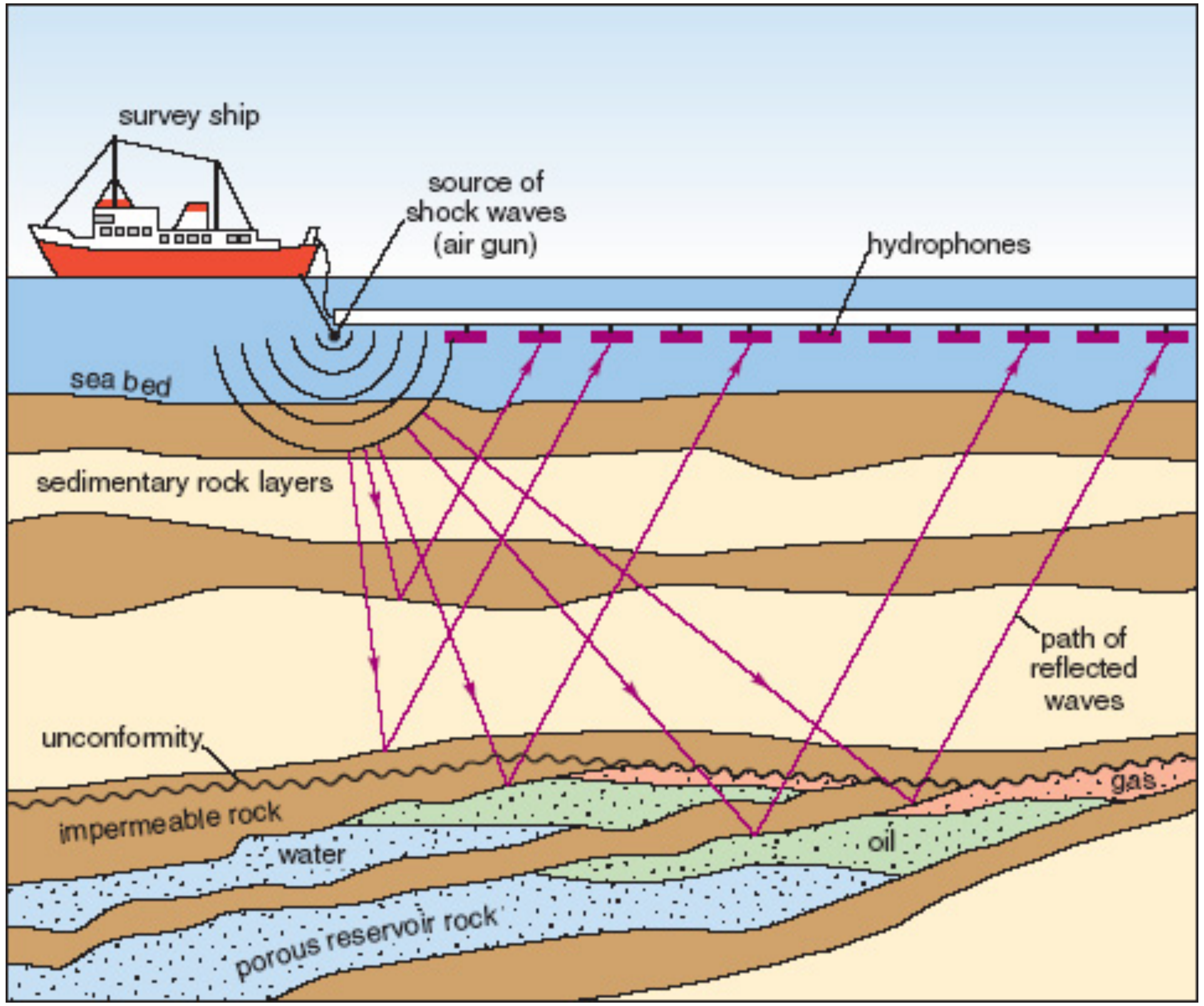
Intersections and sums of sets for the regularization of inverse problems

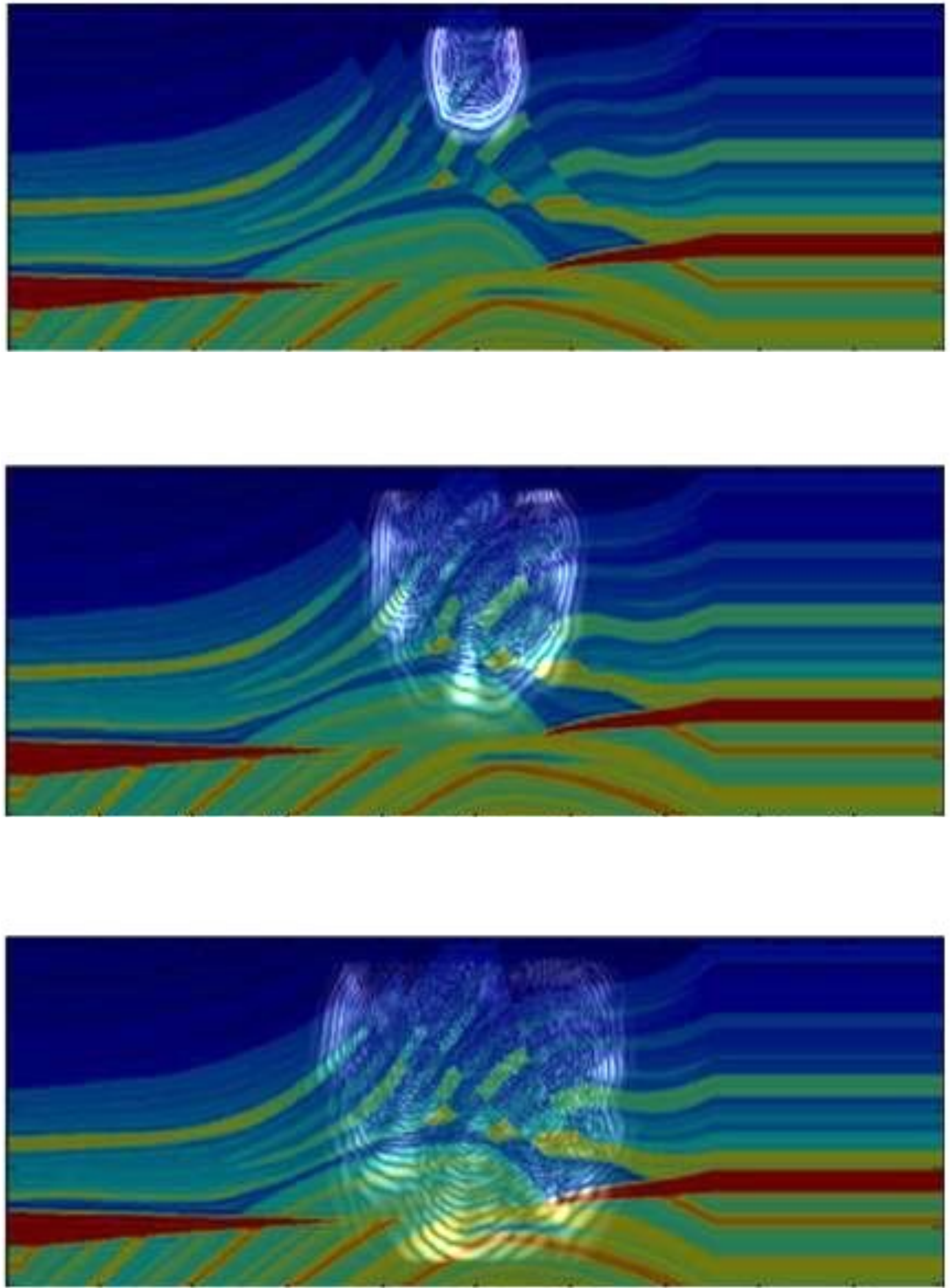
Bas Peters

Final Doctoral Examination
April 29, 2019

SLIM
University of British Columbia

Motivating problem: Seismic full-waveform inversion





Seismic full-waveform inversion (FWI)

matching observed data $d^{\text{obs}} \in \mathbb{R}^M$ to predicted data $\mathcal{F}(m) : \mathbb{R}^N \to \mathbb{R}^M$

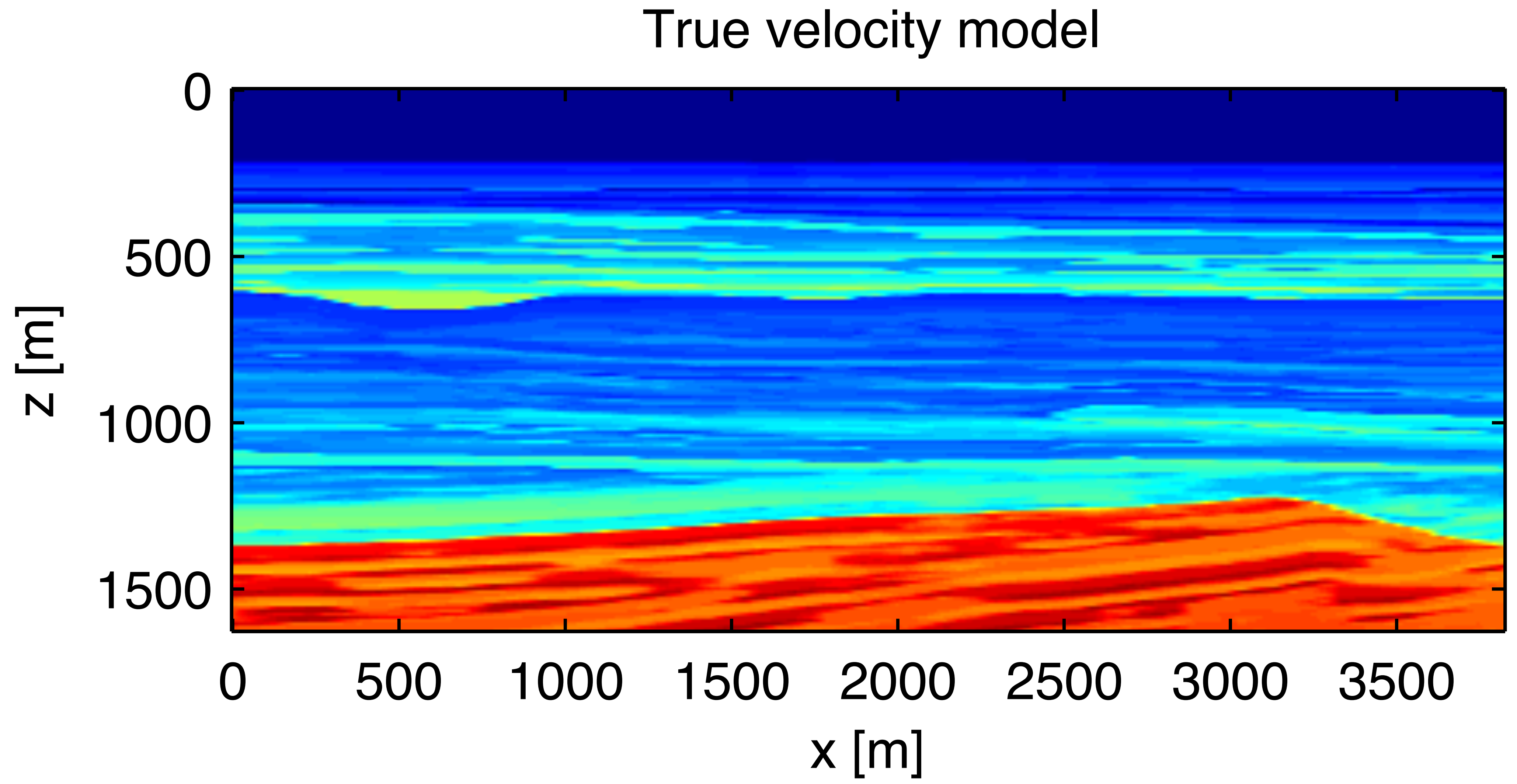
model parameters $m \in \mathbb{R}^N$ are often acoustic velocity

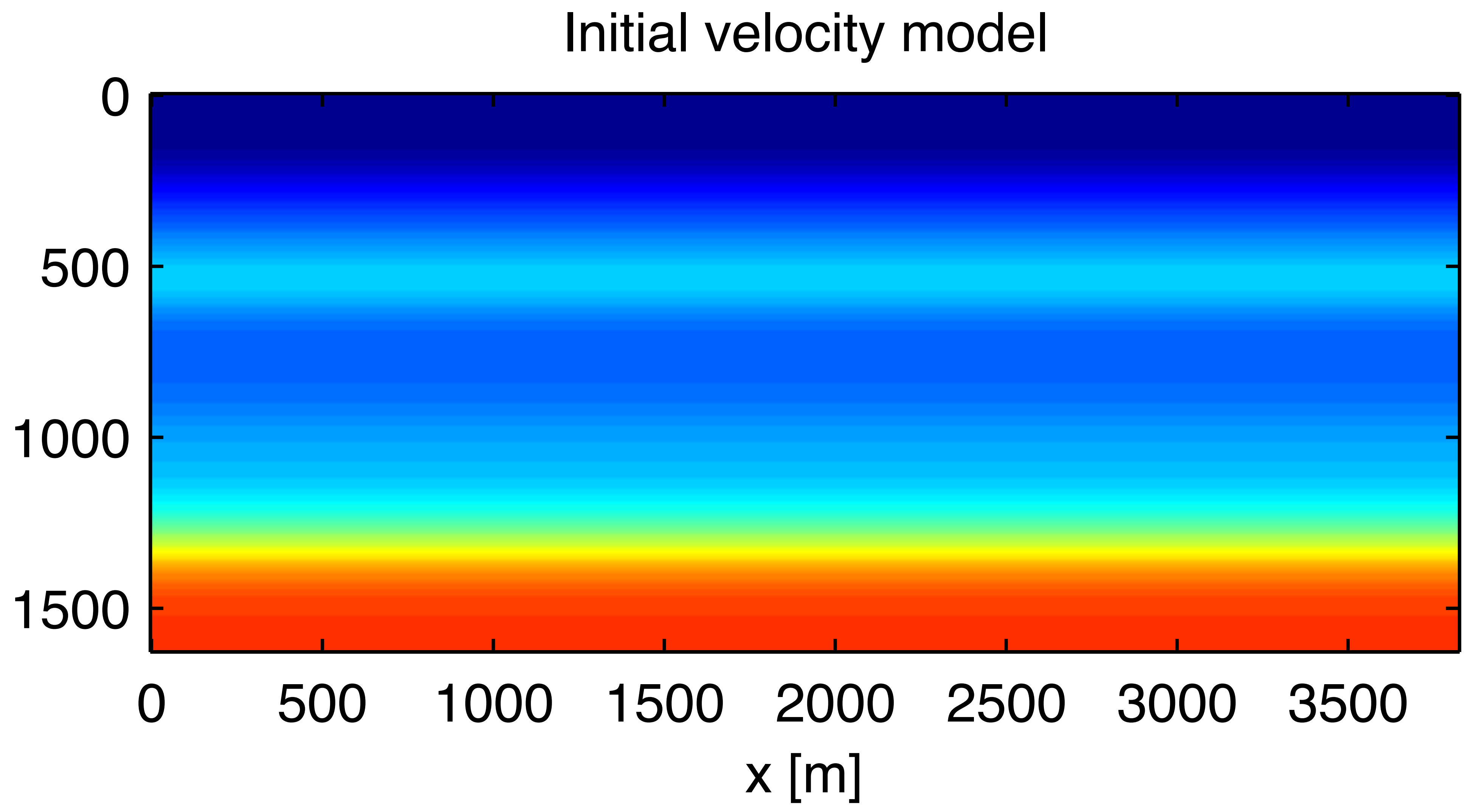
basic nonlinear inverse problem: $\min_m f(\mathcal{F}(m) - d^{\text{obs}})$

