Regularizing waveform inversion by projections onto intersections of convex sets

Bas Peters
Joint work with Brendan Smithyman and Mathias Louboutin.

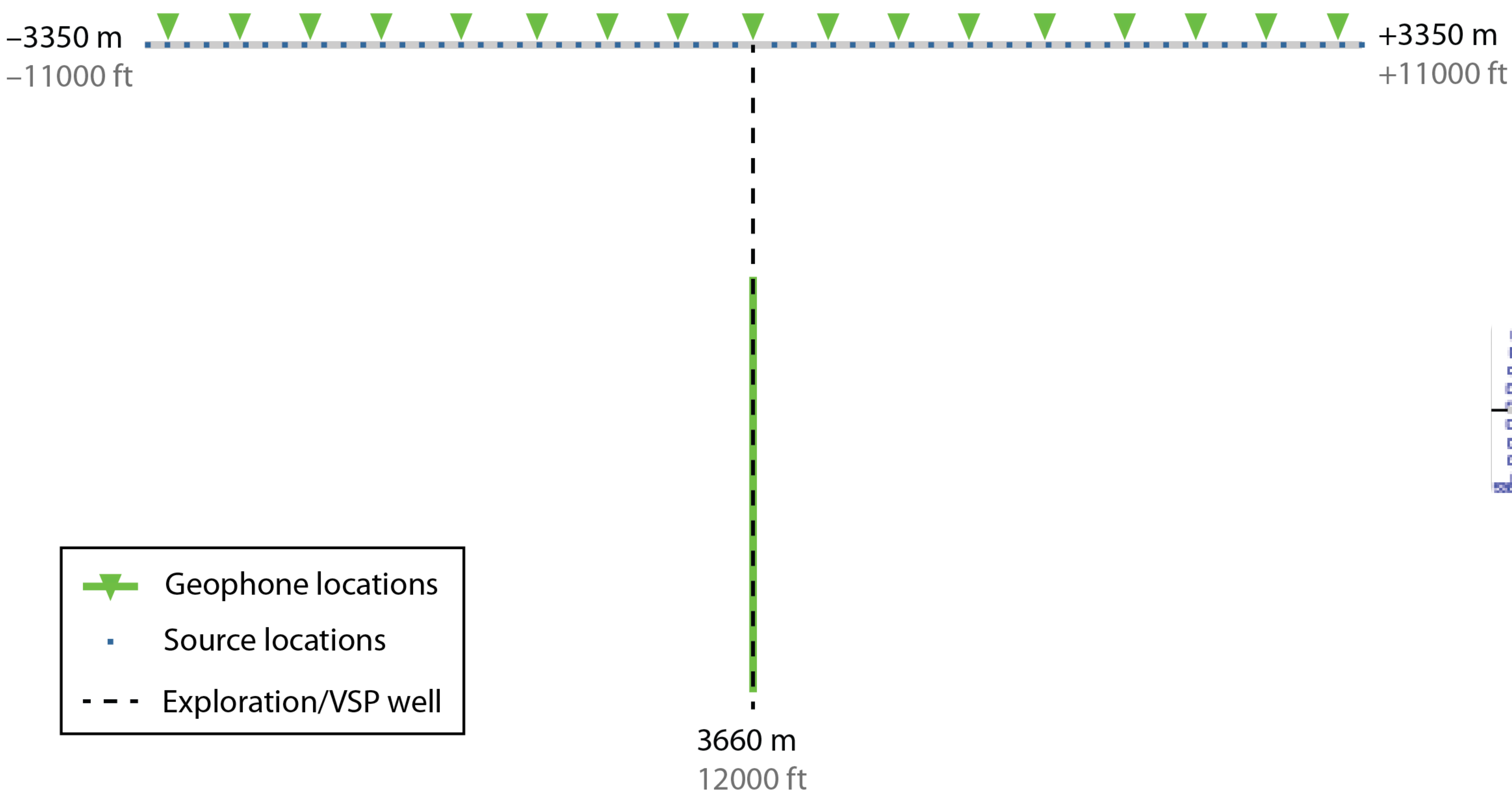
2015 SINBAD Fall Consortium meeting. October 27.

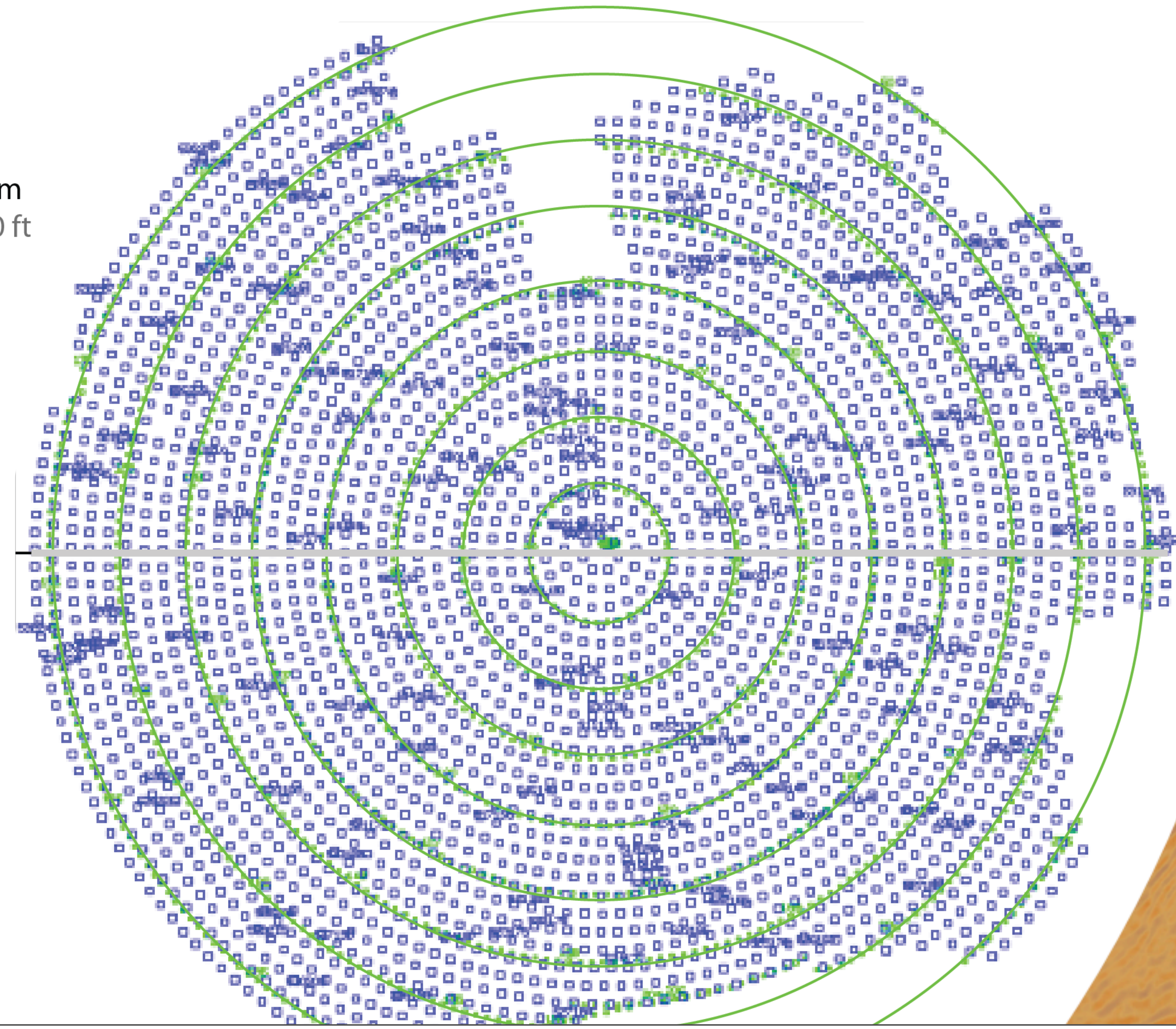
SLIM
University of British Columbia

Motivation

Land data set with surface sources and surface & well receivers

Constant density acoustic inversion

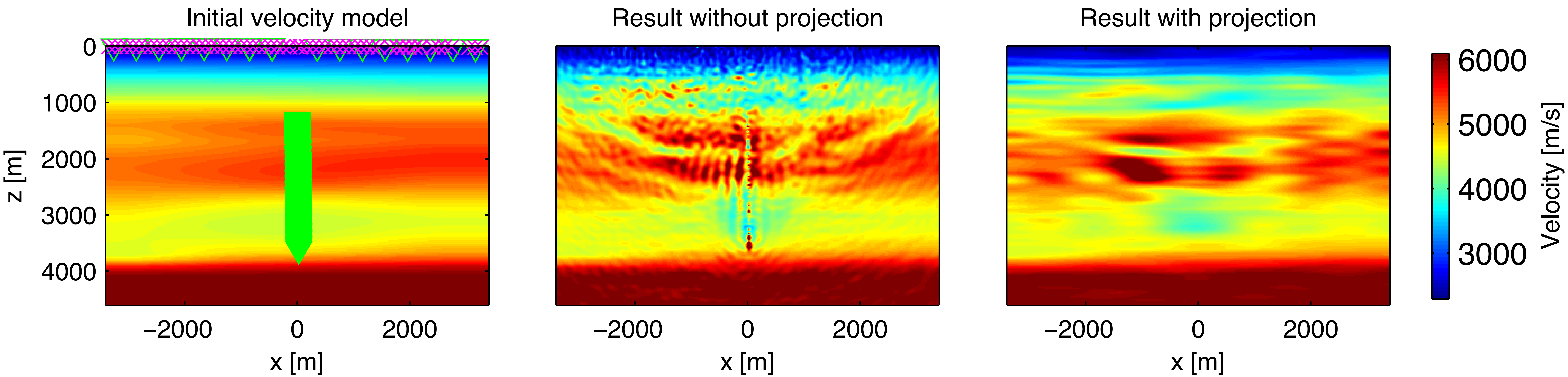




Motivation

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

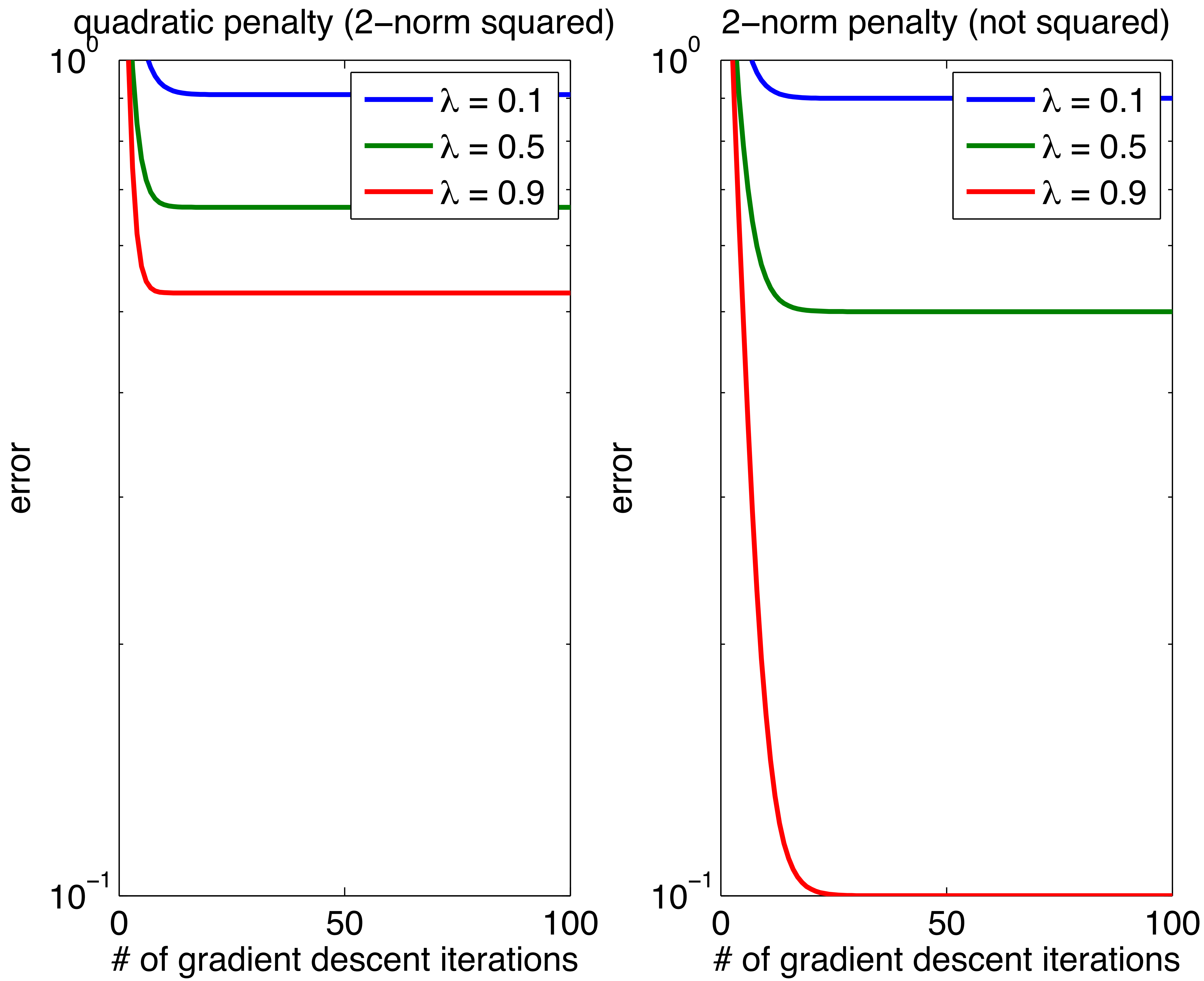
A few regularization strategies

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:

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

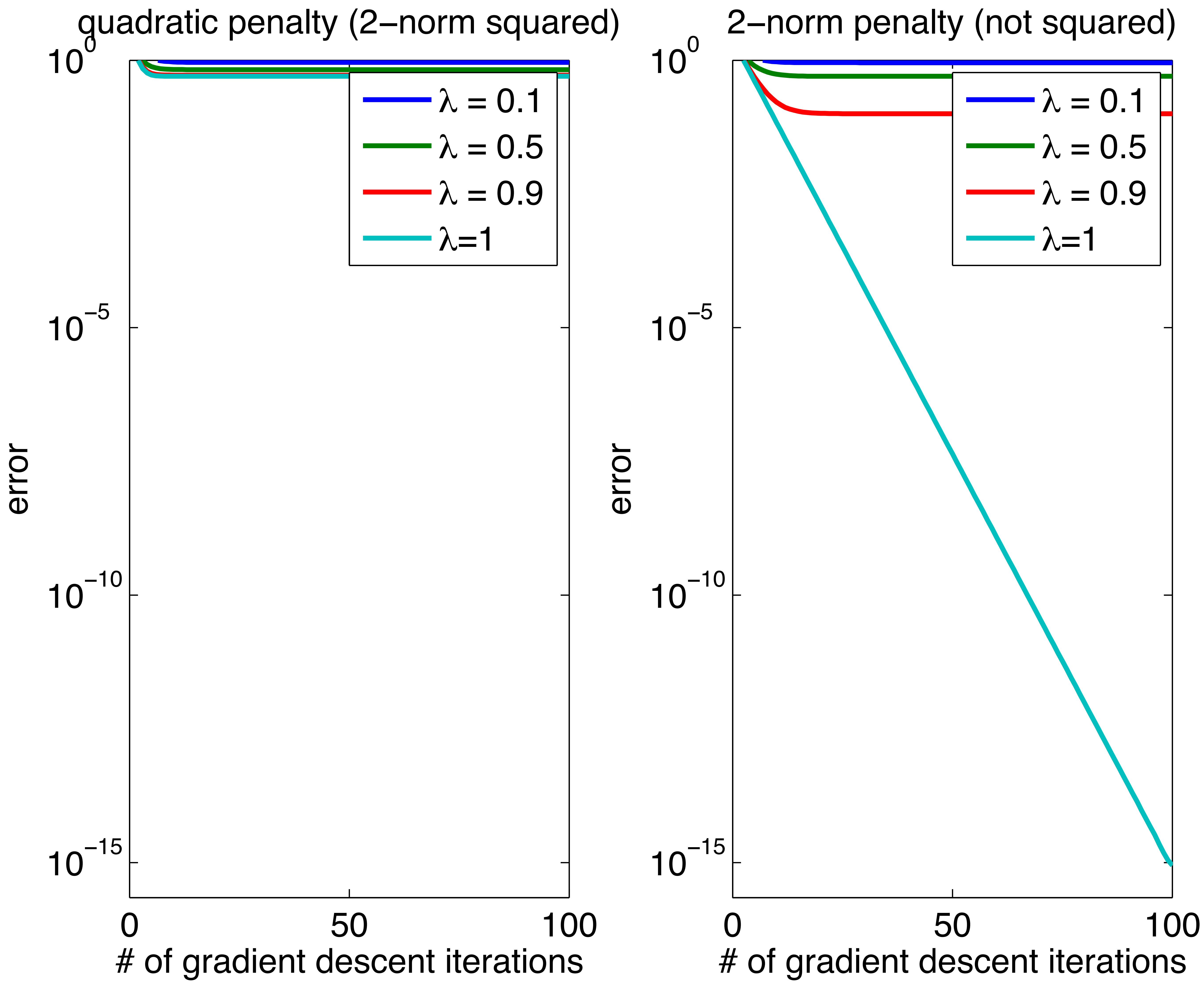
Quadratic-penalty:

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

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.J. Brenders & R.G. Pratt, 2007]

SLIM

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

Can work if F is definite positive

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):

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

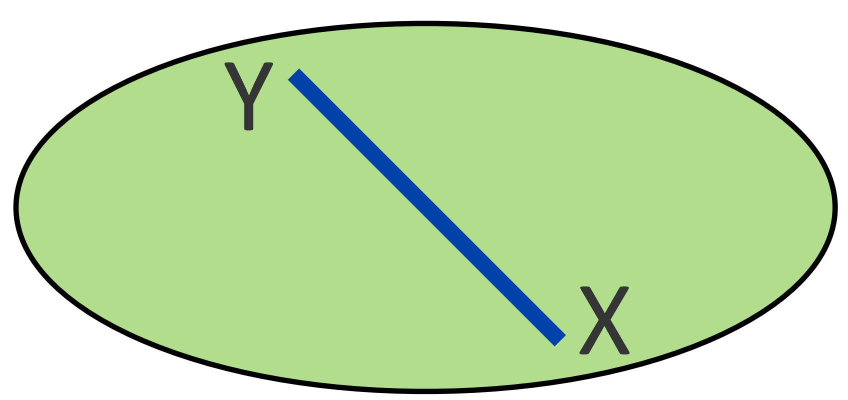
Important property:

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

Convex sets : some properties

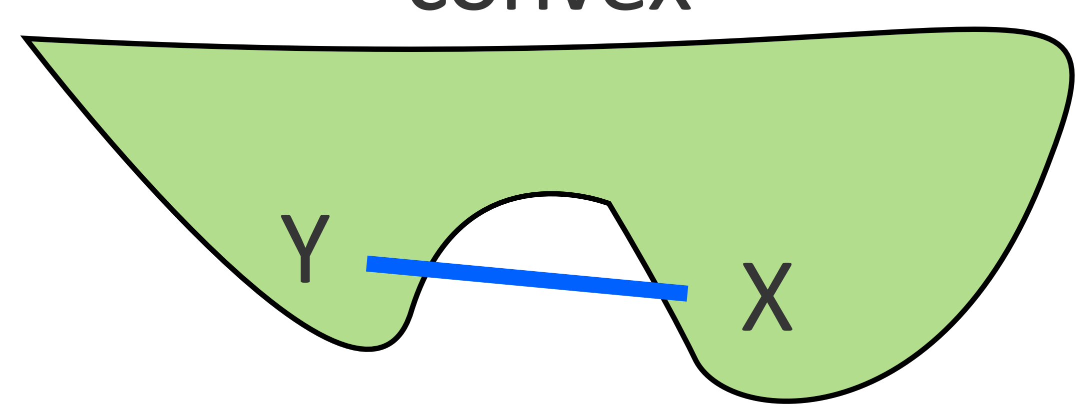
Convex set

- there is a linear path contained in the set between every pair of the set
- every point is linearly reachable from another point
- projection onto a convex set is unique



Convex

Non
convex



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: [B.R. Smithyman et. al., 2015; B. Peters et. al., 2015]

