CRMN Method for Solving Time-Harmonic Wave Equation

Rafael Lago, Art Petrenko, Zhilong Fang, Felix Herrmann

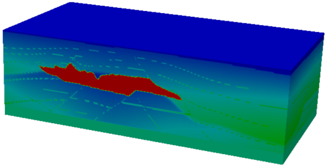
Copper Mountain, April 10, 2014

Frequency Domain Full Waveform Inversion Overview

Define **Full Waveform Inversion** as

$$\min_{m\in\mathcal{A}} \Phi(m) = \sum_{i}^{n_s} \left\| d^i - u^i \right\|_W .$$

for some norm W (usually L^2 with some regularization).



- d^i observed data
- u^i (approximated) computed data
- m Earth parameters; what we are trying to invert!

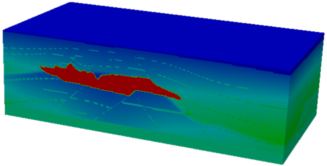
Full Waveform Inversion - Overview

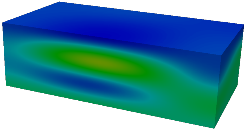
Define **Full Waveform Inversion** as

$$\min_{m\in\mathcal{A}} \Phi(m) = \sum_{i}^{n_s} \left\| d^i - u^i \right\|_W .$$

for some norm W (usually L^2 with some regularization).

- d^i observed data
- u^i (approximated) computed data
- m Earth parameters; what we are trying to invert!



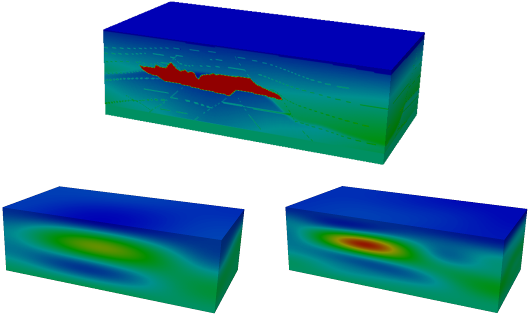


Full Waveform Inversion - Overview

Define **Full Waveform Inversion** as

$$\min_{m \in \mathcal{A}} \Phi(m) = \sum_{i}^{n_s} \left\| d^i - u^i \right\|_W .$$

for some norm W (usually L^2 with some regularization).



- d^i observed data
- u^i (approximated) computed data
- m Earth parameters; what we are trying to invert!

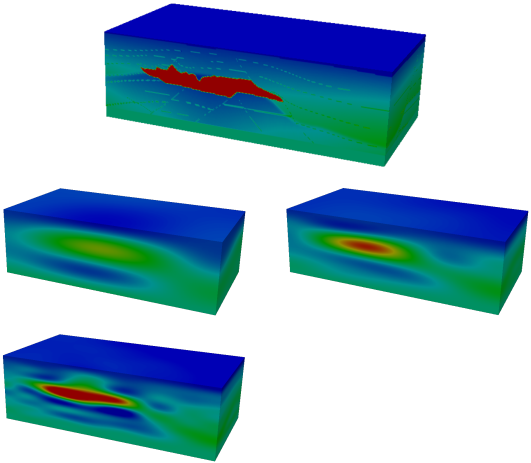
Full Waveform Inversion - Overview

Define **Full Waveform Inversion** as

$$\min_{m \in \mathcal{A}} \Phi(m) = \sum_{i}^{n_s} \left\| d^i - u^i \right\|_W .$$

for some norm W (usually L^2 with some regularization).

- d^i observed data
- u^i (approximated) computed data
- m Earth parameters; what we are trying to invert!

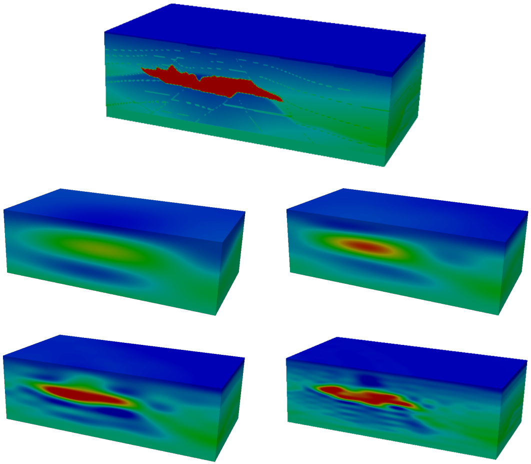


Define **Full Waveform Inversion** as

$$\min_{m\in\mathcal{A}} \Phi(m) = \sum_{i}^{n_s} \left\| d^i - u^i \right\|_W .$$

for some norm W (usually L^2 with some regularization).

