Matrix-Free Interior Point Method for Large-Scale Optimization

Jacek Gondzio

Outline

- Interior Point Methods: Pros & Cons
- Accelerating IPMs
- *Exact* vs *Inexact* Newton Method and IPMs
  → worst-case complexity results
- Inexact Newton → Krylov subspace methods
- Preconditioner is a must
- Computational results
  - Compressed Sensing
  - Google Problem
- Conclusions

Motivation

- First-order methods
  - complexity $\mathcal{O}(1/\varepsilon)$ or $\mathcal{O}(1/\varepsilon^2)$
  - produce a rough approx. of solution quickly
  - but ... struggle to converge to high accuracy

- IPMs are second-order methods
  (they apply Newton method to barrier subprobs)
  - complexity $\mathcal{O}(\log(1/\varepsilon))$
  - produce accurate solution in a few iterations
  - but ... one iteration may be expensive

Just think

For example, $\varepsilon = 10^{-3}$ gives
$1/\varepsilon = 10^3$ and $1/\varepsilon^2 = 10^6$, but $\log(1/\varepsilon) \approx 10$.

For example, $\varepsilon = 10^{-6}$ gives
$1/\varepsilon = 10^6$ and $1/\varepsilon^2 = 10^{12}$, but $\log(1/\varepsilon) \approx 20$.

LP & QP Problems

$$\begin{aligned} \min \quad & c^T x + \tfrac{1}{2} x^T Q\, x \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0, \end{aligned}$$

where $A \in \mathcal{R}^{m \times n}$ has full row rank
and $Q \in \mathcal{R}^{n \times n}$ is symmetric positive semidefinite.

We expect m and n to be large.

Applications: LPs, QPs constructed implicitly

- problems generated by the algebraic mod. language
- problems too large to be stored
  (but generated by some "simple" process)
- LP relaxations of combinatorial (integer) problems
- sparse approximations (compressed sensing)
- PageRank (Google problem)

Assumption: A and Q as *"operators"* $A \cdot u$, $A^T \cdot v$, $Q \cdot u$

Expectation: Low complexity of these operations

Standard Interior Point Method

The First Order Optimality Conditions

$$\begin{aligned} Ax &= b, \\ -Qx + A^T y + s &= c, \\ XSe &= \mu e, \\ (x, s) &> 0. \end{aligned}$$

Assume primal-dual feasibility:

$$Ax = b \qquad \text{and} \qquad -Qx + A^T y + s = c$$

Apply Newton Method to the FOC

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^T & I \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} b - Ax \\ c - A^T y - s + Qx \\ \sigma \mu e - XSe \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \xi \end{bmatrix}.$$

Central Path:

A set of all solutions to the optimality conds for $\mu > 0$.

Path Following Method:

Stay in the **neighbourhood** (of the central path)

$$\mathcal{N}_2(\theta) := \{(x, y, s) \in \mathcal{F}^0 : \|XSe - \mu e\|_2 \leq \theta\mu\}$$

$$\mathcal{N}_S(\gamma) := \{(x, y, s) \in \mathcal{F}^0 : \gamma\mu \leq x_i s_i \leq (1/\gamma)\mu\}$$

where

$\mathcal{F}^0 := \{(x, y, s) : c - A^T y - s + Qx = 0, Ax = b, x, s > 0\}.$

Standard complexity result

Theorem (Wright, Thm 5.12).

Let $\epsilon > 0$ be the required accuracy of the optimal solution. The (*short-step, feasible*) interior point method finds the ϵ-accurate solution such that

$$\mu^k \leq \epsilon$$

after at most

$$K = \mathcal{O}(\sqrt{n} \ln(1/\epsilon))$$

iterations.

