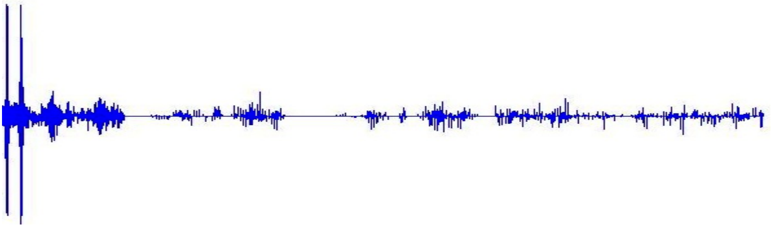
# Putting the Curvature Back into Sparse Solvers

Julie Nutini
*University of British Columbia*



COCANA Seminar Series, UBC Okanagan
March 6th, 2014

# OUTLINE

# GENERAL PROBLEM SETTING

Consider the problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) + g(x) \equiv F(x),$$

where

- $f : \mathbb{R}^n \to \mathbb{R}$ is a **convex** quadratic function, and
- $g : \mathbb{R}^n \to \mathbb{R}$ is a closed **convex** separable function with an inexpensive proximal operator.

# GENERAL PROBLEM SETTING

Consider the problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + g(x) \equiv F(x),$$

where

- $f : \mathbb{R}^n \to \mathbb{R}$ is a **convex** quadratic function, and
- $g : \mathbb{R}^n \to \mathbb{R}$ is a closed **convex** separable function with an inexpensive proximal operator.

$\rightarrow$ $g$ is not necessarily differentiable

# EXISTING METHODS

**First-order methods**

$\rightarrow$ handle non-differentiable problems

e.g.

- Cutting Plane
- Smoothing
- Subgradient
- Bundle
- Proximal Gradient
- ...

# EXISTING METHODS

**First-order methods**

$\rightarrow$ handle non-differentiable problems

e.g.

- Cutting Plane
- Smoothing
- Subgradient
- Bundle
- Proximal Gradient
- ...

$\rightarrow$ can be slow to converge

# VISUALIZE

(OUR MOTIVATION . . . )

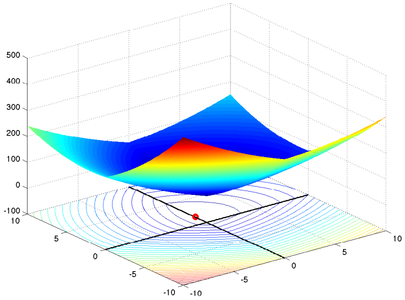
The *basis pursuit denoising problem*:

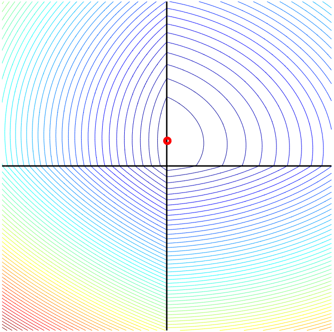
$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1$$

# VISUALIZE
(OUR MOTIVATION . . . )

The *basis pursuit denoising problem*:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|Ax-b\|_2^2 + \lambda\|x\|_1$$

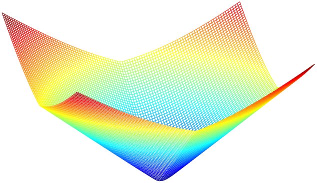


# GOAL

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + g(x) \equiv F(x)$$

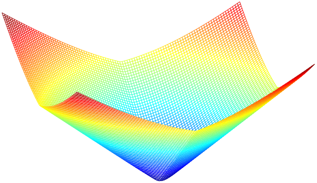
- Exploit curvature
- Create second-order (sparse) solver
- Improve convergence rates



# GOAL

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + g(x) \equiv F(x)$$

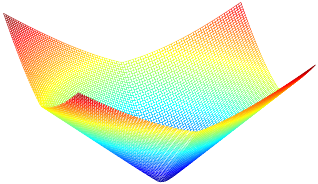
- Exploit curvature
- Create second-order (sparse) solver
- Improve convergence rates



# GOAL

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) + g(x) \equiv F(x)$$

- Exploit curvature
- Create second-order (sparse) solver
- Improve convergence rates



# OUR APPROACH

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + g(x) \equiv F(x)$$

A two-phase algorithm that combines 2 well-known methods:

# OUR APPROACH

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) + g(x) \equiv F(x)$$

A two-phase algorithm that combines 2 well-known methods:

- **Phase 1:** Proximal Gradient Method

# OUR APPROACH

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + g(x) \equiv F(x)$$

A two-phase algorithm that combines 2 well-known methods:

- **Phase 1:** Proximal Gradient Method
- **Phase 2:** Conjugate Gradient Method

# OUR APPROACH

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + g(x) \equiv F(x)$$

A two-phase algorithm that combines 2 well-known methods:

- **Phase 1:** Proximal Gradient Method
- **Phase 2:** Conjugate Gradient Method

**NOTE:** No additional storage overhead required.

Two phases, two beautiful methods …

# PROXIMAL GRADIENT METHOD

( PHASE 1 )

- An iterative method that solves a convex subproblem involving a linearization of $f$ at $x$.

# PROXIMAL GRADIENT METHOD

( PHASE 1 )

- An iterative method that solves a convex subproblem involving a linearization of $f$ at $x$.

The *proximal operator* for a convex function $g$ is defined as

$$\mathbf{prox}_{\alpha g(\cdot)}(x) = \underset{u}{\operatorname{argmin}}\ g(u) + \frac{1}{2\alpha}\|u - x\|^2, \quad (\alpha > 0).$$

# PROXIMAL GRADIENT METHOD

( PHASE 1 )

A *proximal gradient iteration* applies the proximal operator to a **steepest descent** step on $f(x_k)$:

# PROXIMAL GRADIENT METHOD

( PHASE 1 )

A *proximal gradient iteration* applies the proximal operator to a **steepest descent** step on $f(x_k)$:

$$\bar{x} \quad = \quad x_k - \alpha \nabla f(x_k), \quad (\alpha > 0)$$

# PROXIMAL GRADIENT METHOD

( PHASE 1 )

A *proximal gradient iteration* applies the proximal operator to a **steepest descent** step on $f(x_k)$:

$$\begin{aligned} \bar{x} &= x_k - \alpha \nabla f(x_k), \quad (\alpha > 0) \\ x_{k+1} &= \mathbf{prox}_{\alpha g(\cdot)}(\bar{x}) \end{aligned}$$

# PROXIMAL GRADIENT METHOD

( PHASE 1 )

A *proximal gradient iteration* applies the proximal operator to a **steepest descent** step on $f(x_k)$:

$$\begin{aligned} \bar{x} &= x_k - \alpha \nabla f(x_k), \quad (\alpha > 0) \\ x_{k+1} &= \mathbf{prox}_{\alpha g(\cdot)}(\bar{x}) \end{aligned}$$

(**Note:** When $g \equiv 0$, then just steepest descent)

# CONJUGATE GRADIENT METHOD

( PHASE 2 )

- A non-stationary iterative method guarantees convergence in at most $n$ iterations.

