Solving the data-augmented WE

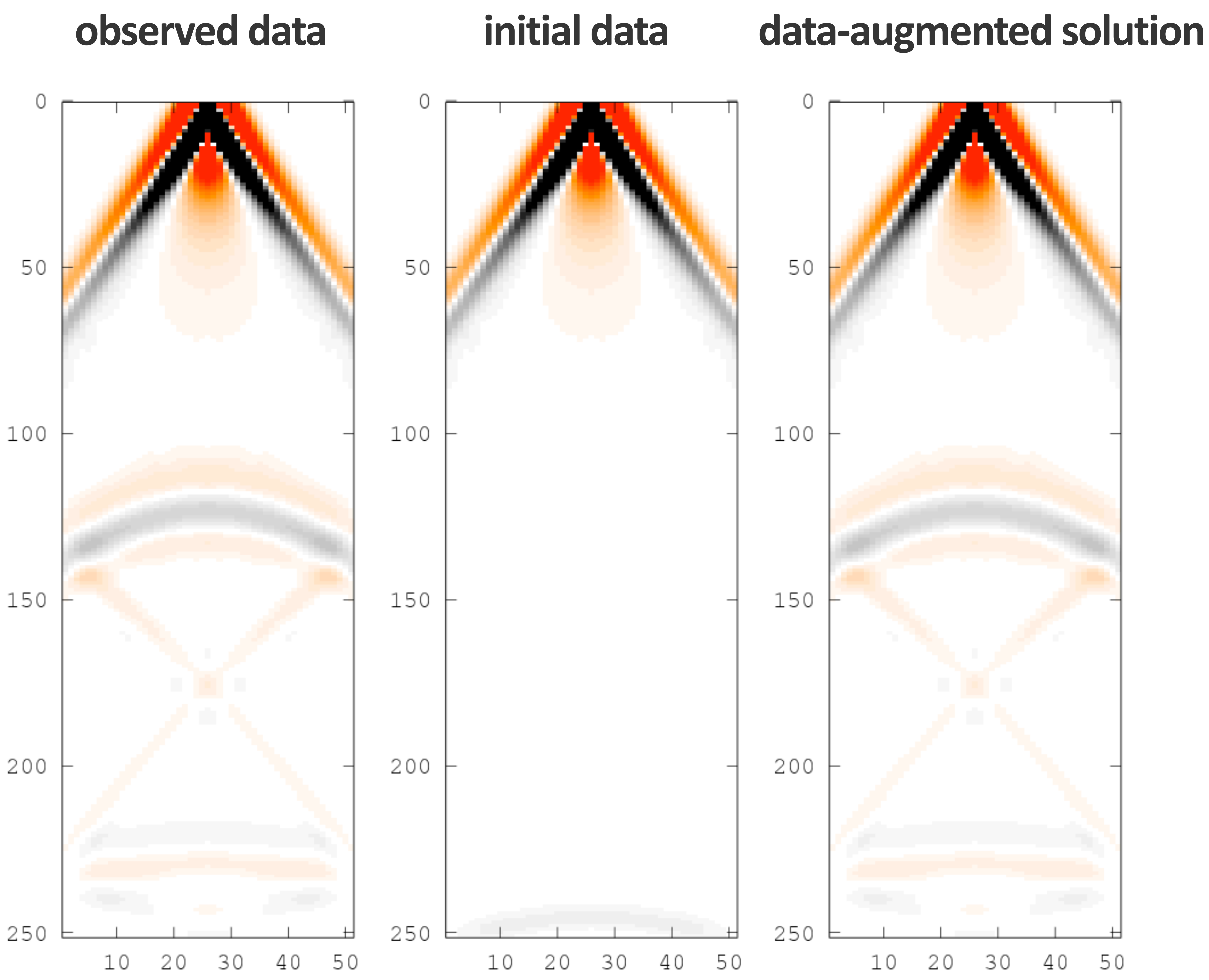
Tristan van Leeuwen
Felix Herrmann
Bas Peters

