Marchenko

Jan Thorbecke, Joeri Brackenhoff,
Lele Zhang

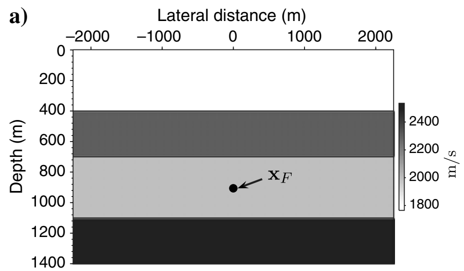
j.w.thorbecke@tudelft.nl
j.a.brackenhoff@tudelft.nl
L.Zhang-1@tudelft.nl

Introduction Marchenko method

Marchenko focusing functions

$$G^{+}(\mathbf{x}_F,\mathbf{x}_R,t) = -\int_{\partial\mathbb{D}_0}\int_{t'=-\infty}^{t} R(\mathbf{x}_R,\mathbf{x},t-t')f_1^{-}(\mathbf{x},\mathbf{x}_F,-t')\mathrm{d}t'\mathrm{d}\mathbf{x} + f_1^{+}(\mathbf{x}_R,\mathbf{x}_F,-t)$$

$$G^{-}(\mathbf{x}_F,\mathbf{x}_R,t) = \int_{\partial\mathbb{D}_0}\int_{t'=-\infty}^{t} R(\mathbf{x}_R,\mathbf{x},t-t')f_1^{+}(\mathbf{x},\mathbf{x}_F,t')\mathrm{d}t'\mathrm{d}\mathbf{x} - f_1^{-}(\mathbf{x}_R,\mathbf{x}_F,t)$$



Velocity model

Reference: Thorbecke, J., Slob, E., Brackenhoff, J., van der Neut, J., and Wapenaar, K., 2017, *Implementation of the Marchenko method*: Geophysics, Vol. 82 (6), WB29-WB45.

Marchenko focusing functions

$$\mathbf{G}^{+} = -\mathbf{R}f_1^{-,*} + f_1^{+,*}$$

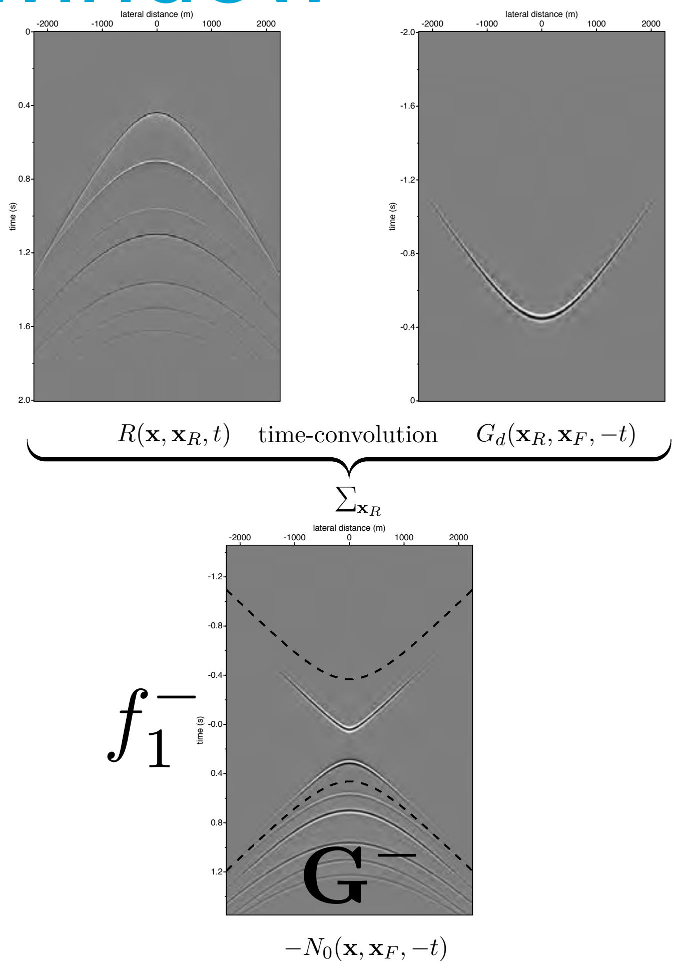
$$\mathbf{G}^{-} = \mathbf{R}f_1^{+} - f_1^{-}$$

Marchenko equations

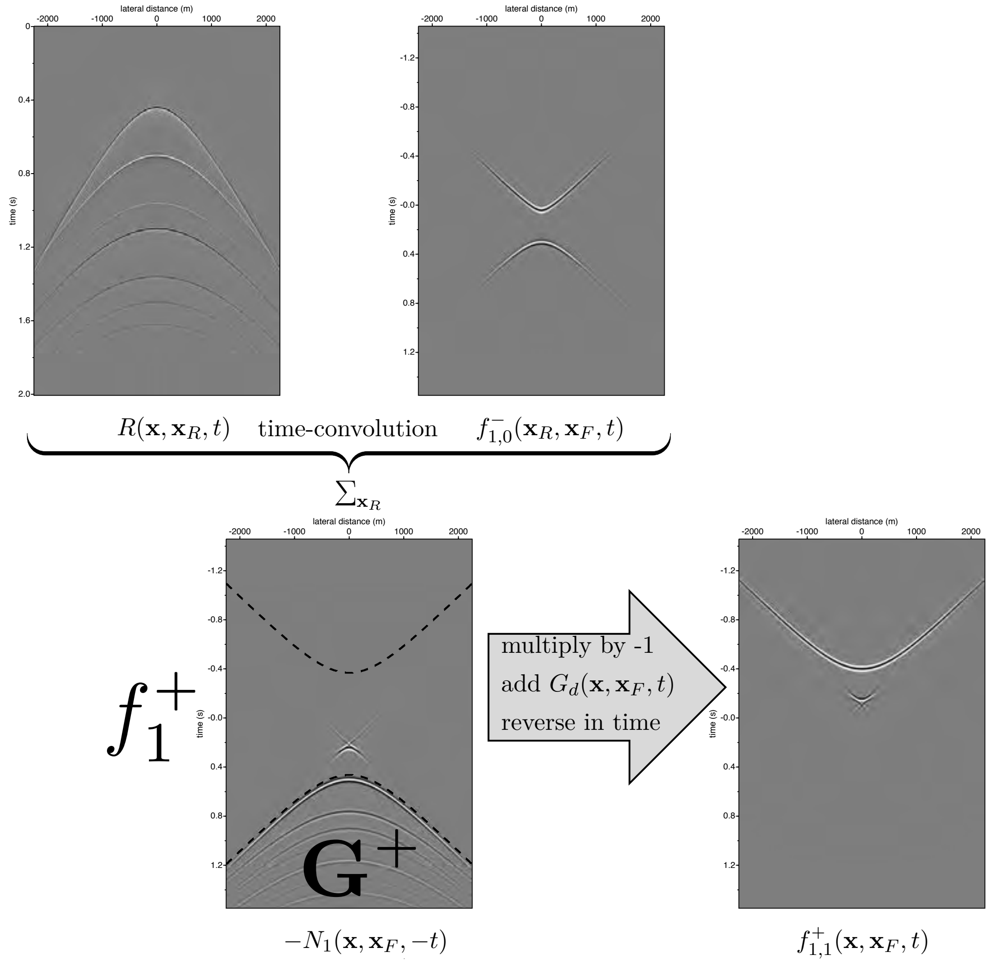
$$0 = -\theta \mathbf{R} f_1^{-,*} + f_1^{+,*}$$

$$0 = \theta \mathbf{R} f_1^{+} - f_1^{-}$$

Time window θ



First iteration



Φ R N_i

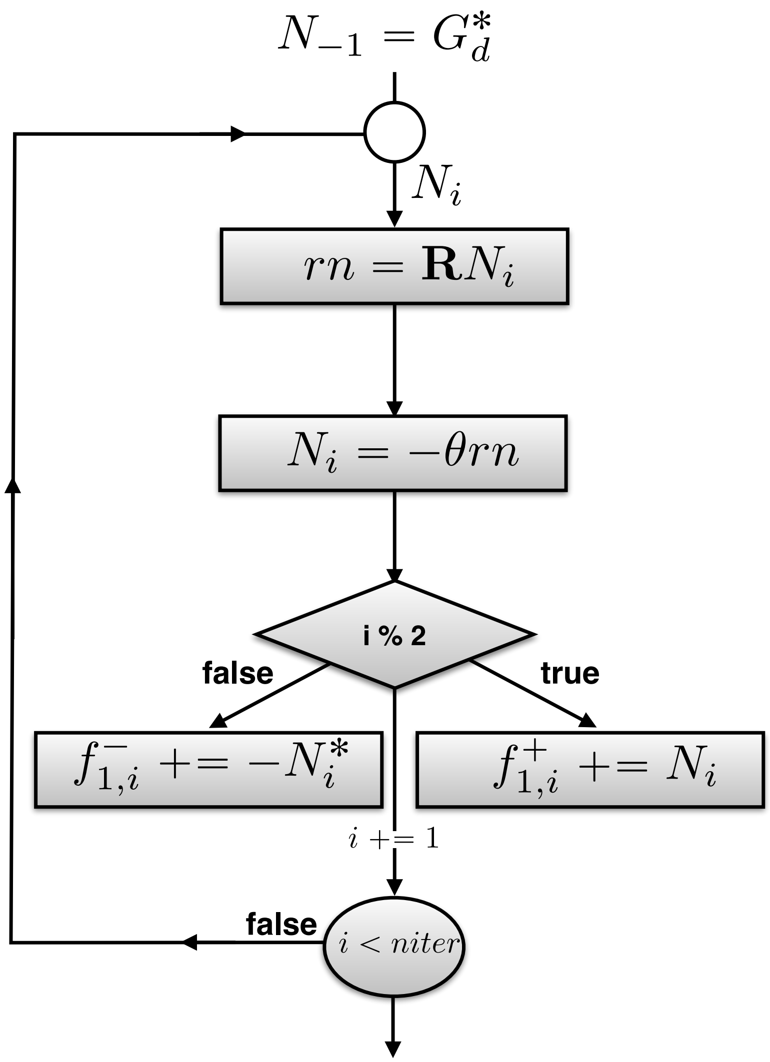
$$N_{-1}(\mathbf{x}_R, \mathbf{x}_F, -t) = G_d(\mathbf{x}, \mathbf{x}_F, -t'),$$

$$N_i(\mathbf{x}_R, \mathbf{x}_F, -t) = -\theta_t \int_{\partial\mathbb{D}_0} \int_{t'} R(\mathbf{x}_R, \mathbf{x}, t - t') N_{i-1}(\mathbf{x}, \mathbf{x}_F, t') \mathrm{d}t' \mathrm{d}\mathbf{x},$$

Time-domain

Time convolution

$$N_i = \Phi \, R \, N_{i-1}$$


$N_{-1} = G_d^*$
N_i
$rn = \mathbf{R}N_i$
$N_i = -\theta rn$
i % 2
false
true
$f_{1,i}^{-} += -N_i^*$
$f_{1,i}^{+} += N_i$
$i += 1$
false
$i < niter$

Algorithm

```
Main begin
    Reading SU-style input Data and Allocate arrays
    Initialisation
    Ni(t) = f2p(t) = f1plus(t) = G_d(-t)
    f1min(t) = pmin(t) = 0.0
    for iter ← 0 to niter do
        synthesis(Refl, Ni, iRN)
        Ni(t)=-iRN(-t)
        pmin(t)+=iRN(t)
        applyMute(Ni, muteW)
        f2p(t) += Ni(t)
        else if (iter % 2 == 0) then
            f1min(t) -= Ni(-t)
        else
            f1plus(t) += Ni(t)
        end
    end
    Green(t) = pmin(t) + f2p(-t)
end
```

$R\,N_i$

Φ

Synthesis: R N_i

```
synthesis(Refl, Ni, iRN)
begin
    iRN = 0
    ∀l, i: Fop(l,ω,i) = ℱ {Ni(l,i,t)}
    for k ← 0 to nshots do
        #pragma omp parallel for
        for l ← 0 to Nfoc do
            for ω ← ω_min to ω_max do
                sum(ω) = 0
                for i ← 0 to nrecv do
                    sum(ω) += Refl(k,ω,i) * Fop(l,ω,i)
                end
            end
            iRN(l,k,t) = ℱ^{-1} {sum(ω)}
        end
    end
end
```

Synthesis: R N_i

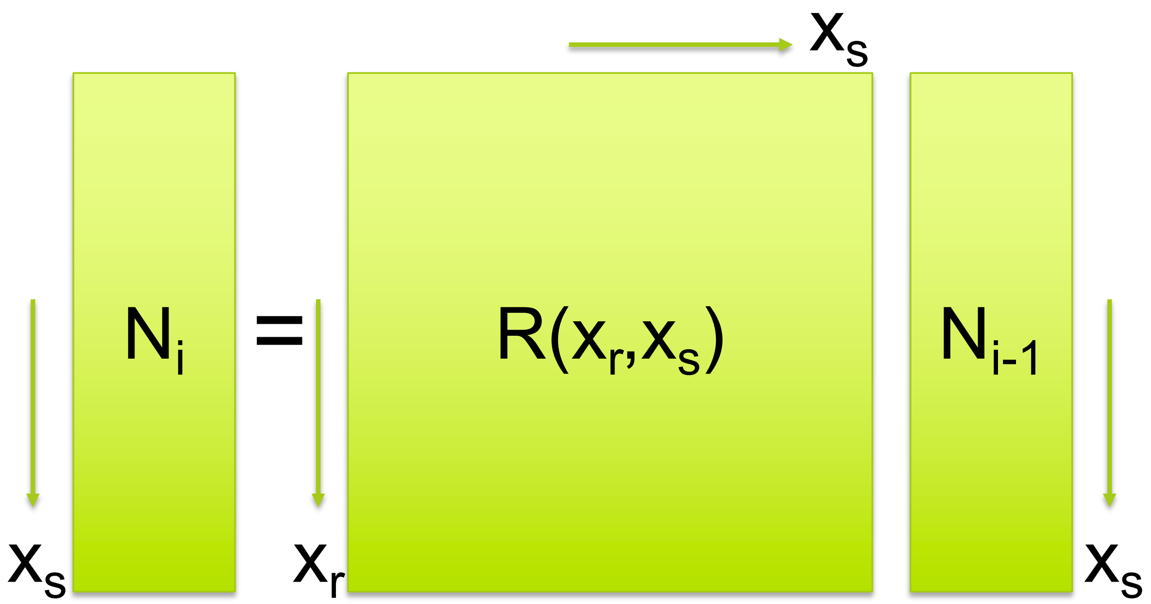
```
synthesis(Refl, Ni, iRN)
begin
    iRN = 0
    ∀l, i: Fop(l,ω,i) = F {Ni(l,i,t)}
    for k ← 0 to nshots do
        #pragma omp parallel for
        for l ← 0 to Nfoc do
            for ω ← ω_min to ω_max do
                sum(ω) = 0
                for i ← 0 to nrecv do
                    sum(ω) += Refl(k,ω,i) * Fop(l,ω,i)
                end
            end
            iRN(l,k,t) = F^-1 {sum(ω)}
        end
    end
end
```

FFT's

Synthesis: R N_i

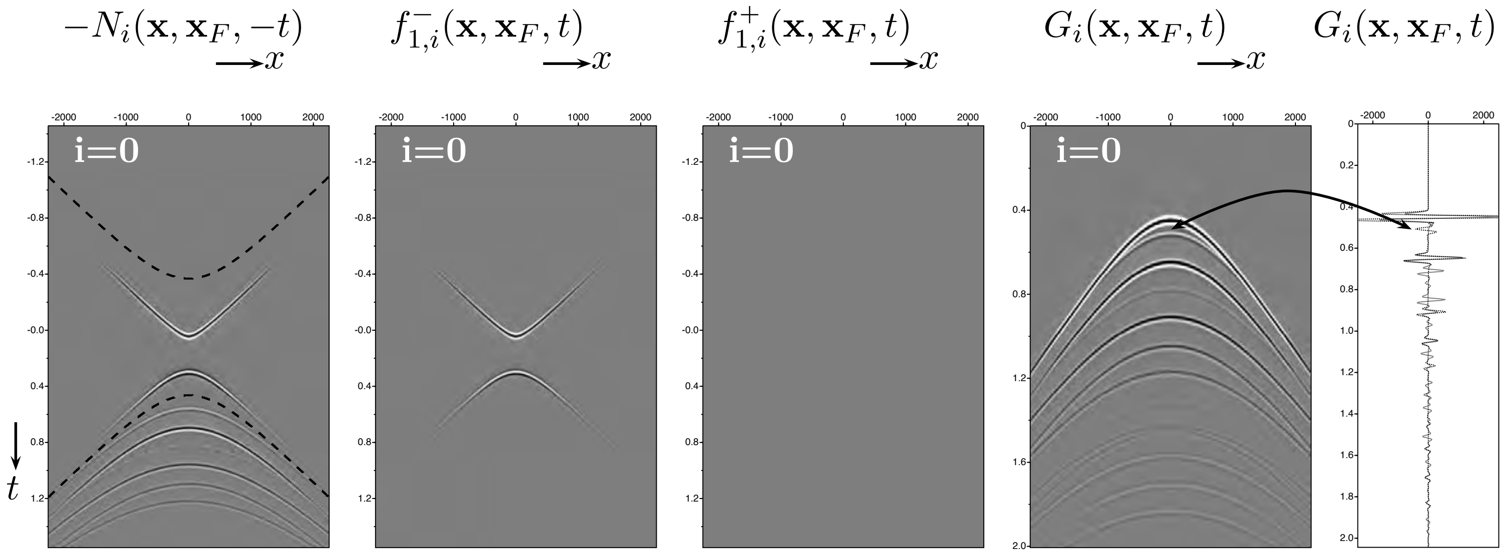
$N_i(x_s,t) \Rightarrow N_i(x_s,\omega)$

$R(x_r,x_s,t) \Rightarrow R(x_r,x_s,\omega)$

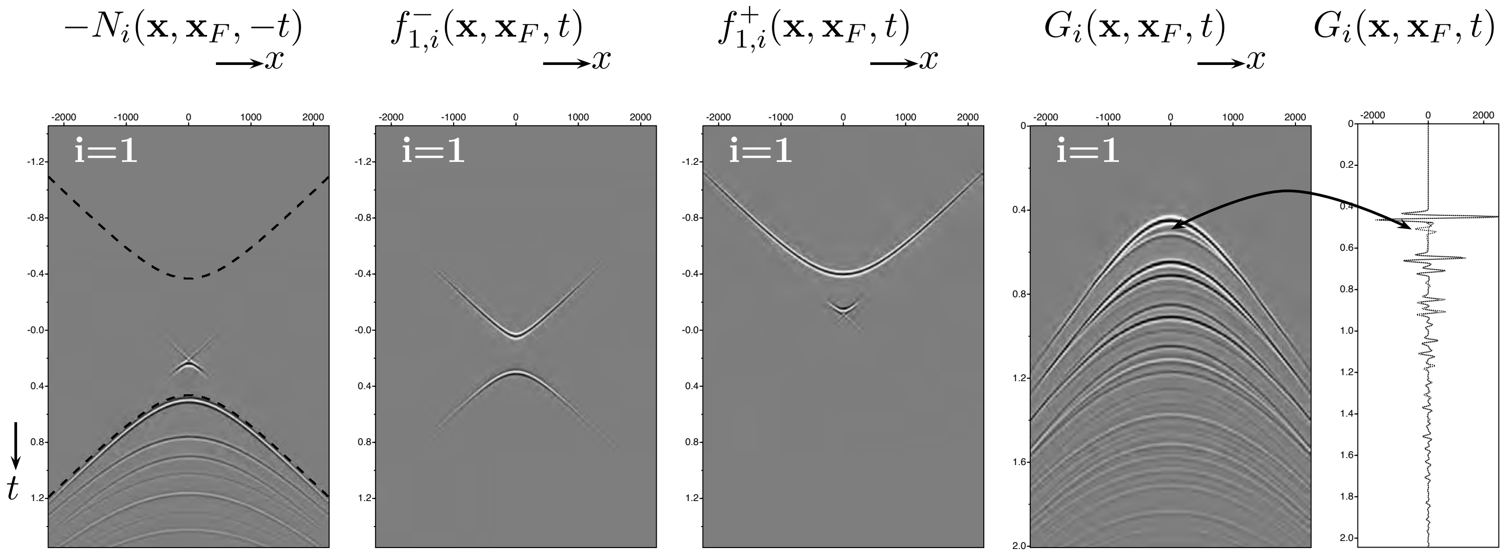


Implementation

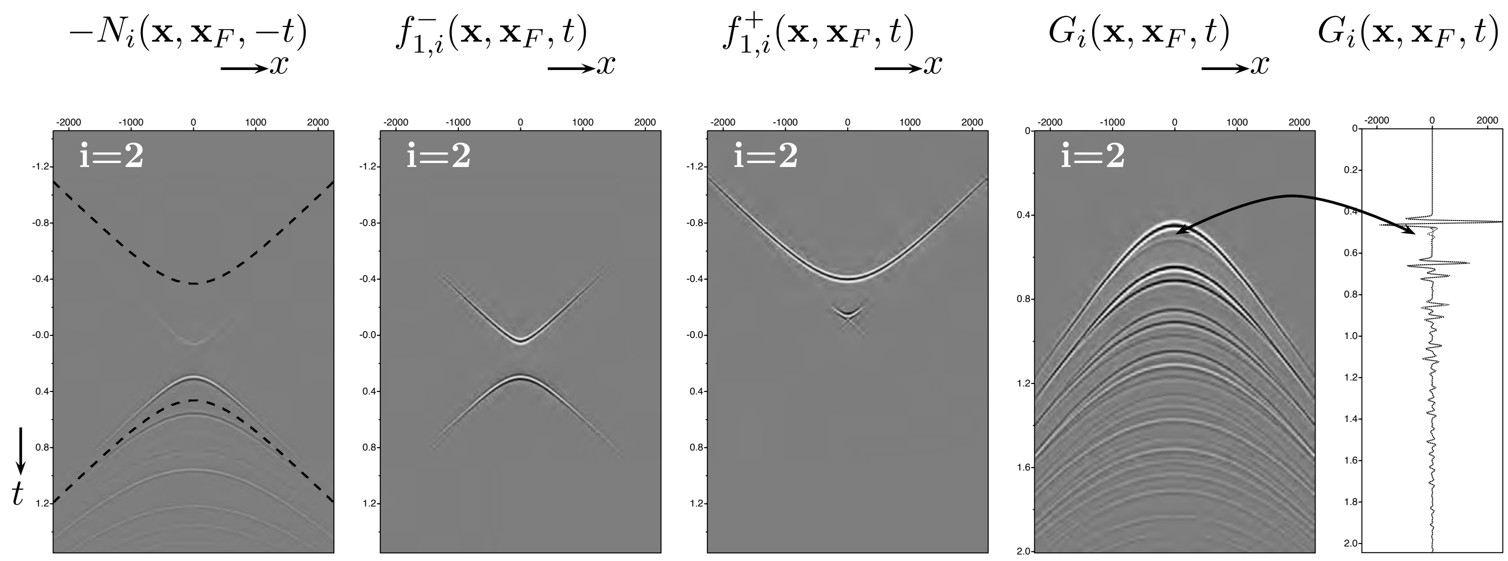
```
#pragma omp for schedule(guided,1)
        for (k=0; k<nshots; k++) {
            ixsrc = NINT((xsrc[k] - fxsb)/dxs);
            for (l = 0; l < Nfoc; l++) {
                memset(&sum[0].r,0,nfreq*2*sizeof(float));
                for (j = nw_low, m = 0; j <= nw_high; j++, m++) {
                    for (i = 0; i < xnx[k]; i++) {
                        ix = ixrcv[k*nx+i];
                        sum[j].r += Refl[k*nw*nx+m*nx+i].r*Fop[l*nw*nxs+m*nxs+ix].r -
                                    Refl[k*nw*nx+m*nx+i].i*Fop[l*nw*nxs+m*nxs+ix].i;
                        sum[j].i += Refl[k*nw*nx+m*nx+i].i*Fop[l*nw*nxs+m*nxs+ix].r +
                                    Refl[k*nw*nx+m*nx+i].r*Fop[l*nw*nxs+m*nxs+ix].i;
                    }
                }
                /* transfrom result back to time domain */
                cr1fft(sum, rtrace, ntfft, 1);
                /* place result at source position ixsrc; dx = receiver distance */
                for (j = 0; j < nts; j++)
                    iRN[l*size+ixsrc*nts+j] += rtrace[j]*scl*dx;
            } /* end of Nfoc loop */
        } /* end of parallel nshots (k) loop */
```

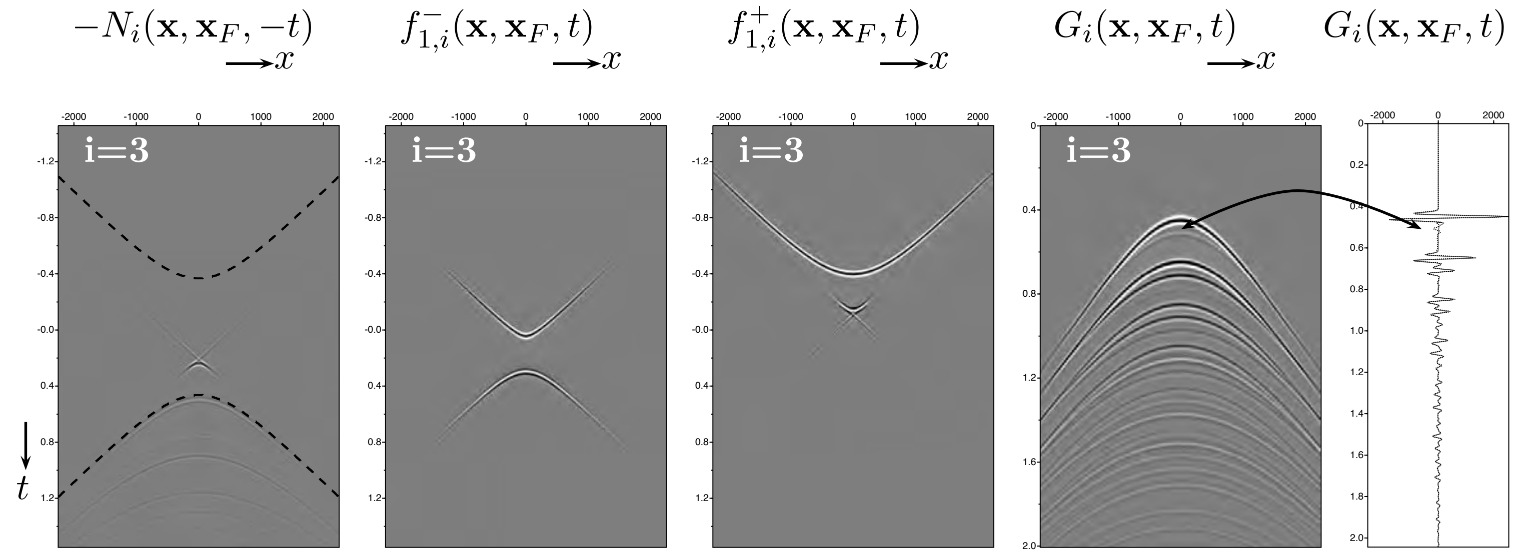


$-N_i(\mathbf{x}, \mathbf{x}_F, -t)$
$f^-_{1,i}(\mathbf{x}, \mathbf{x}_F, t)$
$f^+_{1,i}(\mathbf{x}, \mathbf{x}_F, t)$
$G_i(\mathbf{x}, \mathbf{x}_F, t)$
$G_i(\mathbf{x}, \mathbf{x}_F, t)$
x
t
i=0
i=0
i=0
i=0



$-N_i(\mathbf{x}, \mathbf{x}_F, -t)$
$f^-_{1,i}(\mathbf{x}, \mathbf{x}_F, t)$
$f^+_{1,i}(\mathbf{x}, \mathbf{x}_F, t)$
$G_i(\mathbf{x}, \mathbf{x}_F, t)$
$G_i(\mathbf{x}, \mathbf{x}_F, t)$
x
t
i=1


$-N_i(\mathbf{x}, \mathbf{x}_F, -t)$
$f^-_{1,i}(\mathbf{x}, \mathbf{x}_F, t)$
$f^+_{1,i}(\mathbf{x}, \mathbf{x}_F, t)$
$G_i(\mathbf{x}, \mathbf{x}_F, t)$
$G_i(\mathbf{x}, \mathbf{x}_F, t)$
x
t
i=2


$-N_i(\mathbf{x}, \mathbf{x}_F, -t)$
$f^-_{1,i}(\mathbf{x}, \mathbf{x}_F, t)$
$f^+_{1,i}(\mathbf{x}, \mathbf{x}_F, t)$
$G_i(\mathbf{x}, \mathbf{x}_F, t)$
$G_i(\mathbf{x}, \mathbf{x}_F, t)$
x
t
i=3

access to compute system

- Read MarketingPartnerNetwork agreement
  - citizens from Cuba, Iran, North Korea, Sudan, or Syria contact me.
- Fill in and sign: Training Class User Record
- You will get user name and passwd to access swan:
- ssh –Y trxx@swan.cray.com

  xx: trainee number

accounts are closed, data removed, on June 15

swan.cray.com

- Cray XC50
- Shared resource with other users
- 120 BDW 22-core 2.2 GHz nodes
- 64 SKL 28-core 2.1 GHz nodes
- 64 SKL 20-core 2.4 GHz nodes
- node has 2 processors (40, 44 or 56 cores)
- workload manager is PBSpro

Display of results

```
suximage < result.su perc=99 &
```

or

```
supsimage < result.su perc=99 > result.eps
convert result.eps ~/result.png
```

Then copy files to your local machine

```
scp trxx@swan.cray.com:~/*png .
```

Environment of swan

- set up environment,
  create working directory,
  `/lus/scratch/$USER`
  copy source code and data

```
source ~jan/WS15setup.sh
```

Installation Marchenko code

Latest version can be found on

```
git clone https://github.com/JanThorbecke/OpenSource.git
```

for this workshop copy has been prepared

```
rsync -av /lus/scratch/jan/OpenSource /lus/scratch/$USER/
```

Continue with step 2 in

```
OpenSource/marchenko/demo/WS15/README.1
```

and (re)compile the code.

Working on demo/twoD

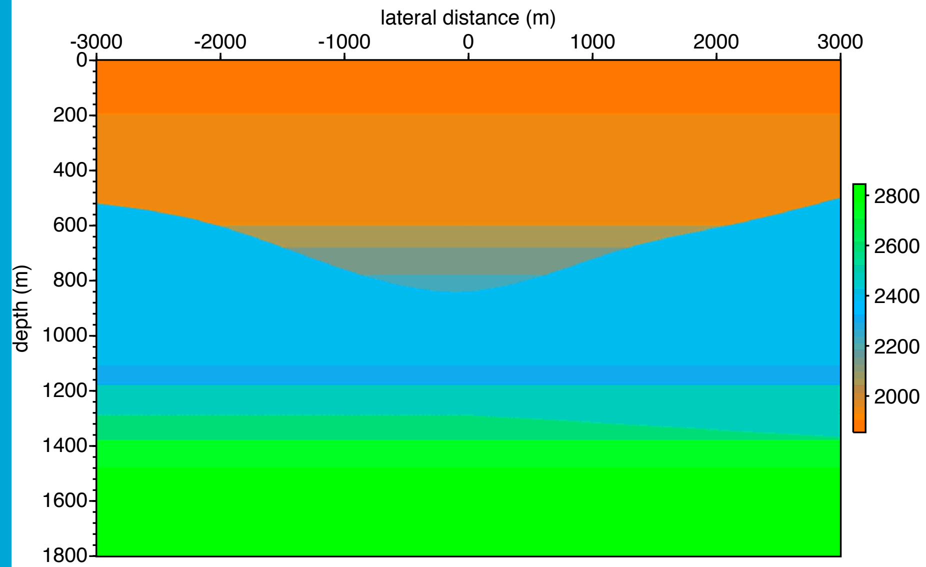
- demo/WS15/README.2
- Reproduce 1D examples from paper on 2D model
- Experiment with
  - Number of iterations.
  - Change amplitude of R by 2 or 0.5.
  - What wavelet is used to model R?
  - Investigate intermediate results.

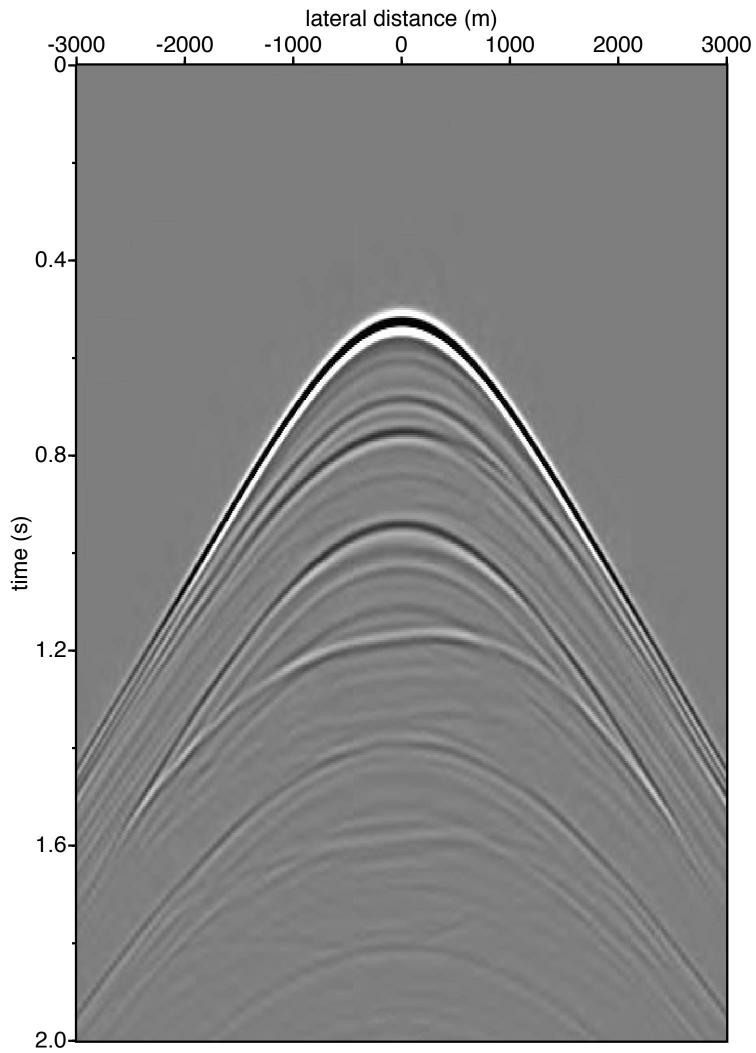
45 min.

