Enabling wave-based inversion on GPUs with randomized trace estimation

Mathias Louboutin and Felix J. Herrmann*

June 9 2022

SLIM
Georgia Institute of Technology

ML4Seismic

Released to public domain under Creative Commons license type BY (https://creativecommons.org/licenses/by/4.0)
Copyright (c) 2021, Mathias Louboutin (Georgia Tech)
Motivation

High-memory footprint adjoint-state methods

Computationally expensive checkpointing

Case specific/internal solutions to manage memory

- Fourier (BP patent)
- Compression (No existing GPU porting)
- Serialization/Disk (High IO)
- Boundary methods (reversible only)
- ...

Solutions

Take advantage of large-scale randomized linear algebra

Leverage our work on full-subsurface offset Image Volumes

Build on lessons learned form machine learning (convolutional layers)

According to stochastic optimization

- inaccurate gradients can still lead to accurate inversion
- undergirds our compressive imaging & randomized FWI & WRI
Randomized linear algebra

Randomized SVD:

\[ A \approx U S V^T \quad \text{with} \quad \begin{bmatrix} Q, \sim \\ \tilde{U}, S, V \end{bmatrix} = qr(AZ) \]
\[ U = Q\tilde{U} = \text{svd}(Q^TA) \]

- information is reaped during random probing \( AZ \) w/ \( Z = [z_1, \ldots, z_r] \)
- only need access to action of \( A \) (in parallel)
- memory friendly
- unbiased estimator when \( \mathbb{E}(zz^T) = I \) w/ accuracy \( \propto r \ll N \), # of sketches w/ random vectors \( z_i \)
Random Trace Estimation

\[
\text{tr}(A) \approx \frac{1}{r} \sum_{j=1}^{r} z_j^\top A z_j = \frac{1}{r} \text{tr}(Z^\top A Z)
\]
Randomized linear algebra

Randomized Trace Estimation:

\[ \text{tr}(A) \approx \frac{1}{r} \sum_{j=1}^{r} z_j^T A z_j = \frac{1}{r} \text{tr}(Z^T A Z) \]

- only needs matrix-free access to actions of \( A \)
- unbiased estimator when \( \mathbb{E}(zz^T) = I \) w/ accuracy \( \propto r \ll N \), # of sketches w/ random vectors \( z_j \)
- errors studied & understood

Why should we care?
Adjoint state gradient

FWI objective

\[ \Phi(m) = \frac{1}{2} \| P_r A^{-1}(m) P_s^T q - d \|_2^2 \]

with gradient with respect to \( m \)

\[ \delta m(x) = \sum_{t=1}^{n_t} \dot{u}[t, x] v[t, x] \]
Randomized trace estimation

Approximate FWI gradient calculation for $r \ll n_t$:

$$\delta m[x] = \text{tr} \left( \ddot{u}[t, x]v[t, x]^\top \right) \approx \frac{1}{r} \text{tr} \left( (Z^\top \ddot{u}[x])(v[x]^\top Z) \right)$$

- $\ddot{u}$ second time derivative solution forward wave equation
- $v$ solution adjoint wave equation
- $\sum x_i y_i = x^\top y = \text{tr}(xy^\top)$
- probing vectors $Z = [z_1 \cdots z_r]$ with $\mathbb{E}(z_i^\top z_i) = 1$
Choice of probing vectors

Range of $u$ leads to more accurate probing
QR decomposition on the range of $u$ is too expensive
Use the observed data as a proxy (restriction of $u$)

\[
\begin{bmatrix} Q, \sim \end{bmatrix} = \text{qr}(AZ) \quad \text{with} \quad A = D_{\text{obs}}D_{\text{obs}}^T
\]

Data $D_{\text{obs}}$ corresponds to
- restriction of the true wavefield to the receivers
- is representative of its range (frequency content, travel time, ...)

Graff-Kray, M., Kumar, R., and Herrmann, F. J., 2017, Low-rank representation of omnidirectional subsurface extended image volumes:
Crosstalk

Stronger diagonal

Less crosstalk

Less coherent noise

\( Z : \) Random \(+/-1\)

\( F : \) DFT
Approximate gradient FWI/RTM

Algorithm:

0. for \( t=2:nt-1 \) \# forward propagation
1. \( \mathbf{u}[t+1] = f(\mathbf{u}[t], \mathbf{u}[t-1], \mathbf{m}, \mathbf{q}[t]) \)
2. \( \ddot{\mathbf{u}}[r, x] += Q[r, t] \dot{\mathbf{u}}[t, x] \quad \forall r \)
3. end for
4. for \( t=nt:-1:1 \) \# back propagation
5. \( \mathbf{v}[t-1] = f^\top(\mathbf{v}[t], \mathbf{v}[t+1], \mathbf{m}, \delta d[t]) \)
6. \( \nabla[r, x] += Q[r, t] \mathbf{v}[t, x] \quad \forall r \)
7. end for
8. output: \( \frac{1}{r} \text{tr}(\ddot{\mathbf{u}} \nabla^\top) \)