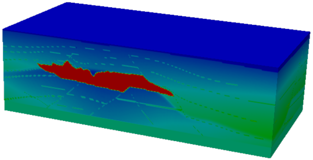
- d^i observed data
- u^i (approximated) computed data
- m Earth parameters; what we are trying to invert!



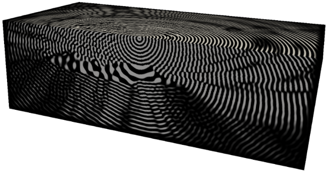
Forward Modelling - Overview

Define forward problem as

$$u^i = P_r A^{-1}(m) q^i$$



- $A(m)$ operator governing the physics of Earth
- q^i source
- m Earth parameters; what we are trying to invert!
- P_r restricts the computed data to the receivers



Backward Modelling - Overview

Define **backward problem** as

$$w^i = A^{-H}(m) P_r^H (d^i - u^i).$$

Then

$$\nabla\Phi(m) = \sum_i^{n_s} \mathcal{R}\left\{ (u^i)^T \left[\frac{\partial A}{\partial m}\right]^T w^i \right\}.$$

- $A(m)$ — operator governing the physics of Earth
- d^i — observed data
- u^i — simulated data
- m — Earth parameters; what we are trying to invert!
- $\nabla\Phi(m)$ — gradient of the reduced formulation of the inverse problem

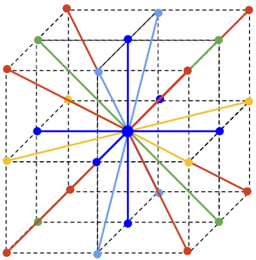
Frequency Domain Modelling - Overview

Isotropic acoustic wave progation is described by the **Helmholtz Equation**:

$$\Delta u - \left(\frac{2\pi f}{v}\right)^2 u = q.$$

We need **absorbing boundary conditions** to simulate infinite domain!!

- $v(x, y, z)$ velocity of propagation of the waves
- Yelds **very large** sparse matrices when discretized
- Unsymmetric, non-Hermitian - **no special property!**



27-points second-order stencil [Operto et al., 2007]

Frugal FWI Overview

Gradient-Descent with Errors

Let

$$\nabla\tilde{\Phi}(m_k) = \nabla\Phi(m_k) + \boldsymbol{e_k}$$

for some **error** e_k. Then, for **strongly convex** problems:

$$\Phi(m_k) - \Phi(m_*) < a_k(\Phi(m_0) - \Phi(m_*))$$

$$a_k = max\left\{c^k, \|\boldsymbol{e_k}\|_{\mathbf{2}}^{\mathbf{2}}\right\}, \qquad 0 \leq c \leq 1$$

where c is the condition number of the problem.

[Friedlander and Schmidt, 2012]

Frugal FWI - Source Subsampling

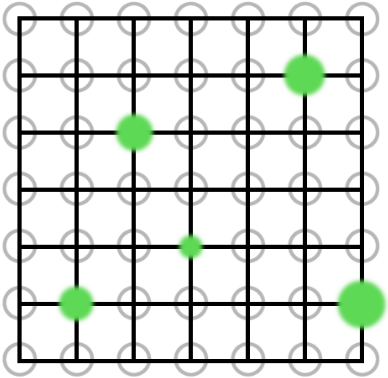
$$\tilde{\Phi}(m) = \sum_{i \in \mathcal{I}_k}^{b_k} \left\| d^i - u^i \right\|_W$$

$$\mathcal{I}_k \subset \{1, 2, ..., n_s\}, \mathcal{I}_k^{\#} = b_k$$

$\mathcal{I}_k$ is chosen at random *without replacement*. The expected error is given by

$$\|e_k\|_2 \propto \sqrt{\frac{1}{b_k} - \frac{1}{n_s}}$$

$$b_k \sim \min\left\{ \left(\epsilon^k + \frac{1}{n_s} \right)^{-1}, n_s \right\}.$$