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

"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 $k = 0, 1, \ldots$

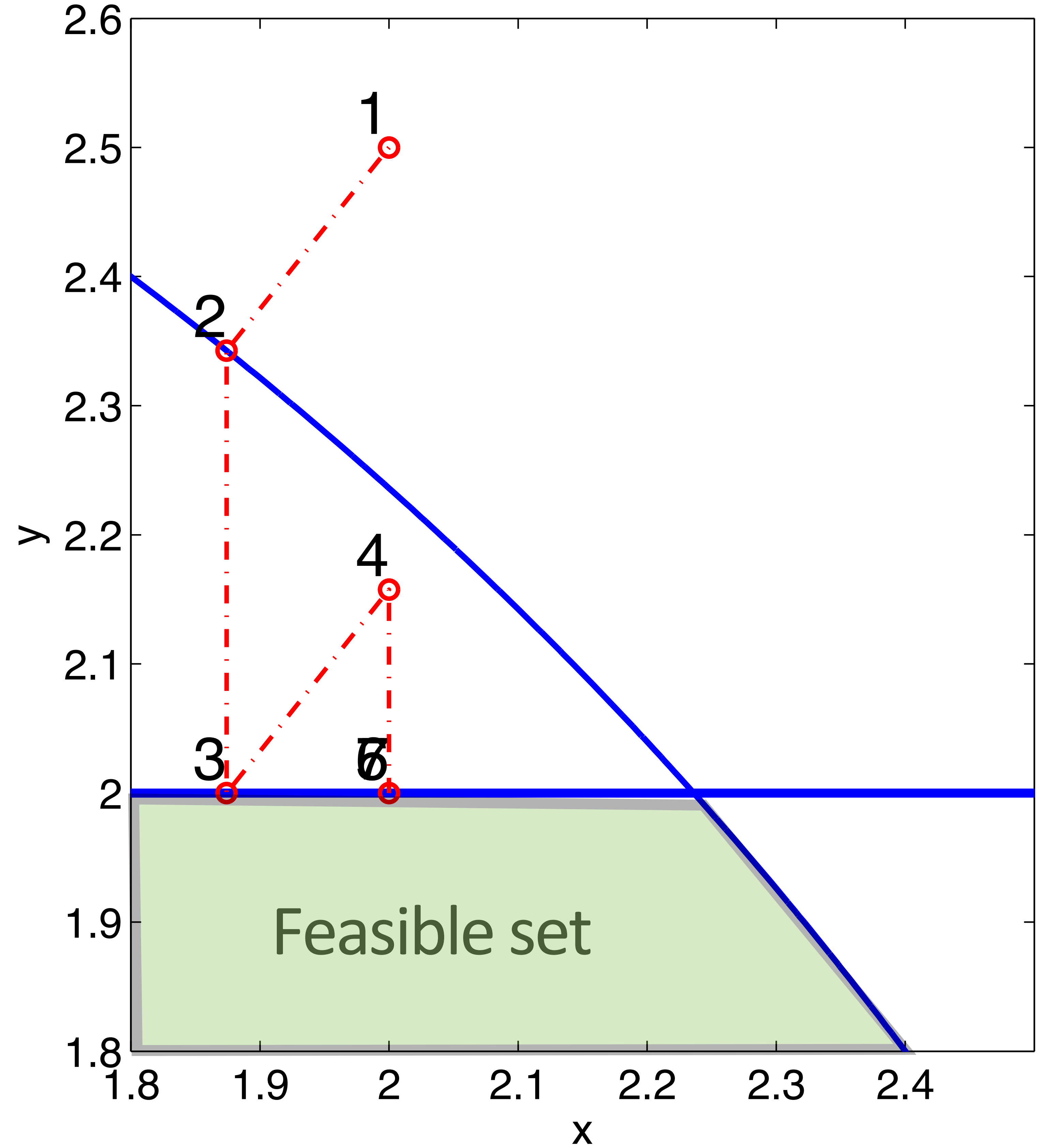
$$y_k = \mathscr{P}_{C_1}(x_k + p_k)$$

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

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

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

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 $k = 0, 1, \ldots$

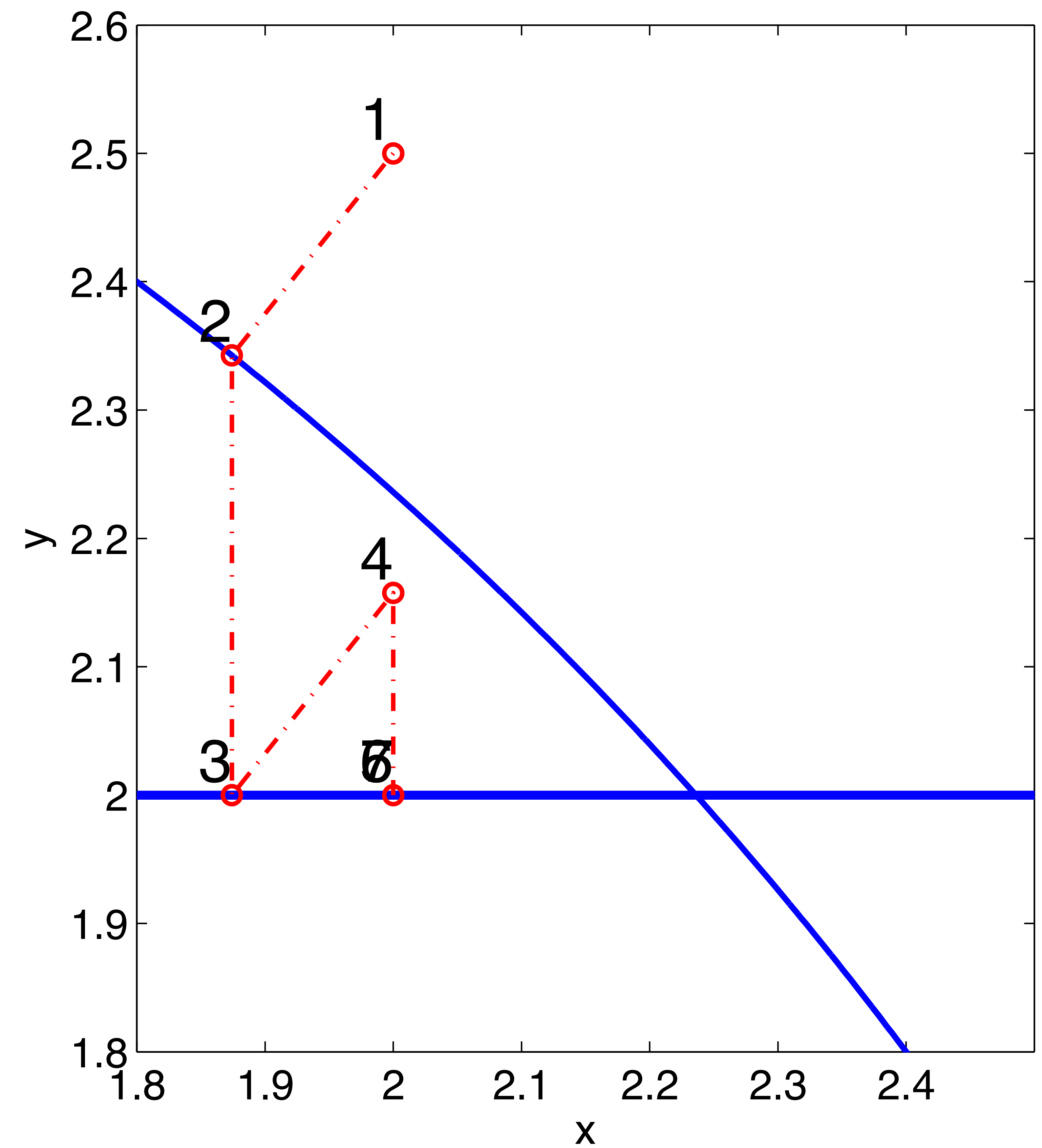
→ $y_k = \mathscr{P}_{C_1}(x_k + p_k)$

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

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

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

End



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 $k = 0, 1, \ldots$

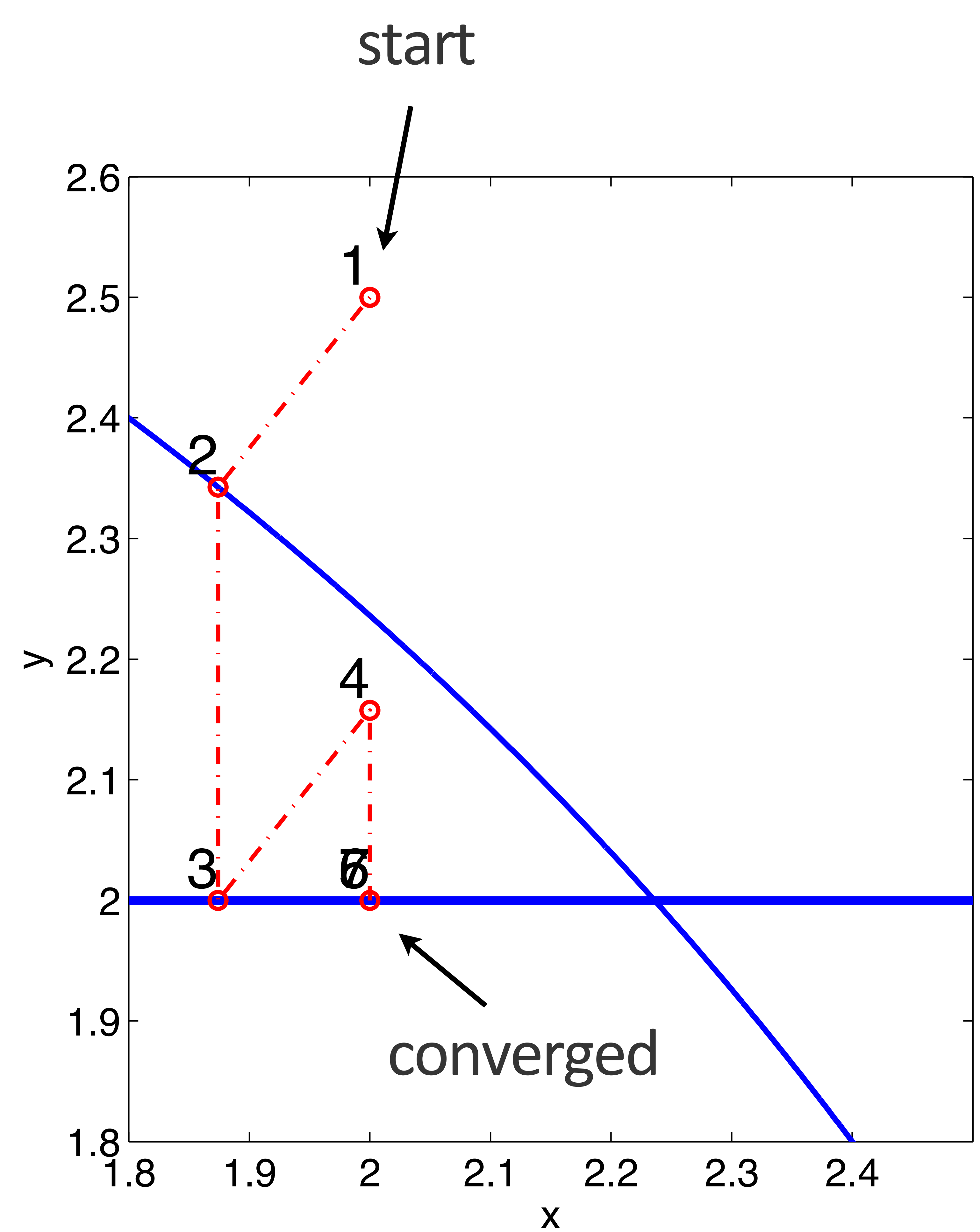
$$y_k = \mathscr{P}_{C_1}(x_k + p_k)$$

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

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

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

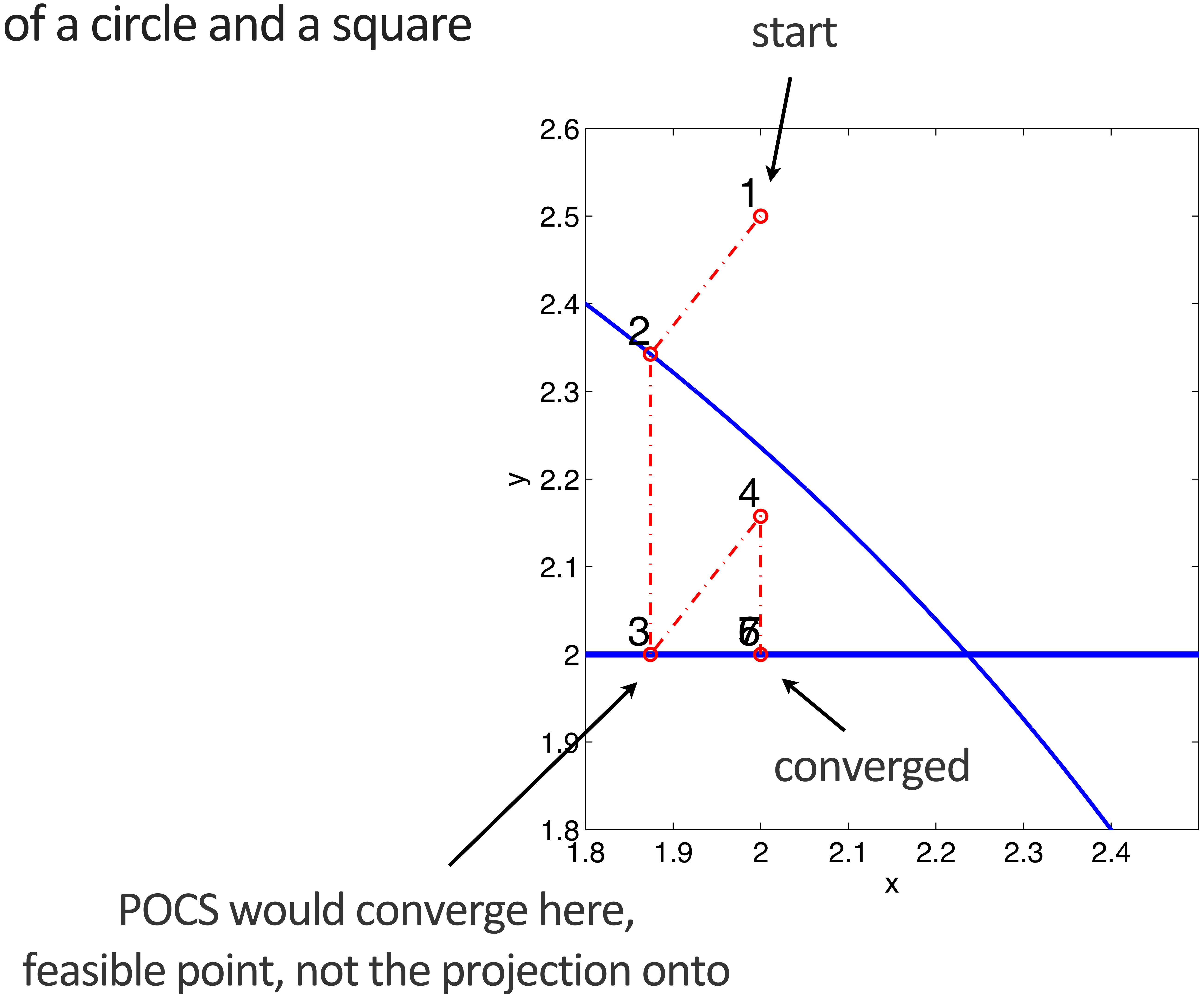
End



Dykstra splitting

Toy example:

find projection onto intersection of a circle and a square



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:

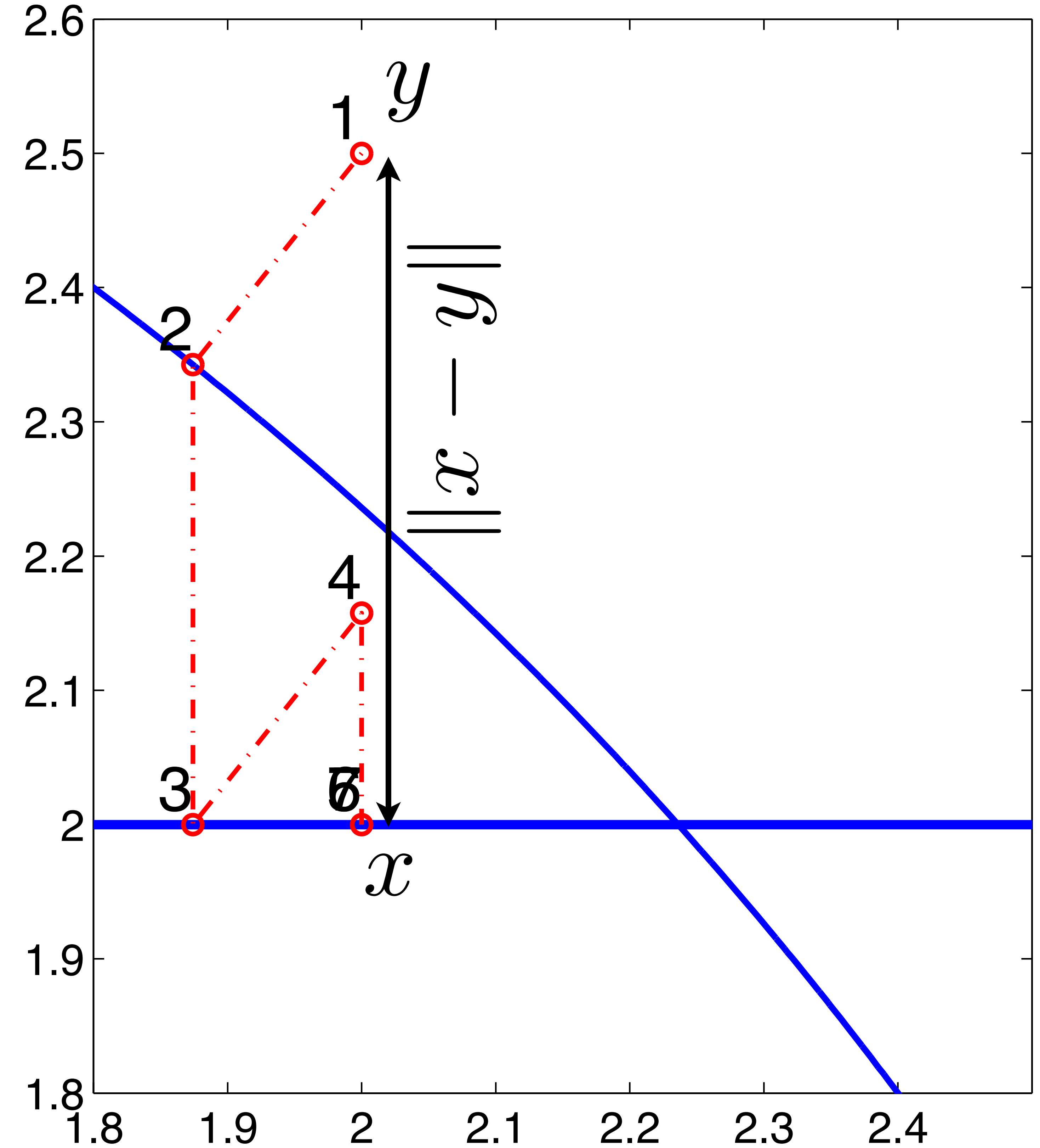
$$\min_x \frac{1}{2}\|x - y\|^2 \quad \text{s.t.} \quad x \in \mathcal{C}_1 \bigcap \mathcal{C}_2$$

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

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

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

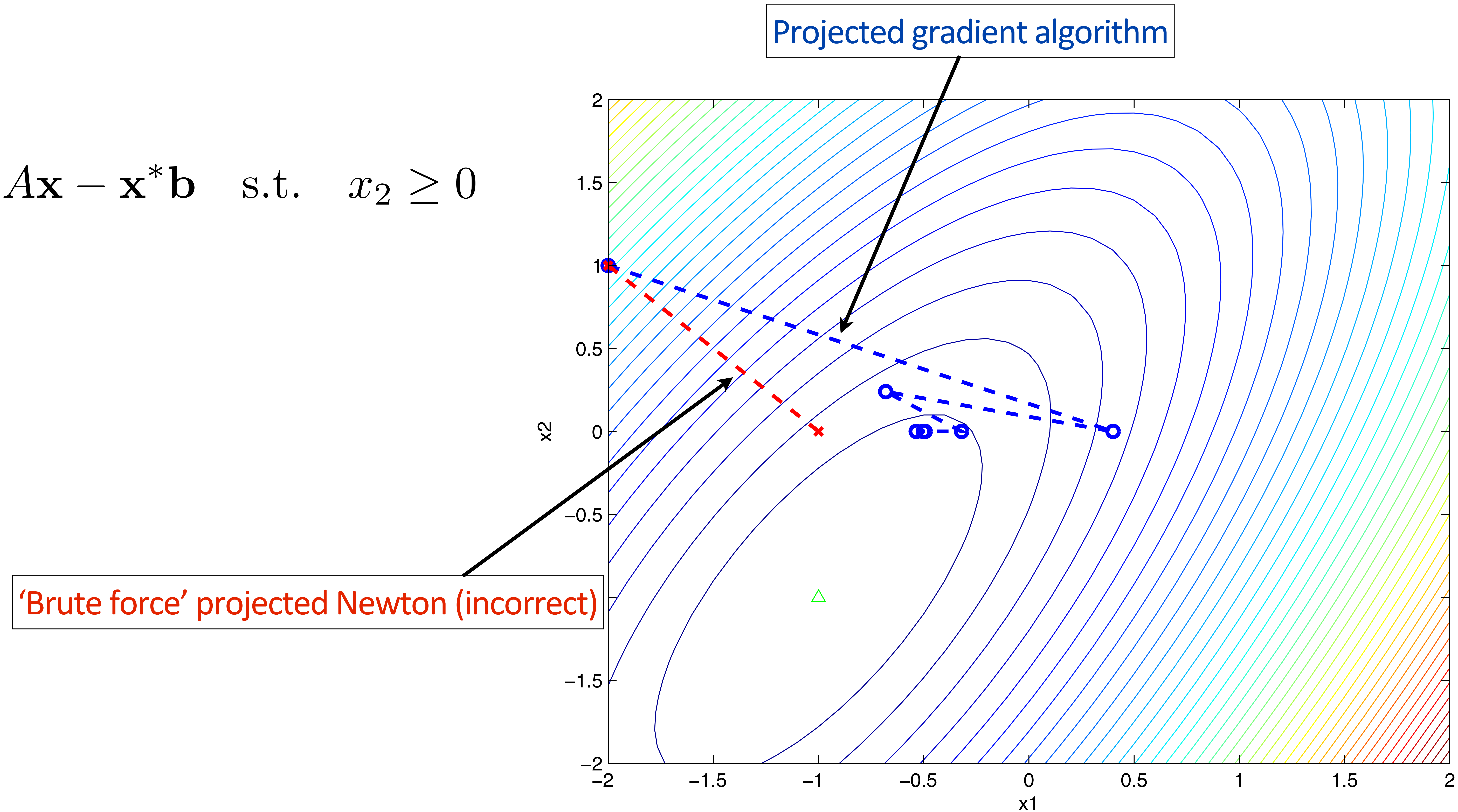
Projected-gradient: $\mathbf{m}_{k+1} = \mathcal{P}_{\mathcal{C}}(\mathbf{m}_k - \gamma \nabla_{\mathbf{m}} f(\mathbf{m}_k))$

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

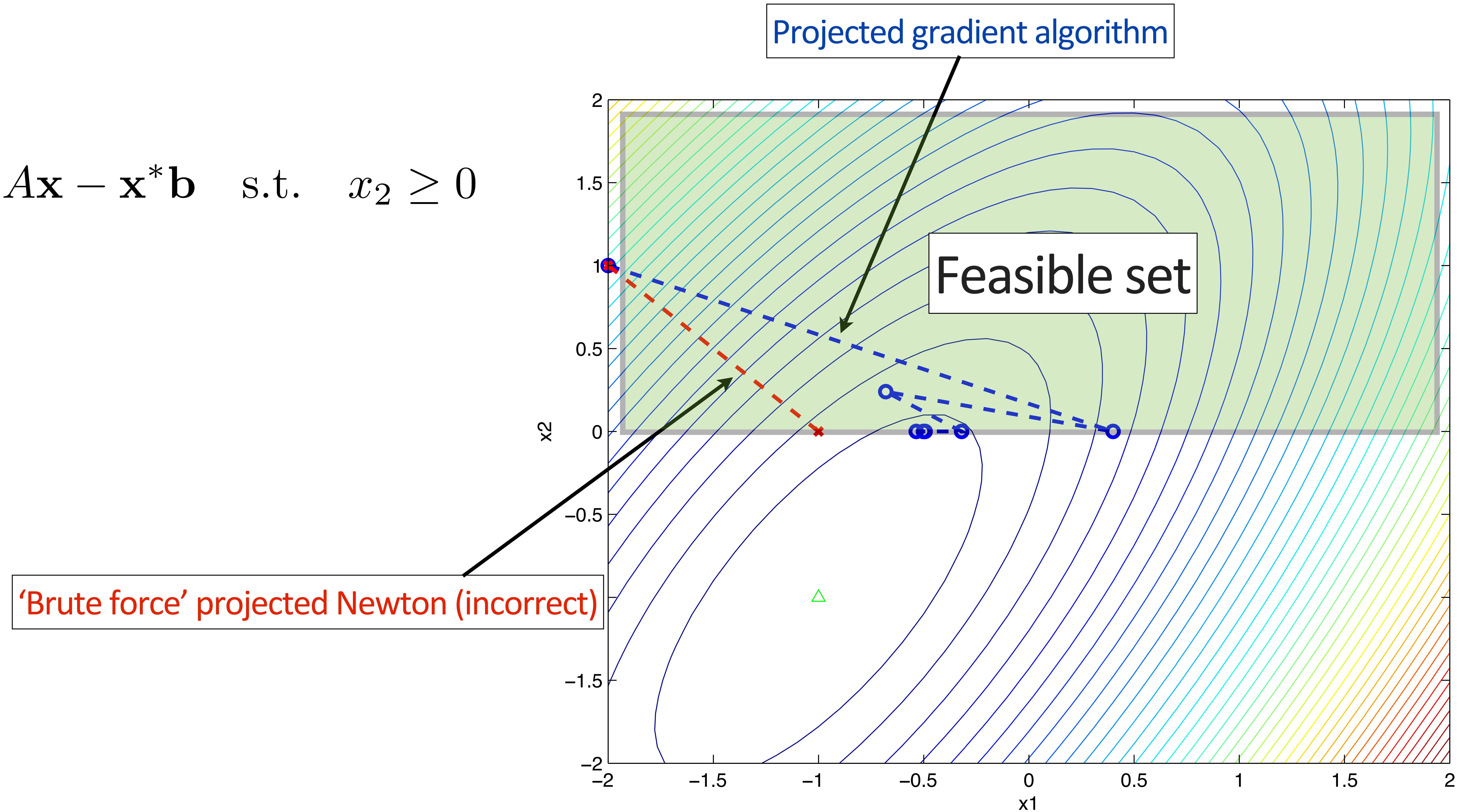
$$\mathbf{m}_{k+1} = \mathcal{P}_{\mathcal{C}}(\mathbf{m}_k - \gamma B(\mathbf{m}_k)^{-1} \nabla_{\mathbf{m}} f(\mathbf{m}_k))$$