Proof (technical trick)

Work with the third equation in the Newton system

$$S\Delta x + X\Delta s = \sigma\mu e - XSe = \xi,$$

to get $$s^T\Delta x + x^T\Delta s = \sigma\mu e^T e - x^T s = (\sigma - 1)x^T s.$$

Hence at the new point $(\bar{x}, \bar{y}, \bar{s}) = (x, y, s) + (\Delta x, \Delta y, \Delta s)$ the complementarity gap becomes

$$\begin{aligned} \bar{x}^T\bar{s} &= (x + \Delta x)^T(s + \Delta s) \\ &= x^T s + (s^T\Delta x + x^T\Delta s) + \Delta x^T\Delta s \\ &= \sigma x^T s + \Delta x^T\Delta s. \end{aligned}$$

Make the **error** $\Delta x^T\Delta s$ and $\|\Delta X\Delta Se\|$ small!

Interior Point Methods

Theory: convergence in $\mathcal{O}(\sqrt{n})$ or $\mathcal{O}(n)$ iterations
Practice: convergence in $\mathcal{O}(\log n)$ iterations

Expected number of IPM iterations:

Problem Dimension	LP	QP
1,000	5 - 10	5 - 10
10,000	10 - 20	10 - 15
100,000	15 - 30	10 - 15
1,000,000	20 - 35	15 - 20
10,000,000	25 - 40	15 - 20
100,000,000	30 - 45	20 - 25
1000,000,000	35 - 50	20 - 25

... but one iteration may be expensive!

Make IPMs better

- Find an ϵ-accurate solution in
$$\mathcal{O}(\log n \, \ln(1/\epsilon))$$
iterations (in practice).

- Lower the cost of a single IPM iteration
from $\mathcal{O}\big(n^3\big)$ to $\mathcal{O}\big(n\big)$.
Realistically: make only a few matrix-vector prods.

Use **Matrix-Free Interior Point Method**

Redesign Interior Point Methods

- make a single iteration as fast as possible
  replace *Exact* Newton Method
  with *Inexact* Newton Method
- work in matrix-free regime
- work with limited memory

Use

Inexact Newton Method

Dembo, Eisenstat & Steihaug,
SIAM J. on Num Analysis 19 (1982) 400–408.

Exact Newton Method

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^T & I \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \xi \end{bmatrix}.$$

Inexact Newton Method

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^T & I \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \xi + \mathbf{r} \end{bmatrix}$$

allows for an error in the (linearized) complementarity condition only.

What happens to the complexity result?

General Assumption

The residual r in the inexact Newton Method satisfies:

$$\|r\| \leq \delta \|\xi\|,$$

where $\delta \in (0, 1]$.

What is an acceptable δ ?

Four parameters

- δ: relative *error* in the inexact Newton Method

$$\|r\| \leq \delta \|\xi\|,$$

- σ: *aspired* reduction of duality gap

$$\bar{\mu} = \sigma \mu,$$

- η, ω: *achieved* reduction of duality gap

$$\bar{\mu} = (1 - \frac{\eta}{n^{\omega}}) \mu.$$

Fit into a general scheme

Theorem (Wright, Thm 3.2).
Let $\epsilon > 0$ be the required accuracy of the optimal solution. Suppose the algorithm generates the sequence of iterates that satisfies:

$$\mu^{k+1} = (1 - \frac{\eta}{n^{\omega}}) \mu^k$$

and starts with $\mu^0 \leq 1/\epsilon^{\kappa}$. Then there exists an index

$$K = \mathcal{O}(n^{\omega} \ln(1/\epsilon))$$

such that

$$\mu^k \leq \epsilon, \quad \forall k \geq K.$$

Short-step (Feasible) Algorithm

Stay in the **small** neighbourhood of the central path

$$\mathcal{N}_2(\theta) := \{(x, y, s) \in \mathcal{F}^0 : \|XSe - \mu e\|_2 \leq \theta\mu\}$$

Use $\sigma = (1 - \frac{0.1}{\sqrt{n}})$.

Set $\delta = 0.2$ to achieve the reduction:

$$\bar{\mu} = (1 - \frac{0.01}{\sqrt{n}})\mu$$

hence $\eta = 0.01$ and $\omega = 1/2$.

