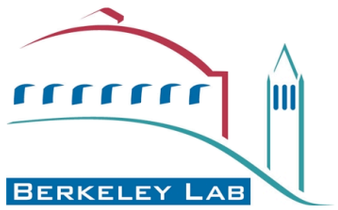
MSST'08 Tutorial: Data-Intensive Scalable Computing for Science

Julio López
Parallel Data Lab -- Carnegie Mellon University

Acknowledgements:
CMU: Randy Bryant, Garth Gibson, Bianca Schroeder (University of Toronto), Greg Ganger.
Intel: Steve Schlosser, David O'Hallaron.
Yahoo!: Milind Bhandarkar, Eric Baldeschwieler.

Acknowledgments

- Material:
  - Randy Bryant, Garth Gibson, Bianca Schroeder (Univ. of Toronto), Greg Ganger.
  - Intel: Steve Schlosser, David O'Hallaron.
  - Yahoo!: Milind Bhandarkar, Eric Baldeschwieler.
- Sponsors and Collaborators:
  National Science Foundation, SciDAC, PDSI, LANL, PNNL, ORNL, SNL, LBNL, Univ. of Michigan, UC Santa Cruz, Pittsburgh Supercomputing Center, Yahoo!, Intel, Hewlett Packard Labs.



ERNEST ORLANDO LAWRENCE
BERKELEY NATIONAL LABORATORY

Agenda

- Motivation
- Turning I/O intensive into CPU intensive workloads
- Failure trends in HPC
- DISC
- Hadoop Intro
- Generating Ground Models for Earthquake Simulations
- M45! Experiences

Methods for compressing large seismic wavefields

Turning I/O intensive data-analysis tasks into massively parallel compute-intensive ones.

