Time–lapse Full–Waveform Inversion using Hamiltonian Monte Carlo: A proof of concept

Maria Kotsi[1], Alison Malcolm[1], Gregory Ely[2]

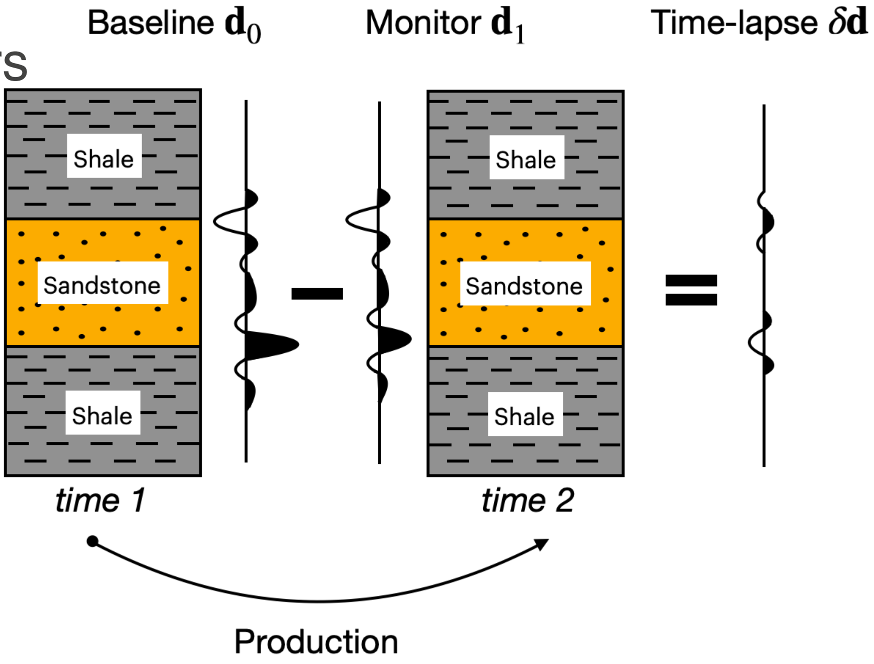
[1] Memorial University of Newfoundland, CA

[2] Massachusetts Institute of Technology, USA

mkotsi@mun.ca

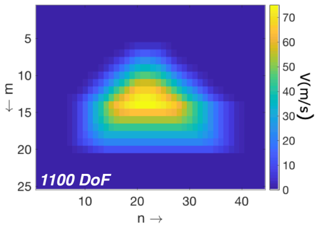
Time–Lapse Seismic Imaging

- Over a reservoir's life its parameters will change
- Time–lapse or 4D seismic: same location but different time
- Full–Waveform Inversion (FWI): delivers a velocity model of the subsurface



Bayesian Time–Lapse Inversion

$\delta\mathbf{m}$: Time-lapse velocity model

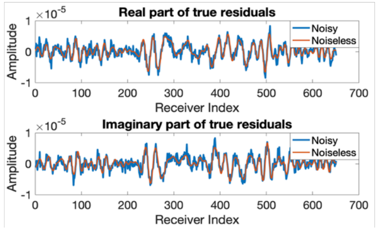


Forward model

$F(\delta\mathbf{m})$

$n \sim \mathcal{N}(0,\Sigma)$

$\delta\mathbf{d}$: Observed data residual



Goal: estimate distribution of time-lapse models given data residuals $p(\delta m \mid \delta d)$

Challenges:

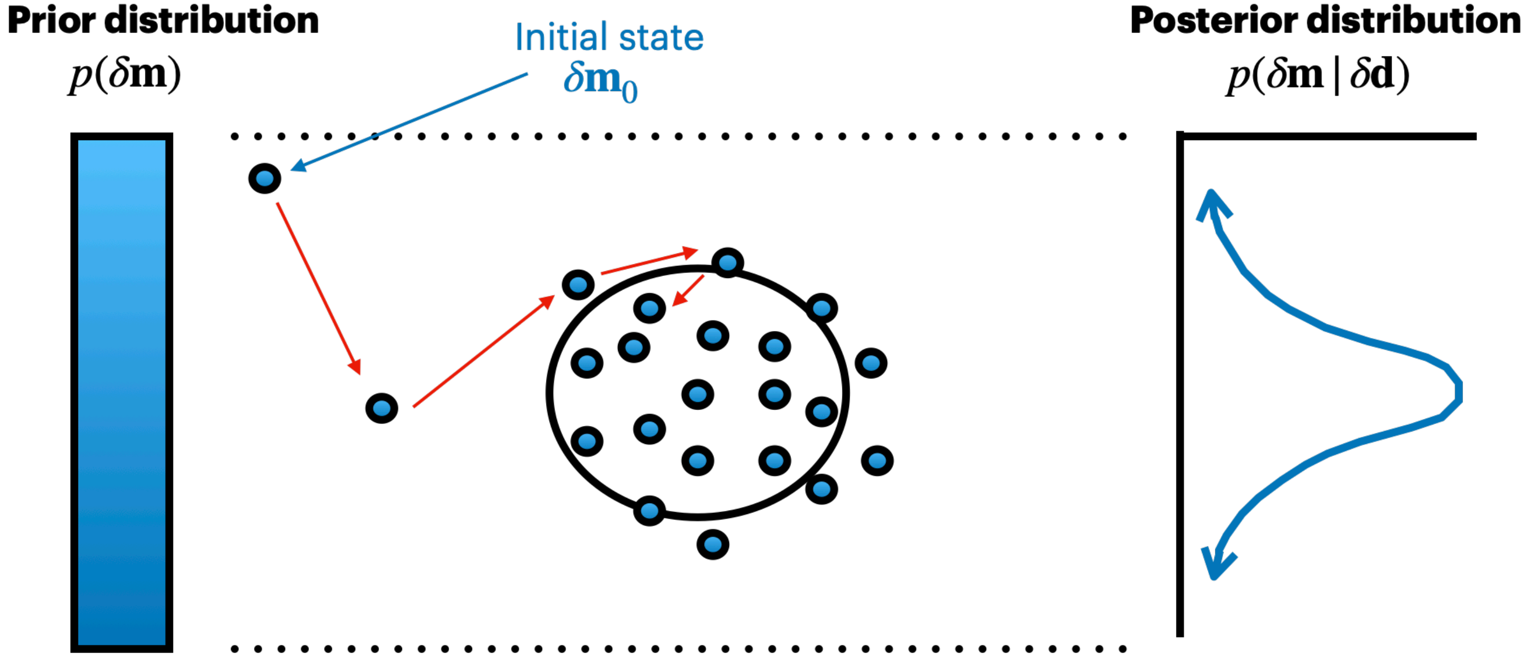
- Large dimensionality (visualization, covariance)
- Expensive forward solvers (finite difference, finite element)
- Non-linear forward model (multi-modal or non-Gaussian model distribution)