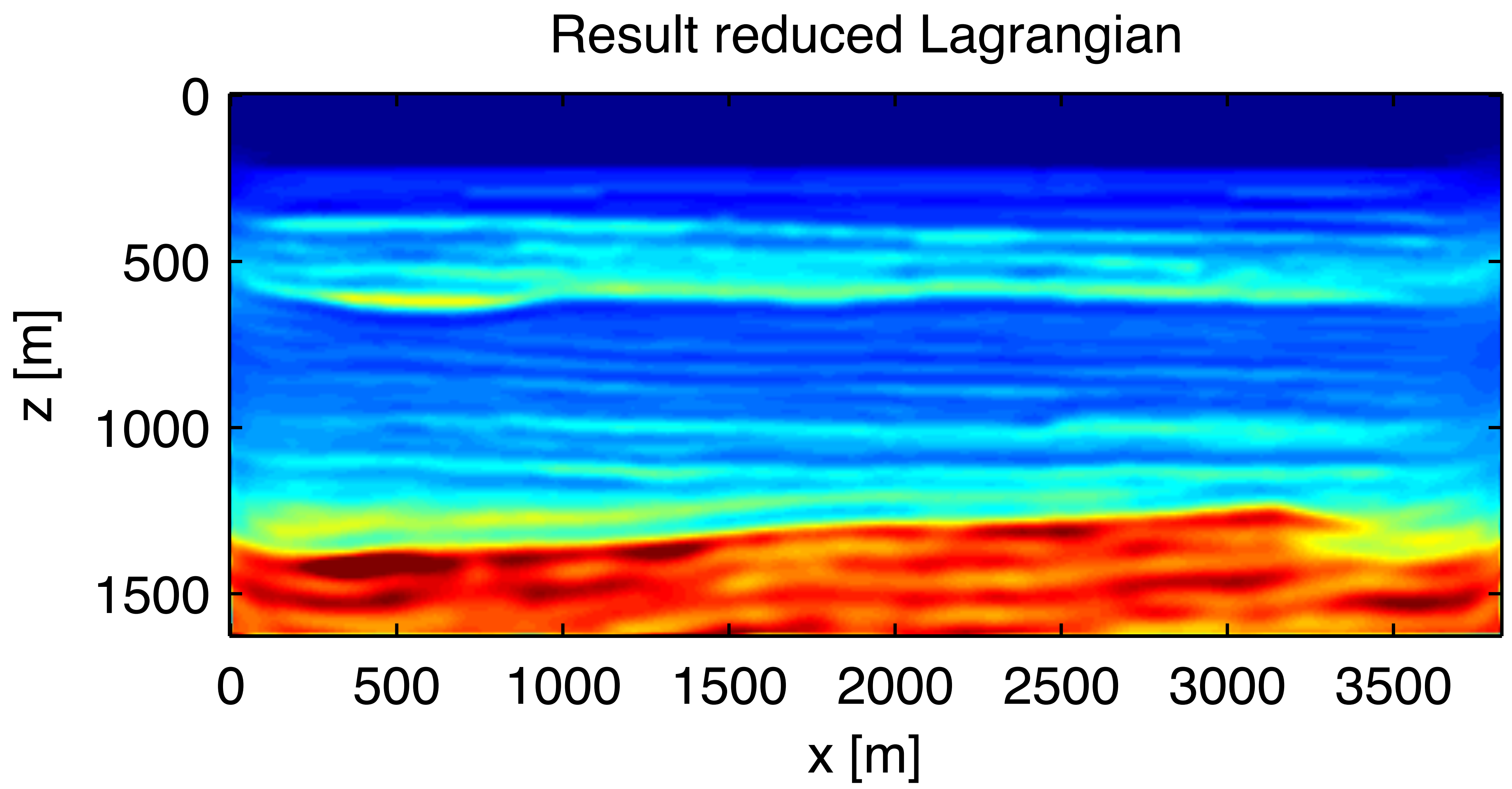
d*ata-misfit :* $f(m) : \mathbb{R}^N \to \mathbb{R}$

