Regularizing waveform inversion by projections onto convex sets

Bas Peters
Joint work with Brendan Smithyman

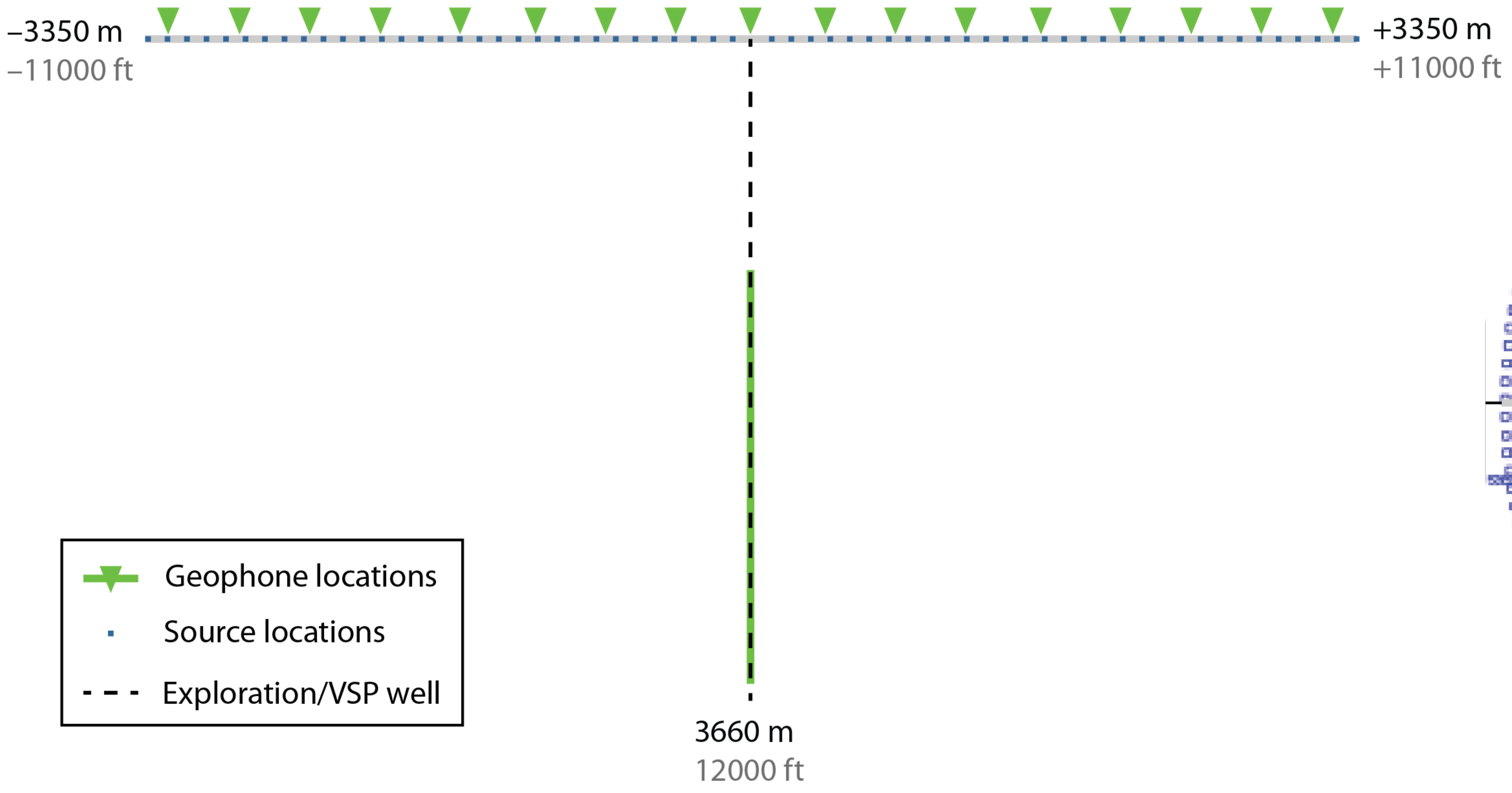
SINBAD Consortium Meeting Spring 2015, June 5

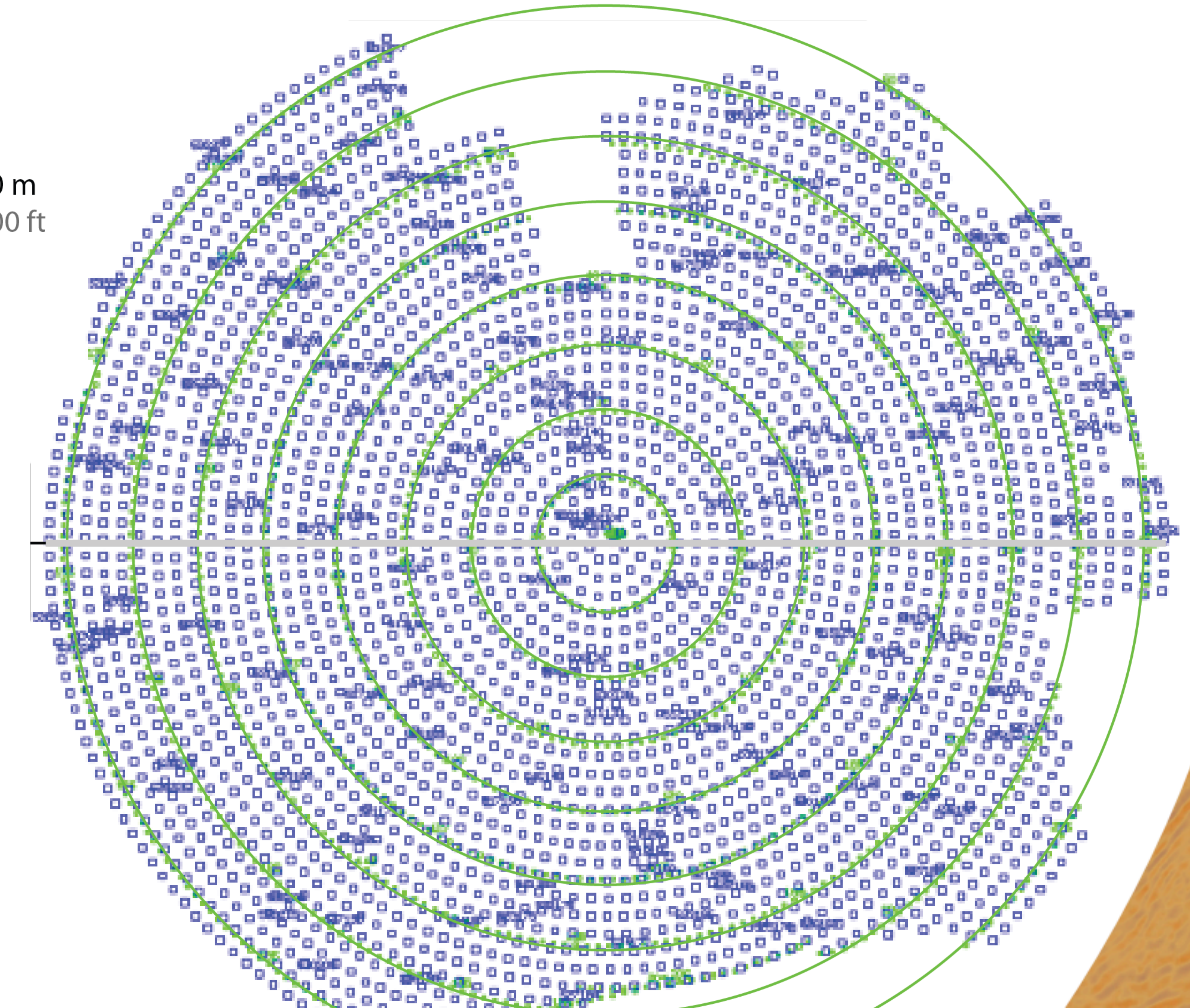
SLIM
University of British Columbia

Motivation

Land data set with surface sources and surface & well receivers

Constant density acoustic inversion

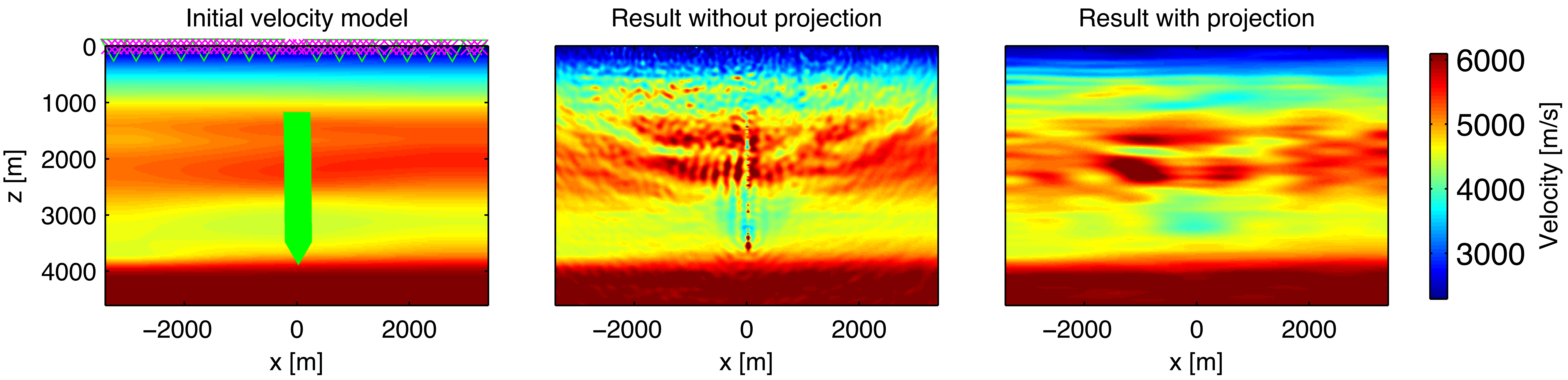




Motivation

Land data set with surface sources and surface & well receivers

Constant density acoustic inversion (2D slice)