[Friedlander and Schmidt, 2012] and [Herrmann et al., 2013]

Frugal FWI - Approximating u^i and w^i

Approximate cost function as

$$\left|\Phi(m, \alpha^k \epsilon) - \Phi(m, \boldsymbol{\alpha}^{\boldsymbol{k+1}}\boldsymbol{\epsilon})\right| \leq \boldsymbol{\eta}\Phi(m, \boldsymbol{\alpha}^{\boldsymbol{k+1}}\boldsymbol{\epsilon})$$

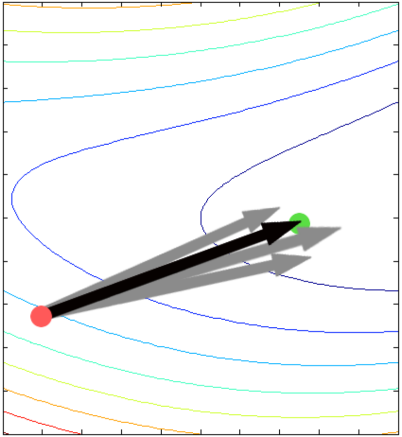
by approximating the forward problem as

$$u^i_{\boldsymbol{\varepsilon}} \approx P_r A^{-1}(m) q^i$$

(analogously for gradient computation)

Chosen Parameters

- $\alpha = 0.5$
- $\epsilon = 10^{-2}$
- $\boldsymbol{\eta} = 5 \times 10^{-2}$



[Herrmann et al., 2013]

CGMN, CRMN & Kaczmarz Sweep

Kaczmarz (Double) Sweep - Overview

$$u_{i+1} = u_i + \frac{\gamma(q_i - a_i^H u_i) a_i}{\|a_i\|_2^2}$$

- q_i ith element of q
- a_i ith row of A as a column vector
- γ relaxation parameter $\in (0, 2)$

Kaczmarz sweeps **guarantees convergence** in a finite (possibly **large**) number of steps.

Kaczmarz (Double) Sweep - Overview

$$Q = Q_1 Q_2 \ldots Q_N Q_N Q_{N-1} \ldots Q_1 \qquad Q_i = I - \frac{\gamma}{\|a_i\|_2^2} a_i a_i^H$$

$$u := Qu + Rq \qquad \Longrightarrow \qquad (I - Q)u = Rq$$

- Q is symmetric positive definite
- We can use **CG** to solve this system
- Neither Q nor R need to be computed in practice
- Equivalent to using **CG on the normal equations**, preconditioned by SSOR

1 (CGMN).

$p_0 = r_0 = dkswp(A, u_0, b, \gamma) - u_0;$
while *not converged* **do**
  $q_k = p_k - dkswp(A, p_k, 0, \gamma);$
  $\alpha_k = \langle r_k, r_k \rangle \,/\, \langle p_k, q_k \rangle;$
  $u_{k+1} = u_k + \alpha_k r_k;$
  $r_{k+1} = r_k - \alpha_k q_k;$
  $\beta_k = \langle r_{k+1}, r_{k+1} \rangle \,/\, \langle r_k, r_k \rangle;$
  $p_{k+1} = r_k + \beta_k p_k;$
  $k = k + 1;$
end while

- Very low memory cost
- Very simple implementation
- Suitable for any matrix A (even nonsquare)

On CG:[Hestenes and Stiefel, 1952], on CGMN: [Björck and Elfving, 1979]

2 (CRMN).

while *not converged* **do**

$Ar_k := r_k - dkswp(A, r_k, 0, \gamma)$;

$\beta_k = \langle r_k, Ar_k \rangle \,/\, \langle r_{k-1}, Ar_{k-1} \rangle$;

$p_k = r_k + \beta_k p_{k-1}$;

$Ap_k = Ar_k + \beta_k Ap_{k-1}$;

$\alpha_k = \langle r_k, Ar_k \rangle \,/\, \langle Ap_k, Ap_k \rangle$;