- non-convex
- often $f(\cdot) = \frac{1}{2}\|\cdot\|_2^2$

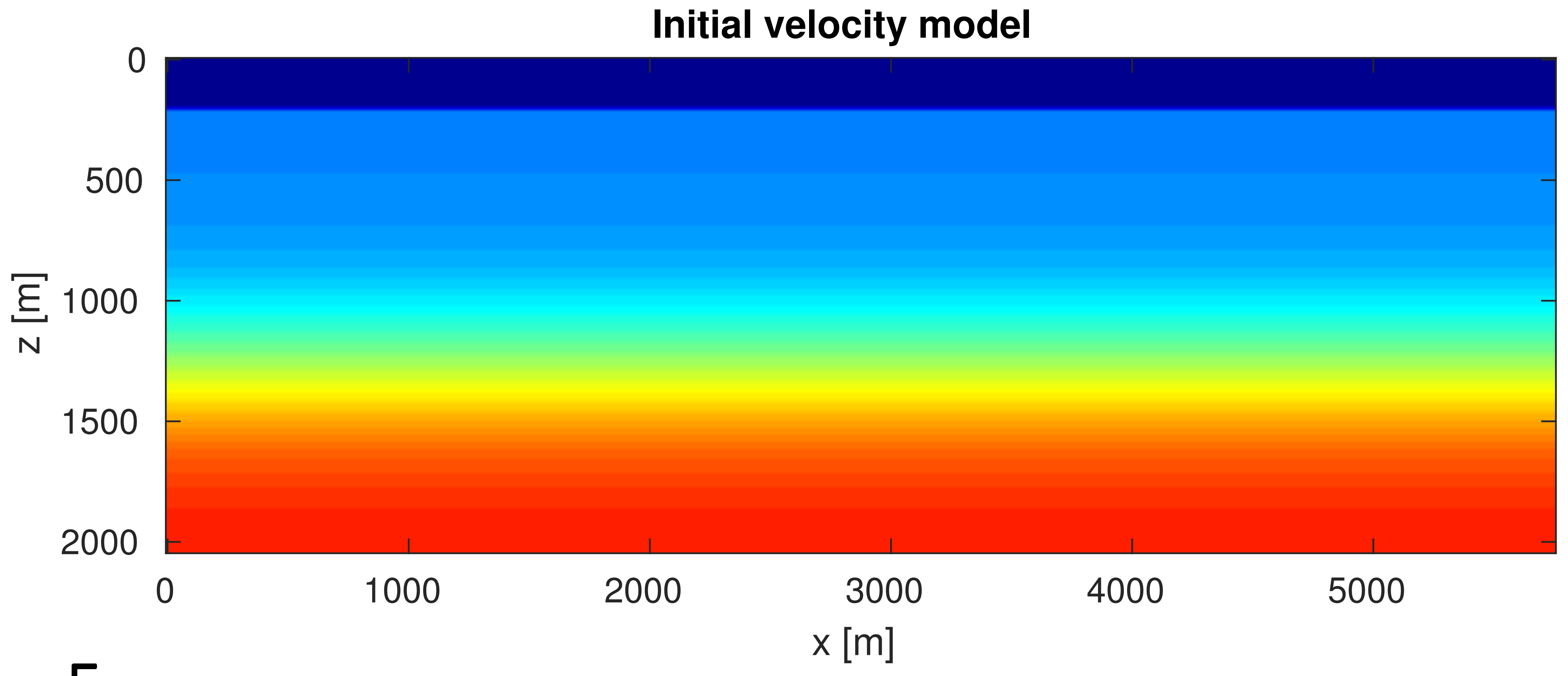
Seismic FWI - best case

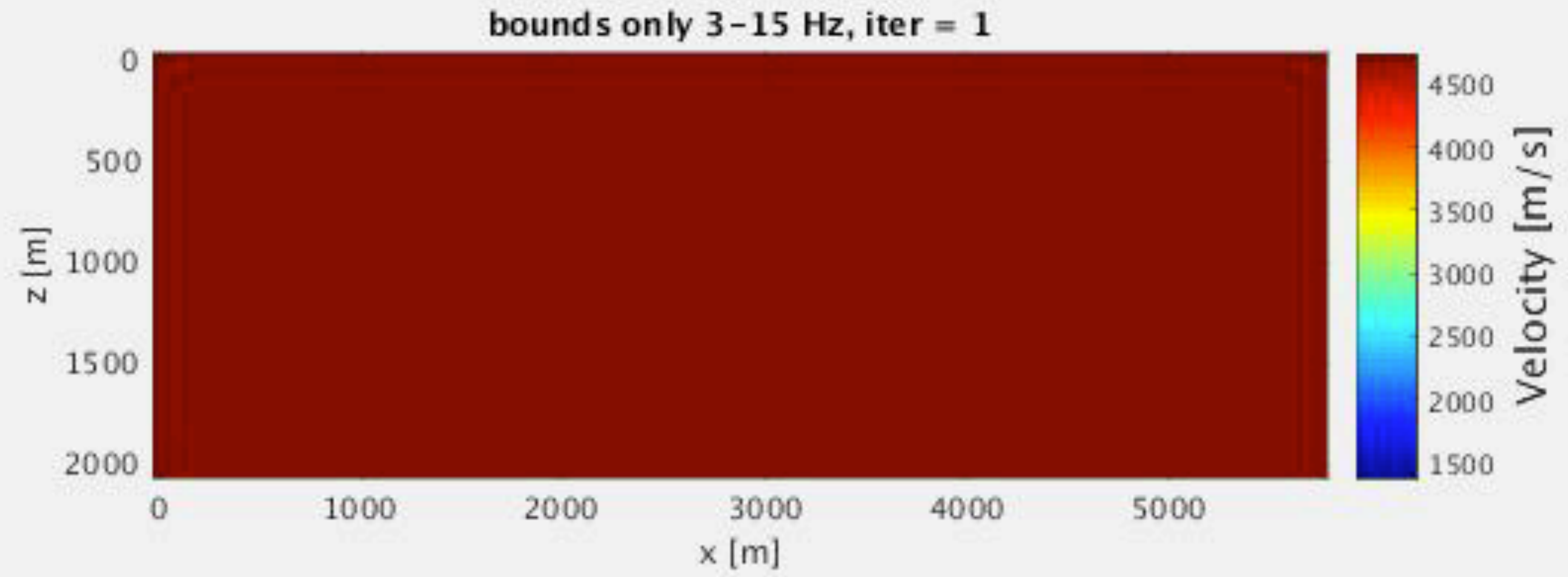






Seismic FWI - problem



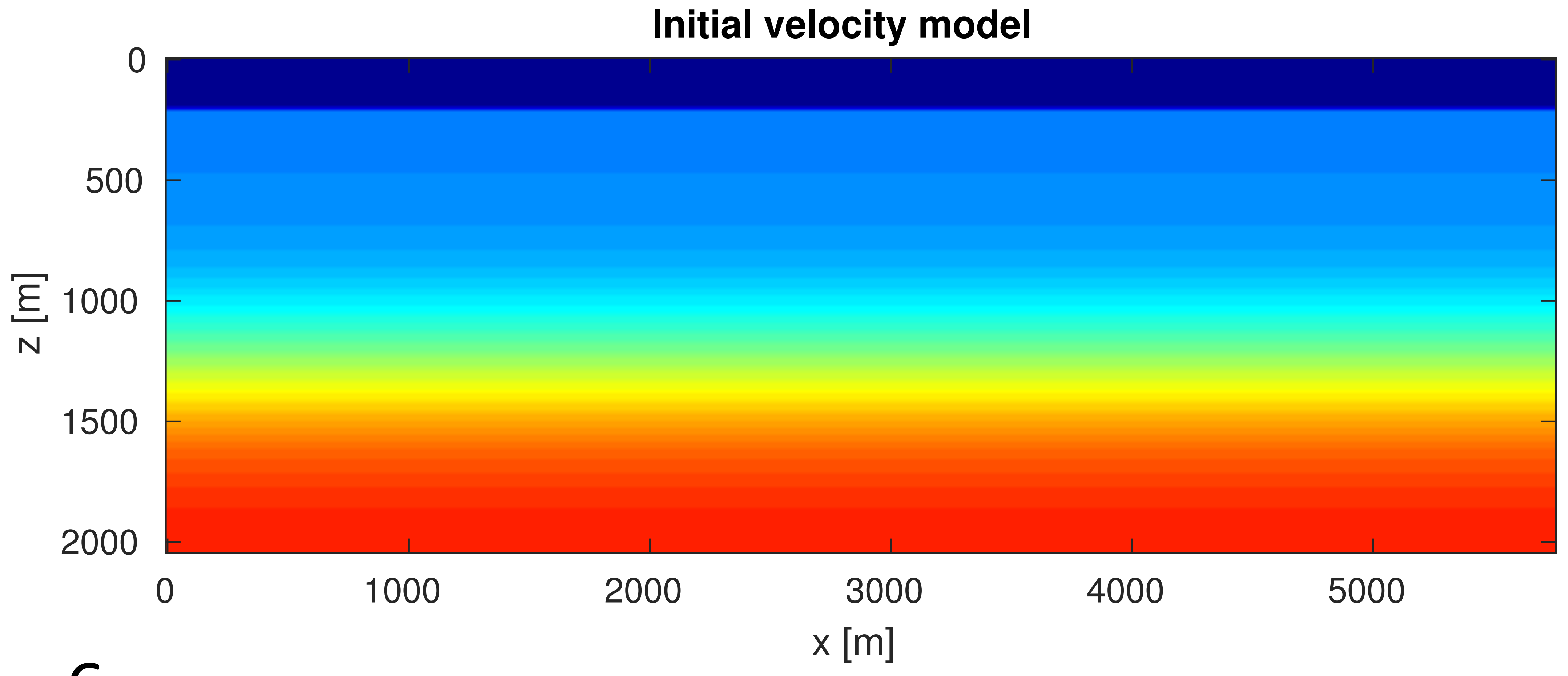


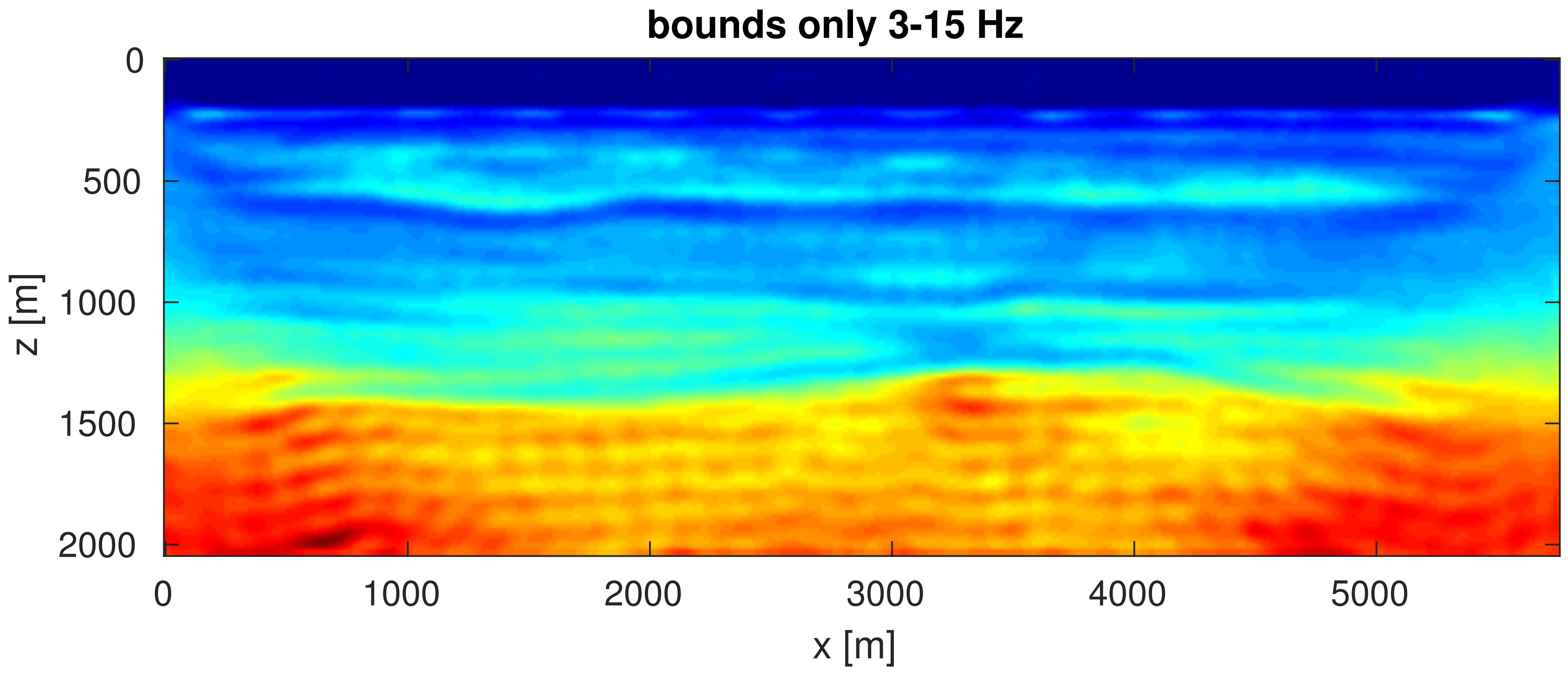
Seismic FWI - problem

problems if:

- initial guess is inaccurate
- low-frequency data is not available
- ***need regularization/prior knowledge***

Realistic, but warped and incorrect





Outline

- **constraints versus penalties (Chapter 2)**
- intersections of multiple constraint sets (Chapter 3)
- computational aspects (Chapter 4)
- sums of sets (Chapter 5)

Constraints vs. penalties

$$\min_m f(m) + \alpha r(m)$$

$r(m) : \mathbb{R}^N \to \mathbb{R}$ describes undesired model properties

the standard in geophysics

$$\min_m f(m) \quad \text{s.t.} \quad m \in \mathcal{V}$$

Constraints vs. penalties

$\min_{m} f(m) + \alpha r(m)$	$\min_{m} f(m) \quad \text{s.t.} \quad m \in \mathcal{V}$
$r(m) : \mathbb{R}^N \rightarrow \mathbb{R}$ describes undesired model properties	$\mathcal{V}$ prescribes model properties
the standard in geophysics	constraint definition closer to intuition for some properties
	constraint+projection: $m \in \mathcal{V}$ always

Constraints vs. penalties

Penalties	Constraints
$\min_m f(m) + \alpha r(m)$	$\min_m f(m) \quad \text{s.t.} \quad m \in \mathcal{V}$
$r(m) : \mathbb{R}^N \rightarrow \mathbb{R}$ describes undesired model properties	$\mathcal{V}$ prescribes model properties
the standard in geophysics	constraint definition closer to intuition for some properties
	constraint+projection: $m \in \mathcal{V}$ always

may be equivalent for specific cases

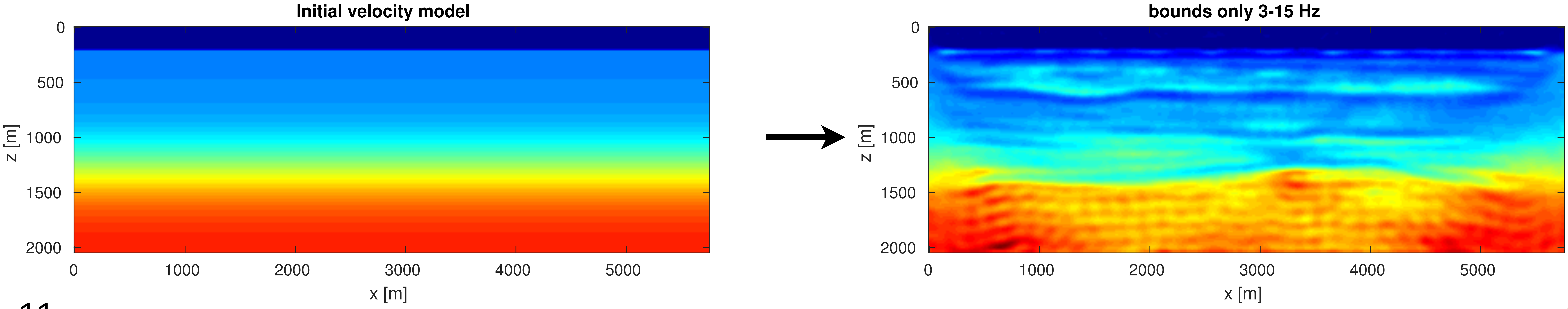
I select constraints for practical reasons.

Seismic full-waveform inversion

Adding smoothing or blockiness (TV) regularization is not enough in this case.

we need:

- new regularization strategies
- use different regularization than the usual
- more than one type of regularization simultaneously

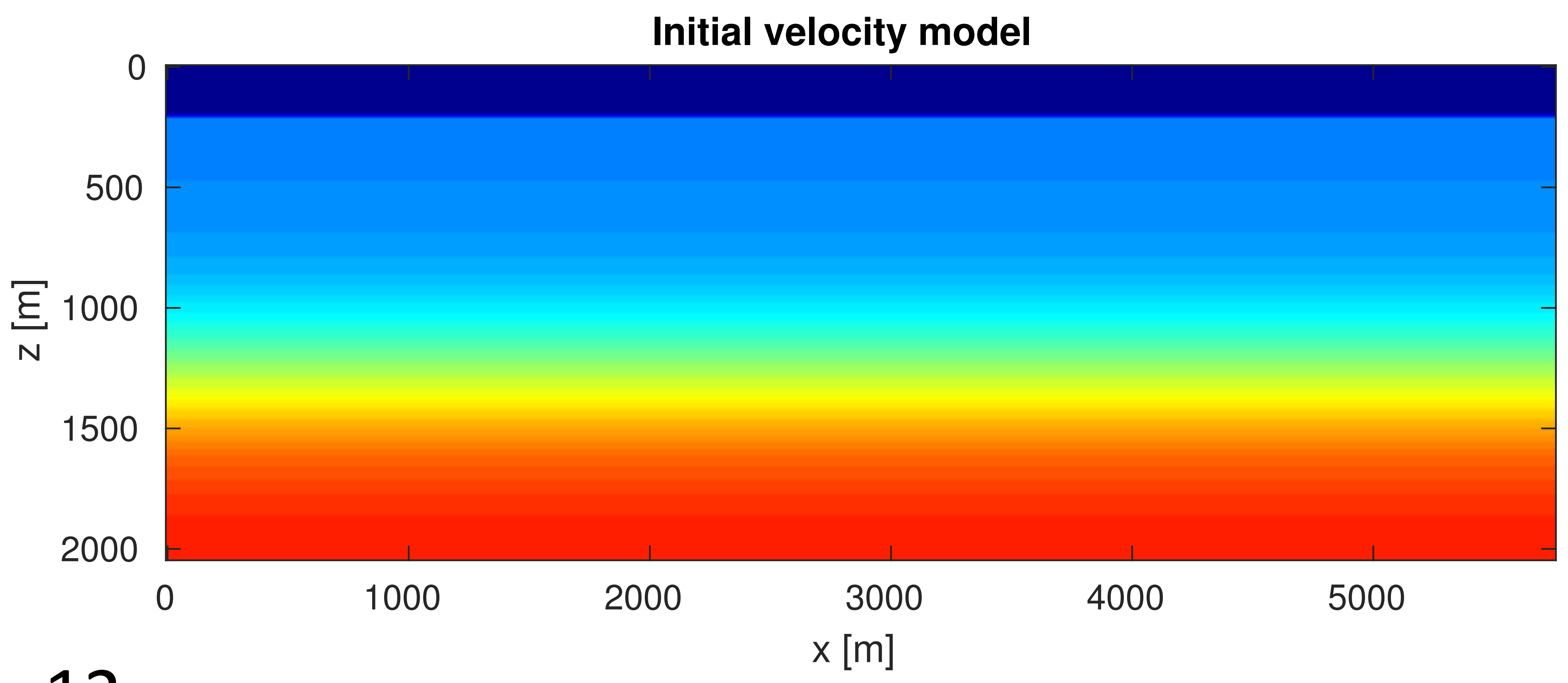


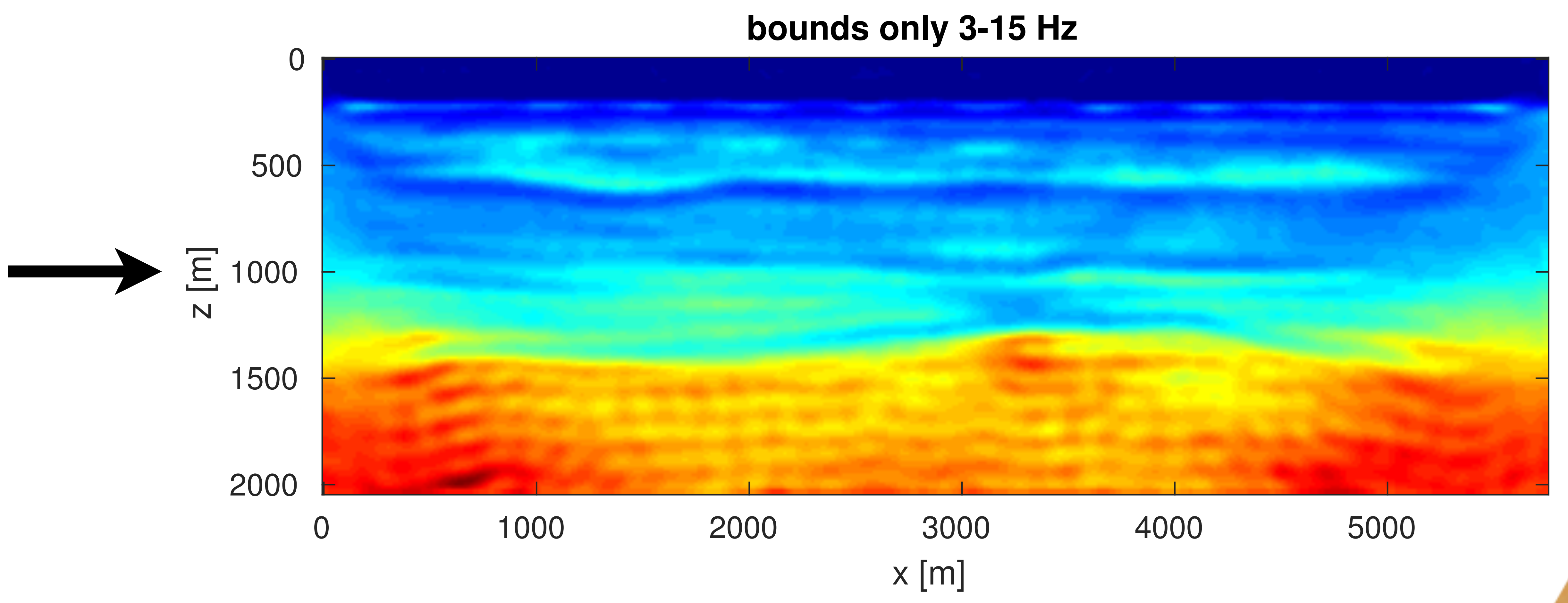
Seismic full-waveform inversion

Adding smoothing or blockiness (TV) regularization is not enough in this case.

we need:

- new regularization strategies
- use different regularization than the usual
- **more than one type of regularization simultaneously**





- constraints versus penalties (Chapter 2)
- **intersections of multiple constraint sets (Chapter 3)**
- computational aspects (Chapter 4)
- sums of sets (Chapter 5)

Multiple penalties

$$\min_{m} f(m) + \sum_{i=1}^{p} \alpha_i r_i(m)$$

- difficult to select many (2 or more) penalty parameters
- the effect of each α_i depends on all other α_i

Multiple penalties

$$\min_m f(m) + \sum_{i=1}^{p} \alpha_i r_i(m)$$

- difficult to select many (2 or more) penalty parameters
- the effect of each α_i depends on all other α_i

Geological models often have strong directionality:

- bounds
- smoothness in x, smoothness in y
- blockiness in depth

Multiple constraints

$$\min_{m} f(m) \quad \text{s.t.} \quad m \in \bigcap_{i=1}^{p} \mathcal{V}_i$$

geophysical applications:

- single $\mathcal{V}$ (bounds) [Zeev et al. (2006) and Bello and Raydan (2007)]
- two sets [Lelièvre and Oldenburg (2009), Baumstein (2013), Smithyman et al. (2015), Esser et al. (2015ab, 2016ab), Peters and Herrmann (2017), Yong et al. (2018), Trinh et al. (2018)]

Multiple constraints

$$\min_{m} f(m) \quad \text{s.t.} \quad m \in \bigcap_{i=1}^{p} \mathcal{V}_i$$

geophysical applications:

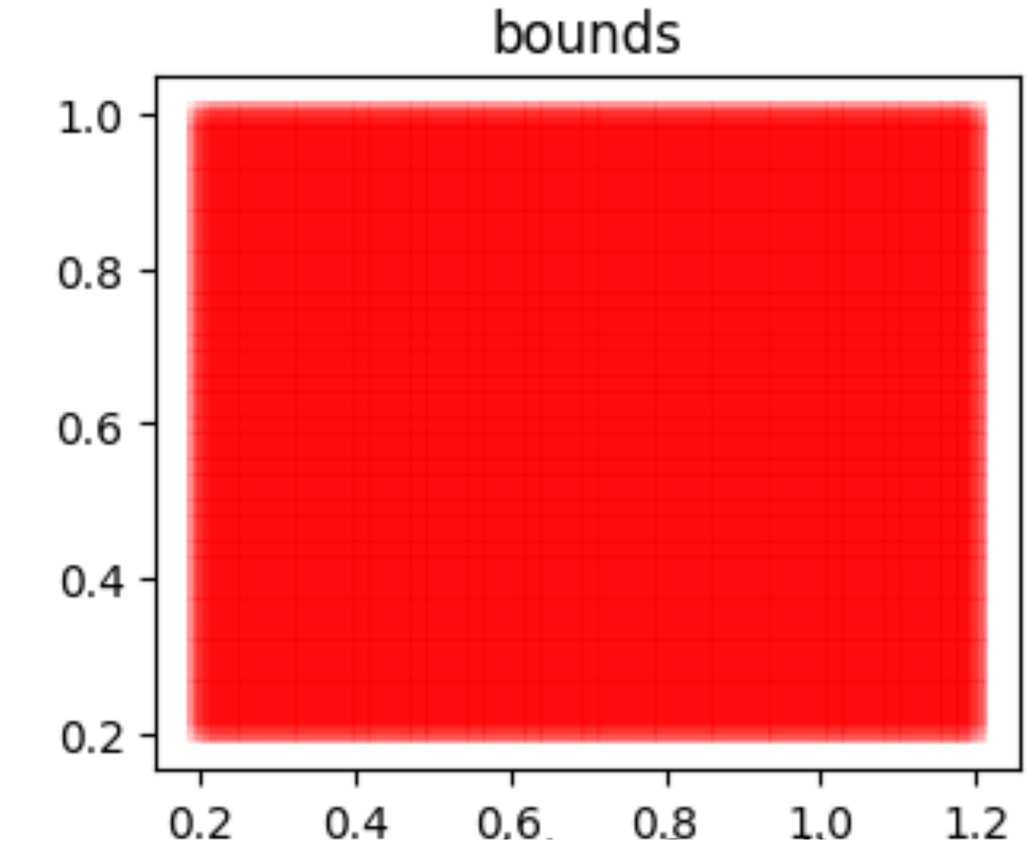
- single $\mathcal{V}$ (bounds) [Zeev et al. (2006) and Bello and Raydan (2007)]
- two sets [Lelièvre and Oldenburg (2009), Baumstein (2013), Smithyman et al. (2015), Esser et al. (2015ab, 2016ab), Peters and Herrmann (2017), Yong et al. (2018), Trinh et al. (2018)]

challenges:

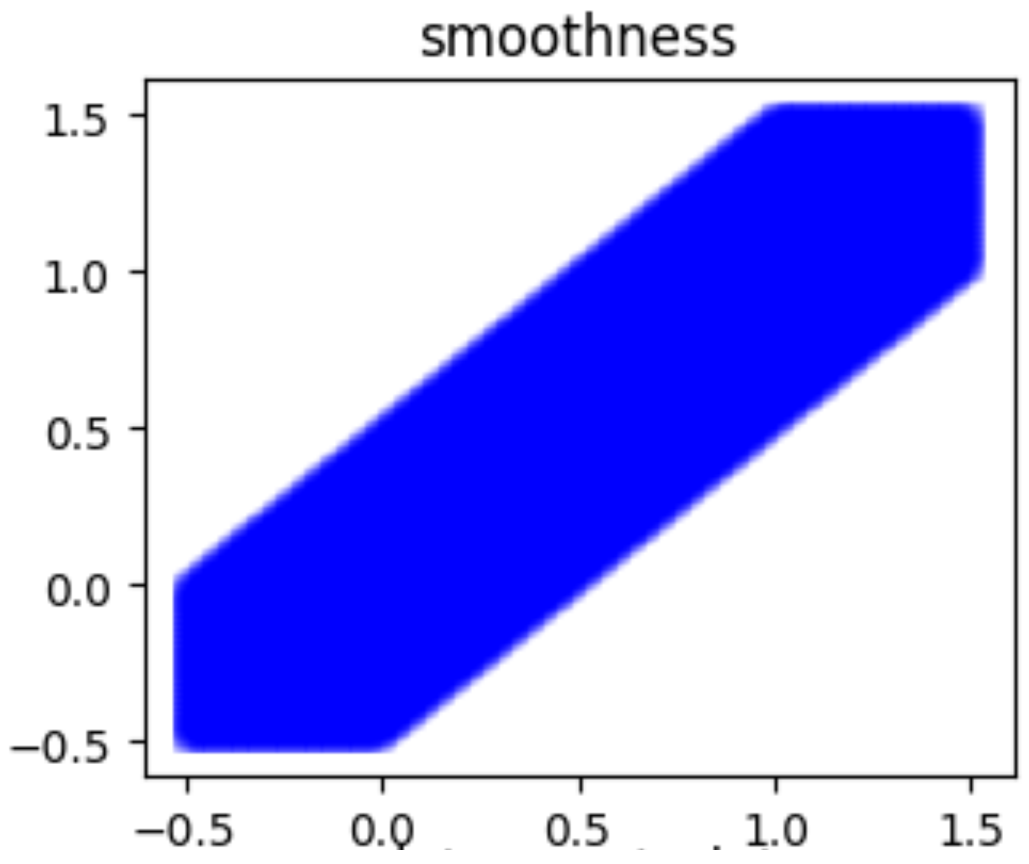
- difficult > 2 sets and projectors not known in closed form
- not suitable to plug in arbitrary sets

Multiple constraints
(2 parameter problem)

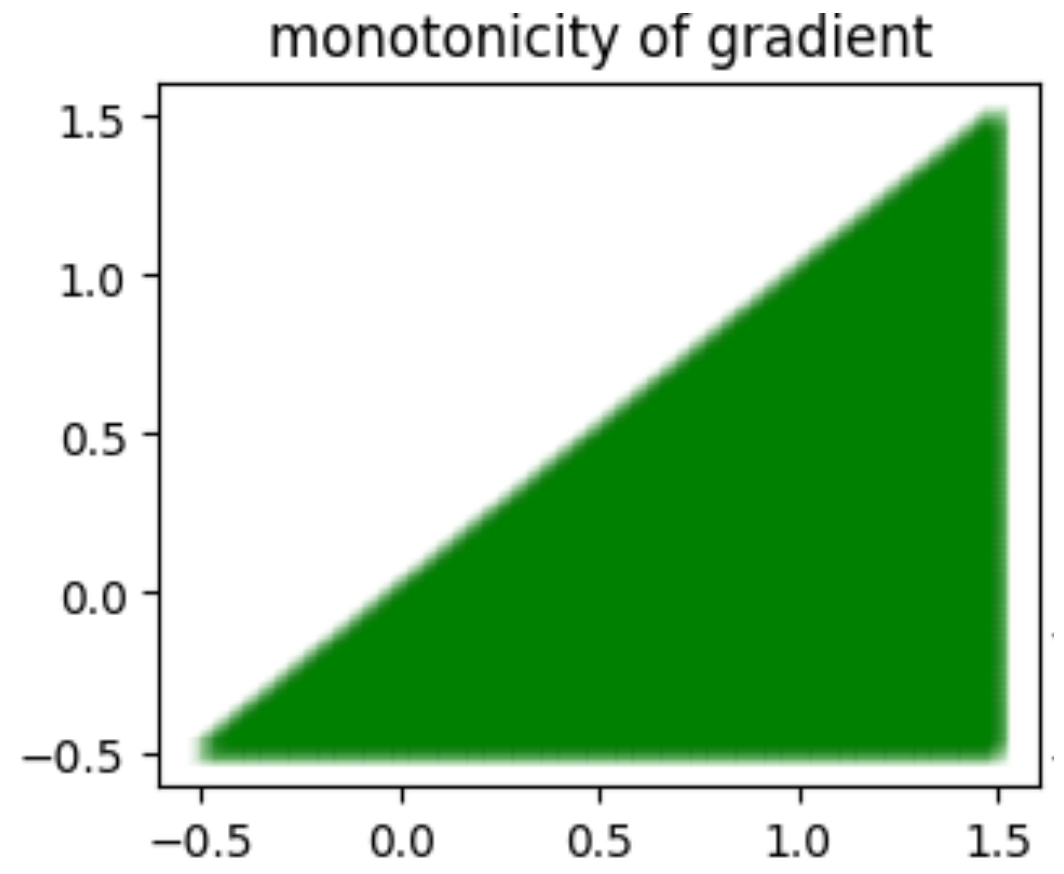
$$\mathcal{V}_1 \equiv \{m \,\|\, l \leq m \leq u\}$$



$$\mathcal{V}_2 \equiv$$
$$\{m \,\|\, -\varepsilon \leq m_1 - m_2 \leq +\varepsilon\}$$

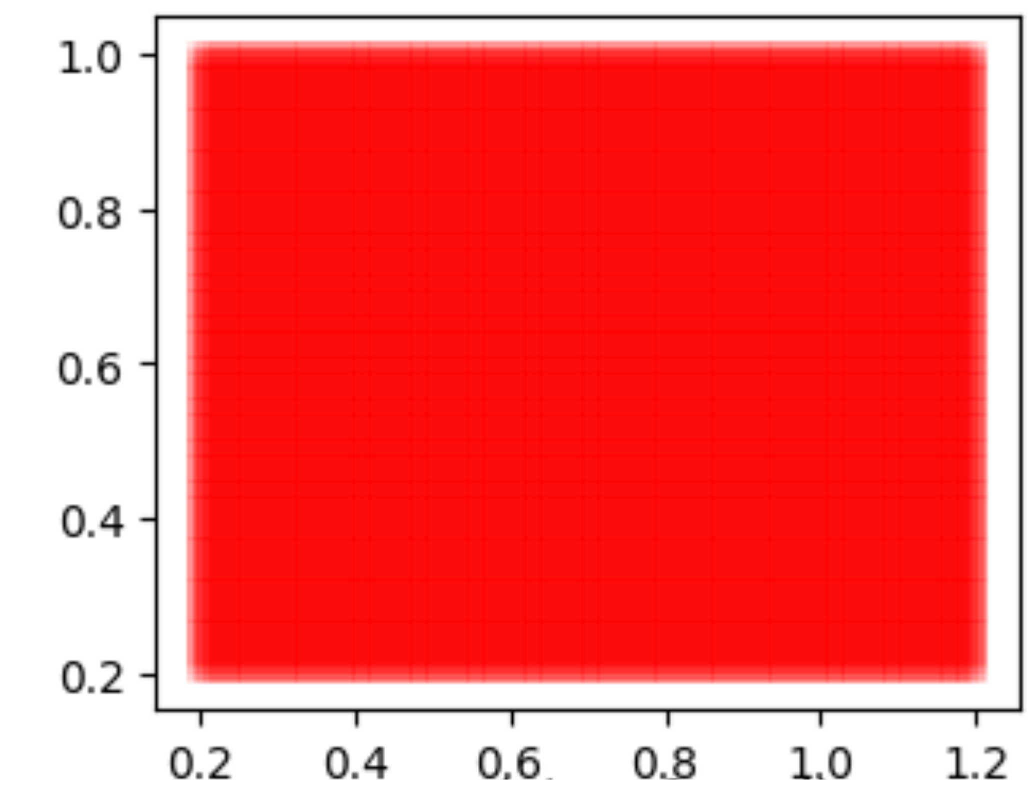


$$\mathcal{V}_3 \equiv$$
$$\{m \,\|\, 0 \leq m_1 - m_2 \leq +\infty\}$$

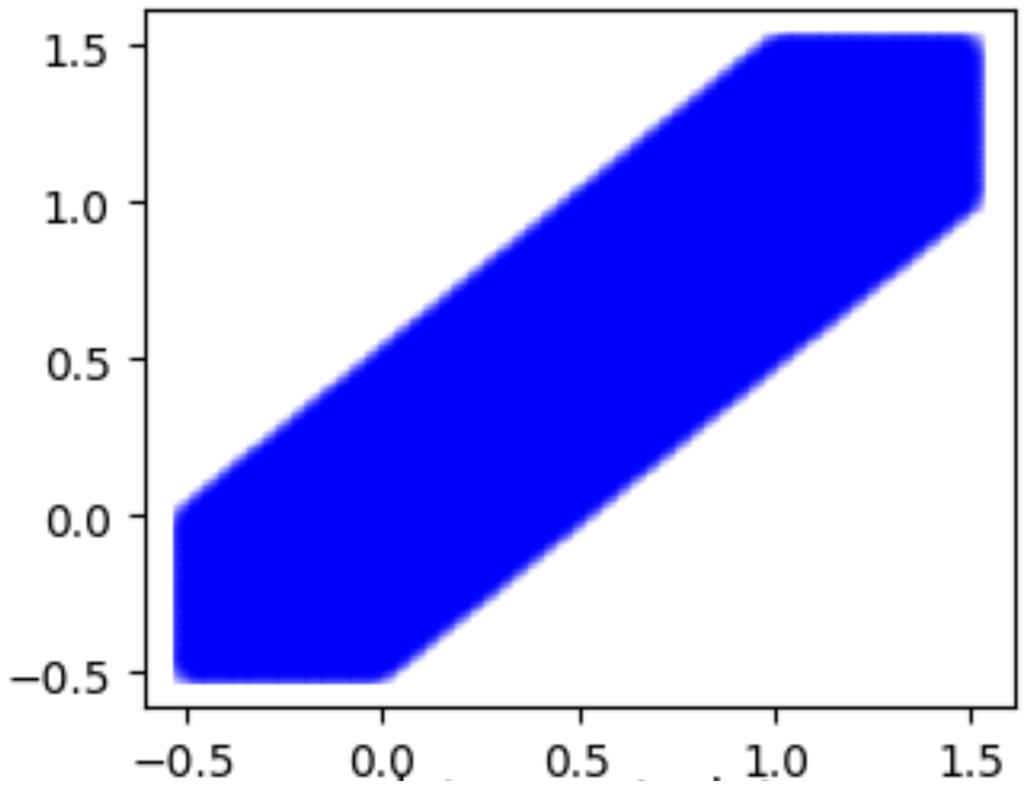


Multiple constraints
(2 parameter problem)