⇒ Convergence in $\mathcal{O}(\sqrt{n} \ln(1/\epsilon))$ iterations.

Long-step (Feasible) Algorithm

Stay in the **large** neighbourhood of the central path

$$\mathcal{N}_S(\gamma) := \{(x,y,s) \in \mathcal{F}^0 : \gamma\mu \leq x_i s_i \leq (1/\gamma)\mu\}$$

Use $\sigma = 0.5$.
Set $\delta = \frac{1}{16}$ to achieve the reduction:

$$\bar{\mu} = (1 - \frac{0.01}{n})\mu$$

hence $\eta = 0.01$ and $\omega = 1$.

⇒ Convergence in $\mathcal{O}(n \ln(1/\epsilon))$ iterations.

Theorem

Suppose the algorithm uses the **inexact** Newton Method.

- If $(x, y, s) \in \mathcal{N}_2(\theta)$ and $\sigma = (1 - \frac{0.1}{\sqrt{n}})$, $\delta = 0.2$ then the algorithm converges in at most

$$K = \mathcal{O}(\sqrt{n} \ln(1/\epsilon))$$

  iterations.

- If $(x, y, s) \in \mathcal{N}_S(\gamma)$ and $\sigma = 0.5$, $\delta = \frac{1}{16}$ then the algorithm converges in at most

$$K = \mathcal{O}(n \ln(1/\epsilon))$$

  iterations.

Proof (key ideas)

For the Short-step Algorithm, show that the *error*

$$\|\Delta X \Delta S e\| = \mathcal{O}(\mu).$$

Use the *full* Newton step.
The proof requires 3 pages of maths.

For the Long-step Algorithm, show that the *error*

$$\|\Delta X \Delta S e\| = \mathcal{O}(n\mu).$$

Use the *damped* Newton step with $\alpha = \mathcal{O}(1/n)$.
The proof requires 5 pages of maths.

J.G.,
Convergence Analysis of Inexact Interior Point Method,
(in preparation).

Conclusion

Replace the **Exact** Newton Method
with the **Inexact** Newton Method

Allow for large residual

$$\|r\| \leq \delta \|\xi\|$$

The worst-case complexity result remains the same!

From Theory to Practice

Linear Algebra

First-order optimality conditions → Newton method →

Augmented System

$$\begin{bmatrix} Q+\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ d \end{bmatrix}$$

Normal Equations

$$(A(Q+\Theta^{-1})^{-1}A^T)\Delta y = g$$

Complementarity in IPMs:

$$x_j \cdot s_j = \mu \rightarrow 0 \qquad \forall j = 1, 2, ..., n.$$

Ill-conditioned scaling matrix $\Theta = XS^{-1}$.

For *"basic"* variables: $\Theta_j = x_j/s_j \rightarrow \infty \qquad \Theta_j^{-1} \rightarrow 0$;

For *"non-basic"* variables: $\Theta_j = x_j/s_j \rightarrow 0 \qquad \Theta_j^{-1} \rightarrow \infty$.

We want to use iterative (Krylov subspace) methods hence

Preconditioner is needed

Challenge:

- It must work in a **matrix-free** regime.
- It must work with a **limited-memory**.

Examples

- General LPs and QPs (**difficult**)
- Compressed Sensing
- Google Problem

Augmented System Matrix

Original: $$\mathcal{H} = \begin{bmatrix} -Q - \Theta^{-1} & A^T \\ A & 0 \end{bmatrix}$$

and *regularized*: $$\mathcal{H}_R = \begin{bmatrix} -(Q + \Theta^{-1} + R_p) & A^T \\ A & R_d \end{bmatrix}.$$

Normal Equation Matrix

Original: $$\mathcal{G} = (A(Q + \Theta^{-1})^{-1} A^T)$$

and *regularized*: $$\mathcal{G}_R = (A(Q + \Theta^{-1} + R_p)^{-1} A^T + R_d).$$

Altman & G., *OMS* 11-12 (1999) 275-302.

