Seismic imaging is a supercomputing problem: the surrogate-and-GPU stack reshaping RTM and FWI economics

Reverse-time migration and full-waveform inversion are among the largest commercial high-performance-computing workloads on Earth. An exploration FWI project routinely consumes millions of core-hours, and the cost concentrates in the forward-and-adjoint wave solve: two anisotropic propagations per shot per iteration, times thousands of shots, times tens of iterations, on grids of billions of cells. For thirty years the response was to buy more cluster. The response now is to change the arithmetic of the solve itself. Neural-operator surrogates make a forward call up to three orders of magnitude cheaper. Differentiable simulators return the exact gradient by automatic differentiation, retiring the hand-coded adjoint. Mixed-precision computation and quantized inference fit bigger models on fewer cards. Compute-aware orchestration stops paying for solves that do not move the answer, and sample-efficient methods run fewer of them in the first place. Stacked, these levers do not shave a percentage; they change the regime: velocity building shifts from an overnight batch on a fixed cluster toward an interactive, elastically sized loop, and a solve cheap enough to run a thousand times turns a single best-fit image into a calibrated posterior. The limits are real: surrogates extrapolate silently outside their training prior, learned gradients are approximate, and on a real cluster the bottleneck is frequently I/O rather than FLOPs. The discipline the stack demands is the one it rewards: gate the surrogate, keep an exact reference in the loop, and measure the whole pipeline.

Tannistha Maiti
Senior AI Researcher

SEISMIC-IMAGING / FULL-WAVEFORM-INVERSION / REVERSE-TIME-MIGRATION / HPC / NEURAL-OPERATORS / DIFFERENTIABLE-SIMULATION

CONTENTS

Reverse-time migration and full-waveform inversion are among the largest commercial high-performance-computing workloads on Earth. An exploration FWI project routinely consumes millions of core-hours, and the bill is dominated by one inner loop: solving the wave equation, forward and adjoint, over and over. For thirty years the response was to buy more cluster. The response now is to change the arithmetic of the solve itself.

Four levers do the changing. Neural-operator surrogates make a forward call up to three orders of magnitude cheaper [5]. Differentiable simulators return the exact gradient without a separate adjoint [7]. Mixed-precision and quantized inference fit bigger models on fewer cards. Compute-aware orchestration stops paying for solves that do not move the answer. This whitepaper maps the seismic HPC workload, locates the bottleneck, and lays out the four levers that are moving depth imaging from an overnight batch job on a CapEx cluster toward an interactive, uncertainty-aware, elastic-compute workflow. Each lever is deep enough to deserve its own treatment; here they are assembled into one argument.

Why seismic imaging is a supercomputing problem

A modern survey is tens of terabytes to petabytes of prestack data, and the imaging algorithms that turn it into a depth volume are wave-equation solvers. Reverse-time migration propagates source and receiver wavefields through a velocity model and cross-correlates them at every grid point and time step. Full-waveform inversion goes further: it iterates, updating the velocity model, and, done properly, the anisotropy model, until modelled data match recorded data.