Solution: Markov Chain Monte Carlo with a fast forward solver

Outline

- Motivation
- Markov Chain Monte Carlo
- Hamiltonian Monte Carlo
- Local solver
- Numerical Example
- Summary

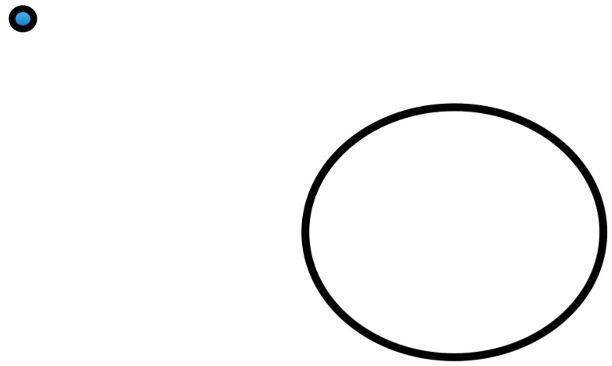
Markov Chain Monte Carlo



Hamiltonian Monte Carlo

HMC treats a model as virtual Hamiltonian particle that moves along a trajectory.

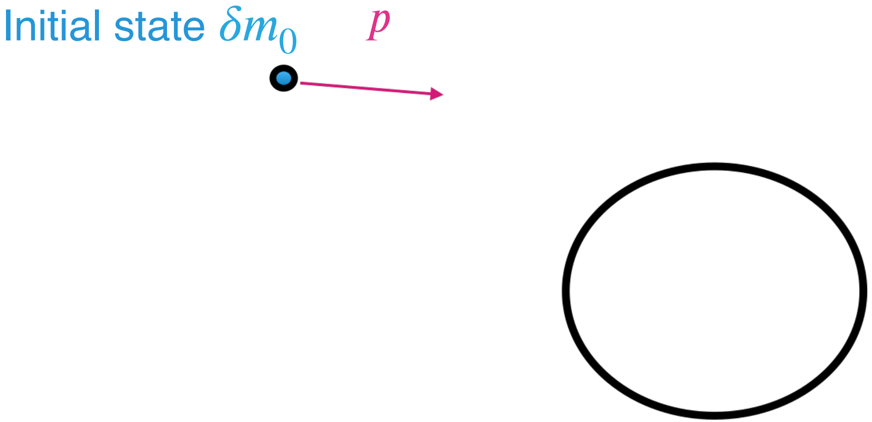
Initial state δm_0



1. Initial model
   - Associated potential energy $U(\delta m)$

Hamiltonian Monte Carlo

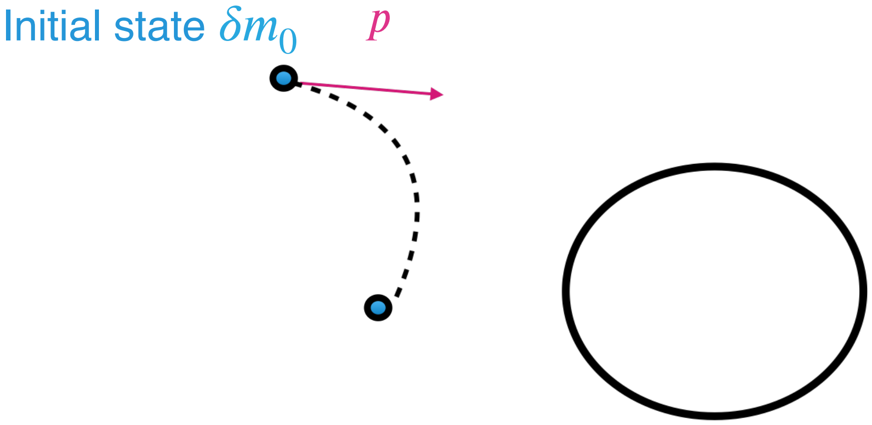
HMC treats a model as virtual Hamiltonian particle that moves along a trajectory.