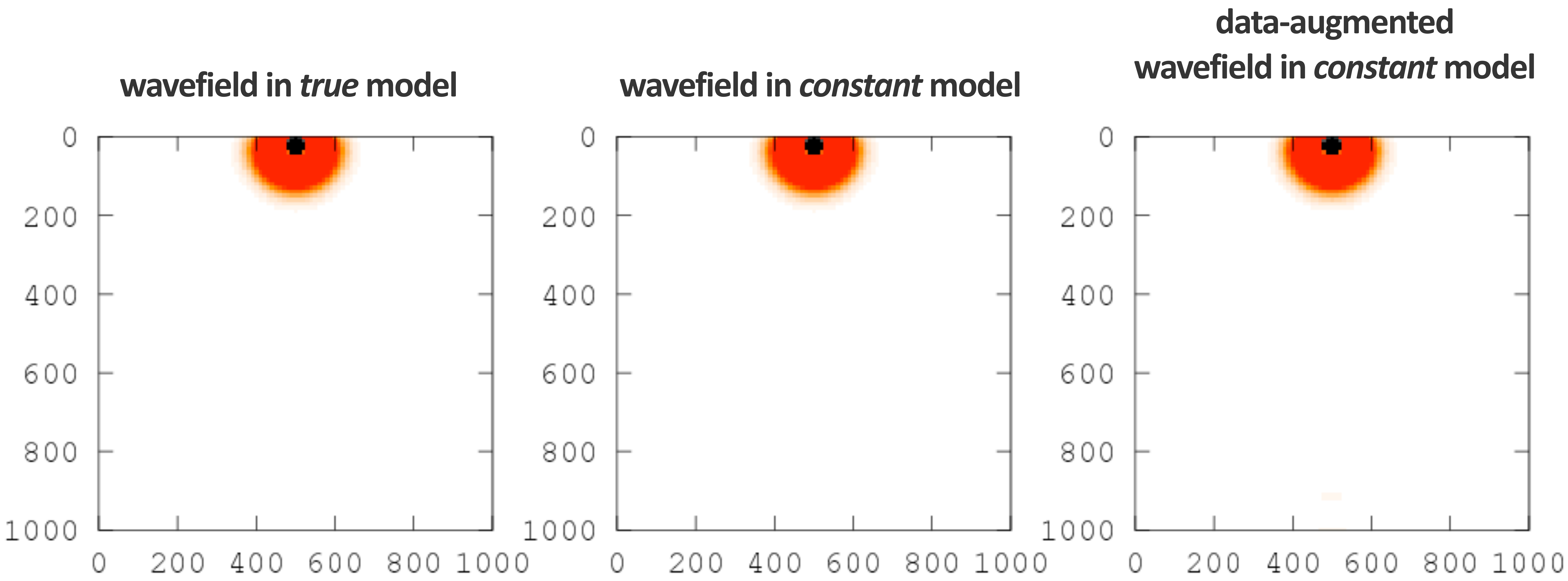
SLIM
University of British Columbia

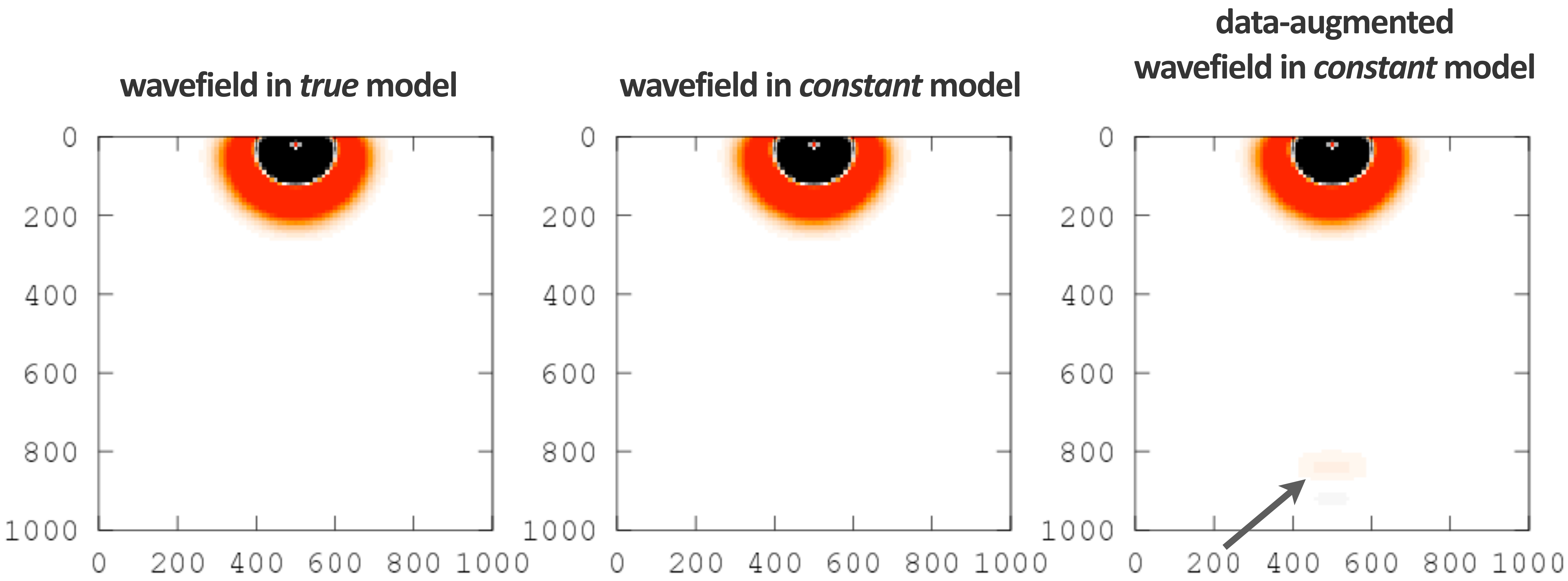
wave-equation X wavefield = source

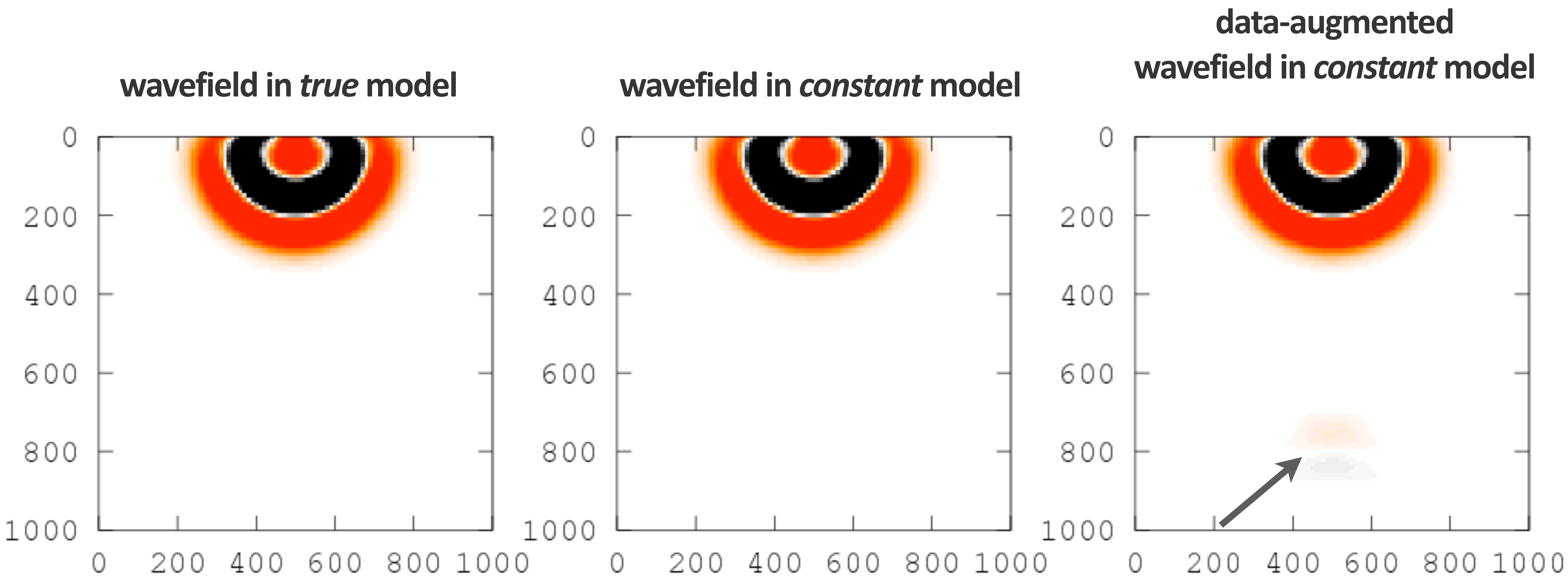
versus

(wave-equation / sampling operator) X wavefield = (source / data)


observed data
initial data
data-augmented solution
0
50
100
150
200
250
10
20
30
40
50


wavefield in *true* model
wavefield in *constant* model
data-augmented
wavefield in *constant* model
0
200
400
600
800
1000
0 200 400 600 800 1000


wavefield in *true* model
wavefield in *constant* model
data-augmented
wavefield in *constant* model
0
200
400
600
800
1000
0 200 400 600 800 1000

