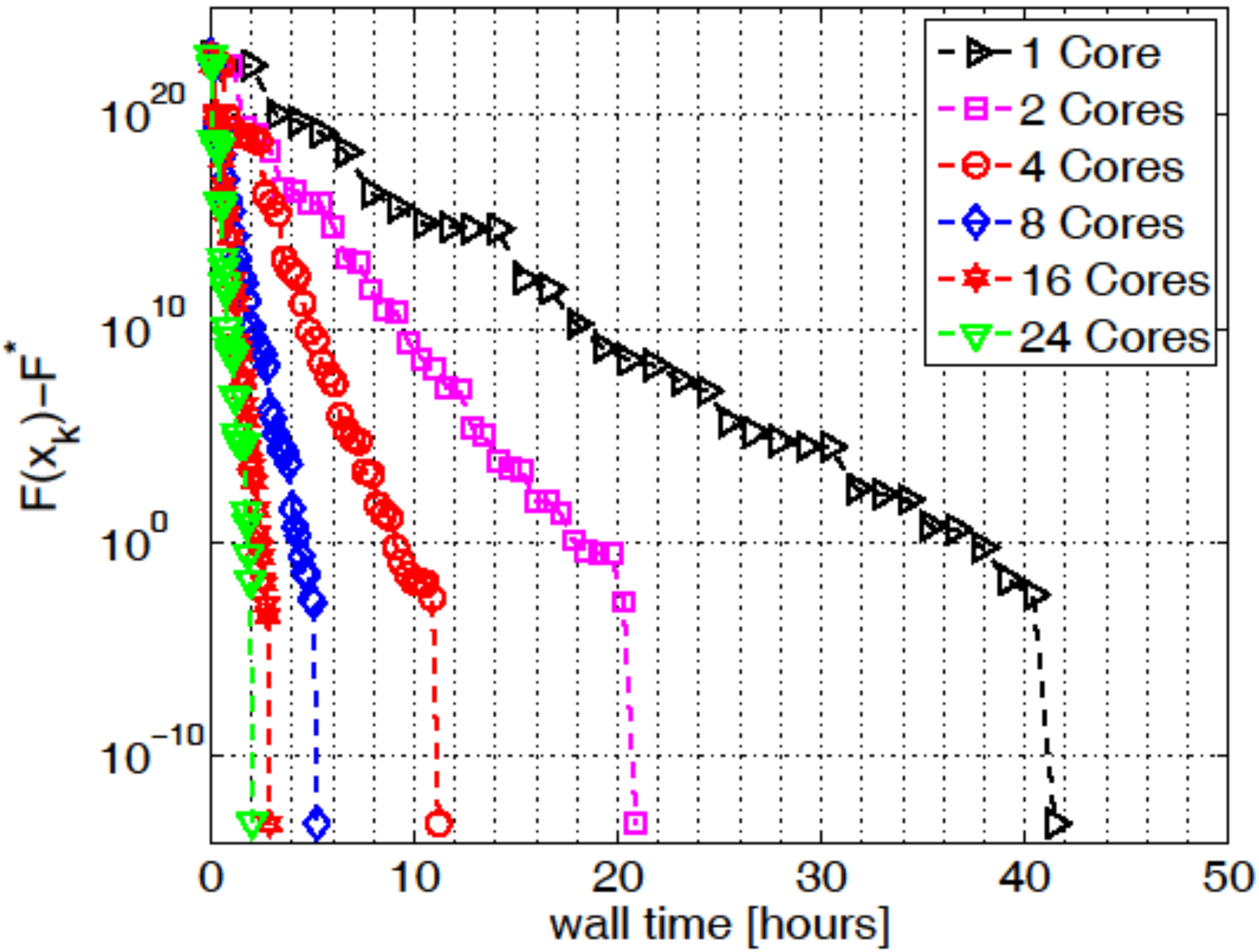
LASSO problem with $A \in \mathbb{R}^{m \times n}$, where $\boxed{n = 10^9 \text{ and } m = 2 \times 10^9}$