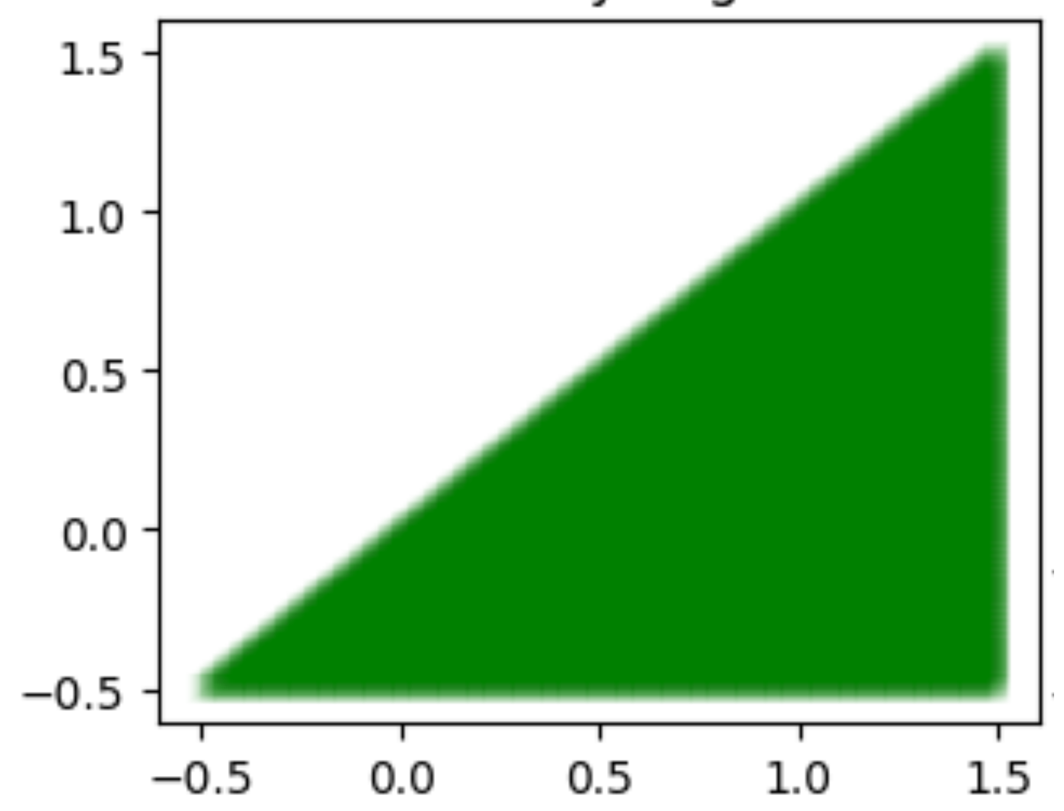
$$\mathcal{V}_1 \equiv \{m \parallel l \leq m \leq u\}$$

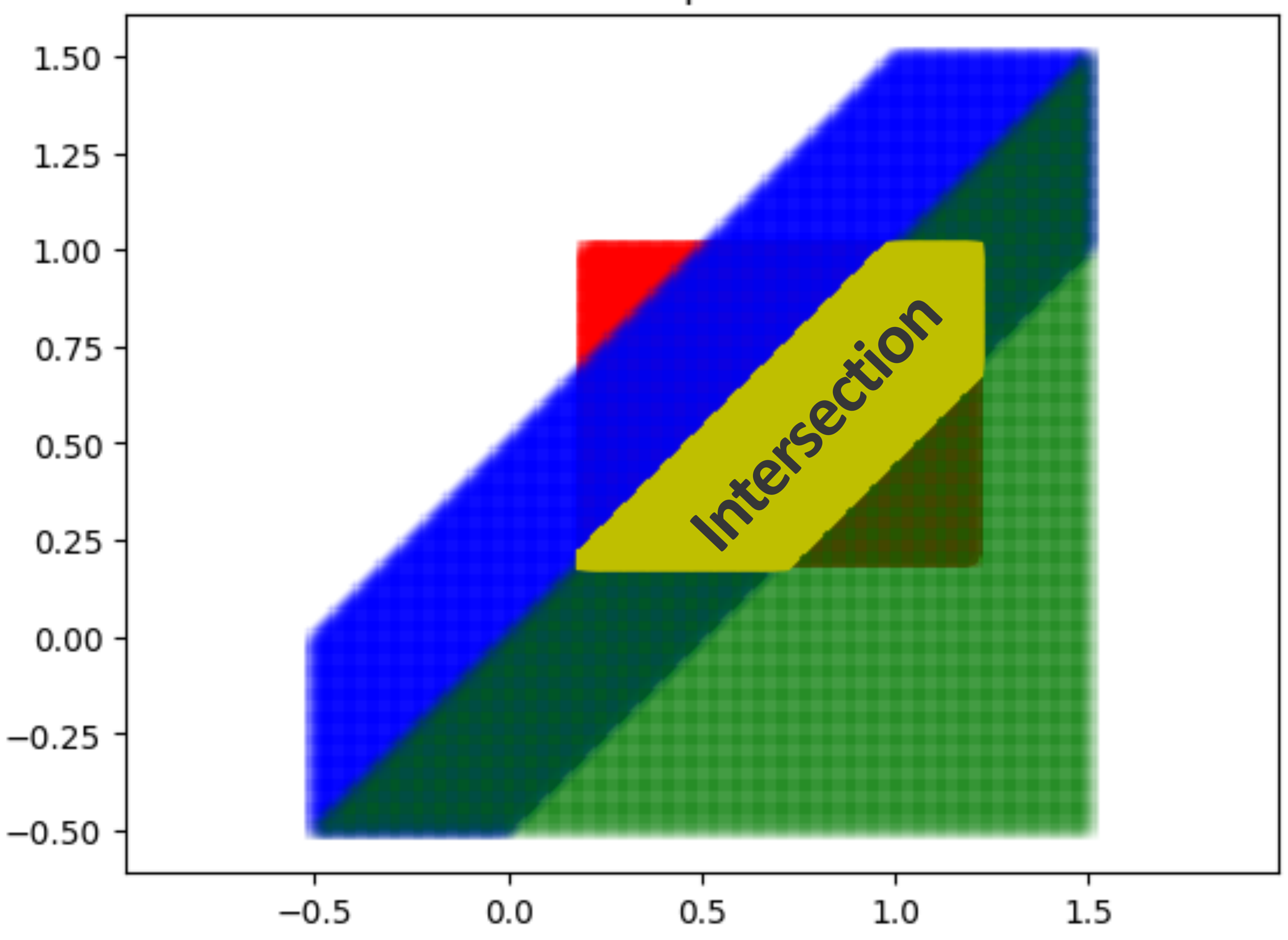


$$\mathcal{V}_2 \equiv \{m \parallel -\varepsilon \leq m_1 - m_2 \leq +\varepsilon\}$$



$$\mathcal{V}_3 \equiv \{m \parallel 0 \leq m_1 - m_2 \leq +\infty\}$$





$$m \in \bigcap_{i=1}^{p} \mathcal{V}_i$$

Multiple constraints

$$\min_{m} f(m) \quad \text{s.t.} \quad m \in \bigcap_{i=1}^{p} \mathcal{V}_i$$

Projection based algorithms:

- guarantee that m satisfies *all* constraints at *every* iteration.

[Birgin et. al. (1999); Schmidt et. al. (2009); Schmidt et. al. (2012)]

Multiple constraints

$$\min_{m} f(m) \quad \text{s.t.} \quad m \in \bigcap_{i=1}^{p} \mathcal{V}_i$$

Main iterations for gradient descent + projections (SPG):

$$m^{k+1} = (1-\gamma)m^k - \gamma \mathcal{P}_{\mathcal{V}}(m^k - \beta \nabla_m f(m^k))$$

β : Barzilai-Borwein scaling

γ : non-monotone line search step length

Projection onto an intersection

$$\mathcal{P}_{\mathcal{V}}(m) = \arg\min_{x} \|x - m\|_2$$

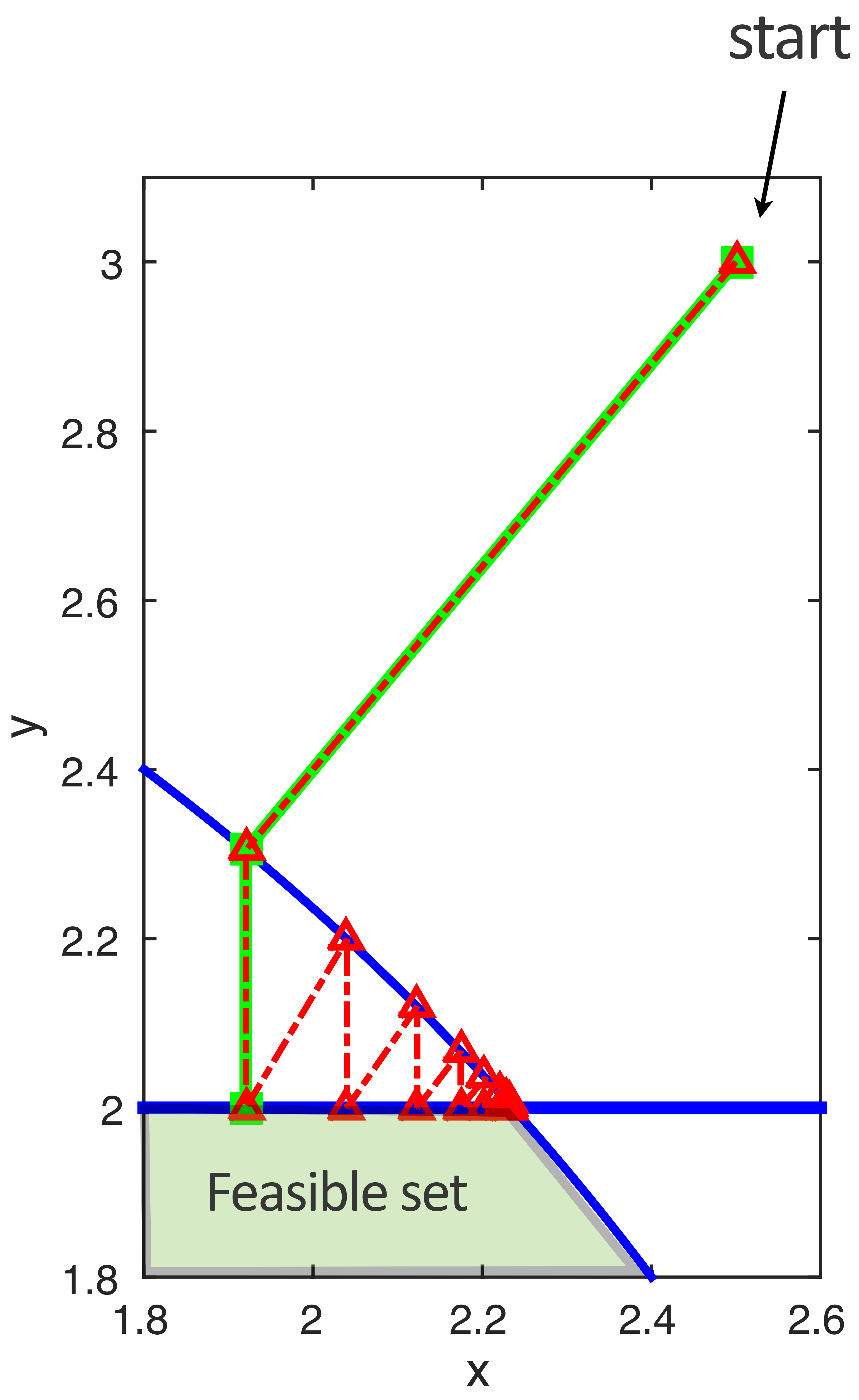
$$\text{s.t.} \;\; \in \bigcap_{i=1}^{p} \mathcal{V}_i.$$

(parallel) black-box algorithms
e.g., Dykstra's algorithm

[Dykstra, 1983 ; Boyle & Dykstra, 1986 ; Censor, 2006; Bauschke & Koch, 2015]