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

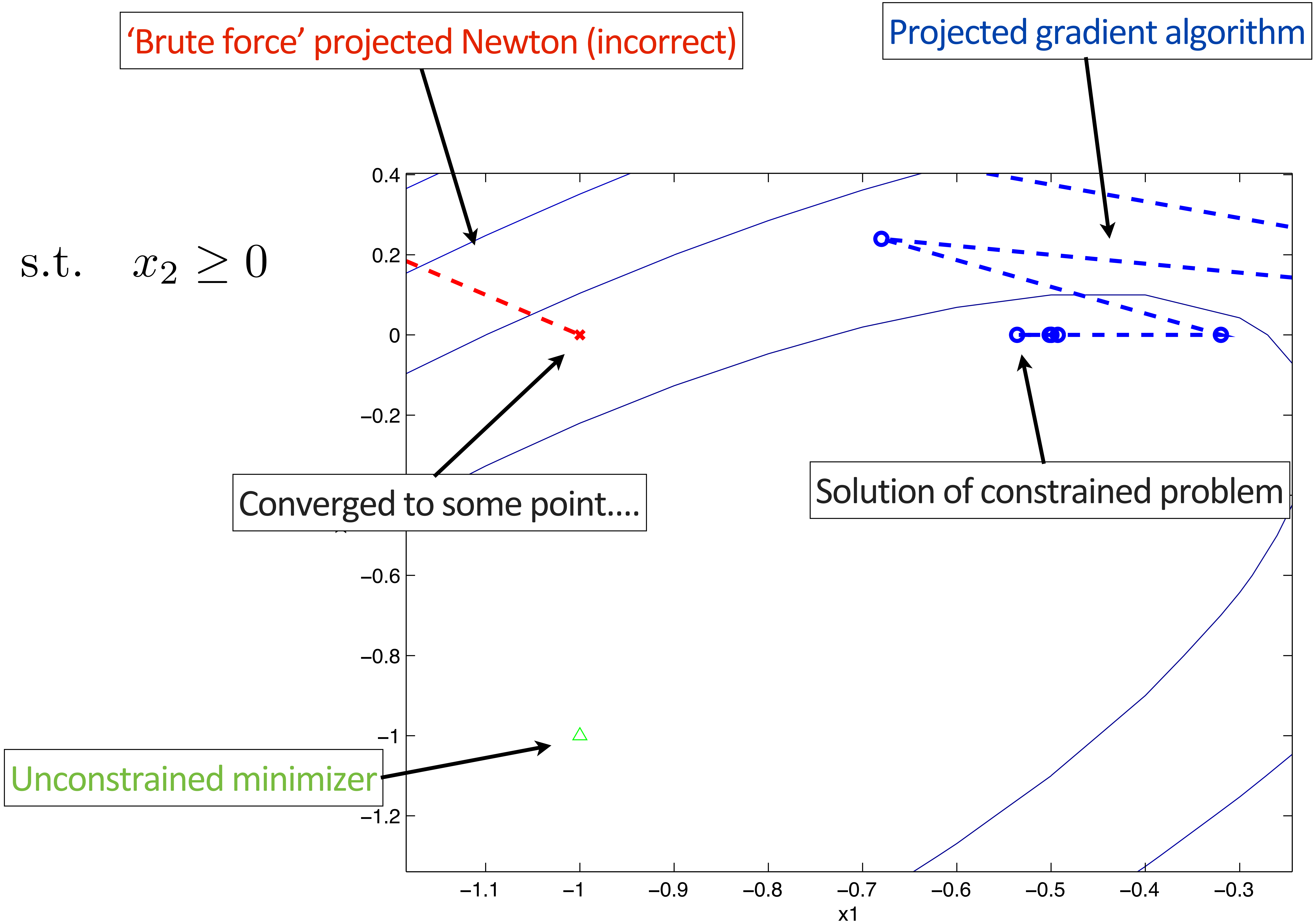
$$\min_{\mathbf{x}} \mathbf{x}^* A\mathbf{x} - \mathbf{x}^*\mathbf{b} \quad \text{s.t.} \quad x_2 \geq 0$$



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



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



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]**

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

(unconstrained Newton-step)

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

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

Algorithmic development

Projection methods do not modify the gradient or Hessian.

Instead, they find an updated model which still satisfies the constraints.

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:

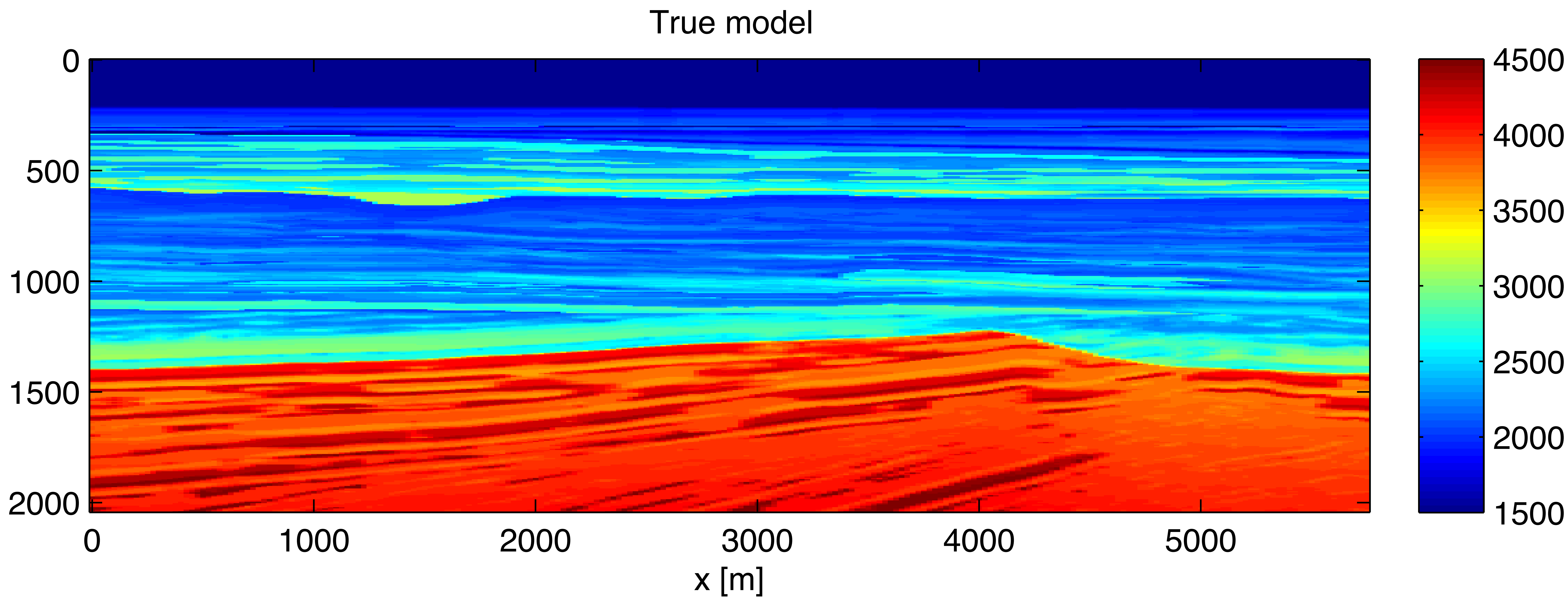
- solve PDE's
- Solve quadratic problem with constraints using ADMM
- at every iteration of ADMM:
  - invert Hessian (possibly iteratively)
  - solve projection problem onto an intersection of convex sets using Dykstra's algorithm
  - at every iteration of Dykstra's algorithm:
    - compute projections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

Example 1 - FWI with noise

- Sources near water surface, receivers at ocean bottom
- 3-13 Hz data
- $\|\text{noise}\|_2 / \|\text{signal}\|_2 = 1$
- used bound constraints and minimum smoothness constraints

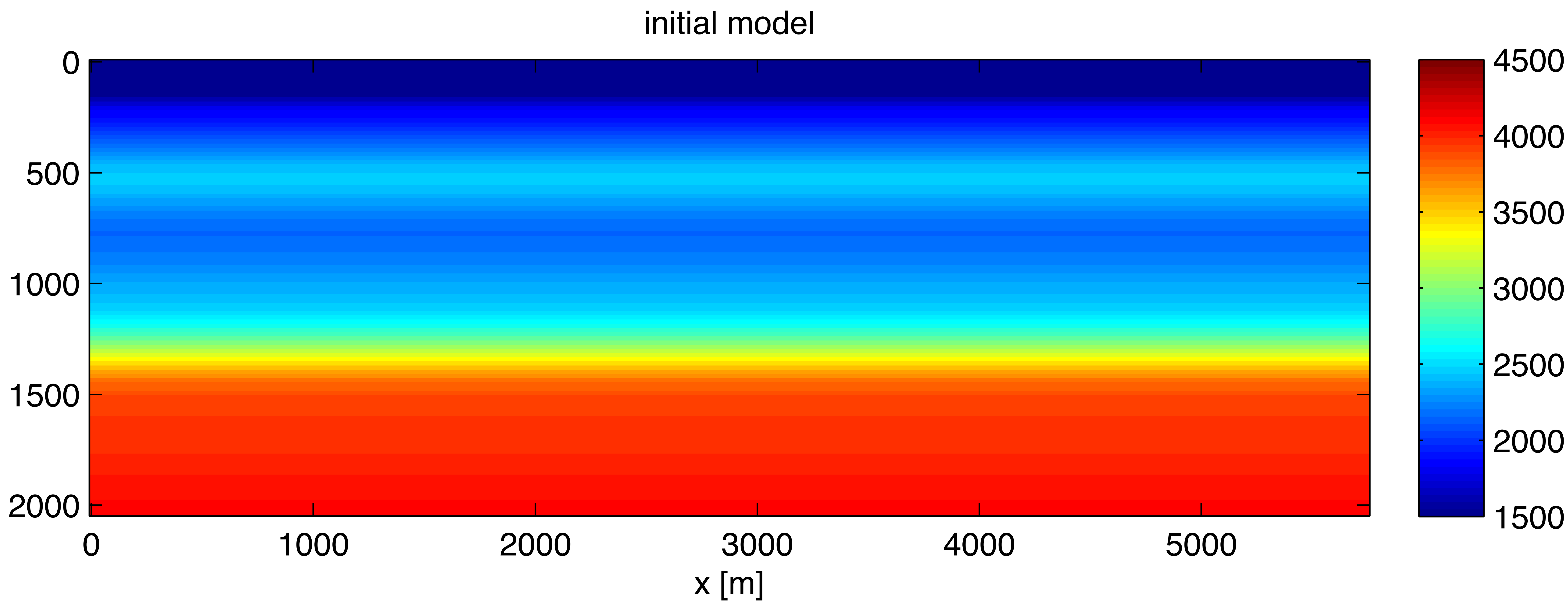
Example 1 - FWI with noise

true model



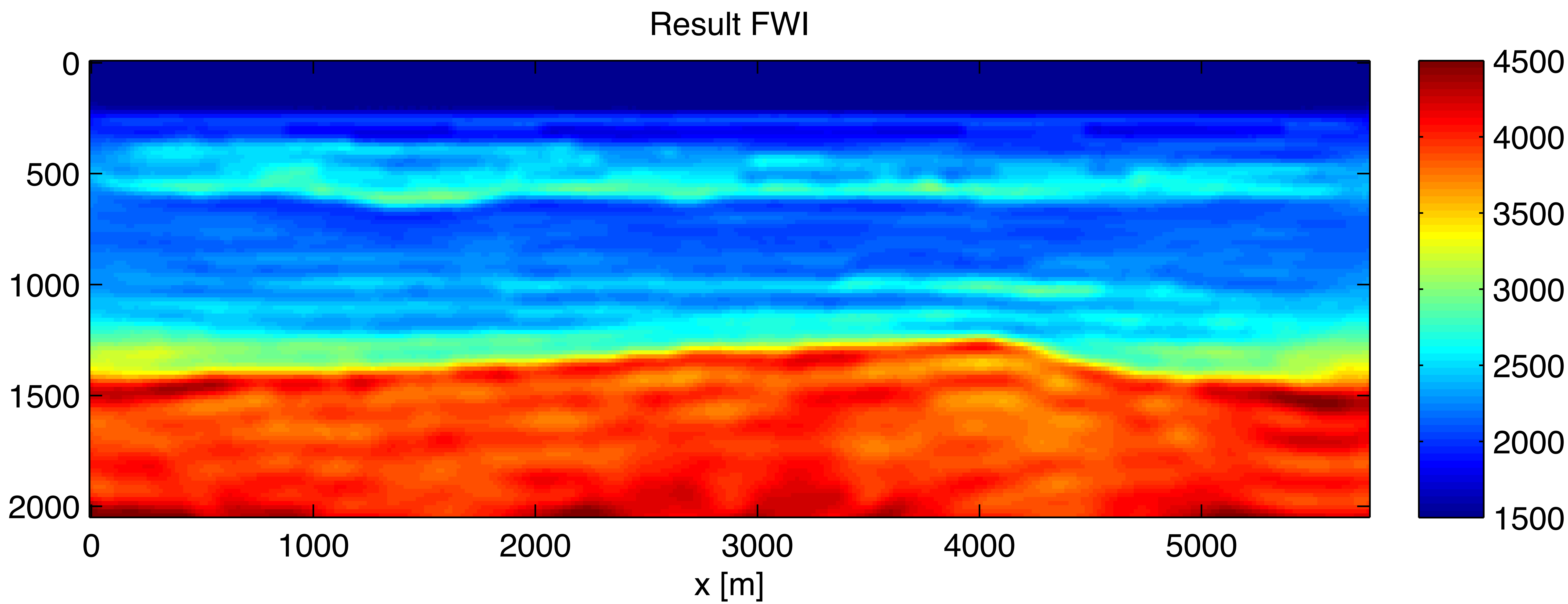
Example 1 - FWI with noise

start model



Example 1 - FWI with noise

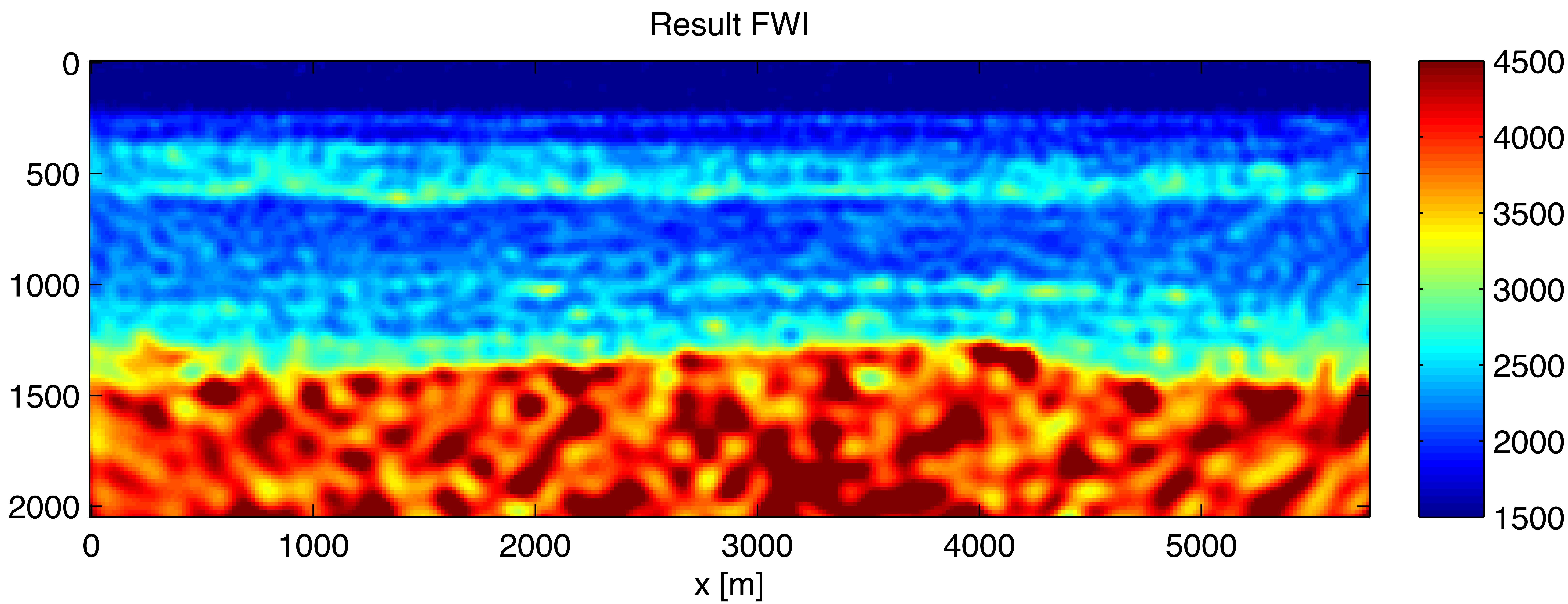
No noise in data - bound constraints only