1. Initial model
    - Associated potential energy $U(\delta m)$
2. Add auxiliary momentum
    - Associated kinetic energy $K(p)$

Hamiltonian Monte Carlo

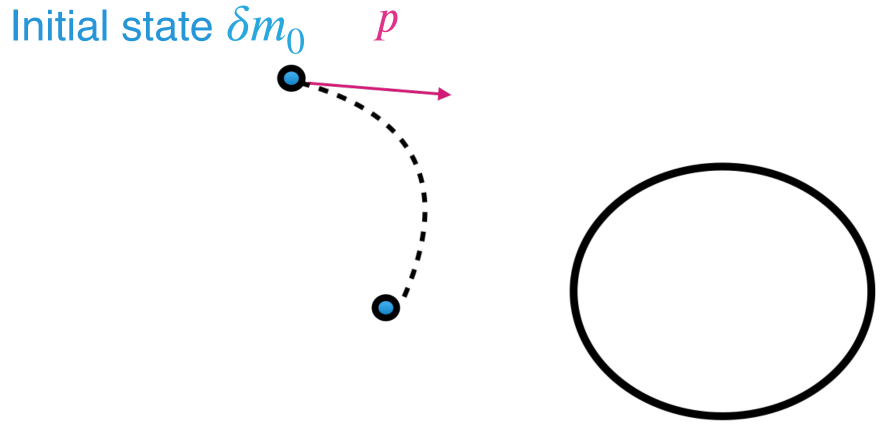
HMC treats a model as virtual Hamiltonian particle that moves along a trajectory.



1. Initial model
   - Associated potential energy $U(\delta m)$
2. Add auxiliary momentum
   - Associated kinetic energy $K(p)$
3. Solve Hamilton's equation
   - $H\left(\delta m, p\right) = U(\delta m) + K(p)$
4. Get a new proposed model

Hamiltonian Monte Carlo

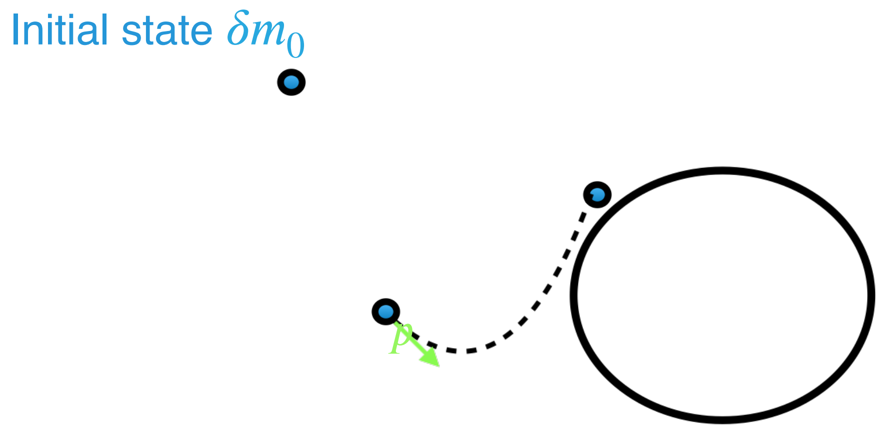
HMC treats a model as virtual Hamiltonian particle that moves along a trajectory.



1. Initial model
   - Associated potential energy $U(\delta m)$
2. Add auxiliary momentum
   - Associated kinetic energy $K(p)$
3. Solve Hamilton's equation
   - $H\left(\delta m, p\right) = U(\delta m) + K(p)$
4. Get a new proposed model
5. Accept/Reject based on Metropolis criterion

Hamiltonian Monte Carlo

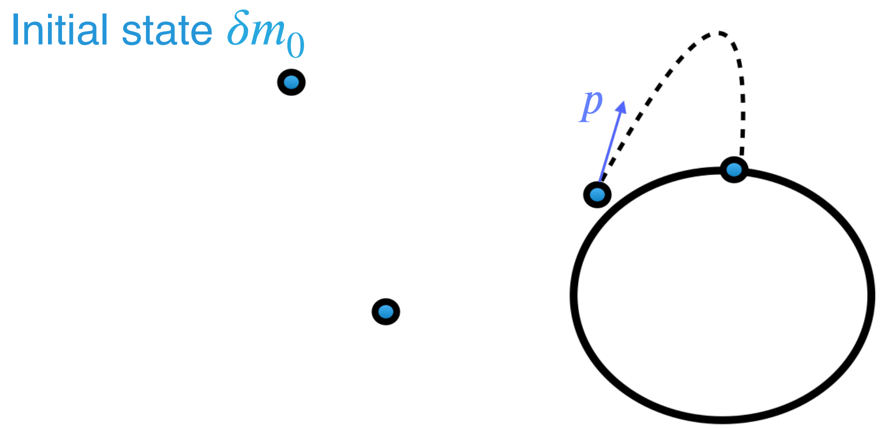
HMC treats a model as virtual Hamiltonian particle that moves along a trajectory.