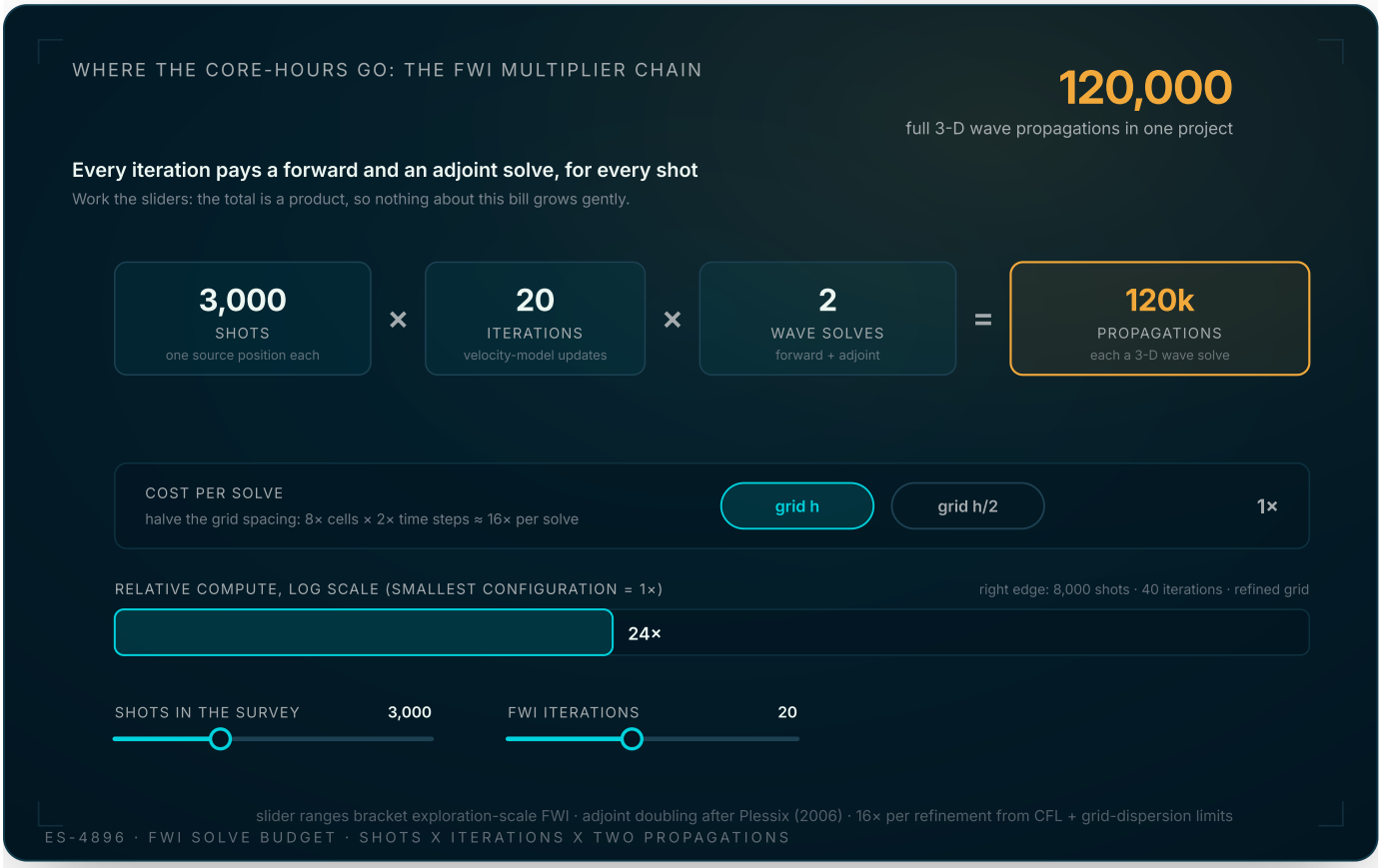
Each RTM shot is a pair of 3-D wave propagations. Each FWI iteration is a forward plus an adjoint propagation per shot, times thousands of shots, times tens of iterations. The grids are billions of cells, the time steps are tens of thousands, and the whole thing is memory-bandwidth-bound stencil computation. This is not a workload with an HPC component. It is an HPC workload with a geophysics label.

The bottleneck is the wave solve, and the adjoint doubles it

The cost concentrates in the forward operator. A finite-difference or spectral-element wave solve is bound by two constraints an asset team knows viscerally: a Courant-Friedrichs-Lewy stability limit tying the time step to the fastest velocity, and a grid-dispersion limit

forcing the cell size below a fraction of the shortest wavelength. Halving the grid spacing therefore multiplies the cell count by eight and the time steps by two, a roughly sixteen-fold cost per refinement.