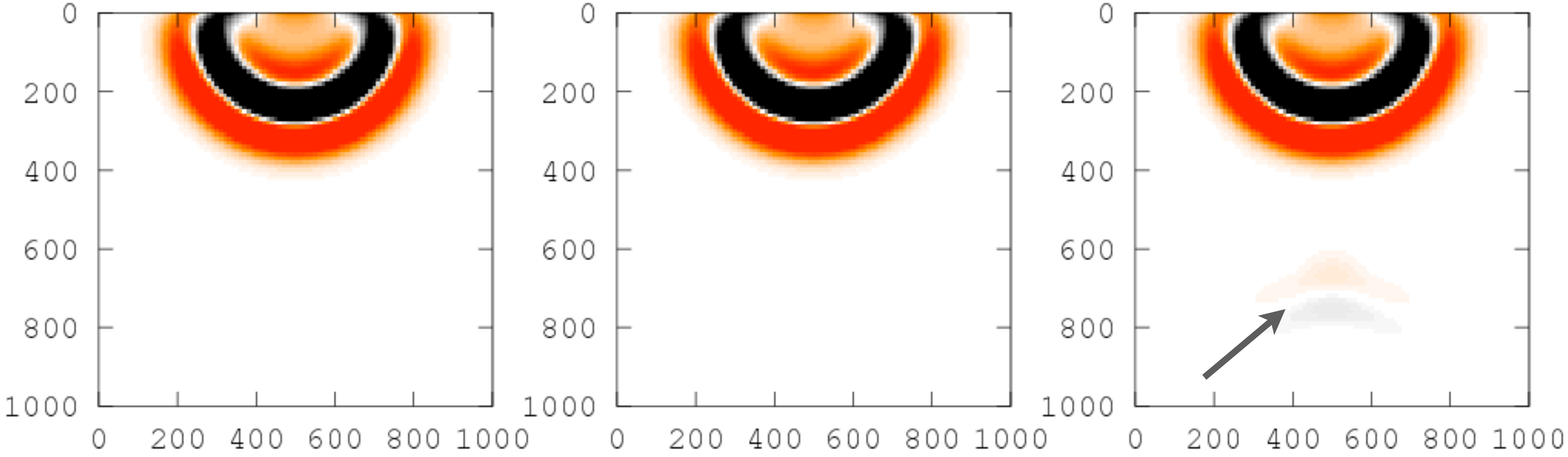


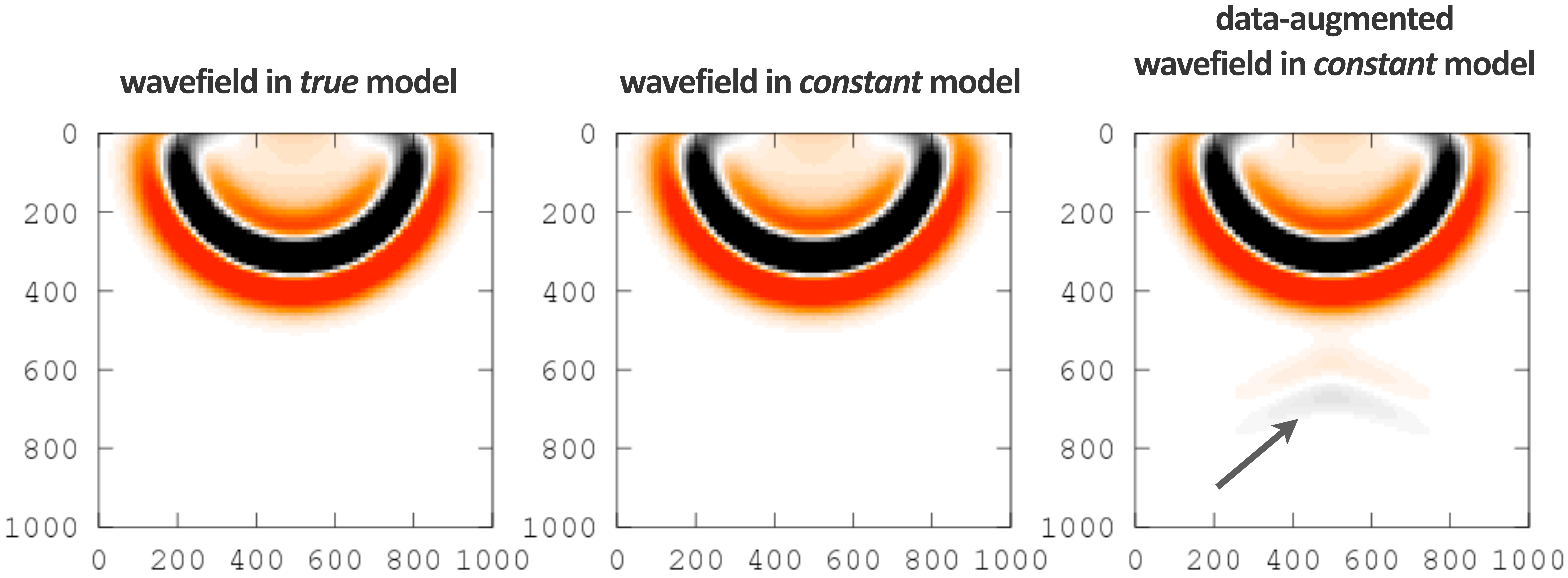
wavefield in *true* model
wavefield in *constant* model
data-augmented
wavefield in *constant* model
0
200
400
600
800
1000
0 200 400 600 800 1000

wavefield in *true* model

wavefield in *constant* model

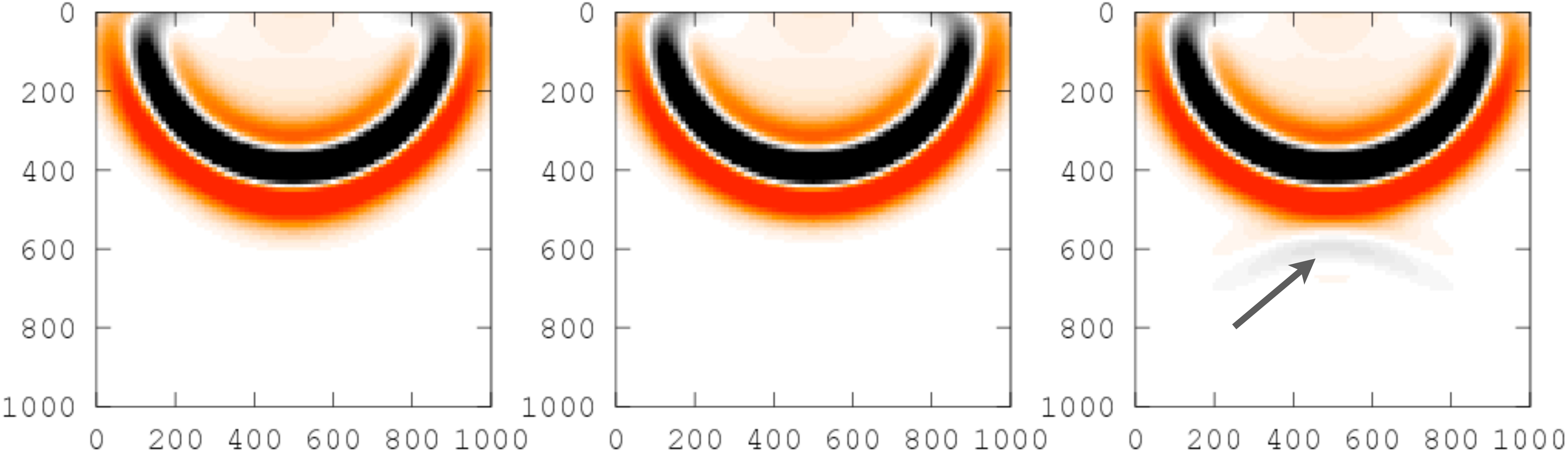
data-augmented wavefield in *constant* model