Example 1 - FWI with noise

$$\|\text{noise}\|_2 / \|\text{signal}\|_2 = 1$$

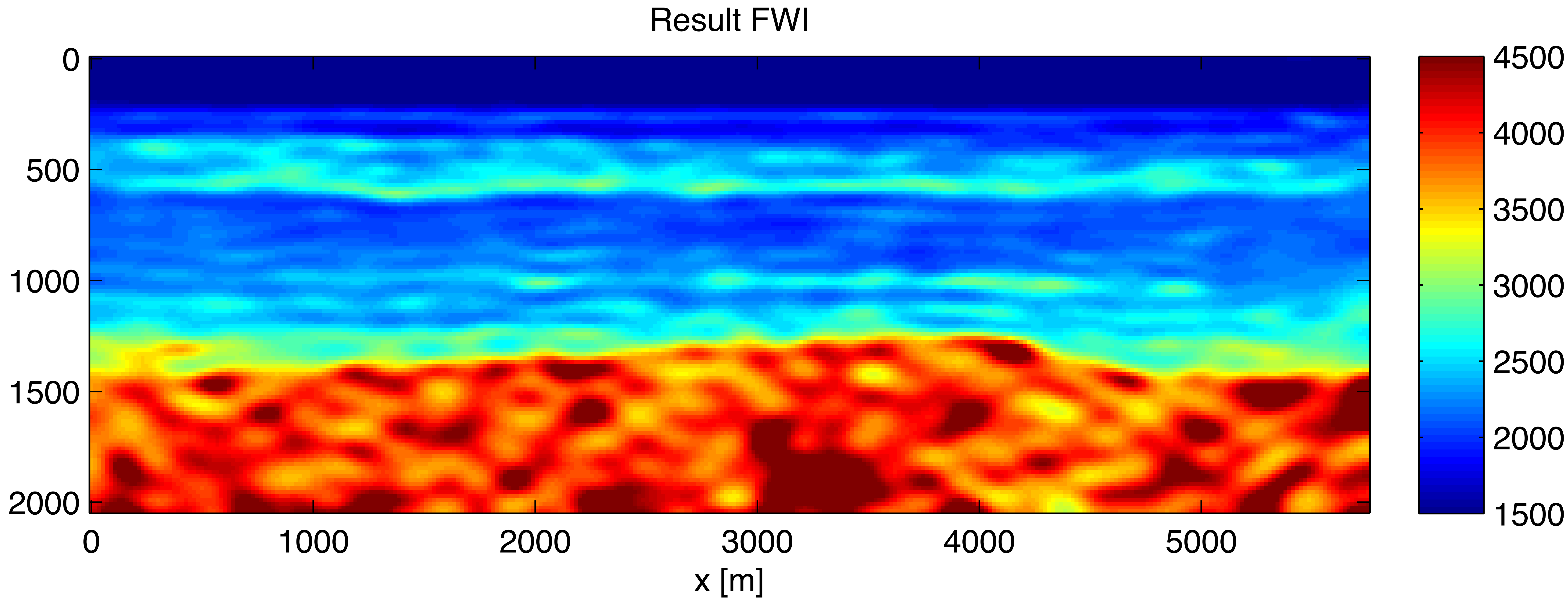
noise in data - bound constraints only



Example 1 - FWI with noise

$\|\text{noise}\|_2/\|\text{signal}\|_2 = 1$

noise in data - bound constraints and minimum smoothness constraints



Example 1 - FWI with noise

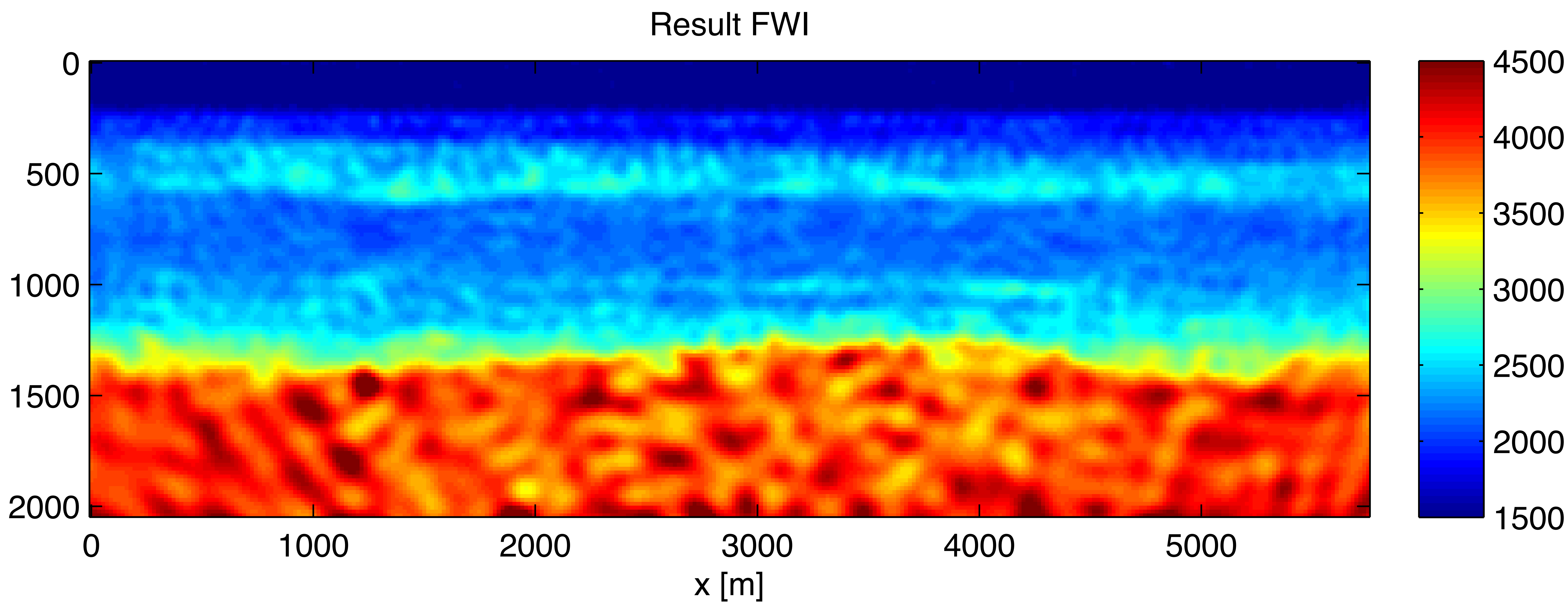
In this case the constraints act as an image-domain noise filter.

Example 2 - FWI with 1 simultaneous source

- Sources near water surface, receivers at ocean bottom
- 3-13 Hz data
- used bound constraints and minimum smoothness constraints
- 1 simultaneous source, no redrawing

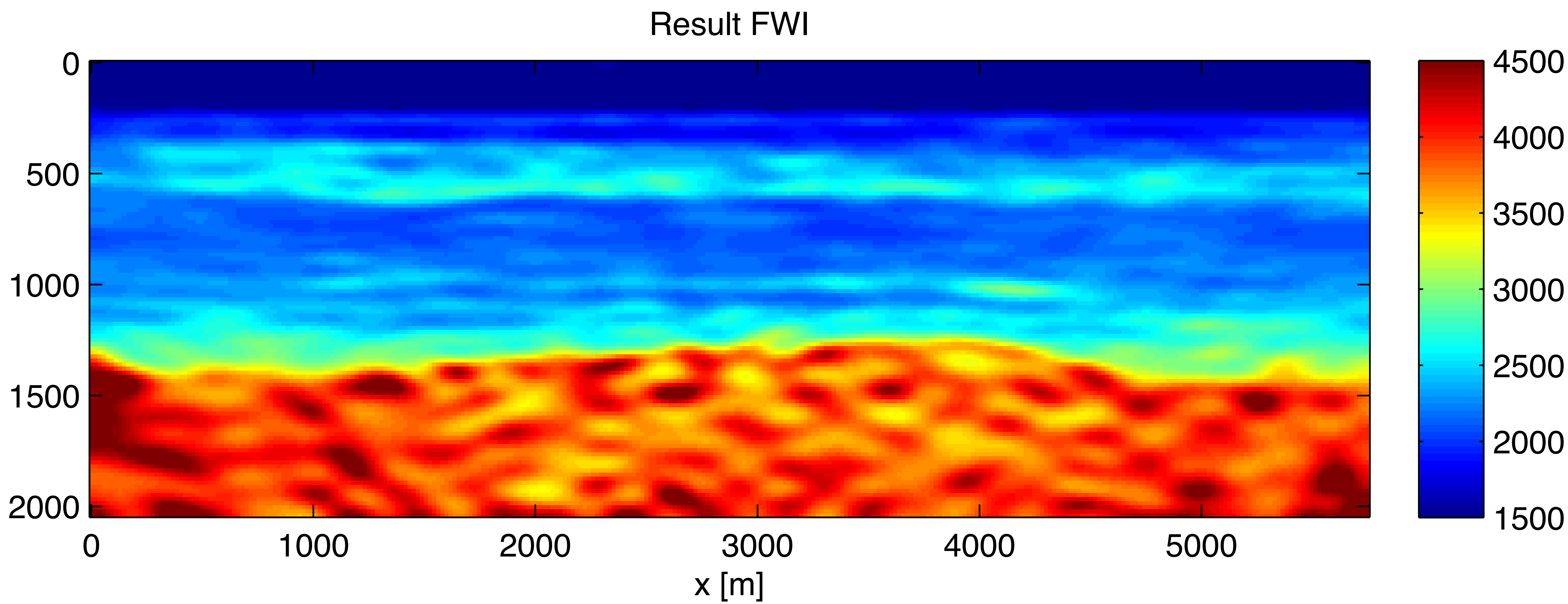
Example 2 - FWI with 1 simultaneous source

only bound constraints



Example 2 - FWI with 1 simultaneous source

using bound constraints and minimum smoothness

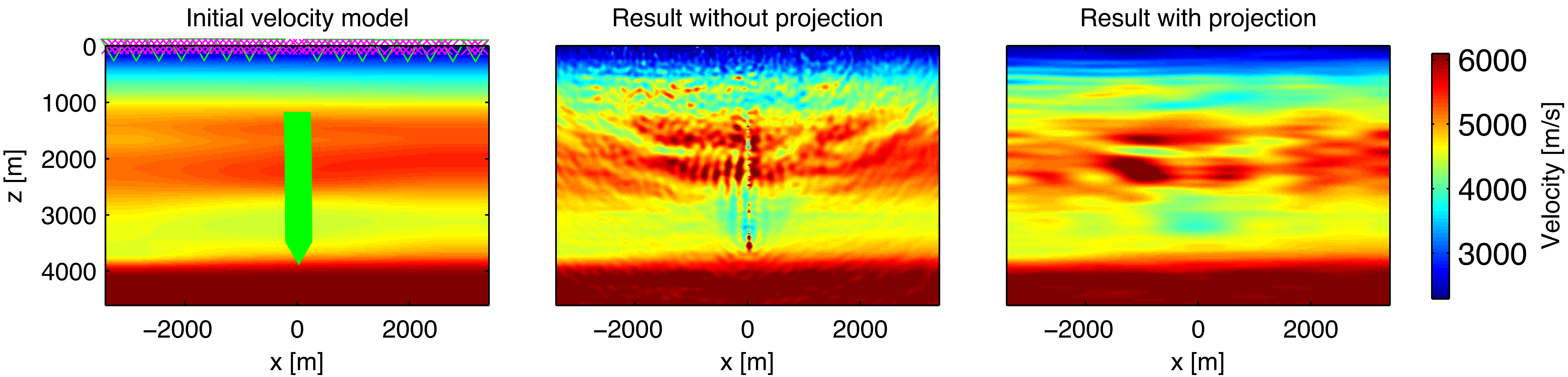


[B.R. Smithyman et. al., 2015; B. Peters et. al., 2015]

Example 3 - FWI on a real land dataset

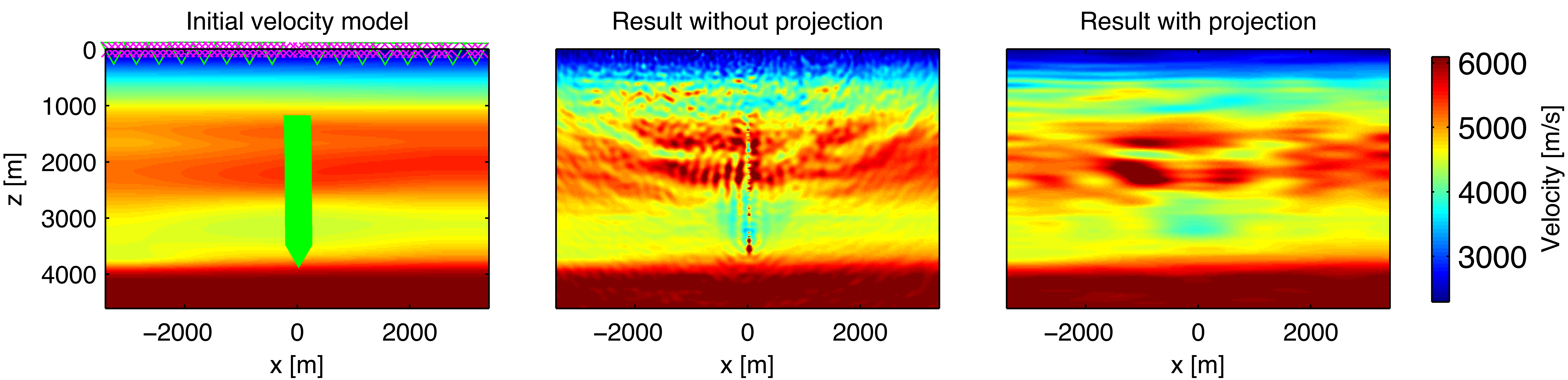
Land data set with surface sources and surface & well receivers

Constant density acoustic inversion (2D slice)



Example 3 - FWI on a real land dataset

- Bound constraints
- Minimum smoothness constraints



Application to time-domain (adjoint-state) (By Mathias)

Memory efficient method

- 5 to 20 times more memory efficient than usual FWI via randomization techniques.
- Less i/o or computationally more efficient (No extra computation during back-propagation).

Constraints used:

- Bound constraints.
- Minimum smoothness constraints to remove most of the method related artifacts.