Iterations



LASSO problem with $A \in \mathbb{R}^{m \times n}$, where $n = 10^9$ and $m = 2 \times 10^9$

Wall Time



LASSO problem with $A \in \mathbb{R}^{m \times n}$, where $n = 10^9$ and $m = 2 \times 10^9$

Minibatching & Quartz

[Qu, R & Zhang 14]

Data Sparsity

$$1 \leq \tilde{\omega} \leq n$$

"Fully sparse data"

A normalized measure of average sparsity of the data

"Fully dense data"

Quartz [Qu, R & Zhang 14]

Complexity of Quartz

Fully sparse data ($\tilde{\omega} = 1$)	$\frac{n}{\tau} + \frac{\max_i L_i}{\lambda\gamma\tau}$
Fully dense data ($\tilde{\omega} = n$)	$\frac{n}{\tau} + \frac{\max_i L_i}{\lambda\gamma}$
Any data ($1 \leq \tilde{\omega} \leq n$)	$\frac{n}{\tau} + \frac{\left(1 + \frac{(\tilde{\omega}-1)(\tau-1)}{n-1}\right)\max_i L_i}{\lambda\gamma\tau}$

$\equiv T(\tau)$

Speedup

Assume the data is normalized: $L_i \equiv \lambda_{\max}(A_i^\top A_i) \leq 1$

Then:

$$T(\tau) \;=\; \frac{\left(1 + \frac{(\tilde{\omega}-1)(\tau-1)}{(n-1)(1+\lambda\gamma n)}\right)}{\tau} \times T(1)$$

Linear speedup up to a certain data-independent minibatch size:

$$\tau \leq 2 + \lambda\gamma n \quad \Rightarrow \quad T(\tau) \leq \frac{2}{\tau} \times T(1)$$

Further data-dependent speedup, up to the extreme case:

$$\tilde{\omega} = \mathcal{O}(\lambda\gamma n) \quad \Rightarrow \quad T(\tau) = \mathcal{O}\left(\frac{T(1)}{\tau}\right)$$

Quartz [Qu, R & Zhang 14]