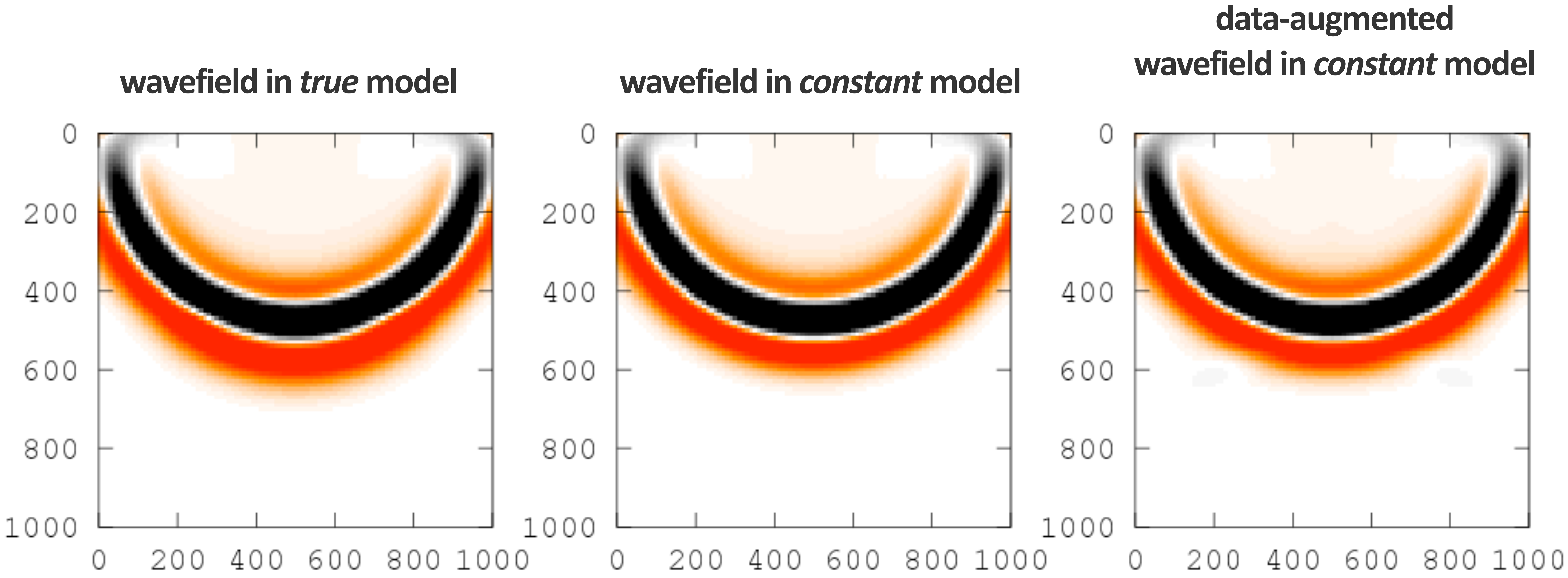


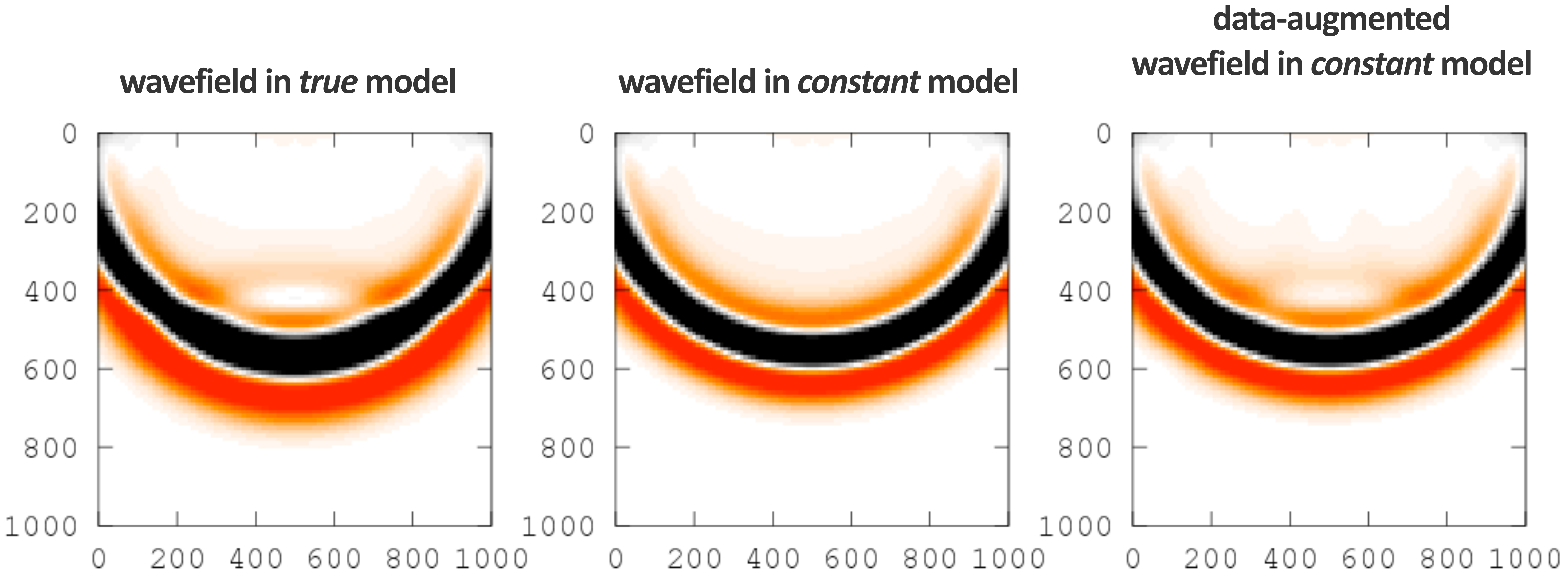


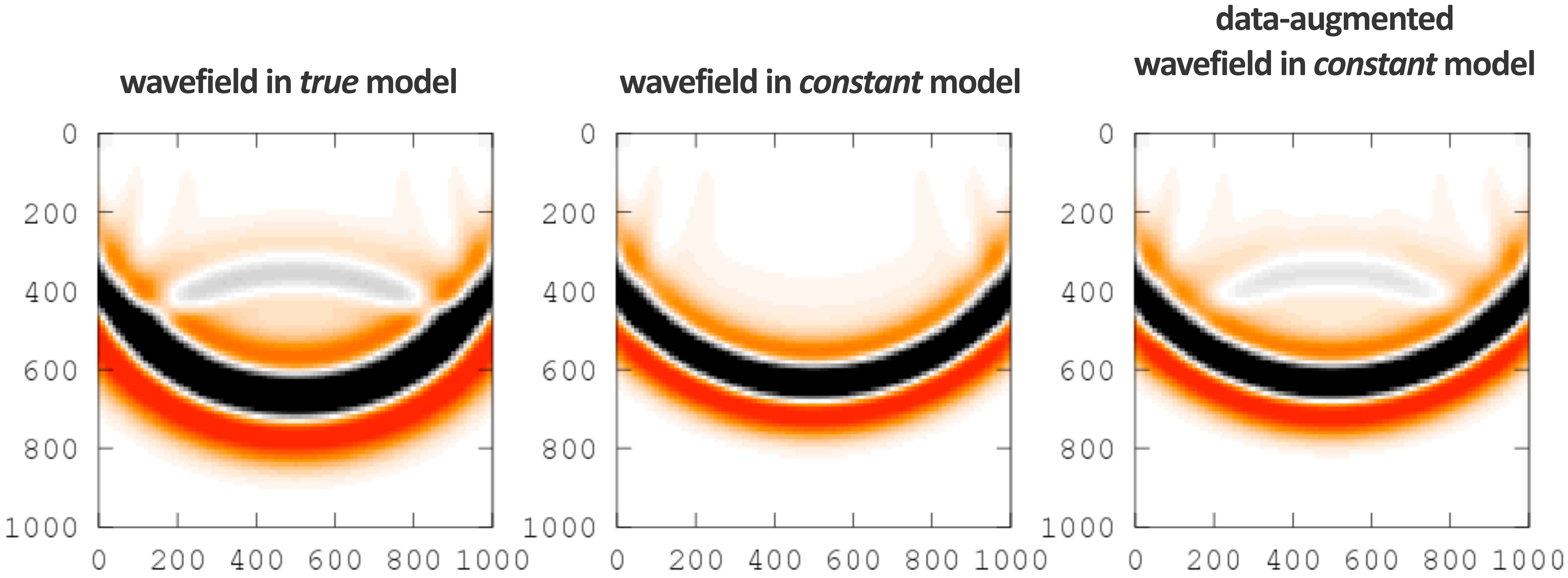
wavefield in *true* model
wavefield in *constant* model
data-augmented
wavefield in *constant* model
0
200
400
600
800
1000
0 200 400 600 800 1000

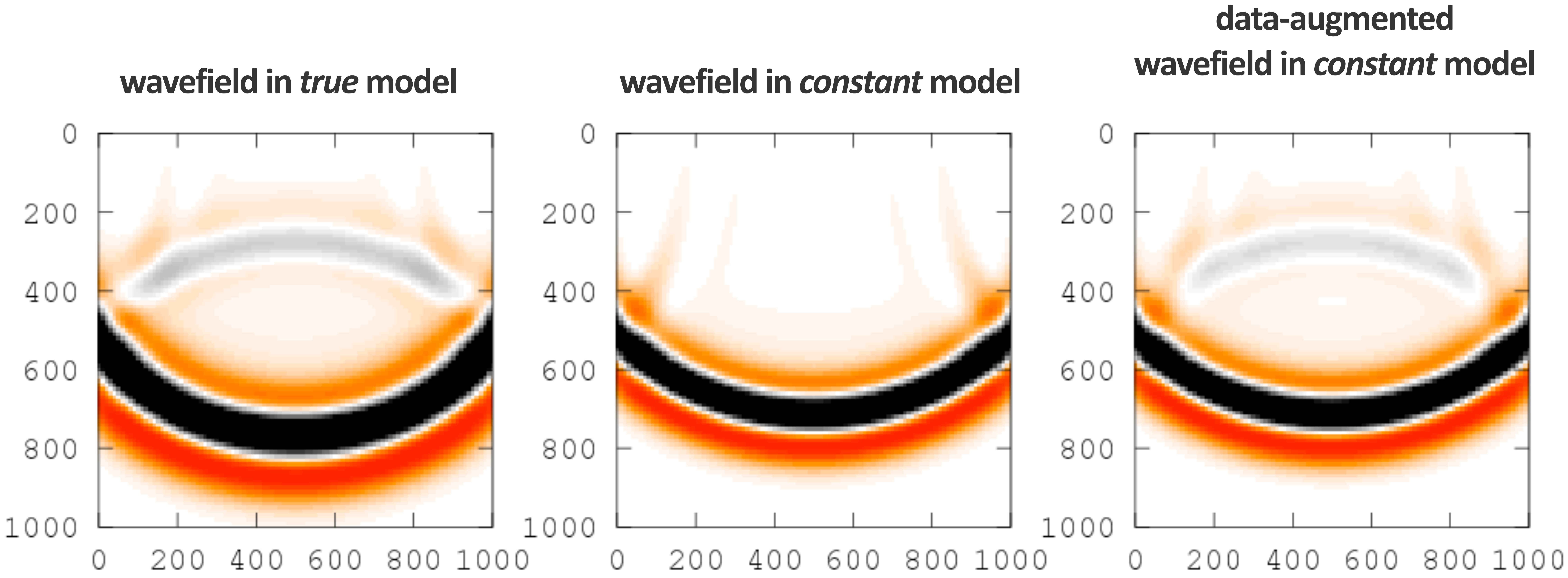
wavefield in *true* model | **wavefield in *constant* model** | **data-augmented wavefield in *constant* model**