Fore challenging problems, some regularization is required

A few regularization strategies

Objective function: $f(\mathbf{m})$ (differentiable, time or frequency)

Tikhonov / quadratic: $\phi(\mathbf{m}) = f(\mathbf{m}) + \frac{\alpha}{2}\|R_1\mathbf{m}\|^2 + \frac{\beta}{2}\|R_2\mathbf{m}\|^2$

Gradient filtering: $\mathbf{m}_{k+1} = \mathbf{m}_k - \gamma F \nabla_{\mathbf{m}} f(\mathbf{m})$

Constrained formulation: $\min_{\mathbf{m}} f(\mathbf{m}) \quad \text{s.t.} \quad \mathbf{m} \in \mathcal{C}_1 \bigcap \mathcal{C}_2$

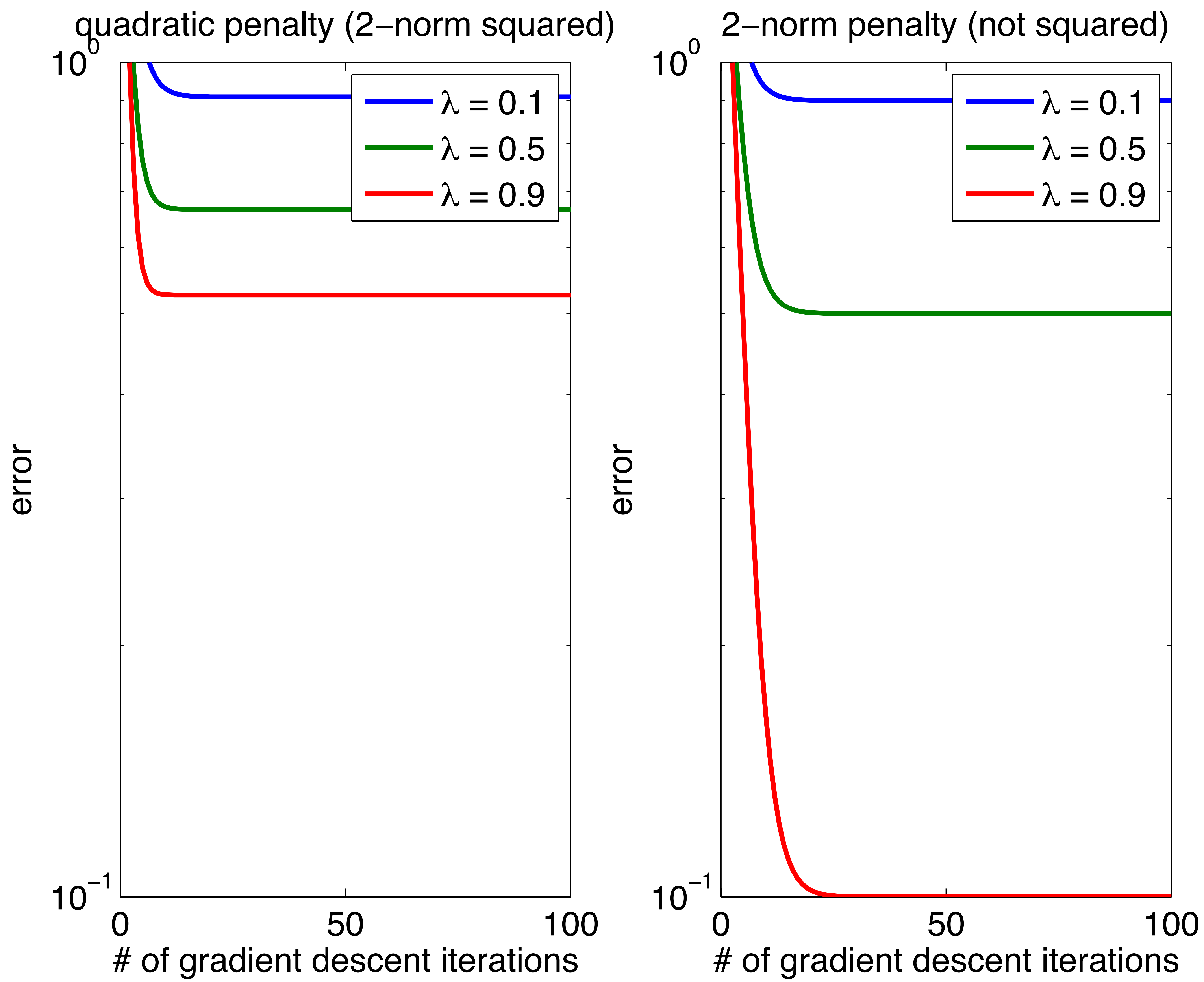
A few regularization strategies

Tikhonov / quadratic: $$\phi(\mathbf{m}) = f(\mathbf{m}) + \frac{\alpha}{2}\|R_1\mathbf{m}\|^2 + \frac{\beta}{2}\|R_2\mathbf{m}\|^2$$

Potential problems:

- squared norm is not an exact penalty
- difficult/costly to determine penalty-parameters
- potentially ill-conditioned Hessian
- may not be obvious which constrained problem is solved for a given penalty parameter

A few regularization strategies



exact versus non-exact penalty

Toy problem:

$$\min_x \frac{1}{2}\|x-1\|_2^2 \quad \text{s.t.} \quad x = 2$$

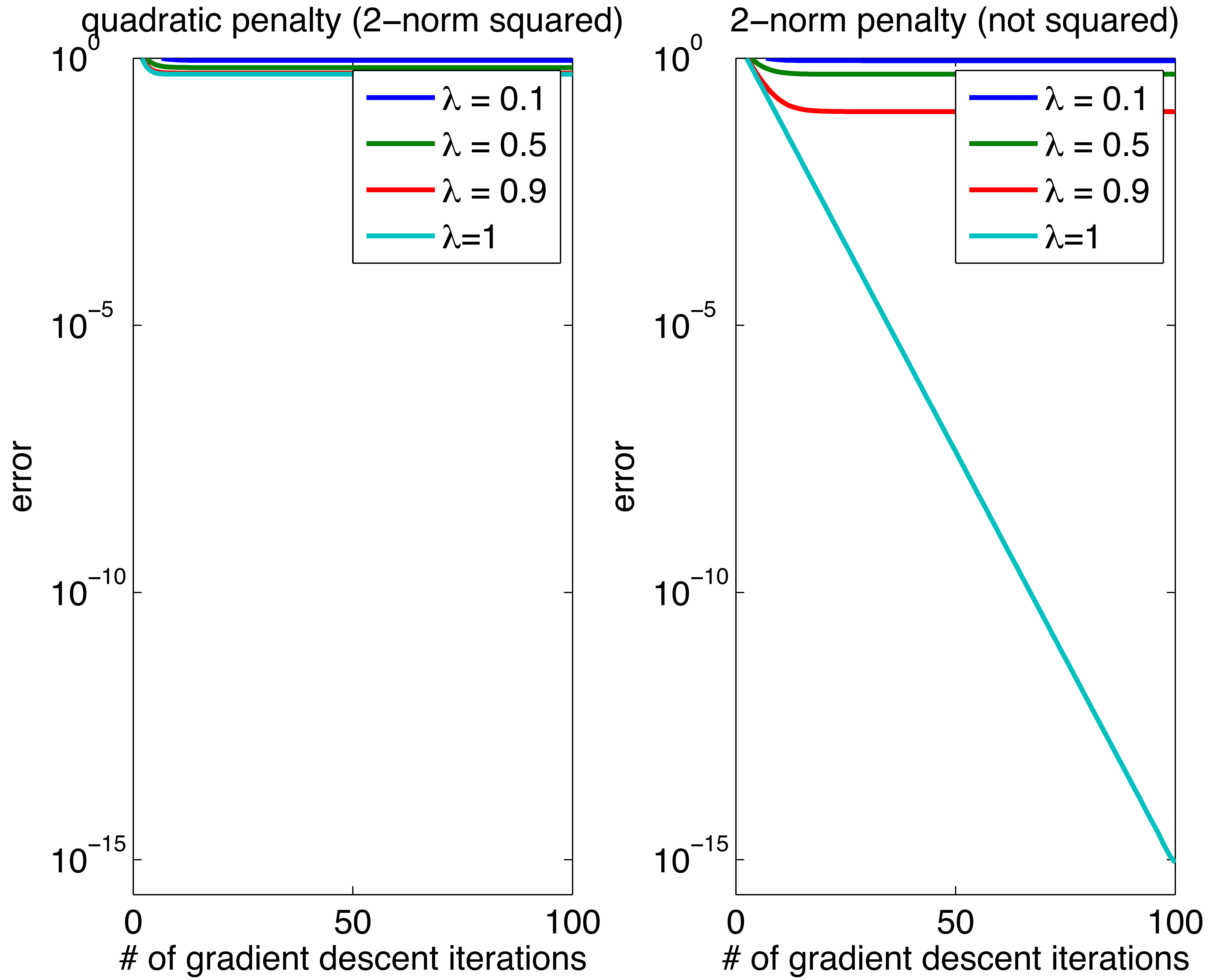
Quadratic-penalty:

$$\min_x \frac{1}{2}\|x-1\|_2^2 + \lambda\|x-2\|_2^2$$

2-norm penalty:

$$\min_x \frac{1}{2}\|x-1\|_2^2 + \lambda\|x-2\|_2$$

A few regularization strategies



exact versus non-exact penalty

Toy problem:

$$\min_x \frac{1}{2}\|x-1\|_2^2 \quad \text{s.t.} \quad x = 2$$

Quadratic-penalty:

$$\min_x \frac{1}{2}\|x-1\|_2^2 + \lambda\|x-2\|_2^2$$

2-norm penalty:

$$\min_x \frac{1}{2}\|x-1\|_2^2 + \lambda\|x-2\|_2$$

A few regularization strategies

Gradient filtering: $\mathbf{m}_{k+1} = \mathbf{m}_k - \gamma F \nabla_{\mathbf{m}} f(\mathbf{m})$