# CONJUGATE GRADIENT METHOD

( PHASE 2 )

- A non-stationary iterative method guarantees convergence in at most $n$ iterations.

Solves linear systems of the form

$$A^\top A x = A^\top b, \qquad \text{i.e.,} \quad \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|Ax - b\|_2^2$$

# CONJUGATE GRADIENT METHOD

( PHASE 2 )

- A non-stationary iterative method guarantees convergence in at most $n$ iterations.

Solves linear systems of the form

$$A^\top A x = A^\top b, \qquad \text{i.e.,} \quad \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|Ax - b\|_2^2$$

We want to solve a linear system of the form

$$A^\top A x = A^\top b - g_x, \qquad \text{i.e.,} \quad \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|Ax - b\|_2^2 + g(x)$$

with $g_x \in \partial g(x)$.

# CONJUGATE GRADIENT METHOD
( PHASE 2 )

- A non-stationary iterative method guarantees convergence in at most $n$ iterations.

Solves linear systems of the form

$$A^\top A x = A^\top b, \qquad \text{i.e.,} \quad \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|Ax-b\|_2^2$$

We want to solve a linear system of the form

$$A^\top A x = A^\top b - g_x, \qquad \text{i.e.,} \quad \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|Ax-b\|_2^2 + g(x)$$

with $g_x \in \partial g(x)$.

**CATCH**: CG method requires a differentiable problem.

# OUR APPROACH

(A FEW MORE DETAILS . . . )

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) + g(x) \equiv F(x)$$

# OUR APPROACH

(A FEW MORE DETAILS . . . )

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + g(x) \equiv F(x)$$

- **Phase 1:** Proximal Gradient Method

# OUR APPROACH

(A FEW MORE DETAILS . . . )

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + g(x) \equiv F(x)$$

- **Phase 1:** Proximal Gradient Method
  - Find an *active set* of $F$ at current iteration.

# OUR APPROACH

(A FEW MORE DETAILS . . . )

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) + g(x) \equiv F(x)$$

- **Phase 1:** Proximal Gradient Method
  - Find an *active set* of $F$ at current iteration.
- **Phase 2:** Conjugate Gradient Method

# OUR APPROACH

(A FEW MORE DETAILS . . . )

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) + g(x) \equiv F(x)$$

- **Phase 1:** Proximal Gradient Method
  - Find an *active set* of $F$ at current iteration.
- **Phase 2:** Conjugate Gradient Method
  - Explore *smooth reduced subproblem* defined by the *free set*.

# OUR APPROACH

(A FEW MORE DETAILS . . . )

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + g(x) \equiv F(x)$$

- **Phase 1:** Proximal Gradient Method
  - Find an *active set* of $F$ at current iteration.
- **Phase 2:** Conjugate Gradient Method
  - Explore *smooth reduced subproblem* defined by the *free set*.
- Check if updated active set is *stationary*.
  - **Yes**: Continue exploring in Phase 2.
  - **No**: Define new working set in Phase 1.

# Proximal Gradient Conjugate Gradient Algorithm
# ( `pgcg` )

# VISUALIZE

$g(\cdot) = \delta_{\Omega}(\cdot)$

# VISUALIZE

$g(\cdot) = \delta_\Omega(\cdot)$

Consider a quadratic bound constrained problem:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + \delta_\Omega(x) \equiv F(x),$$

where $\Omega = \{y : l \leq y \leq u\}$ and $\delta_\Omega(x) = \begin{cases} 0 & \text{if } x \in \Omega; \\ +\infty & \text{if } x \notin \Omega. \end{cases}$

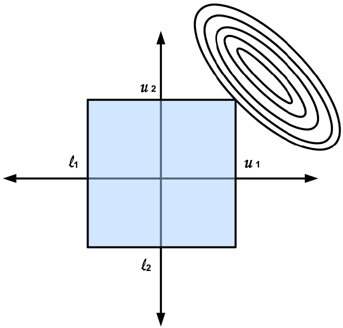
# VISUALIZE

$g(\cdot) = \delta_\Omega(\cdot)$

Consider a quadratic bound constrained problem:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + \delta_\Omega(x) \equiv F(x),$$

where $\Omega = \{y : l \leq y \leq u\}$ and $\delta_\Omega(x) = \begin{cases} 0 & \text{if } x \in \Omega; \\ +\infty & \text{if } x \notin \Omega. \end{cases}$



# VISUALIZE

$g(\cdot) = \delta_\Omega(\cdot)$

Consider the proximal operator for $\delta_\Omega(x)$:

# VISUALIZE

$g(\cdot) = \delta_\Omega(\cdot)$

Consider the proximal operator for $\delta_\Omega(x)$:

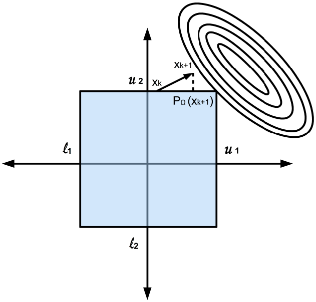
$$\begin{aligned} \mathbf{prox}_{\alpha\delta_\Omega(\cdot)}(x) &= \underset{u \in \mathbb{R}^n}{\text{argmin}} \quad \delta_\Omega(u) + \frac{1}{2\alpha}\|u - x\|_2^2 \\ &= \underset{u \in \mathbb{R}^n}{\text{argmin}} \quad \alpha\,\delta_\Omega(u) + \frac{1}{2}\|u - x\|_2^2 \\ &= \underset{u \in \Omega}{\text{argmin}} \quad \frac{1}{2}\|u - x\|_2^2 \end{aligned}$$

# VISUALIZE

$g(\cdot) = \delta_\Omega(\cdot)$

Consider the proximal operator for $\delta_\Omega(x)$:

$$\begin{aligned}
\mathbf{prox}_{\alpha\delta_\Omega(\cdot)}(x) &= \underset{u \in \mathbb{R}^n}{\text{argmin}} \quad \delta_\Omega(u) + \frac{1}{2\alpha}\|u-x\|_2^2 \\
&= \underset{u \in \mathbb{R}^n}{\text{argmin}} \quad \alpha\,\delta_\Omega(u) + \frac{1}{2}\|u-x\|_2^2 \\
&= \underset{u \in \Omega}{\text{argmin}} \quad \frac{1}{2}\|u-x\|_2^2 \\
&= P_\Omega(x)
\end{aligned}$$