On top of that, the classical gradient, the adjoint-state method, is a second full propagation per shot. So the honest unit cost of FWI is two anisotropic wave solves per iteration per shot, and every lever below is really a lever on that number.



Where the core-hours go. An FWI project pays two full 3-D wave propagations, one forward and one adjoint, per shot per iteration, so the total solve count is a product: shots times iterations times two. Drag the sliders through the ranges a real exploration project occupies, thousands of shots and tens of iterations, and the count runs into the hundreds of thousands. The lower strip adds the resolution tax: halving the grid spacing multiplies cells by eight and time steps by two, roughly sixteen times the cost per solve. The chain is exact arithmetic; the slider ranges are chosen to bracket the exploration-scale workload described in the text rather than quoted from any single project, and the adjoint accounting follows Plessix (2006).

Lever one: surrogates change the cost of the solve

The first lever replaces or accelerates the solve. A neural operator, a Fourier neural operator or a DeepONet, learns the model-to-wavefield map from a trusted solver and then emulates it at up to three orders of magnitude lower per-call cost in the operator-learning literature

[5], differentiable end to end so its gradient comes with the call. Seismic-specific results sit lower still, in the two-to-three order range, and depend heavily on how the comparison is set up. That figure is a per-call number, and it is not the whole bill: the training set has to be generated by the trusted solver first, so a surrogate pays for itself only where the call count amortises that up-front simulation cost. Where it does, the per-call economics is the enabler for ensemble and uncertainty FWI that a single expensive solver cannot touch.

A differentiable classical simulator, such as the JAX-based j-Wave or PyTorch Deepwave, keeps the same discretised physics as the solver it replaces and returns the exact gradient of that discretised objective by automatic differentiation, retiring the hand-coded adjoint. Amortised inversion, the simulation-based-inference route, pushes the idea to its conclusion: pay the simulation cost once to train, then invert new data in a forward pass.

Lever two: fit it on the hardware

The second lever is memory and precision. A continent- or basin-scale anisotropic velocity volume plus its wavefields does not fit under a single GPU's memory ceiling, so teams tile, shard, or offload, each with a cost, or they shrink the footprint. Mixed-precision training and computation roughly halve memory and roughly double throughput, on tensor-core hardware for the learned parts of the stack and through halved memory traffic for the bandwidth-bound stencil kernels themselves, at a controlled accuracy cost. Quantized inference, stepping from fp32 to fp16 to int8, shrinks a trained model to run on constrained or field hardware. Batch-of-one training with the right normalisation keeps the largest single model trainable on one card.