1. Initial model
   - Associated potential energy $U(\delta m)$
2. Add auxiliary momentum
   - Associated kinetic energy $K(p)$
3. Solve Hamilton's equation
   - $H\left(\delta m, p\right) = U(\delta m) + K(p)$
4. Get a new proposed model
5. Accept/Reject based on Metropolis criterion
6. Move on

Hamiltonian Monte Carlo

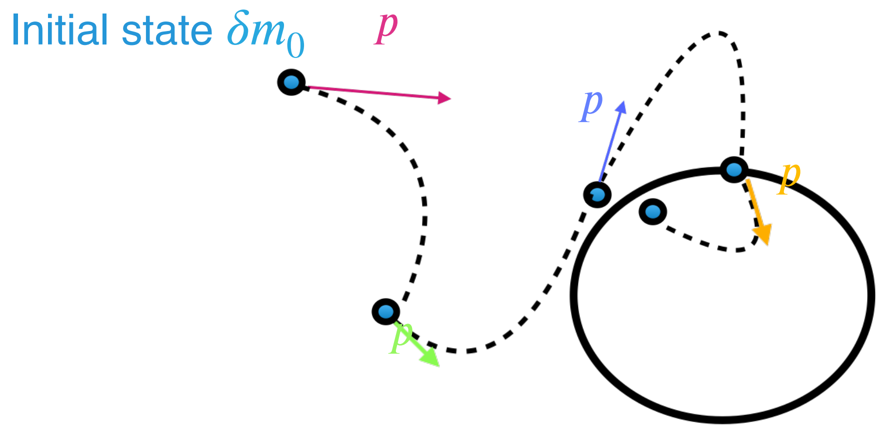
HMC treats a model as virtual Hamiltonian particle that moves along a trajectory.



1. Initial model
   - Associated potential energy $U(\delta m)$
2. Add auxiliary momentum
   - Associated kinetic energy $K(p)$
3. Solve Hamilton's equation
   - $H\left(\delta m, p\right) = U(\delta m) + K(p)$
4. Get a new proposed model
5. Accept/Reject based on Metropolis criterion
6. Move on

Hamiltonian Monte Carlo

HMC treats a model as virtual Hamiltonian particle that moves along a trajectory.

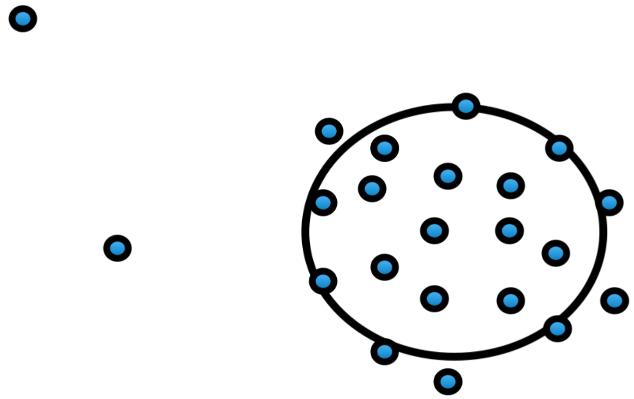


1. Initial model
   - Associated potential energy $U(\delta m)$
2. Add auxiliary momentum
   - Associated kinetic energy $K(p)$
3. Solve Hamilton's equation
   - $H\left(\delta m, p\right) = U(\delta m) + K(p)$
4. Get a new proposed model
5. Accept/Reject based on Metropolis criterion
6. Move on

Hamiltonian Monte Carlo

HMC treats a model as virtual Hamiltonian particle that moves along a trajectory.

Initial state δm_0



1. Initial model
   - Associated potential energy $U(\delta m)$
2. Add auxiliary momentum
   - Associated kinetic energy $K(p)$
3. Solve Hamilton's equation
   - $H\left(\delta m, p\right) = U(\delta m) + K(p)$
4. Get a new proposed model
5. Accept/Reject based on Metropolis criterion
6. Move on

Hamiltonian Monte Carlo

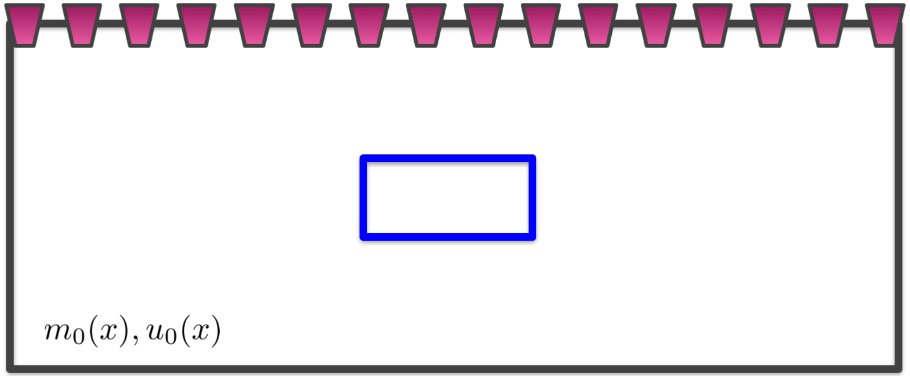
	Metropolis Hastings	Hamiltonian Monte Carlo
DoF	$1 - 10^2$	$10^3 - 10^6$
Iterations	$10^4 - 10^6$	$10^3 - 10^5$
Computational cost	Low	High
Pixel-by-pixel UQ	No	Yes
Gradient calculation	No	Yes