BG Compass 2D

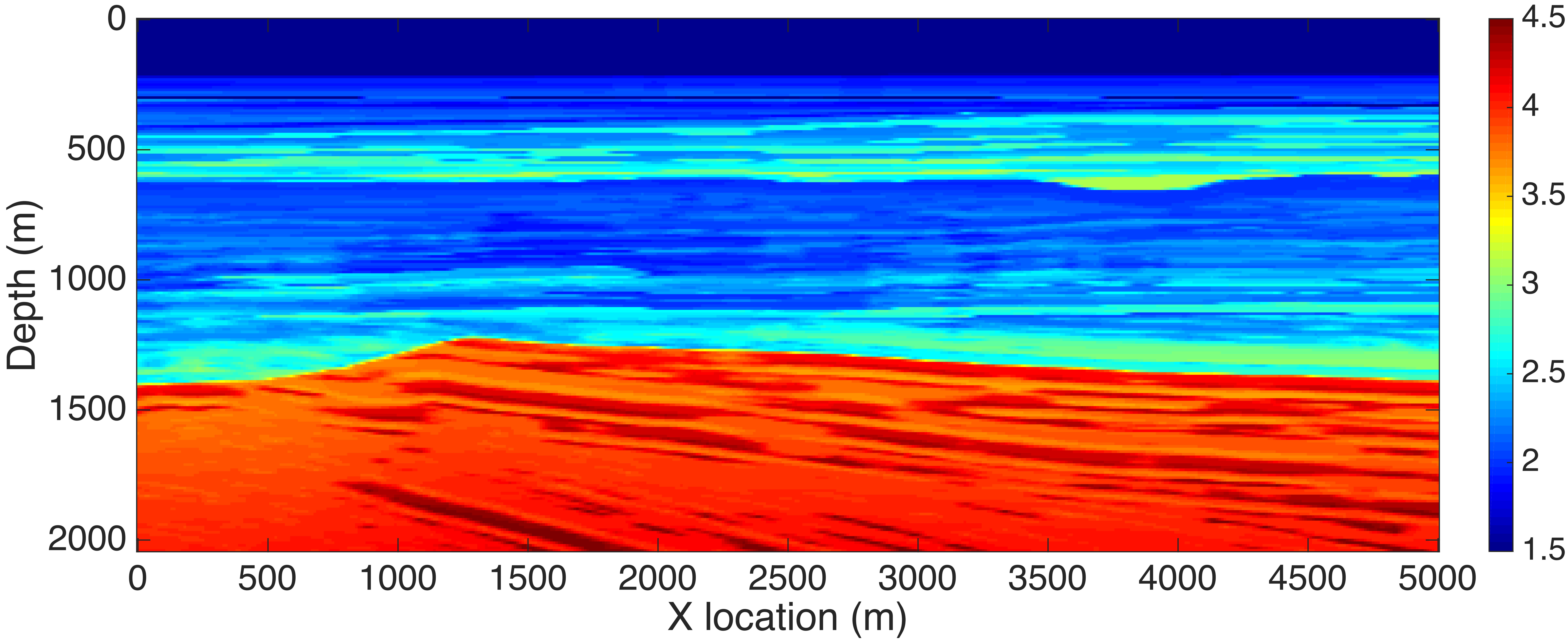
Data:

- Ricker wavelet at 15Hz, 2.4s recording
- 51 sources at 100m interval
- 201 receivers at 25m interval

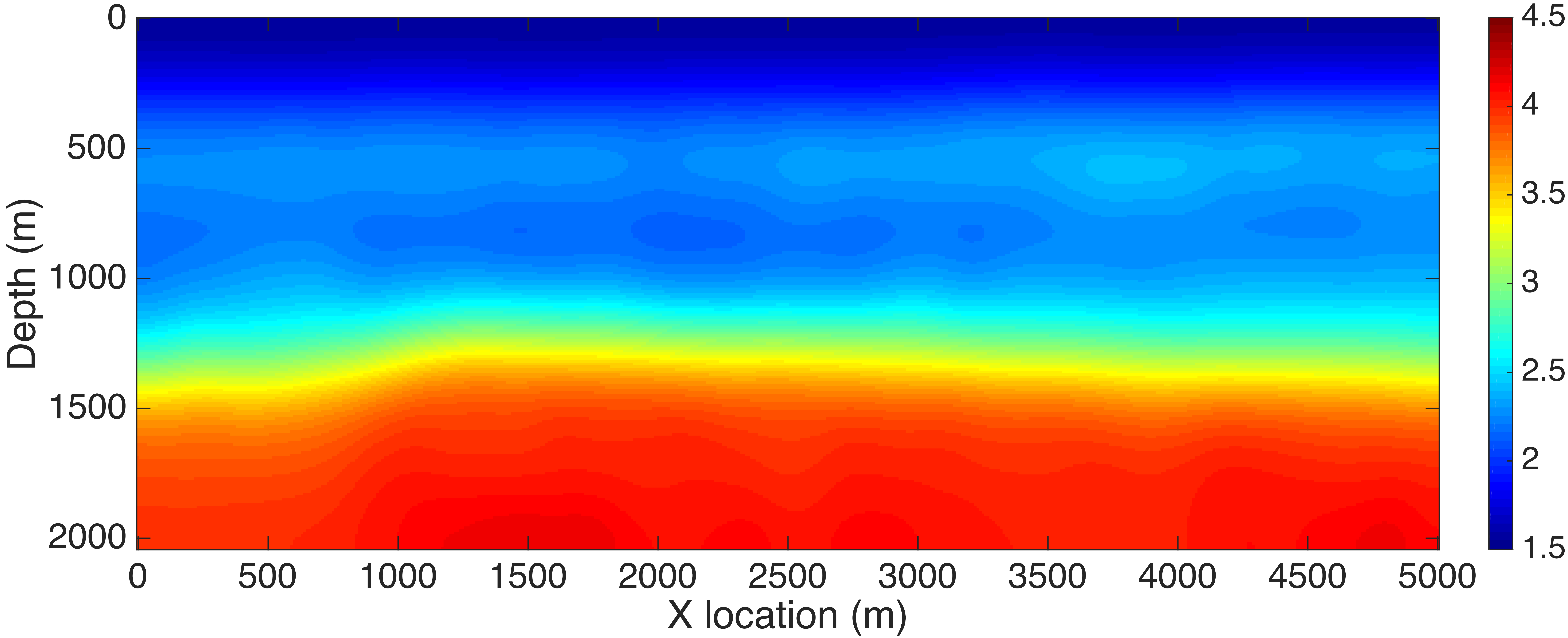
Acoustic modeling and inversion

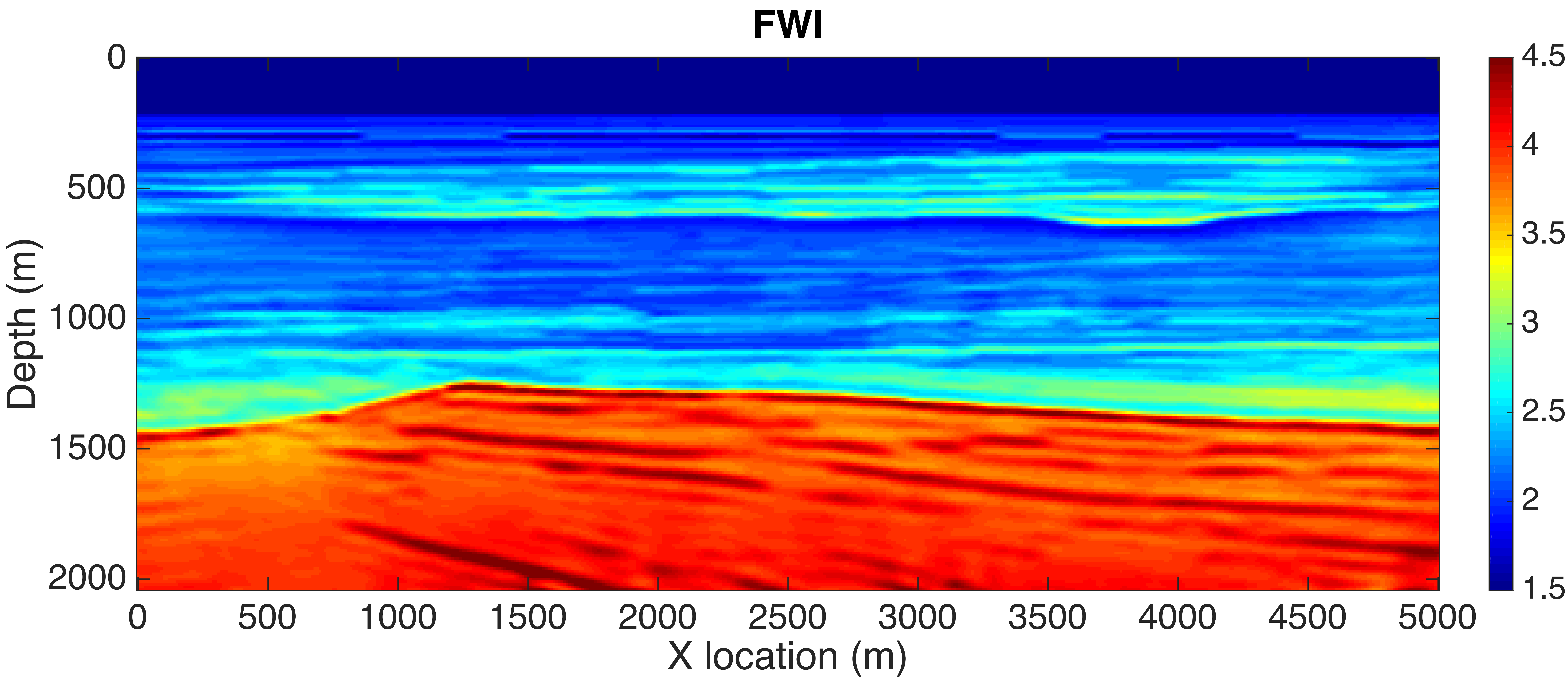
10 PQN iterations

True velocity



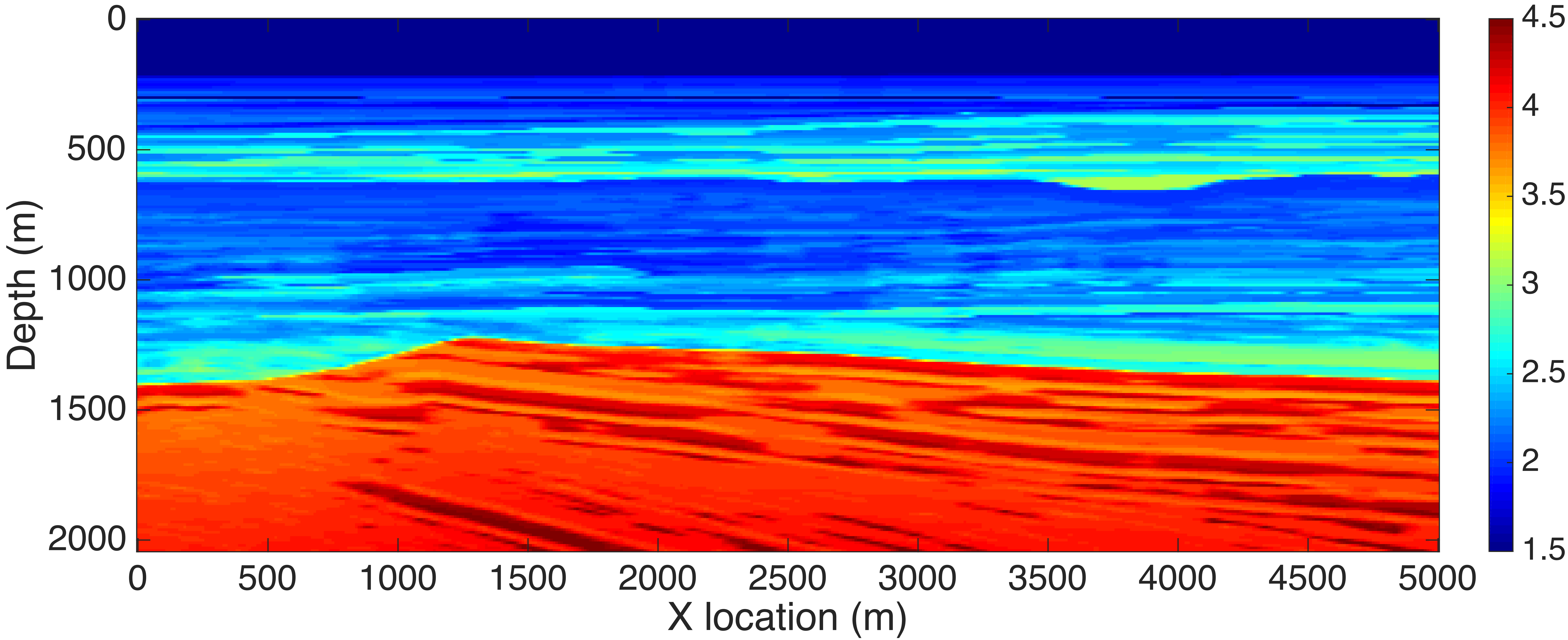
Good initial model



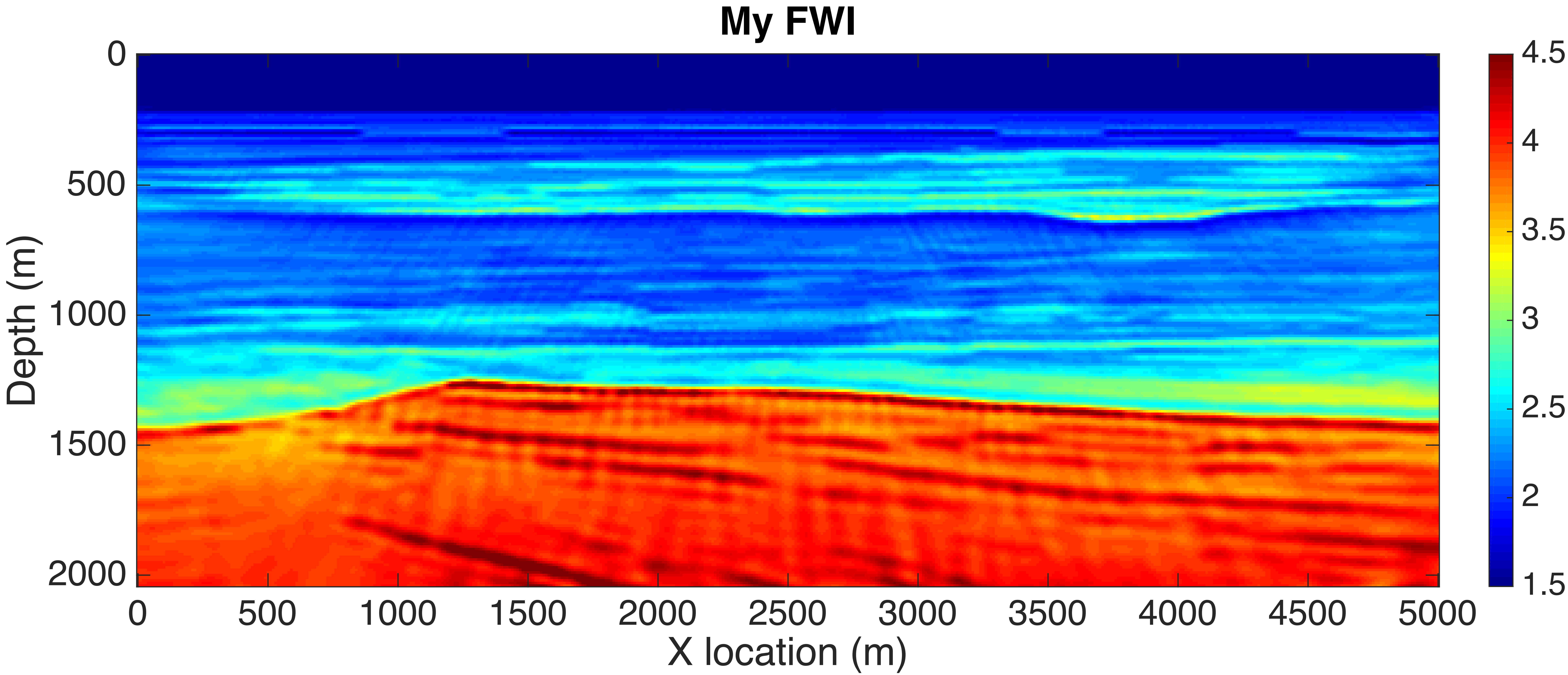


FWI
Depth (m)
X location (m)
0
500
1000
1500
2000
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
4.5
4
3.5
3
2.5
2
1.5