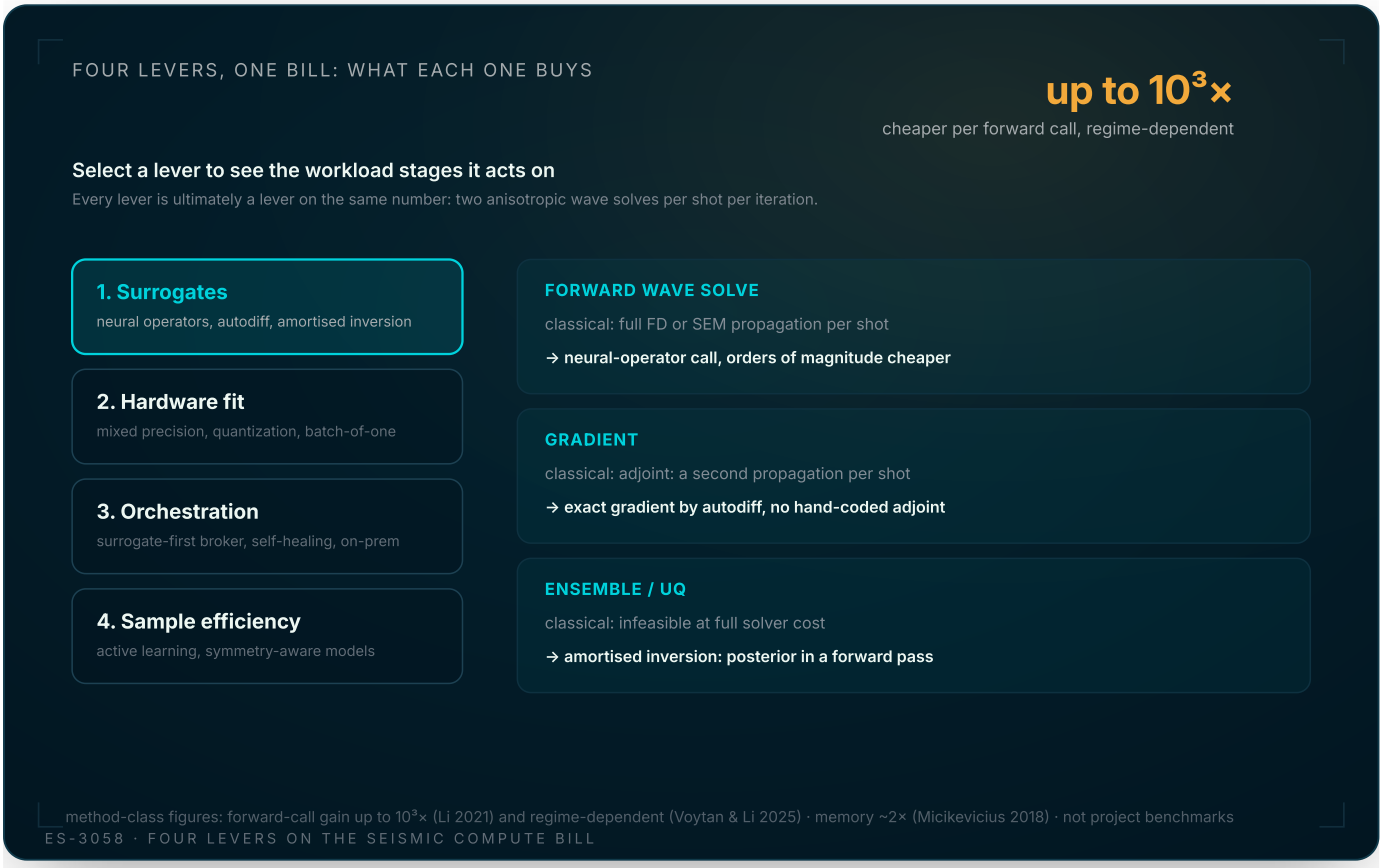
Lever three: compute-aware orchestration

The third lever is not making any single solve cheaper but refusing to run the ones that do not matter. A surrogate-first HPC broker answers with a cheap neural-operator prediction by default and escalates to a true finite-difference or spectral-element run only when the surrogate's uncertainty is high, so the cluster is spent where it changes the answer. Self-healing pipelines checkpoint the wavefield and restart, reroute, or autoscale when a long SLURM job dies six hours in, so a crustal- or reservoir-scale run finishes without a human

babysitting it. And for the data that cannot leave the building, on-premises, air-gapped MLOps runs the whole stack on operator hardware, DGX-class nodes behind the operator's own walls.