If the gradient filter F is the inverse Hessian, this is just Newton's method

Potential problems:

- filtered gradient is not a gradient of the objective anymore
- no obvious generalization to include multiple filters

A few regularization strategies

Constrained formulation: $$\min_{\mathbf{m}} f(\mathbf{m}) \quad \text{s.t.} \quad \mathbf{m} \in \mathcal{C}_1 \bigcap \mathcal{C}_2$$

"find a model which satisfies all pieces of prior info simultaneously"

- constraints can be satisfied at every iteration
- feasible part of the objective function is unmodified
- works with gradient/quasi-Newton/Newton-type methods
- can define more than two constraint-sets

- no weights or other parameters required, just define the sets

Prior information as convex sets

Projection (Euclidean, minimum-distance projection):

$$\mathcal{P}_{\mathcal{C}}(\mathbf{m}) = \underset{\mathbf{x}}{\arg\min} \, \|\mathbf{x} - \mathbf{m}\|_2 \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{C}_1 \bigcap \mathcal{C}_2$$

Important property:

$$\mathcal{P}_{\mathcal{C}}(\mathbf{m}) = \mathcal{P}_{\mathcal{C}}(\mathcal{P}_{\mathcal{C}}(\mathbf{m}))$$

Prior information as convex sets

example 1: (spatially varying) bound constraints:

$$\mathcal{C}_1 \equiv \{\mathbf{m} \mid \mathbf{b}_l \leq \mathbf{m} \leq \mathbf{b}_u\}$$

can include reference models as:

$$\mathbf{b}_l = \mathbf{m}_{\text{ref}} - \delta\mathbf{m}$$

Projector: (element-wise)

$$\mathcal{P}_{\mathcal{C}_1}(\mathbf{m}) = \text{median}\{\mathbf{b}_l, \mathbf{m}, \mathbf{b}_u\}$$

Prior information as convex sets

example 2: minimum smoothness of the model:

$$\mathcal{C}_2 \equiv \{\mathbf{m} \mid E^* F^* (I - S) F E \mathbf{m} = 0\}$$

“the 2D spatial Fourier-transform of the mirror-extended model is contained within an ellipse”

$E \in \mathbb{R}^{4N \times N}$ Mirror-extention

$\mathbf{m} \in \mathbb{R}^{N}$ medium parameters

$F \in \mathbb{C}^{N \times N}$ DFT matrix

$S \in \mathbb{R}^{N \times N}$ Selection matrix (diagonal), 'filter coefficients'

Prior information as convex sets

example 2: minimum smoothness of the model:

$$\mathcal{C}_2 \equiv \{\mathbf{m} \mid E^* F^* (I - S) F E \mathbf{m} = 0\}$$

1. 2D mirror extension of the model (to avoid periodic boundaries)
2. 2D DFT
3. Remove coefficients outside ellipse (highest spatial frequencies)
4. 2D inverse DFT

ellipse takes directional varying smoothness (geology) into account

Prior information as convex sets

example 2: minimum smoothness of the model:

$$\mathcal{C}_2 \equiv \{\mathbf{m} \mid E^* F^* (I - S) F E \mathbf{m} = 0\}$$

- Choose initial ellipse based on the lowest frequency band and the smoothness of the start model.
- Adapt to different frequency bands by stretching the ellipse based on a formula like: $d\dfrac{f_{\max}}{v_{\min}}$

Projector: $\mathcal{P}_{\mathcal{C}_2}(\mathbf{m}) = E^* F^* S F E \mathbf{m}$

Algorithmic development

$$\min_{\mathbf{m}} f(\mathbf{m}) \quad \text{s.t.} \quad \mathbf{m} \in \mathcal{C}_1 \bigcap \mathcal{C}_2$$

$\mathcal{C}_1 \bigcap \mathcal{C}_2$ is convex if $\mathcal{C}_1$ and $\mathcal{C}_2$ are convex

We would like the model to be in $\mathcal{C}_1 \bigcap \mathcal{C}_2$ at every iteration

One possibility:

$$\min_{\mathbf{m}} f(\mathbf{m}) + \iota_{\mathcal{C}_1}(\mathbf{m}) + \iota_{\mathcal{C}_2}(\mathbf{m})$$

$$\iota_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{C}, \\ +\infty & \text{if } x \notin \mathcal{C}. \end{cases}$$

Algorithmic development

$$\min_{\mathbf{m}} f(\mathbf{m}) \quad \text{s.t.} \quad \mathbf{m} \in \mathcal{C}_1 \bigcap \mathcal{C}_2$$

$$\min_{\mathbf{m}} f(\mathbf{m}) + \iota_{\mathcal{C}_1}(\mathbf{m}) + \iota_{\mathcal{C}_2}(\mathbf{m})$$ $\longrightarrow$ not differentiable

Can use forward-backward splitting / proximal-gradient algorithms.

Algorithmic development

$$\min_{\mathbf{m}} f(\mathbf{m}) \quad \text{s.t.} \quad \mathbf{m} \in \mathcal{C}_1 \bigcap \mathcal{C}_2$$

Project onto an intersection of convex sets:

- sometimes known analytically
- otherwise compute numerically; Dykstra's algorithm is used in this work

Dykstra splitting

Toy example:
find projection onto intersection of a circle and a square

Algorithm 1 Dykstra.

$x_0 = \mathbf{m}$, $p_0 = 0$, $q_0 = 0$

For $k = 0, 1, \ldots$

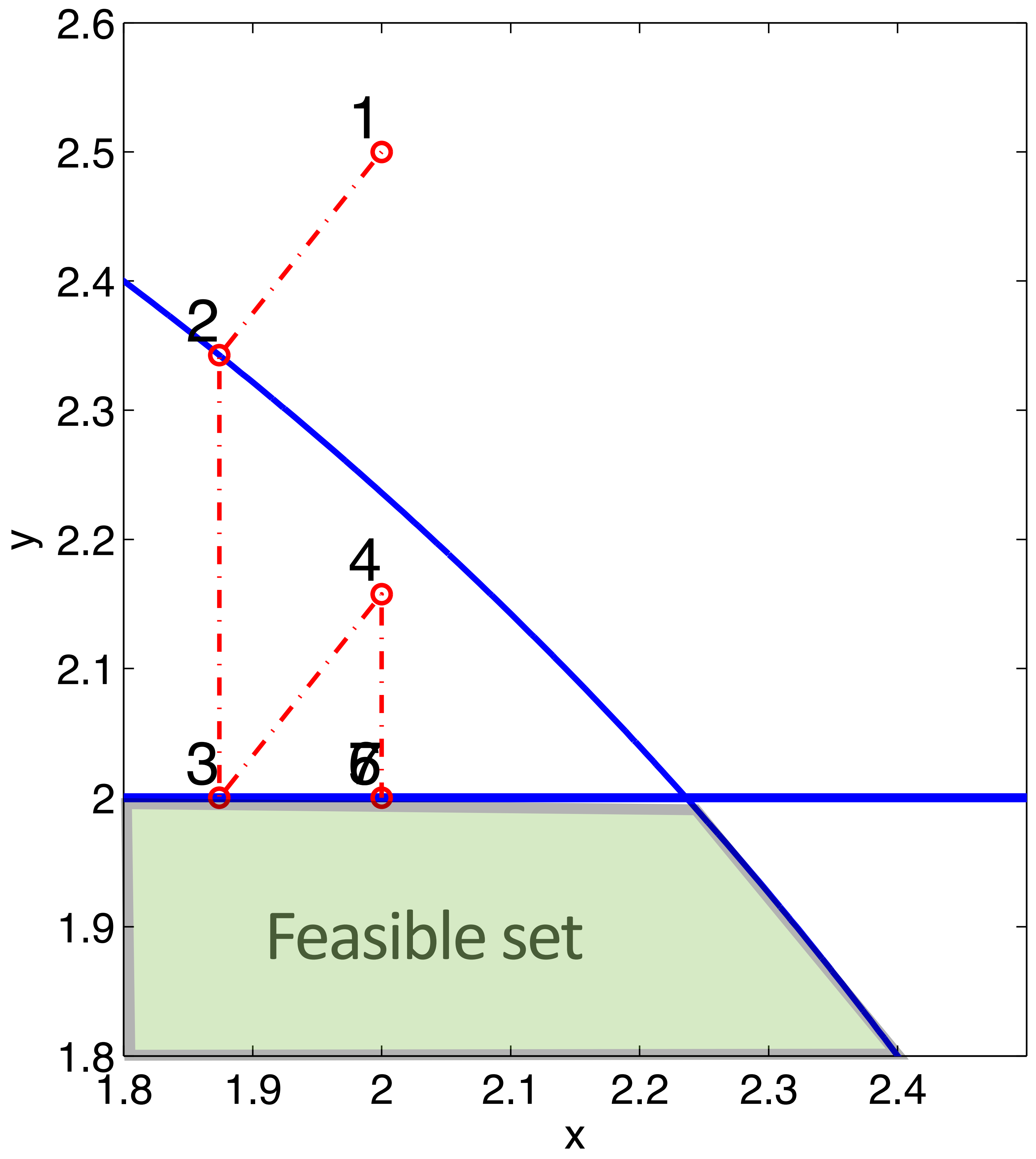
$$y_k = \mathscr{P}_{C_1}(x_k + p_k)$$

$$p_{k+1} = x_k + p_k - y_k$$

$$x_{k+1} = \mathscr{P}_{C_2}(y_k + q_k)$$

$$q_{k+1} = y_k + q_k - x_{k+1}$$

End



Dykstra splitting

Toy example:
find projection onto intersection of a circle and a square

Algorithm 1 Dykstra.