Decompose the regularized NE system

Use complete pivoting to compute

$$\mathcal{G}_R = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix} \begin{bmatrix} D_L & \\ & S \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ & I \end{bmatrix},$$

where $L = \begin{bmatrix} L_{11} \\ L_{21} \end{bmatrix}$ is a trapezoidal matrix:
(the first k columns of Cholesky factor of $\mathcal{G}_R$);
$S \in \mathcal{R}^{(m-k)\times(m-k)}$ is the corresp. **Schur complement**.

Order diagonal elements of D_L and $D_S = diag(S)$:

$$\underbrace{d_1 \geq d_2 \geq \cdots \geq d_k}_{D_L} \;\geq\; \underbrace{d_{k+1} \geq d_{k+2} \geq \cdots \geq d_m}_{D_S}.$$

Preconditioner

Use the decomposition

$$\mathcal{G}_R = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix} \begin{bmatrix} D_L & \\ & S \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ & I \end{bmatrix}$$

and precondition $\mathcal{G}_R$ with

$$P = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix} \begin{bmatrix} D_L & \\ & D_S \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ & I \end{bmatrix},$$

where D_S is a diagonal of S.

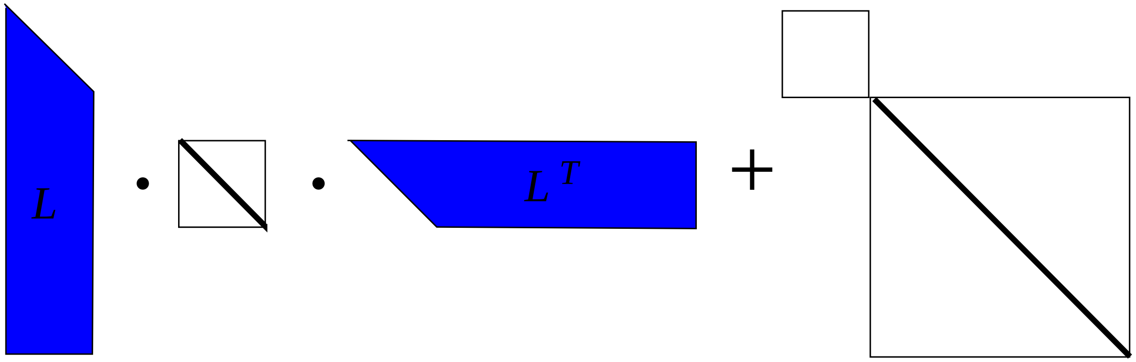
Do **not** compute S.
Update only its diagonal.

Preconditioner

Partial Cholesky of NE system

$$\mathcal{G}_R = (A(Q + \Theta^{-1} + R_p)^{-1} A^T + R_d) \approx L D_L L^T + D_S$$