# THE WORKING SET

( PHASE 1 )

The *active set*:

$$\mathcal{A}(x) = \{i : \partial g_i(x_i) \text{ is } \textit{not} \text{ a singleton}\}$$

The *free set*:

$$\mathcal{F}(x) = \{i : i \notin \mathcal{A}(x)\}$$

# THE WORKING SET

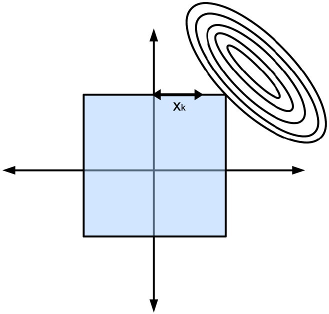
( PHASE 1 )

The *active set*:

$$\mathcal{A}(x) = \{i : \partial g_i(x_i) \text{ is } \textit{not} \text{ a singleton}\}$$

The *free set*:

$$\mathcal{F}(x) = \{i : i \notin \mathcal{A}(x)\}$$



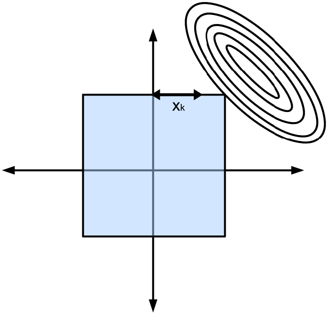
# THE WORKING SET
( PHASE 1 )

The *active set*:

$$\mathcal{A}(x) = \{i : \partial g_i(x_i) \text{ is } \textit{not} \text{ a singleton}\}$$

The *free set*:

$$\mathcal{F}(x) = \{i : i \notin \mathcal{A}(x)\}$$




**Result:** $g$ is differentiable with respect to the free variables.

# SMOOTH REDUCED SUBPROBLEM

( PHASE 2 )

# SMOOTH REDUCED SUBPROBLEM

( PHASE 2 )

Consider a restriction of the objective along $x_k + d$:

# SMOOTH REDUCED SUBPROBLEM

( PHASE 2 )

Consider a restriction of the objective along $x_k + d$:

$$
\begin{aligned}
& F(x_k + d) \\
= \quad & f(x_k + d) + g(x_k + d)
\end{aligned}
$$

# SMOOTH REDUCED SUBPROBLEM

( PHASE 2 )

Consider a restriction of the objective along $x_k + d$:

$$F(x_k + d)$$

$$= f(x_k + d) + g(x_k + d)$$

$$= \frac{1}{2}(x_k + d)^\top H(x_k + d) + b^\top (x_k + d) + g(x_k + d)$$

# SMOOTH REDUCED SUBPROBLEM

( PHASE 2 )

Consider a restriction of the objective along $x_k + d$:

$$
\begin{aligned}
& F(x_k + d) \\
= \quad & f(x_k + d) + g(x_k + d) \\
= \quad & \frac{1}{2}(x_k + d)^\top H(x_k + d) + b^\top (x_k + d) + g(x_k + d) \\
= \quad & \ldots \text{linearize, substitute, simplify} \ldots
\end{aligned}
$$

# SMOOTH REDUCED SUBPROBLEM

( PHASE 2 )

Use truncated CG to *approximately* solve

# SMOOTH REDUCED SUBPROBLEM

( PHASE 2 )

Use truncated CG to *approximately* solve

$$\underset{w}{\text{minimize}} \quad \frac{1}{2} w^\top H_k w + r_k^\top w \equiv F_k(w),$$

where

- $Z_k = \mathcal{I}[:, \mathcal{F}(x_k)]$,
- $H_k = Z_k^\top H Z_k$ is a reduced Hessian, and
- $r_k = Z_k^\top (\nabla f(x_k) + g_k)$ is a reduced gradient of $F$, $g_k \in \partial g(x_k)$.

# SMOOTH REDUCED SUBPROBLEM

( PHASE 2 )

Use truncated CG to *approximately* solve

$$\underset{w}{\text{minimize}} \quad \frac{1}{2} w^\top H_k w + r_k^\top w \equiv F_k(w),$$

where

- $Z_k = \mathcal{I}[:, \mathcal{F}(x_k)]$,
- $H_k = Z_k^\top H Z_k$ is a reduced Hessian, and
- $r_k = Z_k^\top (\nabla f(x_k) + g_k)$ is a reduced gradient of $F$, $g_k \in \partial g(x_k)$.

Then

$$d_k = Z_k w^* = \begin{cases} w_i^*, & \text{for } i \in \mathcal{F}(x_k) \\ 0, & \text{for } i \in \mathcal{A}(x_k) \end{cases}.$$

# STATIONARY WORKING SET

( PHASE 1 $\overset{?}{\longleftrightarrow}$ PHASE 2 )

# STATIONARY WORKING SET

( PHASE 1 $\overset{?}{\leftrightarrow}$ PHASE 2 )

The *binding set* of $F$ at a point $x$:

$$\mathcal{B}(x) = \{ i : i \in \mathcal{A}(x) \text{ and } [-\nabla f(x)]_i \in \partial g_i(x_i) \},$$

i.e., the set of indices corresponding to the active variables that are stationary:

$$-\nabla f(x) \in \partial g(x).$$

# STATIONARY WORKING SET

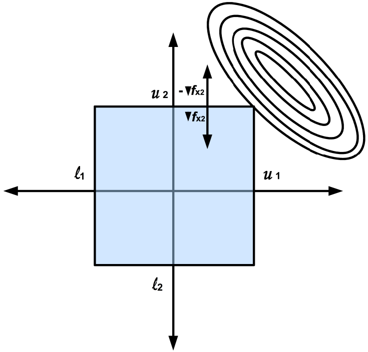
( PHASE 1 $\overset{?}{\leftrightarrow}$ PHASE 2 )

The *binding set* of $F$ at a point $x$:

$$\mathcal{B}(x) = \{i : i \in \mathcal{A}(x) \text{ and } [-\nabla f(x)]_i \in \partial g_i(x_i)\},$$

i.e., the set of indices corresponding to the active variables that are stationary:

$$-\nabla f(x) \in \partial g(x).$$



## PROXIMAL GRADIENT CONJUGATE GRADIENT ( PGCG ) ALGORITHM:

## PROXIMAL GRADIENT CONJUGATE GRADIENT ( PGCG ) ALGORITHM:

1. **Phase 1: Determine working set via proximal gradient iteration.**

$$y_{j+1} = \mathbf{prox}_{\alpha g(\cdot)}\big(y_j - \alpha \nabla f(y_j)\big), \quad \alpha > 0$$

   Set $x_k$ equal to first $y_j$ satisfying

$$\mathcal{A}(y_j) = \mathcal{A}(y_{j-1}) \quad \text{OR} \quad F(y_{j-1}) - F(y_j) \le \eta_2 \max\{F(y_{l-1}) - F(y_l) : 1 \le l < j\}.$$

2. **Phase 2: Explore working set via truncated conjugate gradient.**

   Set $d_k = Z_k w_j$ for first $w_j$ satisfying

$$\|r_k\|_2 \le \tau_{CG} \quad \text{OR} \quad F_k(w_{j-1}) - F_k(w_j) \le \eta_1 \max\{F_k(w_{l-1}) - F_k(w_l) : 1 \le l < j\}.$$

3. **Line search:**

   Set $x_{k+1} = x_k + \alpha_k d_k$ for some $\alpha_k > 0$.