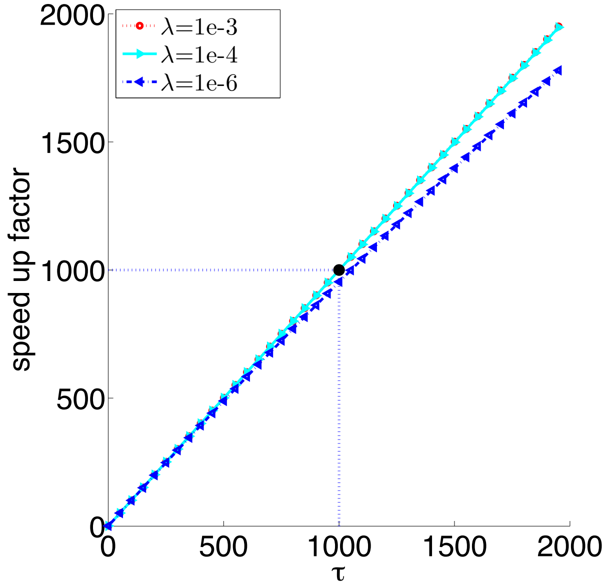
Quartz: Parallelization Speedup

examples: $n = 10^6$

Smoothness of loss functions: $\gamma = 1$

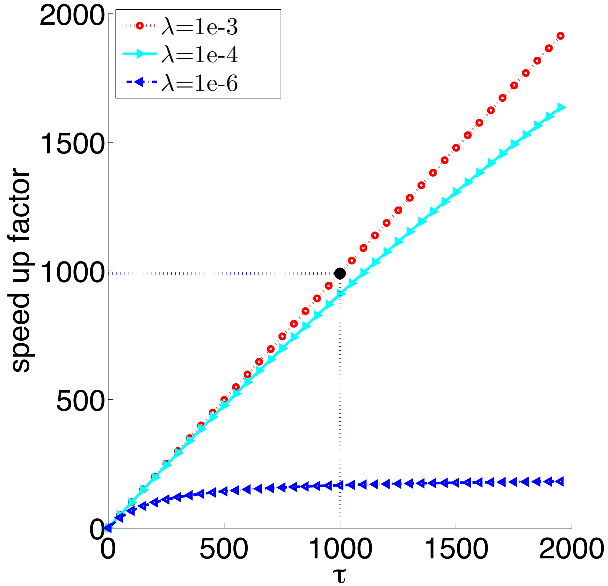
Low regularization: $\lambda = 1/n$

High regularization: $\lambda = 1/\sqrt{n}$



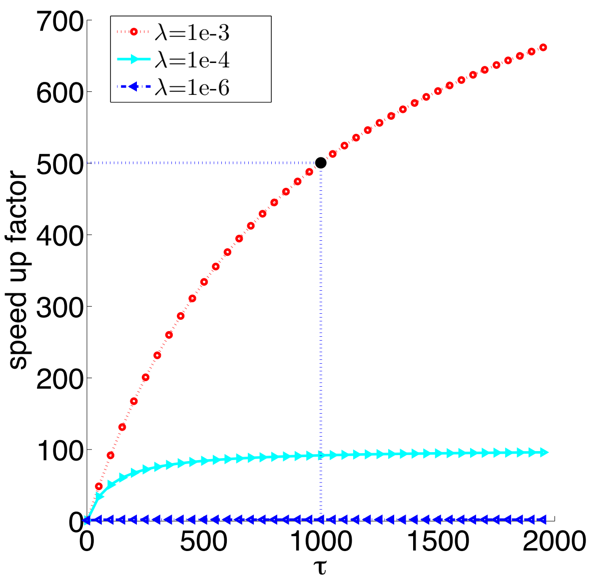
Sparse Data

$\tilde{\omega} = 10^2$



Denser Data

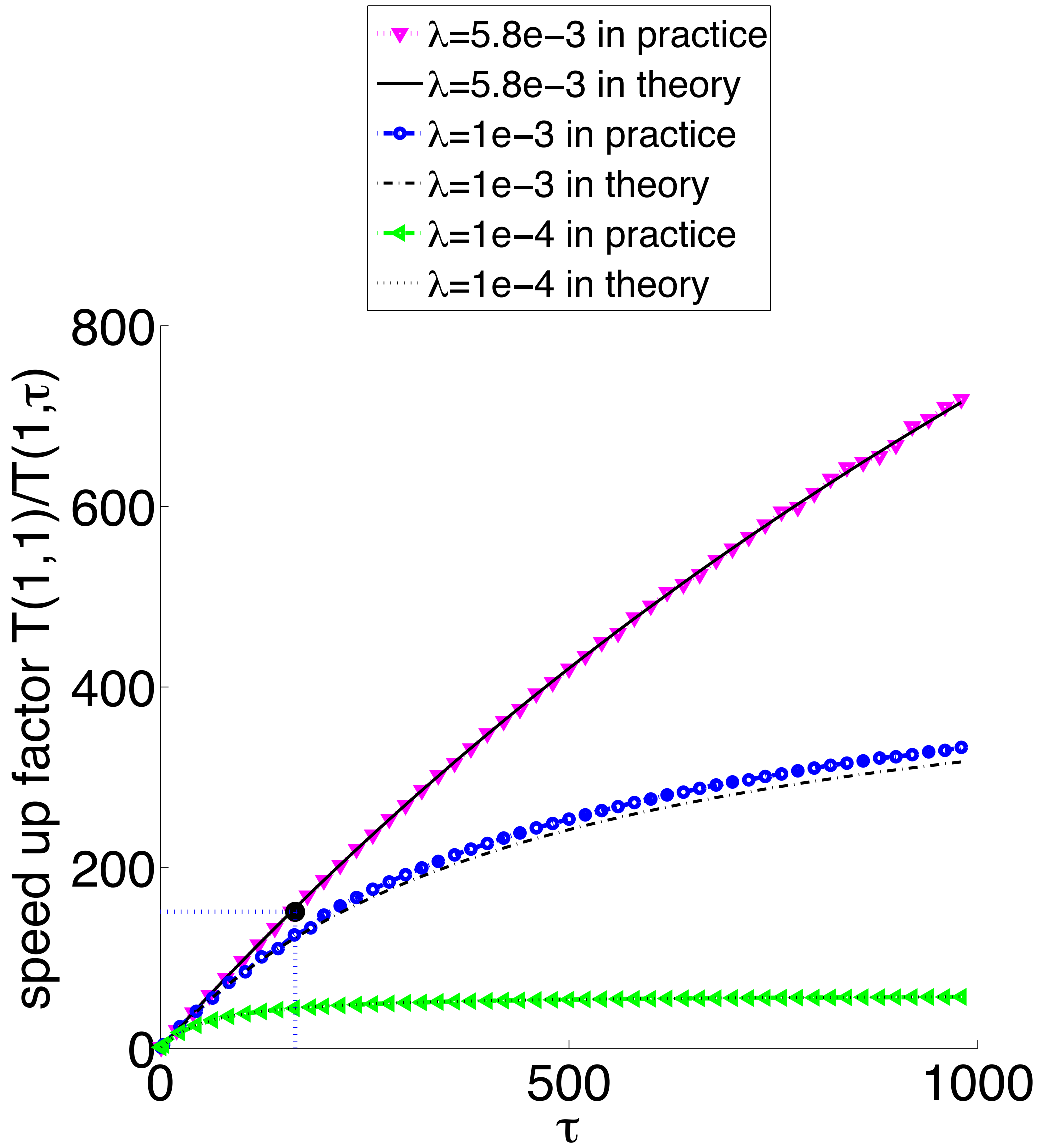
$\tilde{\omega} = 10^4$



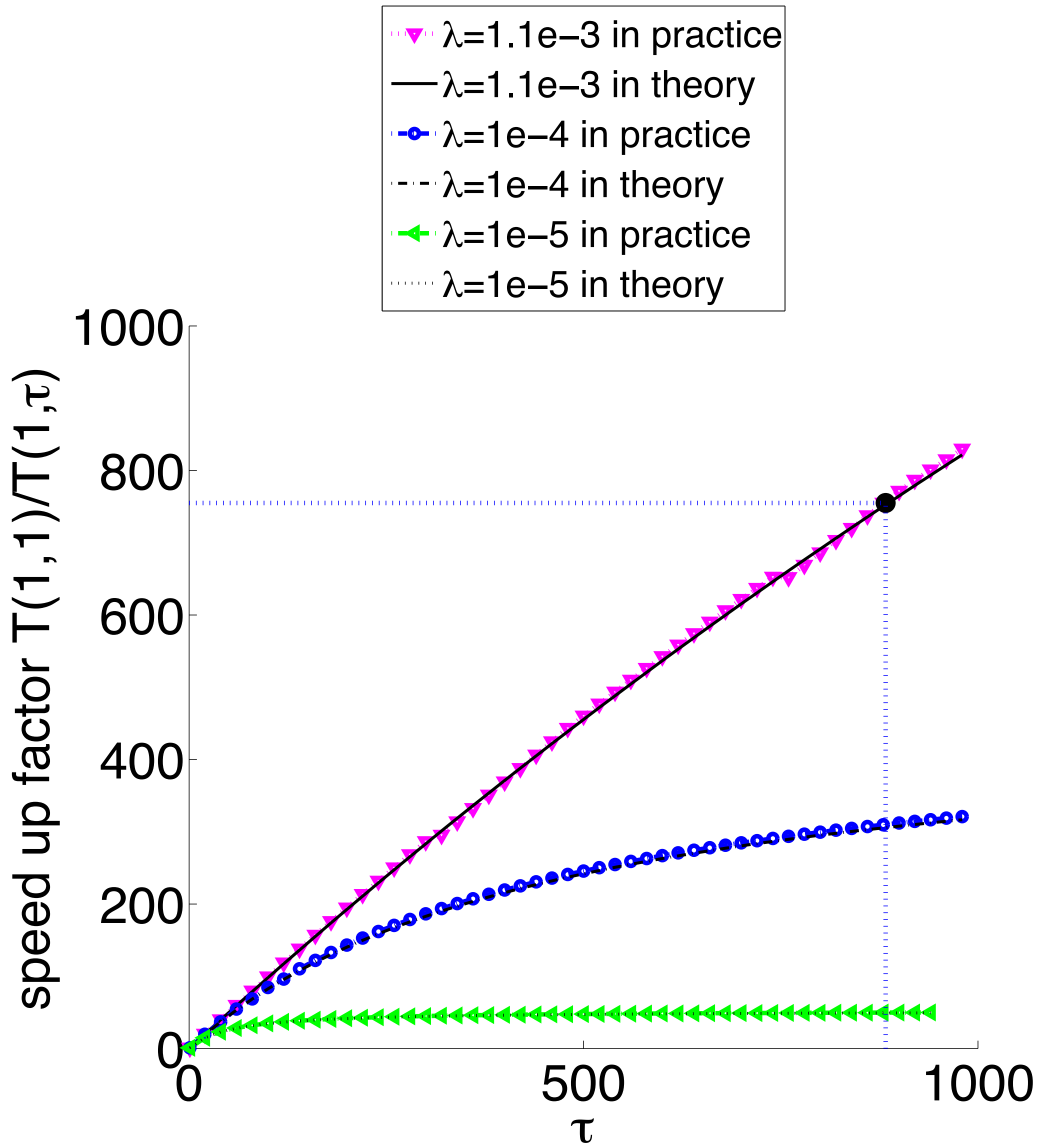
Fully Dense Data

$\tilde{\omega} = 10^6$

astro_ph: *n* = 29,882   density = 0.08%



CCAT: n = 781,265 density = 0.16%



Primal-dual methods with tau-nice sampling