$u_{k+1} = u_k + \alpha_k r_k$;

$r_{k+1} = r_k - \alpha_k q_k$;

$k = k + 1$;

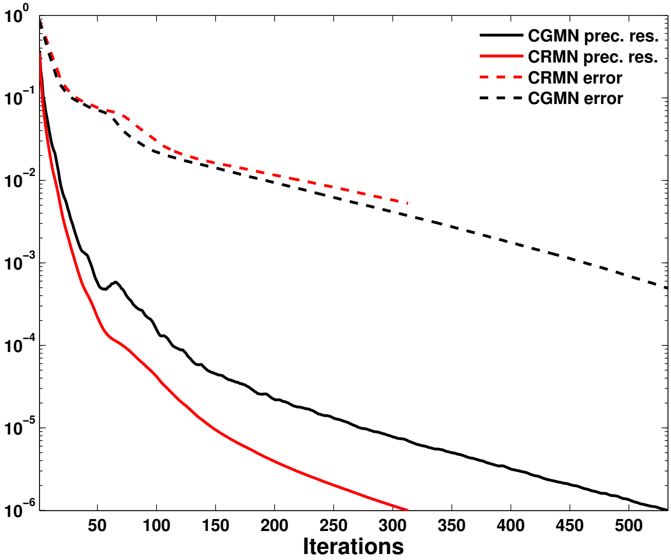
end while

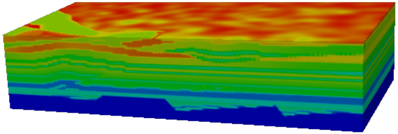
- Very **low memory cost**
- One **extra** vector storage
- One **extra** inner product
- **Minimal residual** properties

On CR: [Stiefel, 1955], comparisons:[Fong and Saunders, 2012, Eiermann and Ernst, 2001]

Numerical Experiment

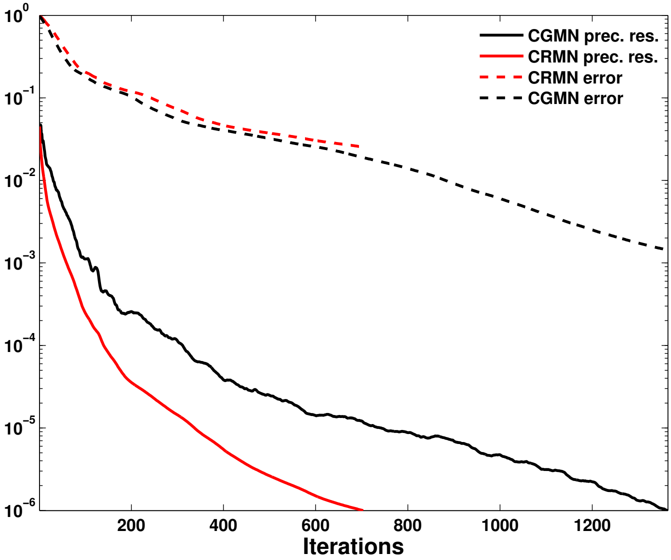
Forward Modeling - SEG/EAGE Overthrust

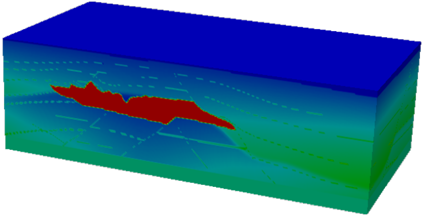




- $20.1 \times 20.1 \times 4.7\ km^3$
- $100m$ grid spacing
- $\mathcal{O}(1.9 \times 10^6)$ points
- 3Hz, $n_\lambda = 7.2$
- $v_{min} = 2179m/s$,
- $v_{max} = 6000m/s$
- PML: 15 points

Forward Modeling - SEG/EAGE Salt Dome





- $4 \times 4 \times 1.2\ km^3$
- $20m$ grid spacing
- $\mathcal{O}(2.5 \times 10^6)$ points
- 3Hz, $n_\lambda = 22.7$
- $v_{min} = 1365m/s$
- $v_{max} = 4991m/s$
- PML: 15 points