   If $\mathcal{A}(x_{k+1}) = \mathcal{B}(x_{k+1})$, continue **Phase 2**.
   Else, loop to **Phase 1**.

## Proximal gradient conjugate gradient ( pgcg ) algorithm:

1. **Phase 1: Determine working set via proximal gradient iteration.**

$$y_{j+1} = \mathbf{prox}_{\alpha g(\cdot)}\big(y_j - \alpha \nabla f(y_j)\big), \quad \alpha > 0$$

Set $x_k$ equal to first $y_j$ satisfying

$$\mathcal{A}(y_j) = \mathcal{A}(y_{j-1}) \quad \text{OR} \quad F(y_{j-1}) - F(y_j) \le \eta_2 \max\{F(y_{l-1}) - F(y_l) : 1 \le l < j\}.$$

2. **Phase 2: Explore working set via truncated conjugate gradient.**

Set $d_k = Z_k w_j$ for first $w_j$ satisfying

$$\|r_k\|_2 \le \tau_{CG} \quad \text{OR} \quad F_k(w_{j-1}) - F_k(w_j) \le \eta_1 \max\{F_k(w_{l-1}) - F_k(w_l) : 1 \le l < j\}.$$

3. **Line search:**

Set $x_{k+1} = x_k + \alpha_k d_k$ for some $\alpha_k > 0$.

If $\mathcal{A}(x_{k+1}) = \mathcal{B}(x_{k+1})$, continue **Phase 2**.
Else, loop to **Phase 1**.

# PROXIMAL GRADIENT CONJUGATE GRADIENT ( PGCG ) ALGORITHM:

1. **Phase 1: Determine working set via proximal gradient iteration.**

$$y_{j+1} = \mathbf{prox}_{\alpha g(\cdot)}\big(y_j - \alpha \nabla f(y_j)\big), \quad \alpha > 0$$

Set $x_k$ equal to first $y_j$ satisfying

$$\mathcal{A}(y_j) = \mathcal{A}(y_{j-1}) \quad \text{OR} \quad F(y_{j-1}) - F(y_j) \le \eta_2 \max\{F(y_{l-1}) - F(y_l) : 1 \le l < j\}.$$

2. **Phase 2: Explore working set via truncated conjugate gradient.**

Set $d_k = Z_k w_j$ for first $w_j$ satisfying

$$\|r_k\|_2 \le \tau_{CG} \quad \text{OR} \quad F_k(w_{j-1}) - F_k(w_j) \le \eta_1 \max\{F_k(w_{l-1}) - F_k(w_l) : 1 \le l < j\}.$$

3. **Line search:**

Set $x_{k+1} = x_k + \alpha_k d_k$ for some $\alpha_k > 0$.

If $\mathcal{A}(x_{k+1}) = \mathcal{B}(x_{k+1})$, continue **Phase 2**.
Else, loop to **Phase 1**.

## PROXIMAL GRADIENT CONJUGATE GRADIENT ( PGCG ) ALGORITHM:

1. **Phase 1: Determine working set via proximal gradient iteration.**

$$y_{j+1} = \mathbf{prox}_{\alpha g(\cdot)}\big(y_j - \alpha \nabla f(y_j)\big), \quad \alpha > 0$$

   Set $x_k$ equal to first $y_j$ satisfying

$$\mathcal{A}(y_j) = \mathcal{A}(y_{j-1}) \quad \text{OR} \quad F(y_{j-1}) - F(y_j) \leq \eta_2 \max\{F(y_{l-1}) - F(y_l) : 1 \leq l < j\}.$$

2. **Phase 2: Explore working set via truncated conjugate gradient.**

   Set $d_k = Z_k w_j$ for first $w_j$ satisfying

$$\|r_k\|_2 \leq \tau_{CG} \quad \text{OR} \quad F_k(w_{j-1}) - F_k(w_j) \leq \eta_1 \max\{F_k(w_{l-1}) - F_k(w_l) : 1 \leq l < j\}.$$

3. **Line search:**

   Set $x_{k+1} = x_k + \alpha_k d_k$ for some $\alpha_k > 0$.

   If $\mathcal{A}(x_{k+1}) = \mathcal{B}(x_{k+1})$, continue **Phase 2**.
   Else, loop to **Phase 1**.

## PROXIMAL GRADIENT CONJUGATE GRADIENT ( PGCG ) ALGORITHM:

1. **Phase 1: Determine working set via proximal gradient iteration.**

$$y_{j+1} = \mathbf{prox}_{\alpha g(\cdot)}\big(y_j - \alpha \nabla f(y_j)\big), \quad \alpha > 0$$

   Set $x_k$ equal to first $y_j$ satisfying

$$\mathcal{A}(y_j) = \mathcal{A}(y_{j-1}) \quad \text{OR} \quad F(y_{j-1}) - F(y_j) \leq \eta_2 \max\{F(y_{l-1}) - F(y_l) : 1 \leq l < j\}.$$

2. **Phase 2: Explore working set via truncated conjugate gradient.**

   Set $d_k = Z_k w_j$ for first $w_j$ satisfying

$$\|r_k\|_2 \leq \tau_{CG} \quad \text{OR} \quad F_k(w_{j-1}) - F_k(w_j) \leq \eta_1 \max\{F_k(w_{l-1}) - F_k(w_l) : 1 \leq l < j\}.$$

3. **Line search:**

   Set $x_{k+1} = x_k + \alpha_k d_k$ for some $\alpha_k > 0$.

   If $\mathcal{A}(x_{k+1}) = \mathcal{B}(x_{k+1})$, continue **Phase 2**.
   Else, loop to **Phase 1**.