- Metropolis Hastings algorithms are used to ground truth probability distribution.
  - Very slow to converge as the number of dimensions grows
  - Requires reduced parameterization techniques
- Hamiltonian Monte Carlo is more advanced
  - Use the geometry of the target distribution for faster exploration

Outline

- Motivation
- Markov Chain Monte Carlo
- Hamiltonian Monte Carlo
- Local solver
- Numerical Example
- Summary

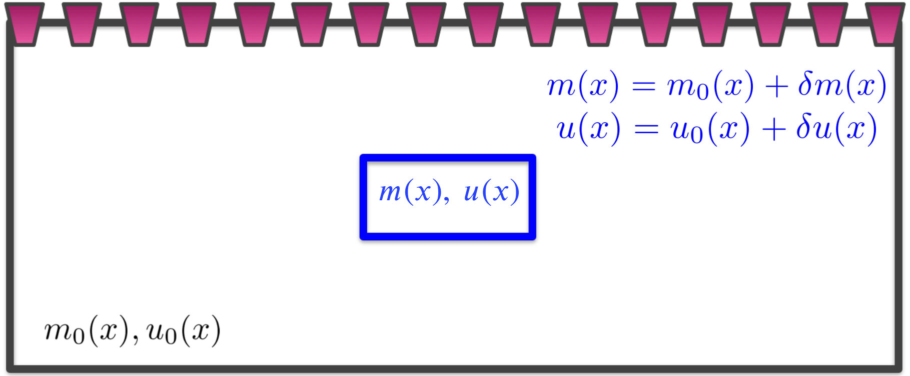
Local Acoustic Solver



1) **Split model to subdomain and exterior.** Model in exterior & initial guess in local domain = background model m_0

Willemsen et al. (2016); Malcolm & Willemsen (2017)

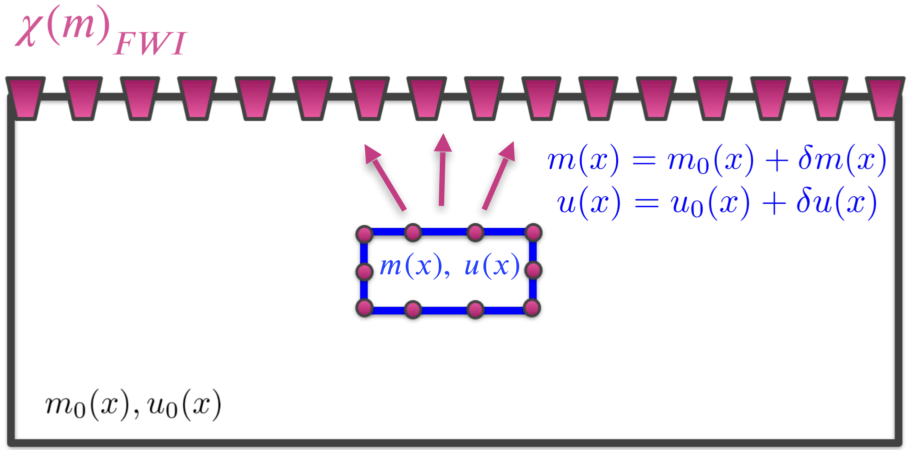
Local Acoustic Solver



1) **Split model to subdomain and exterior.** Model in exterior & initial guess in local domain = background model m_0
2) **Perturbation δm exists only in local domain**
   Get model m and wavefield u

Willemsen et al. (2016); Malcolm & Willemsen (2017)

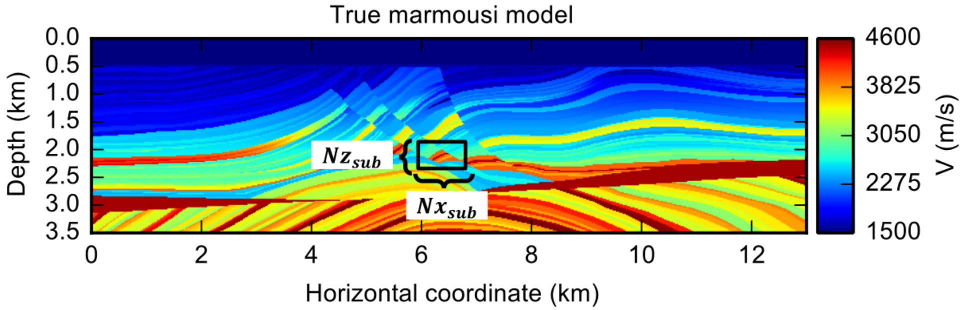
Local Acoustic Solver



1) **Split model to subdomain and exterior.** Model in exterior & initial guess in local domain = background model m_0
2) **Perturbation δm exists only in local domain**
   Get model m and wavefield u
3) **Propagate u to the receivers**

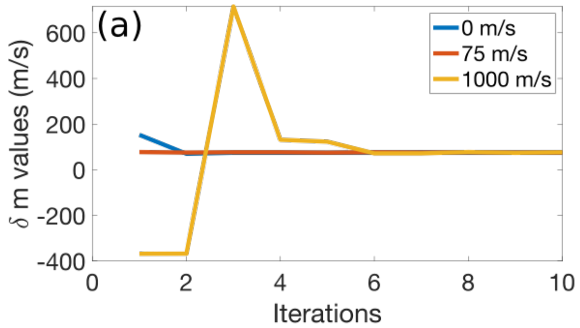
Willemsen et al. (2016); Malcolm & Willemsen (2017)