Julio López & David O'Hallaron

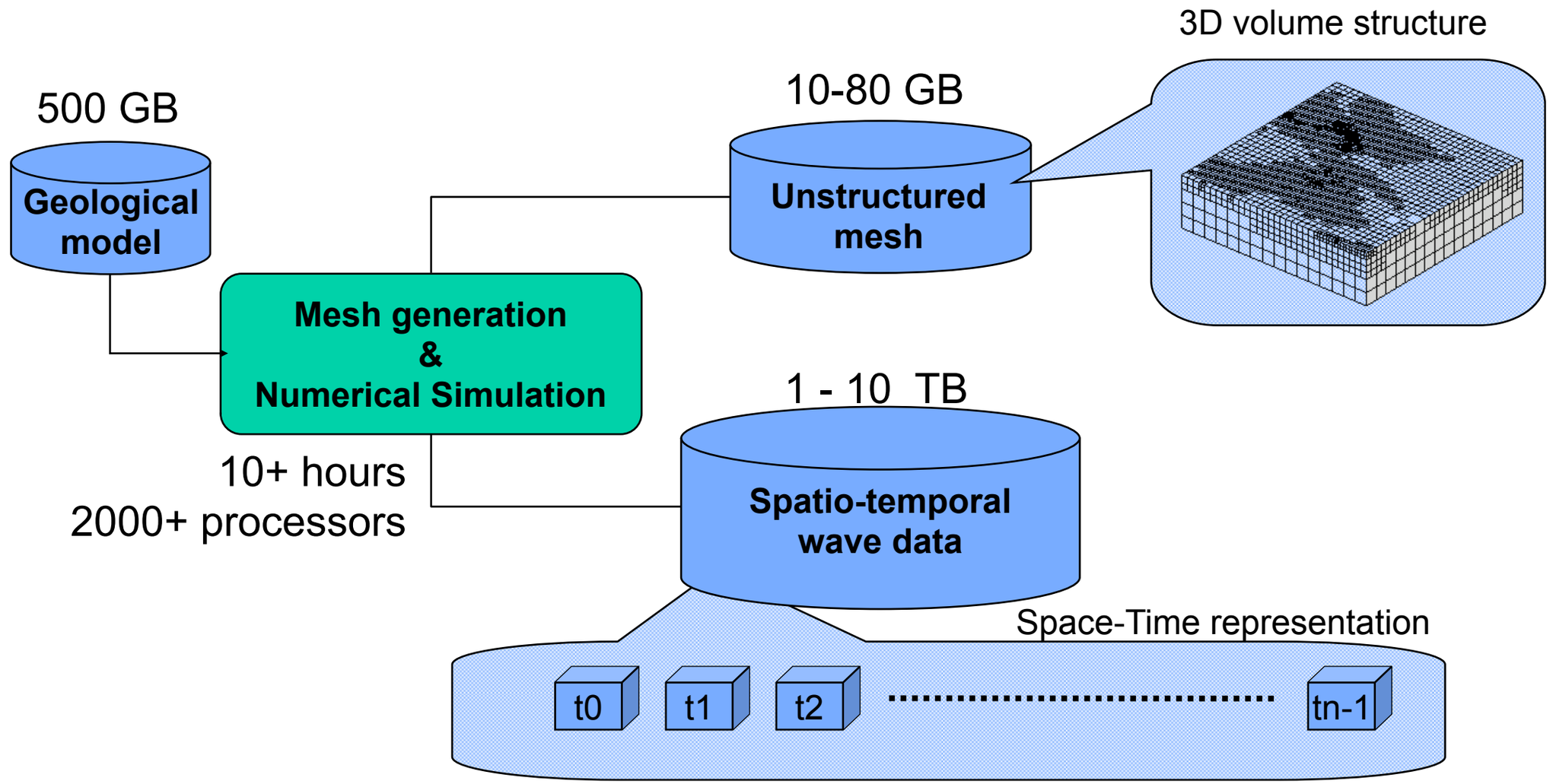
Carnegie Mellon University

Motivation

- Data analytics increasingly difficult at scale
  - Difficult to program
  - Stress I/O and memory systems
  - Dealing with failures
- Search companies routinely process the web graph
- Interest in Data-Intensive Scalable Computing
- How can this be leveraged for science HPC?

Data-Intensive Processing in Seismic Simulation

- Setup: Generating ground models.
- Data-analytics in Quake 4D wavefields: Terabytes per each simulated scenario.



Data analysis challenges

- Resource constraints: I/O BW and memory.
- Complex data operations:
  - Selection, transposition, FFT, filtering…
- Difficult data sharing and management.
- Need for semi-interactive data analysis.
- Data is at risk: write-once / read-never!