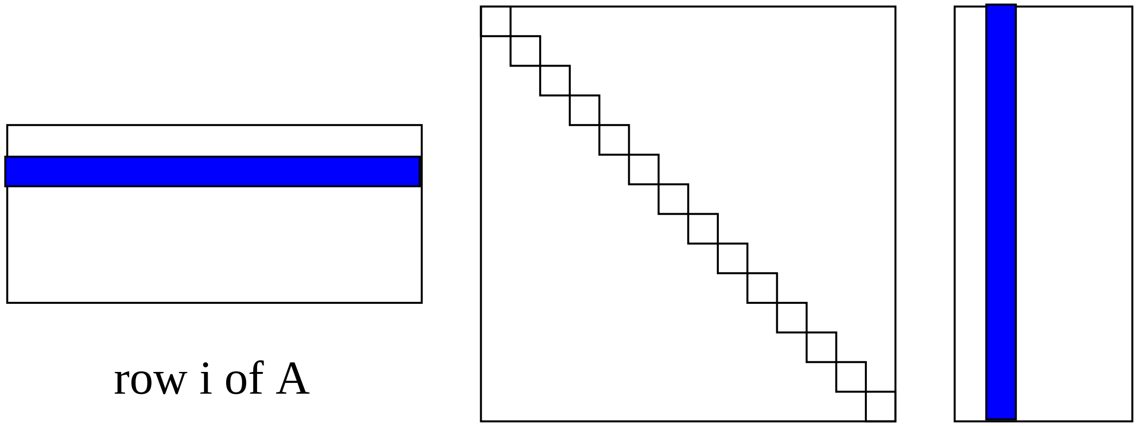
$$L D_L L^T + D_S =$$



- low rank matrix L: $k \ll m$
- D_L contains k largest pivots of $\mathcal{G}_R$

Matrix-Free Implementation

$A\Theta A^T =$



To build the preconditioner we need only:

- a complete diagonal of $A\Theta A^T$ → $d_{ii} = r_i^T \Theta \, r_i$
- a column i of $A\Theta A^T$ → $(A\Theta) \cdot r_i$

both operations are **easy** if we access r_i^T (row i of A).

Example Applications

- Quadratic Assignment Problem (QAP)
- Quantum Physics

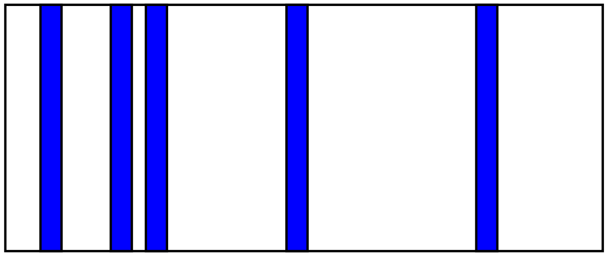
G., Matrix-Free Interior Point Method,
Computational Optimization and Applications,
vol. 51 (2012) 457–480.

G., Interior Point Methods 25 Years Later,
European Journal of Operational Research,
vol. 218 (2012) 587–601.

Sparse Approximations joint work with **Kimonas Fountoulakis** and **Pavel Zhlobich**

$$\min_x \ \frac{1}{2}\|Ax-b\|_2^2 + \tau\|x\|_1,$$

where

$A =$  $\in \mathcal{R}^{m\times n}.$

- A is often available only as an operator
- "Two-way" orthogonality property
- Low complexity $\mathcal{O}(n\log n)$ of operations $A\cdot u,\ A^T\cdot v$

Two-way Orthogonality of A

- *rows* of A are orthogonal to each other (A is built of a subset of rows of an othonormal matrix $U \in \mathcal{R}^{n \times n}$)