Simple Numerical Illustration

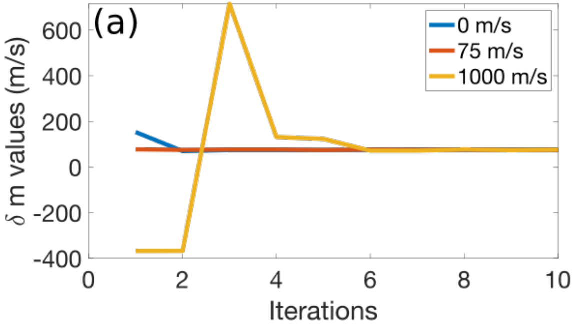


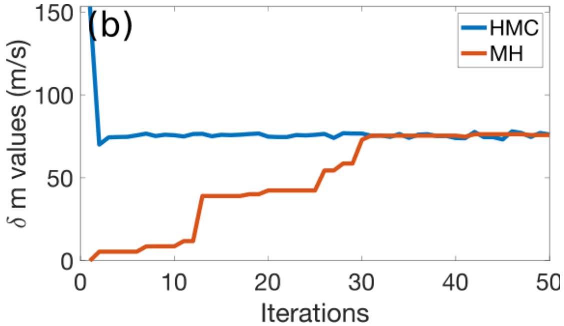
- Marmousi: background baseline model
- Time lapse change : $\delta m = 75\ m/s$
- 1 shot, 651 receivers
- Single frequency of 8 Hz
- Full domain grid points: Nz*Nx = 113925
- Local domain grid points: Nz_{sub}*Nx_{sub} = 1100