$L_i = 1$

Algorithm	Iteration complexity	g
SDCA [S-Shwartz & Zhang 12]	$n + \frac{1}{\lambda\gamma}$	$\frac{1}{2}\|\cdot\|^2$
ASDCA [S-Shwartz & Zhang 13a]	$4 \times \max\left\{\frac{n}{\tau}, \sqrt{\frac{n}{\lambda\gamma\tau}}, \frac{1}{\lambda\gamma\tau}, \frac{n^{\frac{1}{3}}}{(\lambda\gamma\tau)^{\frac{2}{3}}}\right\}$	$\frac{1}{2}\|\cdot\|^2$
SPDC [Zhang & Xiao 14]	$\frac{n}{\tau} + \sqrt{\frac{n}{\lambda\gamma\tau}}$	general
Quartz	$\frac{n}{\tau} + \left(1 + \frac{(\tilde{\omega}-1)(\tau-1)}{n-1}\right)\frac{1}{\lambda\gamma\tau}$	general

For sufficiently sparse data, Quartz wins even when compared against accelerated methods

	Algorithm	$\gamma\lambda n = \Theta(\frac{1}{\tau})$	$\gamma\lambda n = \Theta(1)$	$\gamma\lambda n = \Theta(\tau)$	$\gamma\lambda n = \Theta(\sqrt{n})$
		$\kappa = n\tau$	$\kappa = n$	$\kappa = n/\tau$	$\kappa = \sqrt{n}$
	SDCA	$n\tau$	n	n	n
Accelerated	ASDCA	n	$\frac{n}{\sqrt{\tau}}$	$\frac{n}{\tau}$	$\frac{n}{\tau} + \frac{n^{3/4}}{\sqrt{\tau}}$
Accelerated	SPDC	n	$\frac{n}{\sqrt{\tau}}$	$\frac{n}{\tau}$	$\frac{n}{\tau} + \frac{n^{3/4}}{\sqrt{\tau}}$
	Quartz	$n + \tilde{\omega}\tau$	$\frac{n}{\tau} + \tilde{\omega}$	$\frac{n}{\tau}$	$\frac{n}{\tau} + \frac{\tilde{\omega}}{\sqrt{n}}$

THE END