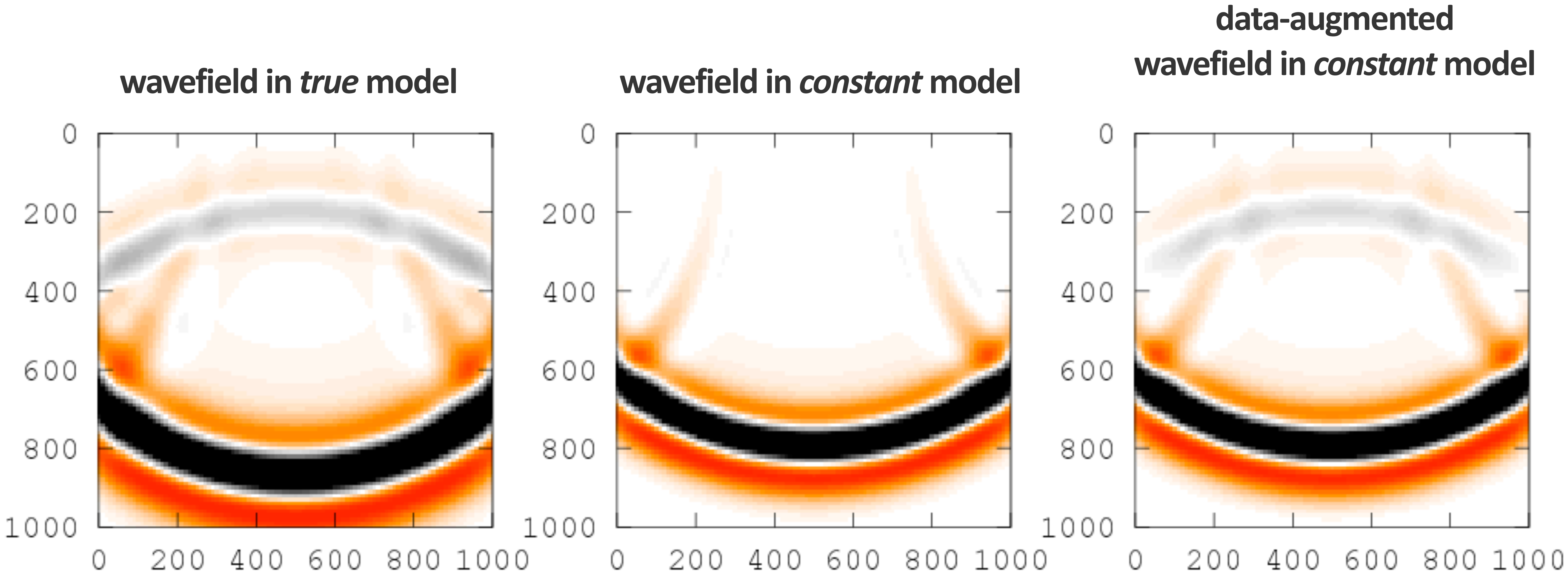



wavefield in *true* model
wavefield in *constant* model
data-augmented
wavefield in *constant* model
0
200
400
600
800
1000
0
200
400
600
800
1000

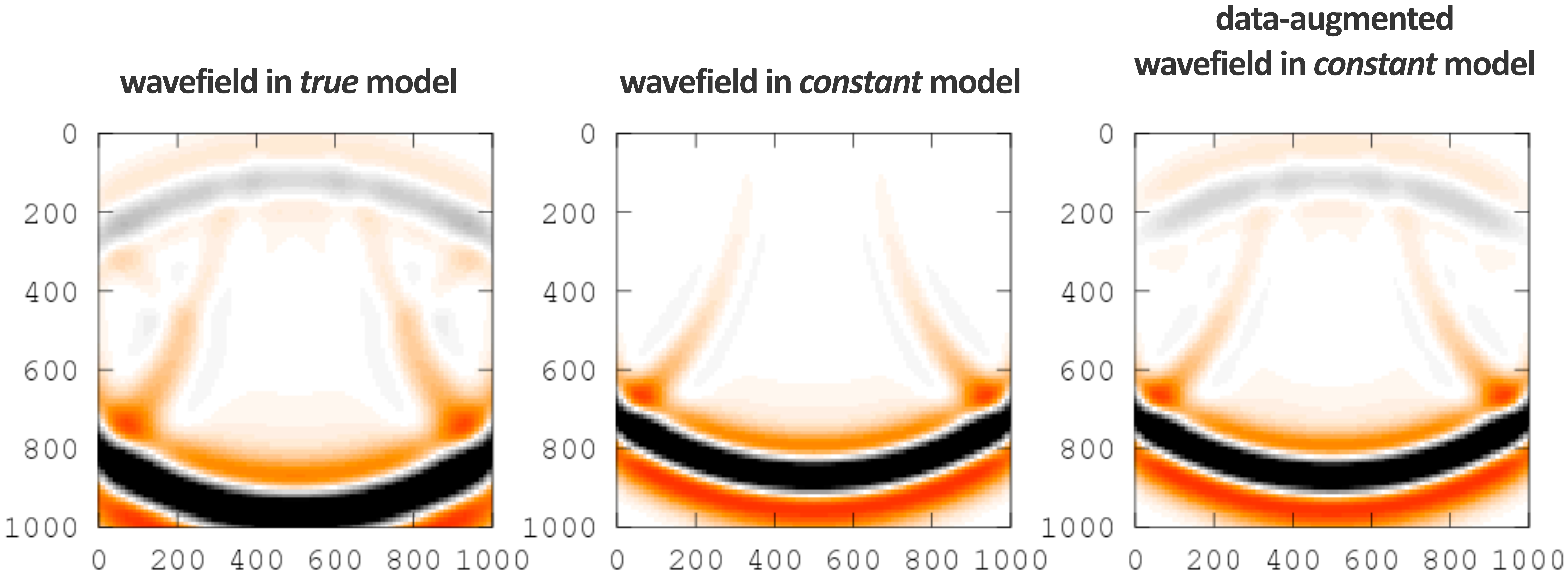

wavefield in *true* model
wavefield in *constant* model
data-augmented
wavefield in *constant* model
0
200
400
600
800
1000
0 200 400 600 800 1000


wavefield in *true* model
wavefield in *constant* model
data-augmented
wavefield in *constant* model
0
200
400
600
800
1000
0
200
400
600
800
1000