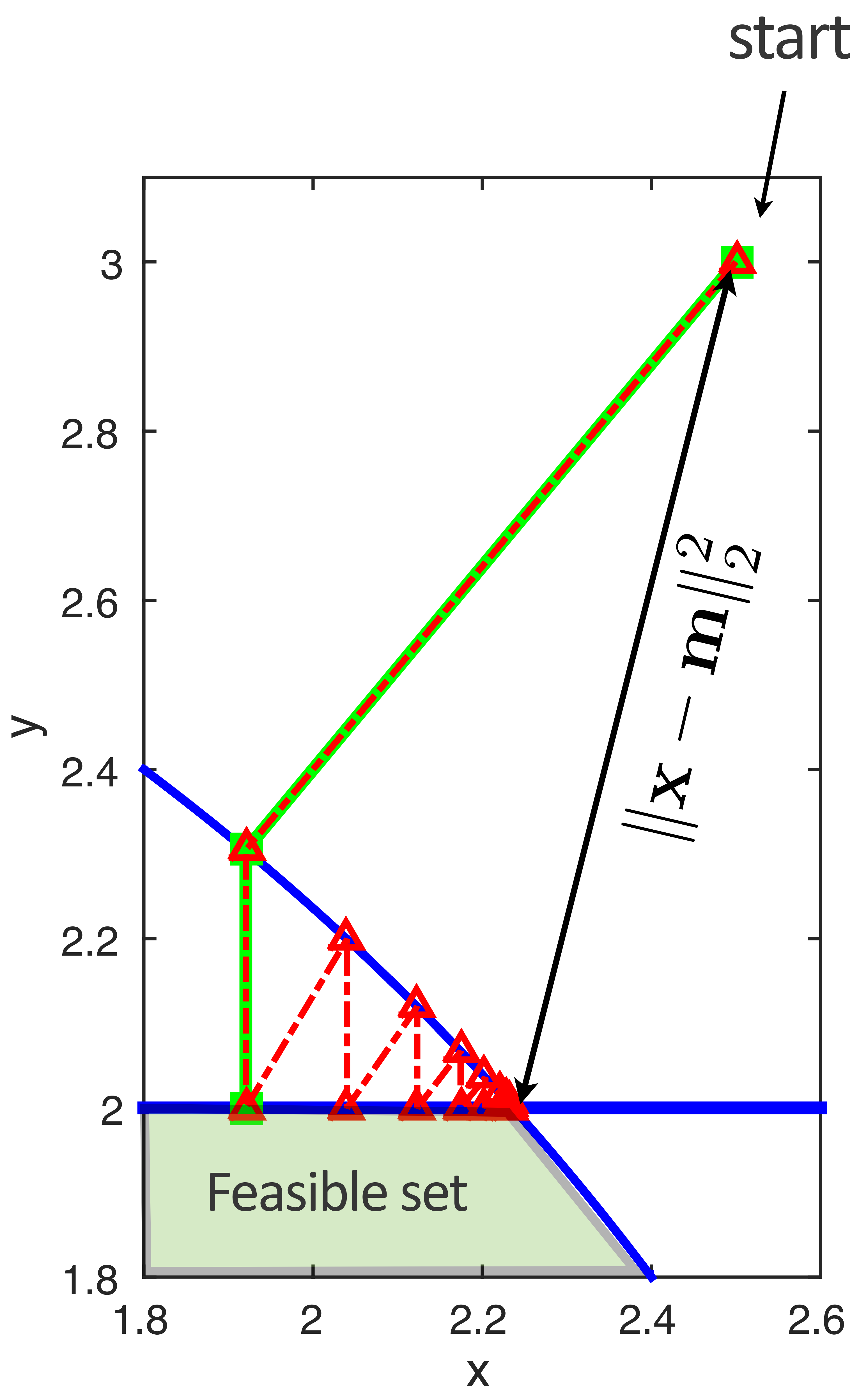
one projection onto each set separately per iteration

input:
  model to project: m
  projectors onto sets $\mathcal{P}_{\mathcal{V}_1}, \mathcal{P}_{\mathcal{V}_2}, \ldots, \mathcal{P}_{\mathcal{V}_p}$ sets
//initialize
0a. $x^0 = m,\; k = 1$
0b. $v_i^0 = x^0$ for $i = 1, 2, \ldots, p$
0c. select weights ρ_i such that $\sum_{i=1}^{p} \rho_i = 1$
  while stopping conditions not satisfied **do**
    FOR $i = 1, 2, \ldots, p$
1. $y_i^{k+1} = \mathcal{P}_{\mathcal{V}_i}(v_i^k)$
    END
2. $x^{k+1} = \sum_{i=1}^{p} \rho_i y_i^{k+1}$
    FOR $i = 1, 2, \ldots, p$
3. $v_i^{k+1} = x^{k+1} + v_i^k - y_i^{k+1}$
    END
4. $k \leftarrow k + 1$
  END
output: x



Intersection of
disk & plane

$$\mathcal{P}_{\mathcal{V}}(m) = \arg\min_{x} \|x - m\|_2 \;\; \text{s.t.} \;\; x \in \bigcap_{i=1}^{p} \mathcal{V}_i.$$



Intersection of
disk & plane

$$\mathcal{P}_{\mathcal{V}}(m) = \arg\min_x \|x - m\|_2 \text{ s.t. } x \in \bigcap_{i=1}^{p} \mathcal{V}_i.$$

Projection onto intersections

Dykstra **Pro**:

- simple and fast if projections are known in closed-form

Projection onto intersections

Dykstra **Pro:**

- simple and fast if projections are known in closed-form

Dykstra **Con**:

- uses another iterative algorithm for other projections
- nested strategy requires two sets of stopping criteria
- does not take similarity between sets into account

Projection onto intersections

Dykstra **Pro**:

- simple and fast if projections are known in closed-form

Dykstra **Con**:

- uses another iterative algorithm for other projections
- nested strategy requires two sets of stopping criteria
- does not take similarity between sets into account

Too many inner iterations for constraints on large 3D grids...

Projection onto intersections

Dykstra **Pro:**

- simple and fast if projections are known in closed-form

Dykstra **Con**:

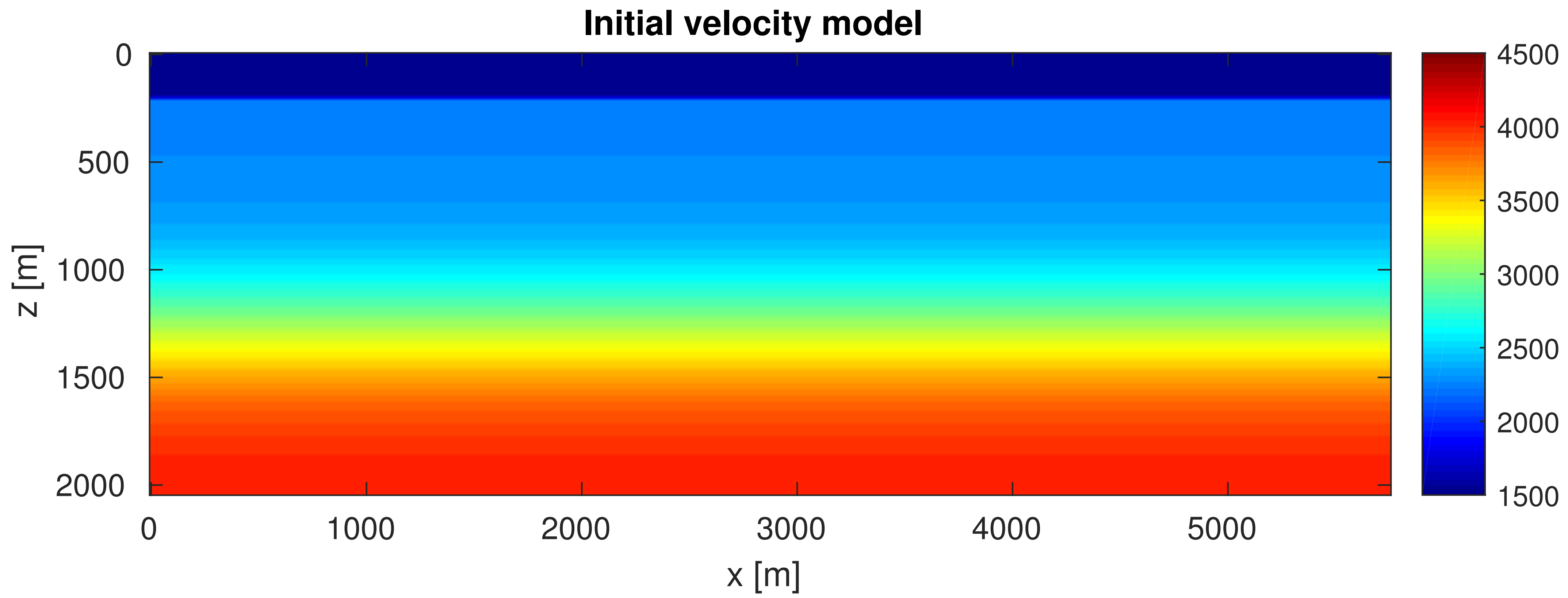
- uses another iterative algorithm for other projections
- nested strategy requires two sets of stopping criteria
- does not take similarity between sets into account

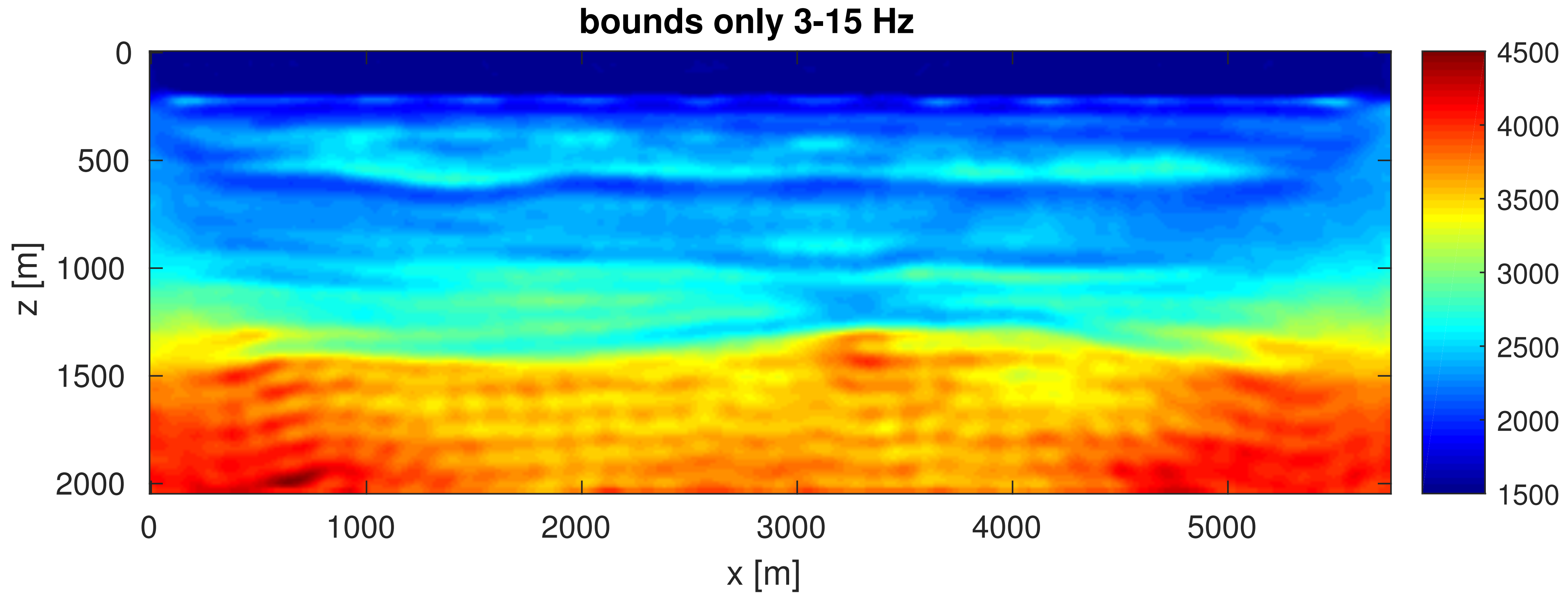
Straightforward applications of operator splitting:

- also leads to black-box algorithm
- suffers from similar problems

Seismic full-waveform inversion

- bound constraints





Heuristics

Problem:

- directly estimating a model is challenging
- what if we do not have much prior knowledge?

Heuristics:

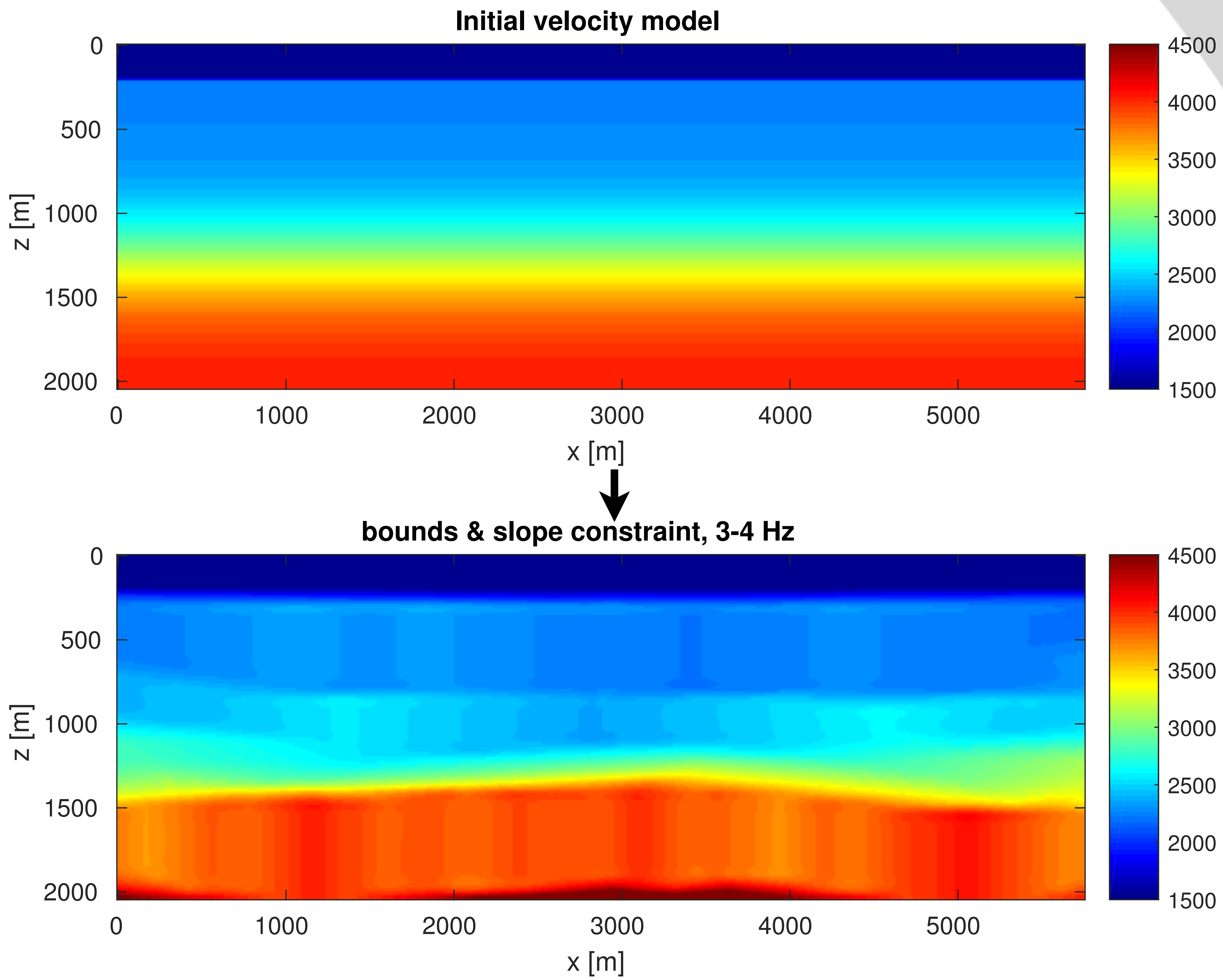
- run w/ 'tight' constraints
- solutions remain feasible
- start over w/ relaxed constraints

Selecting constraints:

- sedimentary geology, mainly layered, no big faults
- starting model should be laterally smooth & velocity increases with depth

1st cycle: invert 3-4 Hz data with:

- bound constraints
- lateral smoothness (slope constraint)
- approximate vertical monotonicity