Simple Numerical Illustration

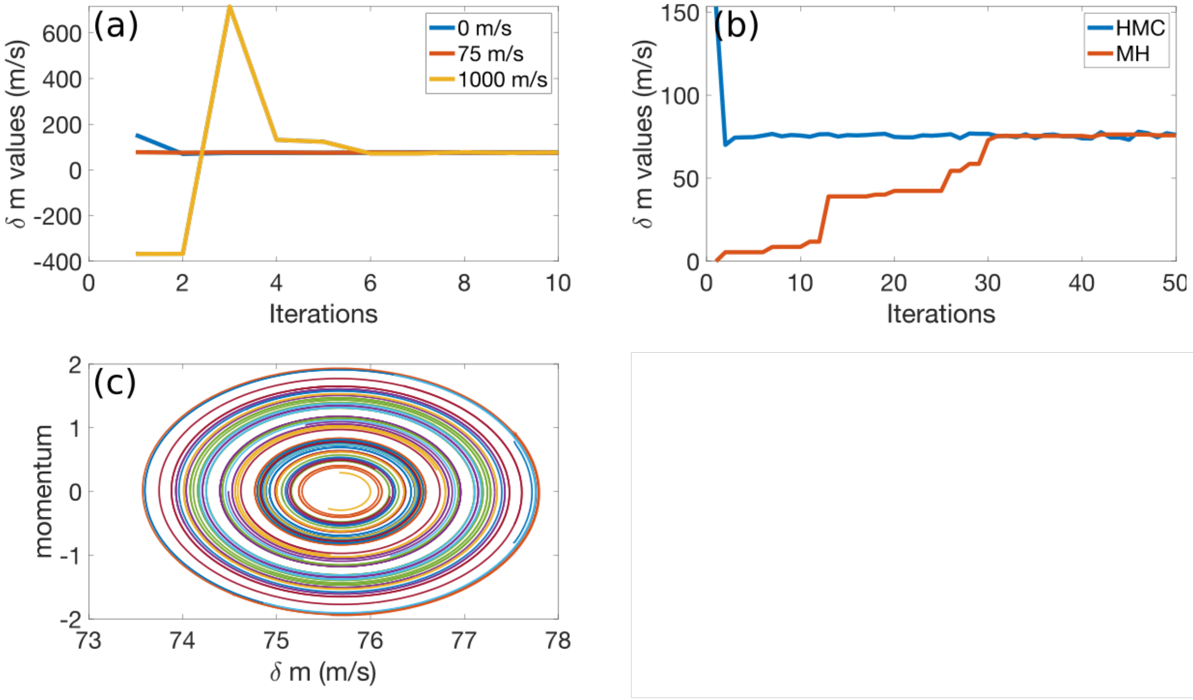


Simple Numerical Illustration

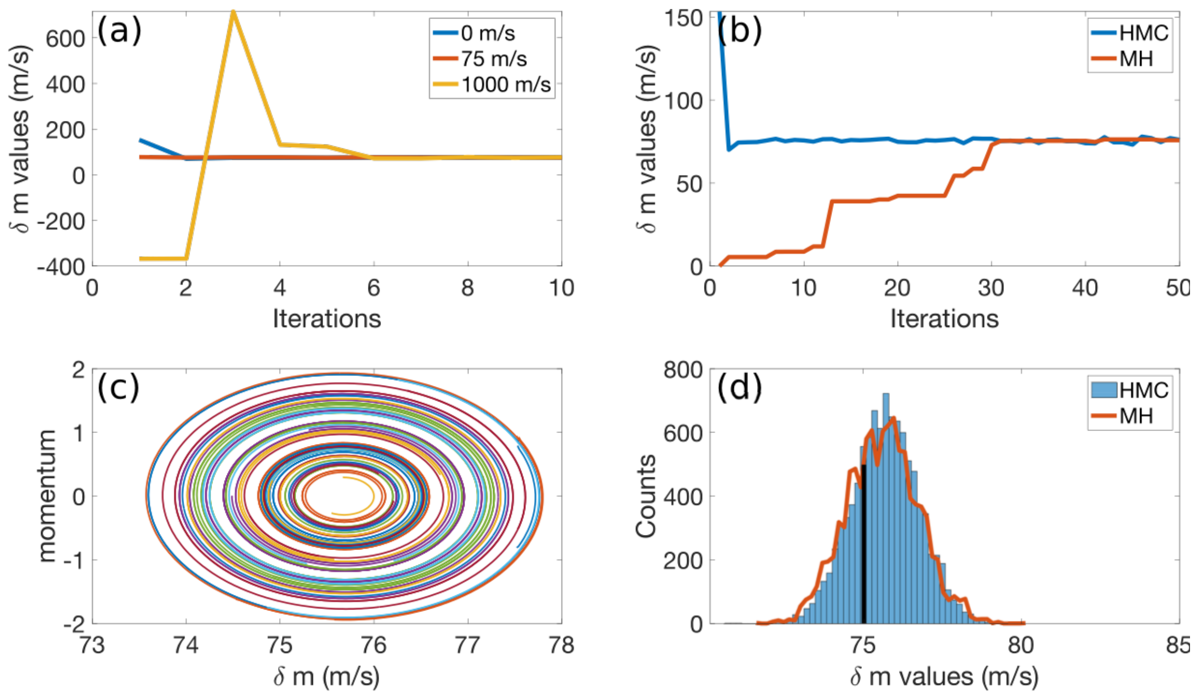




Simple Numerical Illustration

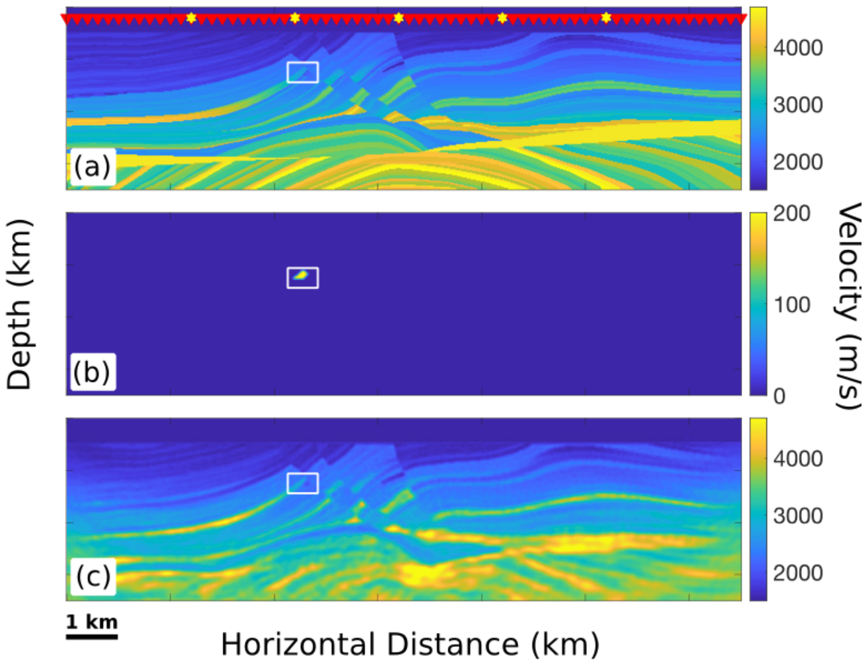


Simple Numerical Illustration



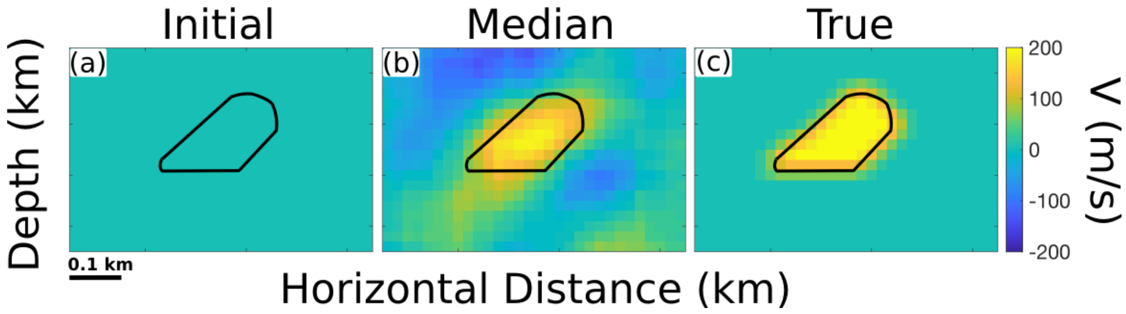
SEG20

Complex Numerical Illustration



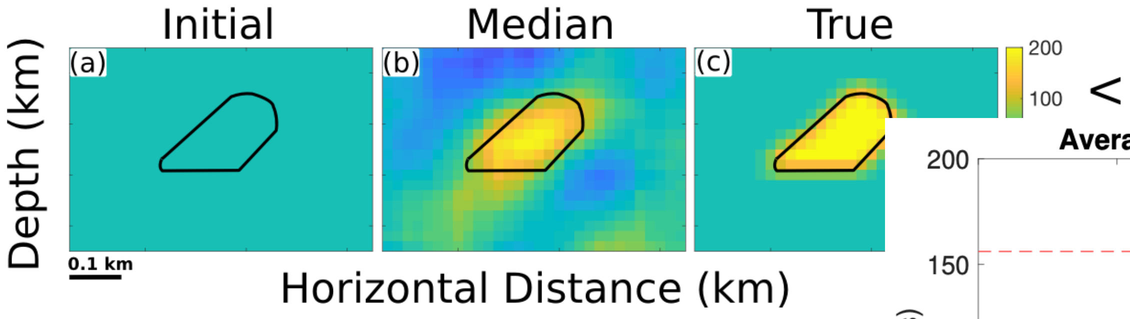
- Setup (a & b)
  - ➢ $\delta m = 200\ m/s$
  - ➢ 600 DoF
  - ➢ 5 shots, 651 receivers
  - ➢ Single frequency: 5 Hz
- Background model (c)
  - ➢ Inverted baseline
  - ➢ 64 shots, 651 receivers
  - ➢ 3, 4, 5, 6.5, 8, 10 Hz sequentially for 15 iterations per frequency.

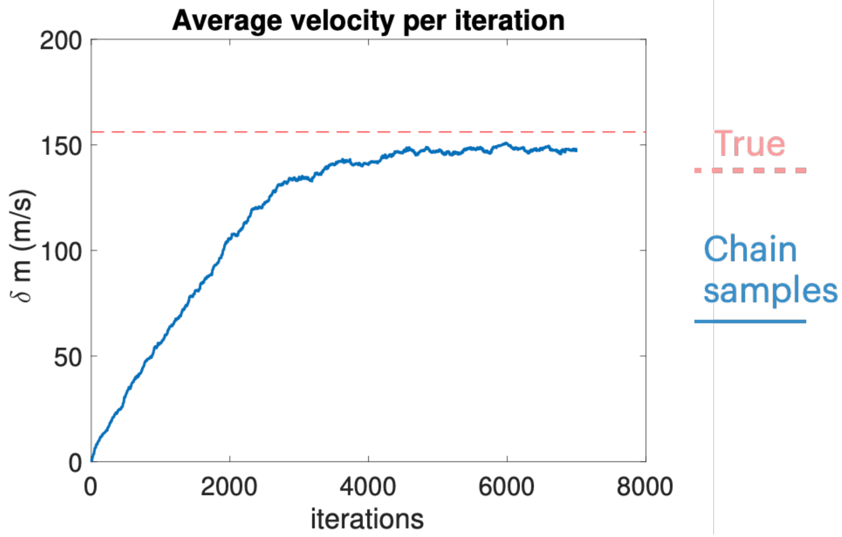
Complex Numerical Illustration



- One chain with 7,000 samples
- Median model:
  - Good approximation of shape
  - Magnitude underestimated

Complex Numerical Illustration





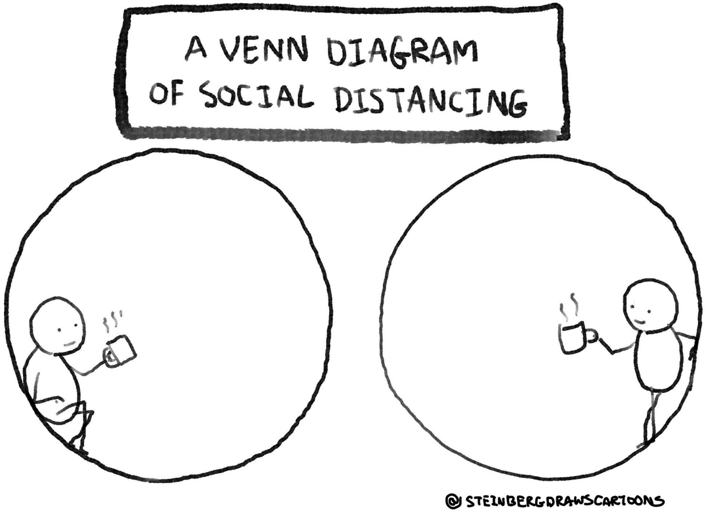
- One chain with 7,000 samples
- Median model:
  - Good approximation of shape
  - Magnitude underestimated

SEG20

Summary

- We generated a framework for uncertainty quantification in a targeted way
- Use of local solver for fast gradient calculations
  - Computational savings are $\frac{t_{local}}{t_{full}} = 0.03$
- Proof of concept of HMC on time-lapse scenario
  - Superior performance when compared to MH
- Future work:
  - Use of Mass Matrix in the kinetic energy formulation
  - Multiple frequencies

THANK YOU!
ΕΥΧΑΡΙΣΤΩ!



https://twitter.com/steinbergart/status/1239253552461164544/photo/1

mkotsi@mun.ca

SEG20