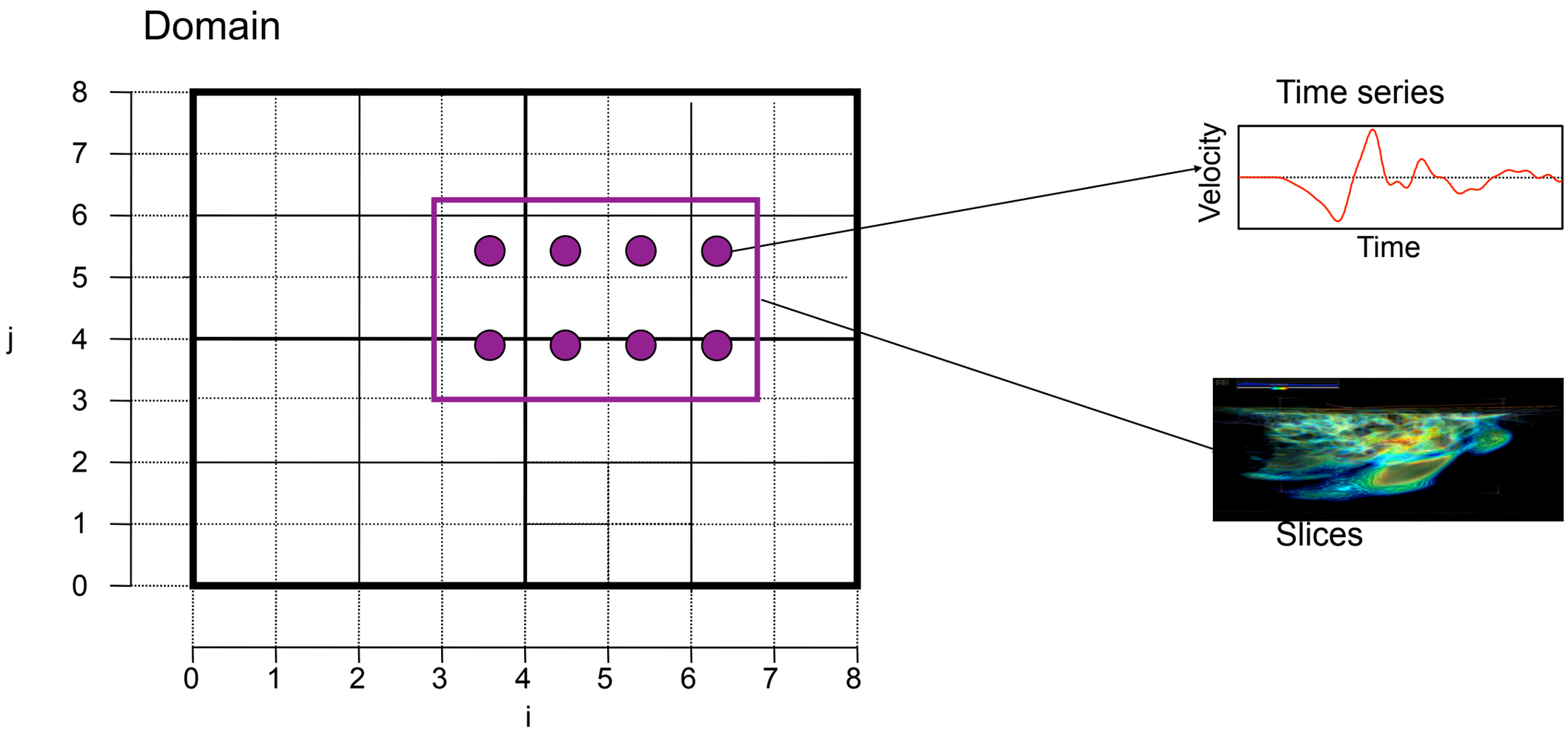
- Need better mechanisms for processing datasets.

Approach

- Field representation:
  spatially indexed compressed data structures
- Effectively, turn data-intensive into compute-intensive.

- Indexed compressed wavefield representations:
  - BEM compression.
  - Frequency-domain compression.
- Microsolver computational query-time model.