$$AA^T = I_m.$$

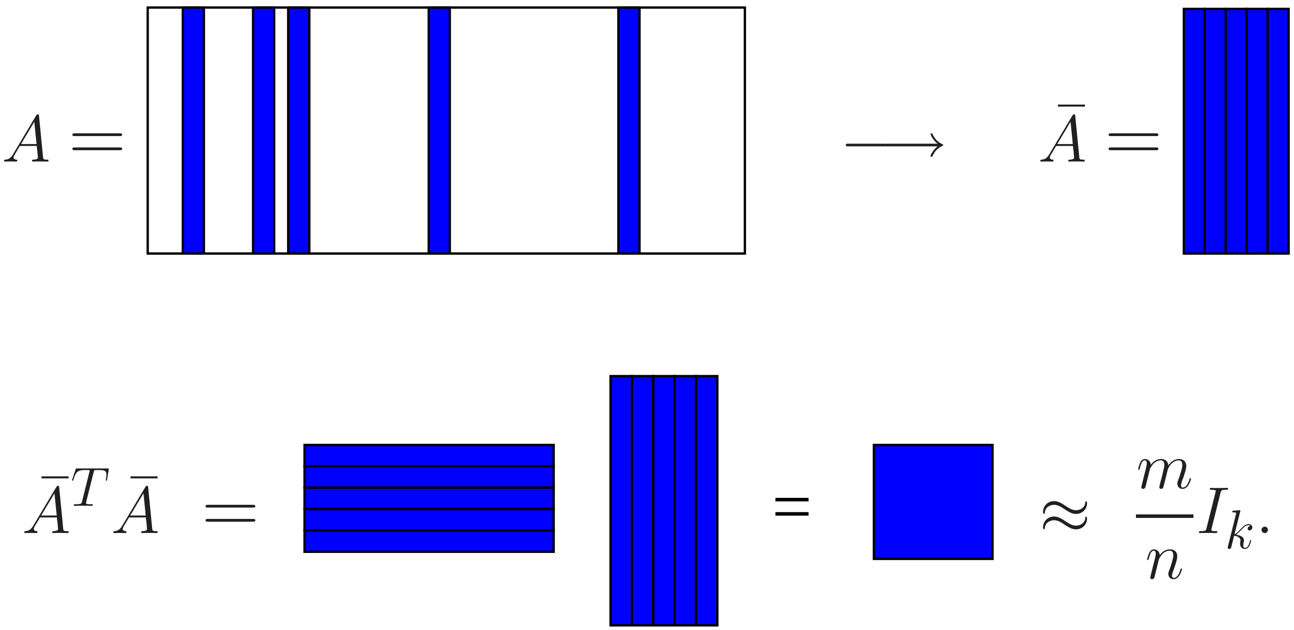
- small subsets of *columns* of A are nearly-orthogonal to each other: *Restricted Isometry Property (RIP)*

$$\|\bar{A}^T \bar{A} - \frac{m}{n} I_k\| \leq \delta \in (0,1).$$

Candès, Romberg & Tao,
Comm on Pure and Appl Maths 59 (2005) 1207-1233.

Restricted Isometry Property

Matrix $\bar{A} \in \mathcal{R}^{m \times k}$ is built of a subset of columns of $A \in \mathcal{R}^{m \times n}$.



$$A = \qquad \longrightarrow \qquad \bar{A} =$$

$$\bar{A}^T \bar{A} \; = \qquad = \qquad \approx \; \frac{m}{n} I_k.$$

This yields a very well conditioned optimization problem.

Problem Reformulation

$$\min_x \ \frac{1}{2}\|Ax - b\|_2^2 + \tau\|x\|_1,$$

Replace $\|x\|_1$ with $\|x\|_1 = 1_{2n}^T z$ using $|x_i| = z_i + z_{i+n}$.
(Increases problem dimension from n to $2n$.)

$$\min_{z \geq 0} \ \frac{1}{2} z^T Q z + c^T z,$$

where

$$Q = \begin{bmatrix} A^T \\ -A^T \end{bmatrix} [\, A \ \ -A \,] = \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix} \in \mathcal{R}^{2n \times 2n}$$

Preconditioner

Approximate

$$\mathcal{M} = \begin{bmatrix} A^TA & -A^TA \\ -A^TA & A^TA \end{bmatrix} + \begin{bmatrix} \Theta_1^{-1} & \\ & \Theta_2^{-1} \end{bmatrix}$$

with

$$\mathcal{P} = \frac{m}{n} \begin{bmatrix} I_n & -I_n \\ -I_n & I_n \end{bmatrix} + \begin{bmatrix} \Theta_1^{-1} & \\ & \Theta_2^{-1} \end{bmatrix}.$$

We expect (*optimal partition*):

- k entries of $\Theta^{-1} \to 0, \quad k \ll 2n,$
- $2n - k$ entries of $\Theta^{-1} \to \infty.$