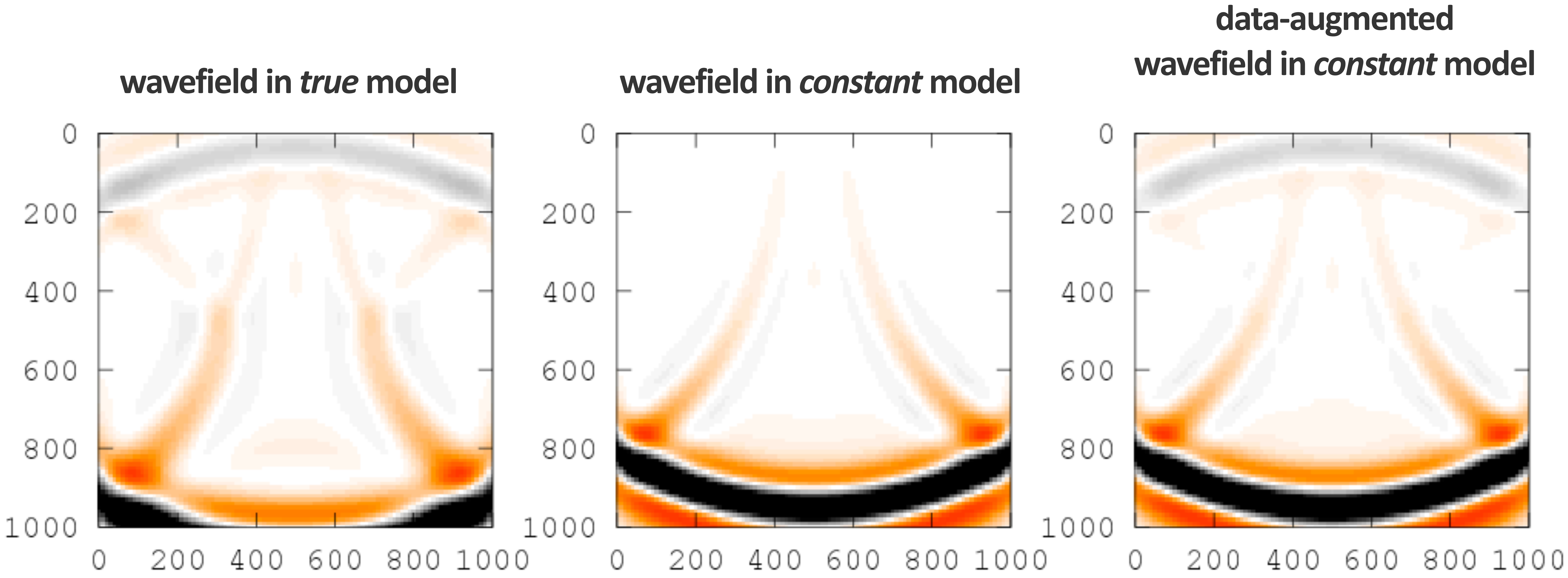

wavefield in true model
wavefield in constant model
data-augmented
wavefield in constant model
0
200
400
600
800
1000
0
200
400
600
800
1000



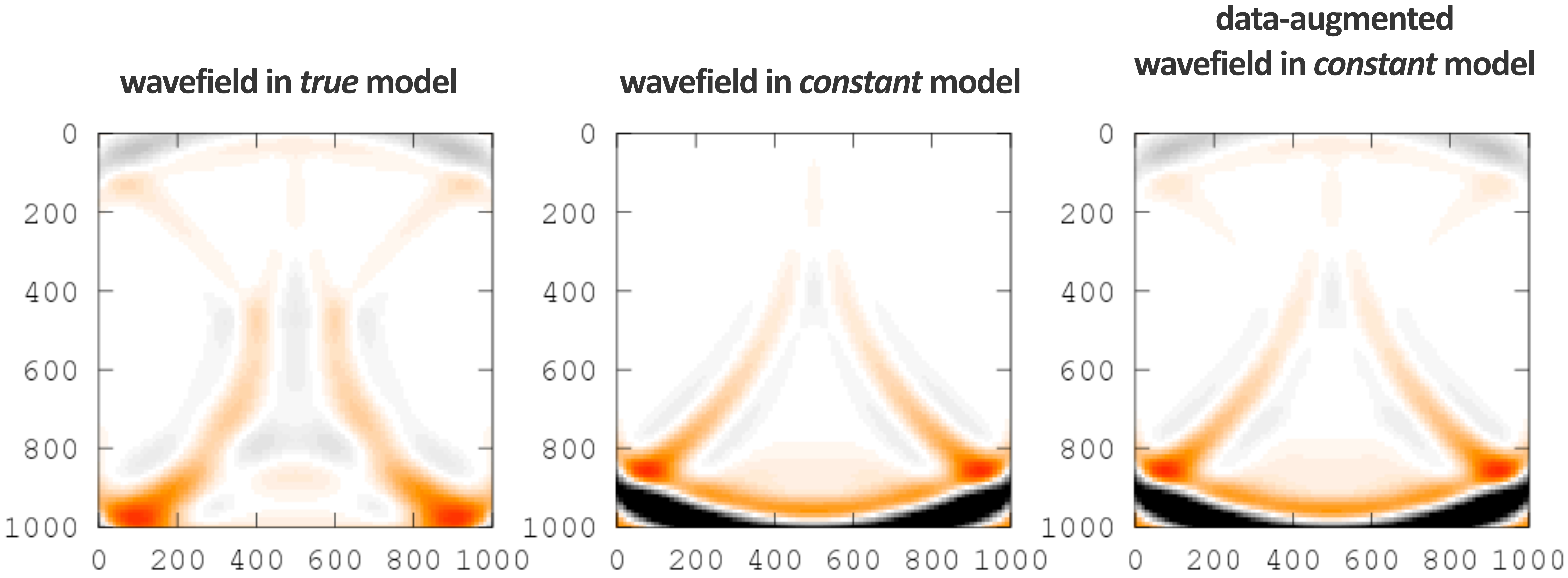
wavefield in *true* model
wavefield in *constant* model
data-augmented
wavefield in *constant* model
0
200
400
600
800
1000



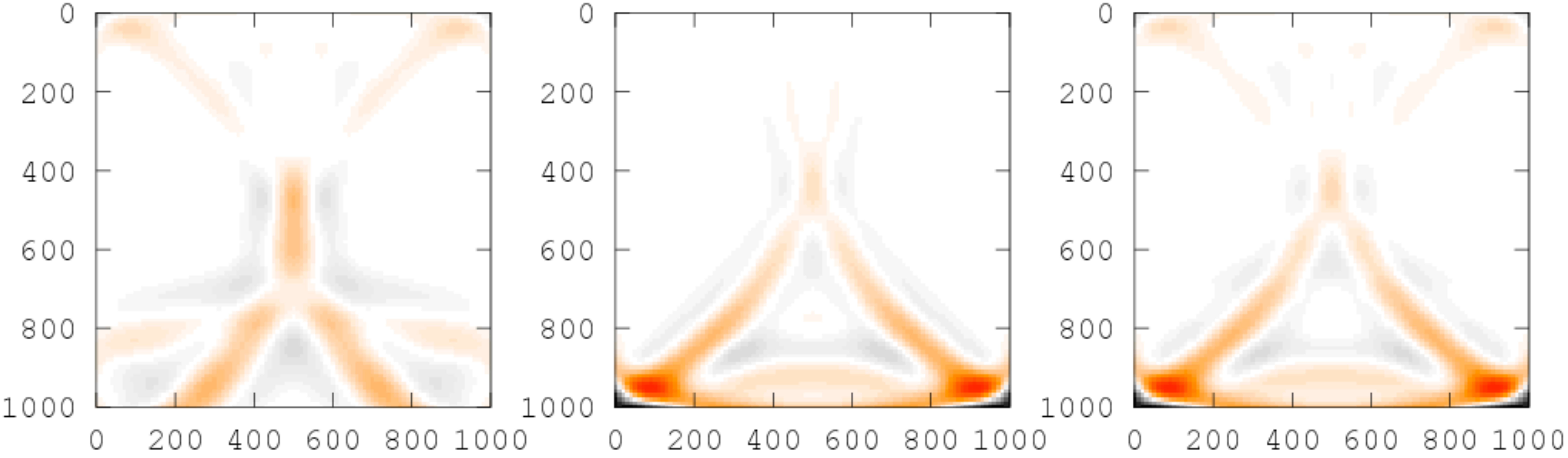
wavefield in *true* model
wavefield in *constant* model
data-augmented
wavefield in *constant* model
0
200
400
600
800
1000
0 200 400 600 800 1000