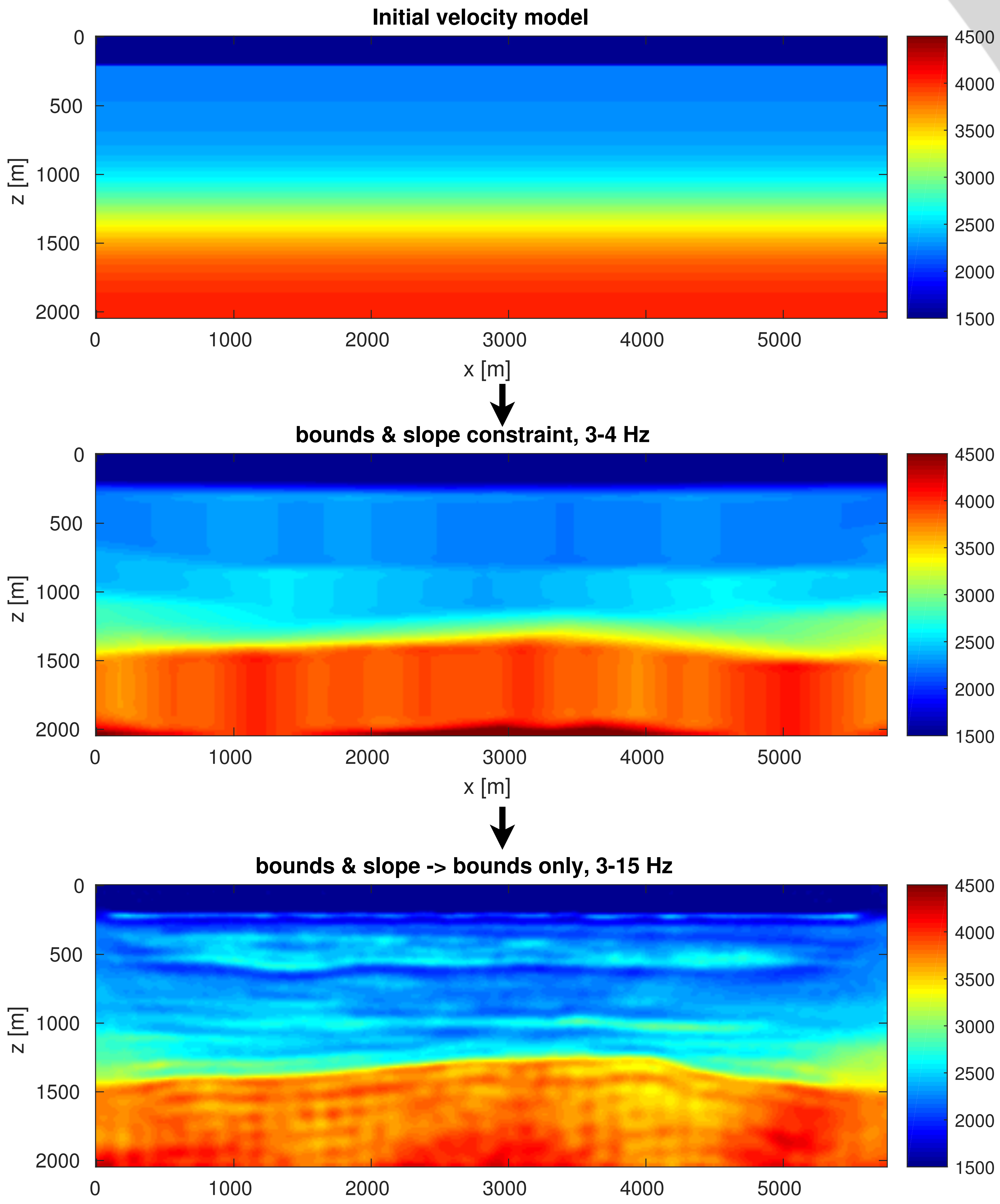


1st cycle: invert 3-4 Hz data with:

- bound constraints
- lateral smoothness (slope constraint)
- approximate vertical monotonicity

2nd cycle:

- use 1st cycle result as new starting model
- invert all data with bound constraints



Observations - intersections of sets

each set defined independently of all other sets

use projection onto intersection as part of, e.g., projected gradient

intersections of constraint sets + constraint adaptation strategies enables full-waveform inversion in more challenging settings

- constraints versus penalties (Chapter 2)
- intersections of multiple constraint sets (Chapter 3)
- **computational aspects (Chapter 4)**
- sums of sets (Chapter 5)

New algorithm

Goals:

Construct an algorithm to project onto an intersection

- allow non-orthogonal linear operators in set definitions
- exploit similarity between sets

New algorithm

[Afonso et. al., 2011],
[Combettes & Pesquet, 2011 ; Kitic et. al. 2016]

Iterations: equivalent to SDMM + over/under relaxation)

$$
\begin{aligned}
x^{k+1} &= \Big[\sum_{i=1}^{p}(\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N\Big]^{-1} \sum_{i=1}^{p+1}\Big[A_i^\top(\rho_i^k y_i^k + v_i^k)\Big] \\
\bar{x}_i^{k+1} &= \gamma_i^k A_i x_i^{k+1} + (1-\gamma_i^k) y_i^k \\
y_i^{k+1} &= \operatorname{prox}_{f_i,\rho_i^k}(\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k}) \\
v_i^{k+1} &= v_i^k + \rho_i^k (y_i^{k+1} - \bar{x}_i^{k+1}).
\end{aligned}
$$

New algorithm

$$x^{k+1} = \Big[\sum_{i=1}^{p}(\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N\Big]^{-1} \sum_{i=1}^{p+1}\Big[A_i^\top(\rho_i^k y_i^k + v_i^k)\Big] \longrightarrow \text{warm-start CG}$$

$$\bar{x}_i^{k+1} = \gamma_i^k A_i x_i^{k+1} + (1-\gamma_i^k) y_i^k$$

$$y_i^{k+1} = \mathrm{prox}_{f_i,\rho_i^k}(\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k})$$

$$v_i^{k+1} = v_i^k + \rho_i^k(y_i^{k+1} - \bar{x}_i^{k+1}).$$

New algorithm

$$
\begin{aligned}
x^{k+1} &= \Big[\sum_{i=1}^{p}(\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N\Big]^{-1} \sum_{i=1}^{p+1}\Big[A_i^\top(\rho_i^k y_i^k + v_i^k)\Big] \\
\bar{x}_i^{k+1} &= \gamma_i^k A_i x_i^{k+1} + (1-\gamma_i^k) y_i^k \\
y_i^{k+1} &= \text{prox}_{f_i,\rho_i^k}(\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k}) \\
v_i^{k+1} &= v_i^k + \rho_i^k(y_i^{k+1} - \bar{x}_i^{k+1}).
\end{aligned}
$$

system-mat always pos-def,

New algorithm

$$
\begin{aligned}
x^{k+1} &= \Big[\sum_{i=1}^{p}(\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N\Big]^{-1} \sum_{i=1}^{p+1}\Big[A_i^\top(\rho_i^k y_i^k + v_i^k)\Big] \\
\bar{x}_i^{k+1} &= \gamma_i^k A_i x_i^{k+1} + (1-\gamma_i^k) y_i^k \\
y_i^{k+1} &= \operatorname{prox}_{f_i,\rho_i^k}(\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k}) \\
v_i^{k+1} &= v_i^k + \rho_i^k (y_i^{k+1} - \bar{x}_i^{k+1}).
\end{aligned}
$$

we can decide on how many CG iterations we need for the sub-problem

New algorithm

discrete derivatives matrices, DFT, Wavelet T, ...

$$x^{k+1} = \Big[\sum_{i=1}^{p} (\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N\Big]^{-1} \sum_{i=1}^{p+1} \Big[A_i^\top (\rho_i^k y_i^k + v_i^k)\Big]$$

$$\bar{x}_i^{k+1} = \gamma_i^k A_i x_i^{k+1} + (1 - \gamma_i^k) y_i^k$$

sum structure

$$y_i^{k+1} = \mathrm{prox}_{f_i, \rho_i^k}(\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k})$$

$$v_i^{k+1} = v_i^k + \rho_i^k (y_i^{k+1} - \bar{x}_i^{k+1}).$$

New algorithm

$$x^{k+1} = \Big[\sum_{i=1}^{p}(\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N\Big]^{-1} \sum_{i=1}^{p+1}\Big[A_i^\top(\rho_i^k y_i^k + v_i^k)\Big]$$

$$\bar{x}_i^{k+1} = \gamma_i^k A_i x_i^{k+1} + (1-\gamma_i^k) y_i^k$$

$$y_i^{k+1} = \mathrm{prox}_{f_i,\rho_i^k}(\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k})$$

$$v_i^{k+1} = v_i^k + \rho_i^k(y_i^{k+1} - \bar{x}_i^{k+1}).$$

in parallel for every index i

New algorithm

$$x^{k+1} = \Big[\sum_{i=1}^{p}(\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N\Big]^{-1} \sum_{i=1}^{p+1}\Big[A_i^\top(\rho_i^k y_i^k + v_i^k)\Big]$$

$$\bar{x}_i^{k+1} = \gamma_i^k A_i x_i^{k+1} + (1-\gamma_i^k) y_i^k \longrightarrow \text{over/under relaxation}$$

$$y_i^{k+1} = \operatorname{prox}_{f_i,\rho_i^k}(\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k})$$

$$v_i^{k+1} = v_i^k + \rho_i^k(y_i^{k+1} - \bar{x}_i^{k+1}).$$

New algorithm

$$
\begin{aligned}
x^{k+1} &= \Big[\sum_{i=1}^{p}(\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N\Big]^{-1} \sum_{i=1}^{p+1}\Big[A_i^\top(\rho_i^k y_i^k + v_i^k)\Big] \\
\bar{x}_i^{k+1} &= \gamma_i^k A_i x_i^{k+1} + (1-\gamma_i^k) y_i^k \\
y_i^{k+1} &= \mathrm{prox}_{f_i,\rho_i^k}(\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k}) \\
v_i^{k+1} &= v_i^k + \rho_i^k (y_i^{k+1} - \bar{x}_i^{k+1}).
\end{aligned}
$$

→ simple (closed-form) projections: norm-ball/bounds/cardinality/rank

New algorithm

additional sets do not increase time per iteration (much)

$$
\begin{aligned}
x^{k+1} &= \Big[\sum_{i=1}^{p}(\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N\Big]^{-1} \sum_{i=1}^{p+1}\Big[A_i^\top(\rho_i^k y_i^k + v_i^k)\Big] \\
\bar{x}_i^{k+1} &= \gamma_i^k A_i x_i^{k+1} + (1-\gamma_i^k) y_i^k \\
y_i^{k+1} &= \operatorname{prox}_{f_i,\rho_i^k}(\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k}) \\
v_i^{k+1} &= v_i^k + \rho_i^k(y_i^{k+1} - \bar{x}_i^{k+1}).
\end{aligned}
$$

Iterations are just iterations…

For fast algorithms we also need:

- stopping conditions
- adaptive parameter selection [Xu et. al. ,2016a ; Xu et. al. ,2017]
- hybrid coarse-fine parallel implementation
- multilevel acceleration
- use multithreaded compressed diagonal MVPs for banded matrices

- couple more things…

Outline

- constraints versus penalties (Chapter 2)
- intersections of multiple constraint sets (Chapter 3)
- computational aspects (Chapter 4)
- **sums of sets (Chapter 5)**

Example

original image

=

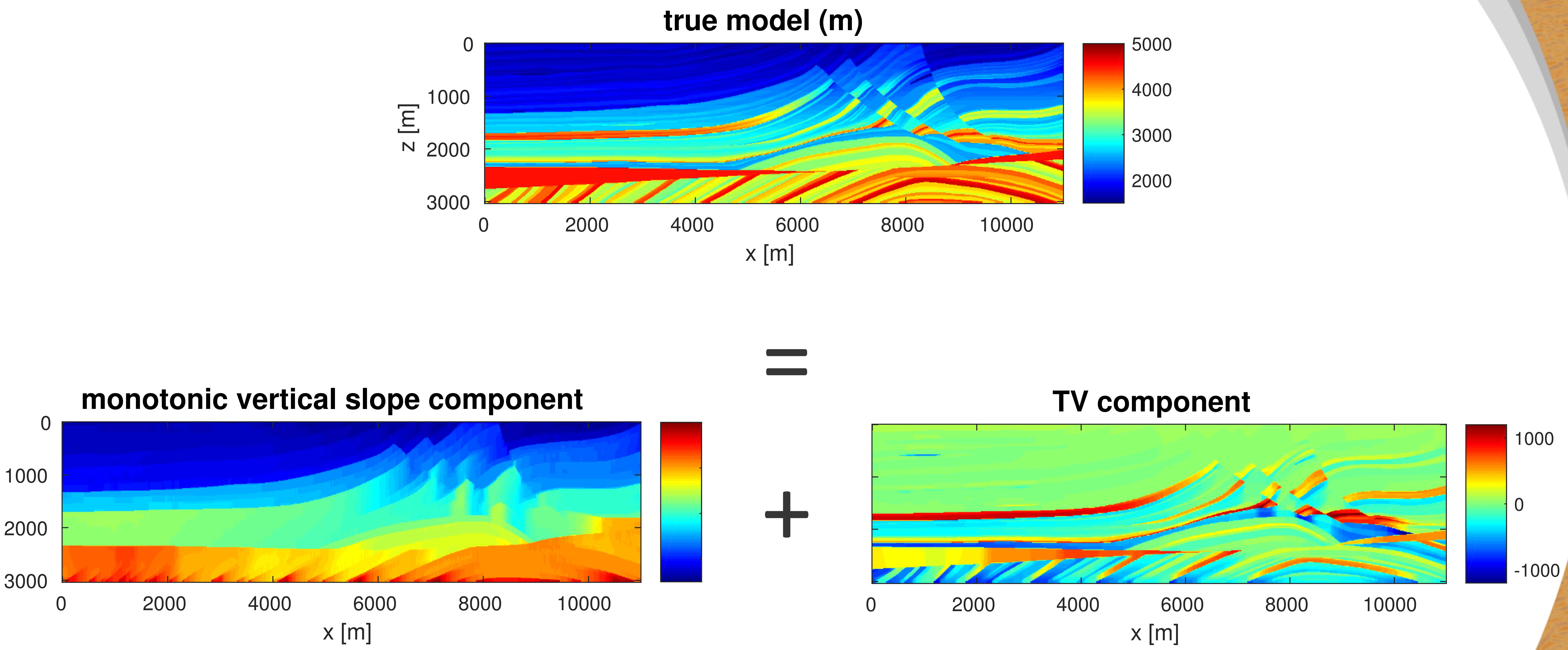
cartoon part

+

texture part

[Szolgay D, & Szirányi T., 2012]

Minkowski decomposition



Inspiration

cartoon-texture decomposition / morphological component analysis

[Osher et al. (2003); Starck et al. (2005); Schaeffer and Osher (2013); Ono et al. (2014)]