\( \ddot{\mathbf{u}}, \mathbf{v} \in \mathbb{R}^{n_t \times N} \Rightarrow \ddot{\mathbf{u}}, \nabla \in \mathbb{R}^{r \times N}, r \ll n_t \)
Randomized trace estimation

**Ultra-low memory use:**

<table>
<thead>
<tr>
<th>Compute</th>
<th>FWI</th>
<th>DFT</th>
<th>Probing</th>
<th>Optimal checkpointing</th>
<th>Boundary reconstruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>$N \times n_t$</td>
<td>$2r \times N$</td>
<td>$r \times N$</td>
<td>$O(\log(n_t)) \times N \times n_t$</td>
<td>$n_t \times N$</td>
</tr>
</tbody>
</table>

For fixed $r \ll n_t$

- half memory cost of DFT
- half compute cost of DFT
- simple real-valued algorithm

In practice, needs much smaller $r$ compared to DFT.
Randomized trace estimation

Ultra-cheap imaging conditions:

\[ Q^T (D_x u[\cdot, x]) = D_x (Q^T u[\cdot, x]) \]

Apply space-only imaging condition to time-compressed wavefields:

- \( k \)-space filter
- inverse-scattering imaging condition (ISIC)

Imaging condition usually costs extra(s) PDEs (ISIC = 1 PDE)

\[ \mathcal{I}(u[t], v[t]) = \sum_t m u'[t] v[t] + \nabla u[t] \cdot \nabla v[t] \]
Randomized trace estimation

Subsurface Common-Image Gathers (CIGs):

\[
\delta M[x, h] \approx \frac{1}{r} \operatorname{tr} \left( \bar{u}[\cdot, x + h] \bar{v}[\cdot, x - h]^\top \right)
\]

\(h\) subsurface offset

- computed in compressed space
- reduced memory footprint
- less computational cost
FWI example

2D overthrust model
OBN acquisition

Comparisons:
- standard FWI
- on-the-fly DFT
- randomized trace estimation
Accuracy – gradients

- converges to true gradient as $r \rightarrow n_t$
- less accurate near source

Acceptable accuracy
50 × – 100 × memory saving
Accuracy – gradients

- Exact for large $r$
- Noisy error

Accurate but becomes expensive
FWI w/ randomized Trace estimation
FWI w/ on-the-fly DFT
3D, first gradient

**Overthrust 3D:**

- Marine acquisition
- 12.5Hz Ricker wavelet bandpass filtered at 3-15Hz
- 32 probing vectors $\rightarrow$ 40 $\times$ memory reduction
- Probing on GPU (NVidia M60, $45/hr$)
- True gradient on CPU (Intel Skylake, $65/hour$)
True gradient
Estimated gradient with 32 vectors

50 × Memory gain
Imaging

- Sparse OBN
- TTI imaging
- Subsurface common image gather

Makes RTM conducive to acceleration w/ GPUs
All on Azure NC6 (NVidia M60 w/ 8Gb memory)
RTM

SEAM 2D:

- 44 OBN 1km apart
- 3521 sources 12.5m apart
- 14.5Hz Ricker wavelet
- 64 probing vectors (160 × memory savings, 84Gb vs .5Gb)

Makes RTM conducive to acceleration w/ GPUs
Noisy but accurate

RTM ($r = 64 \rightarrow 160 \times$ memory savings)
TTI RTM

BP 2D TTI:
- 1600 Marine sources
- 8km offset
- 19.5Hz Ricker wavelet
- 64 probing vectors (160 × memory savings, 84Gb vs .5Gb)

Makes TTI-RTM conducive to acceleration w/ GPUs
Noise stacks out /w dense source sampling

RTM ($r = 64 \rightarrow 160 \times$ memory savings)
Subsurface CIG

Depth (km)

Location (km)
Conclusions

Leverage Randomized Linear Algebra

Low-memory footprint and low algorithmic complexity

Controllable error

Allows for accelerators (GPUs)

Drop-in extension for existing open source framework JUDI/Devito
Discussion

Proof of concept
- incurs somewhat of computational hit (negating acceleration of GPUs)
- we stay 100% on GPU
- as w/ optimal checkpointing memory gains offset by computational overhead
- implementation can be improved so overhead is minimal

In theory, we outperform DFT methods
- similar structure
- real valued
- require fewer basis vectors
Open source software

**TimeProbeSeismic.jl:**
- Open-source MIT license
- Built on top of **JUDI.jl**
- Leverages Devito
- [https://github.com/slimgroup/TimeProbeSeismic.jl](https://github.com/slimgroup/TimeProbeSeismic.jl)

**ImageGather.jl:**
- Open-source MIT license
- Built on top of **JUDI.jl**
- Leverages Devito
- [https://github.com/slimgroup/ImageGather.jl](https://github.com/slimgroup/ImageGather.jl)
This research was carried out with the support of Georgia Research Alliance and partners of the ML4Seismic consortium.