CRAY

TU Delft

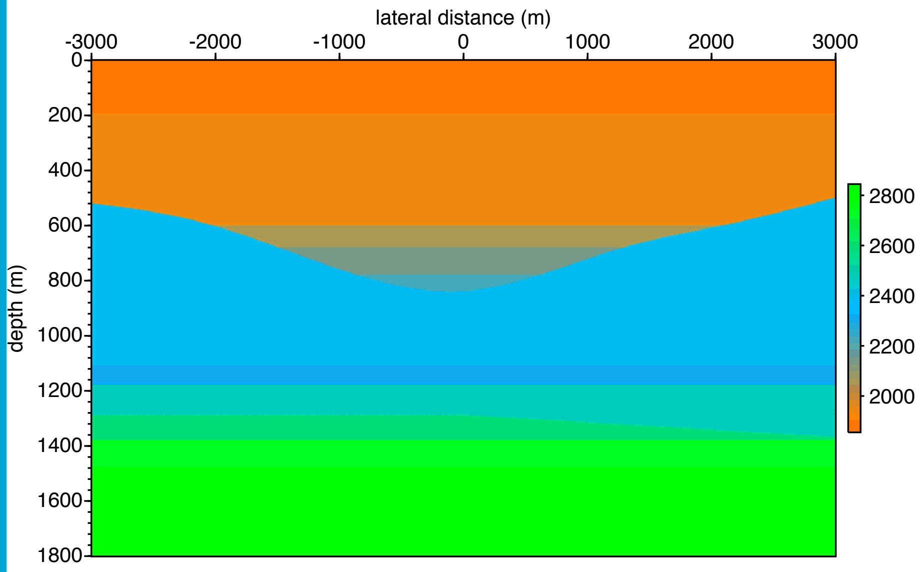
twoD figures

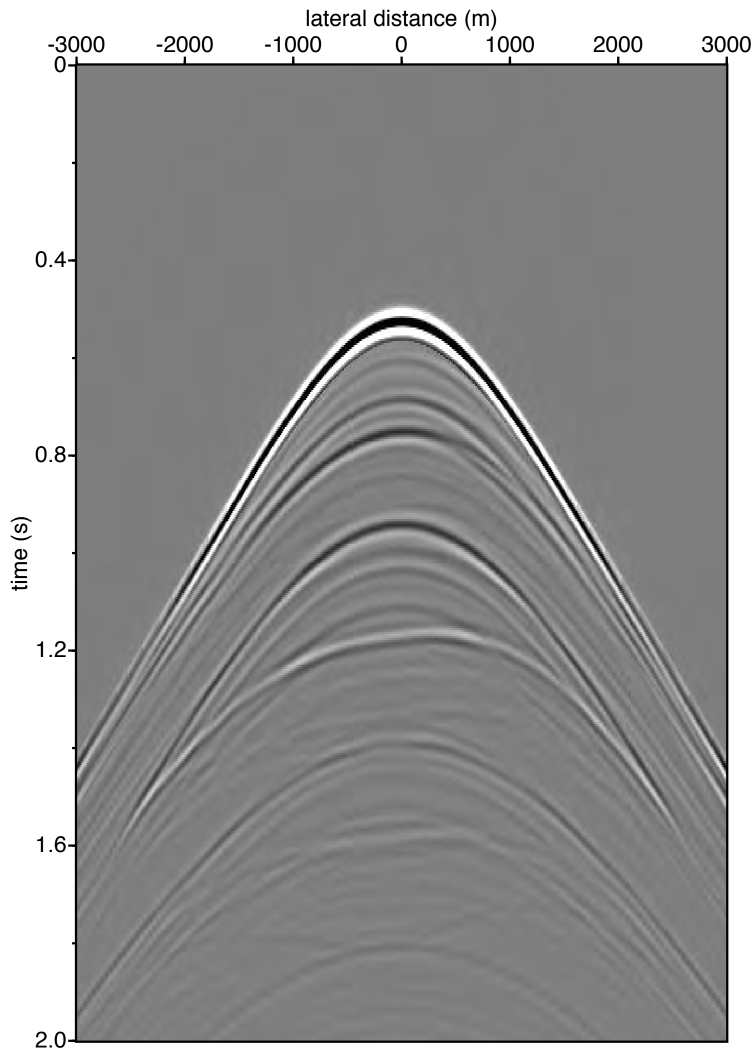




Reference

twoD figures





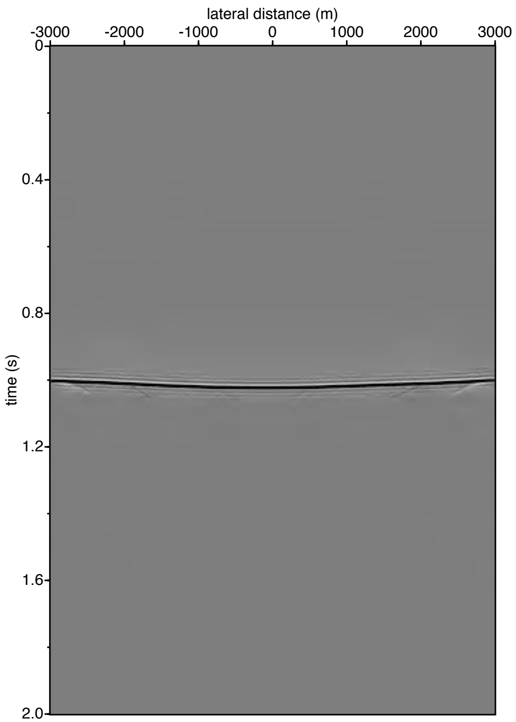
Marchenko

Use of plane-waves in Marchenko

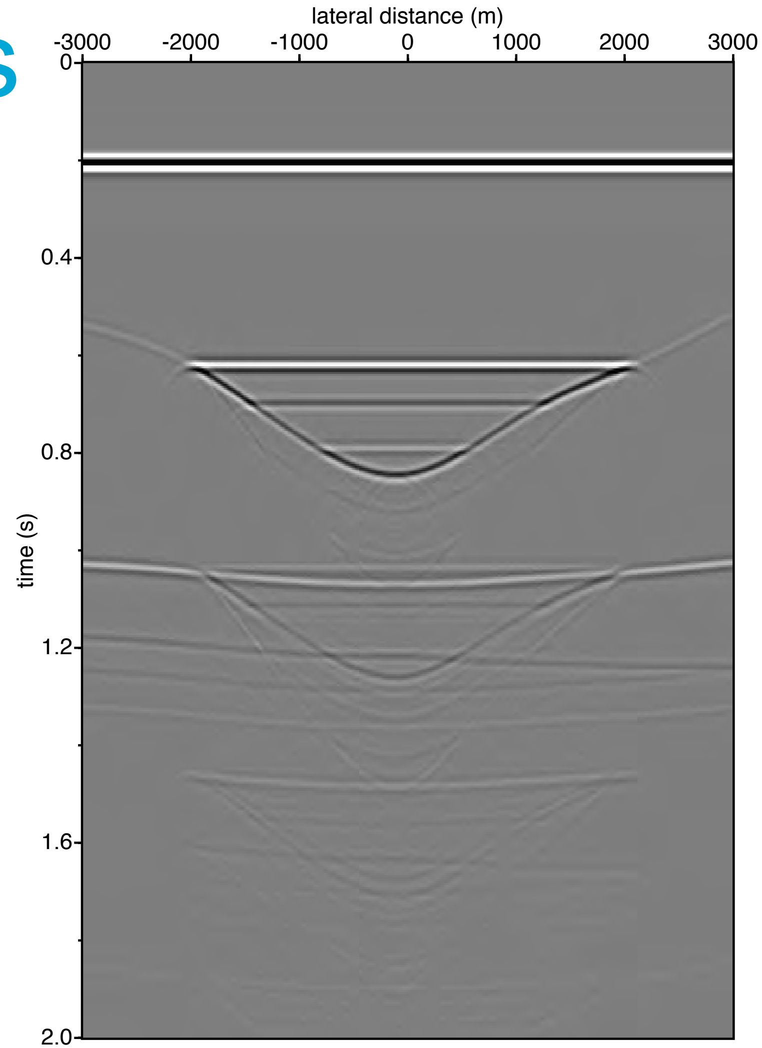
- README.3
- Marchenko plane-wave method
- Based on R-data in demo/twoD
- Based on work of Giovanni Meles **15 min.**

Reference: Meles, G.A., Wapenaar, K., and Thorbecke, J., 2018, *Virtual plane-wave imaging via Marchenko redatuming*: Geoph. J. Int., Vol. 214, 508-519.

Plane-wave results

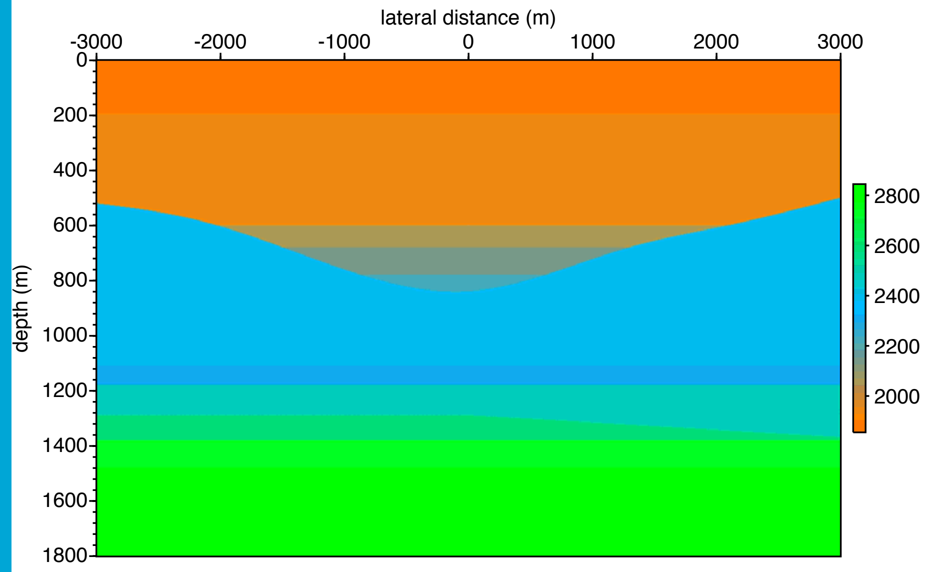


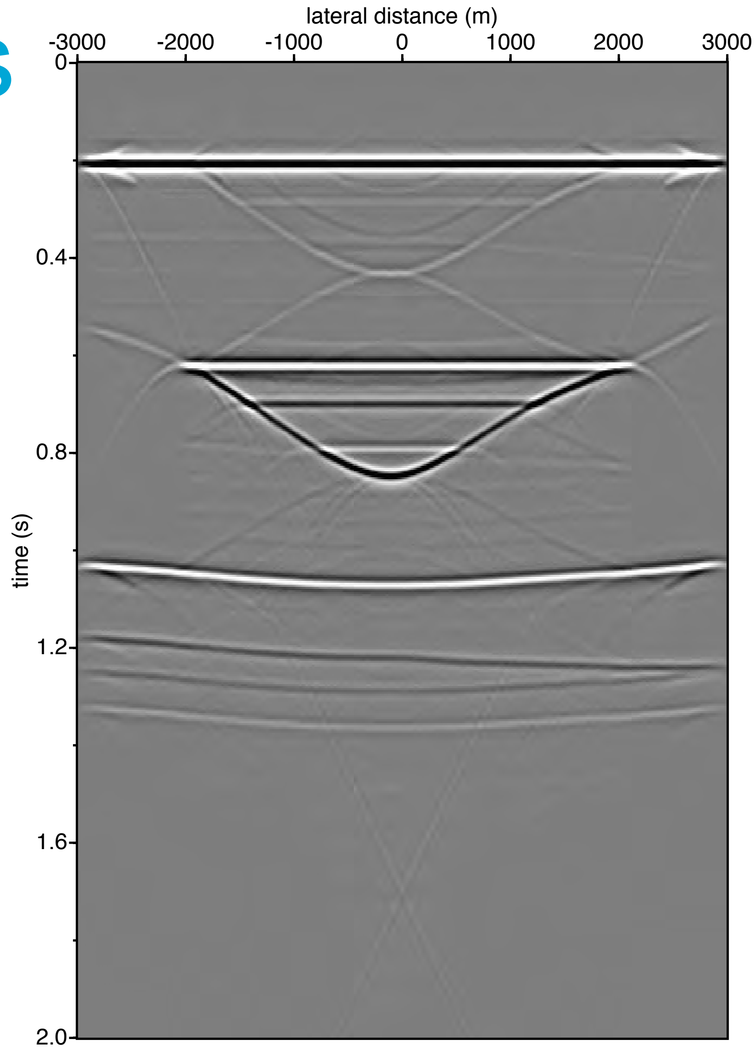
iniPlane_z1800_rp_muted



ZOW

Plane-wave results



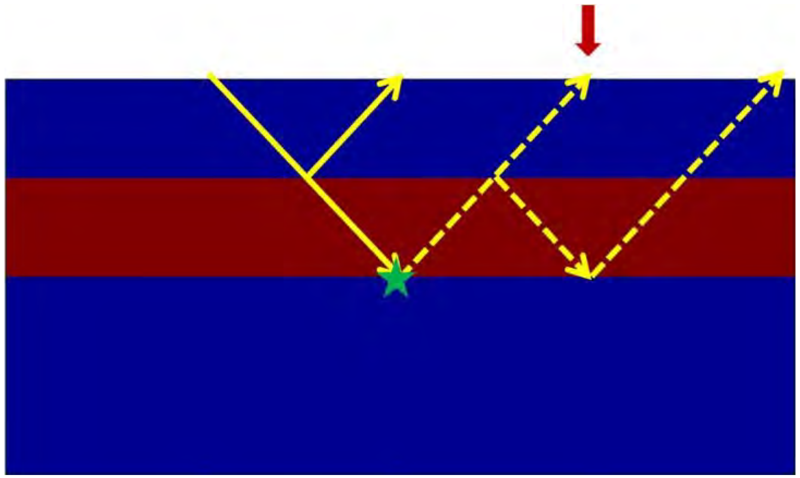


convPlanew

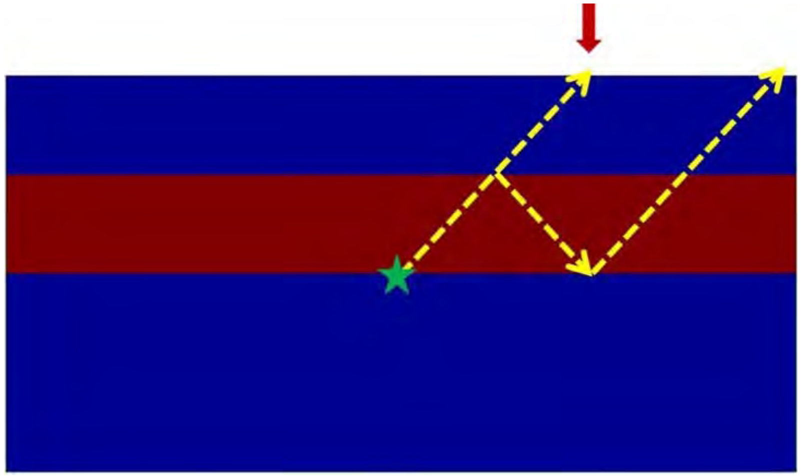
Extracting primaries with Marchenko

Reference: Zhang, L., Thorbecke, J., Wapenaar, K., and Slob, E., 2019, *Transmission compensated primary reflection retrieval in the data domain and consequences for imaging*: Geophysics, Vol. 84 (4), Q27-Q36.

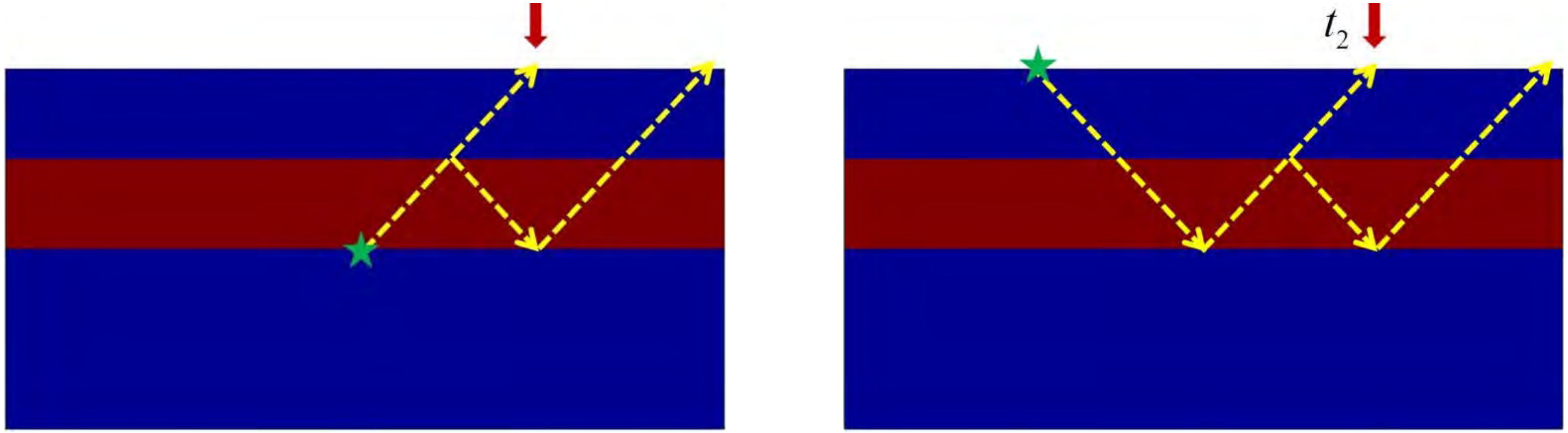
Marchenko Multiple Elimination (MME)



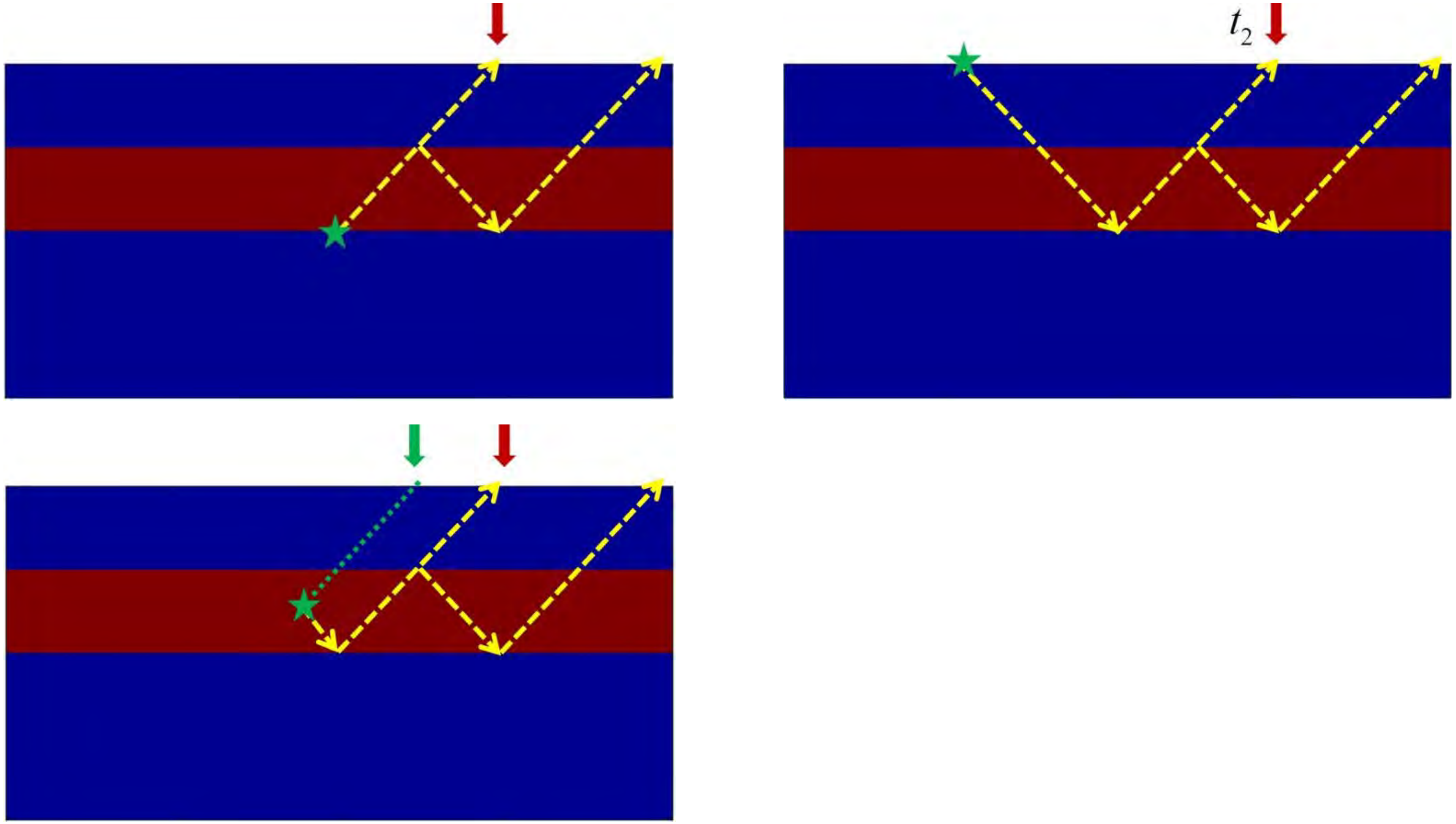
Marchenko Multiple Elimination (MME)



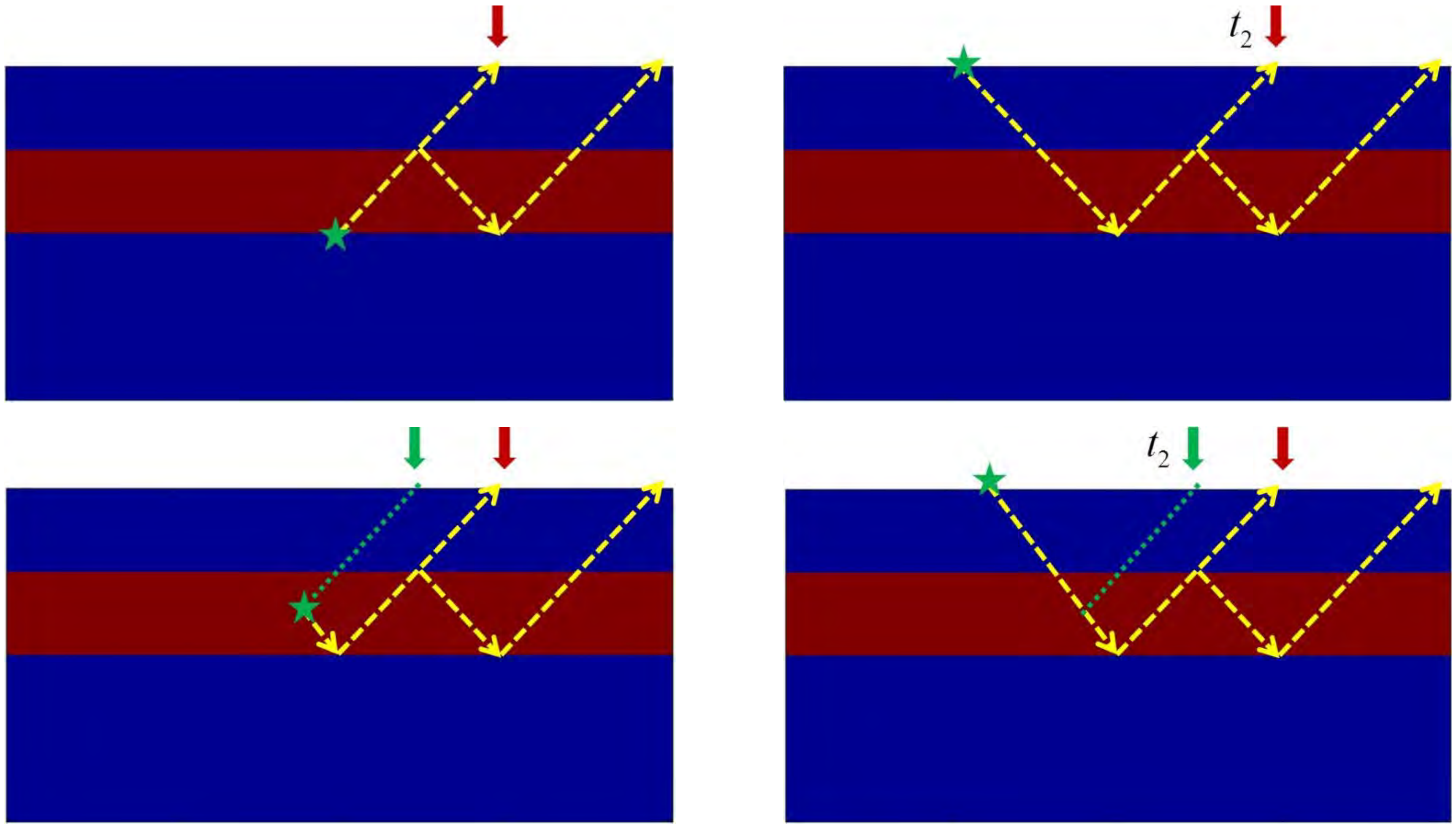
Marchenko Multiple Elimination (MME)

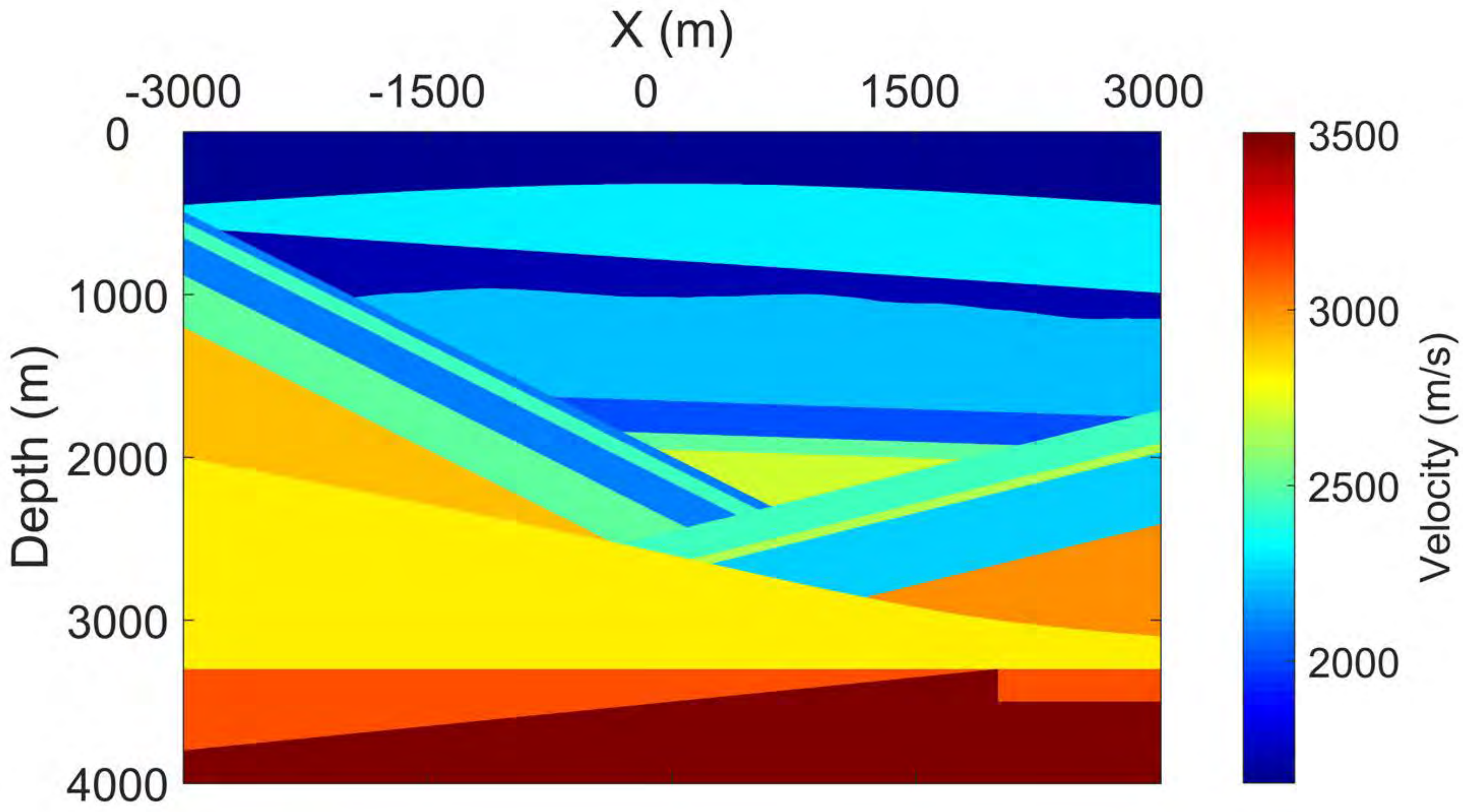


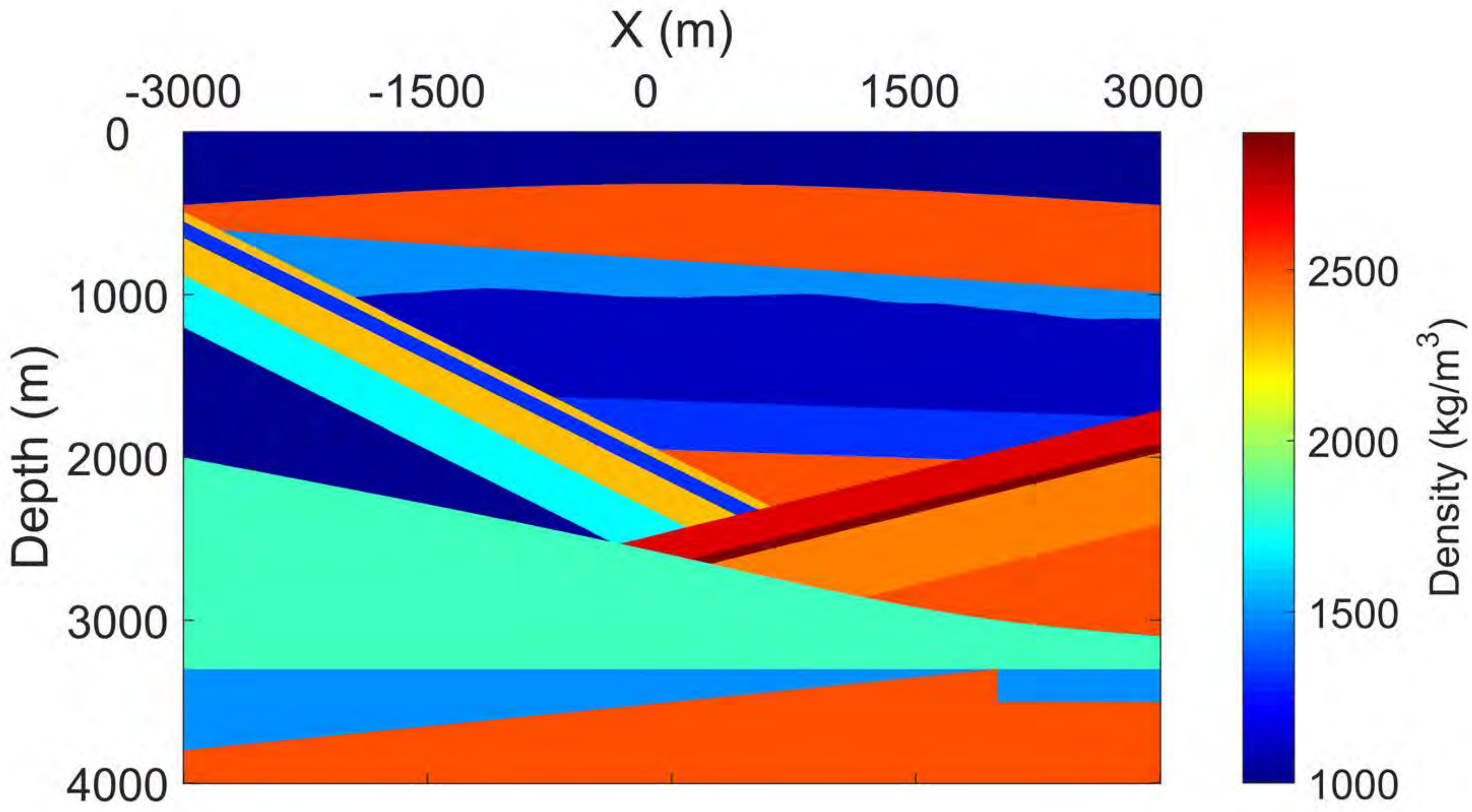
Marchenko Multiple Elimination (MME)



Marchenko Multiple Elimination (MME)

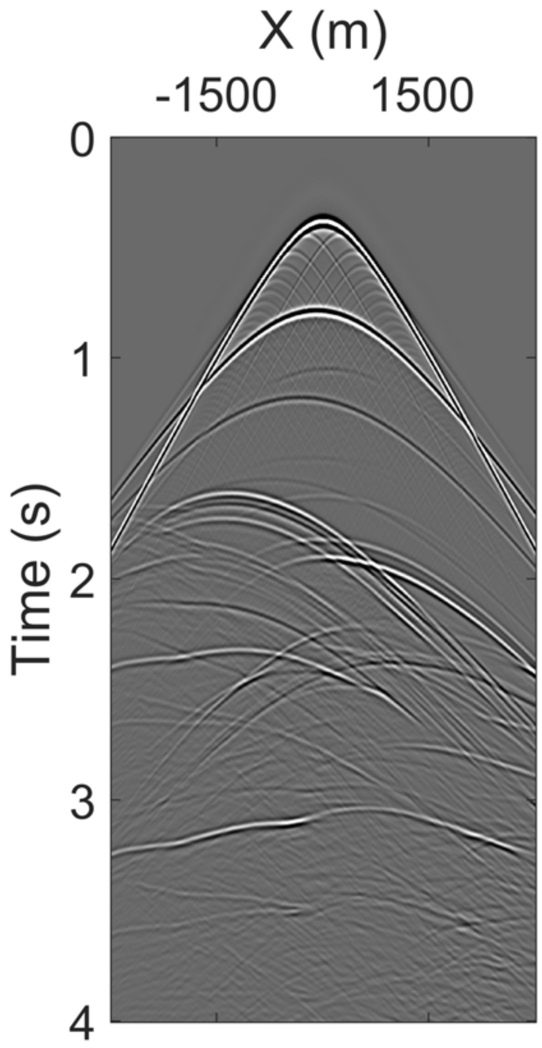



X (m)
-3000
-1500
0
1500
3000
0
1000
2000
3000
4000
Depth (m)
3500
3000
2500
2000
Velocity (m/s)

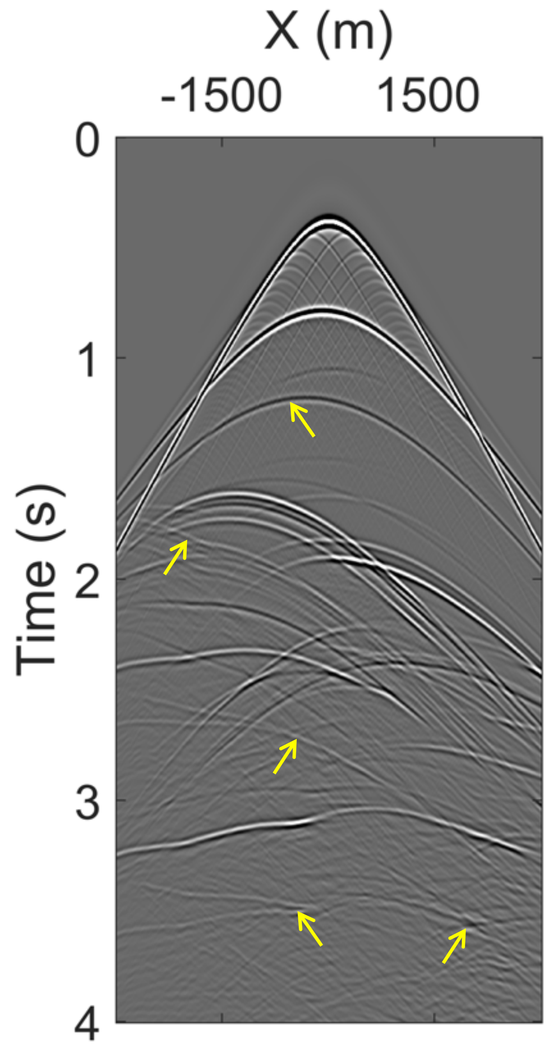


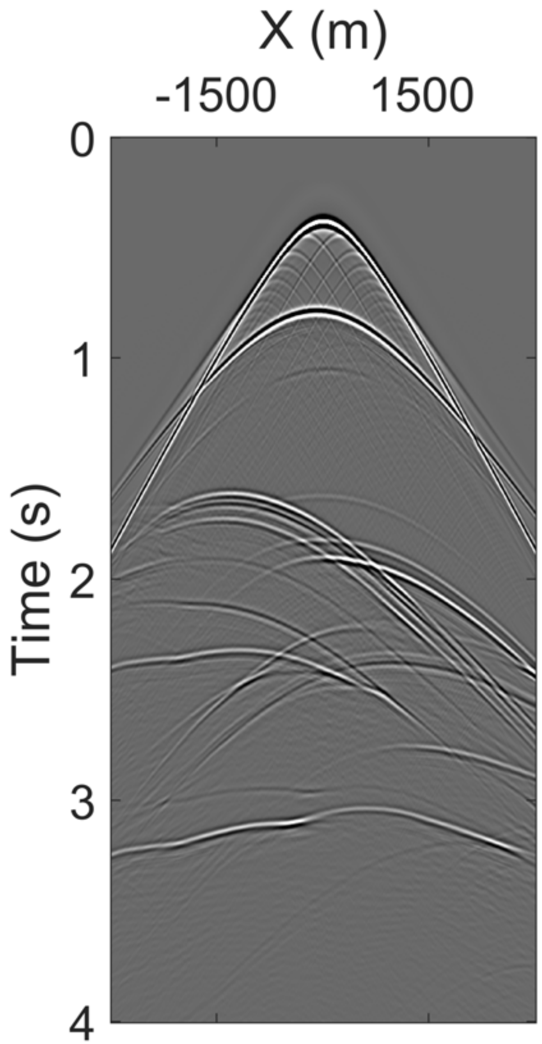
CRAY

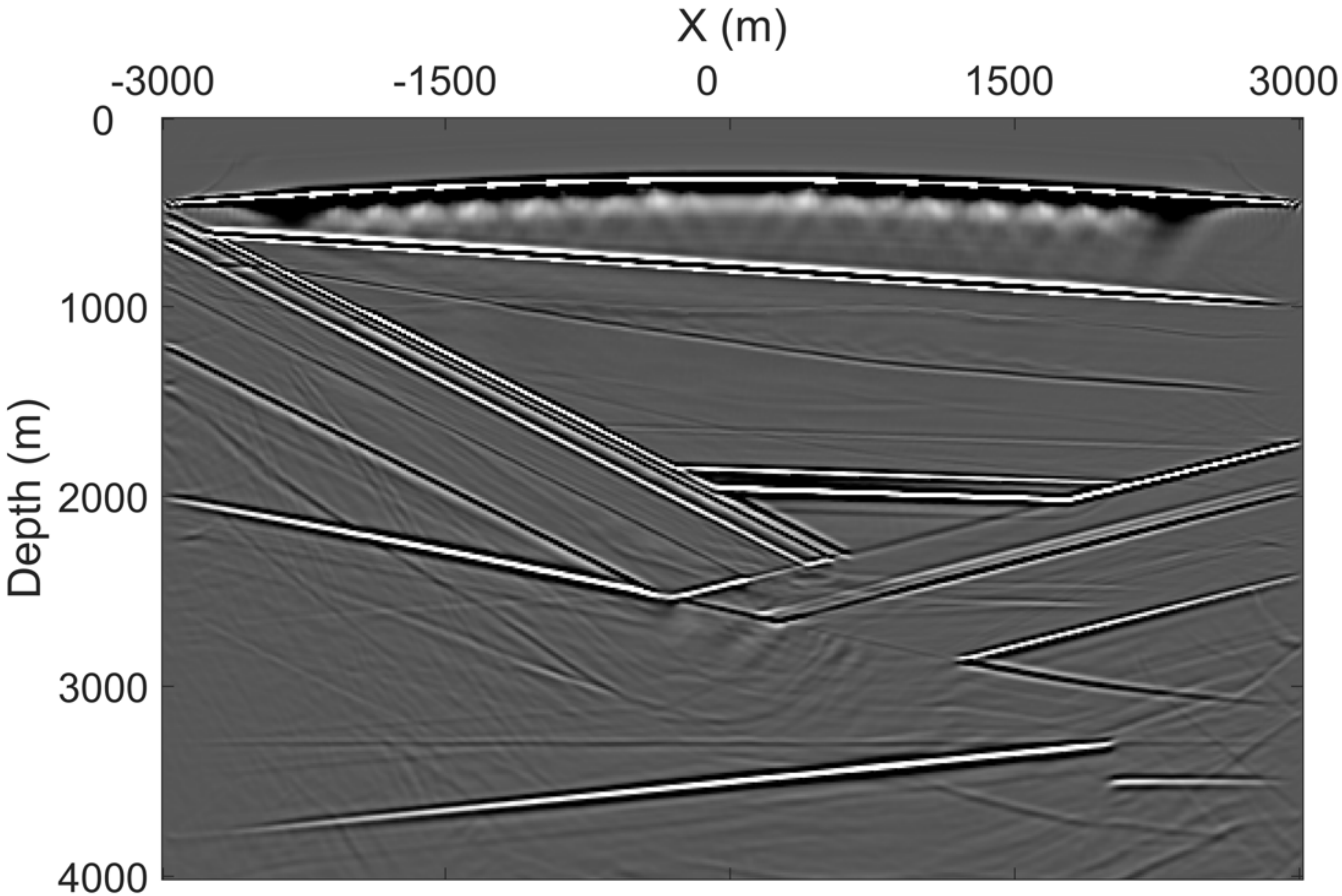
TUDelft



X (m)
-1500
1500
0
1
2
3
4
Time (s)