Lever four: spend less compute

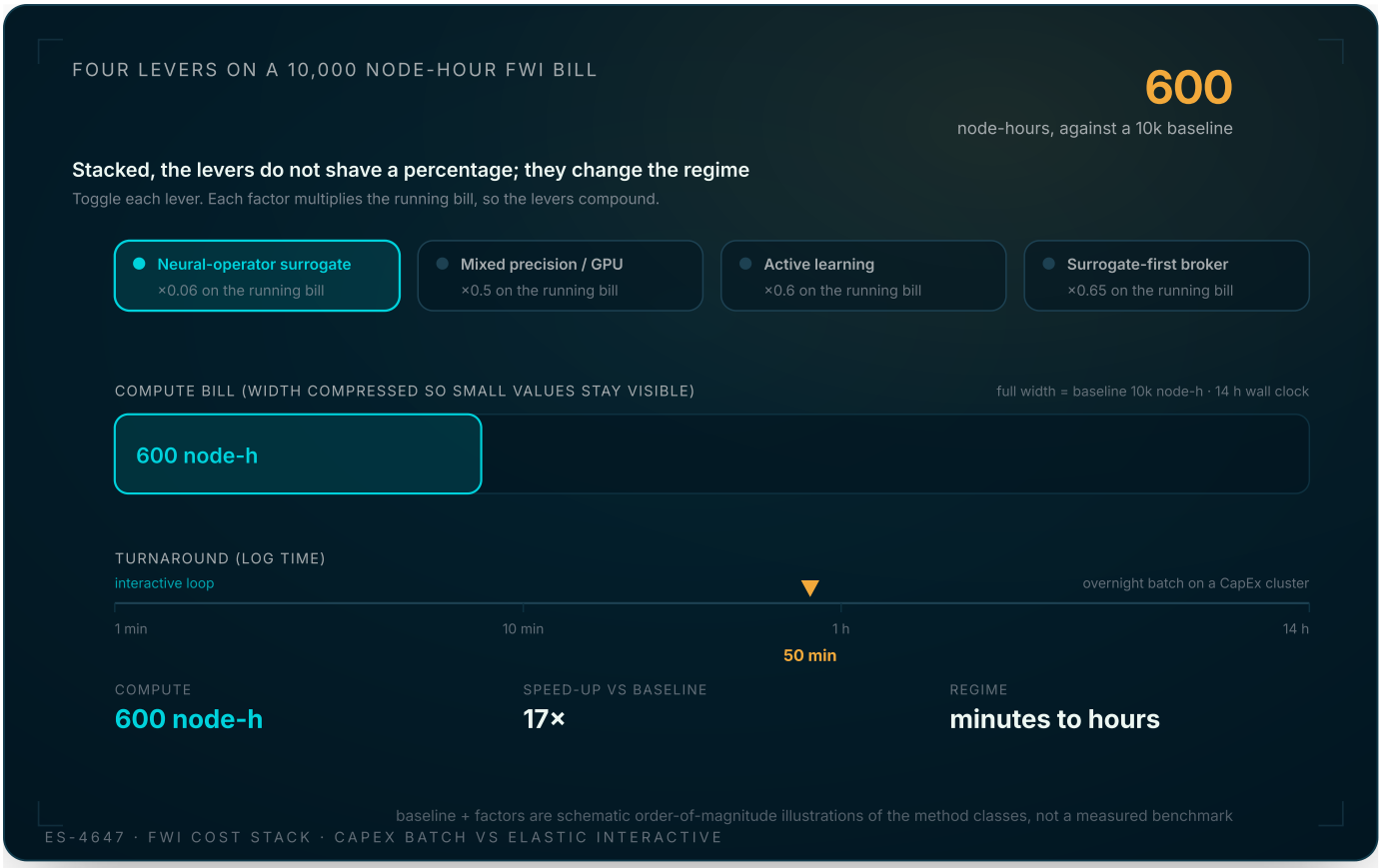
The fourth lever reduces how many expensive solves you run at all. Active learning chooses the next simulation by model uncertainty, so a training set for a surrogate is built from the fewest informative sims rather than a blind sweep. Symmetry-aware models, equivariant networks that respect the rotational structure of the elastic tensor, learn anisotropy from far fewer examples. Both convert compute into sample efficiency.