Spectral Properties of $\mathcal{P}^{-1}\mathcal{M}$

Theorem

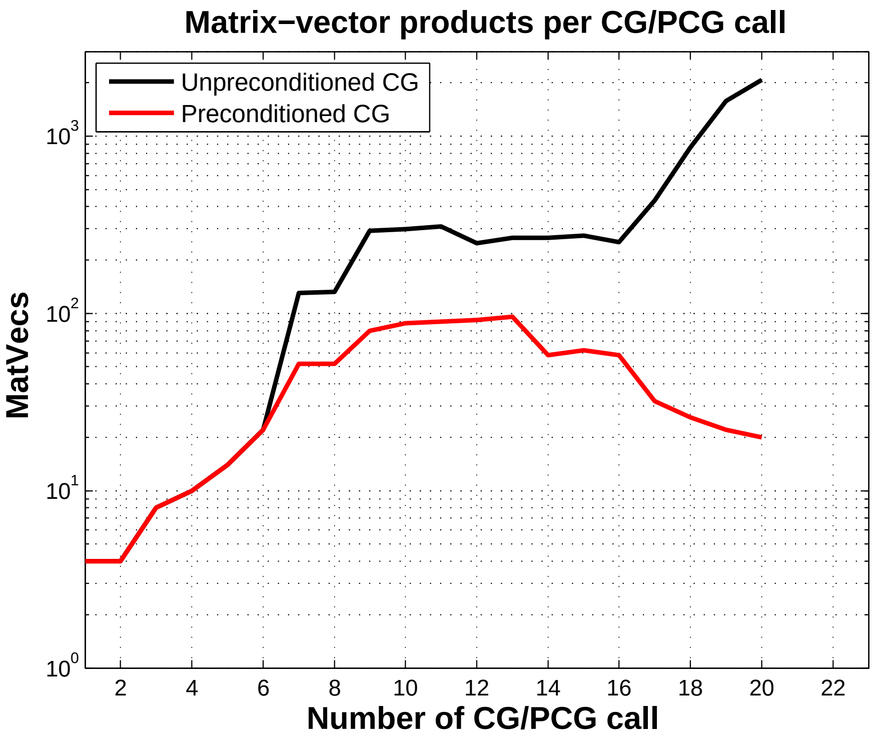
- Exactly n eigenvalues of $\mathcal{P}^{-1}\mathcal{M}$ are 1.
- The remaining n eigenvalues satisfy

$$|\lambda(\mathcal{P}^{-1}\mathcal{M}) - 1| \leq \delta_k + \frac{n}{m\delta_k L},$$

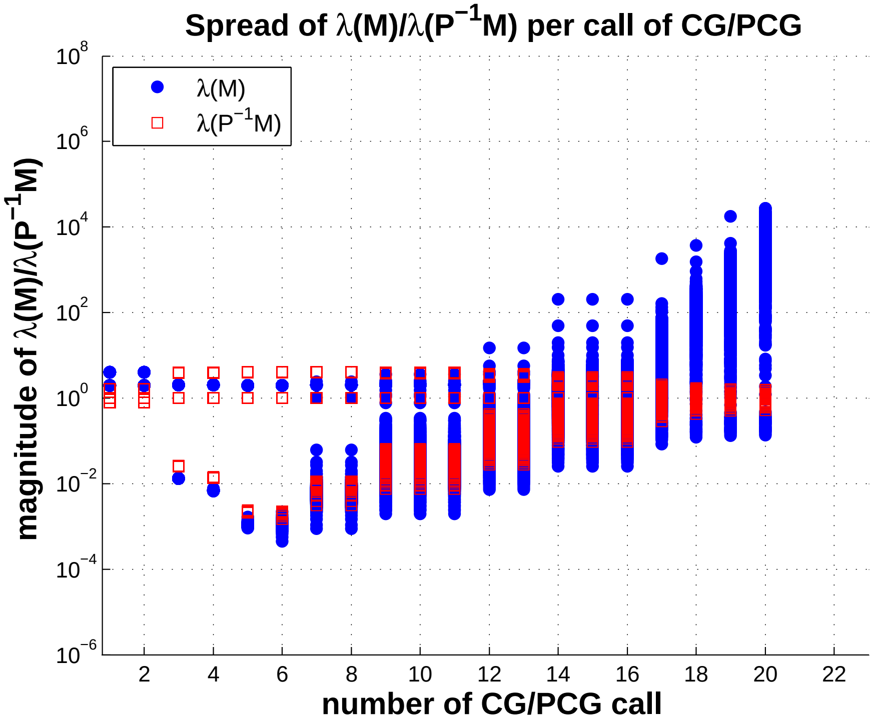
where δ_k is the RIP-constant, and
L is a threshold of "large" $(\Theta_1 + \Theta_2)^{-1}$.