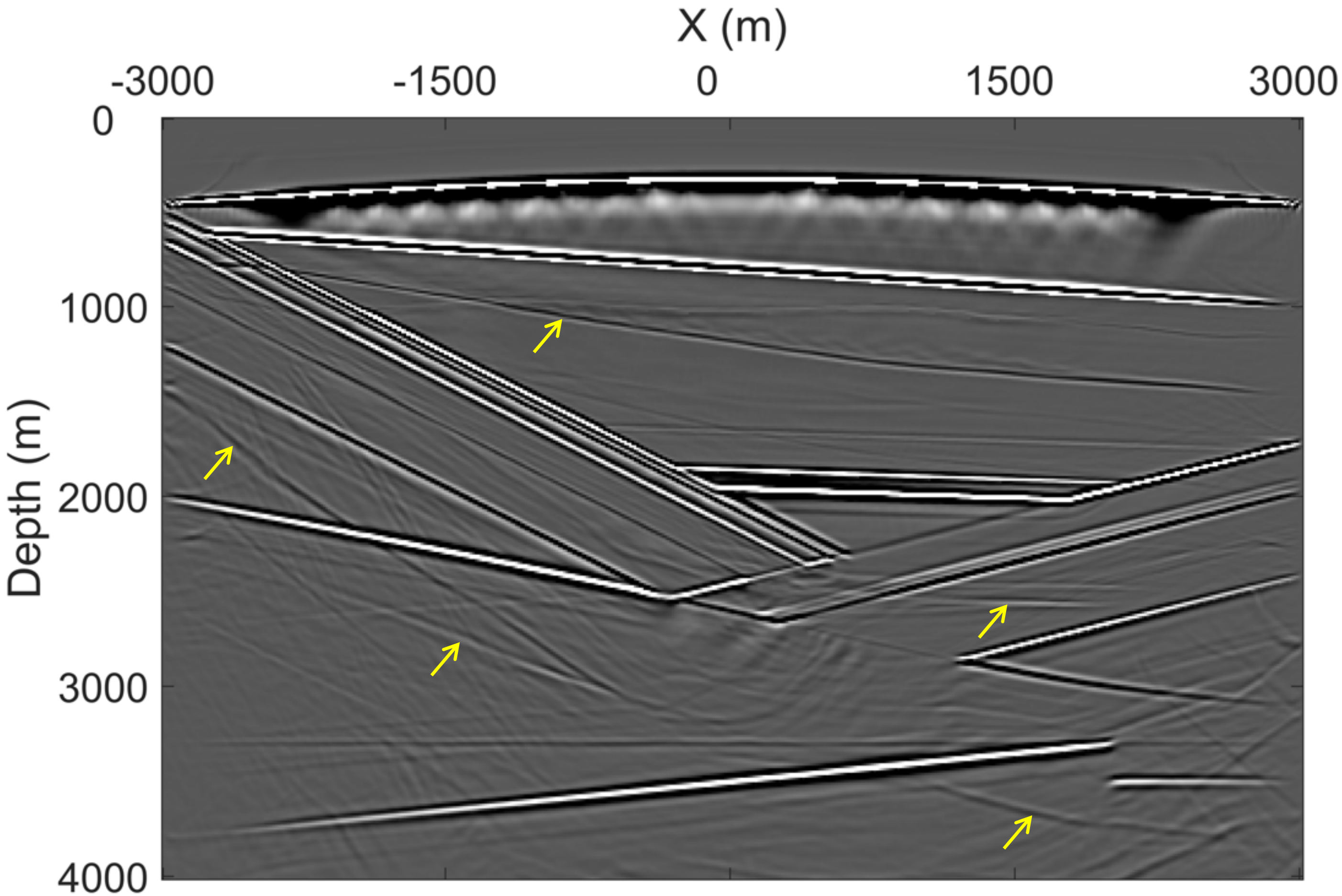

X (m)
-1500
1500
0
1
2
3
4
Time (s)


X (m)
-1500
1500
0
1
2
3
4
Time (s)



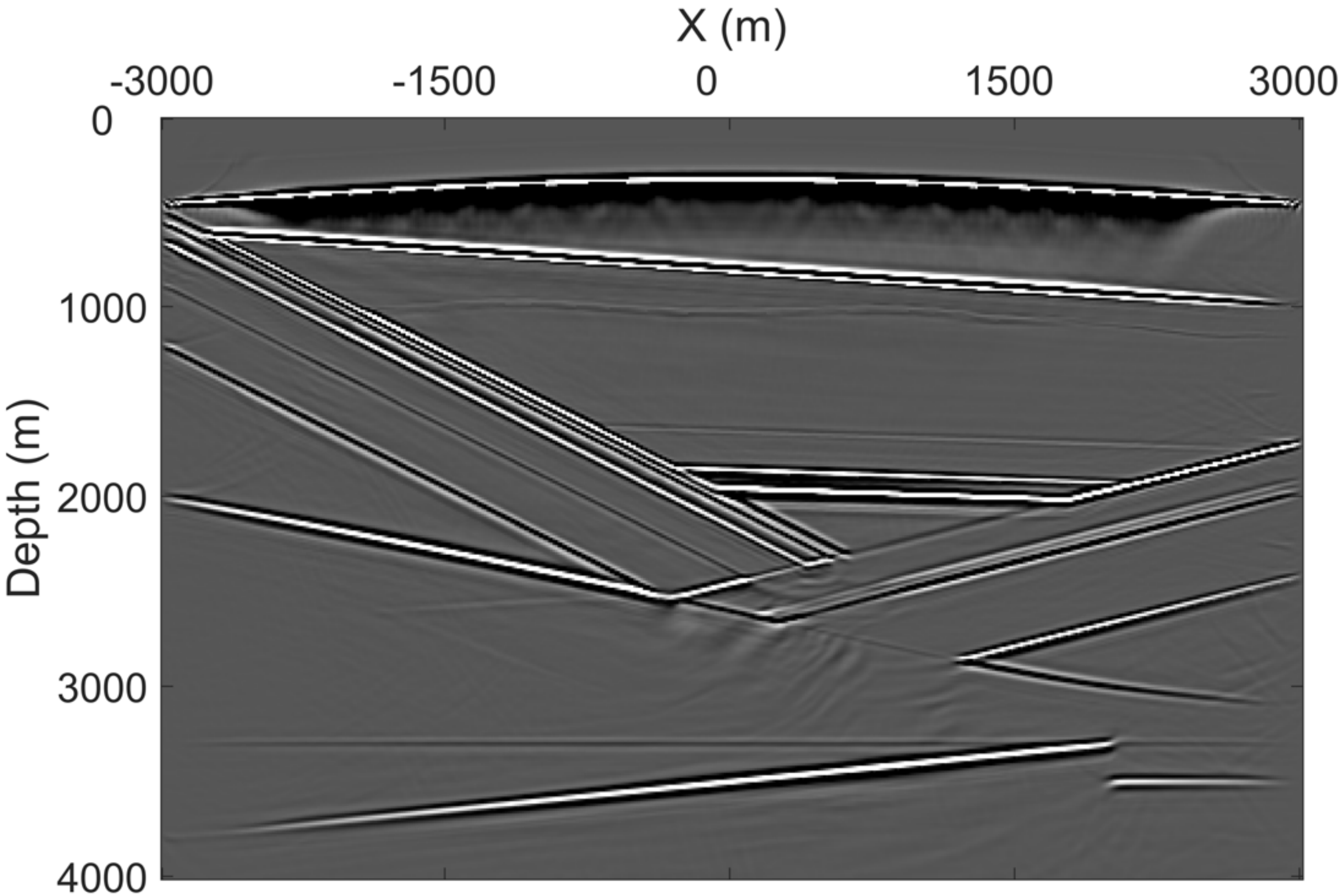
CRAY

TUDelft



CRAY

TUDelft


X (m)
-3000
-1500
0
1500
3000
0
1000
2000
3000
4000
Depth (m)

Extracting primaries with Marchenko

- README.4

- generate ‘primaries only’ for single shot
  - 2D example
  - Invisible medium

15 min.

Reference: Zhang, L., Thorbecke, J., Wapenaar, K., and Slob, E., 2019, *Transmission compensated primary reflection retrieval in the data domain and consequences for imaging*: Geophysics, Vol. 84 (4), Q27-Q36.

Invisible medium puzzle

- Find out how many reflectors there are in the data in demo/invisible. **Only** by inspecting data:

  `shotsdx4_rp.su`

  and using the program

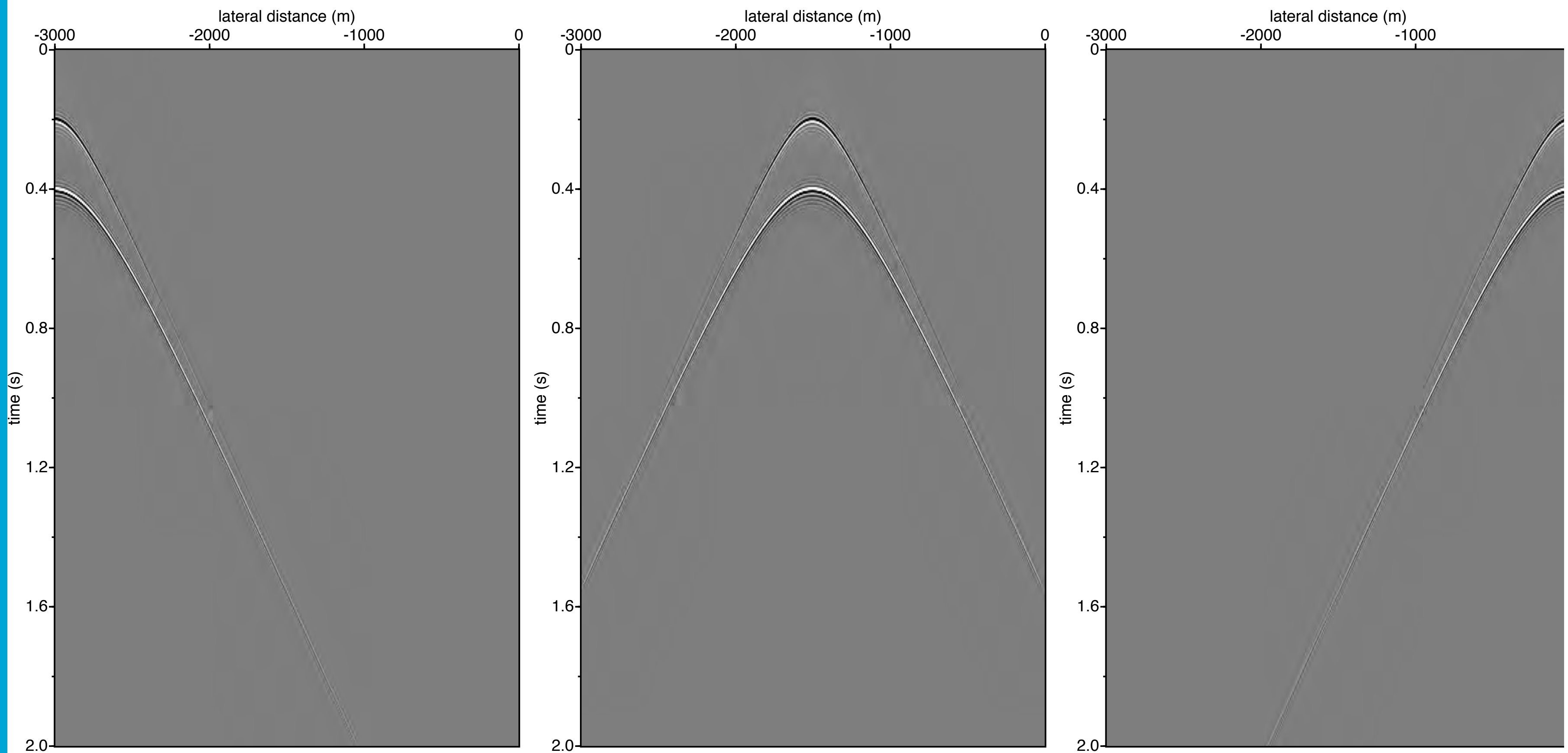
  `marchenko_primaries`

10 min.

*The solution can be found in the *.scr and README, but you are not allowed to read those.*

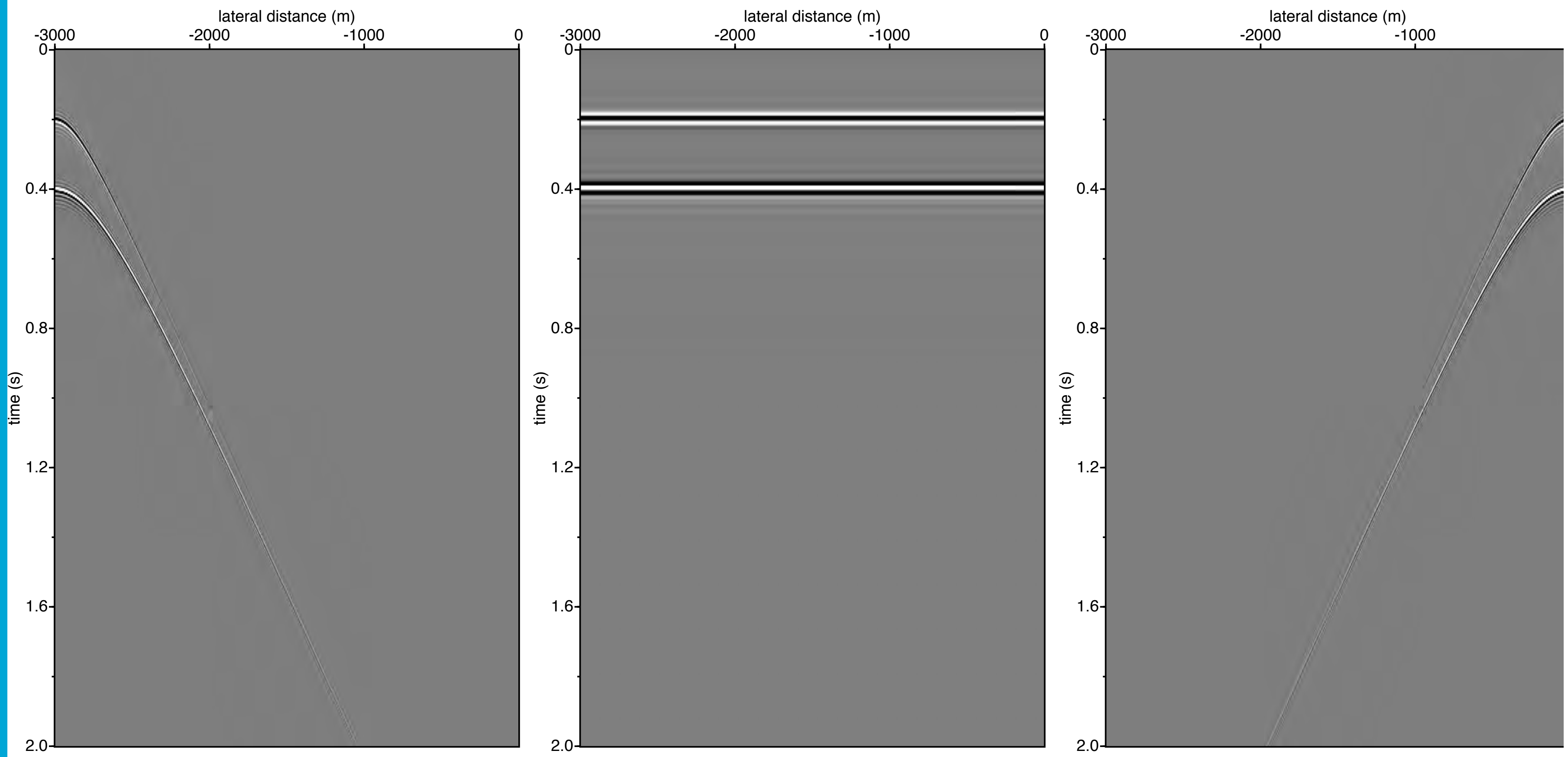
Reference: Lele Zhang, Jan Thorbecke, Kees Wapenaar, Evert Slob: *Data-driven internal multiple elimination and its consequences for imaging: A comparison of strategies*, accepted Geophysics

shotsdx4_rp.su



shot 1 shot 375 shot 751

shotsdx4_rp.su

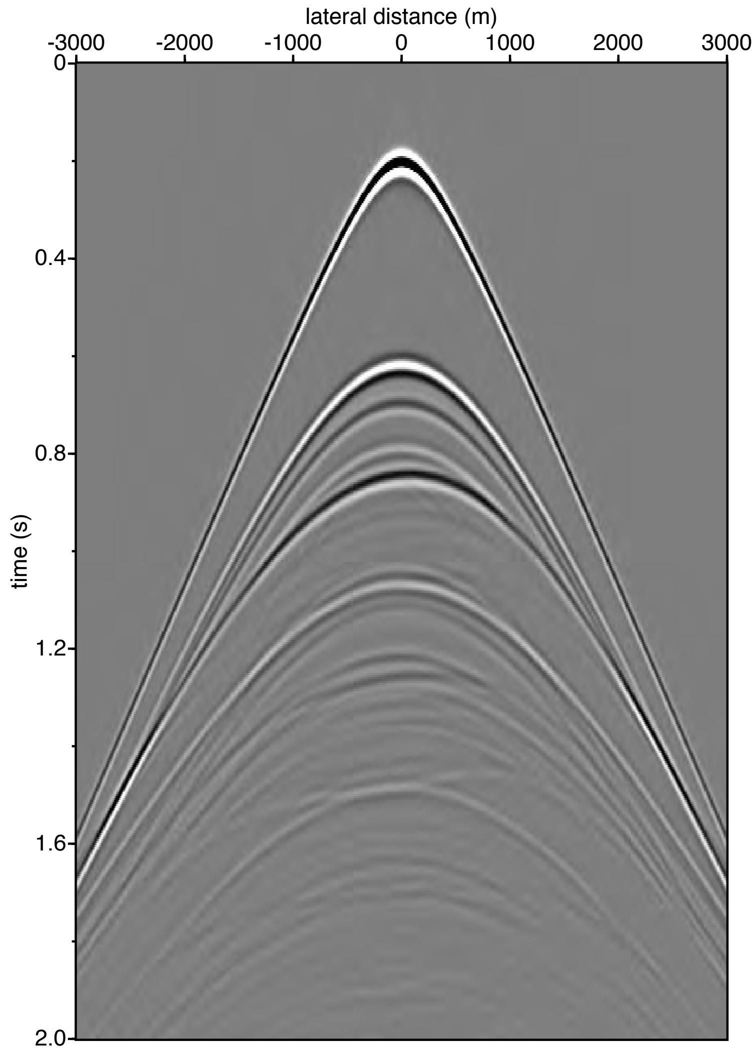


shot 1

zow

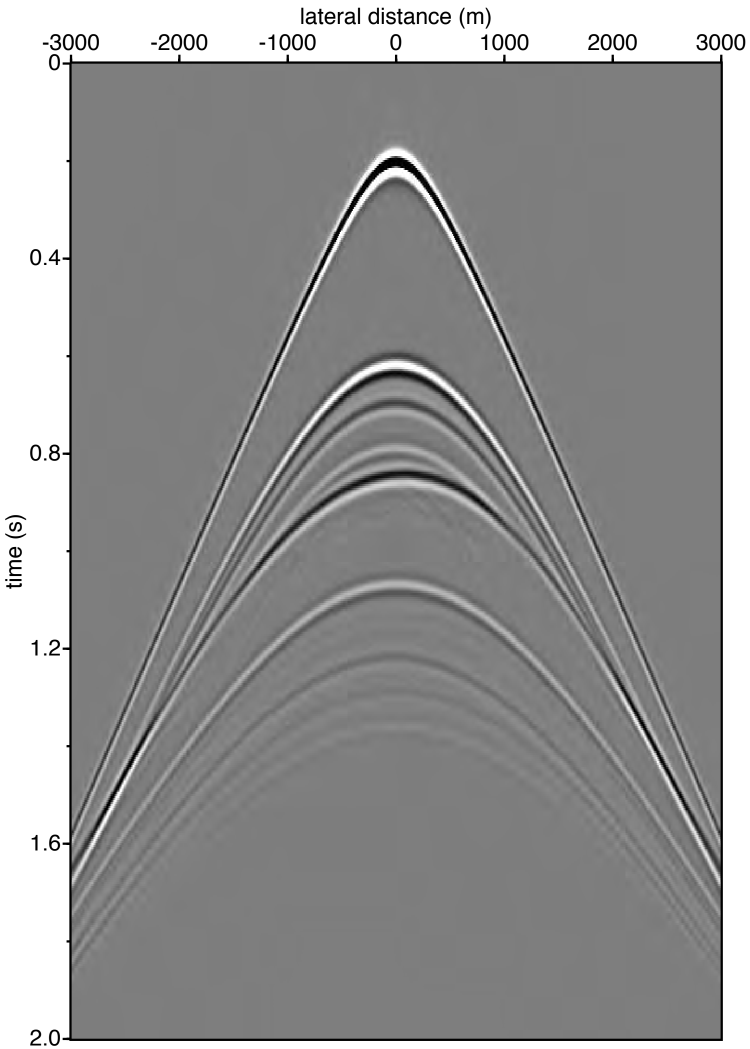
shot 751

twoD figures



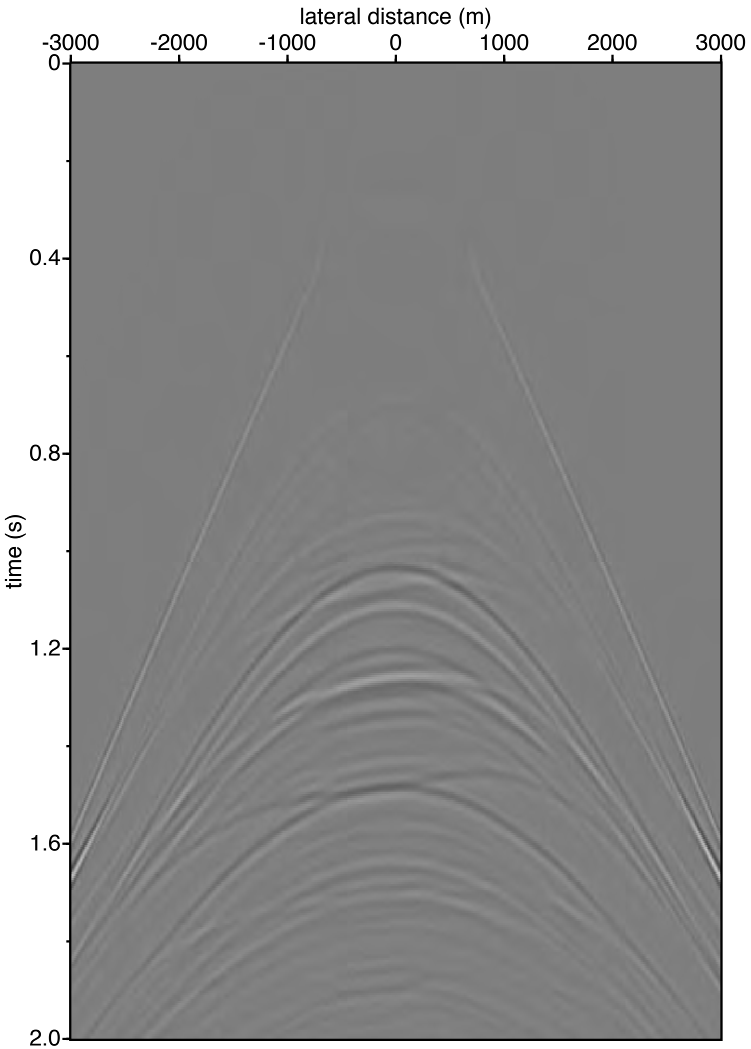
original shot

twoD figures



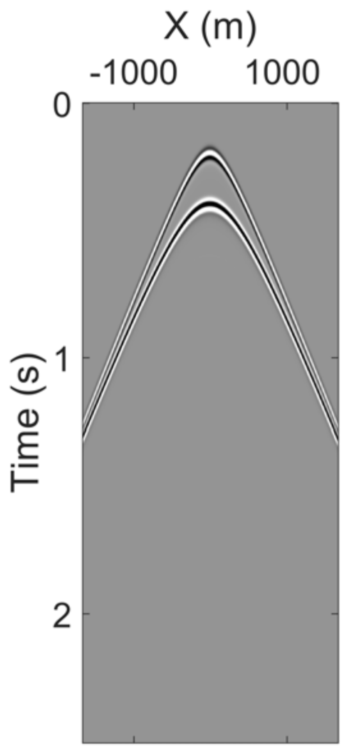
primaries

twoD figures

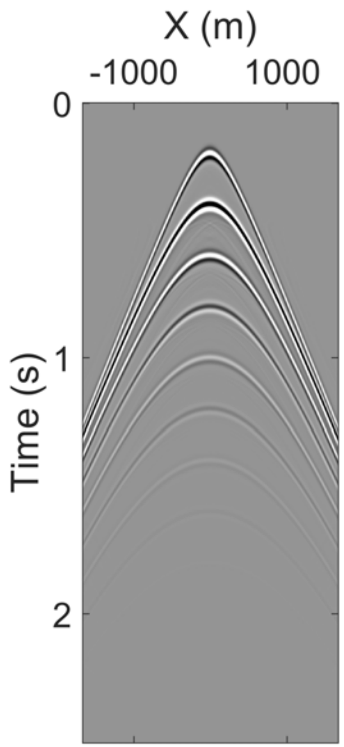


removed multiples

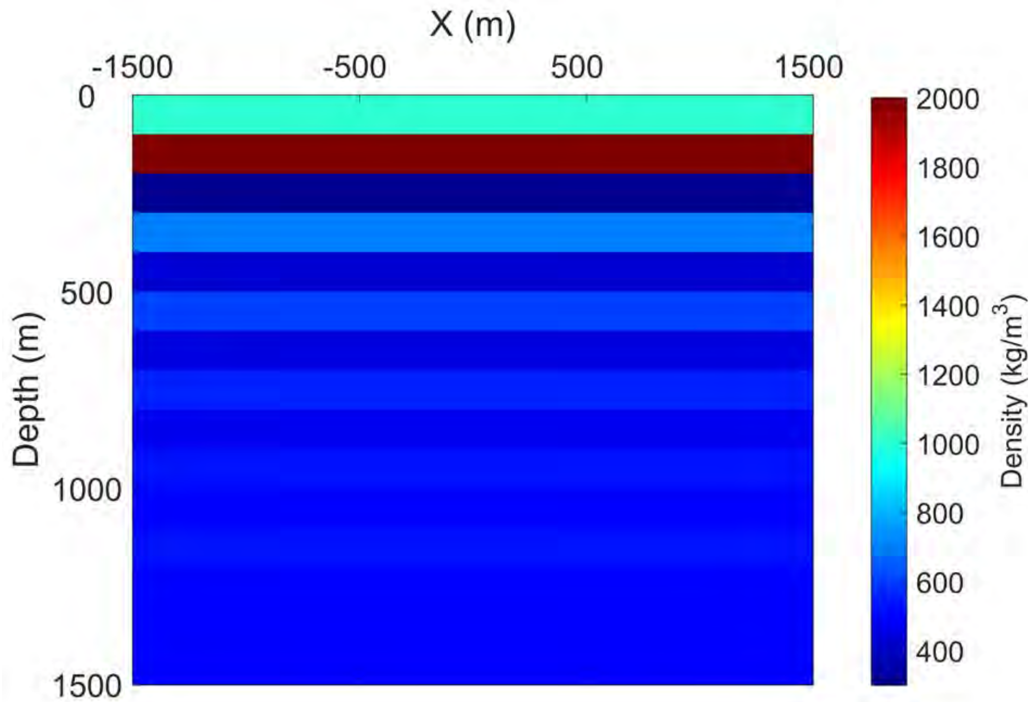
Invisible medium



Invisible medium



Invisible medium

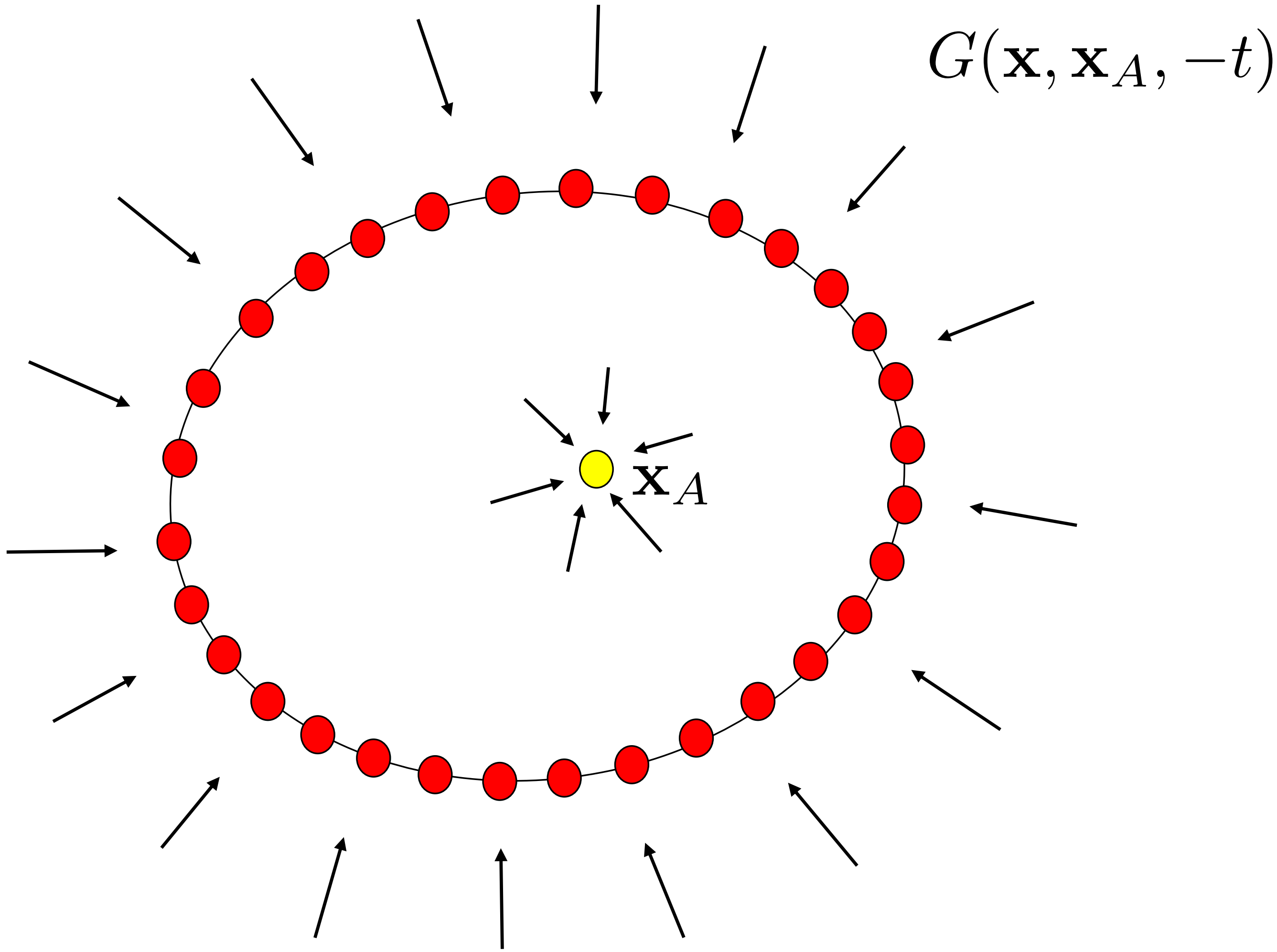


CRAY

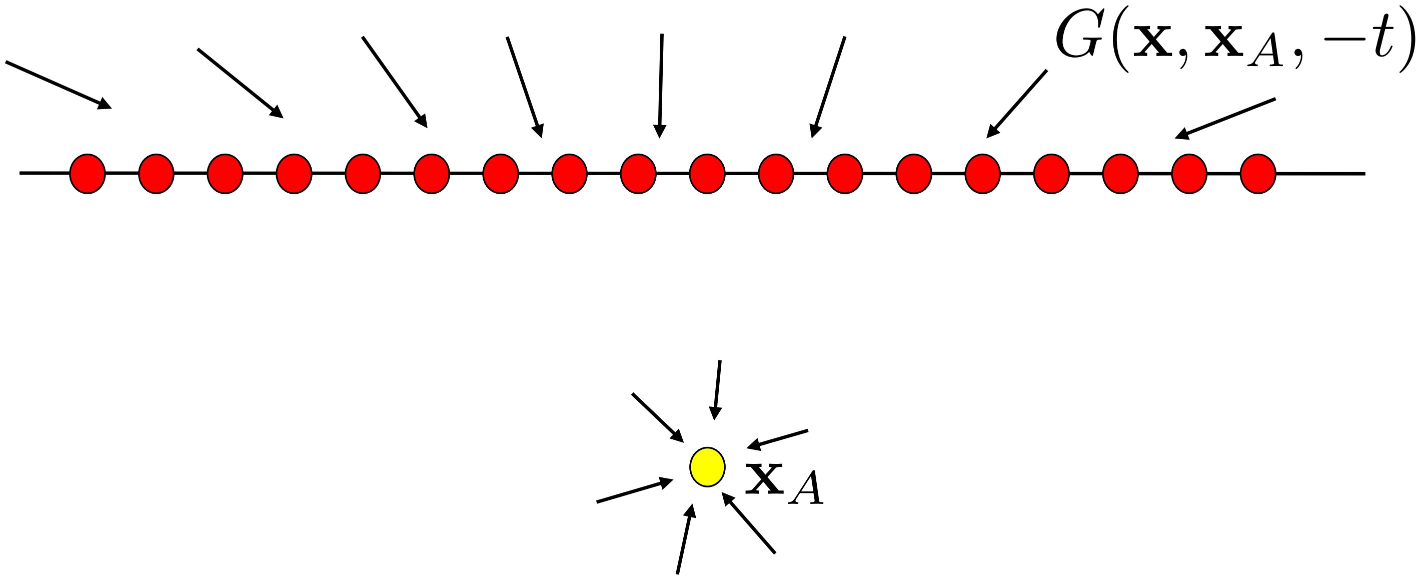
TUDelft

Homogeneous Green's function

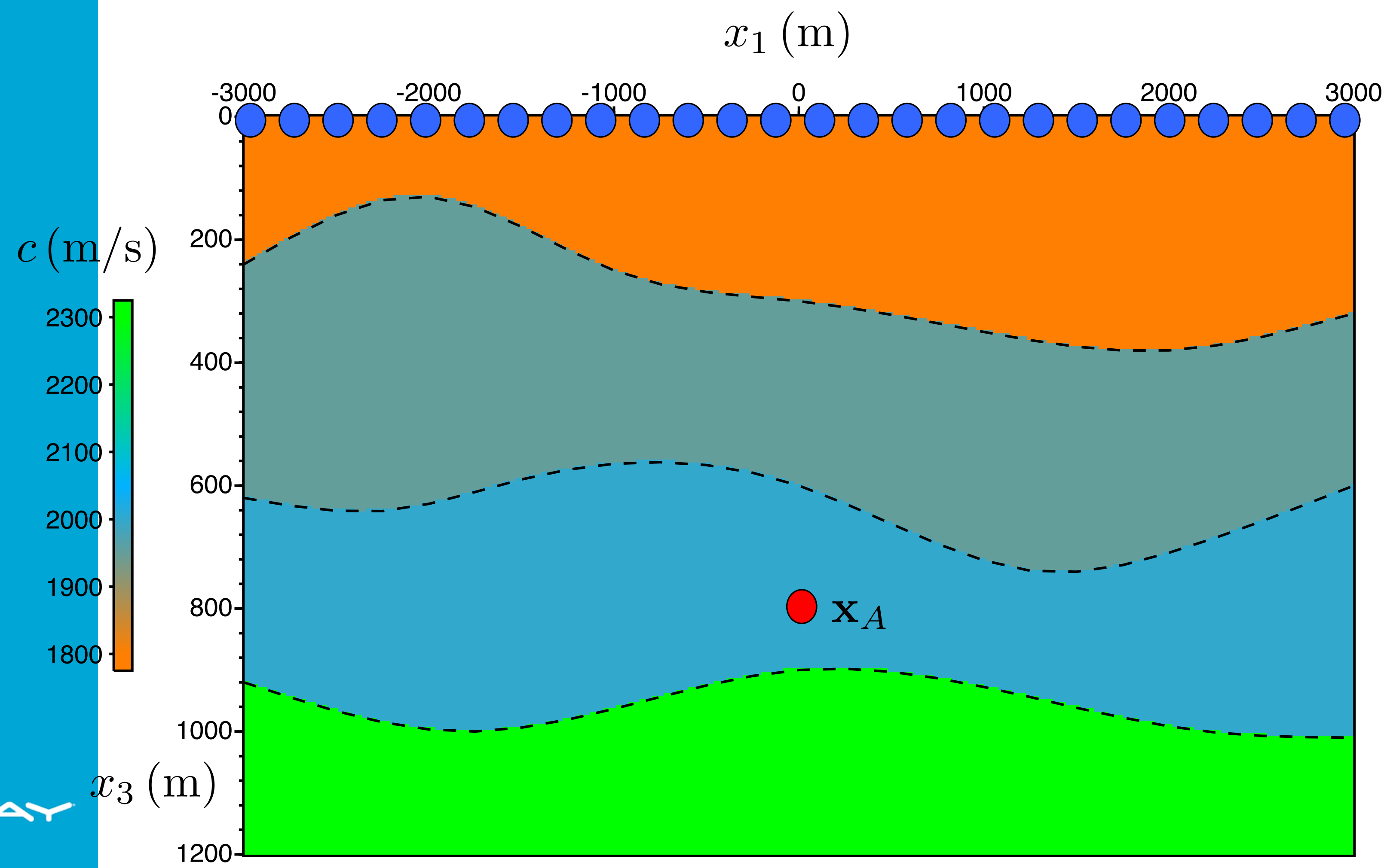
Reference: Brackenhoff, J., Thorbecke, J., and Wapenaar, K.: *Monitoring induced distributed double-couple sources using Marchenko-based virtual receivers*, Solid Earth Discuss., https://doi.org/10.5194/se-2018-142, in review, 2019.