The four levers mapped to the stages of the workload they act on. Surrogates change the unit cost of the forward call and, through automatic differentiation and amortised inversion, the gradient and the ensemble. Hardware fit roughly halves memory through mixed precision and raises throughput with it, then shrinks serving footprints by quantization. Orchestration refuses to run true solves that do not move the answer and keeps long runs alive. Sample efficiency builds surrogate training sets from the fewest informative simulations. Effects shown are the documented orders of magnitude for each method class, not benchmarks from a specific project, and the forward-call gain in particular is strongly regime-dependent: at matched spatial resolution an operator surrogate can be slower than optimised finite difference.

The economics actually shift

Stacked, these levers do not shave a percentage; they change the regime. A forward call cheap enough to invoke a million times turns velocity building from an overnight batch into an interactive loop. An exact autodiff gradient turns a bespoke adjoint into a drop-in. A surrogate cheap enough to run a thousand times turns a single best-fit model into a calibrated posterior, the uncertainty dividend that a regulated CCS or reservoir decision actually needs. And a surrogate-heavy loop changes the shape of the demand: bursty and interactive rather than a standing overnight queue, which argues for capacity sized to the question rather than to the peak. Whether that capacity is rented or owned is a separate decision, and for confidential data it is frequently owned, as lever three assumes. The workload does not get smaller; the cost per answer does.



The economics shift, one toggle at a time. Baseline is a representative basin-scale anisotropic FWI at roughly 10,000 node-hours and a 14-hour wall clock: an overnight batch job on a CapEx cluster. The surrogate cuts the forward-solve cost by orders of magnitude; mixed precision roughly halves what remains; active learning runs fewer informative simulations; the broker escalates to a true solver only when the surrogate is uncertain. Each factor multiplies the running bill, and with all four on, the same question is answered in an interactive loop on elastic compute. Note that the surrogate lever is shown at a factor of 0.06 rather than the thousand-fold gain quoted for a single forward call: only part of a project is forward solves, and the broker still escalates a fraction of them to a true solver, so the whole-project saving is far smaller than the per-call one. Baseline and per-lever factors are schematic order-of-magnitude illustrations of the documented method classes, not a measured benchmark.

Limitations

None of this is free. A neural operator is only as trustworthy as its training prior is broad, and it extrapolates silently, so out-of-distribution gating is mandatory, not optional. A learned gradient is approximate; where the final image must be exact, the differentiable simulator, not the surrogate, owns the last iterations, and automatic differentiation buys that exact gradient at a checkpointing cost in memory or recomputation that a hand-tuned adjoint was written to avoid. Mixed precision and quantization trade accuracy for footprint and must be validated against a full-precision reference. The levers also overlap rather than stack cleanly: a surrogate that removes most of the forward solves has already removed most of the ground a precision or orchestration lever could have gained on those same

solves. And on a real cluster the bottleneck is frequently not FLOPs but I/O, moving terabytes of prestack data and checkpoints, so a compute lever that ignores data movement can disappoint.

The surrogate speed-up in particular is a claim to interrogate rather than accept. It is measured against a chosen baseline, and the comparison is easy to set up favourably. Voytan and Li benchmarked tensorised Fourier neural operators against optimised finite difference for time-domain wave propagation and found that at matched spatial resolution the operator was consistently slower, by factors of 1.4 to 80, across every physics formulation they tested; the large accelerations appeared only on coarser grids or in restricted settings such as surface-only boundary-field prediction [10]. The honest reading is that the surrogate lever is real but regime-dependent: it pays where the call count is high, the required resolution is modest, and the quantity of interest is narrow, and it should be demonstrated on the operator's own geometry before it is budgeted for.

Two further cautions bound this whitepaper itself. The per-call gains quoted here are the documented orders of magnitude for the method classes, not re-run benchmarks, and a specific project's mileage depends on its geology, acquisition, and prior. And the interactive-regime picture in the final exhibit is a schematic composition of those documented factors, useful for reasoning about where spend goes, not a quotation. The discipline the AI-native stack demands is the same one it rewards: measure the whole pipeline, gate the surrogate, and keep an exact reference in the loop.