$x_0 = \mathbf{m}$, $p_0 = 0$, $q_0 = 0$

For $k = 0, 1, \ldots$

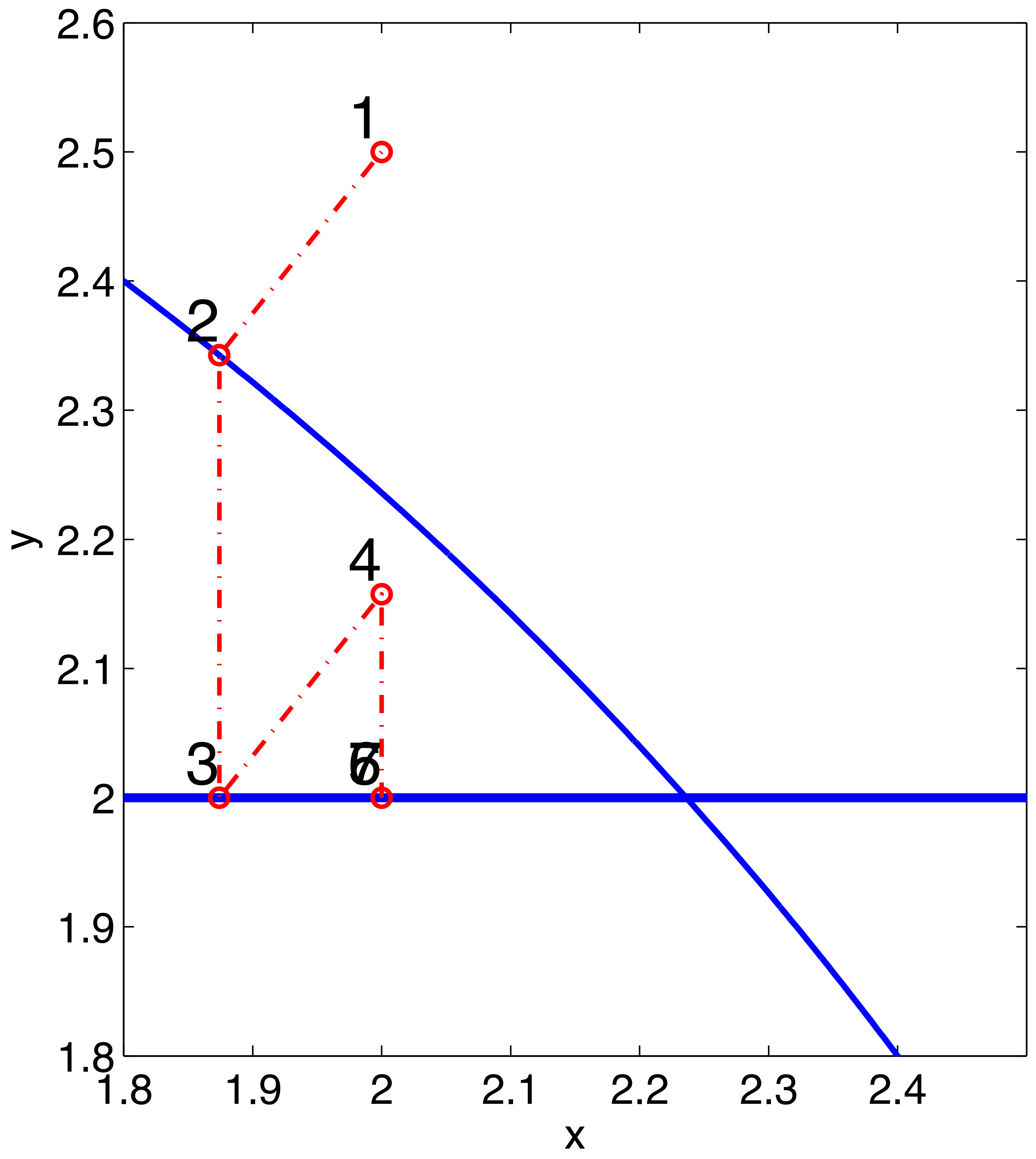
$\longrightarrow \boxed{y_k = \mathscr{P}_{C_1}(x_k + p_k)}$

$p_{k+1} = x_k + p_k - y_k$

$\longrightarrow \boxed{x_{k+1} = \mathscr{P}_{C_2}(y_k + q_k)}$

$q_{k+1} = y_k + q_k - x_{k+1}$

End



only need projection onto each set separately

Dykstra splitting

Toy example:
find projection onto intersection of a circle and a square

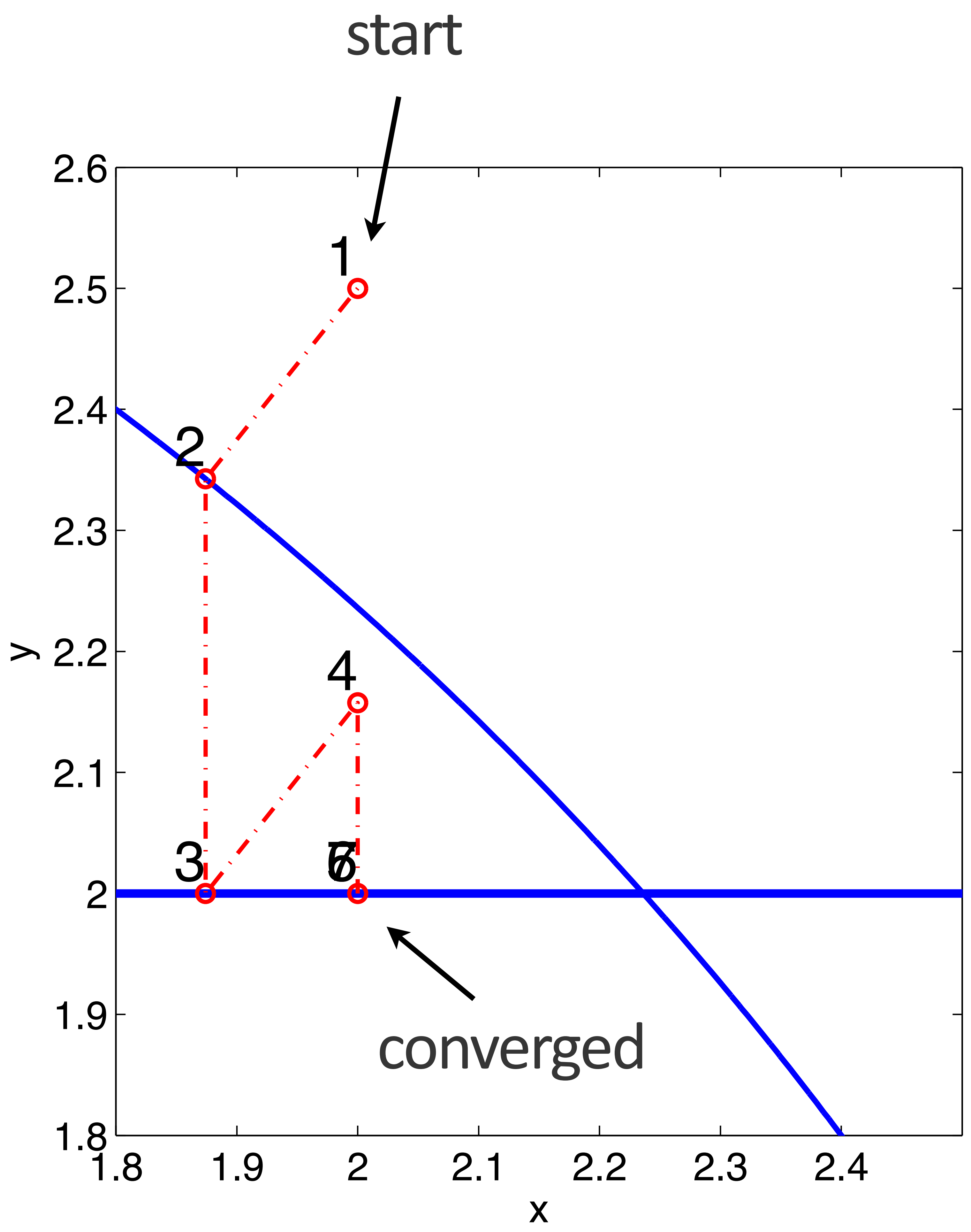
Algorithm 1 Dykstra.