The True Stopping Criterion

CGMN/CRMN stops when $\frac{\|r_j\|_2}{\|r_0\|_2} < \alpha^{k+1}\epsilon$

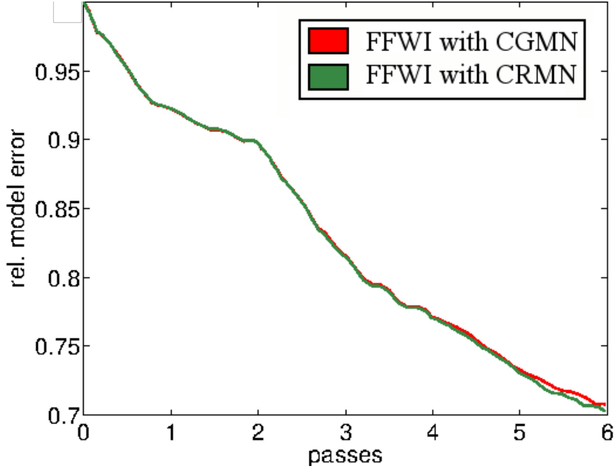
For k satisfying

$$\left|\Phi(m, \alpha^k \epsilon) - \Phi(m, \alpha^{k+1}\epsilon)\right| \leq \eta \Phi(m, \alpha^{k+1}\epsilon)$$

Chosen Parameters

- $\alpha = 0.5$
- $\epsilon = 10^{-2}$
- $\eta = 5 \times 10^{-2}$

FFWI-CGMN vs. FFWI-CRMN - Overthrust



	FFWI with CGMN	FFWI with CRMN	Speedup
$4\ Hz$	23,403	19,846	18%
$6\ Hz$	30,189	24,387	24%
$8\ Hz$	34,724	26,265	32%
Total	88,316	70,498	25%

Table: *Total number of iterations of CGMN and CRMN during the inversion for each frequency slice*

Conclusions & Future Work

Conclusions & Future Work

- Smaller error computed by CGMN does not bring any improvement to Frugal FWI
- CRMN seems to be a feasible option
- Does the performance gain of CRMN grow with the frequency?
- Does the same result hold for other kind of PDEs?
- Does this behaviour holds for other models? (no, not always!)

Questions?

Björck, Å. and Elfving, T. (1979).
Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations.
BIT Numerical Mathematics, 19(2):145–163.

Eiermann, M. and Ernst, O. G. (2001).
Geometric aspects in the theory of Krylov subspace methods.
Acta Numerica, 10(10):251–312.

Fong, D. and Saunders, M. (2012).
CG versus MINRES: An empirical comparison.
SQU Journal for Science, 17:1:44–62.

Friedlander, M. P. and Schmidt, M. (2012).
Hybrid deterministic-stochastic methods for data fitting.
SIAM Journal on Scientific Computing, 34(3):A1380–A1405.

Gordon, D. and Gordon, R. (2005).

Component-averaged row projections: A robust, block-parallel scheme for sparse linear systems.
SIAM J. Scientific Computing, 27(3):1092–1117.

Gordon, D. and Gordon, R. (2010).
CARP-CG: A robust and efficient parallel solver for linear systems, applied to strongly convection dominated pdes.
Parallel Computing, 36(9):495–515.

Gordon, D. and Gordon, R. (2012).
Parallel solution of high frequency Helmholtz equations using high order finite difference schemes.
Applied Mathematics and Computation, 218(21):10737–10754.