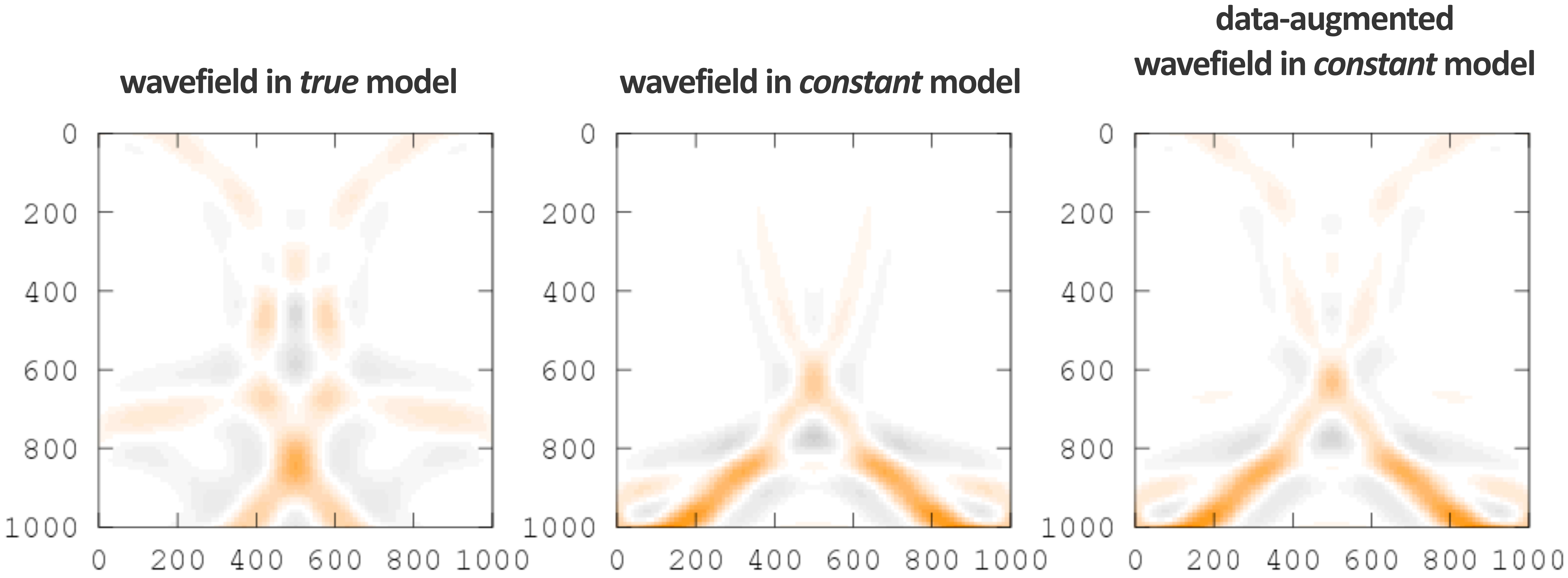


wavefield in *true* model
wavefield in *constant* model
data-augmented
wavefield in *constant* model
0
200
400
600
800
1000


wavefield in *true* model
wavefield in *constant* model
data-augmented
wavefield in *constant* model
0
200
400
600
800
1000
0 200 400 600 800 1000

wavefield in *true* model | **wavefield in *constant* model** | **data-augmented wavefield in *constant* model**




wavefield in *true* model
wavefield in *constant* model
data-augmented
wavefield in *constant* model
0
200
400
600
800
1000

Outline

Frequency-domain

- direct factorization
- iterative methods

Time-domain

- time-stepping for overdetermined ODEs

Frequency-domain

(mildly) Overdetermined system:

Helmholtz ($n \times n$) → A

Sampling ($nr \times n$) → P

$$\begin{pmatrix} A \\ P \end{pmatrix} \mathbf{u} \approx \begin{pmatrix} \mathbf{q} \\ \mathbf{d} \end{pmatrix}$$

$\mathbf{q}$ ← source

$\mathbf{d}$ ← data

Normal equations

$$(A^*A + P^*P)\,\mathbf{u} = \mathbf{A}^*\mathbf{q} + P^*\mathbf{d}$$

A^*A: full-rank

P^*P: rank nr

Direct factorization

QR factorization:

$$\begin{pmatrix} A \\ P \end{pmatrix} = QR$$

orthogonal (Q) lower-triangular (R)

Choleski:

$$(A^*A + P^*P) = LL^*$$

lower-triangular (L)

OK for 2D, prohibitive for 3D?!

Iterative methods

Solve overdetermined system

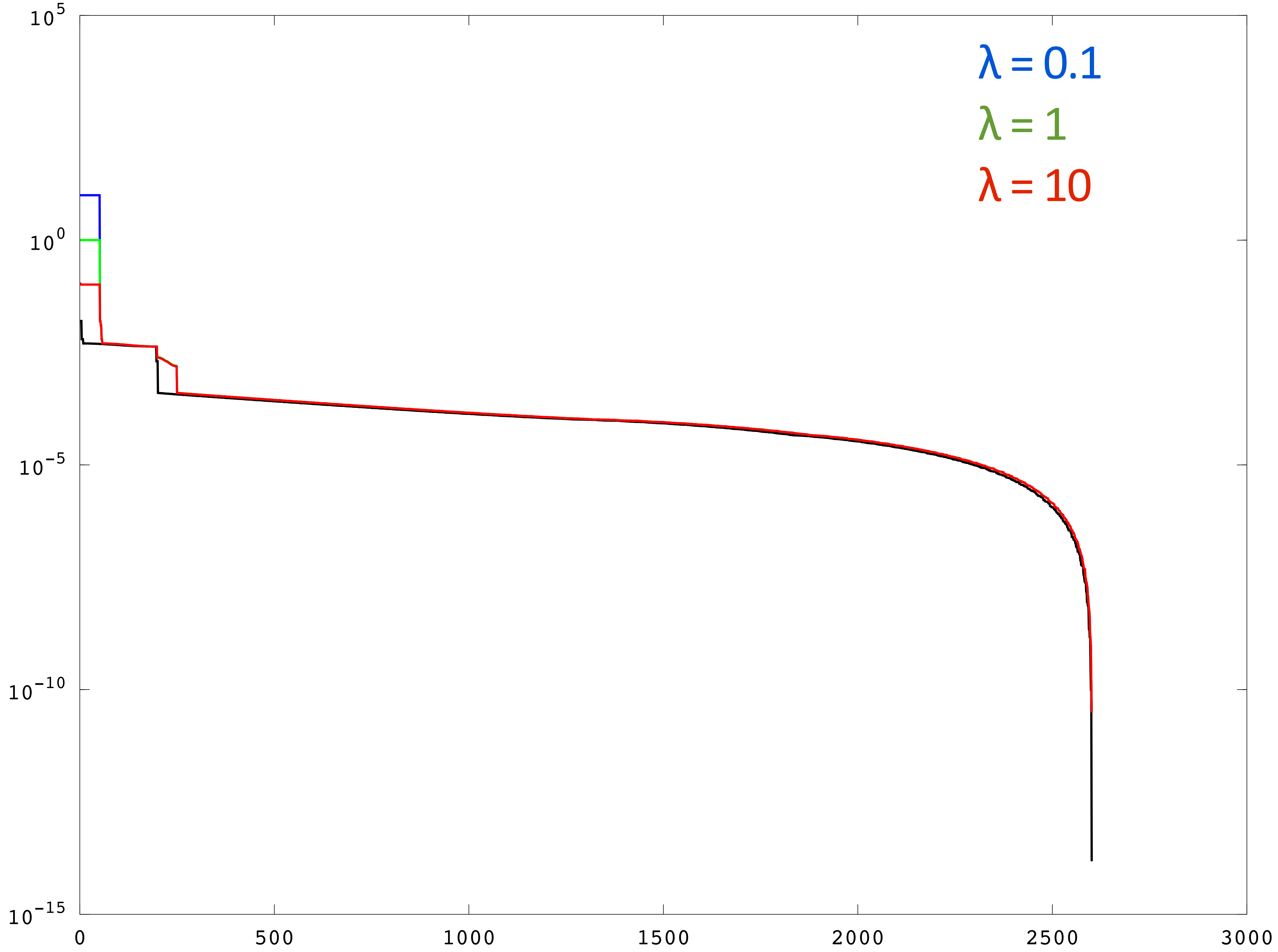
$$\begin{pmatrix} A \\ P \end{pmatrix} \mathbf{u} \approx \begin{pmatrix} \mathbf{q} \\ \mathbf{d} \end{pmatrix}$$ or

$$\begin{pmatrix} A \\ P \end{pmatrix} \delta\mathbf{u} \approx \begin{pmatrix} 0 \\ \mathbf{d} - P\mathbf{u}_0 \end{pmatrix}$$ where $$\begin{aligned} \mathbf{u}_0 &= A^{-1}\mathbf{q} \\ \mathbf{u} &= \mathbf{u}_0 + \delta\mathbf{u} \end{aligned}$$

- LSQR/LSMR
- CGLS?!