$x_0 = \mathbf{m}$, $p_0 = 0$, $q_0 = 0$

For $k = 0, 1, \ldots$

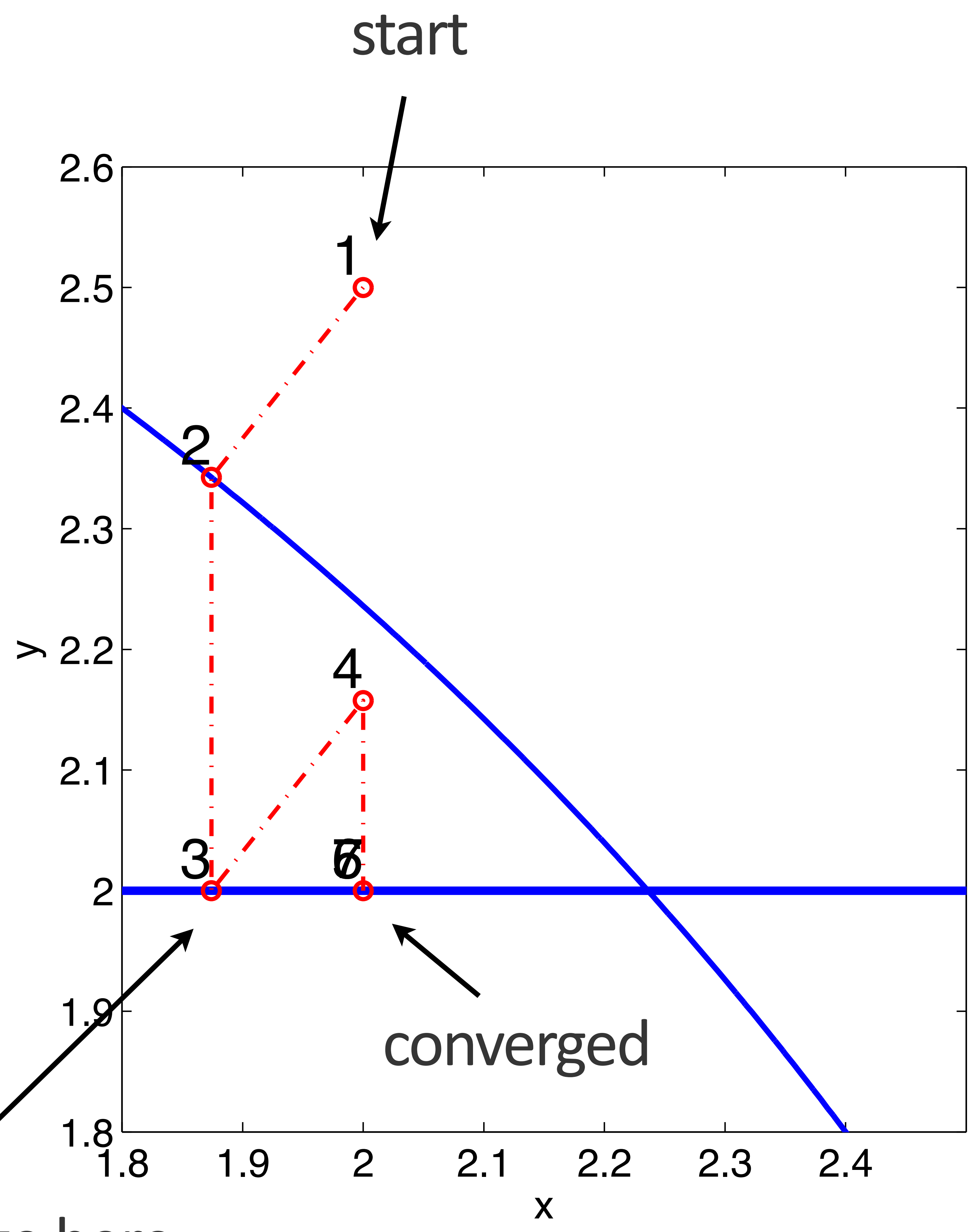
$$y_k = \mathscr{P}_{C_1}(x_k + p_k)$$
$$p_{k+1} = x_k + p_k - y_k$$
$$x_{k+1} = \mathscr{P}_{C_2}(y_k + q_k)$$
$$q_{k+1} = y_k + q_k - x_{k+1}$$

End



Dykstra splitting

Toy example:
find projection onto intersection of a circle and a square



POCS would converge here,
feasible point, not the projection onto

Dykstra splitting

Projection-onto-convex-sets (POCS) solves the convex feasibility problem:

$$\text{find} \quad x \in \mathcal{C}_1 \bigcap \mathcal{C}_2$$

Dykstra's algorithm solves:

$$\min_x \iota_{\mathcal{C}_1}(x) + \iota_{\mathcal{C}_2}(x) + \frac{1}{2}\|x - y\|^2$$

with indicator function:

$$\iota_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{C}, \\ +\infty & \text{if } x \notin \mathcal{C}. \end{cases}$$

Dykstra splitting

Projection-onto-convex-sets (POCS) solves the convex feasibility problem:

$$\text{find} \quad x \in \mathcal{C}_1 \bigcap \mathcal{C}_2$$

Dykstra's algorithm solves:

$$\min_x \iota_{\mathcal{C}_1}(x) + \iota_{\mathcal{C}_2}(x) + \frac{1}{2}\|x - y\|^2$$

is equivalent to:

$$\min_x \frac{1}{2}\|x - y\|^2 \quad \text{s.t.} \quad x \in \mathcal{C}_1 \bigcap \mathcal{C}_2$$

Dykstra splitting

Projection-onto-convex-sets (POCS):

$$\text{find} \quad x \in \mathcal{C}_1 \bigcap \mathcal{C}_2$$

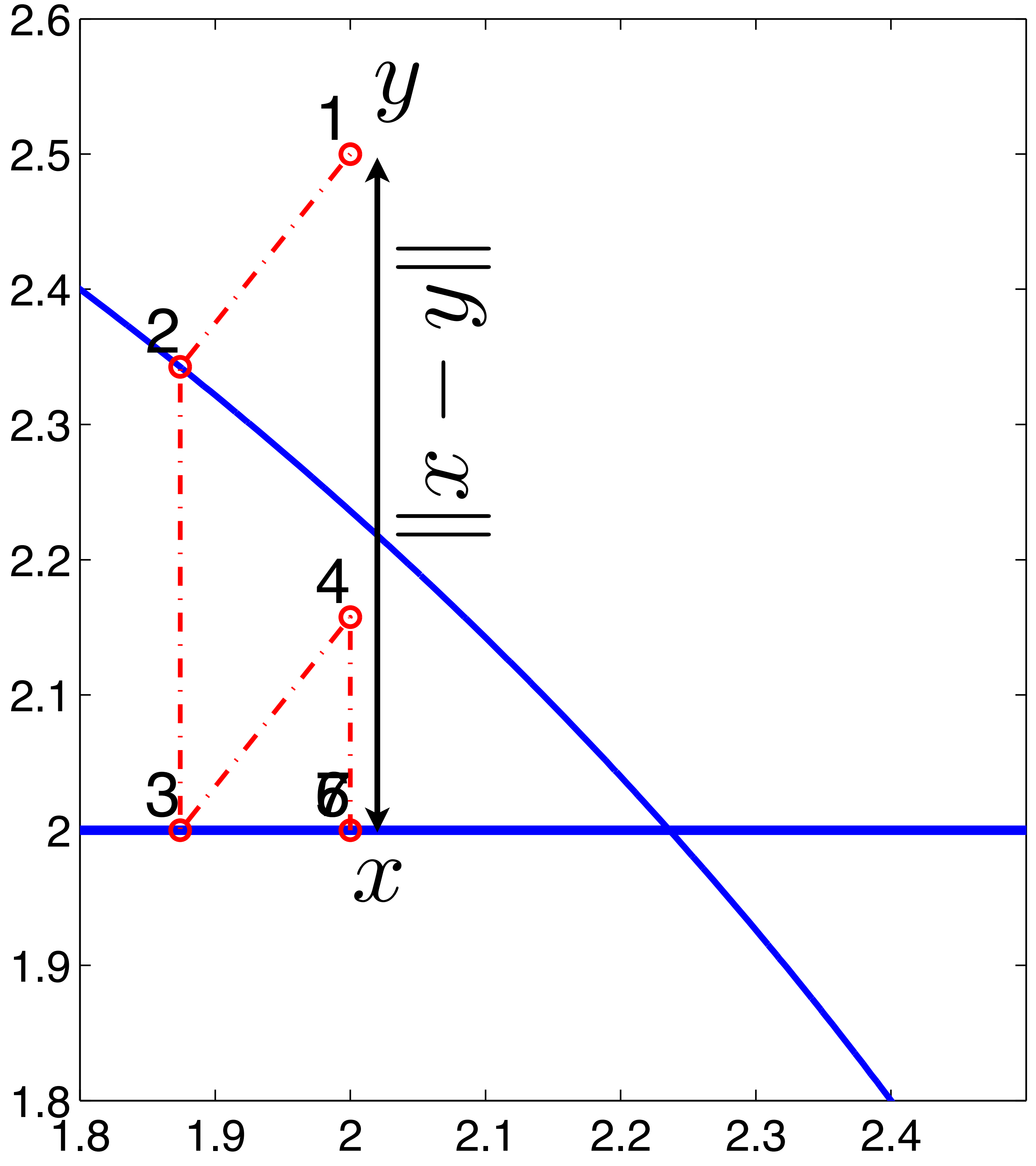
find any point in the intersection,
may be the closest point

Dykstra's algorithm solves:

$$\min_x \iota_{\mathcal{C}_1}(x) + \iota_{\mathcal{C}_2}(x) + \frac{1}{2}\|x-y\|^2$$

is equivalent to:

$$\min_x \frac{1}{2}\|x-y\|^2 \quad \text{s.t.} \quad x \in \mathcal{C}_1 \bigcap \mathcal{C}_2$$



Algorithmic development

$$\min_{\mathbf{m}} f(\mathbf{m}) \quad \text{s.t.} \quad \mathbf{m} \in \mathcal{C}_1 \bigcap \mathcal{C}_2$$

Projected-gradient: $\mathbf{m}_{k+1} = \mathcal{P}_{\mathcal{C}}(\mathbf{m}_k - \gamma \nabla_{\mathbf{m}} f(\mathbf{m}_k))$

SLIM

Algorithmic development

$$\min_{\mathbf{m}} f(\mathbf{m}) \quad \text{s.t.} \quad \mathbf{m} \in \mathcal{C}_1 \bigcap \mathcal{C}_2$$

Projected-gradient: $\mathbf{m}_{k+1} = \mathcal{P}_{\mathcal{C}}(\mathbf{m}_k - \gamma \nabla_{\mathbf{m}} f(\mathbf{m}_k))$

Can this simply be accelerated using Hessian approximation $B(\mathbf{m}_k)$?

$$\mathbf{m}_{k+1} = \mathcal{P}_{\mathcal{C}}(\mathbf{m}_k - \gamma B(\mathbf{m}_k)^{-1} \nabla_{\mathbf{m}} f(\mathbf{m}_k))$$