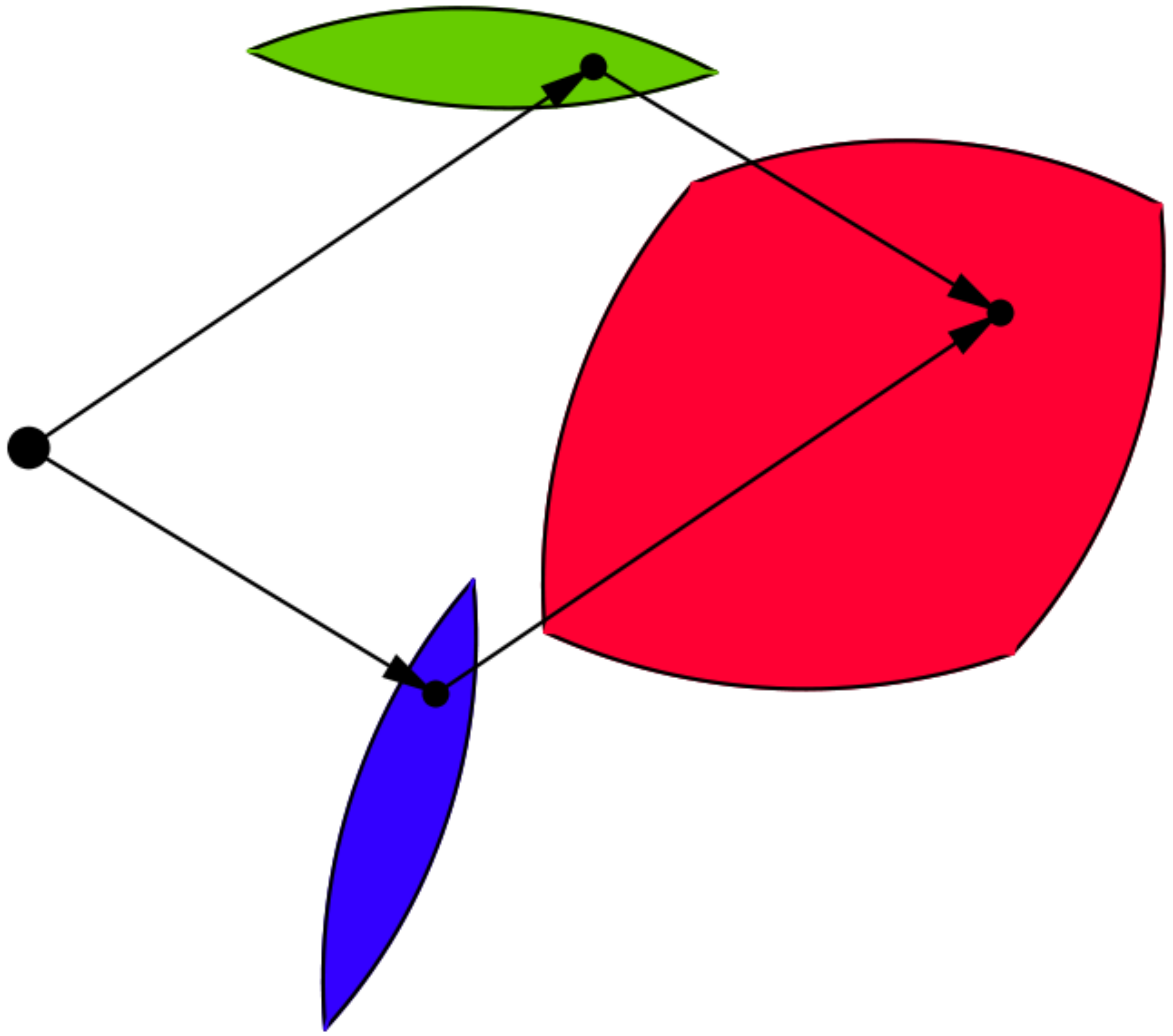
often stated as: $\min_{u,v} \|m - u - v\| + \frac{\alpha}{2}\|Au\| + \frac{\beta}{2}\|Bv\|.$

approximately decompose m into:

1. u : cartoon/background/piecewise-smooth or constant component
2. v : texture/details/pattern/oscillatory component

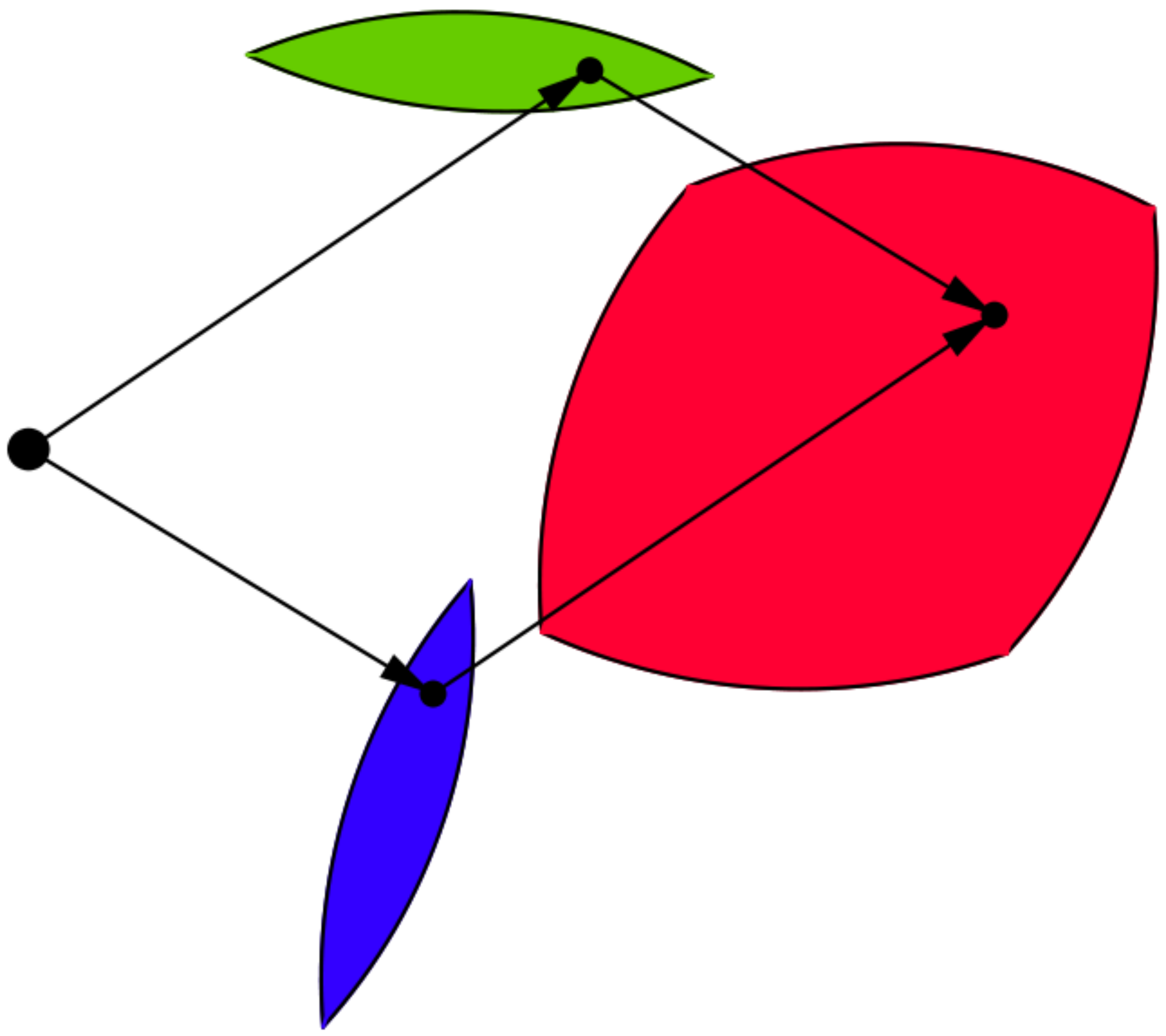
closely related to robust PCA and variants [Candes et al. (2011); Gao et al., 2011a ; Gao et al., 2011b]

Minkowski sum



https://en.wikipedia.org/wiki/Minkowski_addition

Minkowski sum



construct ‘complicated’ sets from ‘simple’ sets

Minkowski sum constraint sets

$$\mathcal{V} \equiv \mathcal{C}_1 + \mathcal{C}_1 = \{m = u + v \mid u \in \mathcal{C}_1, v \in \mathcal{C}_2\}$$

Minkowski set not suitable by itself:

- need bound constraints and more on m
- would like more than one constraint on u and v

Definition 1: Generalized Minkowski set

$$\mathcal{M} \equiv \{m = u + v \,|\, u \in \bigcap_{i=1}^{p} \mathcal{D}_i,\; v \in \bigcap_{j=1}^{q} \mathcal{E}_j,\; m \in \bigcap_{k=1}^{r} \mathcal{F}_k\}$$

Definition 1: Generalized Minkowski set

$$\mathcal{M} \equiv \{m = u + v \,|\, u \in \bigcap_{i=1}^{p} \mathcal{D}_i,\; v \in \bigcap_{j=1}^{q} \mathcal{E}_j,\; m \in \bigcap_{k=1}^{r} \mathcal{F}_k\}$$

Proposition 1. *The generalized Minkowski set is convex if* $\mathcal{D}_i$, $\mathcal{E}_j$, *and* $\mathcal{F}_k$ *are convex sets for all* i, j *and* k.

Proof. It follows from the definition (almost)

Projections onto the generalized Minkowski set

$$\mathcal{M} \equiv \{m = u + v \mid u \in \bigcap_{i=1}^{p} \mathcal{D}_i, \; v \in \bigcap_{j=1}^{q} \mathcal{E}_j, \; m \in \bigcap_{k=1}^{r} \mathcal{F}_k\}$$

Projection:

$$\underset{u,v,w}{\arg\min} \; \frac{1}{2}\|w-m\|_2^2 + \sum_{i=1}^{p} \iota_{\mathcal{D}_i}(A_i u) + \sum_{j=1}^{q} \iota_{\mathcal{E}_j}(B_j v) + \sum_{k=1}^{r} \iota_{\mathcal{F}_k}(C_k w) + \iota_{w=u+v}(w, u, v)$$

- follow same recipe as for intersections
- matrices -> block-structured linear systems
- same algorithm in the end, different inputs

Conclusions (1)

Intersections of sets are particularly suitable in case of many constraint sets.

Introduced the generalized Minkowski set.

Intersections and sums of sets allow for the inclusion of more prior info.

Solve inverse problems directly as a projection, or
include projection as part of projected gradient-type algorithm.

Developed new constraint relaxation strategies for seismic inverse problems.

Conclusions (2)

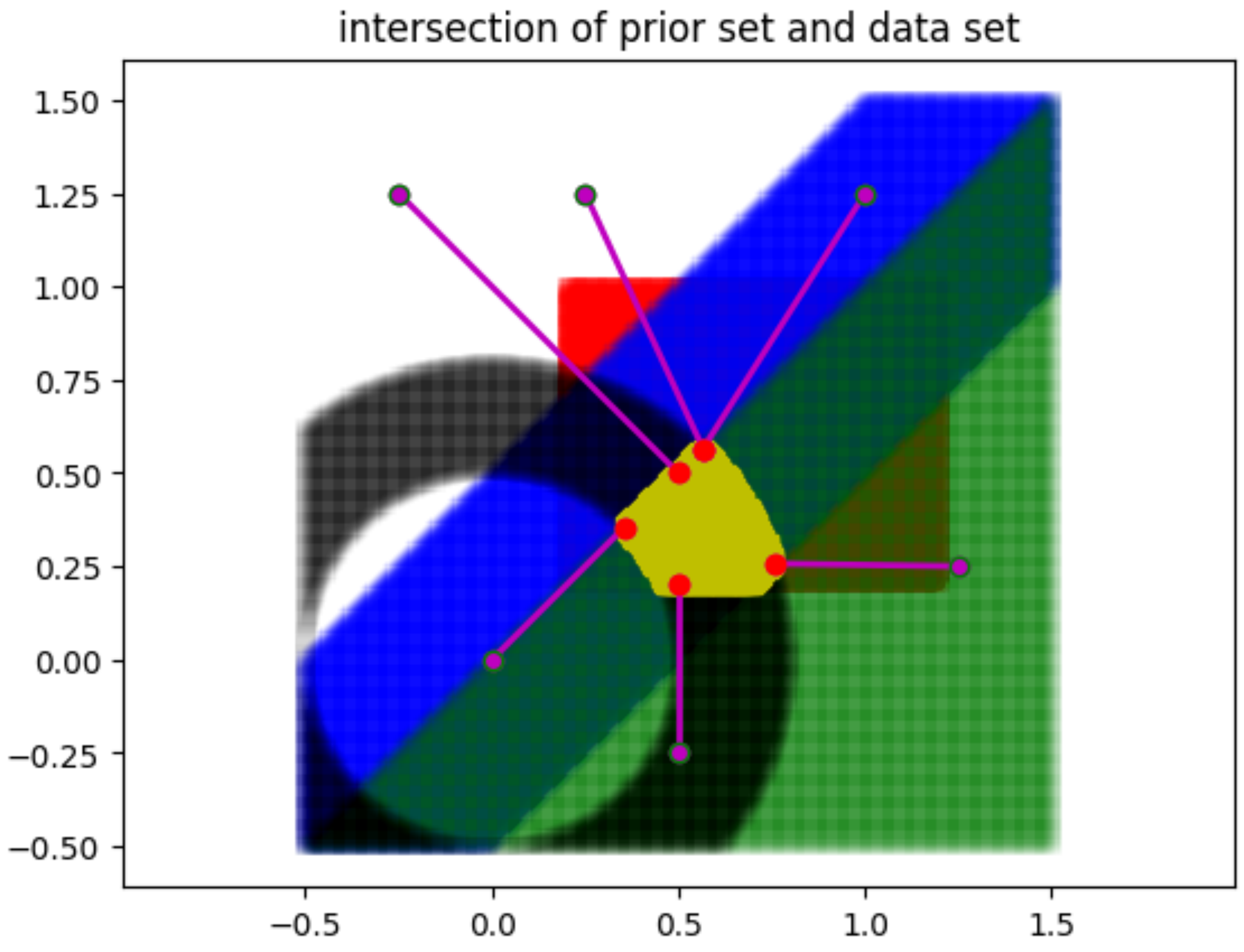
Developed a new algorithm for projecting onto an intersection, as well as the generalized Minkowski set.

High performance implementation in Julia:

- hybrid worker-threading parallelism
- multilevel
- tested on inverse problems with up to 12 constraint sets
- tested on geophysical and video problems of up to 400^3 cube

Future research directions

set-theoretic uncertainty quantification



combine neural networks and projections onto intersections of sets

Thank you!