Conditioning

$$(\lambda A^*A + P^*P)$$



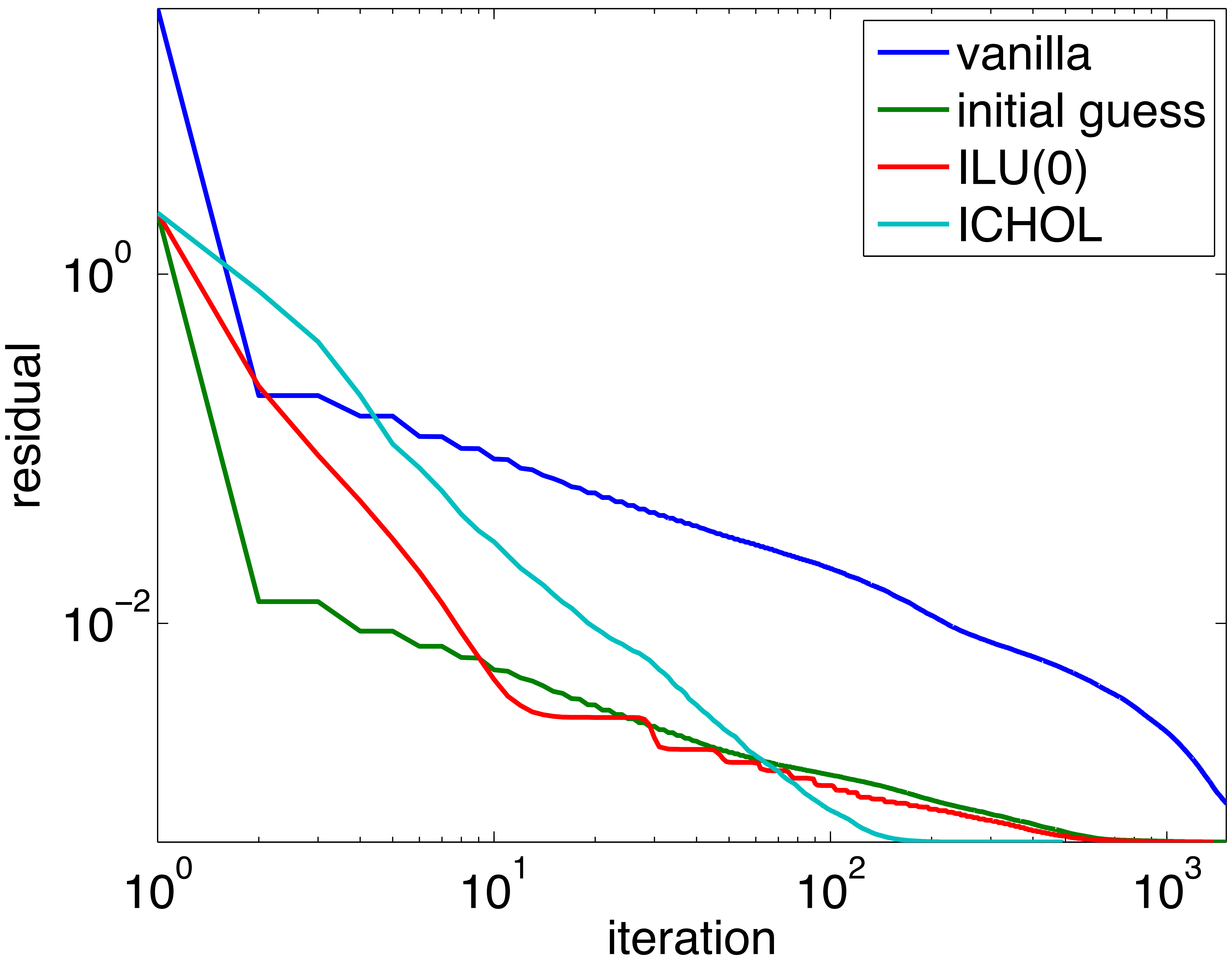
Preconditioners

Use right-preconditioner and solve

$$\begin{pmatrix} A \\ P \end{pmatrix} M^{-1}\mathbf{v} \approx \begin{pmatrix} \mathbf{q} \\ \mathbf{d} \end{pmatrix}$$

$$\mathbf{u} = M^{-1}\mathbf{v}$$

- incomplete LU of A
- incomplete Choleski of $A^*A + P^*P$
- ...


vanilla
initial guess
ILU(0)
ICHOL
residual
iteration
10^0
10^{-2}
10^0
10^1
10^2
10^3

Preconditioners

Helmholtz:

- Multigrid (Erlangga ’05)
- Sweeping (Enquist ’10, Pouson ’12)
- Factorization (Wang ’11)

Data-augmented (incomplete QR etc.)

- Multi-Level QR (Li & Saad ‘06)
- HSS-based methods

Time-domain

$$\begin{pmatrix} M \\ 0 \end{pmatrix} \mathbf{u}''(t) + \begin{pmatrix} S \\ P \end{pmatrix} \mathbf{u}(t) \approx \begin{pmatrix} \mathbf{q}(t) \\ \mathbf{d}(t) \end{pmatrix}$$

Time-discretization

Standard Leap-frog leads to

$$\begin{pmatrix} M \\ 0 \end{pmatrix} \mathbf{u}^{n+1} + \begin{pmatrix} \Delta t^2 S - 2M \\ \Delta t^2 P \end{pmatrix} \mathbf{u}^{n} + \begin{pmatrix} M \\ 0 \end{pmatrix} \mathbf{u}^{n-1} \approx \Delta t^2 \begin{pmatrix} \mathbf{q}(t) \\ \mathbf{d}(t) \end{pmatrix}$$

We need to compute the least-squares solution globally over all time-steps!

Time-discretization

$$
\begin{pmatrix}
M & & & & \\
\Delta t^2 S - 2M & M & & & \\
M & \Delta t^2 S - 2M & M & & \\
 & M & \Delta t^2 S - 2M & M & \\
 & & \ddots & \ddots & \ddots \\
P & & & & \\
 & P & & & \\
 & & P & & \\
 & & & \ddots &
\end{pmatrix}
\begin{pmatrix}
\mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_N
\end{pmatrix}
\approx
\begin{pmatrix}
\mathbf{q}_0 \\ \mathbf{q}_1 \\ \vdots \\ \mathbf{q}_N \\ \\ \mathbf{d}_0 \\ \mathbf{d}_2 \\ \vdots \\ \mathbf{d}_N
\end{pmatrix}
$$

Time-discretization

Standard leap-frog leads to large (block) overdetermined system

lower-triangular →
diagonal →

$$\begin{pmatrix} L \\ D \end{pmatrix} \mathbf{u} \approx \begin{pmatrix} \mathbf{q} \\ \mathbf{d} \end{pmatrix}$$

Normal equations lead to penta-diagonal system

$$(L^T L + D^T D)\mathbf{u} = L^T \mathbf{q} + D^T \mathbf{d}$$

We can never form the complete matrices, have to solve by forward/ backward substitution!

Conclusions & Future work

Frequency-domain:

- A number of standard tools can be re-used
- preconditioning requires extra attention

Time-domain:

- need to solve LS problem globally over all time-steps
- forward substitution may require many sweeps

Acknowledgements

Thank you!

This work was in part financially supported by the Natural Sciences and Engineering Research Council of Canada Discovery Grant (22R81254) and the Collaborative Research and Development Grant DNOISE II (375142-08). This research was carried out as part of the SINBAD II project with support from the following organizations: BG Group, BGP, BP, CGG, Chevron, ConocoPhillips, ION, Petrobras, PGS, Total SA, WesternGeco, and Woodside.