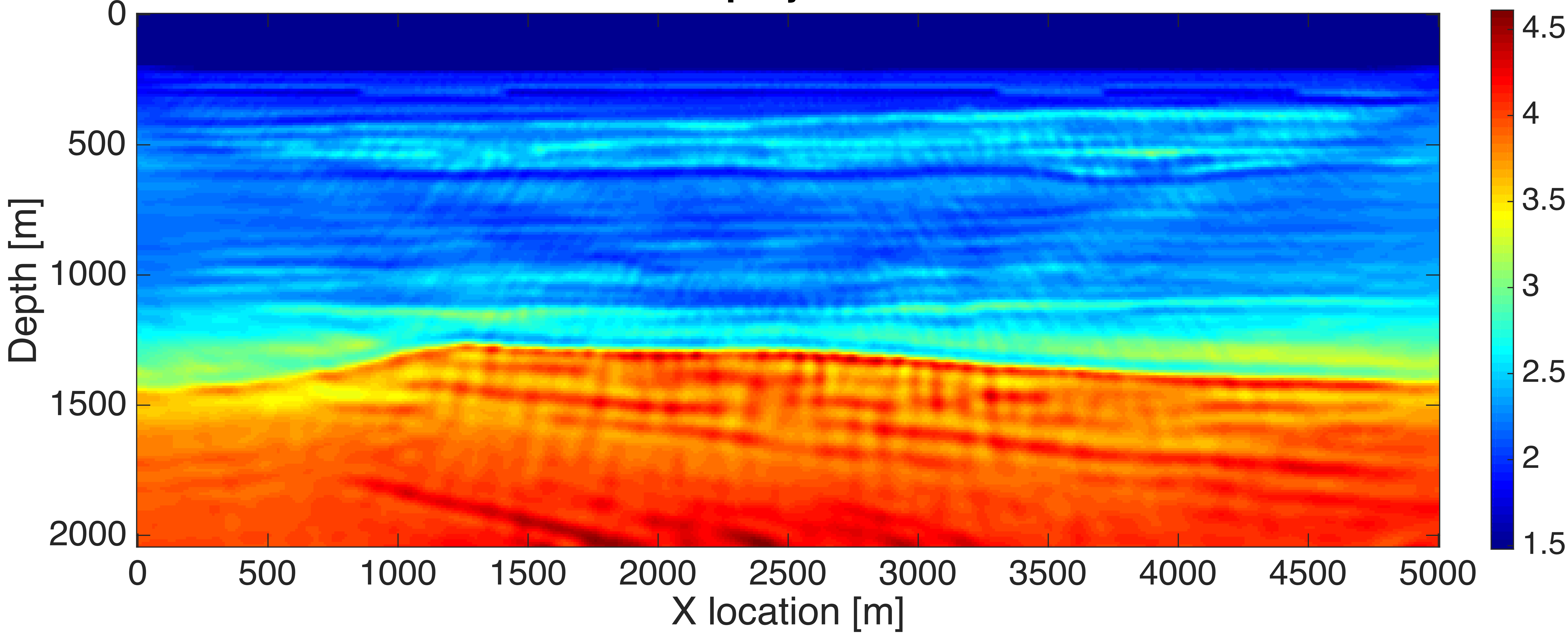
True velocity



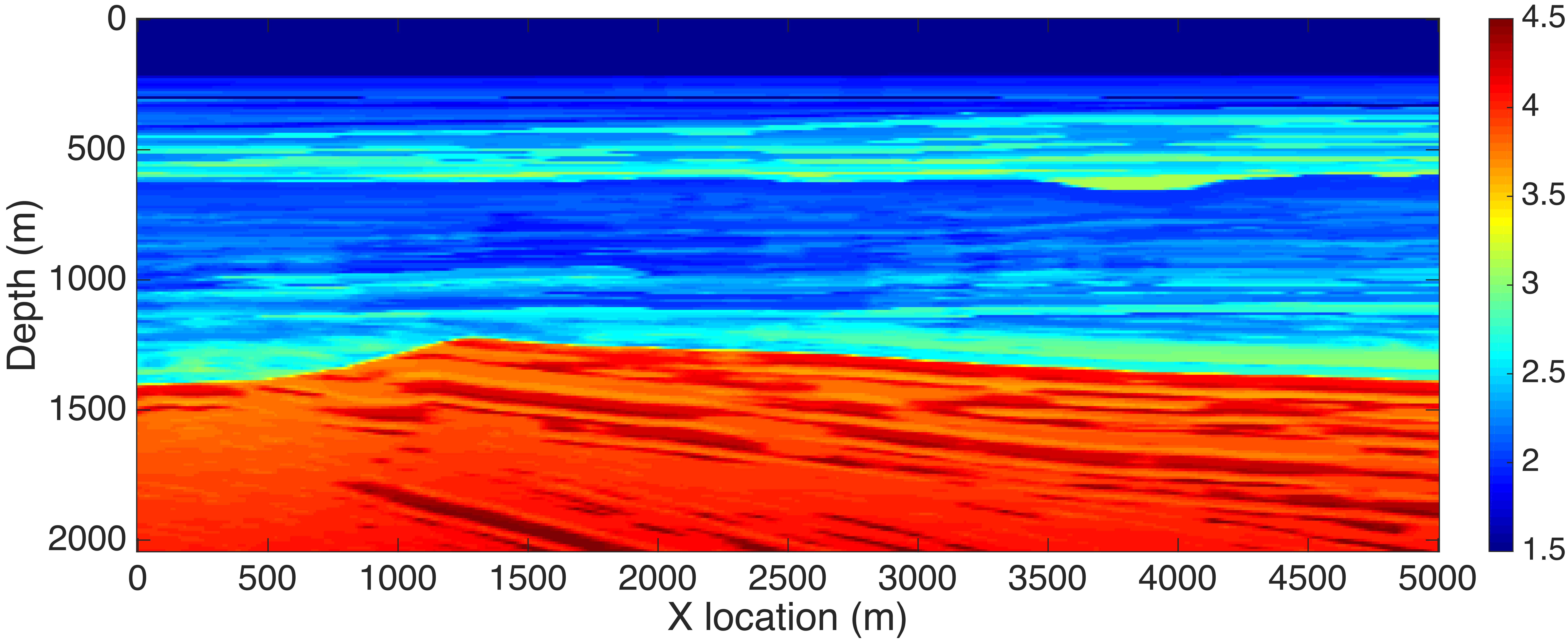
16 times less memory usage than FWI.
Same computational time (Everything in RAM)



No projection



True velocity



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; EAGE 2015). 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] (EAGE,2015). About the land dataset, uses projected quasi-Newton.

[S. Becker et. al., 2015]. (EAGE,2015) Also uses projected quasi-Newton, for projections onto a single set.

[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.

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.

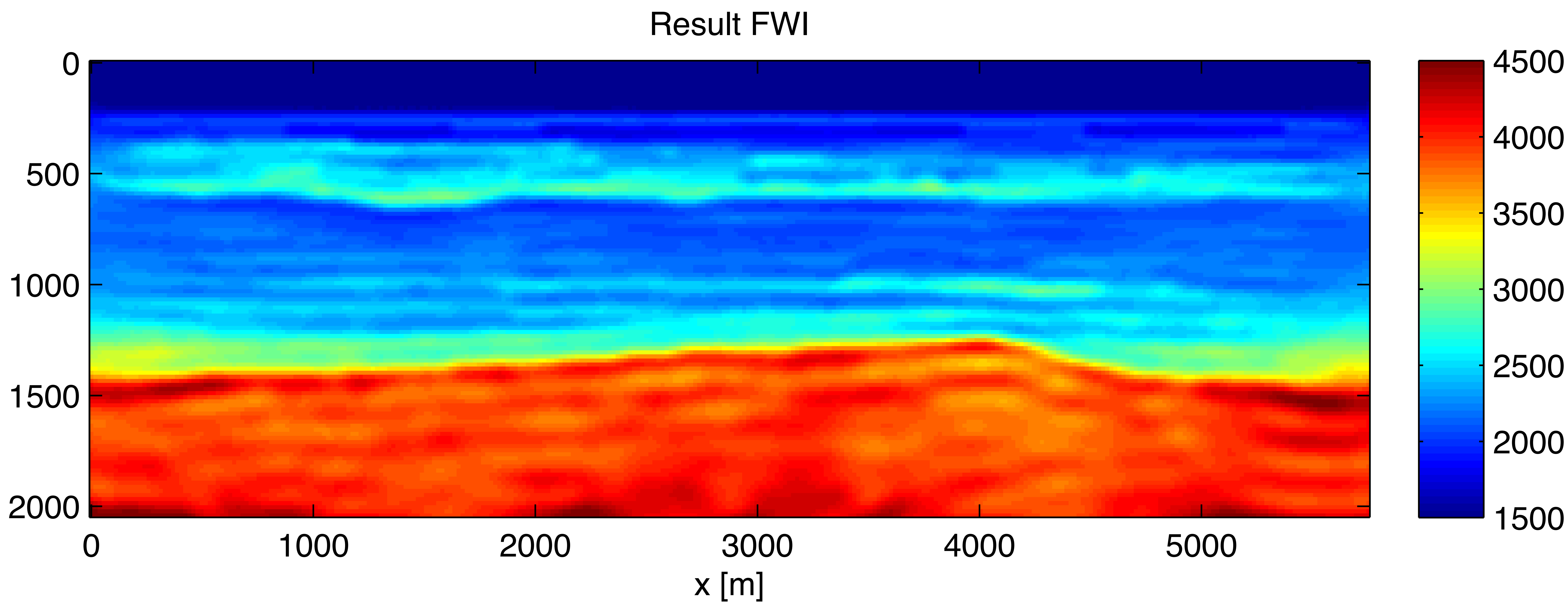
Outlook

Non-convex sets?

- intersection of convex and non-convex sets may be non-convex
- can be used as long as the projection onto the set can be computed
- if the algorithm performs well and is empirically ‘stable’

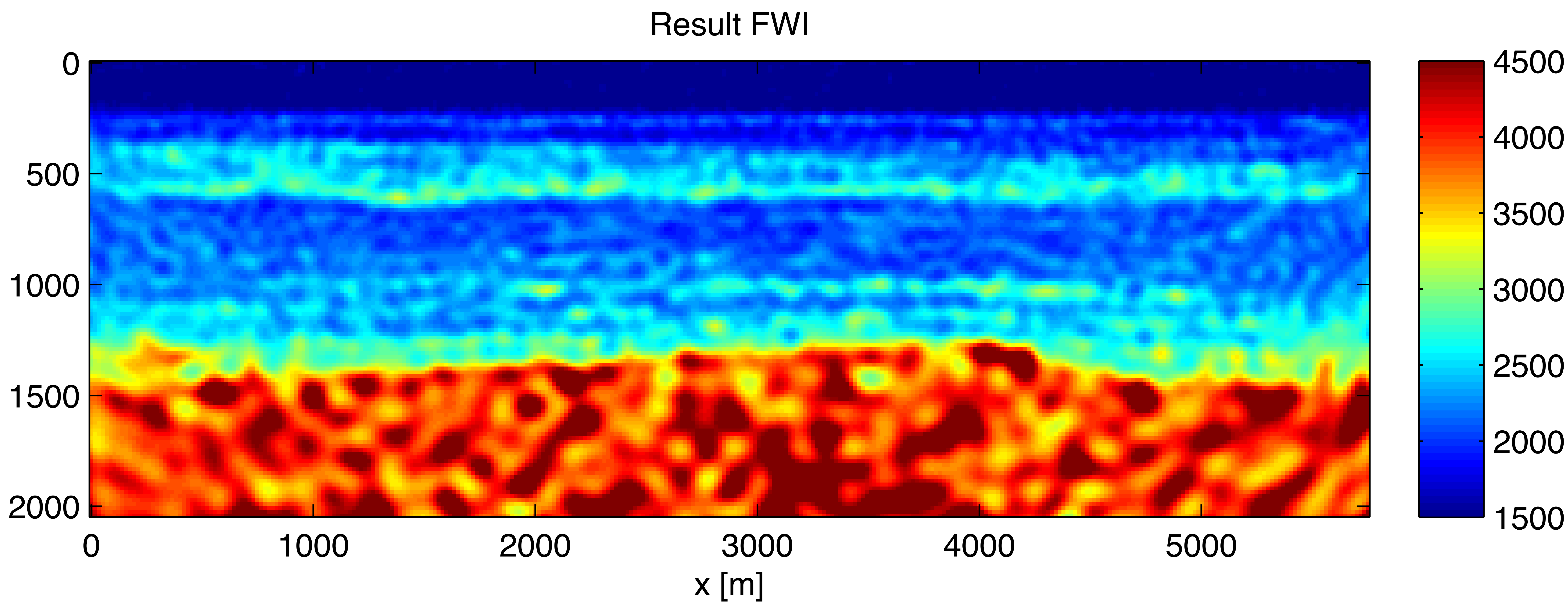
Example 1 revisited - FWI with noise

no noise in data - bound constraints only



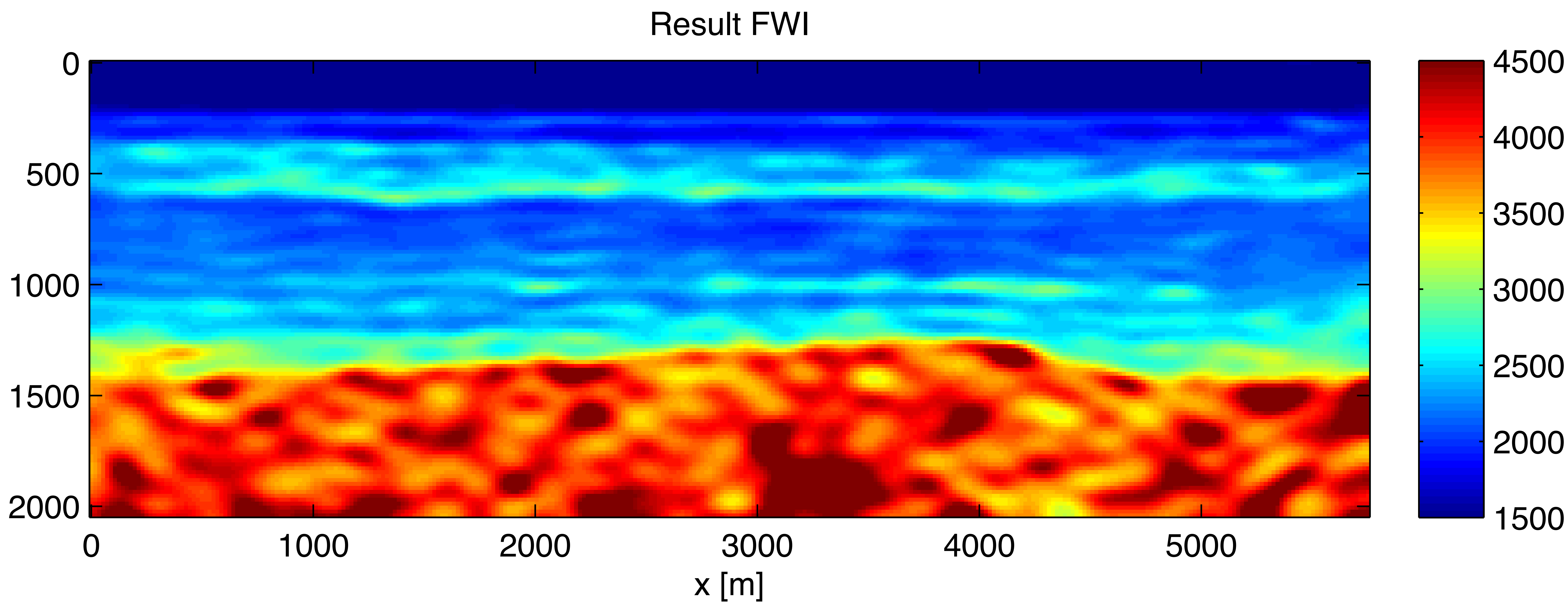
Example 1 revisited - FWI with noise $\|\text{noise}\|_2 / \|\text{signal}\|_2 = 1$