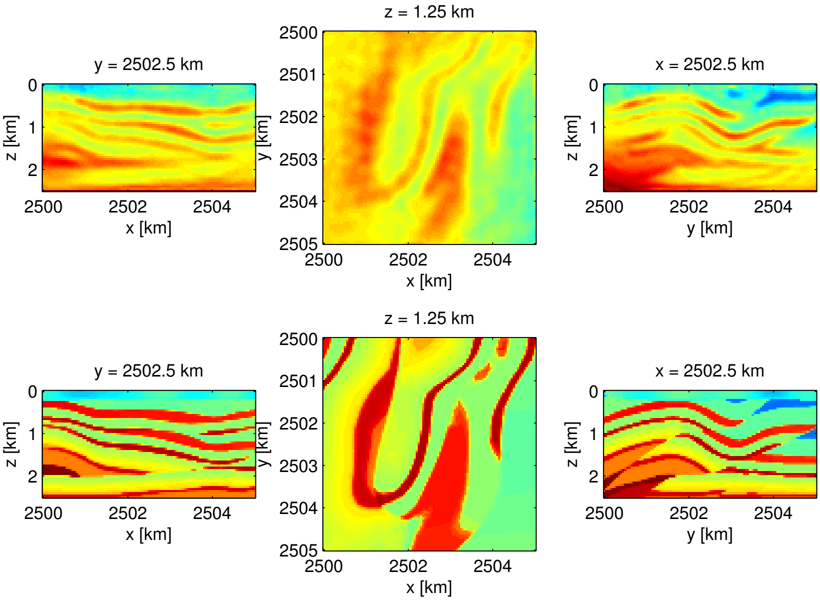
Herrmann, F. J., Calvert, A. J., Hanlon, I., Javanmehri, M., Kumar, R., van Leeuwen, T., Li, X., Smithyman, B., Takougang, E. T., and Wason, H. (2013).
Frugal full-waveform inversion: from theory to a practical algorithm.
The Leading Edge, 32(9):1082–1092.

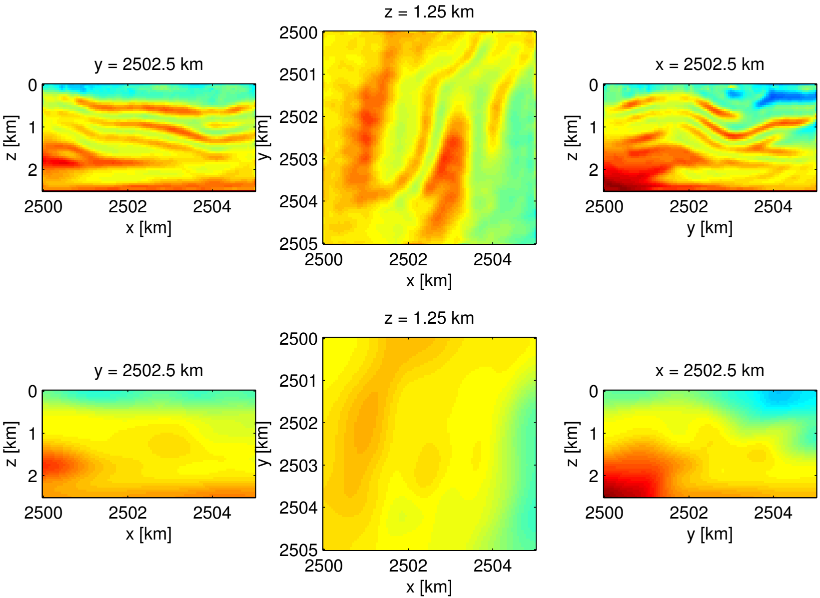
Hestenes, M. R. and Stiefel, E. (1952).

Methods of conjugate gradients for solving linear systems.
Journal of Research of the National Bureau of Standards, 49(6):409–436.

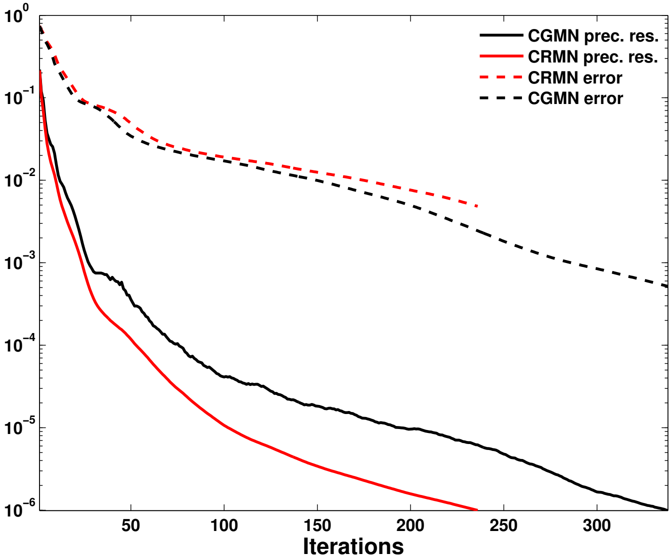
Operto, S., Virieux, J., Amestoy, P., L'Excellent, J.-Y., Giraud, L., and Ali, H. B. H. (2007).
3d finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study.
Geophysics, 72(5):SM195–SM211.