Algorithmic development

$$\min_{\mathbf{m}} f(\mathbf{m}) \quad \text{s.t.} \quad \mathbf{m} \in \mathcal{C}_1 \bigcap \mathcal{C}_2$$

Projected-gradient: $\mathbf{m}_{k+1} = \mathcal{P}_{\mathcal{C}}(\mathbf{m}_k - \gamma \nabla_{\mathbf{m}} f(\mathbf{m}_k))$

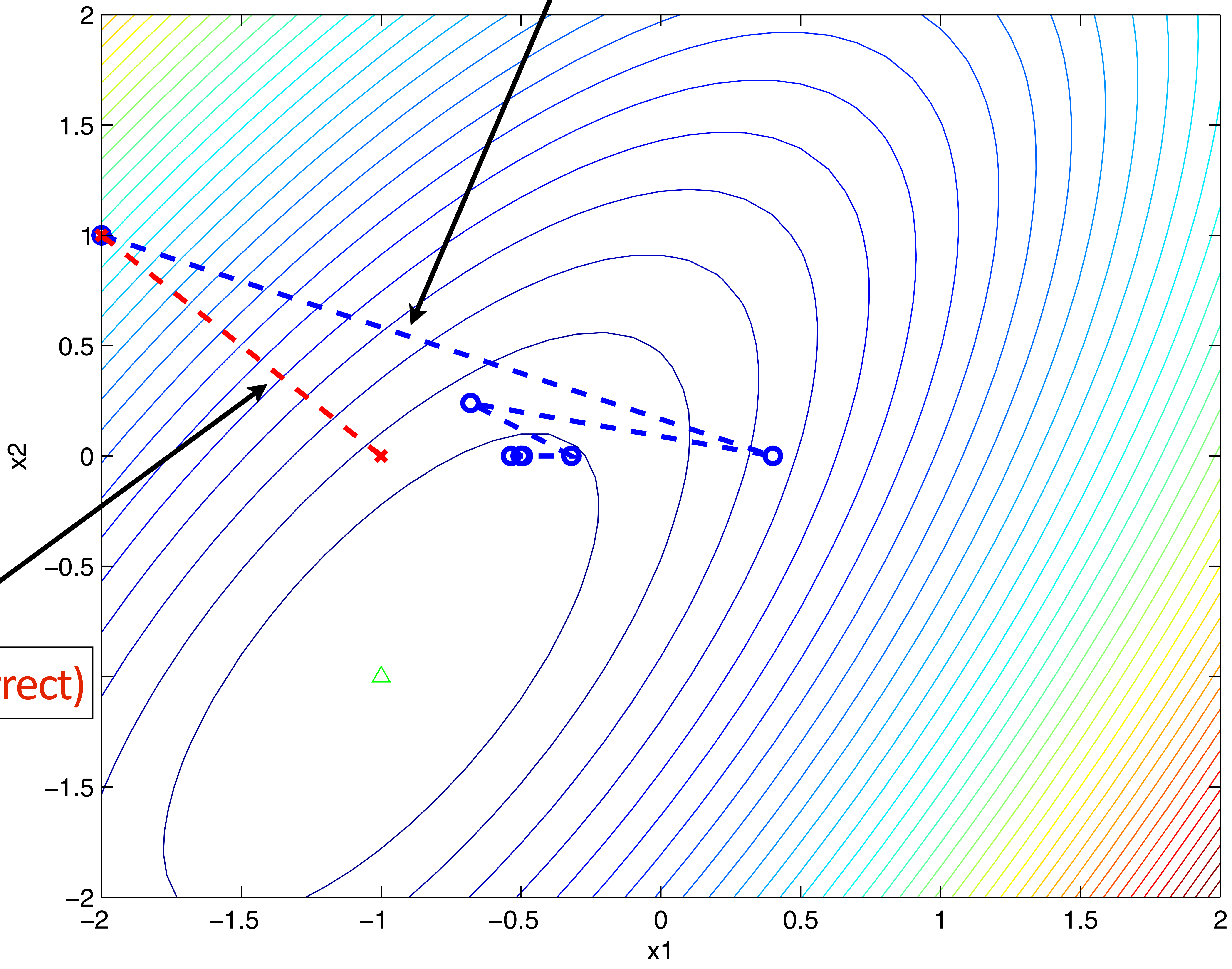
Can this simply be accelerated using Hessian approximation $B(\mathbf{m}_k)$?

$$\xcancel{\mathbf{m}_{k+1} = \mathcal{P}_{\mathcal{C}}(\mathbf{m}_k - \gamma B(\mathbf{m}_k)^{-1} \nabla_{\mathbf{m}} f(\mathbf{m}_k))}$$

Generally not, when using the Euclidean projection and general $B(\mathbf{m}_k)$

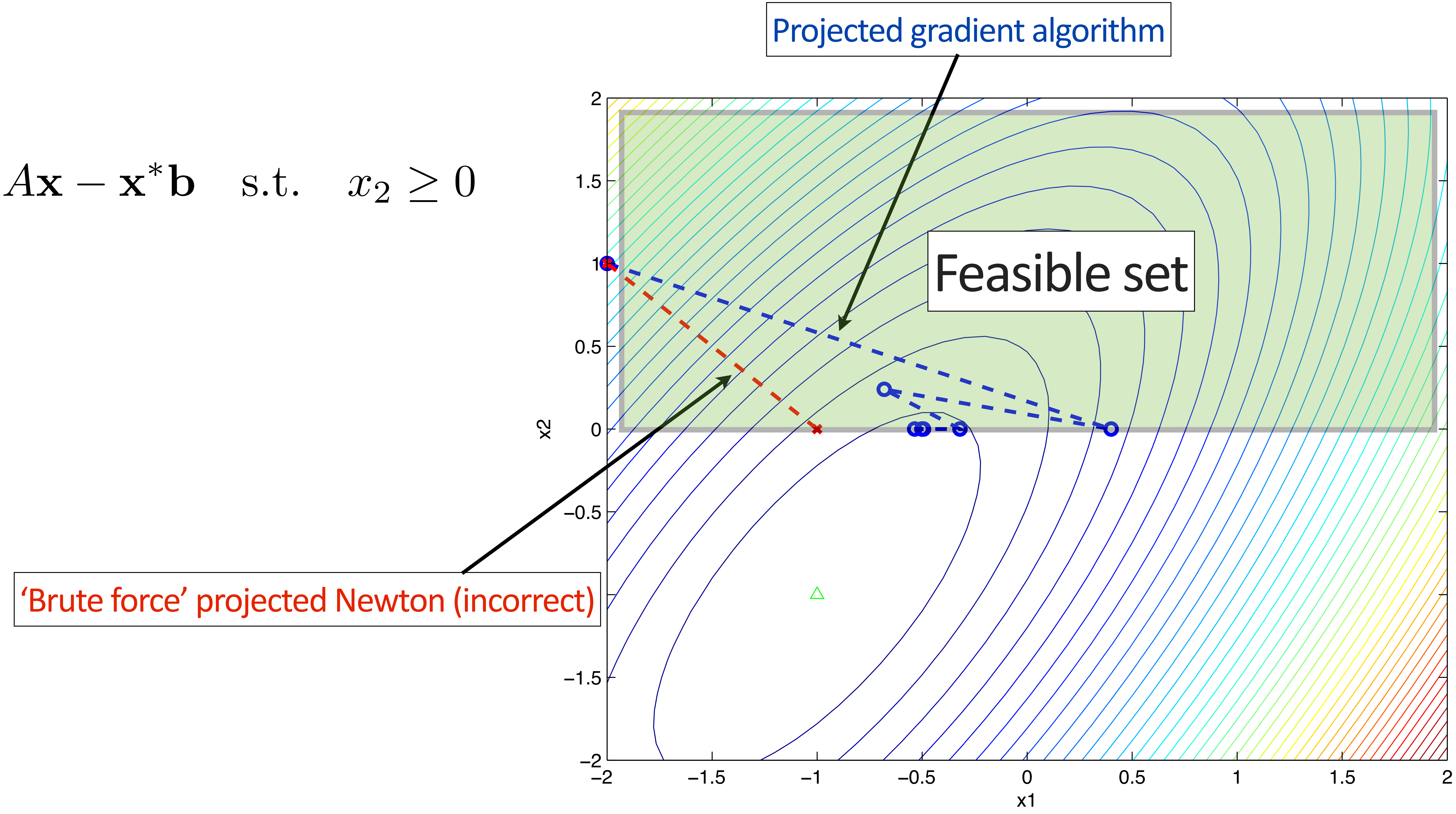
$$\min_{\mathbf{x}} \mathbf{x}^* A\mathbf{x} - \mathbf{x}^*\mathbf{b} \quad \text{s.t.} \quad x_2 \geq 0$$