WHAT THIS WHITEPAPER ARGUES

1. Seismic depth imaging (RTM, FWI) is a first-class supercomputing workload whose cost concentrates in the forward-and-adjoint wave solve: two anisotropic solves per FWI iteration per shot.
2. Four levers move the economics: surrogates that cheapen the solve, hardware fit via mixed precision and quantization, compute-aware orchestration that skips needless solves, and sample-efficient methods that run fewer.
3. The decisive product is not raw speed but the uncertainty dividend: a solve cheap enough to run a thousand times turns a single image into a calibrated posterior.
4. The stack changes the shape of compute demand, from a standing overnight queue toward bursty interactive use, which argues for capacity sized to the question rather than to the peak.
5. The limits are distributional and I/O-bound: gate surrogates for out-of-distribution inputs, keep an exact reference for the final iterations, and profile the whole pipeline, not just the kernel.

Where the compute goes, and what each lever does

WORKLOAD STAGE	CLASSICAL HPC COST	AI-NATIVE LEVER	EFFECT
Forward wave solve	full FD/SEM propagation per shot	neural-operator surrogate	up to ~10^3× cheaper per call
Gradient	adjoint = 2nd propagation per shot	differentiable simulator (autodiff)	exact gradient, no separate adjoint
Model footprint	exceeds single-GPU memory	mixed precision / quantization	~2× memory + throughput
Iteration count	blind parameter sweeps	active learning	fewer informative sims
Cluster utilisation	static jobs, manual restart	surrogate-first broker, self-healing	spend only where it moves the answer
Ensemble / UQ	infeasible at solver cost	amortised (SBI) inversion	posterior in a forward pass

The table summarises method classes and typical orders of magnitude, not a re-run benchmark.

References

[1] Virieux, J., & Operto, S. (2009). An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6), WCC1-WCC26. doi:10.1190/1.3238367

[2] Plessix, R.-E. (2006). A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167, 495-503. doi:10.1111/j.1365-246X.2006.02978.x

[3] Komatitsch, D., & Tromp, J. (1999). Introduction to the spectral element method for three-dimensional seismic wave propagation. *Geophysical Journal International*, 139, 806-822. doi:10.1046/j.1365-246x.1999.00967.x

[4] Louboutin, M., et al. (2019). Devito (v3.1.0): an embedded domain-specific language for finite differences and geophysical exploration. *Geoscientific Model Development*, 12, 1165-1187. doi:10.5194/gmd-12-1165-2019

[5] Li, Z., et al. (2021). Fourier neural operator for parametric partial differential equations. *ICLR 2021*. arXiv:2010.08895

[6] Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3, 218-229. doi:10.1038/s42256-021-00302-5

[7] Stanziola, A., Arridge, S. R., Cox, B. T., & Treeby, B. E. (2023). j-Wave: An open-source differentiable wave simulator. *SoftwareX*, 22, 101338. doi:10.1016/j.softx.2023.101338

[8] Micikevicius, P., et al. (2018). Mixed precision training. *ICLR 2018*. arXiv:1710.03740

[9] Cranmer, K., Brehmer, J., & Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48), 30055-30062. doi:10.1073/pnas.1912789117

[10] Voytan, D., & Li, L. (2025). Are Fourier neural operators really faster for time-domain wave propagation? arXiv:2508.11119

EarthScan builds the surrogate-and-GPU stack for seismic imaging: neural-operator forward models, differentiable and quantized inference, and compute-aware orchestration. If your team is running RTM/FWI at cluster scale and wants interactive turnaround with calibrated uncertainty, talk to us about the full pipeline.

EARTHSCAN

Seismic imaging is a supercomputing problem: the surrogate-and-GPU stack reshaping RTM and FWI economics

Published July 2026. Generated 2026-07-22.
https://www.earthscan.io/whitepapers/seismic-imaging-is-a-supercomputing-problem