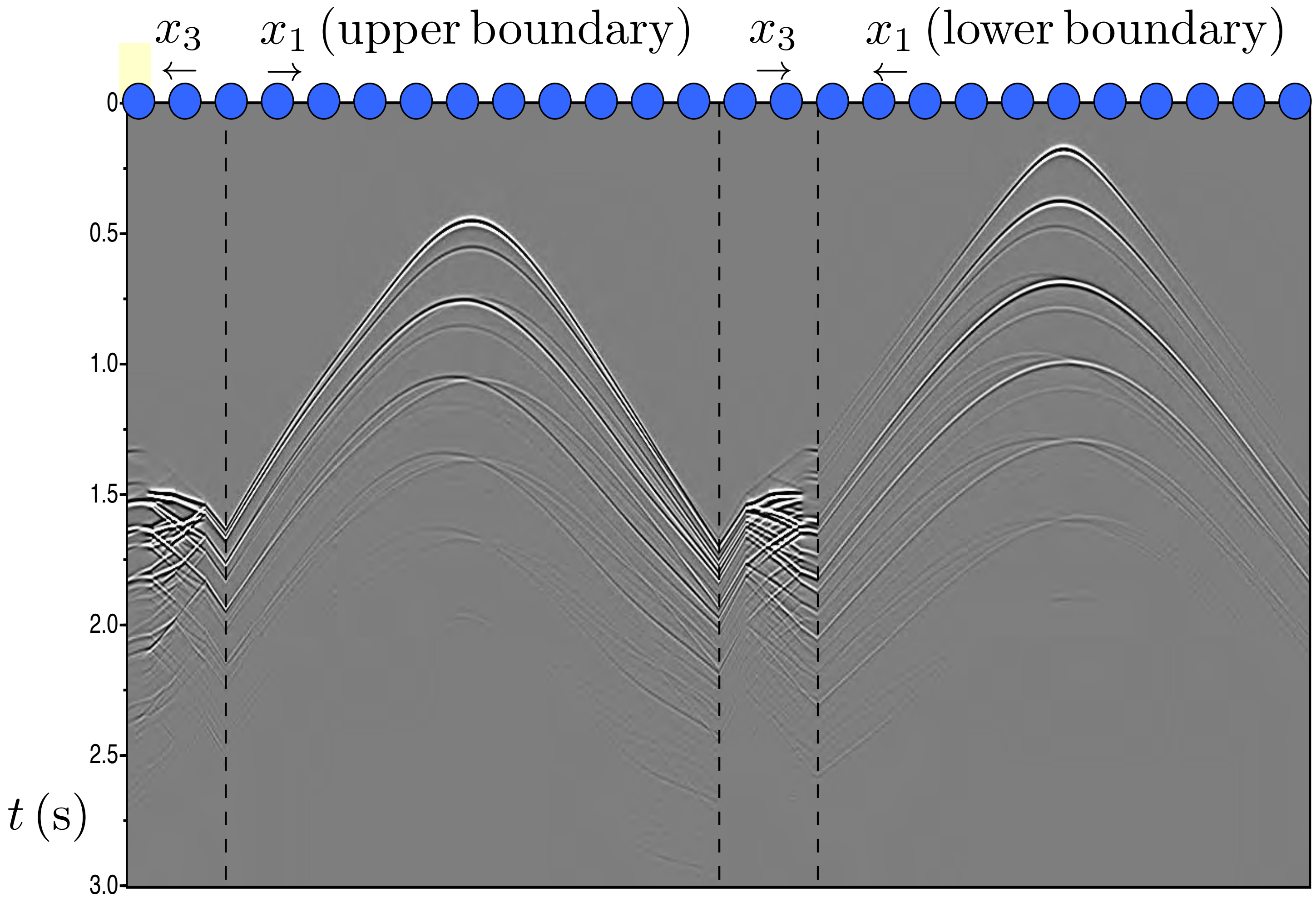
$$G(\mathbf{x}_B, \mathbf{x}_A, t) + G(\mathbf{x}_B, \mathbf{x}_A, -t) \propto \oint_{\partial \mathbb{D}} G(\mathbf{x}_B, \mathbf{x}, t) * G(\mathbf{x}, \mathbf{x}_A, -t) \mathrm{d}^2\mathbf{x}$$



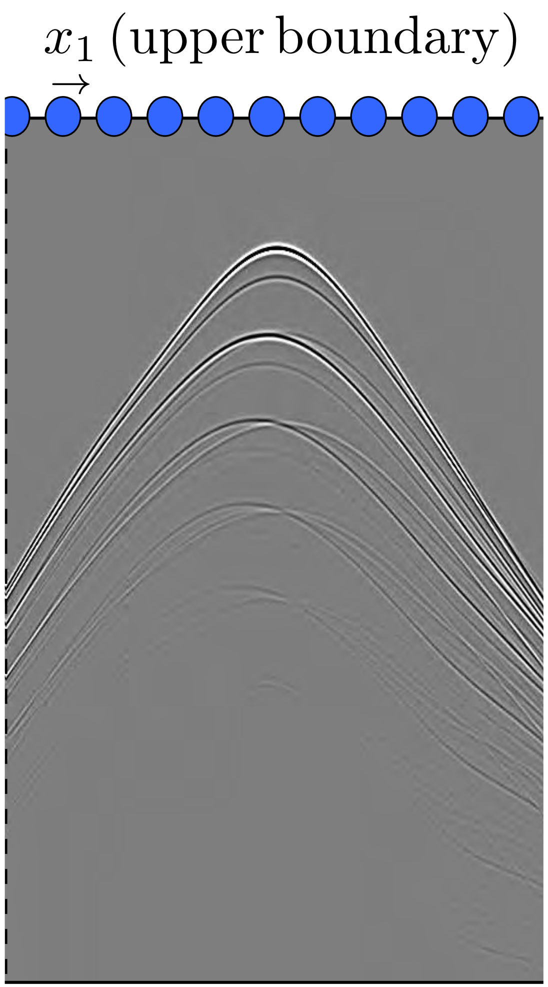
$$G(\mathbf{x}_B, \mathbf{x}_A, t) + G(\mathbf{x}_B, \mathbf{x}_A, -t) \neq \int_{\partial \mathbb{D}} G(\mathbf{x}_B, \mathbf{x}, t) * G(\mathbf{x}, \mathbf{x}_A, -t) \mathrm{d}^2\mathbf{x}$$



Single-sided time-reversal experiment

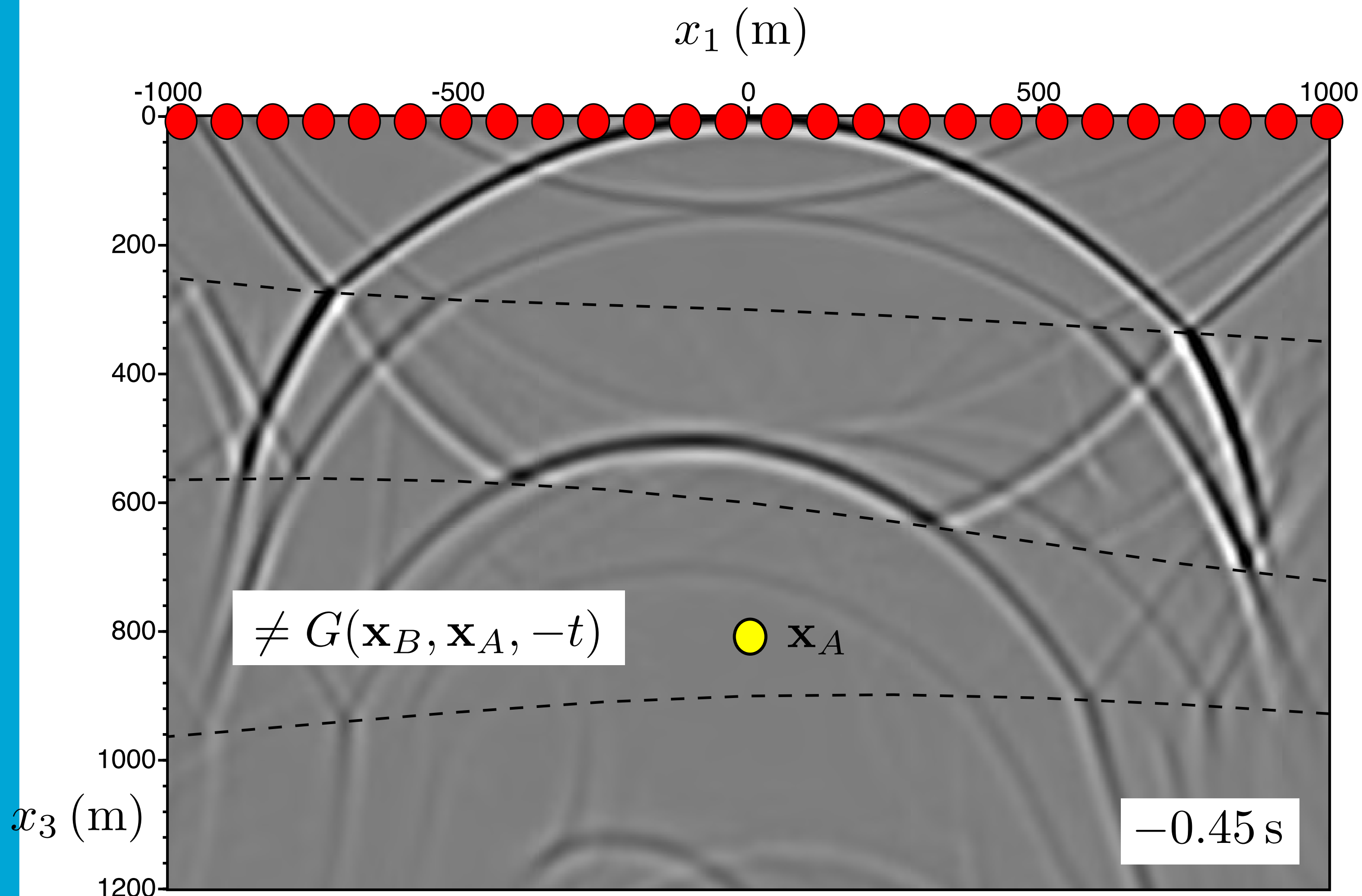


Single-sided time-reversal experiment

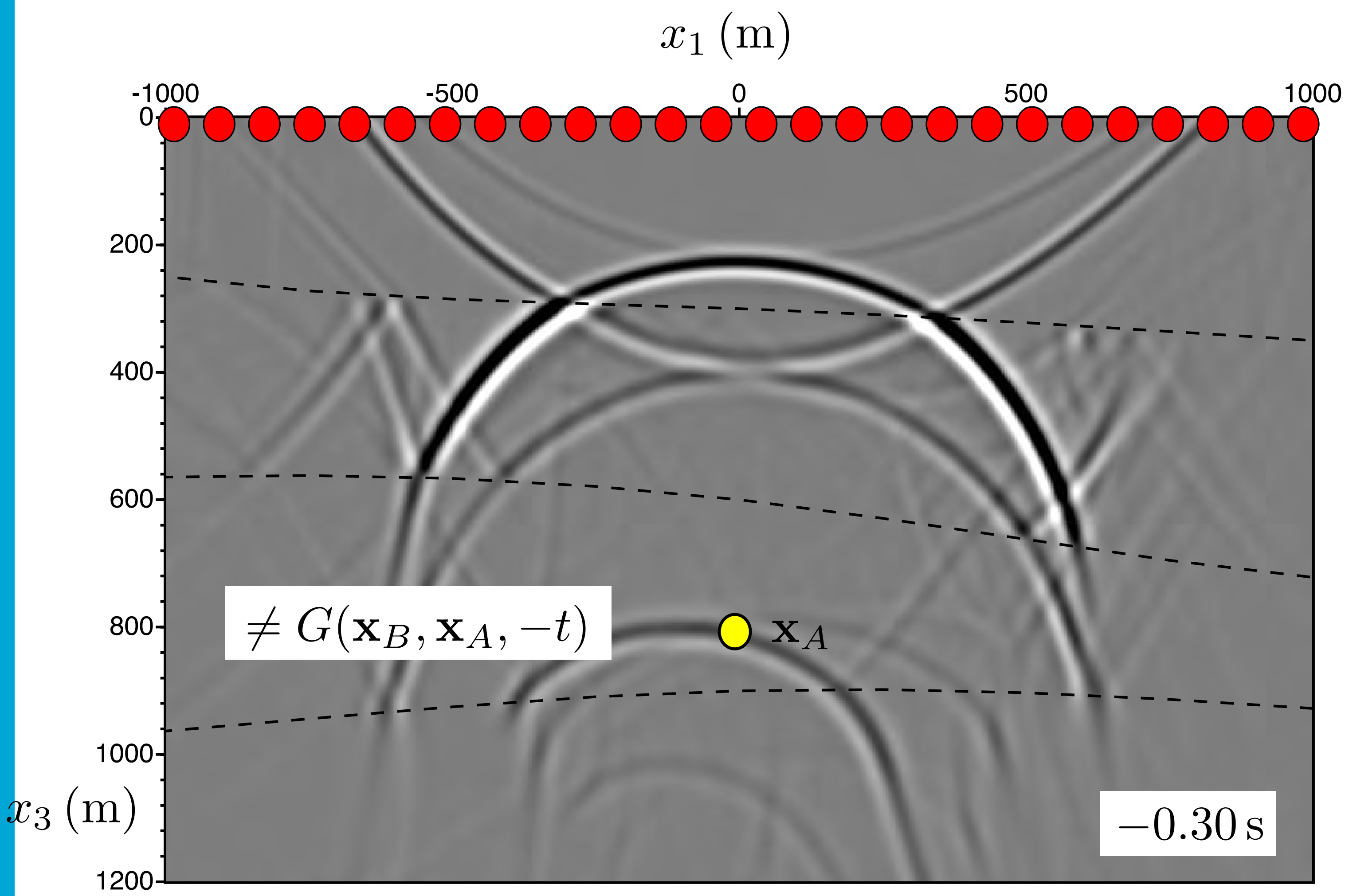


Single-sided time-reversal experiment

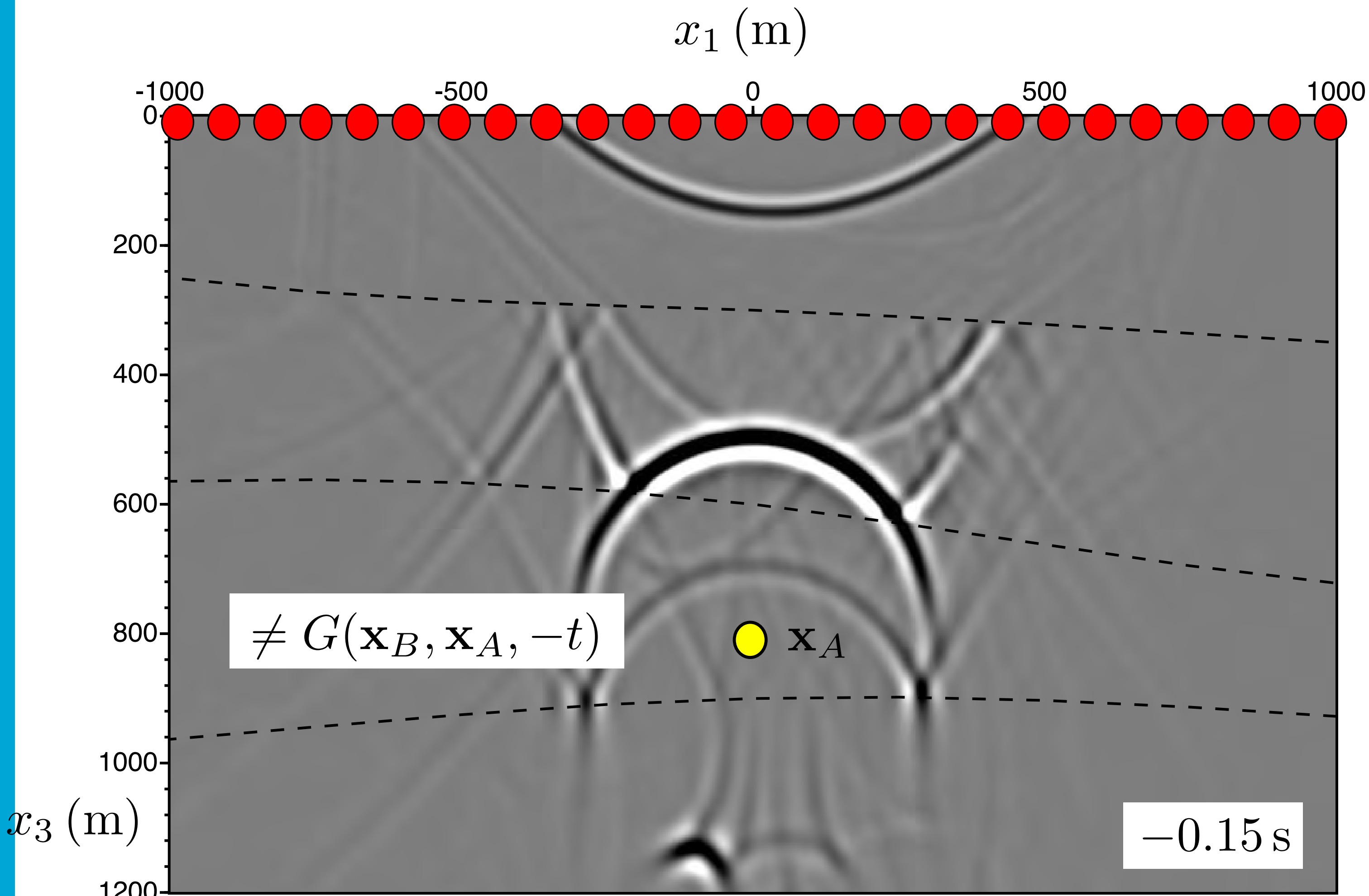
Single-sided time-reversal experiment



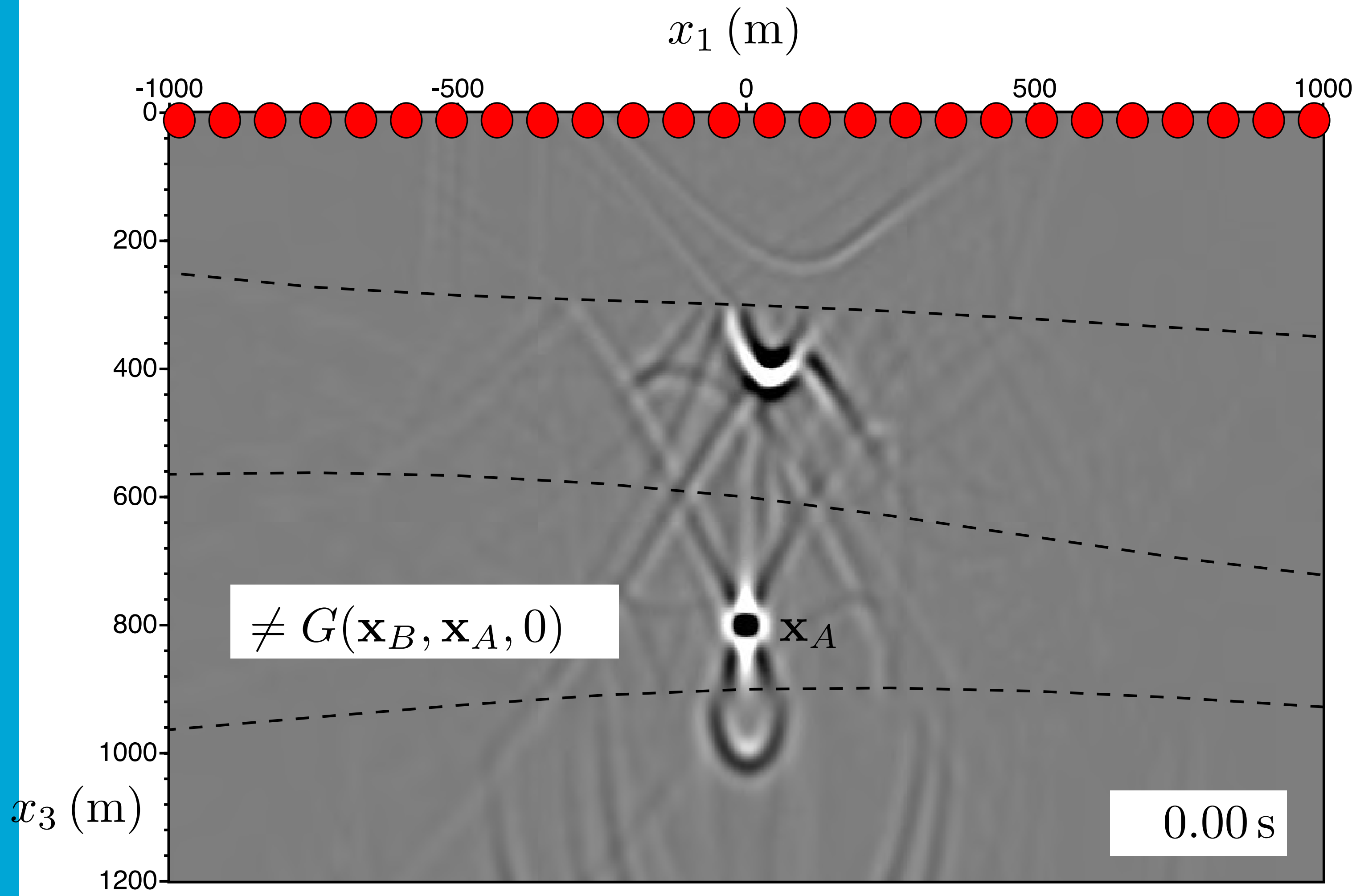
Single-sided time-reversal experiment



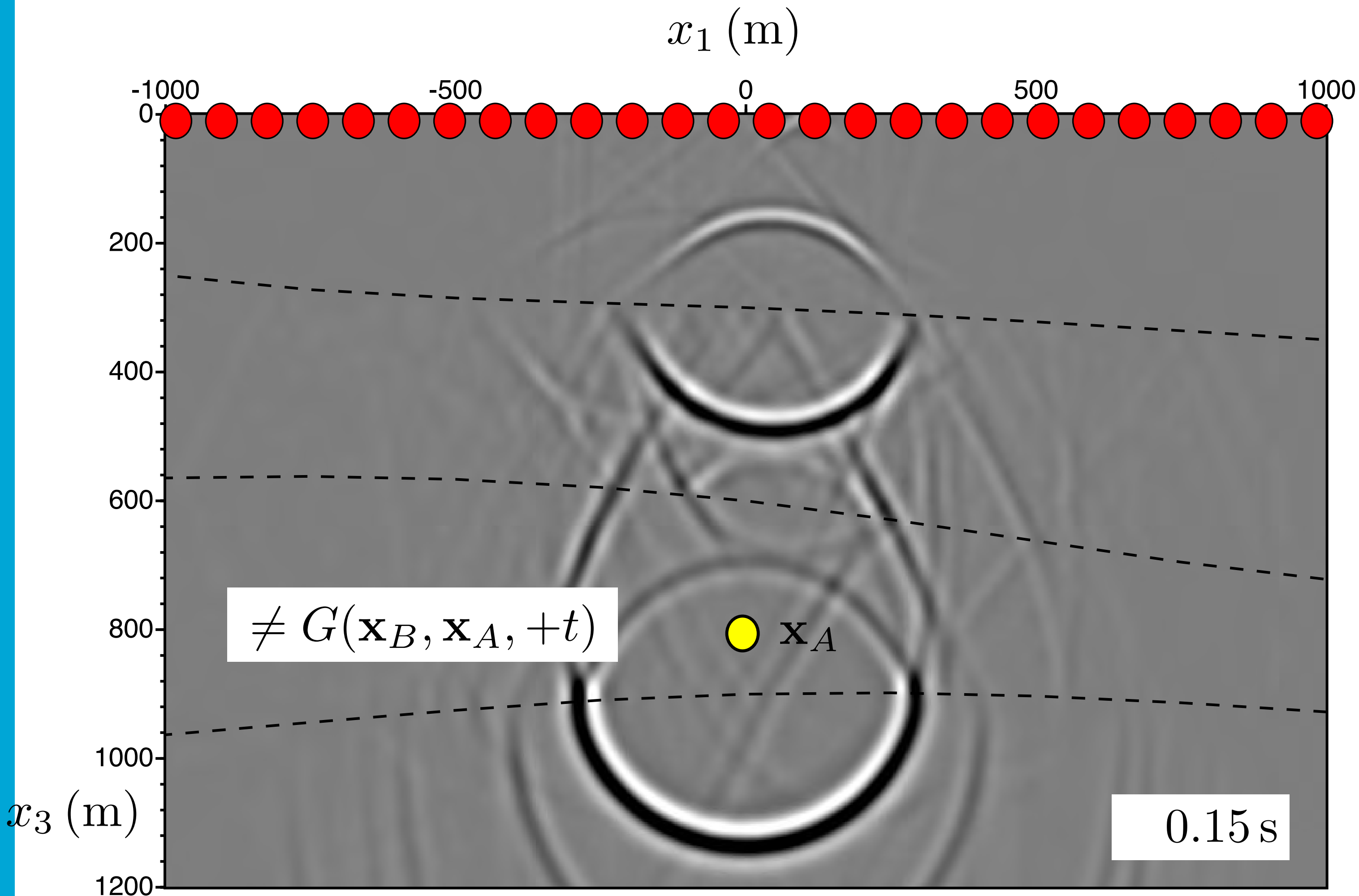
Single-sided time-reversal experiment



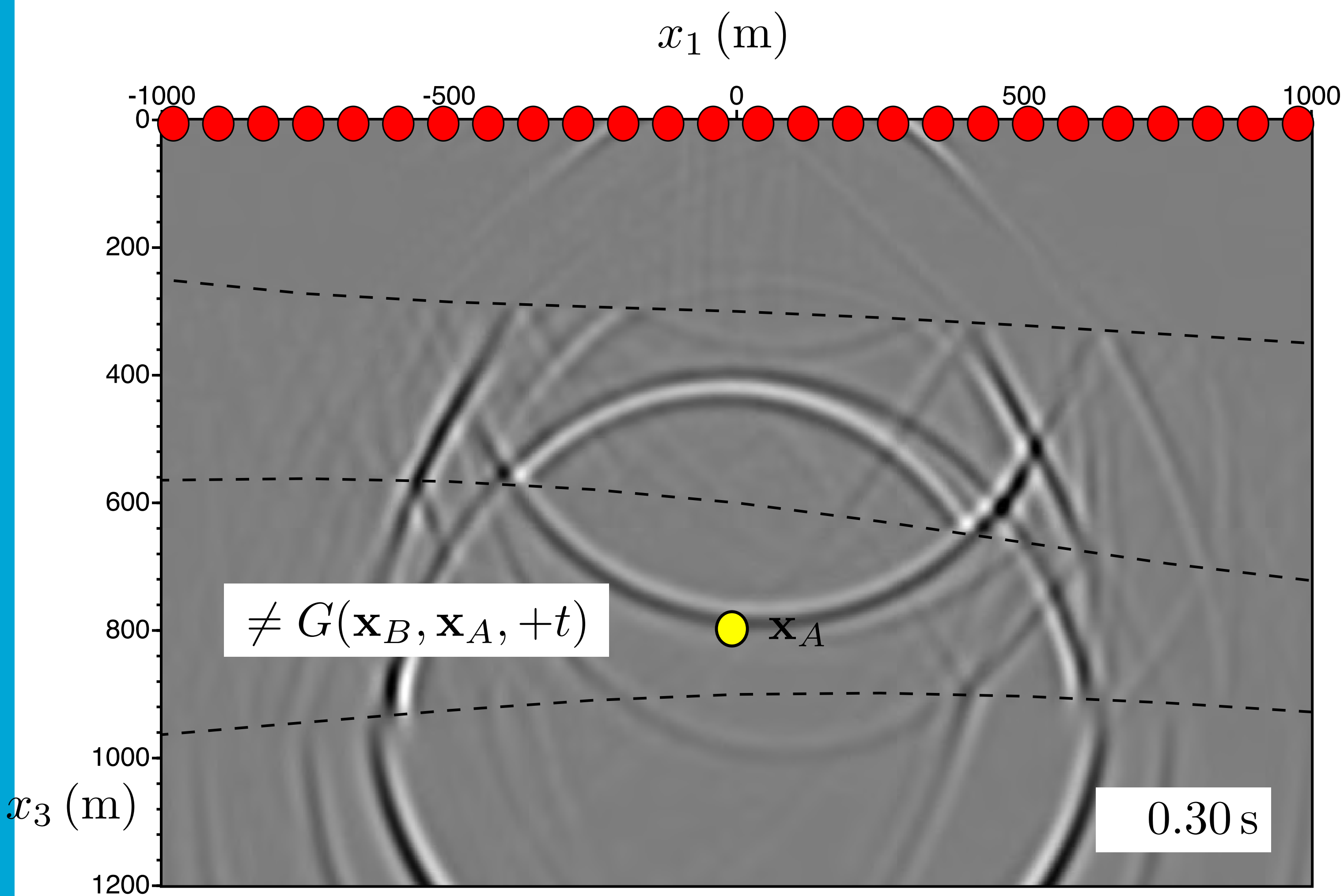
Single-sided time-reversal experiment



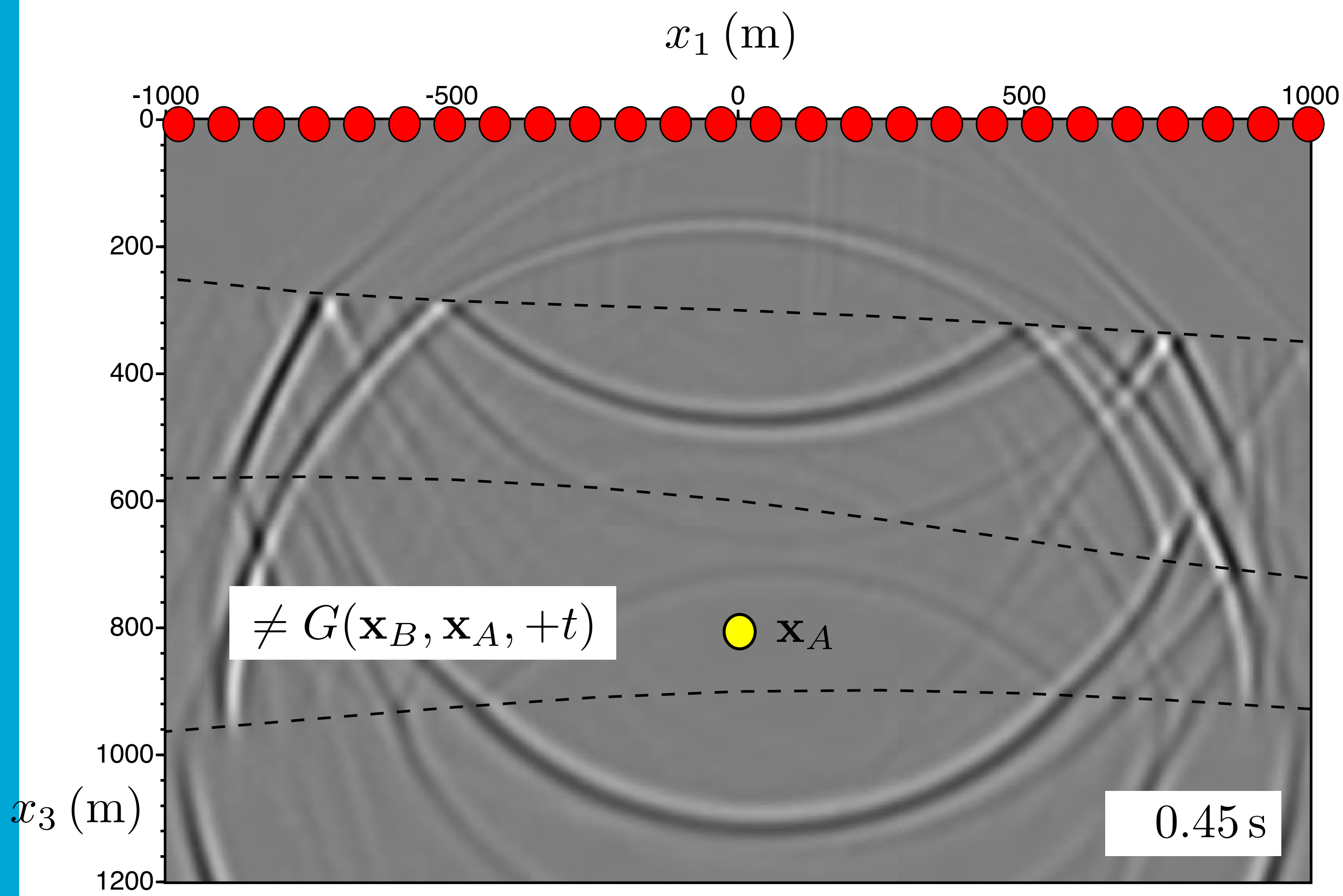
Single-sided time-reversal experiment



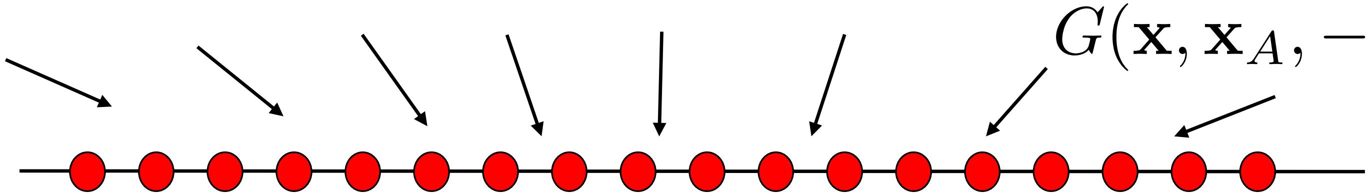
Single-sided time-reversal experiment

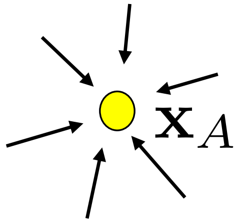


Single-sided time-reversal experiment

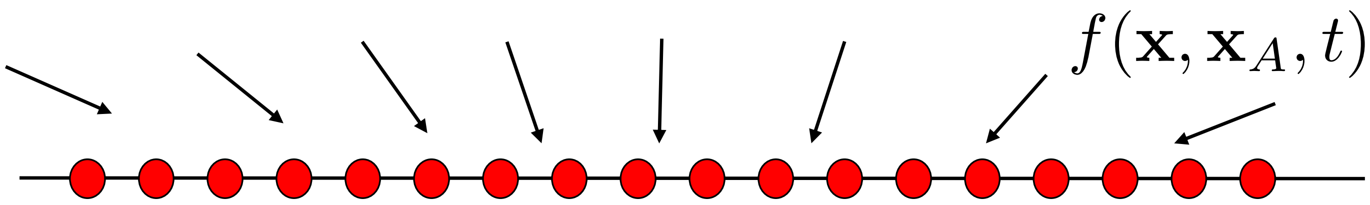


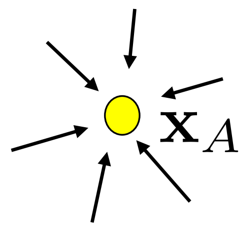
Single-sided time-reversal experiment





$$G(\mathbf{x}_B, \mathbf{x}_A, t) + G(\mathbf{x}_B, \mathbf{x}_A, -t) \neq \int_{\partial \mathbb{D}} G(\mathbf{x}_B, \mathbf{x}, t) * G(\mathbf{x}, \mathbf{x}_A, -t) \mathrm{d}^2\mathbf{x}$$