Projected gradient algorithm

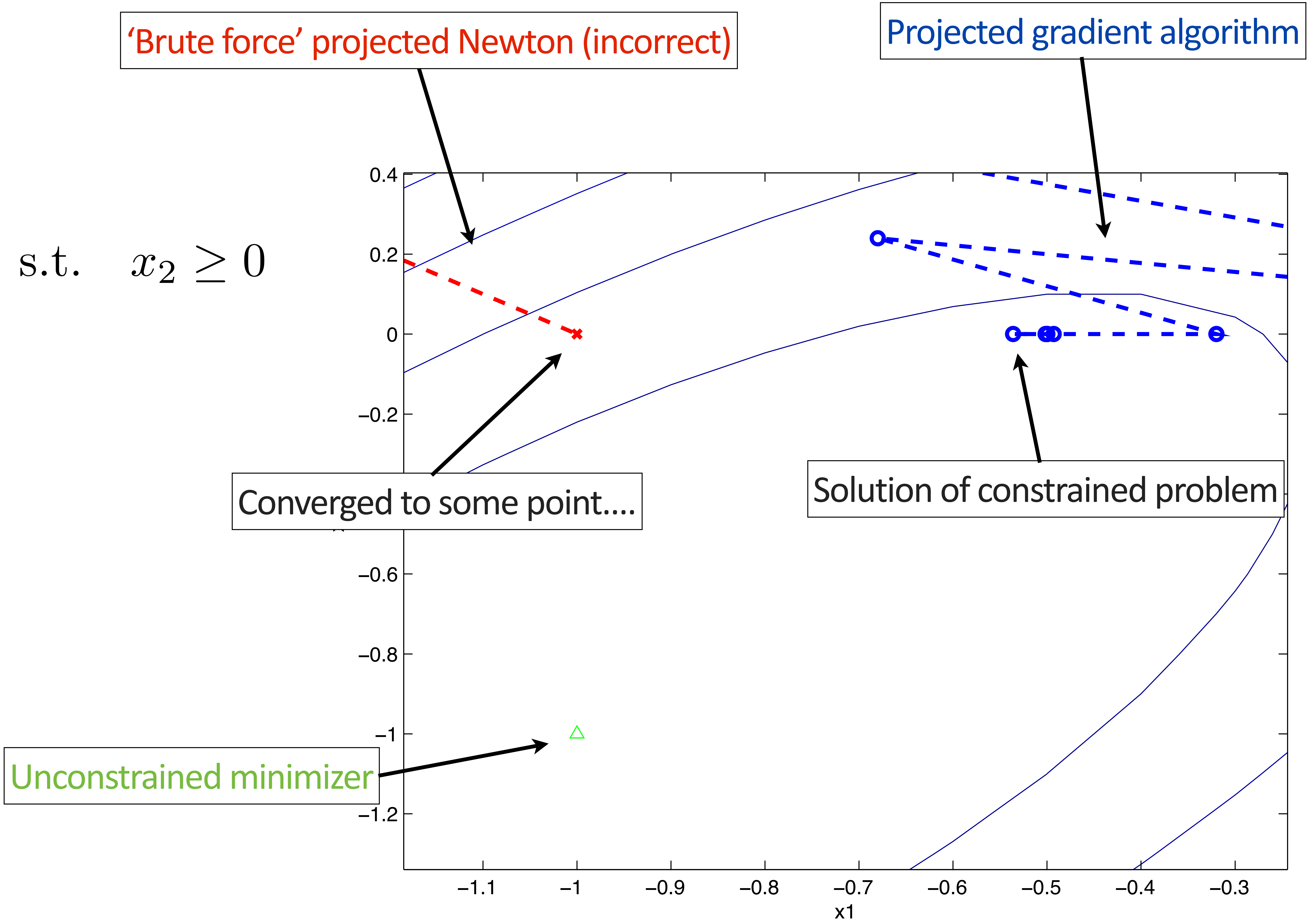


'Brute force' projected Newton (incorrect)

$$\min_{\mathbf{x}} \mathbf{x}^* A\mathbf{x} - \mathbf{x}^*\mathbf{b} \quad \text{s.t.} \quad x_2 \geq 0$$



$$\min_{\mathbf{x}} \mathbf{x}^* A\mathbf{x} - \mathbf{x}^*\mathbf{b} \quad \text{s.t.} \quad x_2 \geq 0$$



Algorithmic development

Projected-gradient: $\mathbf{m}_{k+1} = \mathcal{P}_{\mathcal{C}}(\mathbf{m}_k - \gamma \nabla_{\mathbf{m}} f(\mathbf{m}_k))$

Projected Quasi-Newton **[M. Schmidt et. al., 2009]**

- solves quadratic sub-problem with constraints using the spectral projected-gradient algorithm (inexactly)
- L-BFGS Hessian

Projected Newton-type:

- solves quadratic sub-problem with constraints
- efficient if approximate Hessian is ‘easy to invert’

Algorithmic development

Projected Newton-type:

- solves quadratic sub-problem with constraints:

$$Q(\mathbf{m}) = f(\mathbf{m}_k) + (\mathbf{m} - \mathbf{m}_k)^* \nabla_{\mathbf{m}} f(\mathbf{m}_k) + (\mathbf{m} - \mathbf{m}_k)^* B_k (\mathbf{m} - \mathbf{m}_k)$$

$$\mathbf{m}_{k+1} = \min_{\mathbf{m} \in \mathcal{C}_1 \bigcap \mathcal{C}_2} Q(\mathbf{m})$$

- efficient if approximate Hessian is 'easy to invert' (factored Hessian, sparse & well conditioned, diagonal)

Multiple algorithms can solve the constrained sub-problem

We use Alternating Direction Method of Multipliers (ADMM)

Algorithmic development

Projected Newton-type:

- solves quadratic sub-problem with constraints:

$$\mathbf{m}_{k+1} = \min_{\mathbf{m}\in\mathcal{C}_1 \bigcap \mathcal{C}_2} Q(\mathbf{m})$$

- can be reformulated as: [M. Schmidt et. al., 2011]

$$\mathbf{y}_k = B_k^{-1}\nabla_{\mathbf{m}} f(\mathbf{m}_k)$$

(unconstrained Newton-step)

$$\mathbf{m}_{k+1} = \min_{\mathbf{m}\in\mathcal{C}_1 \bigcap \mathcal{C}_2} \frac{1}{2}\|\mathbf{y}_k - \mathbf{m}\|_{B_k}^2$$

(projection w.r.t. metric induced by the approximate Hessian)

Workflow summary

1. Define convex feasible sets, possibly velocity & frequency dependent
2. Set up Dykstra's algorithm for projection onto intersections of sets
3. Set up an algorithm to solve the quadratic sub-problem with constraints (ADMM)
4. Solve waveform inversion problem using the a Projected Newton-type algorithm

Algorithm

Projected quasi-Newton (PQN) version:

At every iteration of PQN:

- solve PDE's
- solve quadratic problem with constraints using SPG
- at every iteration of SPG:
  - solve projection problem onto an intersection of convex sets using Dykstra's algorithm
  - at every iteration of Dykstra's algorithm:
    - compute projections on each set separately

Algorithm

Projected Newton-type version:

At every iteration of projected Newton-type:

- solve PDE’s
- Solve quadratic problem with constraints using ADMM
- at every iteration of ADMM:
  - invert Hessian (possibly iteratively)
  - solve projection problem onto an intersection of convex sets using Dykstra’s algorithm
  - at every iteration of Dykstra’s algorithm:
    - compute projections on each set separately

Example 1 - FWI with a lot of noise

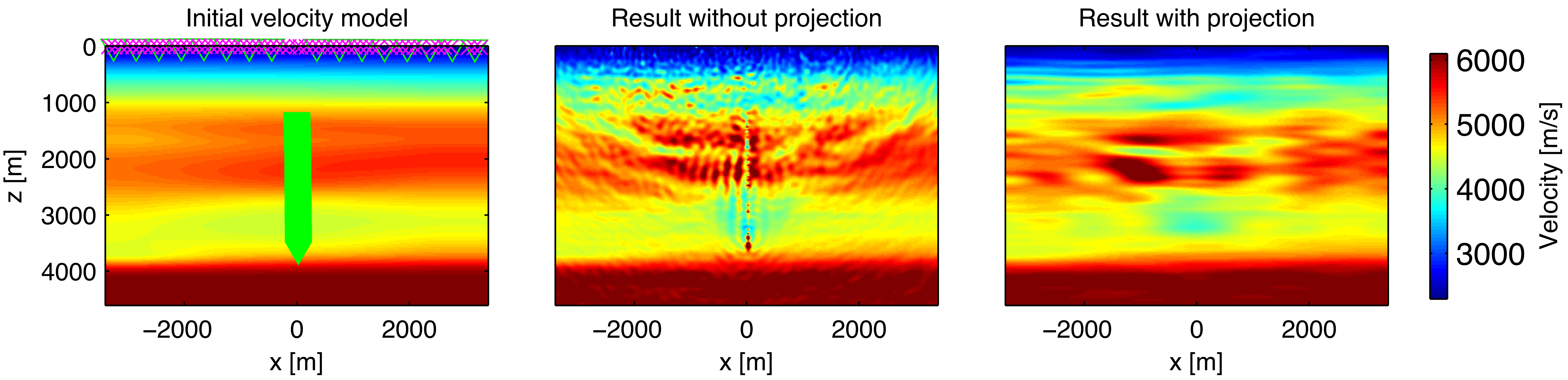
- Sources and receivers at top & bottom of the domain
- 10 Hz data
- $\|\text{noise}\|_2 / \|\text{signal}\|_2 = 0.3$
- used bound constraints and minimum smoothness constraints

acts as an image-domain noise filter in this case

Example 2 - FWI on a real land dataset

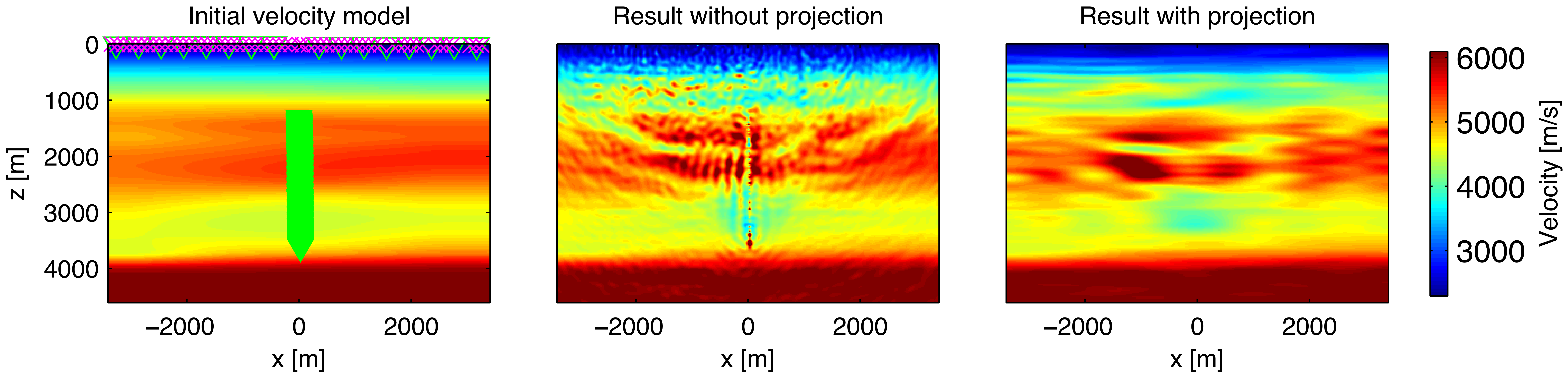
Land data set with surface sources and surface & well receivers