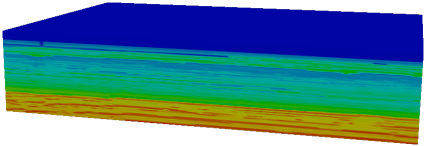
Stiefel, E. (1955).
Relaxationsmethoden bester strategie zur losung linearer gleichungssystem.
Commentarii Mathematici Helvetici, 29(1):157–179.


y = 2502.5 km
z [km]
x [km]
z = 1.25 km
y [km]
x [km]
x = 2502.5 km
z [km]
y [km]
y = 2502.5 km
z [km]
x [km]
z = 1.25 km
y [km]
x [km]
x = 2502.5 km
z [km]
y [km]


y = 2502.5 km
z [km]
0
1
2
2500
2502
2504
x [km]
z = 1.25 km
y [km]
2500
2501
2502
2503
2504
2505
2500
2502
2504
x [km]
x = 2502.5 km
z [km]
0
1
2
2500
2502
2504
y [km]
y = 2502.5 km
z [km]
0
1
2
2500
2502
2504
x [km]
z = 1.25 km
y [km]
2500
2501
2502
2503
2504
2505
2500
2502
2504
x [km]
x = 2502.5 km
z [km]
0
1
2
2500
2502
2504
y [km]

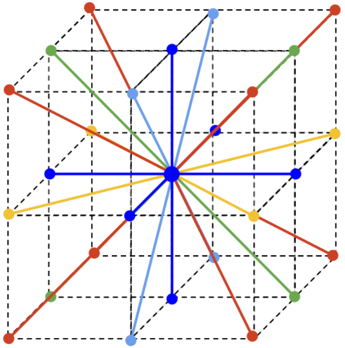
Forward Modeling - BG Compass



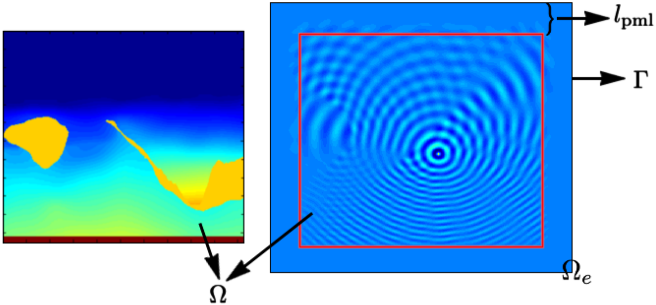


- $10 \times 10 \times 2\ km^3$
- $75m$ grid spacing
- $\mathcal{O}(4.9 \times 10^5)$ points
- 3Hz, $n_\lambda = 6.3$
- $v_{min} = 1420m/s$
- $v_{max} = 4650m/s$
- PML: 15 points

Frequency Modelling - Finite Difference Discretization



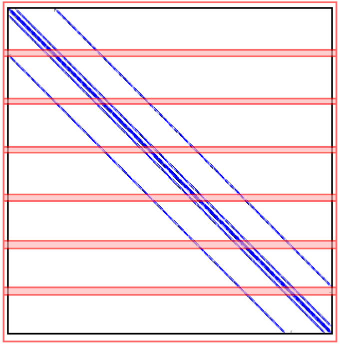
27-points parsimonious staggered second-order stencil



Perfectly Matched Layer (PML) boundary condition

[Operto et al., 2007]

Parallelization - Component Averaging Row Projection



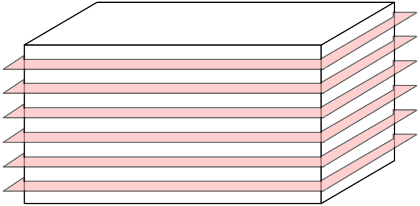


Figure: *Illustration of the parallelization of CGMN (respectively CRMN), resulting on CARPCG method (respectively CARPCR)*

[Gordon and Gordon, 2005][Gordon and Gordon, 2010][Gordon and Gordon, 2012]