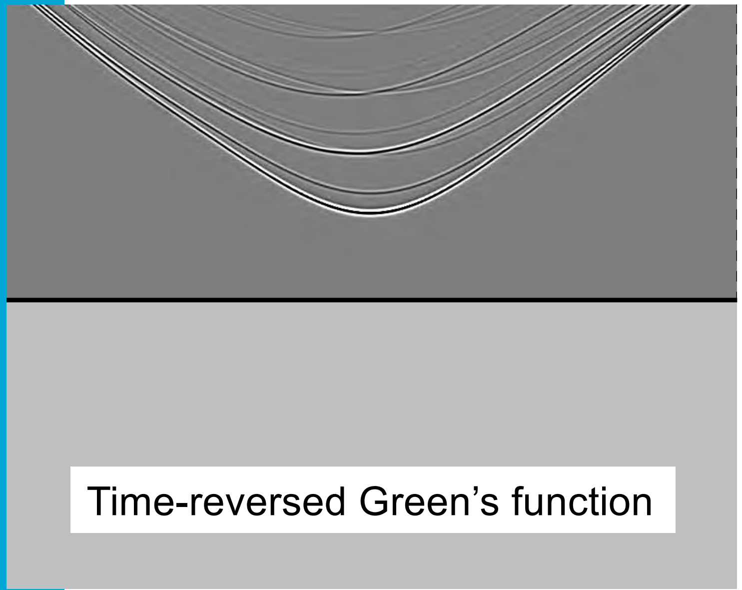


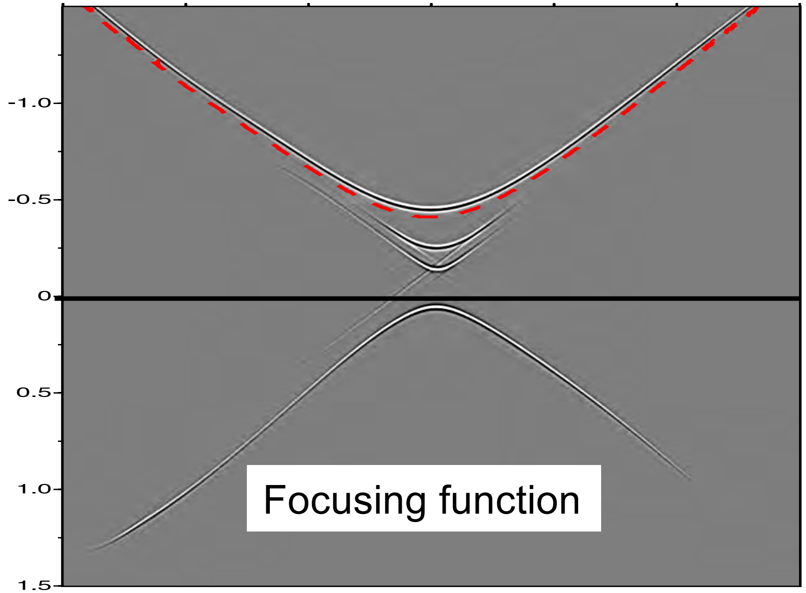


$$G(\mathbf{x}_B, \mathbf{x}_A, t) + G(\mathbf{x}_B, \mathbf{x}_A, -t) \propto \int_{\partial \mathbb{D}} G(\mathbf{x}_B, \mathbf{x}, t) * f(\mathbf{x}, \mathbf{x}_A, t) \mathrm{d}^2\mathbf{x}$$

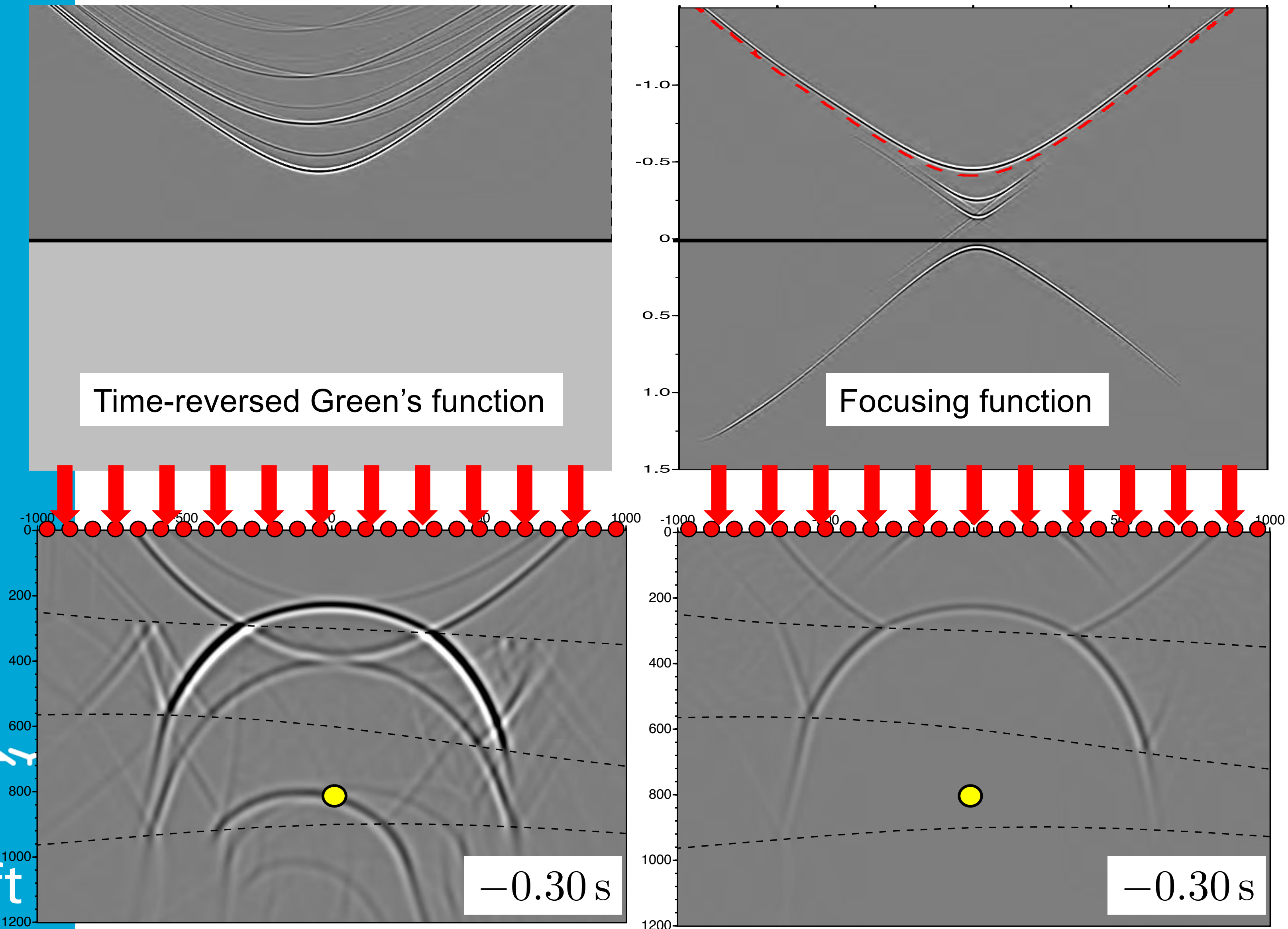
$$+ \int_{\partial \mathbb{D}} G(\mathbf{x}_B, \mathbf{x}, -t) * f(\mathbf{x}, \mathbf{x}_A, -t) \mathrm{d}^2\mathbf{x}$$

References: Wapenaar et al., *Green's function retrieval from reflection data*, JASA 2014; *Virtual sources and their responses*, Geophys. Prosp. 2017

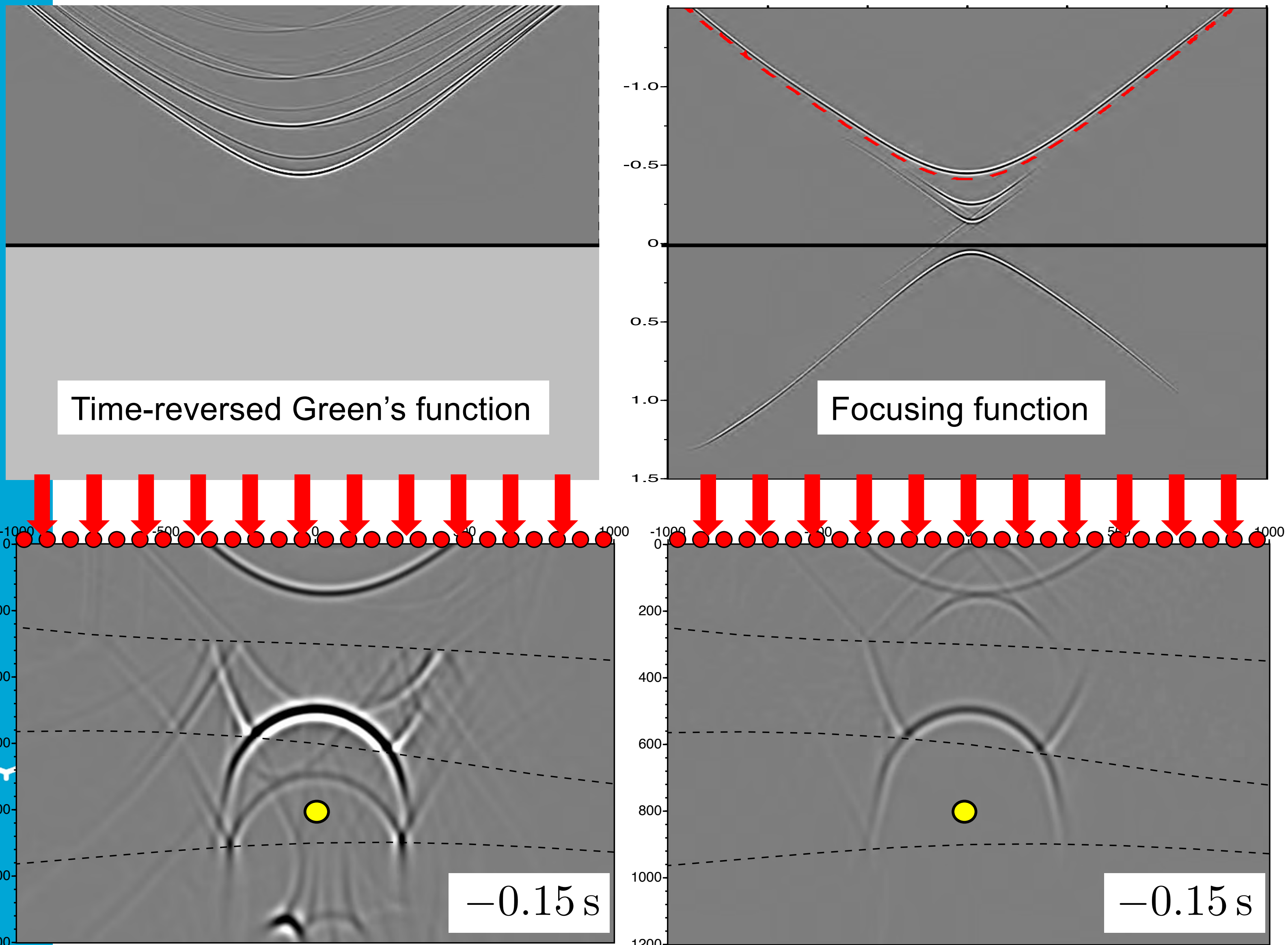


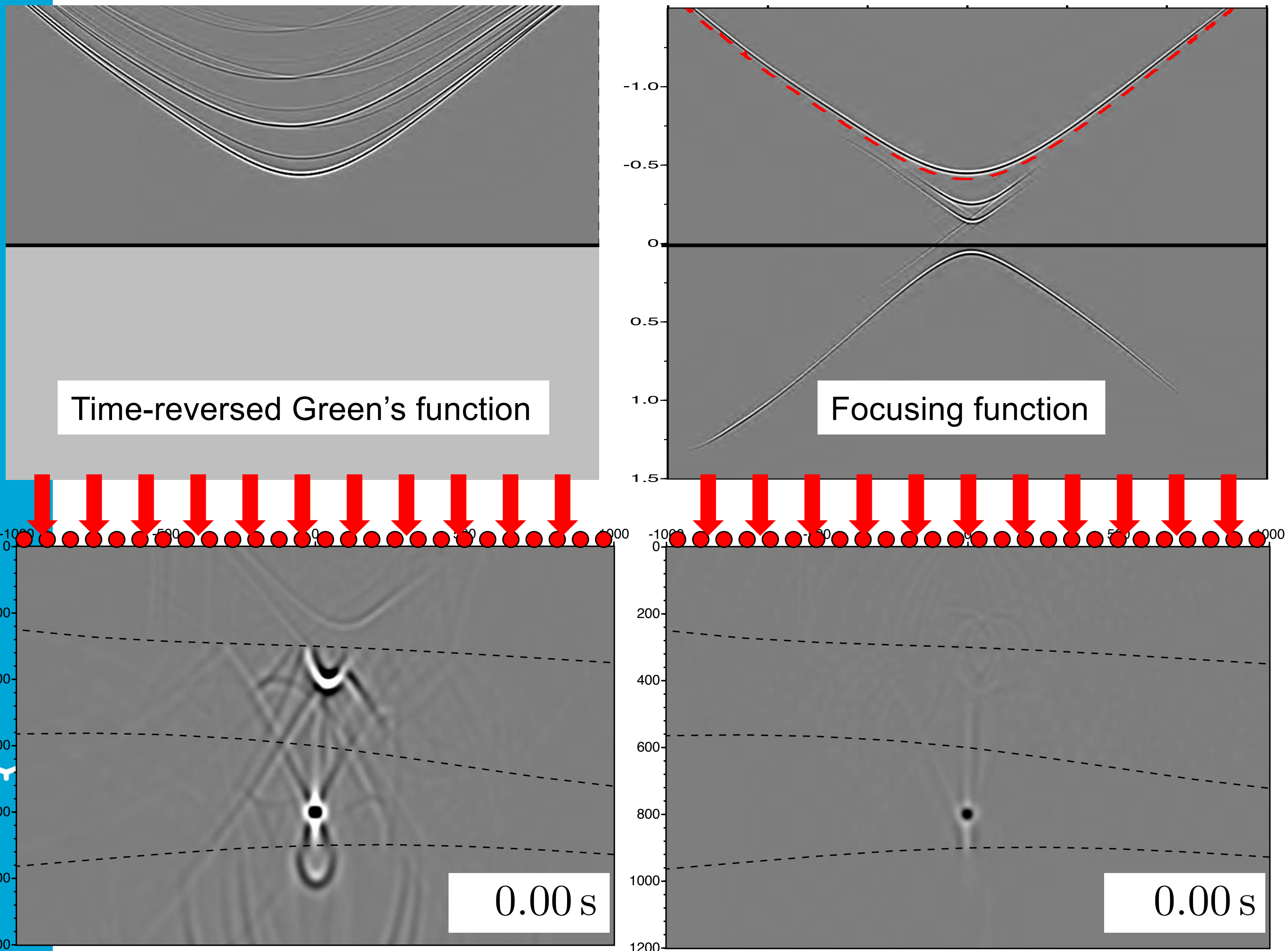


$$G(\mathbf{x}_B, \mathbf{x}_A, t) + G(\mathbf{x}_B, \mathbf{x}_A, -t) \propto \int_{\partial \mathbb{D}} G(\mathbf{x}_B, \mathbf{x}, t) * f(\mathbf{x}, \mathbf{x}_A, t) \mathrm{d}^2\mathbf{x}$$

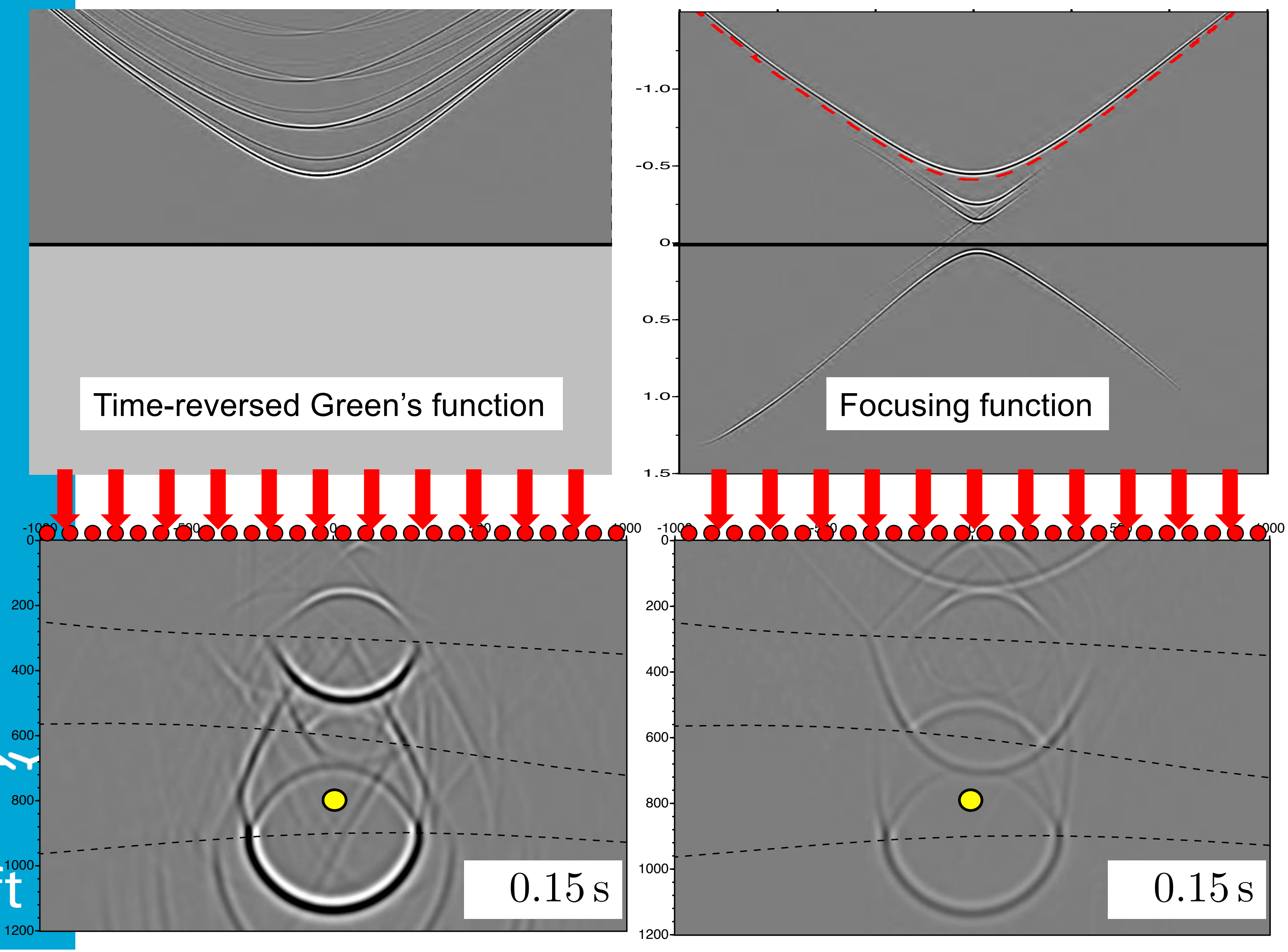


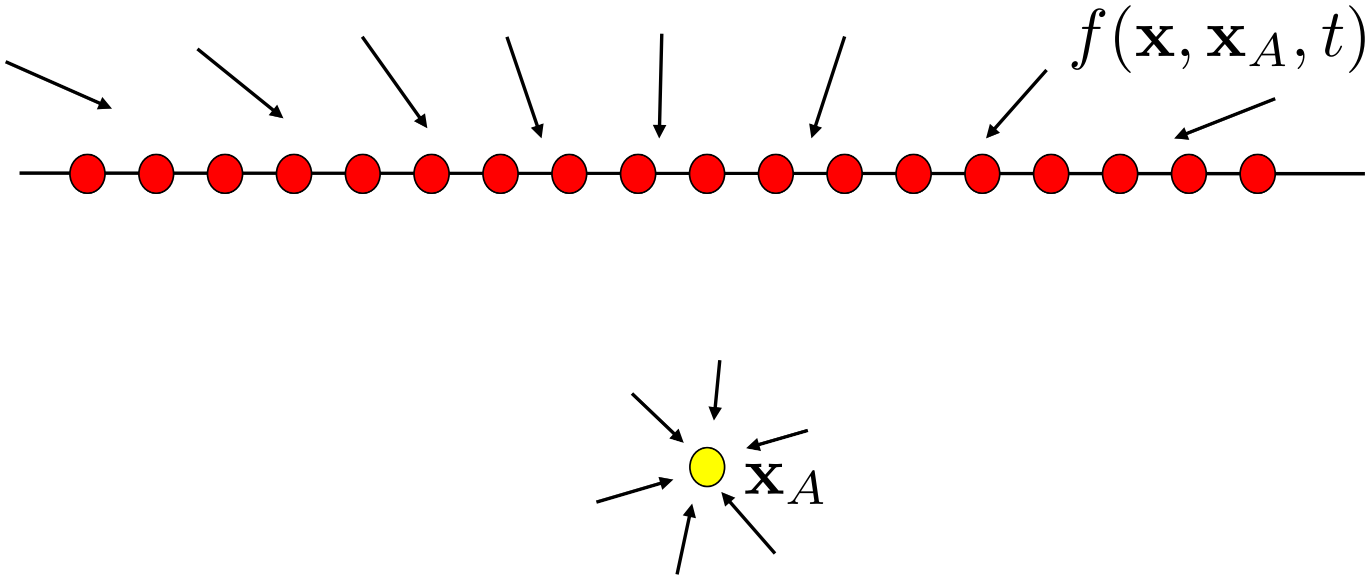
Time-reversed Green's function
Focusing function
-1.0
-0.5
0
0.5
1.0
1.5
-1000
1000
0
200
400
600
800
1000
1200
−0.30 s
−0.30 s


Time-reversed Green's function
Focusing function
−0.15 s
−0.15 s



Time-reversed Green's function
Focusing function
-1.0
-0.5
0
0.5
1.0
1.5
0.00 s
0.00 s
200
400
600
800
1000
1200


Time-reversed Green's function
Focusing function
0.15 s
0.15 s

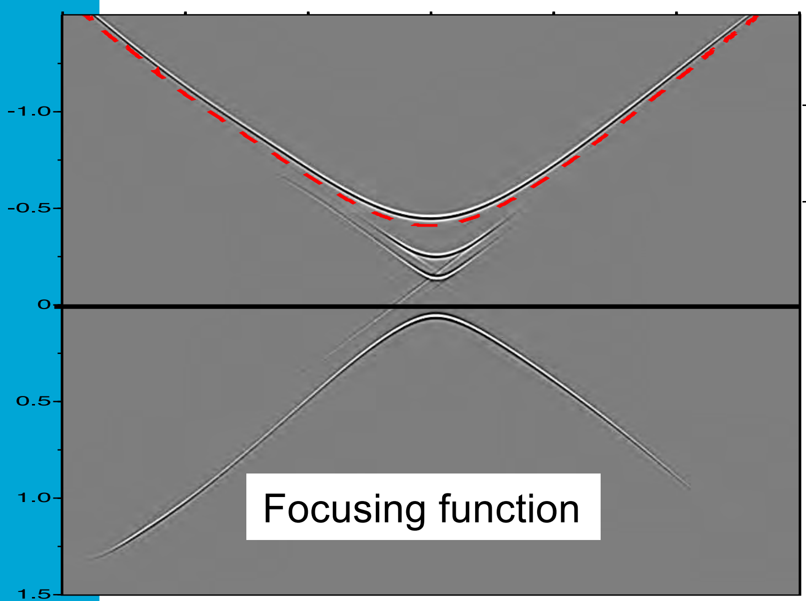


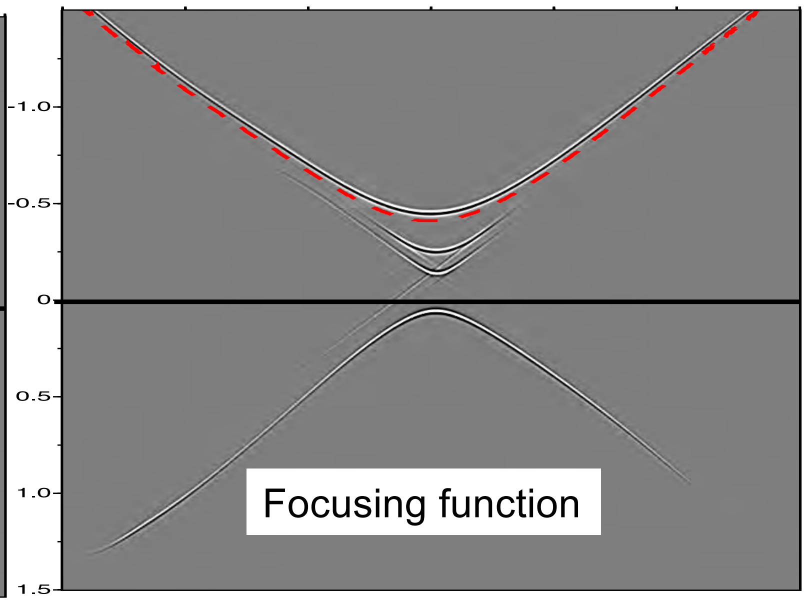
$$G(\mathbf{x}_B, \mathbf{x}_A, t) + G(\mathbf{x}_B, \mathbf{x}_A, -t) \propto \int_{\partial\mathbb{D}} G(\mathbf{x}_B, \mathbf{x}, t) * f(\mathbf{x}, \mathbf{x}_A, t) \mathrm{d}^2\mathbf{x}$$

$$+ \int_{\partial\mathbb{D}} G(\mathbf{x}_B, \mathbf{x}, -t) * f(\mathbf{x}, \mathbf{x}_A, -t) \mathrm{d}^2\mathbf{x}$$

CRAY

TU Delft



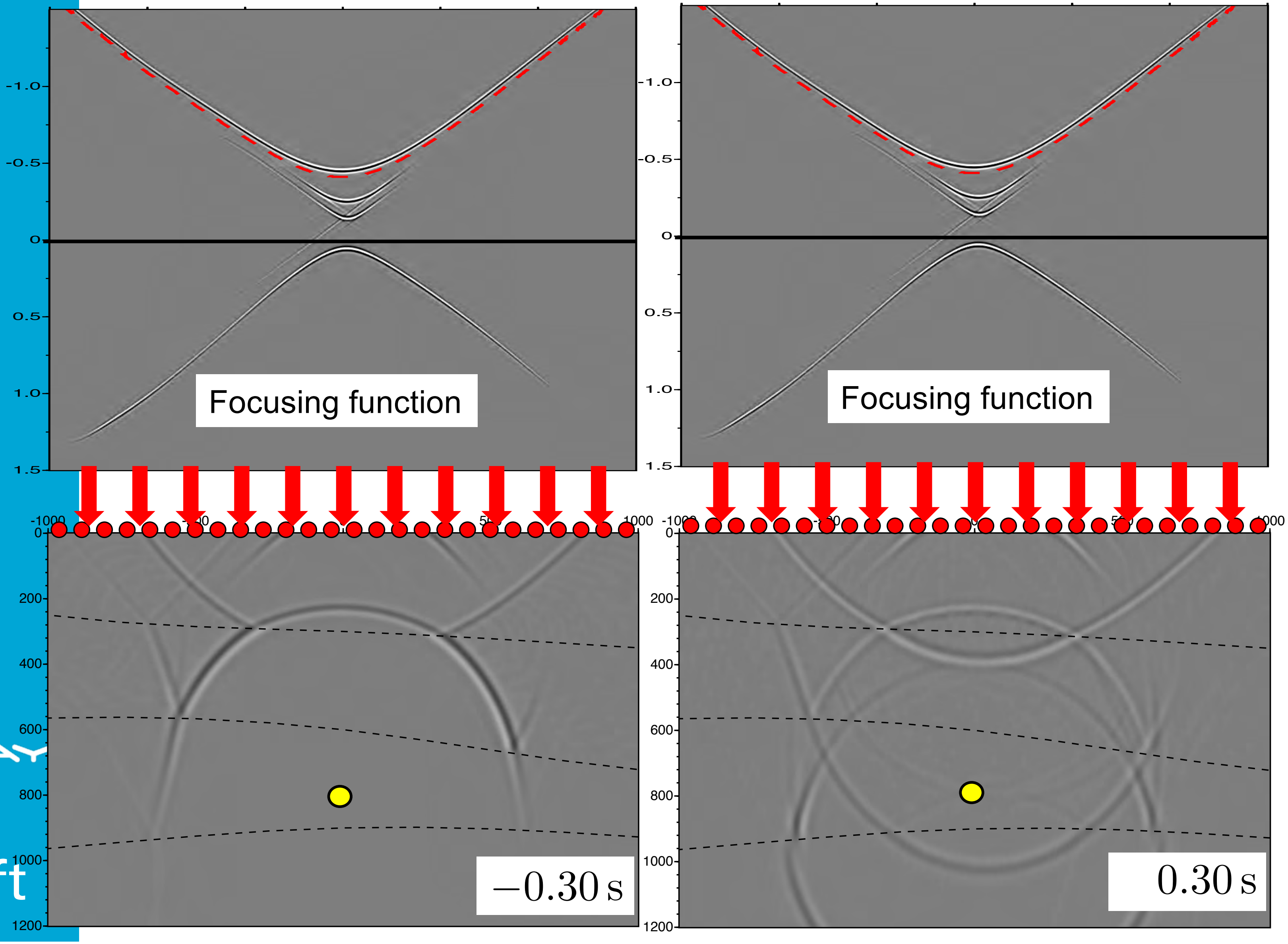


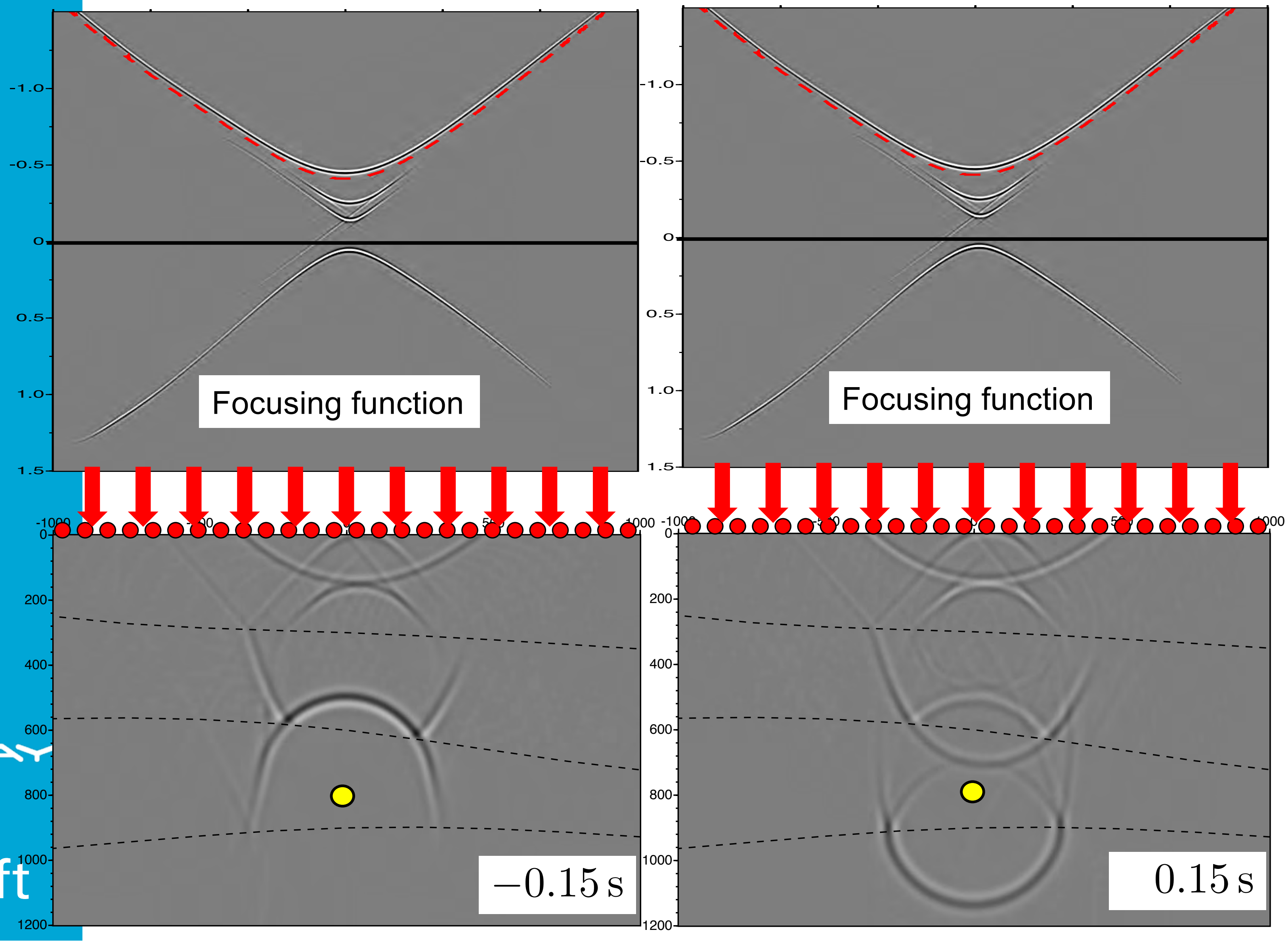
$$G(\mathbf{x}_B, \mathbf{x}_A, t) + G(\mathbf{x}_B, \mathbf{x}_A, -t) \propto \int_{\partial\mathbb{D}} G(\mathbf{x}_B, \mathbf{x}, t) * f(\mathbf{x}, \mathbf{x}_A, t) \mathrm{d}^2\mathbf{x}$$

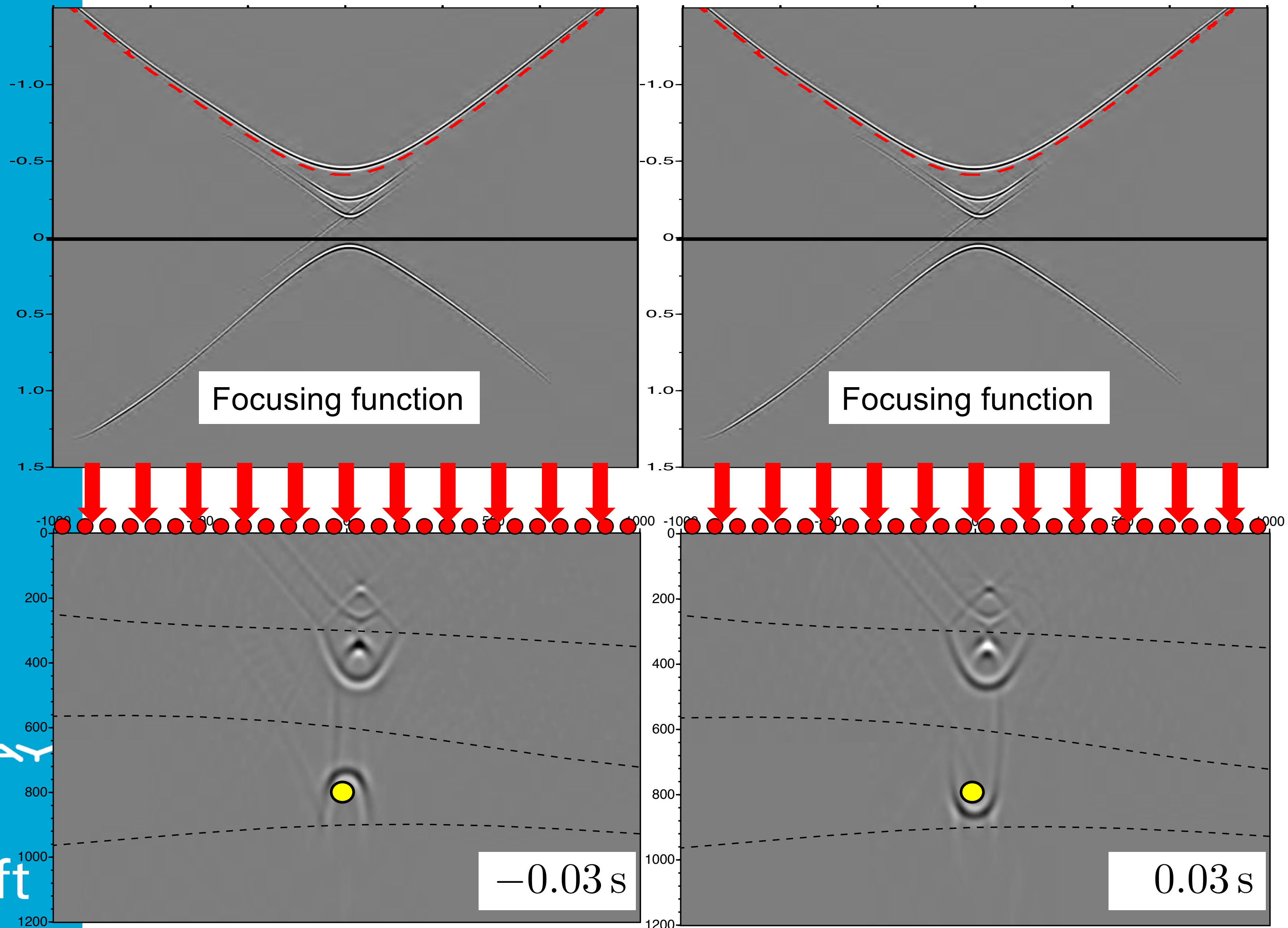
$$+ \int_{\partial\mathbb{D}} G(\mathbf{x}_B, \mathbf{x}, -t) * f(\mathbf{x}, \mathbf{x}_A, -t) \mathrm{d}^2\mathbf{x}$$