## Proximal gradient conjugate gradient ( pgcg ) algorithm:

1. **Phase 1: Determine working set via proximal gradient iteration.**

$$y_{j+1} = \mathbf{prox}_{\alpha g(\cdot)}\big(y_j - \alpha \nabla f(y_j)\big), \quad \alpha > 0$$

Set $x_k$ equal to first $y_j$ satisfying

$$\mathcal{A}(y_j) = \mathcal{A}(y_{j-1}) \quad \text{OR} \quad F(y_{j-1}) - F(y_j) \le \eta_2 \max\{F(y_{l-1}) - F(y_l) : 1 \le l < j\}.$$

2. **Phase 2: Explore working set via truncated conjugate gradient.**

Set $d_k = Z_k w_j$ for first $w_j$ satisfying

$$\|r_k\|_2 \le \tau_{\text{CG}} \quad \text{OR} \quad F_k(w_{j-1}) - F_k(w_j) \le \eta_1 \max\{F_k(w_{l-1}) - F_k(w_l) : 1 \le l < j\}.$$

3. **Line search:**

Set $x_{k+1} = x_k + \alpha_k d_k$ for some $\alpha_k > 0$.

If $\mathcal{A}(x_{k+1}) = \mathcal{B}(x_{k+1})$, continue **Phase 2**.
Else, loop to **Phase 1**.

## Proximal gradient conjugate gradient ( pgcg ) algorithm:

1. **Phase 1: Determine working set via proximal gradient iteration.**

$$y_{j+1} = \mathbf{prox}_{\alpha g(\cdot)}\big(y_j - \alpha \nabla f(y_j)\big), \quad \alpha > 0$$

   Set $x_k$ equal to first $y_j$ satisfying

$$\mathcal{A}(y_j) = \mathcal{A}(y_{j-1}) \quad \text{OR} \quad F(y_{j-1}) - F(y_j) \le \eta_2 \max\{F(y_{l-1}) - F(y_l) : 1 \le l < j\}.$$

2. **Phase 2: Explore working set via truncated conjugate gradient.**
   Set $d_k = Z_k w_j$ for first $w_j$ satisfying

$$\|r_k\|_2 \le \tau_{\text{CG}} \quad \text{OR} \quad F_k(w_{j-1}) - F_k(w_j) \le \eta_1 \max\{F_k(w_{l-1}) - F_k(w_l) : 1 \le l < j\}.$$

3. **Line search**:
   Set $x_{k+1} = x_k + \alpha_k d_k$ for some $\alpha_k > 0$.

   If $\mathcal{A}(x_{k+1}) = \mathcal{B}(x_{k+1})$, continue **Phase 2**.
   Else, loop to **Phase 1**.

## PROXIMAL GRADIENT CONJUGATE GRADIENT ( PGCG ) ALGORITHM:

1. **Phase 1: Determine working set via proximal gradient iteration.**

$$y_{j+1} = \mathbf{prox}_{\alpha g(\cdot)}\big(y_j - \alpha \nabla f(y_j)\big), \quad \alpha > 0$$

   Set $x_k$ equal to first $y_j$ satisfying

$$\mathcal{A}(y_j) = \mathcal{A}(y_{j-1}) \quad \text{OR} \quad F(y_{j-1}) - F(y_j) \le \eta_2 \max\{F(y_{l-1}) - F(y_l) : 1 \le l < j\}.$$

2. **Phase 2: Explore working set via truncated conjugate gradient.**
   Set $d_k = Z_k w_j$ for first $w_j$ satisfying

$$\|r_k\|_2 \le \tau_{\text{CG}} \quad \text{OR} \quad F_k(w_{j-1}) - F_k(w_j) \le \eta_1 \max\{F_k(w_{l-1}) - F_k(w_l) : 1 \le l < j\}.$$

3. **Line search**:
   Set $x_{k+1} = x_k + \alpha_k d_k$ for some $\alpha_k > 0$.

   If $\mathcal{A}(x_{k+1}) = \mathcal{B}(x_{k+1})$, continue **Phase 2**.
   Else, loop to **Phase 1**.

## PROXIMAL GRADIENT CONJUGATE GRADIENT ( PGCG ) ALGORITHM:

1. **Phase 1: Determine working set via proximal gradient iteration.**

   $$y_{j+1} = \mathbf{prox}_{\alpha g(\cdot)}\big(y_j - \alpha \nabla f(y_j)\big), \quad \alpha > 0$$

   Set $x_k$ equal to first $y_j$ satisfying

   $$\mathcal{A}(y_j) = \mathcal{A}(y_{j-1}) \quad \text{OR} \quad F(y_{j-1}) - F(y_j) \leq \eta_2 \max\{F(y_{l-1}) - F(y_l) : 1 \leq l < j\}.$$

2. **Phase 2: Explore working set via truncated conjugate gradient.**
   Set $d_k = Z_k w_j$ for first $w_j$ satisfying

   $$\|r_k\|_2 \leq \tau_{\text{CG}} \quad \text{OR} \quad F_k(w_{j-1}) - F_k(w_j) \leq \eta_1 \max\{F_k(w_{l-1}) - F_k(w_l) : 1 \leq l < j\}.$$

3. **Line search**:
   Set $x_{k+1} = x_k + \alpha_k d_k$ for some $\alpha_k > 0$.

   If $\mathcal{A}(x_{k+1}) = \mathcal{B}(x_{k+1})$, continue **Phase 2**.
   Else, loop to **Phase 1**.

# LINE SEARCHES

The *directional derivative* of $F$ at $x$ along a direction $d$ is given by

$$F'(x;d) = \lim_{h \to 0^+} \frac{F(x+hd) - F(x)}{h}.$$

# LINE SEARCHES

The *directional derivative* of $F$ at $x$ along a direction $d$ is given by

$$F'(x;d) = \lim_{h \to 0^+} \frac{F(x+hd) - F(x)}{h}.$$

**Note:** If $F(x)$ is differentiable, then $F'(x;d) = d^\top \nabla F(x)$.

# LINE SEARCHES

For fixed $\mu \in (0,1)$, find $\alpha > 0$ such that

**Phase 1:**

$$F(p_k) \le F(x_k) + \mu F'(x_k; p_k - x_k), \qquad p_k = \mathbf{prox}_{\alpha g(\cdot)}(x_k - \alpha \nabla f(x_k))$$

# LINE SEARCHES

For fixed $\mu \in (0,1)$, find $\alpha > 0$ such that

**Phase 1:**

$$F(p_k) \leq F(x_k) + \mu F'(x_k; p_k - x_k), \qquad p_k = \mathbf{prox}_{\alpha g(\cdot)}(x_k - \alpha \nabla f(x_k))$$