Constant density acoustic inversion (2D slice)



Example 2 - FWI on a real land dataset

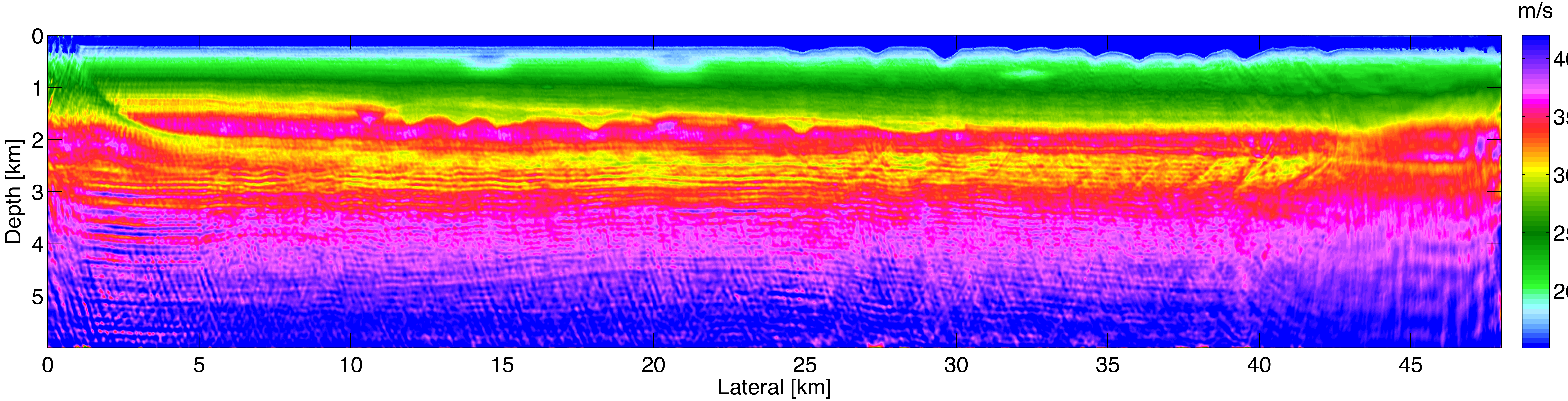
- Bound constraints
- Minimum smoothness constraints



Example 3 - WRI

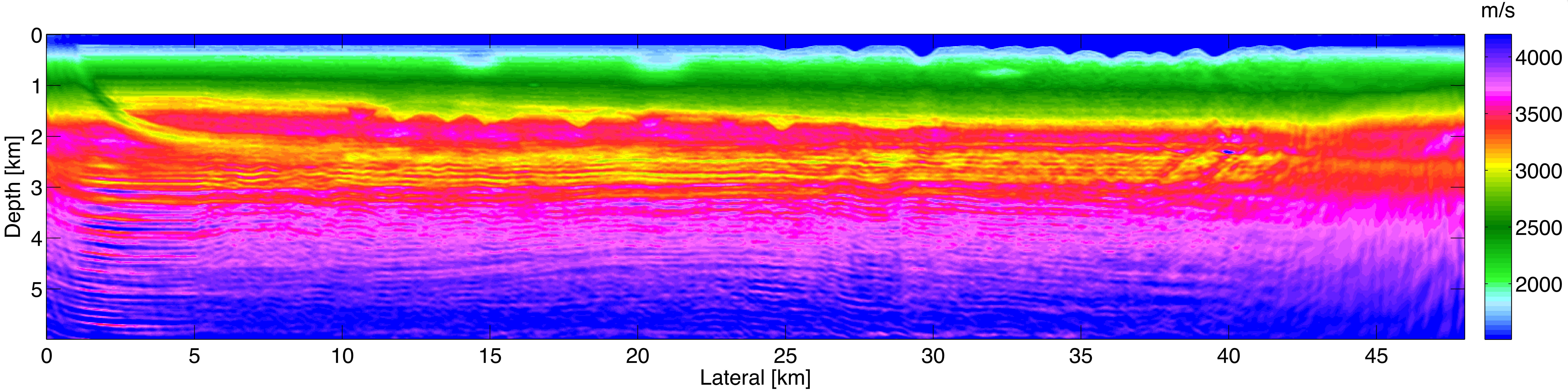
Chevron blind-test elastic data set, work by Zhilong Fang

Acoustic constant density inversion (result in progress, March 2015)



Without minimum smoothness constraint

Example 3 - WRI



Using minimum smoothness constraint

Related geophysical work

- **[A. Baumstein, 2013]** . This work attempts to find the projection onto an intersection using POCS, for different constraints. Includes preconditioner in the Projected-gradient algorithm. May not converge.
- **[E. Esser et. al., 2014; 2015]** (UBC Tech report; this EAGE). Similar philosophy/ideas & problem formulation, different constraints and algorithms.
- **[B. Peters, B. R. Smithyman & F.J. Herrmann, 2015]** (UBC Tech report) projected quasi-Newton based version of this presentation.
- **[B. R. Smithyman, B. Peters & F.J. Herrmann, 2015]** (this EAGE). About the land dataset, uses projected quasi-Newton.
- **[S. Becker et. al., 2015]**. (this EAGE) Also uses projected quasi-Newton, for projections onto a single set. (similar)
- **[B. Peters, Z. Fang, B. R. Smithyman & F.J. Herrmann, 2015]** (submitted to SEG 2015 conference). About the Chevron blind-test dataset (2014). Projected Newton-type using ADMM.

SLIM

Summary & conclusions

- Can combine different regularization approaches as:

$$\min_{\mathbf{m}} f(\mathbf{m}) + \frac{\alpha}{2}\|R_1\mathbf{m}\|_2^2 + \frac{\beta}{2}\|R_2\mathbf{m}\|_2^2 \quad \text{s.t.} \quad \mathbf{m} \in \mathcal{C}_1 \bigcap \mathcal{C}_2$$

- Developed flexible and extendable framework for including constraints for any differentiable objective.
- Works with various optimization algorithms.
- Requires no extra PDE-solves.
- Easy to use, prior information translates into constraints directly, without penalty parameters.

Outlook

- Add more constraint sets with their projectors (suggestions?)
- Will be available in the SLIM software.

Acknowledgements

PhD students and Postdocs at SLIM

This work was in part financially supported by the Natural Sciences and Engineering Research Council of Canada via the Collaborative Research and Development Grant DNOISEII (375142--08). This research was carried out as part of the SINBAD II project with support from the following organizations: BG Group, BGP, CGG, Chevron, ConocoPhillips, DownUnder GeoSolutions, Hess, ION, Petrobras, PGS, Schlumberger, Statoil, Sub Salt Solutions, Total SA, and Woodside.

References (1)

1. Baumstein, A. [2013] Pocs-based geophysical constraints in multi-parameter full wavefield inversion. EAGE.
2. Bauschke, H. and Borwein, J. [1994] Dykstras alternating projection algorithm for two sets. Journal of Approximation Theory, 79(3), 418 – 443, ISSN 0021-9045, doi:http://dx.doi.org/10.1006/jath.1994.1136.
3. Brenders, A.J. and Pratt, R.G. [2007] Full waveform tomography for lithospheric imaging: results from a blind test in a realistic crustal model. Geophysical Journal International, 168(1), 133–151, doi:10.1111/j.1365- 246X.2006.03156.x.
4. Nocedal, J. and Wright, S.J. [2000] Numerical optimization. Springer.
5. Schmidt, M., van den Berg, E., Friedlander, M. and Murphy, K. [2009] Optimizing costly functions with simple constraints: A limited-memory projected quasi-newton algorithm. JMLR, vol. 5, 456–463.
6. Sen, M. and Roy, I. [2003] Computation of differential seismograms and iteration adaptive regularization in prestack waveform inversion. GEOPHYSICS, 68(6), 2026–2039, doi:10.1190/1.1635056.
7. Becker, SR and Horesh, L and Aravkin, AY and van den Berg, E and Zhuk, S . [2015] General Optimization Framework for Robust and Regularized 3D FWI. 77th EAGE Conference and Exhibition 2015
8. Smithyman, B. R., B. Peters, and F. J. Herrmann. "Constrained Waveform Inversion of Colocated VSP and Surface Seismic Data." 77th EAGE Conference and Exhibition 2015. 2015.
9. Schmidt, Mark, Dongmin Kim, and Suvrit Sra. "Projected Newton-type methods in machine learning." (2011).

References (2)

9. Bas Peters, Zhilong Fang, Brendan Smithyman, Felix J. Herrmann. Regularizing waveform inversion by projections onto convex sets –- application to the 2D Chevron 2014 synthetic blind-test dataset. (submitted to the SEG conference). 2015. https://www.slim.eos.ubc.ca/Publications/Private/Conferences/SEG/2015/peters2015SEGrwi/peters2015SEGrwi.html

10. Bas Peters, Brendan Smithyman, Felix J. Herrmann. Waveform inversion by projection onto intersections of convex sets. UBC Tech Report. 2015.https://www.slim.eos.ubc.ca/Publications/Public/TechReport/2015/peters2015EAGErwi/peters2015EAGErwi.html