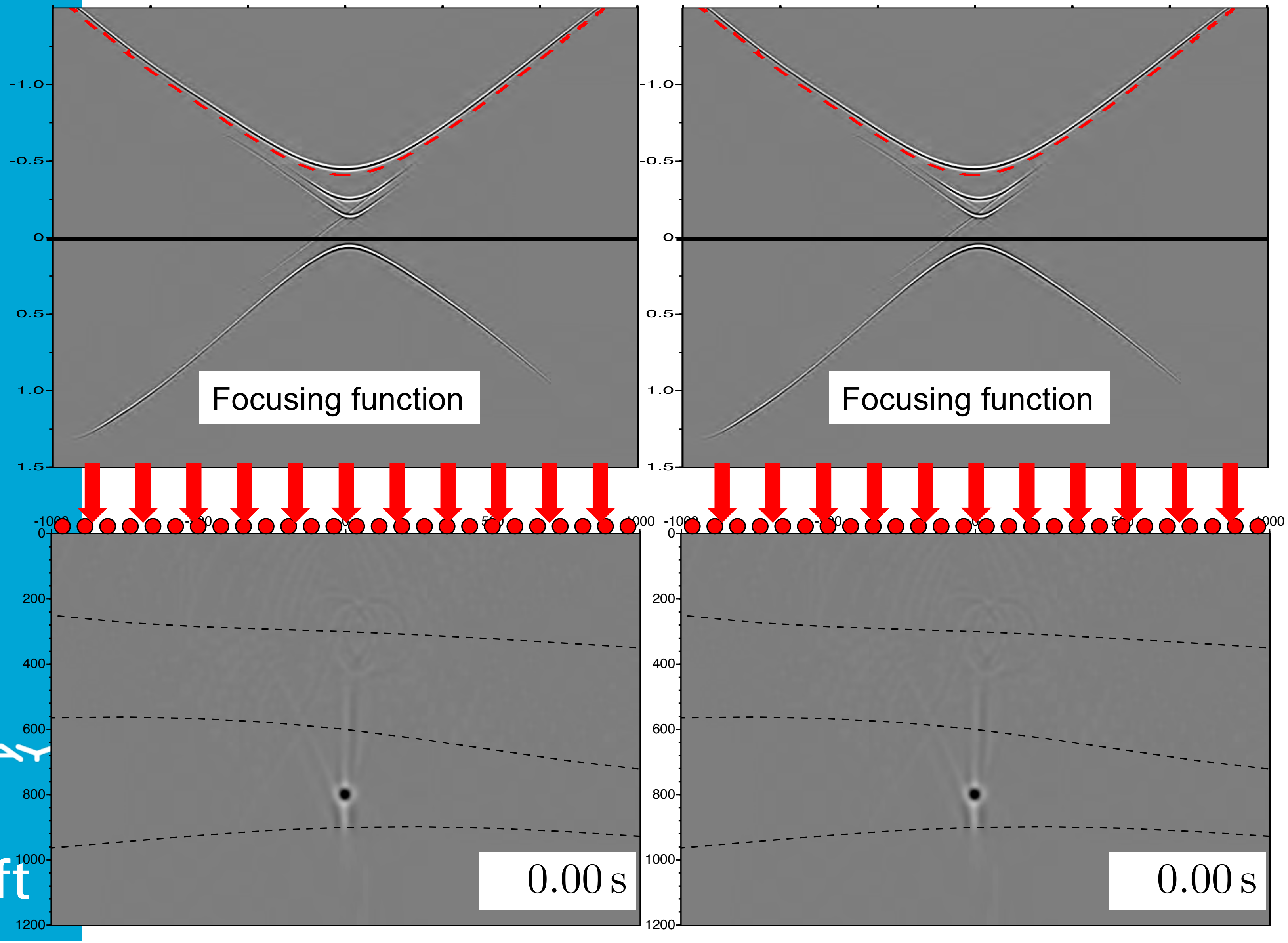
CRAY

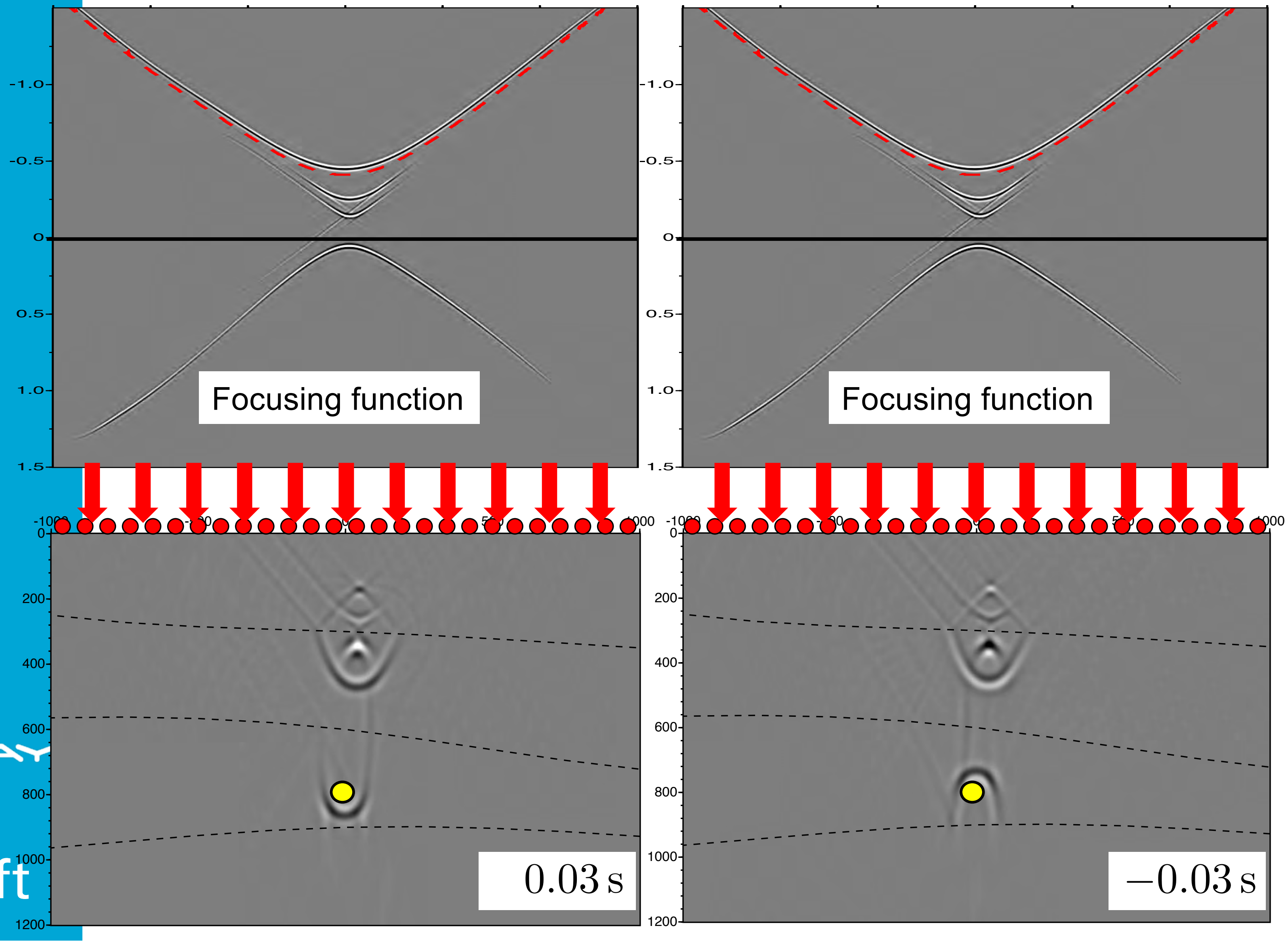
TUDelft

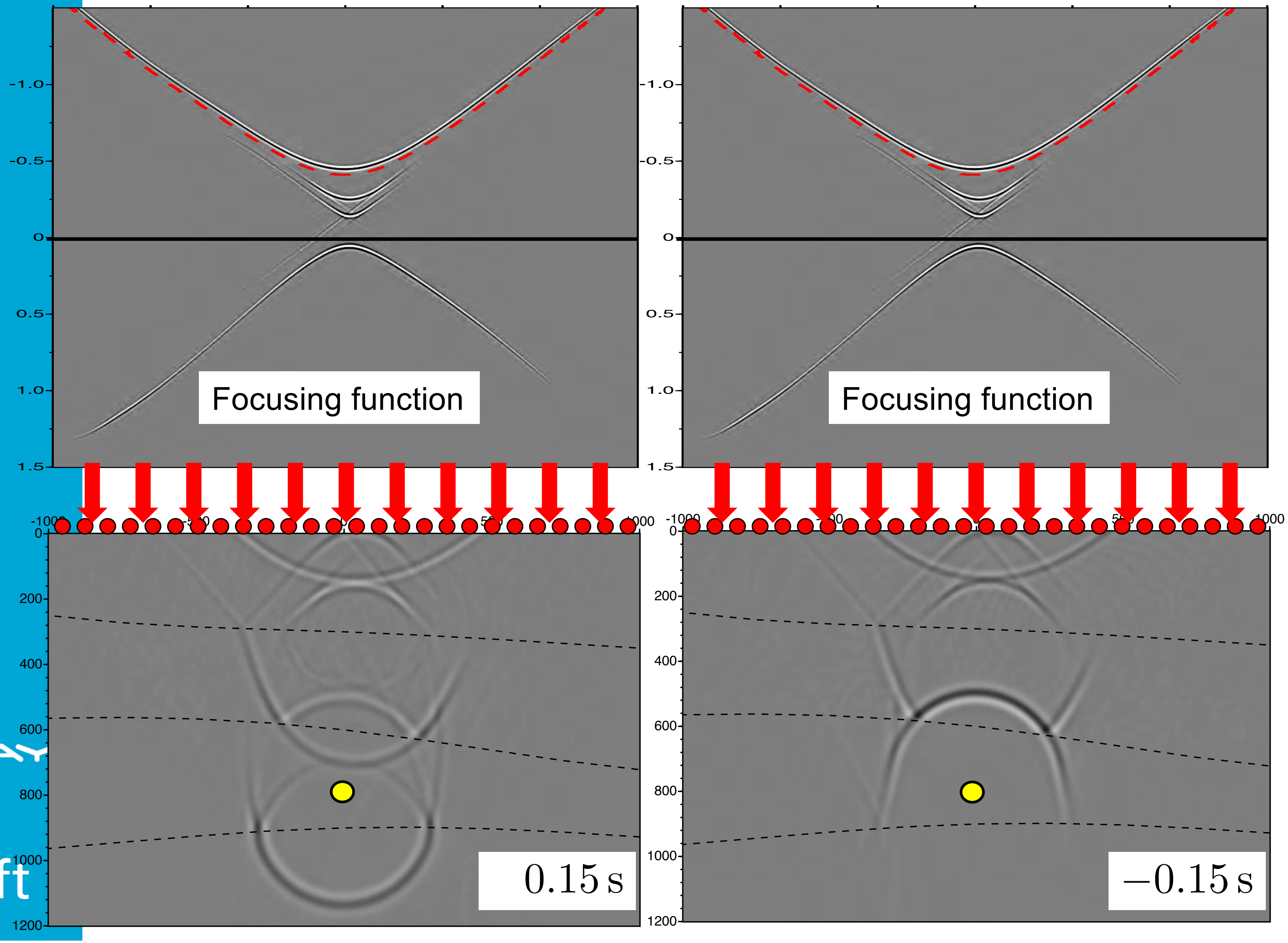

-1.0
-0.5
0
0.5
1.0
1.5
Focusing function
Focusing function
-1000
1000
0
200
400
600
800
1000
1200
−0.30 s
0.30 s

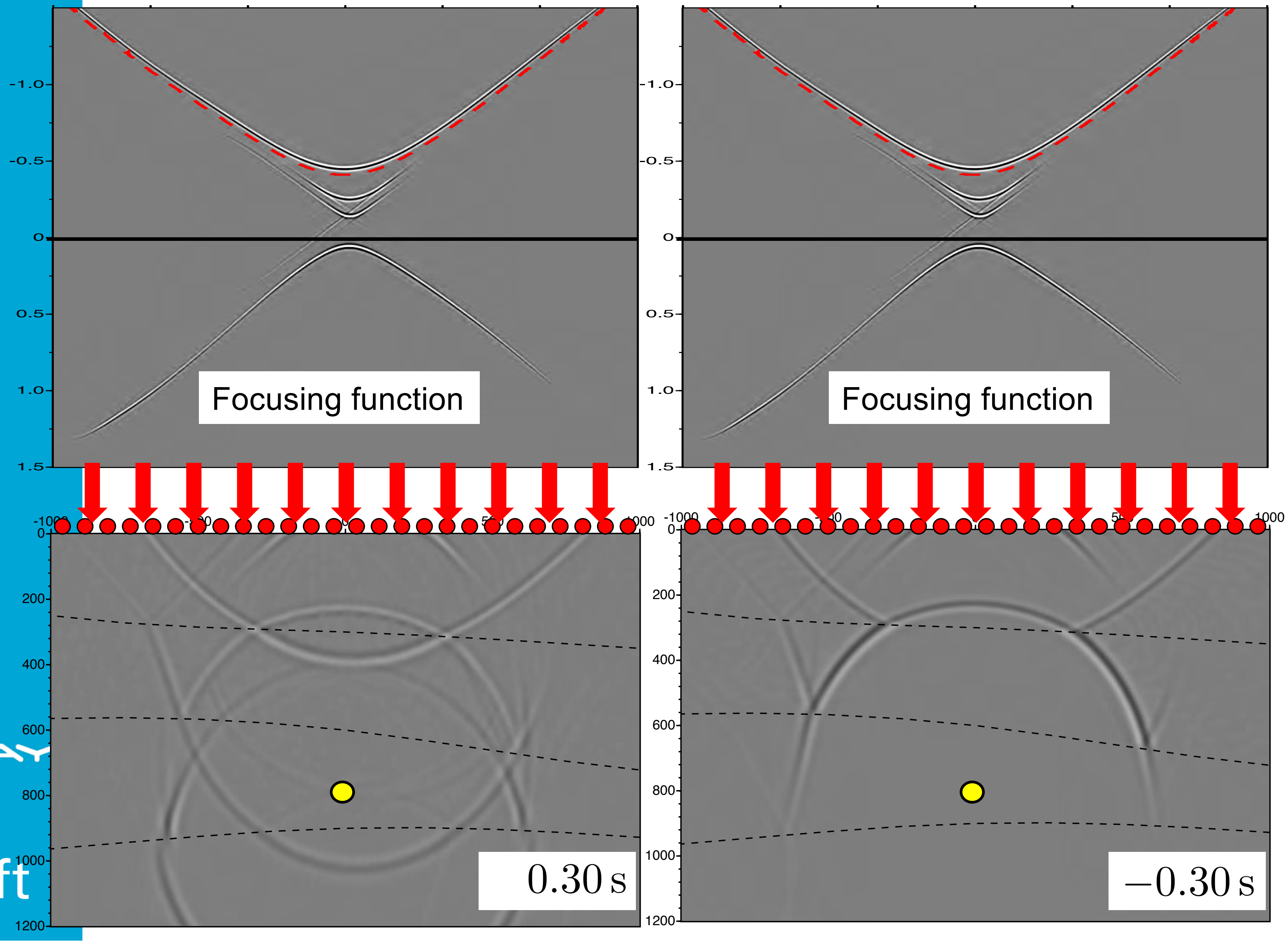

Focusing function
Focusing function
−0.15 s
0.15 s
-1.0
-0.5
0
0.5
1.0
1.5
-1000
1000
0
200
400
600
800
1000
1200

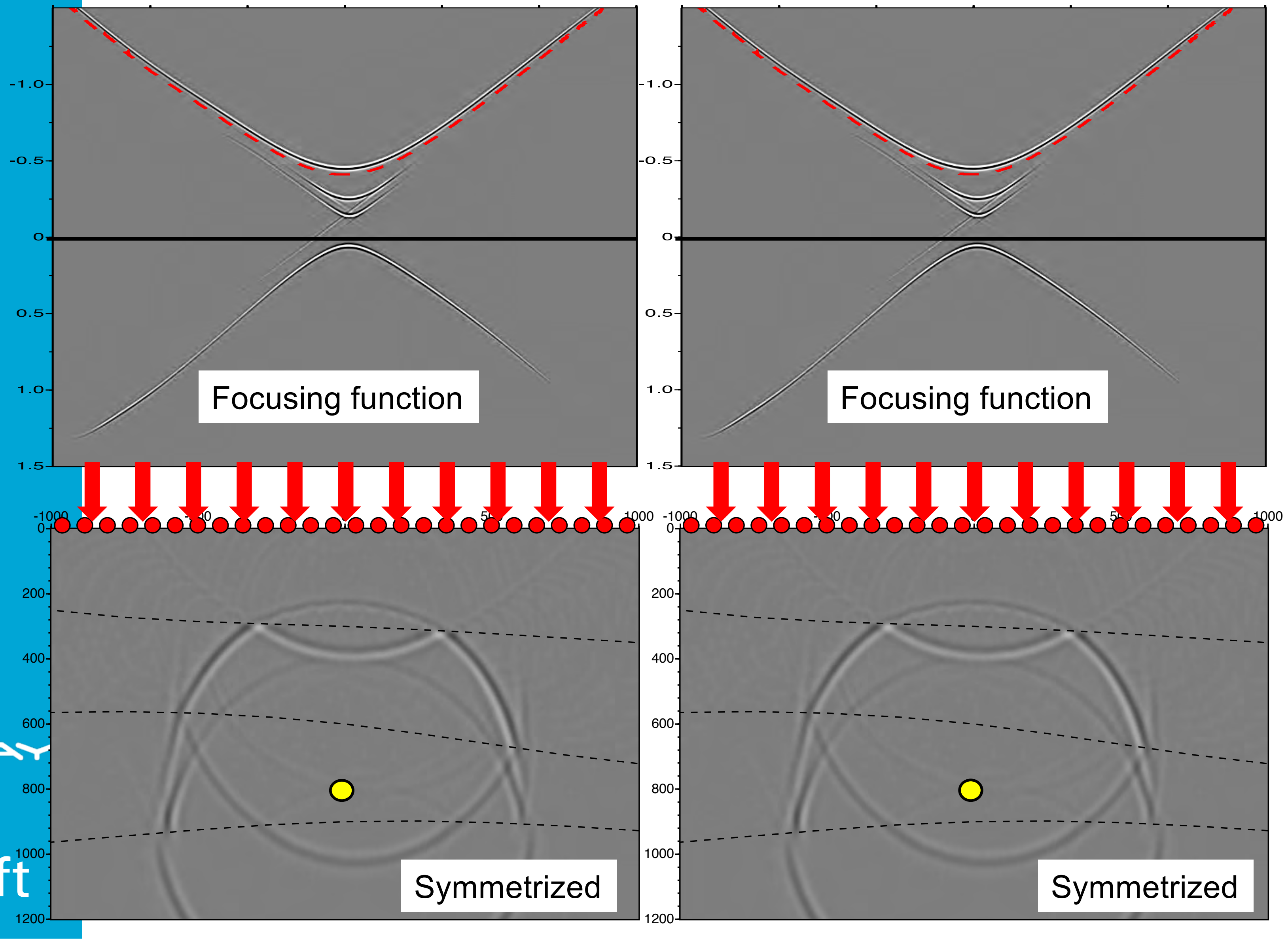

Focusing function
Focusing function
−0.03 s
0.03 s


Focusing function
Focusing function
0.00 s
0.00 s

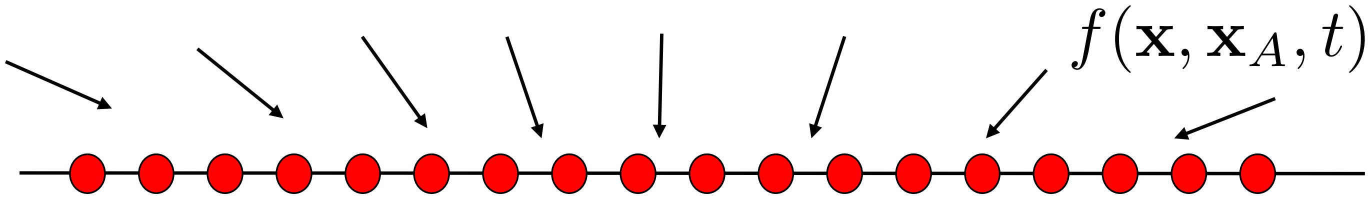

Focusing function
Focusing function
0.03 s
−0.03 s

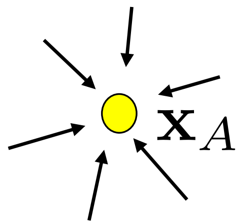

Focusing function
Focusing function
0.15 s
−0.15 s


-1.0
-0.5
0
0.5
1.0
1.5
Focusing function
Focusing function
0
200
400
600
800
1000
1200
0.30 s
−0.30 s



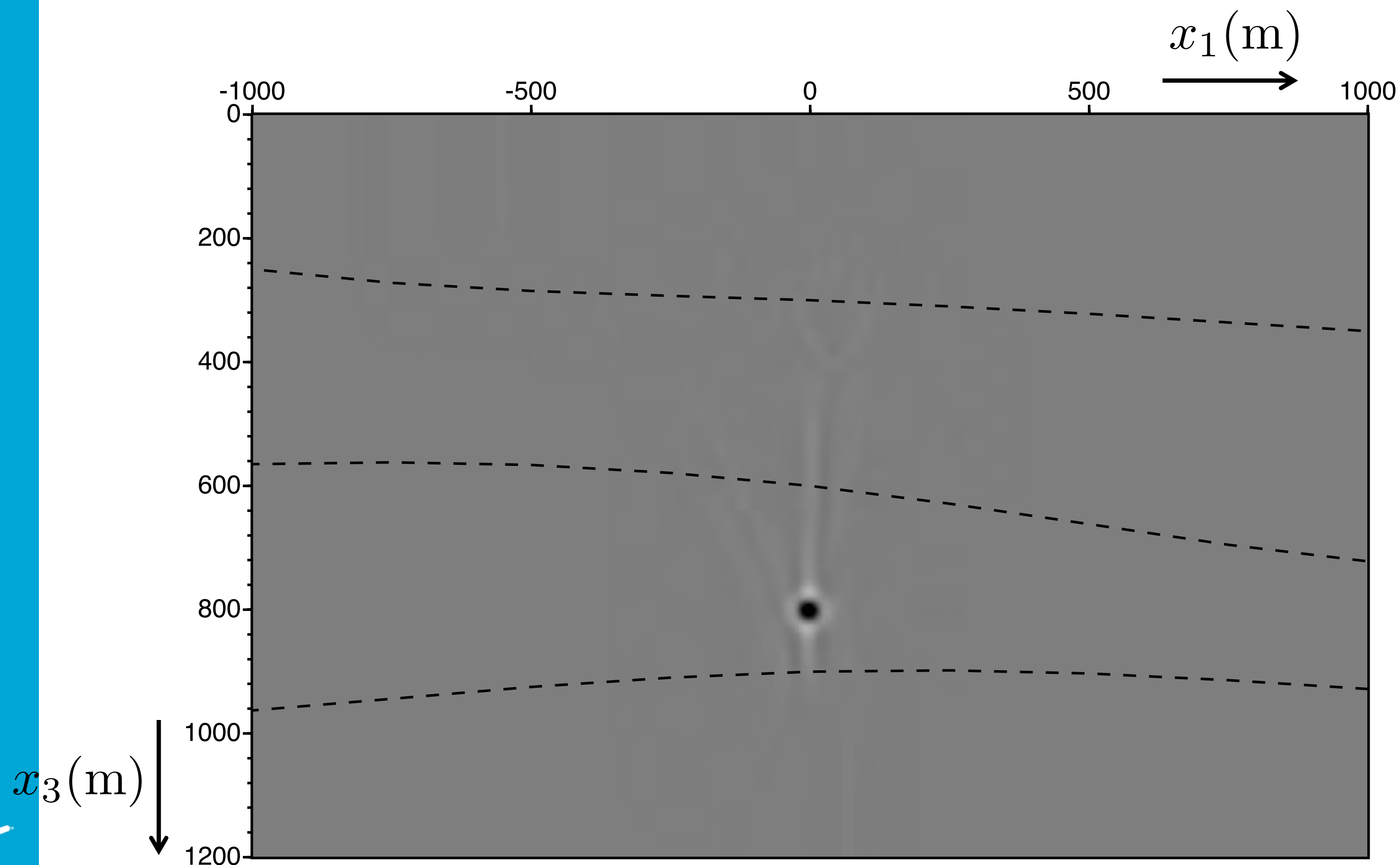
Focusing function
Focusing function
Symmetrized
Symmetrized



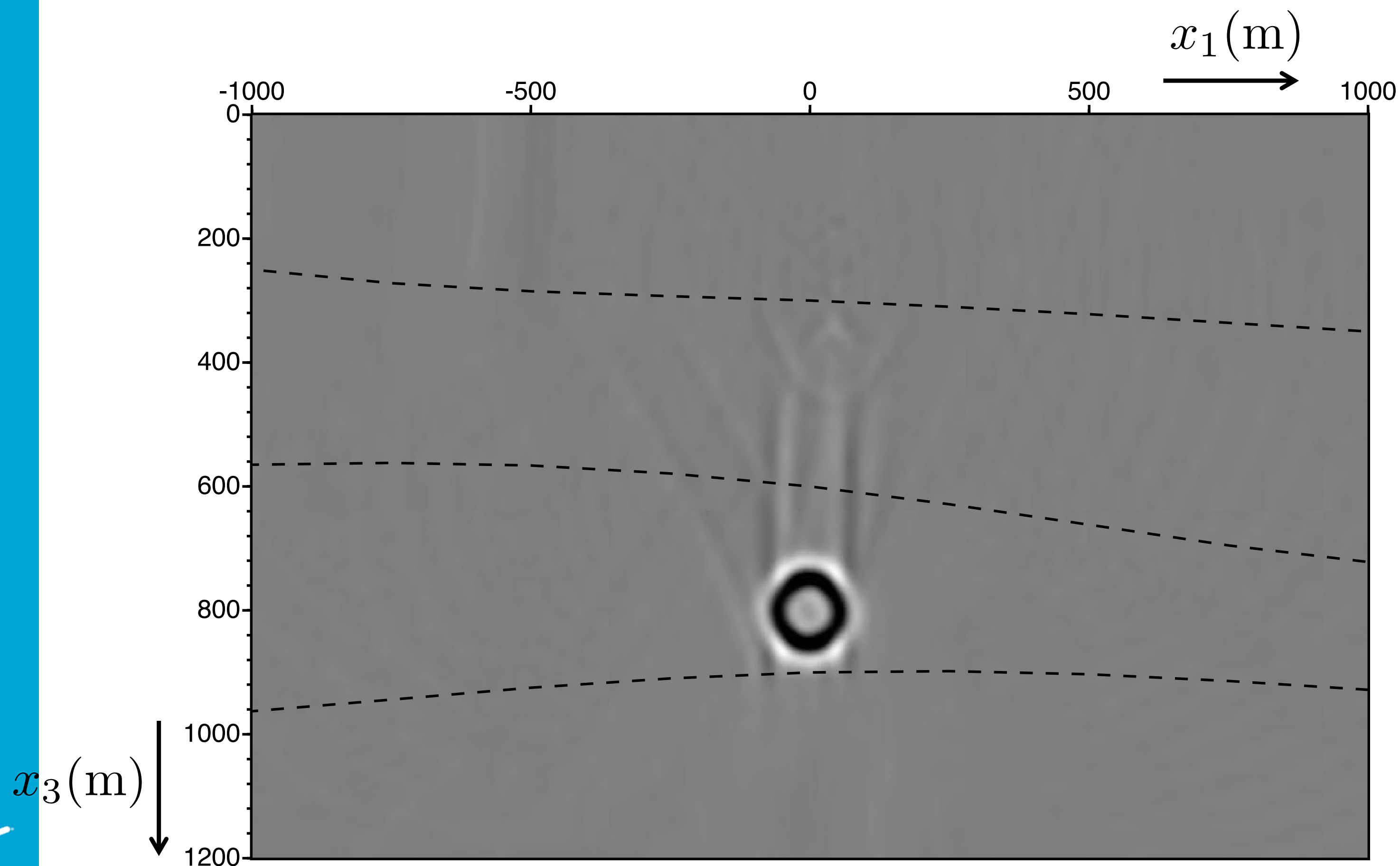


$$G(\mathbf{x}_B, \mathbf{x}_A, t) + G(\mathbf{x}_B, \mathbf{x}_A, -t) \propto \int_{\partial \mathbb{D}} G(\mathbf{x}_B, \mathbf{x}, t) * f(\mathbf{x}, \mathbf{x}_A, t) \mathrm{d}^2\mathbf{x}$$

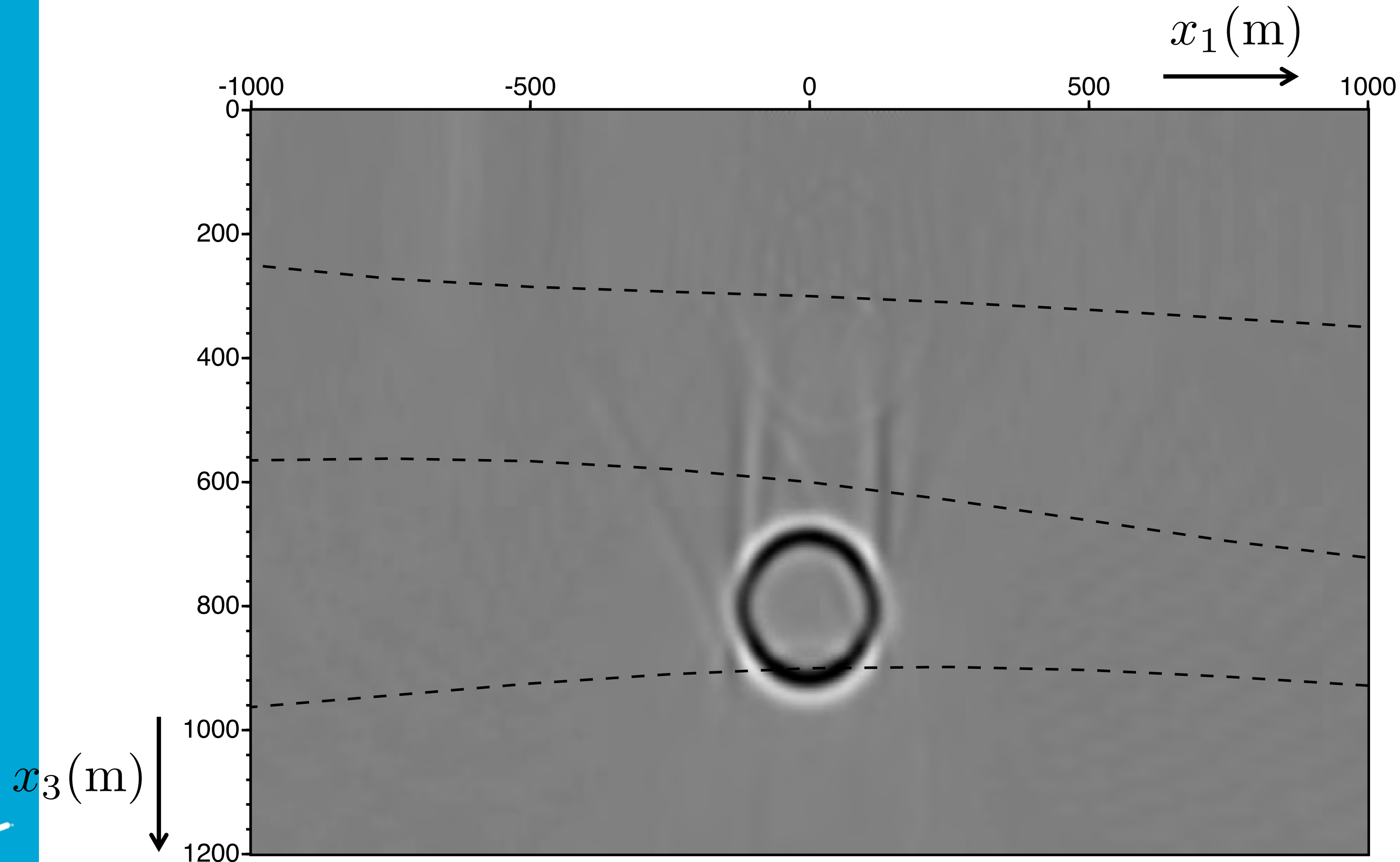
$$+ \int_{\partial \mathbb{D}} G(\mathbf{x}_B, \mathbf{x}, -t) * f(\mathbf{x}, \mathbf{x}_A, -t) \mathrm{d}^2\mathbf{x}$$