**Phase 2:**

$$F(x_k + \alpha d_k) \leq F(x_k) + \mu \alpha F'(x_k; d_k)$$

# LINE SEARCHES

For fixed $\mu \in (0,1)$, find $\alpha > 0$ such that

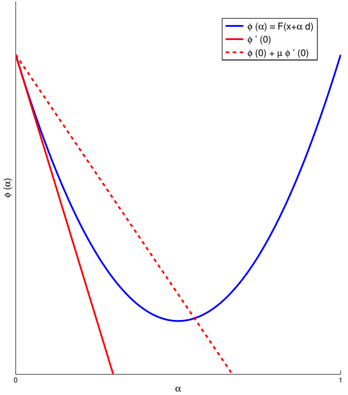
**Phase 1:**

$$F(p_k) \leq F(x_k) + \mu F'(x_k; p_k - x_k), \qquad p_k = \mathbf{prox}_{\alpha g(\cdot)}(x_k - \alpha \nabla f(x_k))$$

**Phase 2:**

$$F(x_k + \alpha d_k) \leq F(x_k) + \mu \alpha F'(x_k; d_k)$$

**Reduction factor:** $\alpha \leftarrow \frac{1}{2}\alpha$



# CONVERGENCE

- A proximal gradient iteration guarantees that

$$F\left(\mathbf{prox}_{\alpha g(\cdot)}(x_k - \alpha \nabla f(x_k))\right) < F(x_k), \quad \text{for some } \alpha > 0,$$

and converges to a fixed point.

# CONVERGENCE

- A proximal gradient iteration guarantees that

  $$F\big(\mathbf{prox}_{\alpha g(\cdot)}(x_k - \alpha \nabla f(x_k))\big) < F(x_k), \qquad \text{for some } \alpha > 0,$$

  and converges to a fixed point.

- The conjugate gradient method generates a $w$ such that when $d = Zw$,

  $$F(x_k + \alpha d) \leq F(x_k), \qquad \text{for some } \alpha > 0,$$

  i.e., $d$ is a descent direction.

# CONVERGENCE

- A proximal gradient iteration guarantees that

$$F\big(\mathbf{prox}_{\alpha g(\cdot)}(x_k - \alpha \nabla f(x_k))\big) < F(x_k), \quad \text{for some } \alpha > 0,$$

and converges to a fixed point.

- The conjugate gradient method generates a $w$ such that when $d = Zw$,

$$F(x_k + \alpha d) \leq F(x_k), \quad \text{for some } \alpha > 0,$$

i.e., $d$ is a descent direction.

$\Rightarrow$ **As $F(x)$ is *non-increasing* in Phase 2, global convergence follows from Phase 1.**

$$g(\cdot) = ?$$

(Examples)

# REQUIREMENTS

For a specific function $g$, we require:

# REQUIREMENTS

For a specific function $g$, we require:

- a **proximal operator** for $g$, and

# REQUIREMENTS

For a specific function $g$, we require:

- a **proximal operator** for $g$, and
- definitions of an **active** set and a **binding** set.

# CONVEX QUADRATIC BOUND CONSTRAINED PROBLEM

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + \delta_\Omega(x) \equiv F(x), \text{ where } \Omega = \{y : l \leq y \leq u\}$$

# CONVEX QUADRATIC BOUND CONSTRAINED PROBLEM

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) + \delta_\Omega(x) \equiv F(x), \text{ where } \Omega = \{y : l \leq y \leq u\}$$

**Proximal operator:**

$$\mathbf{prox}_{\delta_\Omega(\cdot)}(x) = \mathrm{P}_\Omega(x)$$

# CONVEX QUADRATIC BOUND CONSTRAINED PROBLEM

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) + \delta_\Omega(x) \equiv F(x), \text{ where } \Omega = \{y : l \leq y \leq u\}$$

**Proximal operator:**

$$\mathbf{prox}_{\delta_\Omega(\cdot)}(x) = \mathrm{P}_\Omega(x)$$

**Active set:**

$$\mathcal{A}(x) = \{i : x_i = l_i \text{ or } x_i = u_i\}$$

**Binding set:**

$$\mathcal{B}(x) = \left\{ i : \begin{array}{ccc} x_i = l_i & \text{and} & [\nabla f(x)]_i \geq 0 \\ x_i = u_i & \text{and} & [\nabla f(x)]_i \leq 0 \end{array} \right\}$$

# CONVEX QUADRATIC BOUND CONSTRAINED PROBLEM

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) + \delta_\Omega(x) \equiv F(x), \text{ where } \Omega = \{y : l \le y \le u\}$$

**Proximal operator:**

$$\mathbf{prox}_{\delta_\Omega(\cdot)}(x) = \mathrm{P}_\Omega(x)$$

**Active set:**

$$\mathcal{A}(x) = \{i : x_i = l_i \text{ or } x_i = u_i\}$$

**Binding set:**

$$\mathcal{B}(x) = \left\{ i : \begin{array}{rcl} x_i = l_i & \text{and} & [\nabla f(x)]_i \ge 0 \\ x_i = u_i & \text{and} & [\nabla f(x)]_i \le 0 \end{array} \right\}$$

**Reduced subproblem:**

$$\underset{w}{\text{minimize}} \quad \frac{1}{2} w^\top H_k w + r_k^\top w \equiv F_k(w), \text{ where } r_k = Z_k^\top \nabla f(x_k)$$

This specialization is the *gradient projection conjugate gradient* method, [Moré, Toraldo, '91].

# BASIS PURSUIT DENOISING PROBLEM

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1 \equiv F(x)$$

# BASIS PURSUIT DENOISING PROBLEM

(PROXIMAL OPERATOR)

$$\mathbf{prox}_{\lambda\|\cdot\|}(x) \quad = \quad \underset{u}{\operatorname{argmin}} \quad \lambda\|u\|_1 + \frac{1}{2}\|u - x\|_2^2$$

# BASIS PURSUIT DENOISING PROBLEM

(PROXIMAL OPERATOR)

$$\mathbf{prox}_{\lambda|\cdot|}(x) = \underset{u}{\operatorname{argmin}} \quad \lambda|u| + \frac{1}{2}|u - x|^2$$

# BASIS PURSUIT DENOISING PROBLEM

(PROXIMAL OPERATOR)

$$\mathbf{prox}_{\lambda|\cdot|}(x) \quad = \quad \underset{u}{\operatorname{argmin}} \quad \lambda|u| + \frac{1}{2}|u-x|^2$$

$$x - u \in \lambda\partial(|u|)$$

# BASIS PURSUIT DENOISING PROBLEM

(PROXIMAL OPERATOR)

$$\mathbf{prox}_{\lambda|\cdot|}(x) = \underset{u}{\text{argmin}} \quad \lambda|u| + \frac{1}{2}|u-x|^2$$

$$x - u \in \lambda\partial(|u|)$$

$(u \neq 0)$:

$$\begin{aligned} x &= \lambda \text{sgn}(u) + u \\ \Rightarrow \quad \text{sgn}(x) &= \text{sgn}(u) \end{aligned}$$