noise in data - bound constraints only



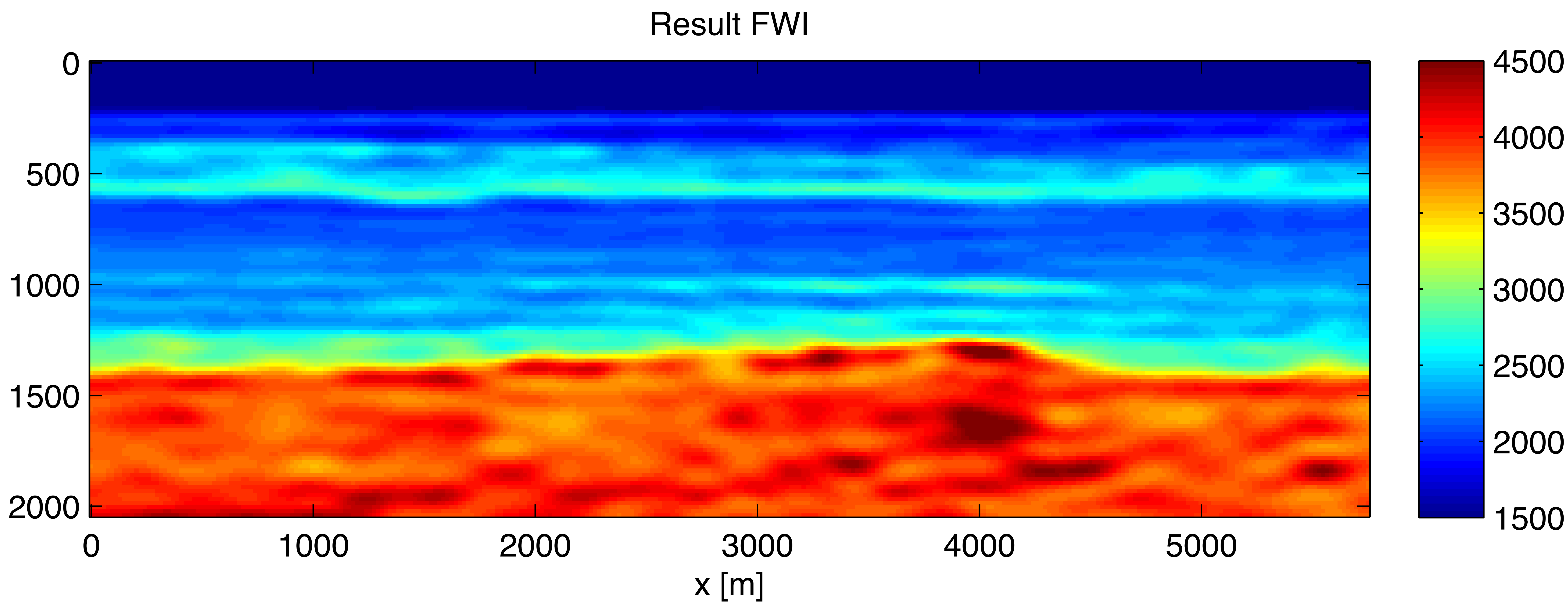
Example 1 revisited - FWI with noise $\|\text{noise}\|_2 / \|\text{signal}\|_2 = 1$

noise in data - bound constraints and minimum smoothness constraints



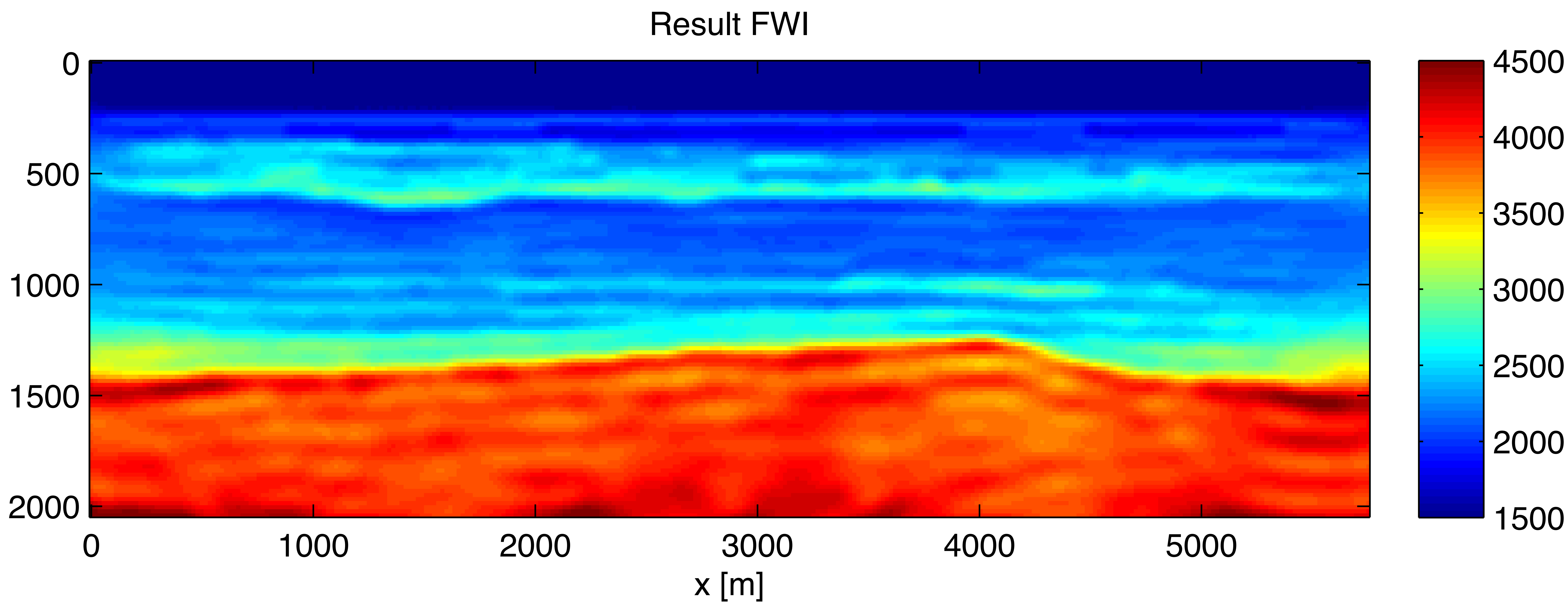
Example 1 revisited - FWI with noise $\|\text{noise}\|_2 / \|\text{signal}\|_2 = 1$

noise in data - nonconvex set (intersection of convex and nonconvex sets)



Example 1 revisited - FWI with noise

no noise in data - bound constraints only



Acknowledgements

PhD students and Postdocs at SLIM

This work was financially supported by SINBAD Consortium members BG Group, BGP, CGG, Chevron, ConocoPhillips, DownUnder GeoSolutions, Hess, Petrobras, PGS, Schlumberger, Statoil, Sub Salt Solutions and Woodside;
and by the Natural Sciences and Engineering Research Council of Canada via NSERC Collaborative Research and Development Grant DNOISEII (CRDPJ 375142--08).

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.
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).

SLIM

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

11. Ernie Esser, Tristan van Leeuwen, Aleksandr Y. Aravkin, Felix J. Herrmann. A scaled gradient projection method for total variation regularized full waveform inversion. UBC TR-EOAS-2014-2. 2014

ADMM

Problem:

$$\min_{\mathbf{m},\mathbf{z}} f(\mathbf{m}) + g(\mathbf{z}) \quad \text{s.t.} \quad A\mathbf{m} + B\mathbf{z} = \mathbf{c}$$

Form augmented-Lagrangian:

$$L_\rho(\mathbf{m},\mathbf{z},\mathbf{v}) = f(\mathbf{m}) + g(\mathbf{z}) + \mathbf{v}^*(A\mathbf{m} + B\mathbf{z} - \mathbf{c}) + \frac{\rho}{2}\|A\mathbf{m} + B\mathbf{z} - \mathbf{c}\|_2^2$$

using dual variable $\mathbf{v}$ and scalar penalty ρ

combines advantages of ‘dual ascend’ and ‘method of multipliers’

ADMM

$$L_\rho(\mathbf{m},\mathbf{z},\mathbf{v}) = f(\mathbf{m}) + g(\mathbf{z}) + \mathbf{v}^*(A\mathbf{m}+B\mathbf{z}-\mathbf{c}) + \frac{\rho}{2}\|A\mathbf{m}+B\mathbf{z}-\mathbf{c}\|_2^2$$

iterations:

$$\begin{aligned}
\mathbf{m}^{k+1} &= \arg\min_{\mathbf{m}} L_\rho(\mathbf{m},\mathbf{z}^k,\mathbf{v}^k) \\
\mathbf{z}^{k+1} &= \arg\min_{\mathbf{z}} L_\rho(\mathbf{m}^{k+1},\mathbf{z},\mathbf{v}^k) \\
\mathbf{v}^{k+1} &= \mathbf{v}^k + \rho(A\mathbf{m}^{k+1} + B\mathbf{z}^{k+1} - \mathbf{c})
\end{aligned}$$

ADMM

$$L_\rho(\mathbf{m}, \mathbf{z}, \mathbf{v}) = f(\mathbf{m}) + g(\mathbf{z}) + \mathbf{v}^*(A\mathbf{m} + B\mathbf{z} - \mathbf{c}) + \frac{\rho}{2}\|A\mathbf{m} + B\mathbf{z} - \mathbf{c}\|_2^2$$

iterations in scaled form ($\mathbf{u} = \mathbf{v}/\rho$) and after some rewriting:

$$\mathbf{m}^{k+1} = \arg\min_{\mathbf{m}} \left(f(\mathbf{m}) + \frac{\rho}{2}\|A\mathbf{m} + B\mathbf{z}^k - \mathbf{c} + \mathbf{u}^k\|_2^2 \right)$$

$$\mathbf{z}^{k+1} = \arg\min_{\mathbf{z}} \left(g(\mathbf{z}) + \frac{\rho}{2}\|A\mathbf{m}^{k+1} + B\mathbf{z} - \mathbf{c} + \mathbf{u}^k\|_2^2 \right)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + A\mathbf{m}^{k+1} + B\mathbf{z}^{k+1} - \mathbf{c})$$

ADMM applied to Projected-Newton subproblem:

$$\mathbf{y}^k = \mathbf{m}^k - (B^k)^{-1}\nabla_{\mathbf{m}} f(\mathbf{m}^k)$$

(unconstrained Newton-step)

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

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

$$= \arg\min_{\mathbf{m}} \frac{1}{2}(\mathbf{y}^k - \mathbf{m})^* B^k(\mathbf{y}^k - \mathbf{m}) + \iota_{\mathcal{C}}(\mathbf{m})$$

$$= \arg\min_{\mathbf{m},\mathbf{z}} \frac{1}{2}(\mathbf{y}^k - \mathbf{m})^* B^k(\mathbf{y}^k - \mathbf{m}) + \iota_{\mathcal{C}}(\mathbf{z}) \quad \text{s.t.} \quad \mathbf{m} = \mathbf{z}$$

ADMM applied to Projected-Newton subproblem:

ADMM solves:

$$\min_{\mathbf{m},\mathbf{z}} f(\mathbf{m}) + g(\mathbf{z}) \quad \text{s.t.} \quad A\mathbf{m} + B\mathbf{z} = \mathbf{c}$$

projected-Newton subproblem (reformulated):

$$\mathbf{m}^{k+1} = \arg\min_{\mathbf{m},\mathbf{z}} \frac{1}{2}(\mathbf{y}^k - \mathbf{m})^* B^k (\mathbf{y}^k - \mathbf{m}) + \iota_{\mathcal{C}}(\mathbf{z}) \quad \text{s.t.} \quad \mathbf{m} = \mathbf{z}$$

identify: $A = I, B = -I, \mathbf{c} = 0$

$$f(\mathbf{m}) = \frac{1}{2}(\mathbf{y}^k - \mathbf{m})^* B^k (\mathbf{y}^k - \mathbf{m})$$

$$g(\mathbf{z}) = \iota_{\mathcal{C}}(\mathbf{z})$$

ADMM applied to Projected-Newton subproblem:

Projected-Newton subproblem (reformulated):

$$\mathbf{m}^{k+1} = \arg\min_{\mathbf{m},\mathbf{z}} \frac{1}{2}(\mathbf{y}^k - \mathbf{m})^* B^k (\mathbf{y}^k - \mathbf{m}) + \iota_{\mathcal{C}}(\mathbf{z}) \quad \text{s.t.} \quad \mathbf{m} = \mathbf{z}$$

Iterations in scaled form (general):

$$\mathbf{m}^{k+1} = \arg\min_{\mathbf{m}} \left(f(\mathbf{m}) + \frac{\rho}{2}\|A\mathbf{m} + B\mathbf{z}^k - \mathbf{c} + \mathbf{u}^k\|_2^2 \right)$$

$$\mathbf{z}^{k+1} = \arg\min_{\mathbf{z}} \left(g(\mathbf{z}) + \frac{\rho}{2}\|A\mathbf{m}^{k+1} + B\mathbf{z} - \mathbf{c} + \mathbf{u}^k\|_2^2 \right)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + A\mathbf{m}^{k+1} + B\mathbf{z}^{k+1} - \mathbf{c})$$

ADMM applied to Projected-Newton subproblem:

Projected-Newton subproblem (reformulated):

$$\mathbf{m}^{k+1} = \arg\min_{\mathbf{m},\mathbf{z}} \frac{1}{2}(\mathbf{y}^k - \mathbf{m})^* B^k (\mathbf{y}^k - \mathbf{m}) + \iota_{\mathcal{C}}(\mathbf{z}) \quad \text{s.t.} \quad \mathbf{m} = \mathbf{z}$$

Iterations in scaled form (problem specific):

$$\mathbf{m}^{k+1} = \arg\min_{\mathbf{m}} \left(f(\mathbf{m}) + \frac{\rho}{2}\|\mathbf{m} - \mathbf{z}^k + \mathbf{u}^k\|_2^2 \right)$$

$$\mathbf{z}^{k+1} = \arg\min_{\mathbf{z}} \left(g(\mathbf{z}) + \frac{\rho}{2}\|\mathbf{m}^{k+1} - \mathbf{z} + \mathbf{u}^k\|_2^2 \right)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{m}^{k+1} - \mathbf{z}^{k+1}$$

ADMM applied to Projected-Newton subproblem:

Projected-Newton subproblem (reformulated):

$$\mathbf{m}^{k+1} = \arg\min_{\mathbf{m},\mathbf{z}} \frac{1}{2}(\mathbf{y}^k - \mathbf{m})^* B^k (\mathbf{y}^k - \mathbf{m}) + \iota_{\mathcal{C}}(\mathbf{z}) \quad \text{s.t.} \quad \mathbf{m} = \mathbf{z}$$

Iterations in scaled form (problem specific):

$$\mathbf{m}^{k+1} = \left(B^k + \rho I\right)^{-1}\left(B^k \mathbf{y}^k - \rho(\mathbf{u} - \mathbf{z})\right)$$

$$\mathbf{z}^{k+1} = \mathbf{prox}_{(\rho,\mathbf{g})}(\mathbf{m}^{k+1} + \mathbf{u}^k) = \Pi_{\mathcal{C}}(\mathbf{m}^{k+1} + \mathbf{u}^k)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{m}^{k+1} - \mathbf{z}^{k+1}$$