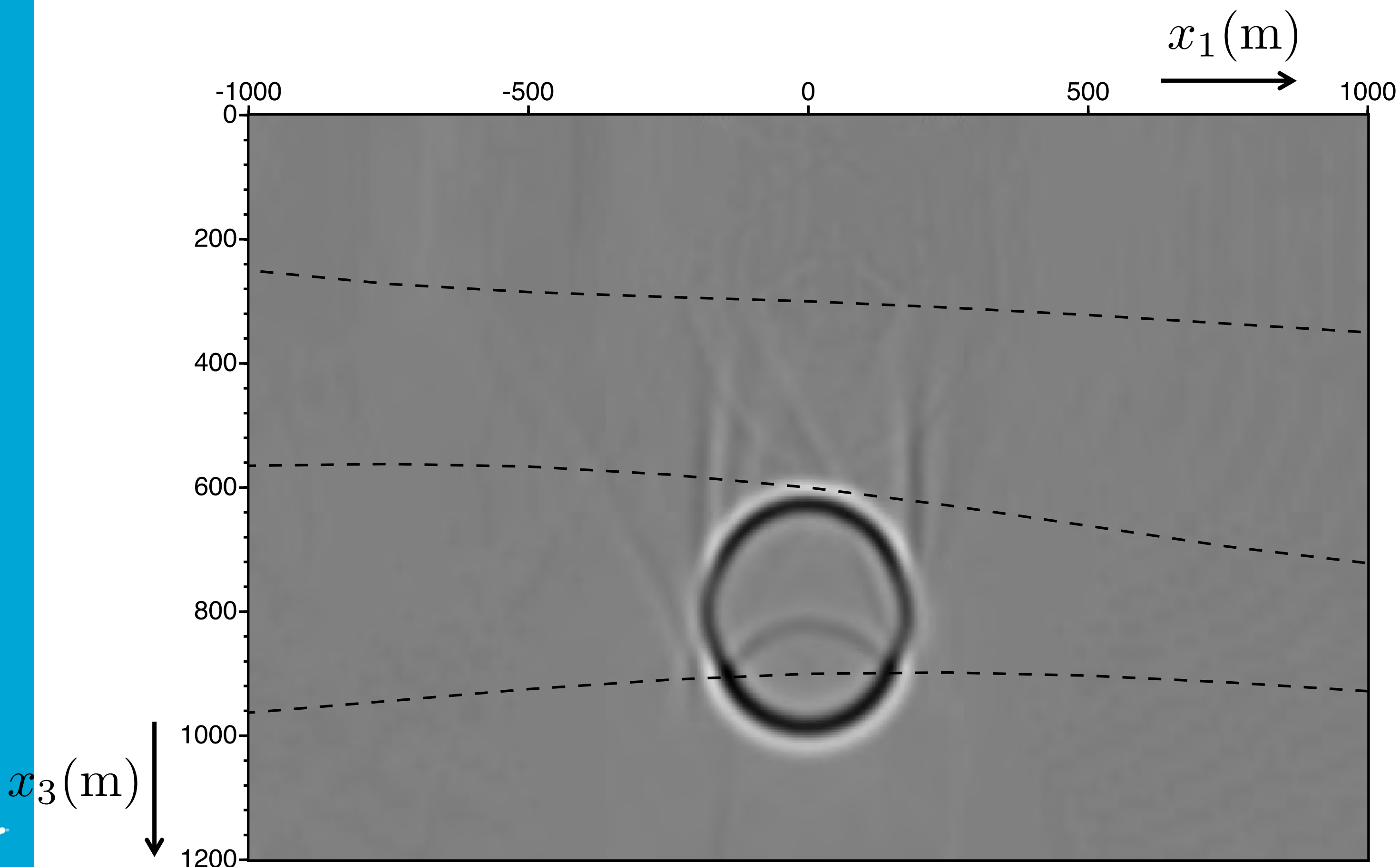
$t = 0.001$s



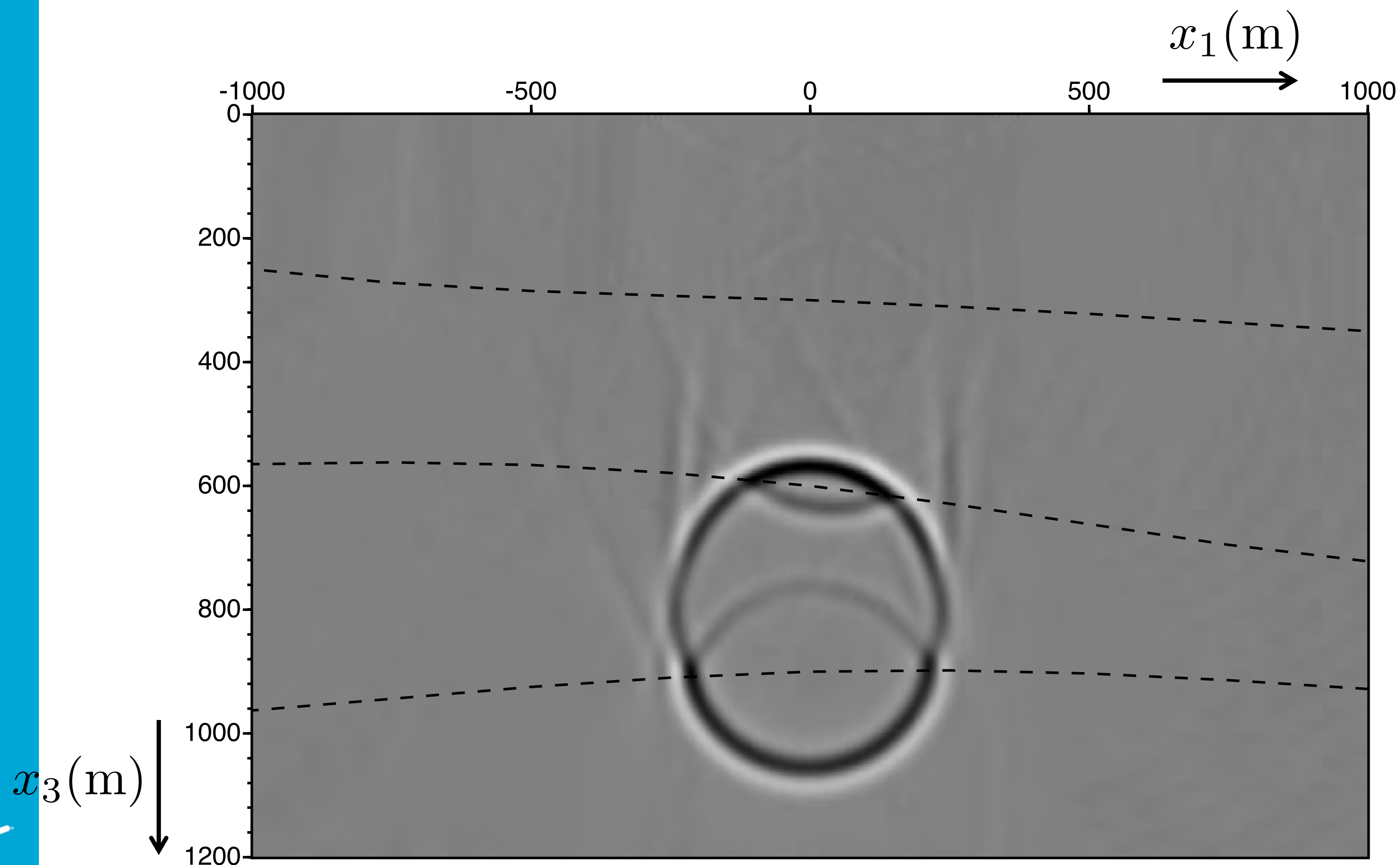
$t = 0.03$s



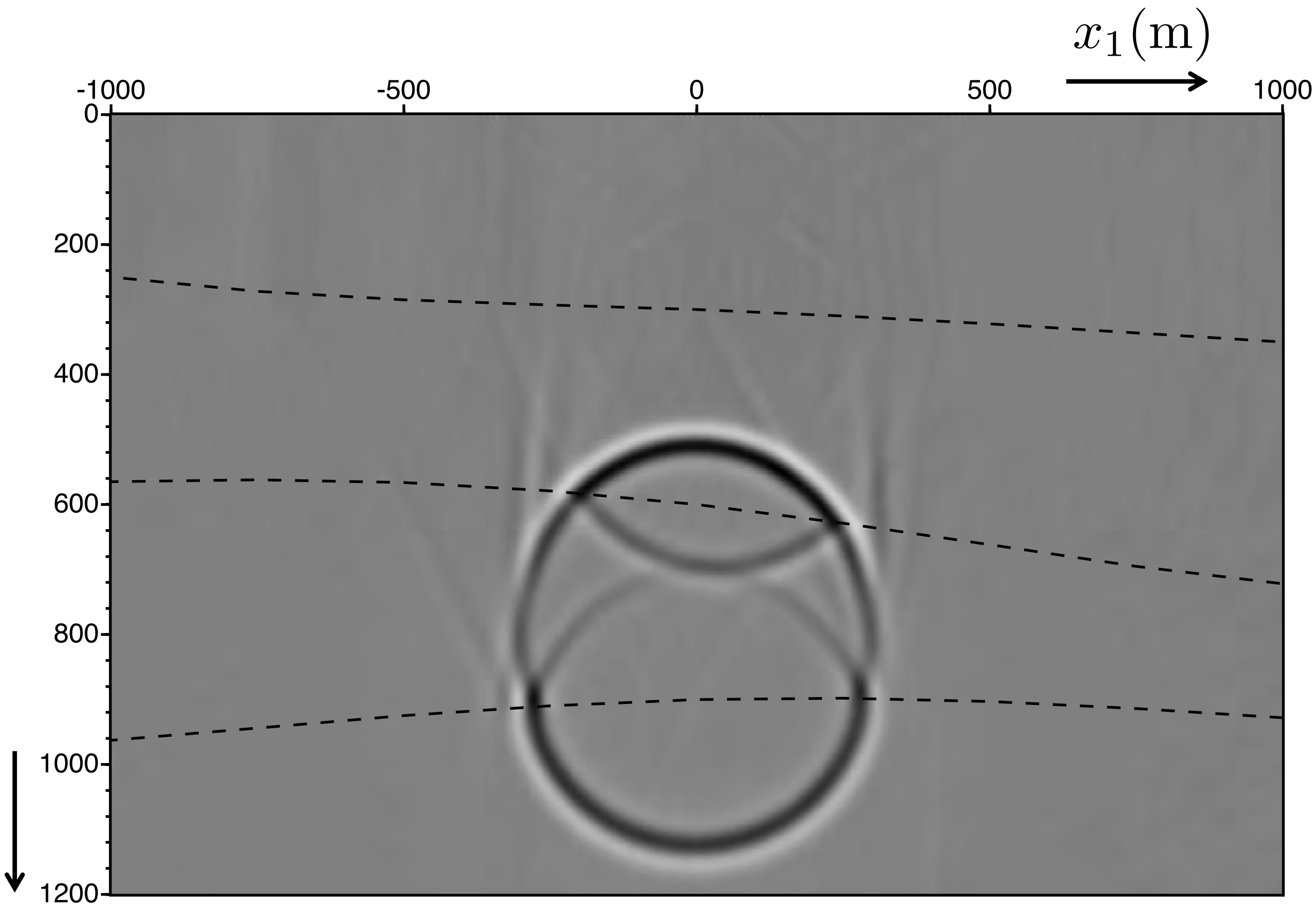
$t = 0.06\mathrm{s}$



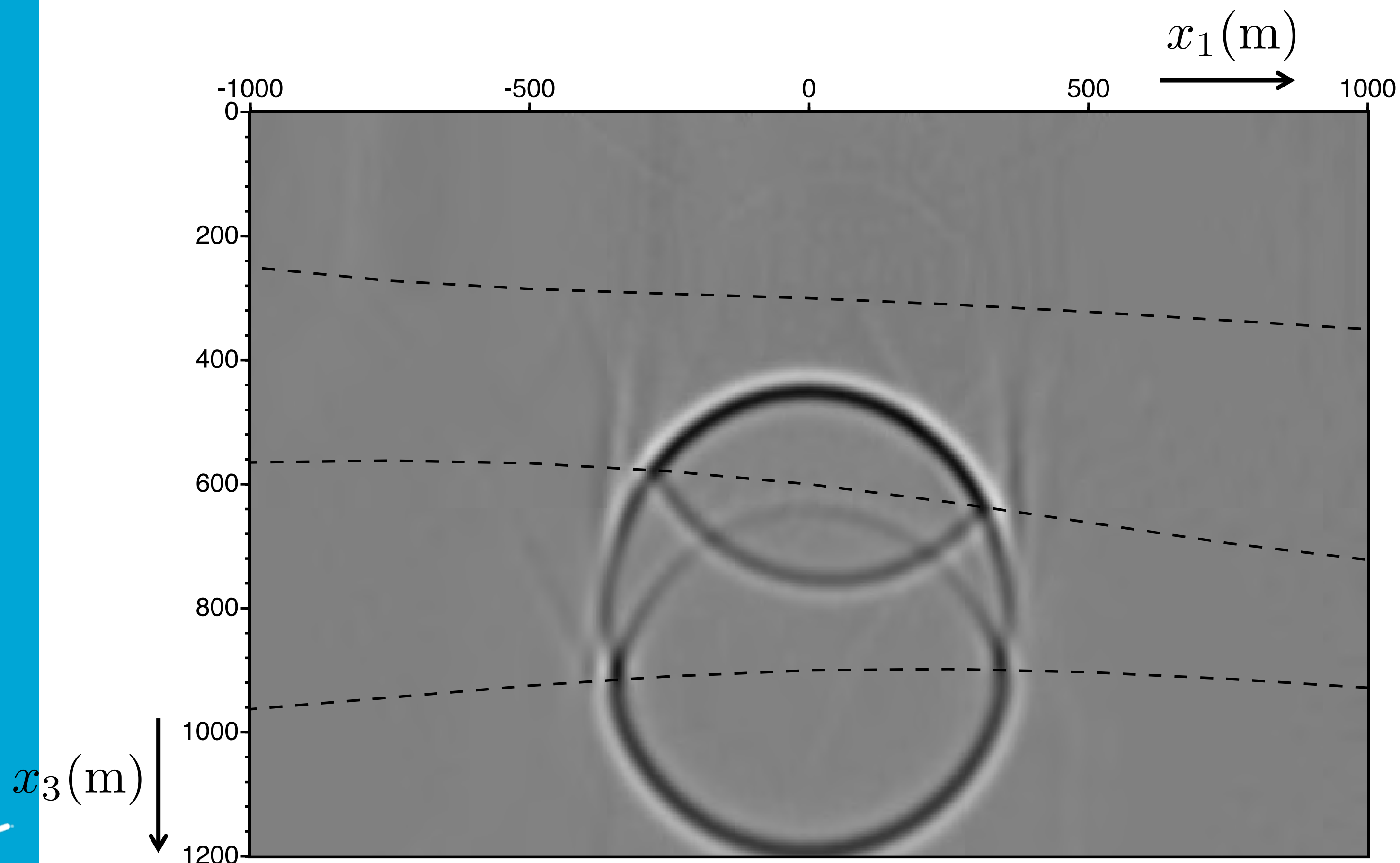
$t = 0.09\text{s}$



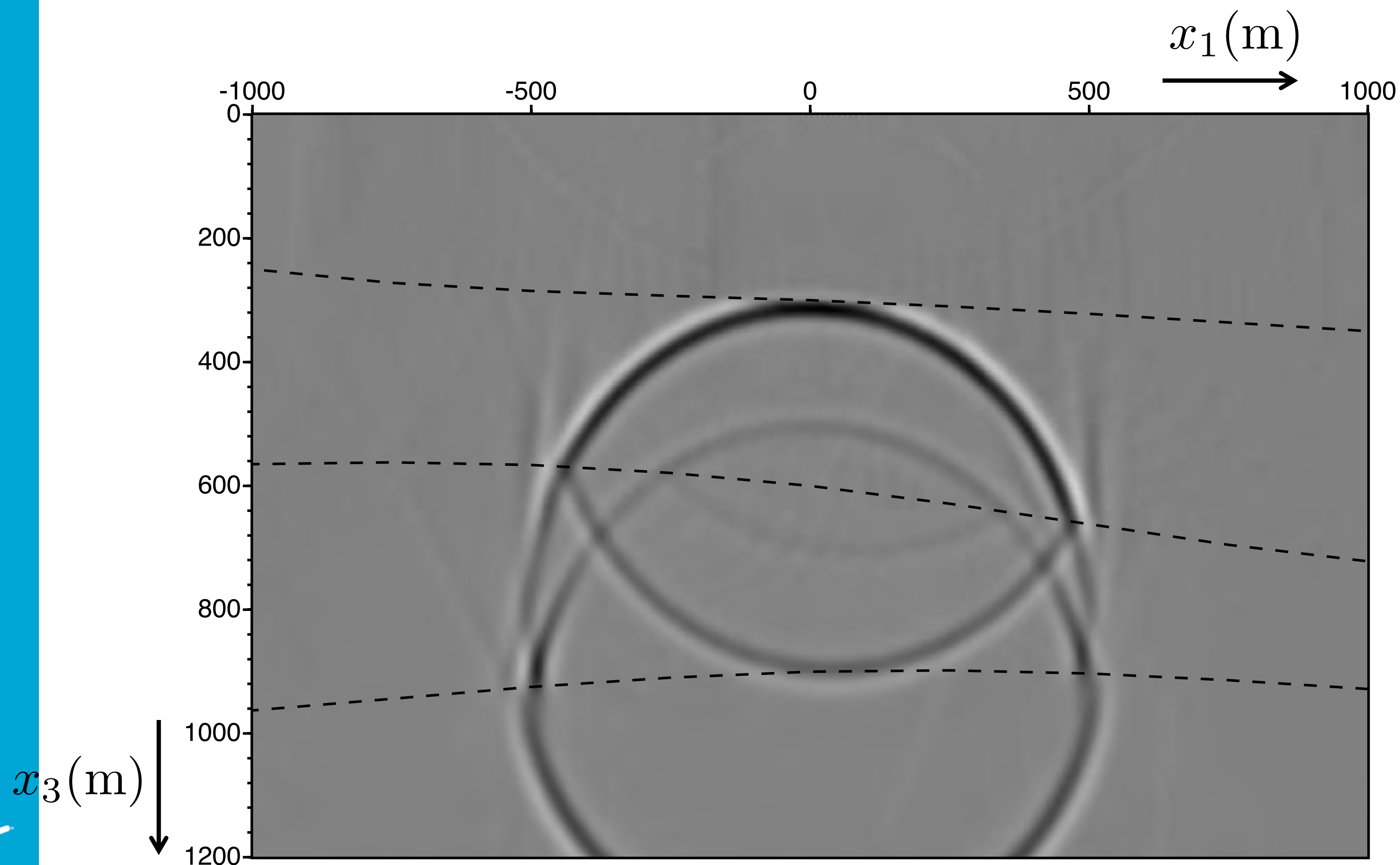
$t = 0.12$s



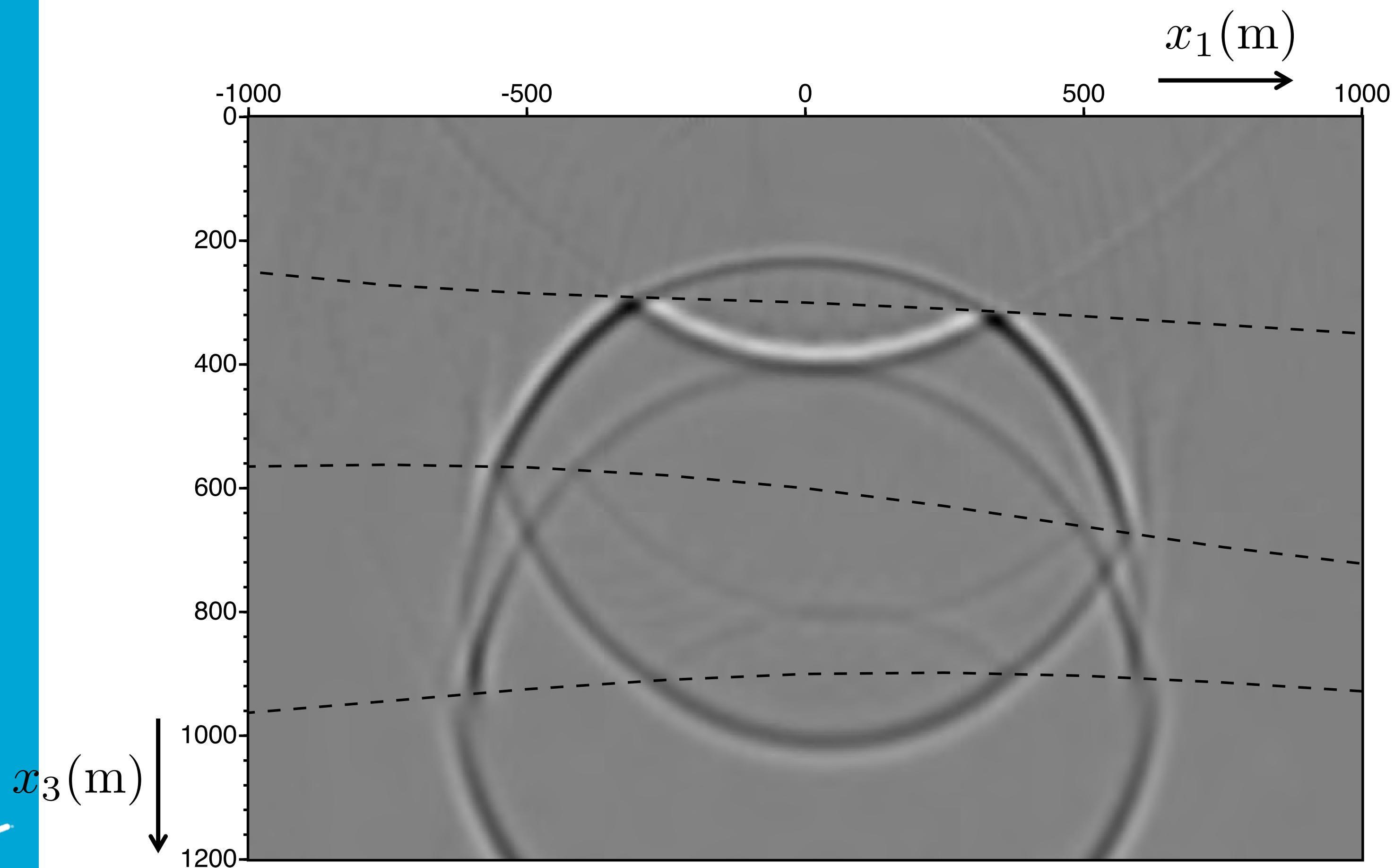
$t = 0.15$s



$t = 0.18$s

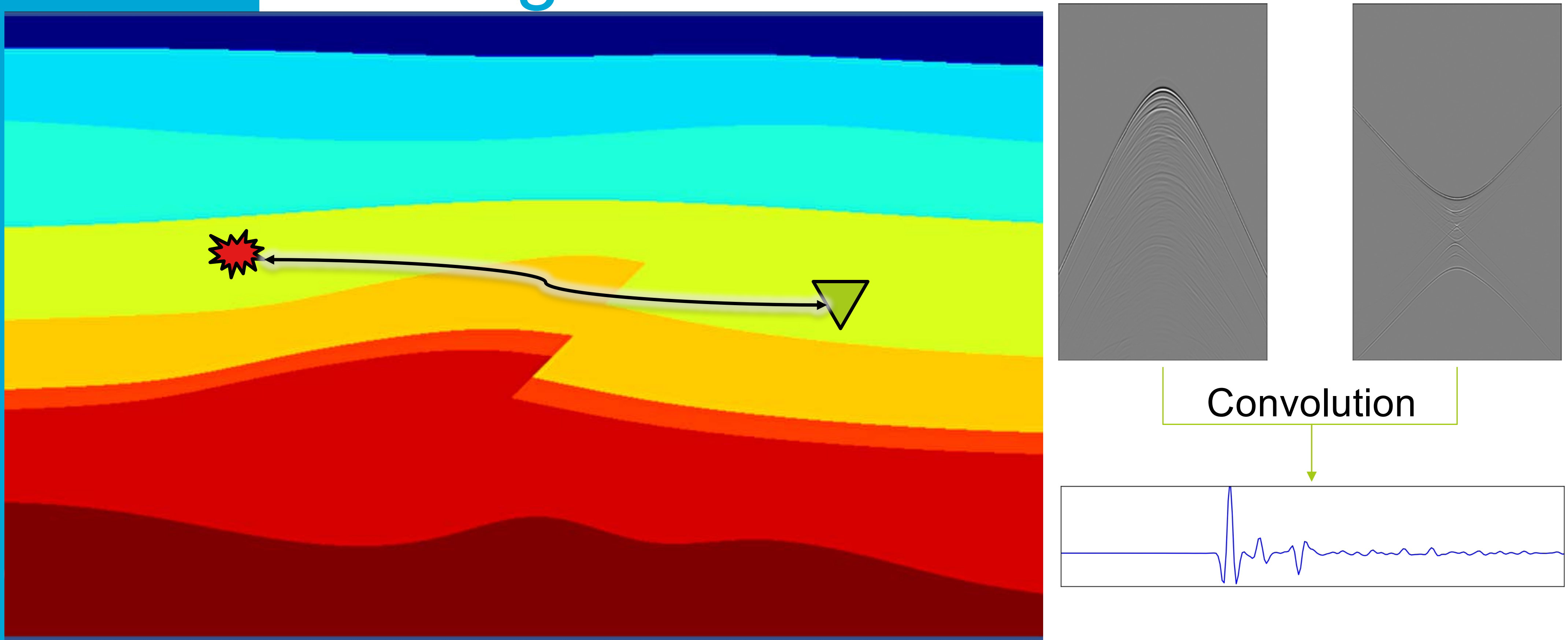


$t = 0.25\mathrm{s}$



$t = 0.30\mathrm{s}$

Homogeneous Green’s function



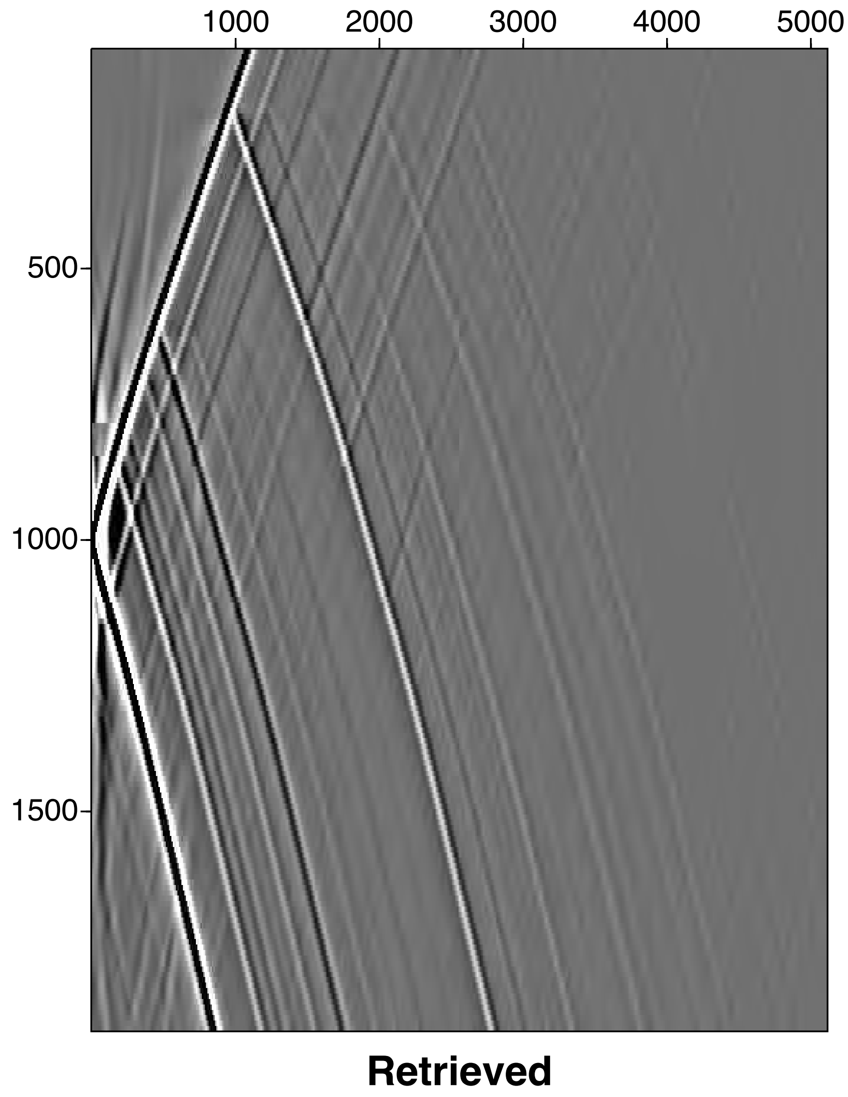
Homogeneous Green’s function

- README.5
- Generate snapshots for homogeneous Green's function

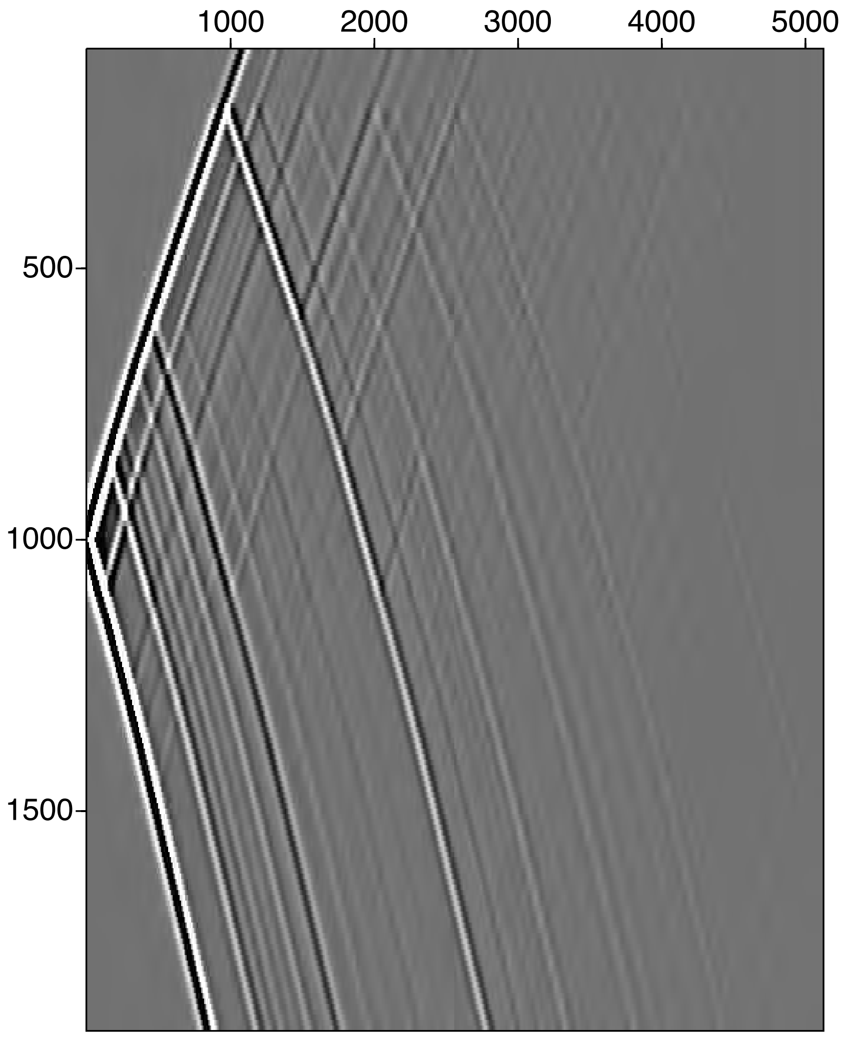
20 min.

Reference: Brackenhoff, J., Thorbecke, J., and Wapenaar, K.: *Monitoring induced distributed double-couple sources using Marchenko-based virtual receivers*, Solid Earth Discuss., https://doi.org/10.5194/se-2018-142, in review, 2019.

demo/twoD



demo/twoD



Reference

Group Discussion

SAGA data examples

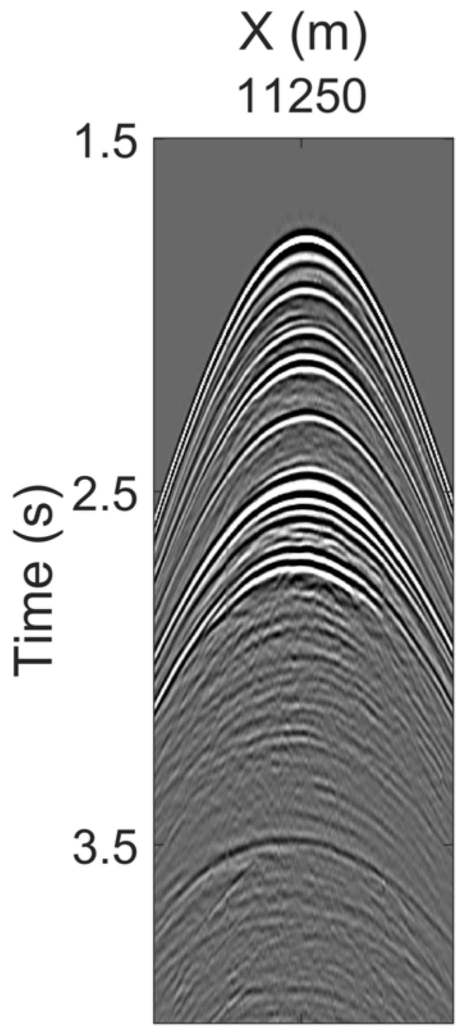
- Homogeneous Green’s function movie directed by Joeri.
- Primaries extraction featured by Lele

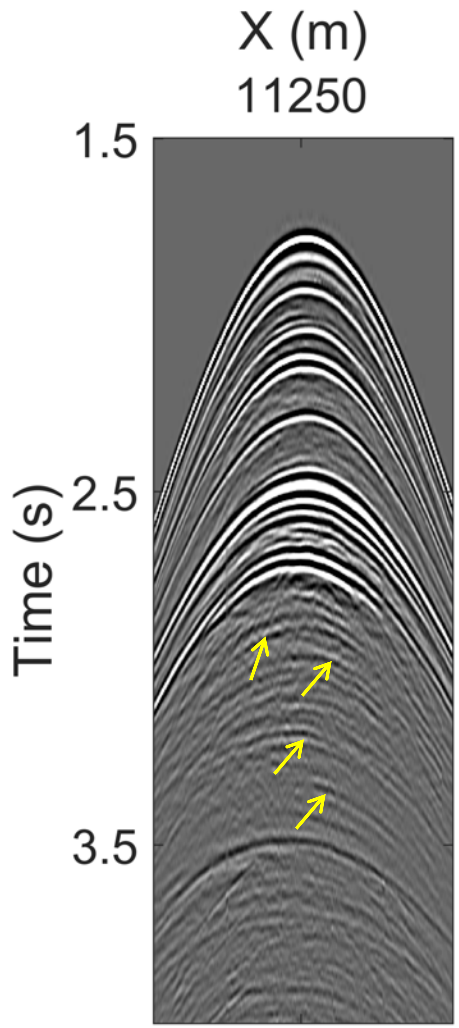
Reference: Brackenhoff, J., Thorbecke, J., and Wapenaar, K.: *Monitoring induced distributed double-couple sources using Marchenko-based virtual receivers,* Solid Earth Discuss., https://doi.org/10.5194/se-2018-142, in review, 2019.

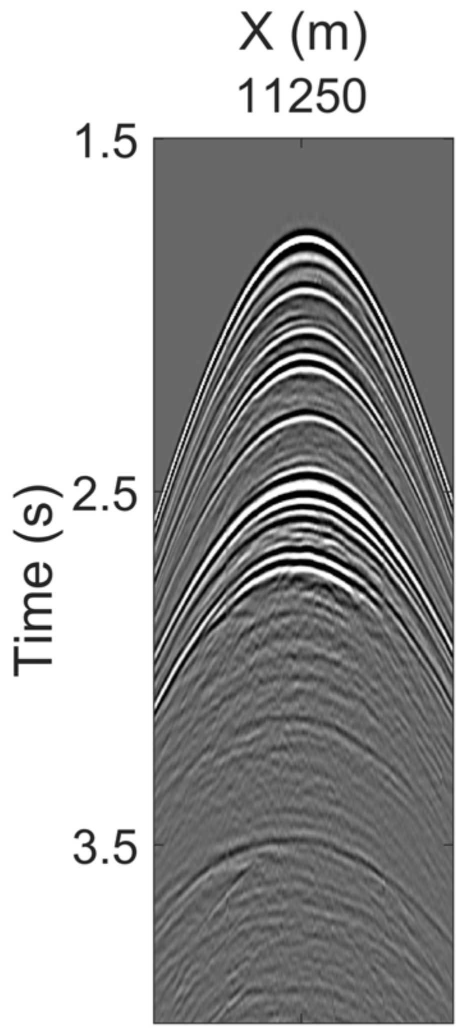
Field data

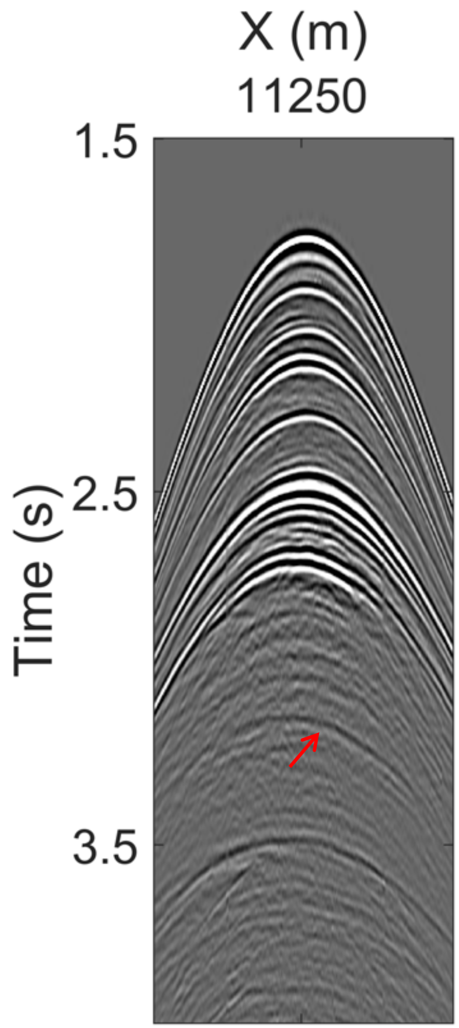
- Data recorded in a marine environment
- Acoustic scheme applied
- No free-surface multiples included
  - EPSI: wavelet deconvolved
- Q-compensation and scalar correction applied