# BASIS PURSUIT DENOISING PROBLEM

(PROXIMAL OPERATOR)

$$\mathbf{prox}_{\lambda|\cdot|}(x) = \underset{u}{\text{argmin}} \quad \lambda|u| + \frac{1}{2}|u-x|^2$$

$$x - u \in \lambda\partial(|u|)$$

$(u \neq 0)$:

$$\begin{aligned} x &= \lambda\text{sgn}(u) + u \\ \Rightarrow \quad \text{sgn}(x) &= \text{sgn}(u) \end{aligned}$$

$$\begin{aligned} u &= x - \lambda\text{sgn}(x) \\ \Rightarrow \quad u &= \text{sgn}(x) \cdot \left[|x| - \lambda\right] \end{aligned}$$

# BASIS PURSUIT DENOISING PROBLEM

(PROXIMAL OPERATOR)

$$\mathbf{prox}_{\lambda|\cdot|}(x) = \underset{u}{\operatorname{argmin}} \quad \lambda|u| + \frac{1}{2}|u-x|^2$$

$$x - u \in \lambda\partial(|u|)$$

$(u \neq 0)$:

$$\begin{aligned} x &= \lambda \mathrm{sgn}(u) + u \\ \Rightarrow \quad \mathrm{sgn}(x) &= \mathrm{sgn}(u) \end{aligned}$$

$$\begin{aligned} u &= x - \lambda \mathrm{sgn}(x) \\ \Rightarrow \quad u &= \mathrm{sgn}(x) \cdot \big[|x| - \lambda\big] \end{aligned}$$

$(u = 0)$:

$$\begin{aligned} x \in \lambda\partial(|u|) &= [-\lambda, \lambda] \\ \therefore \quad \text{if } x \in [-\lambda, \lambda] &\Rightarrow u = 0 \end{aligned}$$

# BASIS PURSUIT DENOISING PROBLEM

(PROXIMAL OPERATOR)

$$\left[\mathbf{prox}_{\lambda|\cdot|}(x)\right]_i = \operatorname{sgn}(x_i) \cdot \left[|x_i| - \lambda\right]_+$$

# BASIS PURSUIT DENOISING PROBLEM

(ACTIVE SET)

$$\mathcal{A}(x) = \{i : \partial g_i(x_i) \text{ is } \textit{not} \text{ a singleton}\}$$

# BASIS PURSUIT DENOISING PROBLEM

(ACTIVE SET)

$$\mathcal{A}(x) = \{i : \partial g_i(x_i) \text{ is } \textit{not} \text{ a singleton}\}$$

$$\mathcal{A}(x) = \{i : x_i = 0\}$$

# BASIS PURSUIT DENOISING PROBLEM

( BINDING SET )

$$\mathcal{B}(x) = \left\{ i : i \in \mathcal{A}(x) \text{ and } [-\nabla f(x)]_i \in \partial g_i(x_i) \right\}$$

# BASIS PURSUIT DENOISING PROBLEM

( BINDING SET )

$$\mathcal{B}(x) = \{ i : i \in \mathcal{A}(x) \text{ and } [-\nabla f(x)]_i \in \partial g_i(x_i) \}$$

$$\mathcal{B}(x) = \{ i : x_i = 0 \text{ and } [-\nabla f(x)]_i \in [-\lambda, \lambda] \}$$

# Basis Pursuit Denoising Problem

(Reduced subproblem)

$$\underset{w}{\text{minimize}} \quad \frac{1}{2} w^\top H_k w + r_k^\top w \equiv F_k(w),$$

$$\text{where} \quad r_k = Z_k^\top \nabla f(x_k) + \text{sgn}(Z_k^\top x_k)$$

# BASIS PURSUIT DENOISING PROBLEM

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|Ax-b\|_2^2 + \lambda\|x\|_1 \equiv F(x)$$

**Proximal operator:**

$$\left[\mathbf{prox}_{\lambda|\cdot|}(x)\right]_i = \text{sgn}(x_i)\cdot\left[|x_i|-\lambda\right]_+$$

**Active set:**

$$\mathcal{A}(x) = \{i : x_i = 0\}$$

**Binding set:**

$$\mathcal{B}(x) = \left\{i : x_i = 0 \text{ and } [-\nabla f(x)]_i \in [-\lambda,\lambda]\right\}$$

**Reduced subproblem:**

$$\underset{w}{\text{minimize}} \quad \frac{1}{2}w^\top H_k w + r_k^\top w \equiv F_k(w),$$

where $r_k = Z_k^\top \nabla f(x_k) + \text{sgn}(Z_k^\top x_k)$.

# Numerical Examples

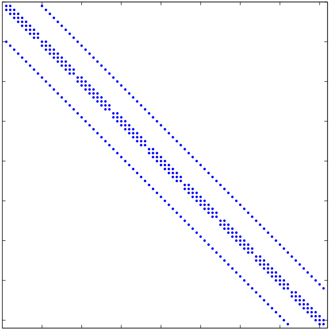
# PROBLEM SETTING

The basis pursuit denoising problem:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|Ax-b\|_2^2 + \lambda\|x\|_1 \equiv F(x)$$

# THEORETICAL EXAMPLE

Consider the finite-difference approximation of the Laplacian operator (discretized on a 2D grid):



$n = 10000$

condition number: 4.1336e+3

# THEORETICAL EXAMPLE

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}x^\top Hx + b^\top x + 0.3\|x\|_1$$

Matrix-vector products:

| $\tau_{\text{opt}}$ | $10^{-2}$ | $10^{-4}$ | $10^{-6}$ |
|---|---|---|---|
| PG | 193 | 795 | 1498 |
| `pgcg` | 76 | 133 | 170 |

# Seismic Data Interpolation

Find sparse representation of $y$ in curvelet operator $C$:

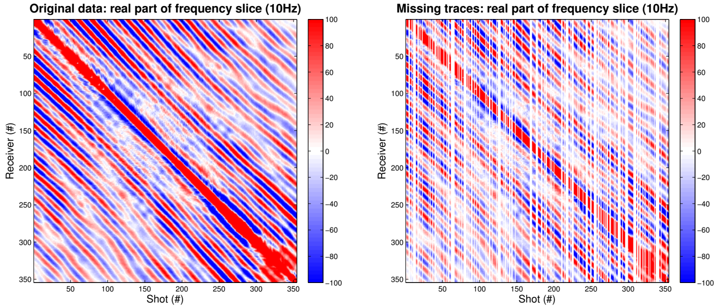
$$\underset{x\in\mathbb{R}^m}{\text{minimize}} \quad \frac{1}{2}\|RMC^\top x - y\|_2^2 + \lambda\|x\|_1,$$

where

- $RM$ is a restriction matrix operator
  - restricts to 60% original data set
- $y$ is the vectorized restricted data

# Seismic Data Interpolation

- Frequency slice (10 Hz) for sequential source acquisition from Gulf of Suez
  - e.g., [ Kumar, Aravkin, and Herrmann, 2012 ]