Fountoulakis, G., Zhlobich
Matrix-free IPM for Compressed Sensing Problems,
ERGO Technical Report, 2012.

Preconditioning



Matrix−vector products per CG/PCG call



Spread of $\lambda(M)/\lambda(P^{-1}M)$ per call of CG/PCG

Computational Results: Comparing MatVecs

Prob size	k	**mf-IPM**	**NestA**	**FPC**
4k	51	301	424	91
16k	204	307	461	91
64k	816	407	453	89
256k	3264	537	589	89
1M	13056	613	576	87

NestA, Nesterov's smoothing gradient
Becker, Bobin & Candés,
`http://www-stat.stanford.edu/~candes/nesta/`

FPC, Fixed-Point Continuation
Yin, Osher, Goldfarb & Darbon, *SIIMS* 1 (2008).

Google Problem joint work with **Kristian Woodsend**

An adjacency matrix $G \in \mathcal{R}^{n \times n}$ of web-page links is given (web-pages are the nodes). G is *column-stochastic*.

Teleportation:

$$M = \lambda G + (1 - \lambda)\frac{1}{n}ee^T,$$

with $\lambda \in (0,1)$, usually $\lambda = 0.85$.

Find the *dominant right eigenvector* x of M with eigenvalue equal to 1

$$Mx = x, \qquad \text{such that} \qquad e^T x = 1,\ x \geq 0.$$

and use x as a **ranking vector**.

Google Problem

$$\begin{array}{rl} \min & \frac{1}{2}\|Mx - x\|_2^2 \\ \text{s.t.} & e^T x = 1, \;\; x \geq 0 \end{array}$$

Rearrange:

$$\|Mx - x\|_2^2 = x^T (M - I)^T (M - I) x$$

to produce a standard QP formulation with

$$Q = (M - I)^T (M - I).$$

A very easy QP problem!

Preconditioner for Google Problem

Approximate

$$\mathcal{M} = \begin{bmatrix} Q + \Theta^{-1} & e \\ e^T & 0 \end{bmatrix}$$

with

$$\mathcal{P} = \begin{bmatrix} D_Q & e \\ e^T & 0 \end{bmatrix},$$

where $D_Q = diag\{Q + \Theta^{-1}\}$.

G., Woodsend
Matrix-free IPM for Google Problems,
ERGO Technical Report (in preparation) 2012.

Computational Results:

	Size	degree	IPM-iters	**MatVecs**	time
$\lambda = 0.85$	4k	20	6	13	0.34
	16k	20	5	8	0.83
	64k	20	4	5	2.67
	256k	20	3	4	9.02
	1M	20	3	11	150.29
$\lambda = 1.0$	4k	20	6	13	0.41
	16k	20	5	8	0.83
	64k	20	4	5	2.65
	256k	20	3	6	11.21
	1M	20	3	14	178.94

Special Structure in A?

Matrix-Free IPM

uses matrix A only to perform matrix-vector products.

Any structure (sparsity, block-sparsity, etc) in A
is naturally exploited!

Conclusions

- The worst-case complexity of IPMs can't be matched
- IPMs are:
  - beyond competition for dense problems
  - competitive for certain sparse problems
  - and can also be specialized for easy problems
- Matrix-free IPM can solve very large problems

Challenge:

Special preconditioners may be needed for particular problem classes.

Thank You!