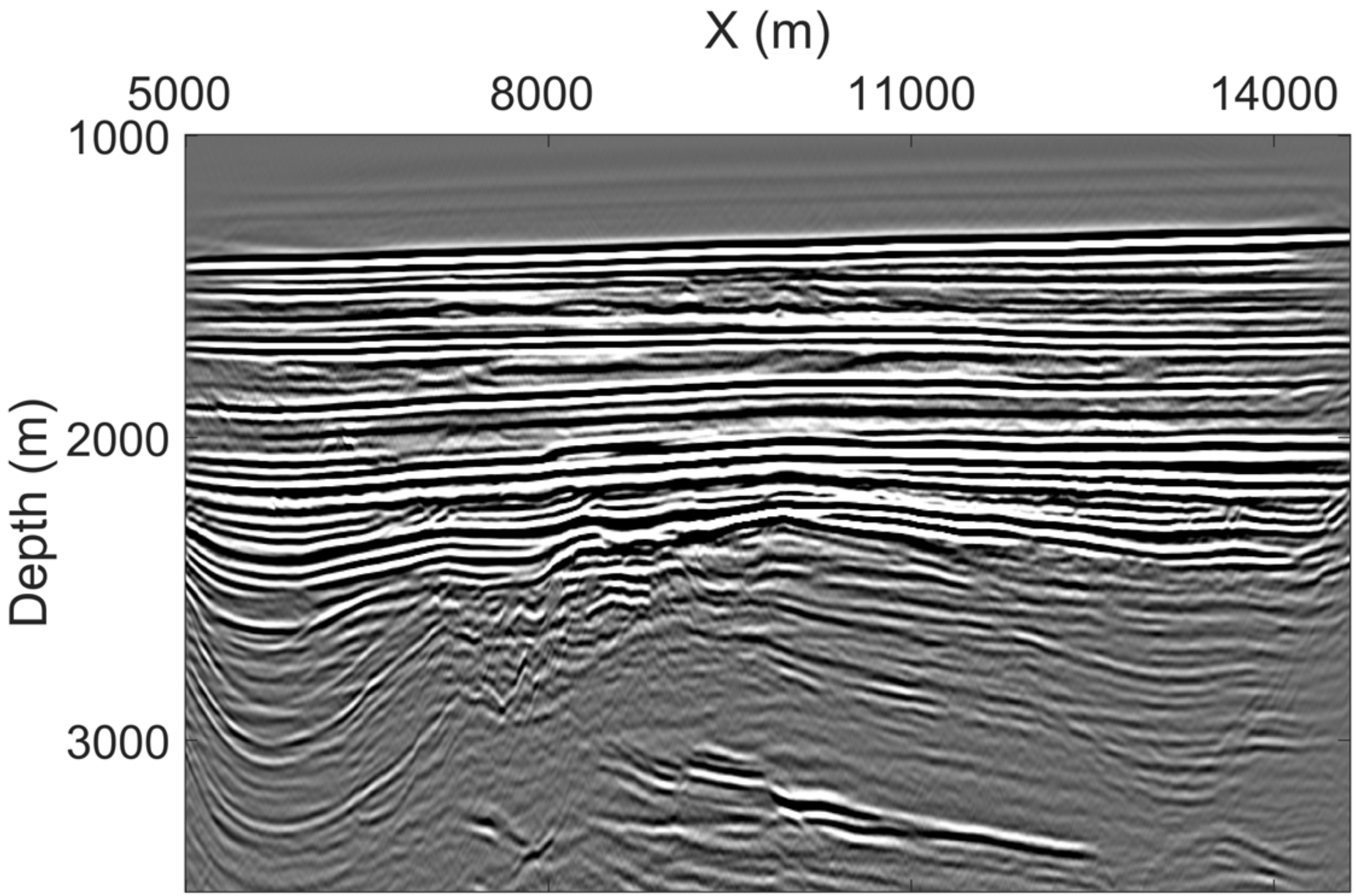
Primaries

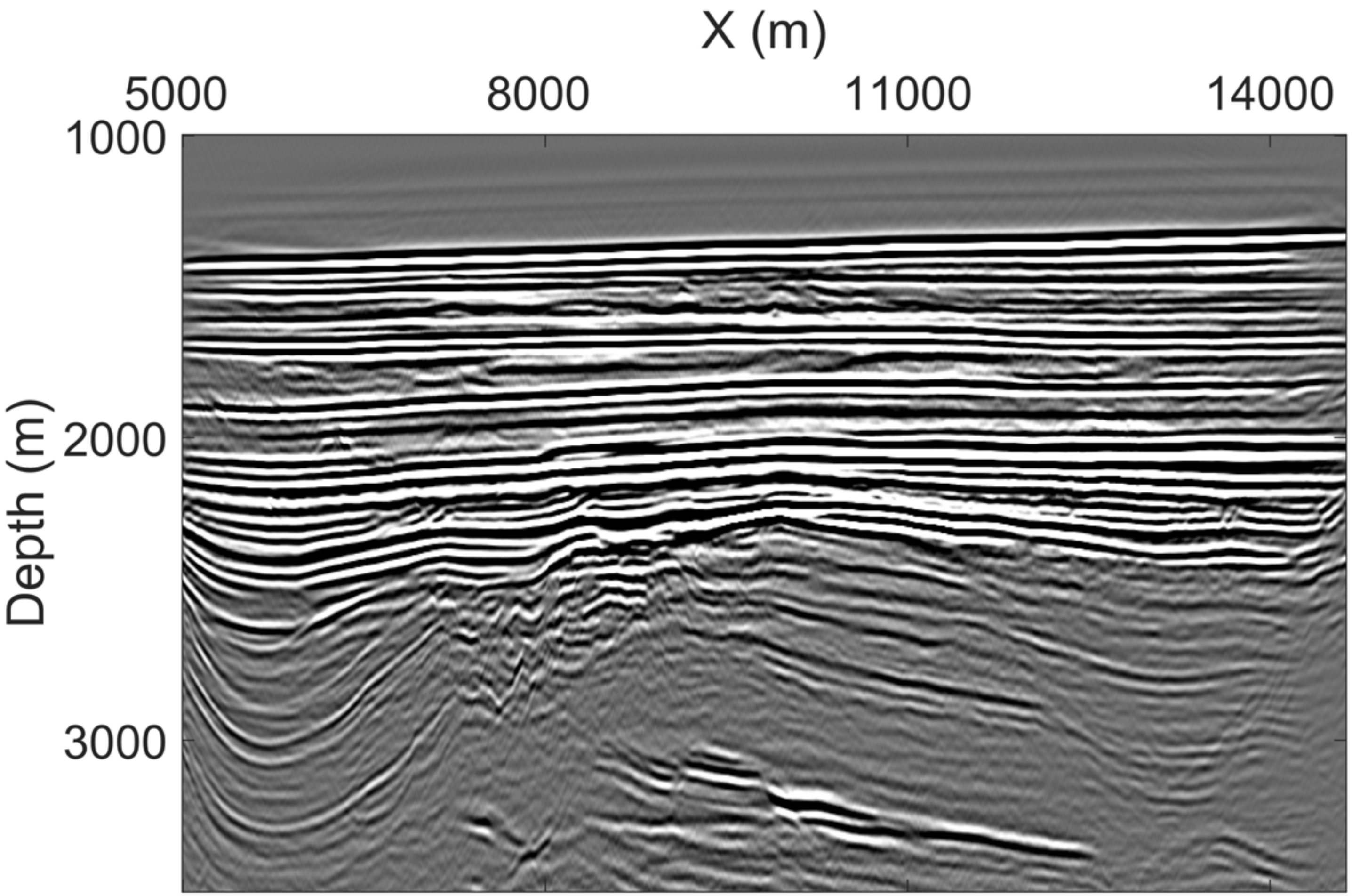

X (m)
11250
1.5
2.5
3.5
Time (s)

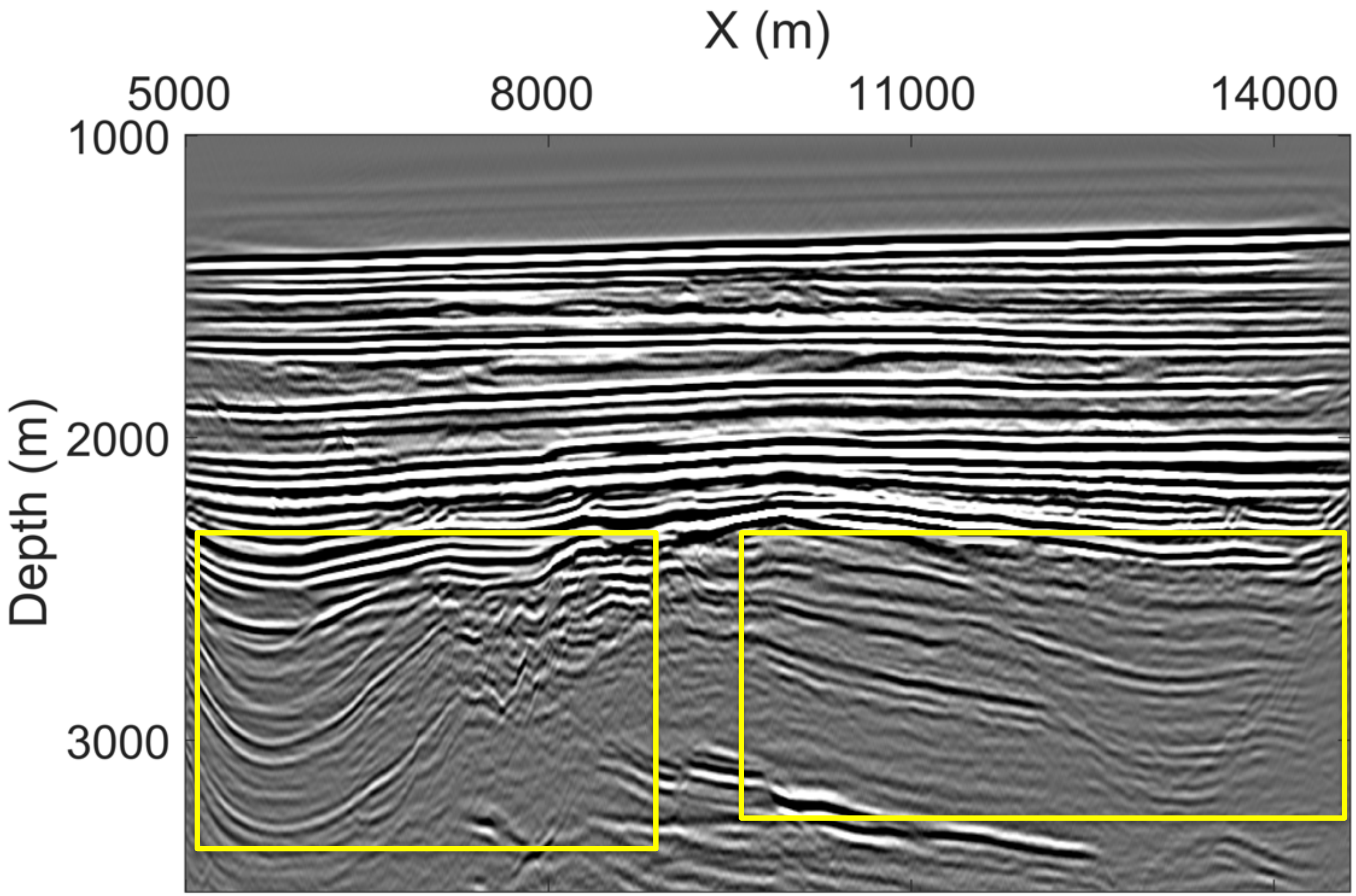

X (m)
11250
1.5
2.5
3.5
Time (s)

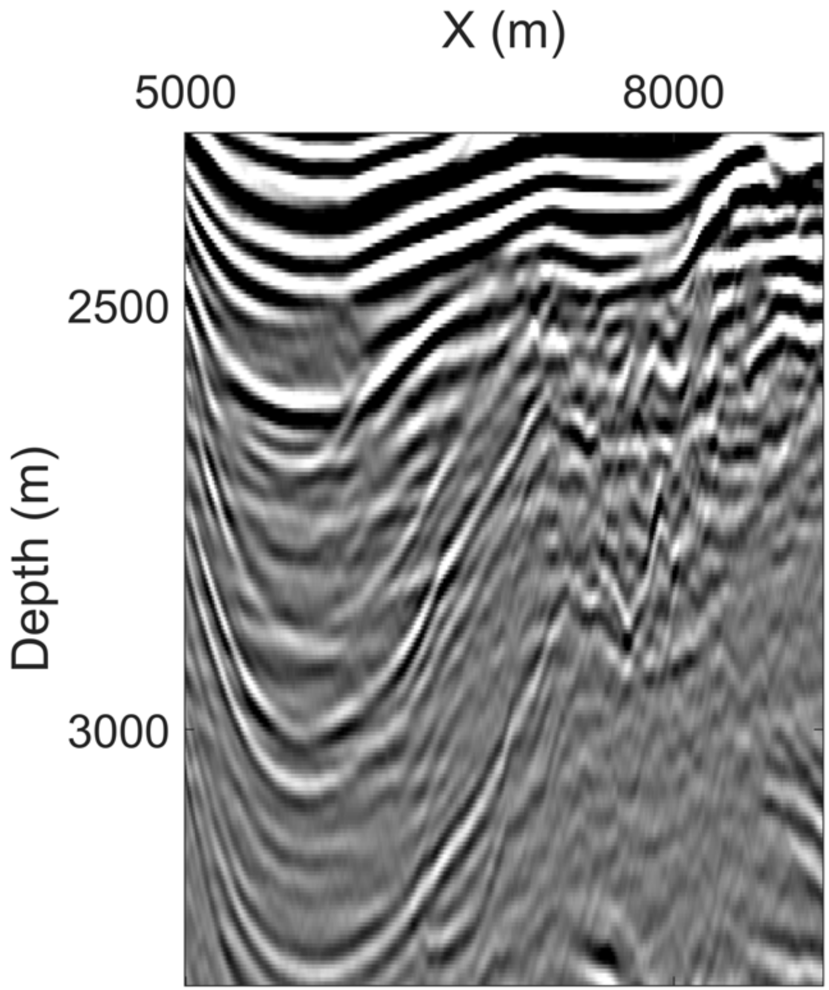

X (m)
11250
1.5
2.5
3.5
Time (s)

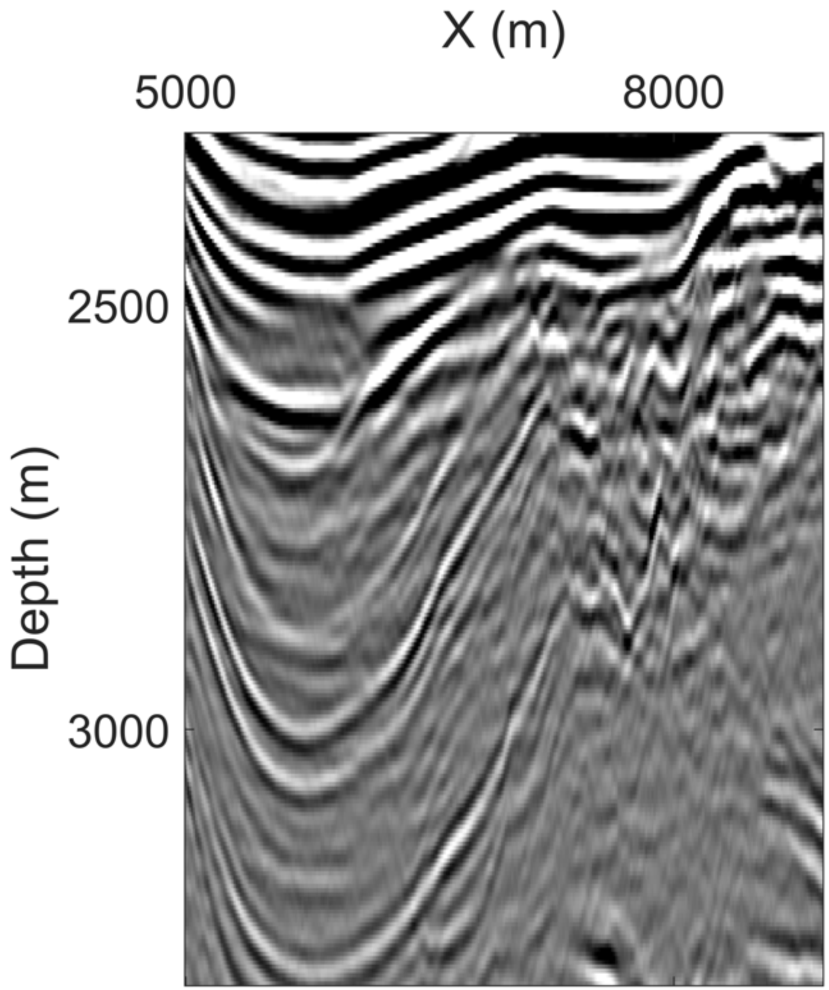

X (m)
11250
1.5
2.5
3.5
Time (s)

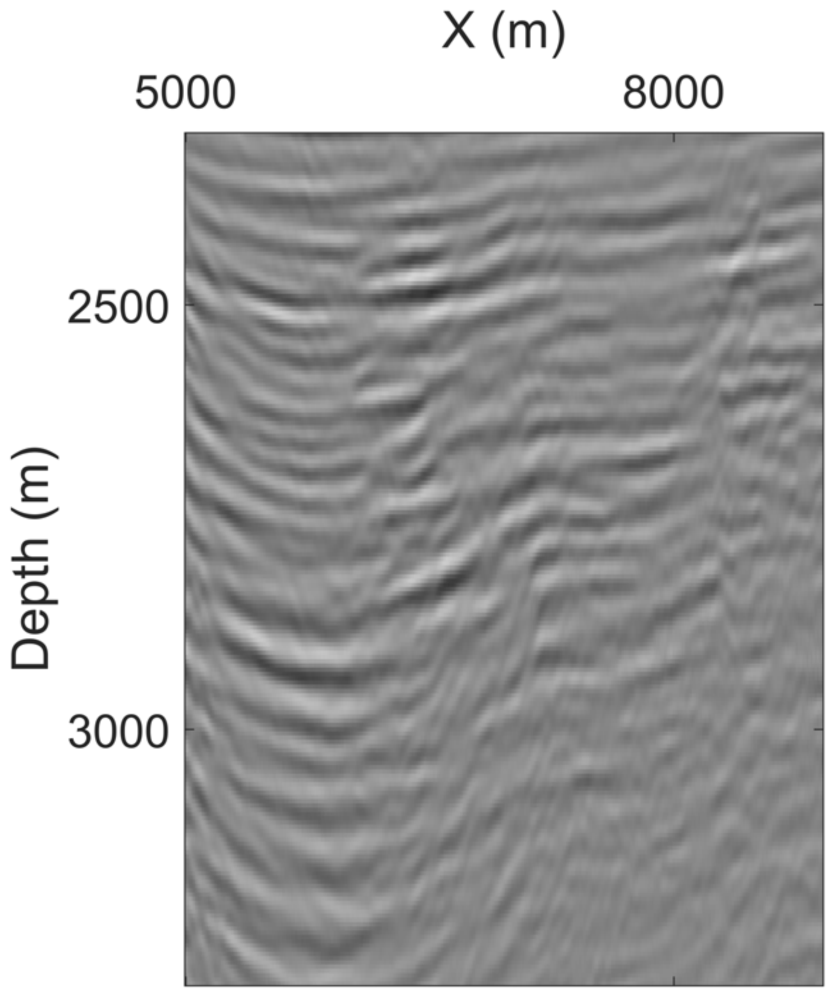

X (m)
5000
8000
11000
14000
1000
2000
3000
Depth (m)

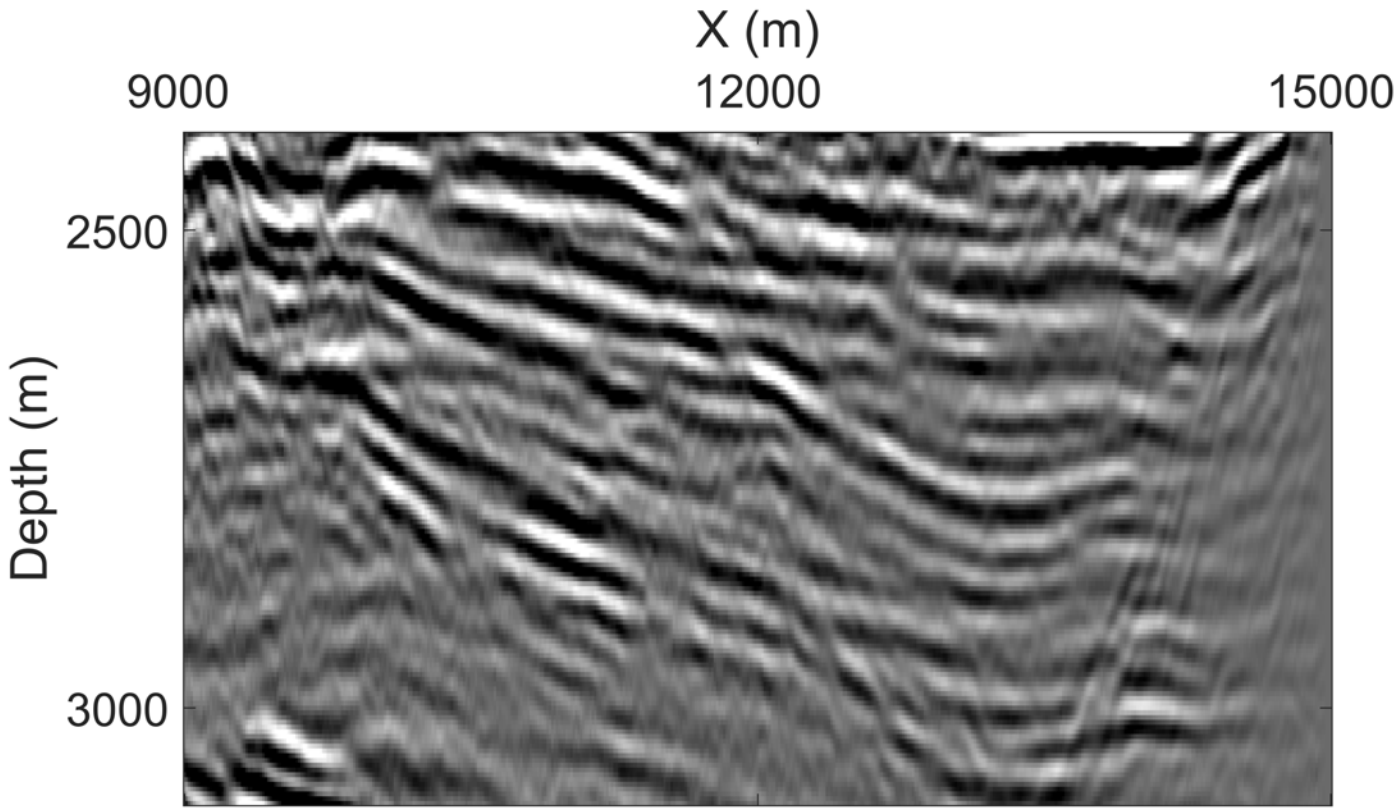

X (m)
5000
8000
11000
14000
Depth (m)
1000
2000
3000

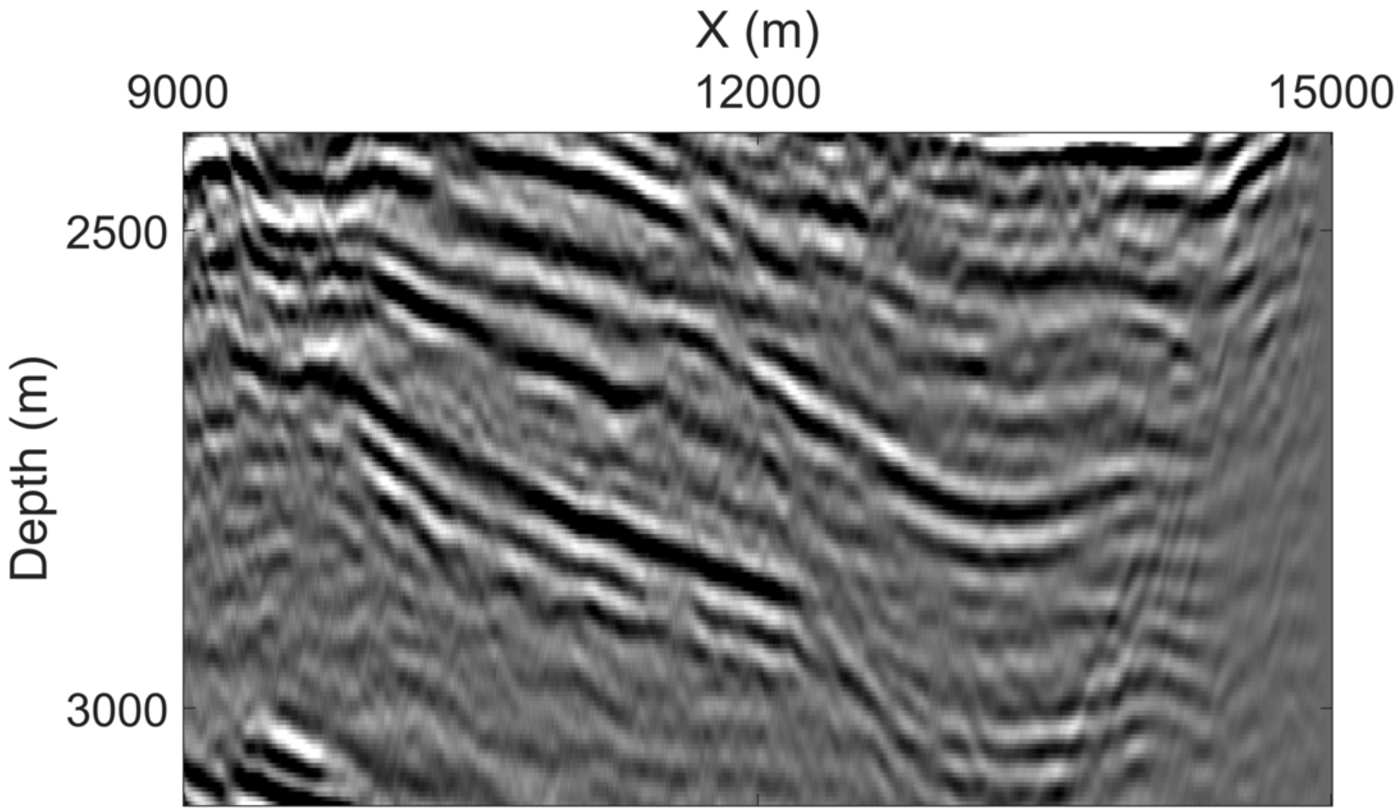

X (m)
5000
8000
11000
14000
1000
2000
3000
Depth (m)

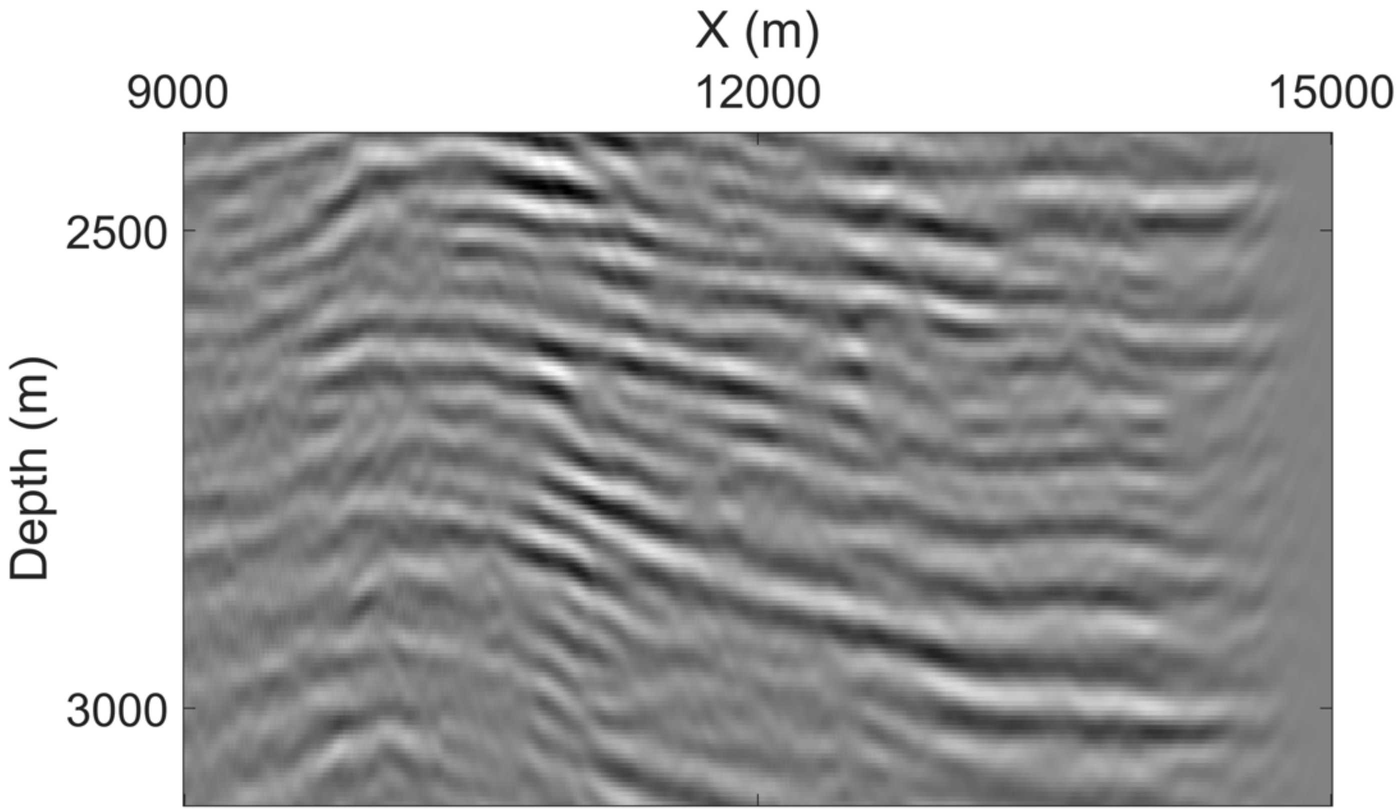

X (m)
5000
8000
2500
3000
Depth (m)


X (m)
5000
8000
Depth (m)
2500
3000


X (m)
5000
8000
2500
3000
Depth (m)

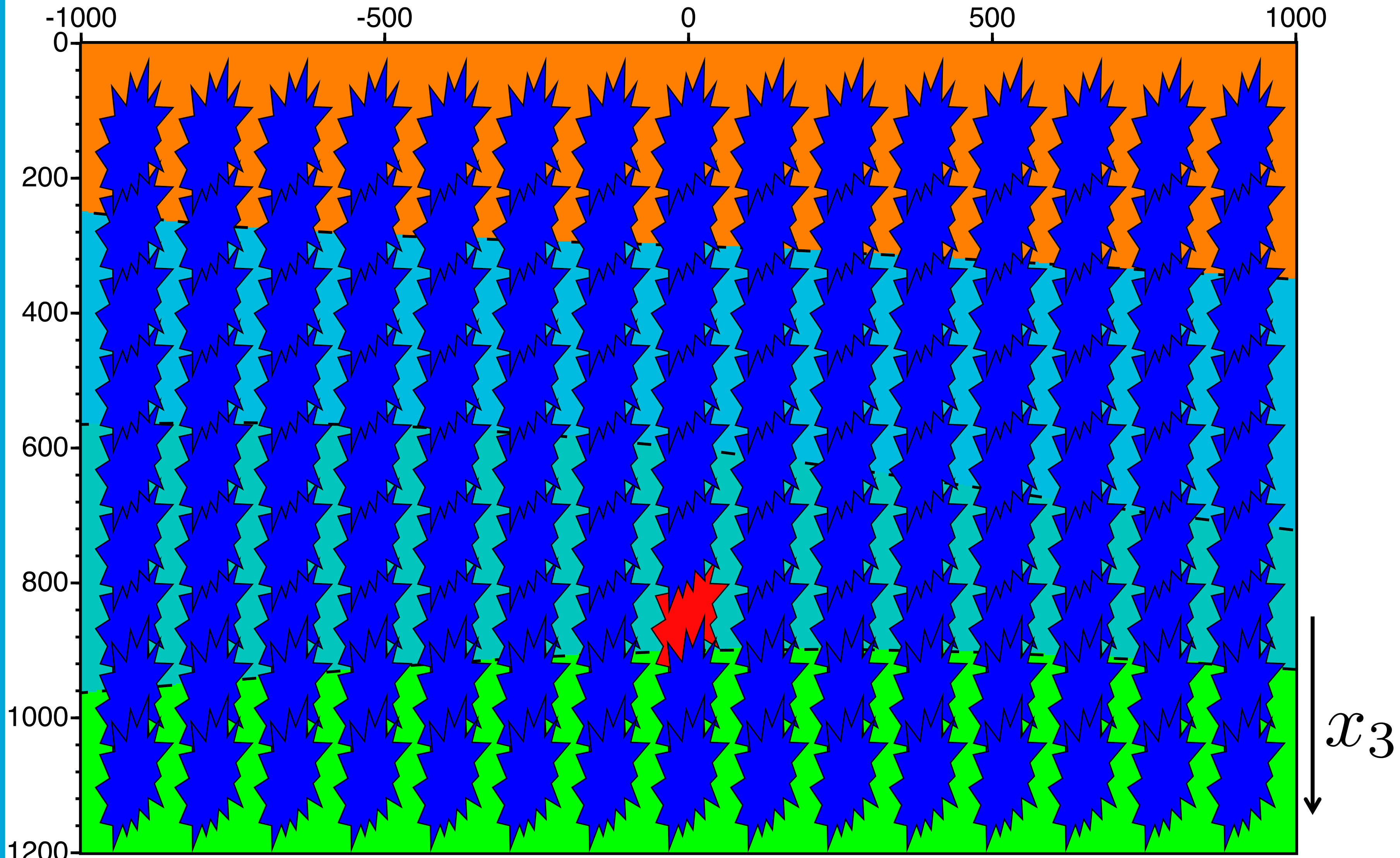

X (m)
9000
12000
15000
Depth (m)
2500
3000


X (m)
9000
12000
15000
2500
3000
Depth (m)

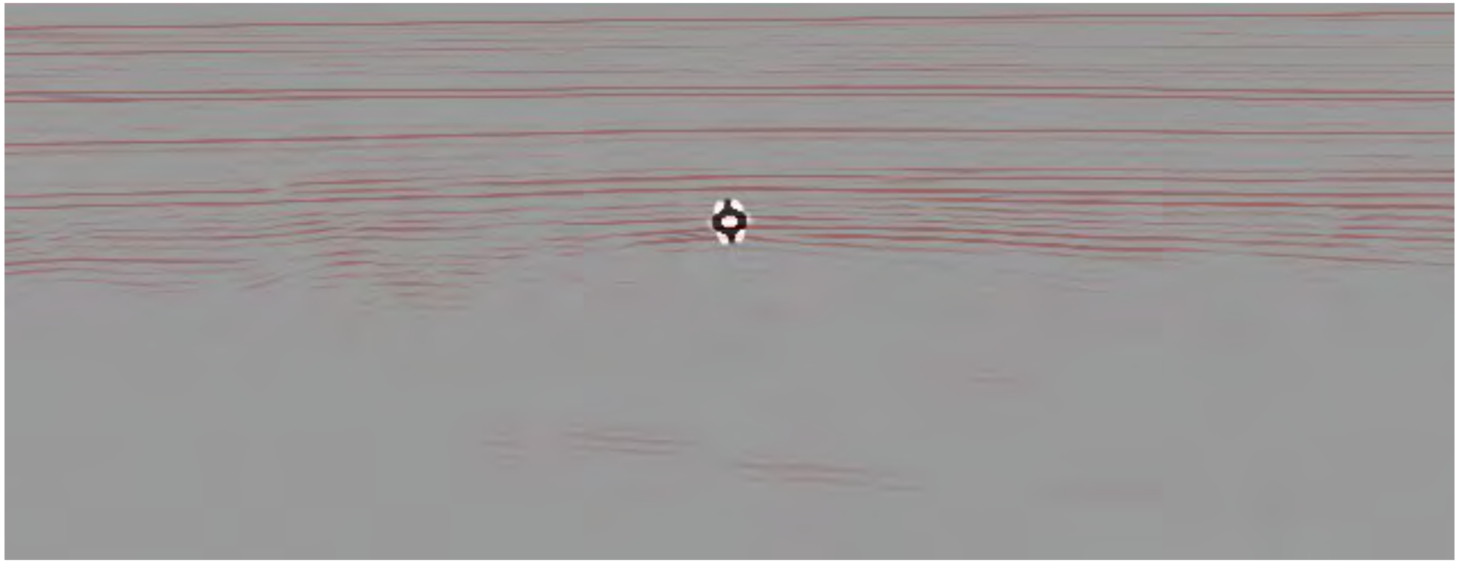

X (m)
9000
12000
15000
2500
3000
Depth (m)

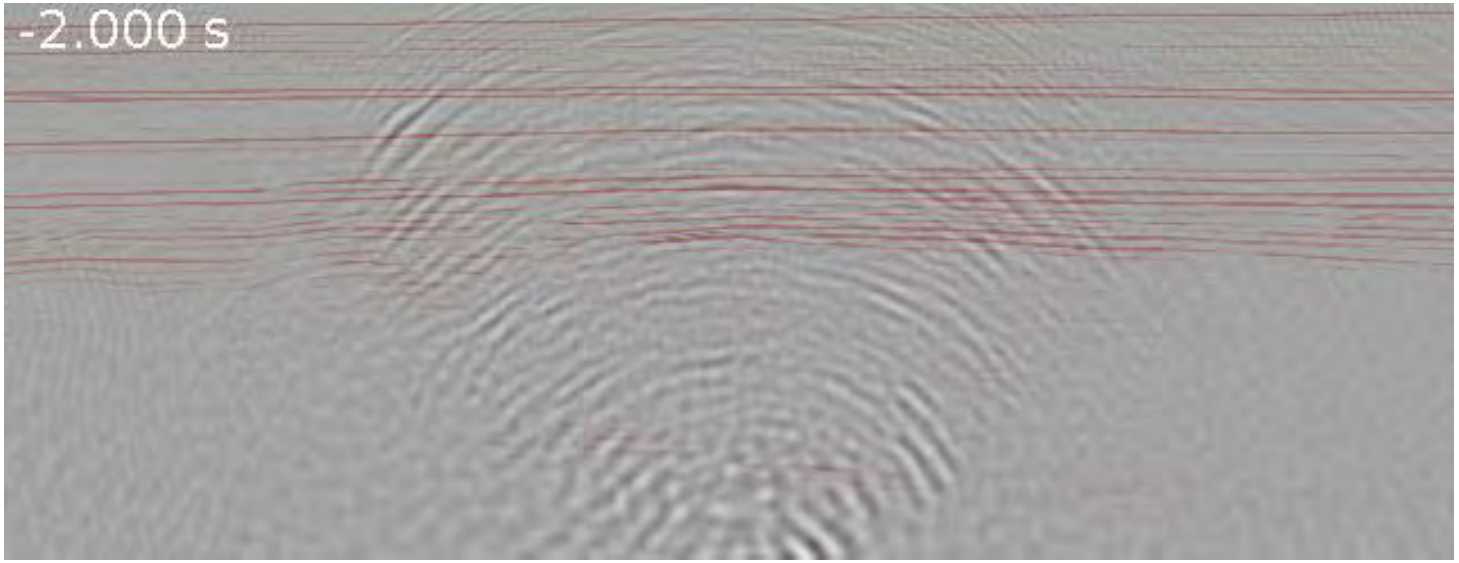
Homogeneous Green’s function

Reference: Wapenaar, K., Brackenhoff, J., Thorbecke, J., van der Neut, J., Slob, E., and Verschuur, E., 2018, *Virtual acoustics in inhomogeneous media with single-sided access*: Scientific Reports, Vol. 8, 2497..



Creation of virtual source *and* virtual receivers




-2.000 s