# SEISMIC DATA INTERPOLATION

Compare to SPGL1, which solves

$$\underset{x\in\mathbb{R}^m}{\text{minimize}} \quad \|x\|_1 \quad \text{s.t. } \|RMC^\top x - y\|_2 \leq \sigma. \tag{1}$$

# SEISMIC DATA INTERPOLATION

Compare to SPGL1, which solves

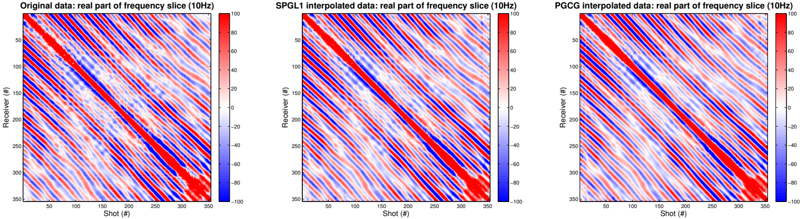
$$\underset{x\in\mathbb{R}^m}{\text{minimize}} \quad \|x\|_1 \quad \text{s.t. } \|RMC^\top x - y\|_2 \leq \sigma. \tag{1}$$

To ensure a valid comparison, we choose regularization parameter

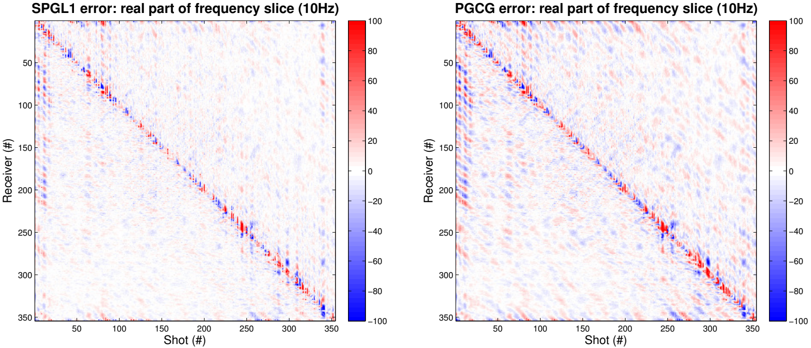
$$\lambda = \|C(RM)^\top r_\sigma\|_\infty,$$

where $r_\sigma$ is the residual corresponding to the solution of (1) found by SPGL1.

# SEISMIC DATA INTERPOLATION



# SEISMIC DATA INTERPOLATION




| Matrix-vector products | | |
|---|---|---|
| | $RMC^\top$ | $(RMC^\top)^\top$ |
| spgl1 | 139 | 102 |
| pgcg | 147 | 148 |

# FUTURE WORK

- A formal proof of convergence
- Using alternative methods for Phase 1
  - e.g., FISTA [Beck and Teboulle, '09]
- Alternative exit conditions
  - e.g., inside CG
- Continuation on $\lambda$, where $\lambda_k \to \lambda$
- Other interesting $g$
  - e.g., $g(z) = \|z\|_1$, where $z$ is complex valued

# FUTURE WORK

- A formal proof of convergence
- Using alternative methods for Phase 1
  - e.g., FISTA [Beck and Teboulle, '09]
- Alternative exit conditions
  - e.g., inside CG
- Continuation on $\lambda$, where $\lambda_k \rightarrow \lambda$
- Other interesting $g$
  - e.g., $g(z) = \|z\|_1$, where $z$ is complex valued

# FUTURE WORK

- A formal proof of convergence
- Using alternative methods for Phase 1
  - e.g., FISTA [Beck and Teboulle, '09]
- Alternative exit conditions
  - e.g., inside CG
- Continuation on $\lambda$, where $\lambda_k \to \lambda$
- Other interesting $g$
  - e.g., $g(z) = \|z\|_1$, where $z$ is complex valued

# FUTURE WORK

- A formal proof of convergence
- Using alternative methods for Phase 1
  - e.g., FISTA [Beck and Teboulle, '09]
- Alternative exit conditions
  - e.g., inside CG
- Continuation on $\lambda$, where $\lambda_k \to \lambda$
- Other interesting $g$
  - e.g., $g(z) = \|z\|_1$, where $z$ is complex valued

# FUTURE WORK

- A formal proof of convergence
- Using alternative methods for Phase 1
  - e.g., FISTA [Beck and Teboulle, '09]
- Alternative exit conditions
  - e.g., inside CG
- Continuation on $\lambda$, where $\lambda_k \to \lambda$
- Other interesting $g$
  - e.g., $g(z) = \|z\|_1$, where $z$ is complex valued

# FUTURE WORK

- A formal proof of convergence
- Using alternative methods for Phase 1
  - e.g., FISTA [Beck and Teboulle, '09]
- Alternative exit conditions
  - e.g., inside CG
- Continuation on $\lambda$, where $\lambda_k \to \lambda$
- Other interesting $g$
  - e.g., $g(z) = \|z\|_1$, where $z$ is complex valued

# FUTURE WORK

- A formal proof of convergence
- Using alternative methods for Phase 1
  - e.g., FISTA [Beck and Teboulle, '09]
- Alternative exit conditions
  - e.g., inside CG
- Continuation on $\lambda$, where $\lambda_k \to \lambda$
- Other interesting $g$
  - e.g., $g(z) = \|z\|_1$, where $z$ is complex valued

# FUTURE WORK

- A formal proof of convergence
- Using alternative methods for Phase 1
  - e.g., FISTA [Beck and Teboulle, '09]
- Alternative exit conditions
  - e.g., inside CG
- Continuation on $\lambda$, where $\lambda_k \to \lambda$
- Other interesting $g$
  - e.g., $g(z) = \|z\|_1$, where $z$ is complex valued

THANK YOU!

# REFERENCES


- U. Ascher and C. Greif. A First Course in Numerical Methods. SIAM, 2011.
- A. Beck and M. Teboulle. *A fast iteration shrinkage-thresholding algorithm for linear inverse problems*. SIAM J. Img. Sci., 2(2009), 183-202.
- E. van den Berg and M. P. Friedlander. *SPGL1: A solver for large-scale sparse reconstruction*. http://www.cs.ubc.ca/labs/scl/spgl1, 2007.
- R. Kumar, A. Y. Aravkin and F. J. Herrmann. *Fast methods for rank minimization with applications in seismic-data interpolation*. Technical Report, 2012.
- J. J. Moré and G. Toraldo. *On the solution of large quadratic programming problems with bound constraints*. SIAM J. Opt. (1):93-113, 1991.
- N. Parikh and S. Boyd. *Proximal algorithms*. To appear in *Foundations and Trends in Optimization*, 2013.