Data-analysis queries

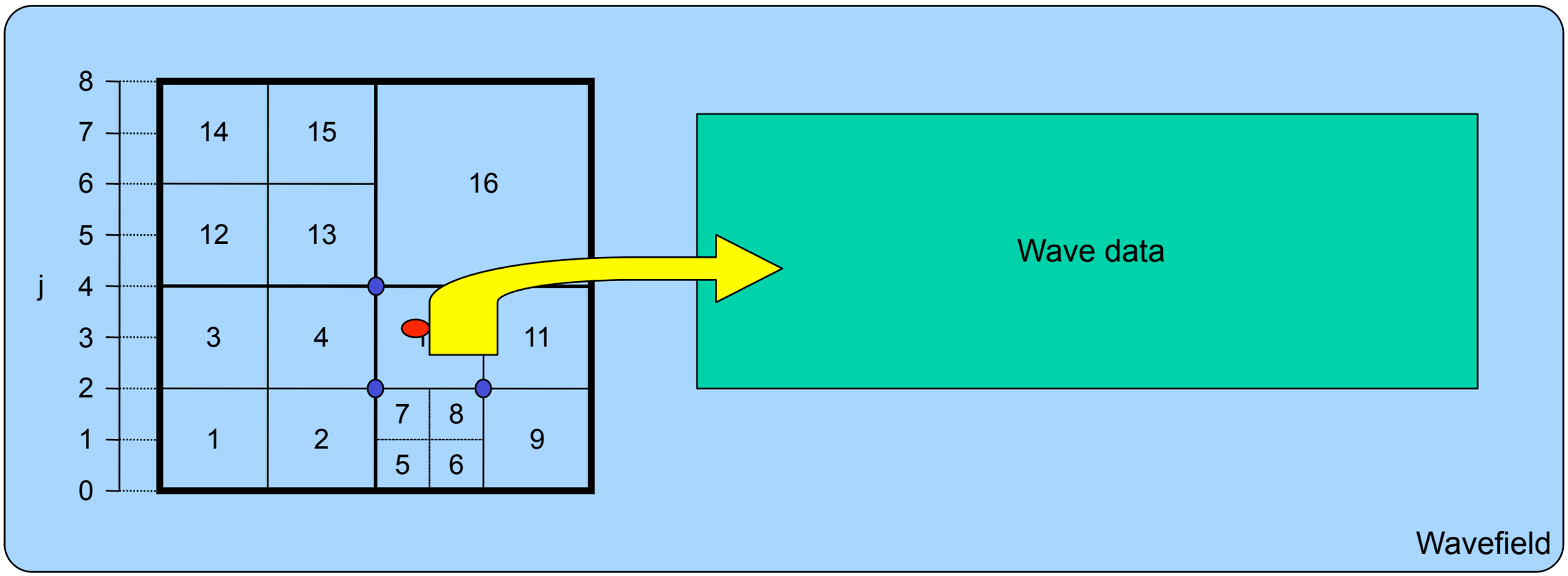


Region of interest (origin, extent, spacing, rotation)

Dimensionality: 0D, 1D, 2D, 3D.

Sampling queries

Querying wavefields



- Mesh: Spatial index, e.g., etrees.
- Find element containing query point.
- Use element corners to retrieve wave data

Compression approach

- Wavefields are floating point datasets

Floating point compression state of the art

- Floating point compression is a hard problem.
- High-entropy bit patterns.

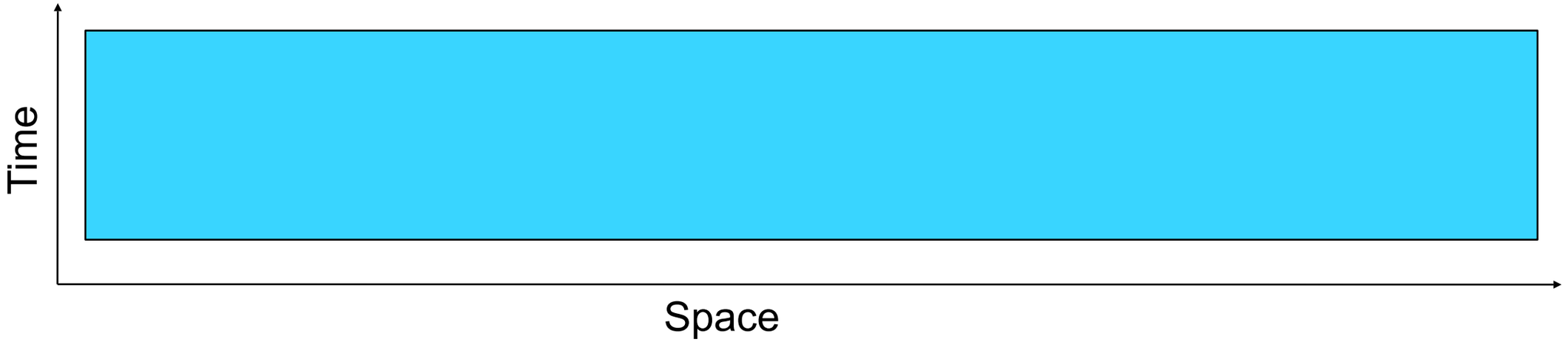
- NASA's image compression: Quantization.
- JPEG 2000: 3D FP images, DCT, DWT, BigInt.
- LLNL: Lorenzo predictor + FP encoding.
- Cornell's value prediction FP encoding.
- Floating point sound compression.

Compression approach

- Wavefields are floating point datasets
- Frequency domain representation
  - Wavefields are band-limited
- Requires a storage layout change
  - Out of core matrix Transposition
- Boundary Element Method Compression (BEMC)
  - Dimensionality-reduction
  - Works in the frequency-domain
  - Allows for incremental reconstruction

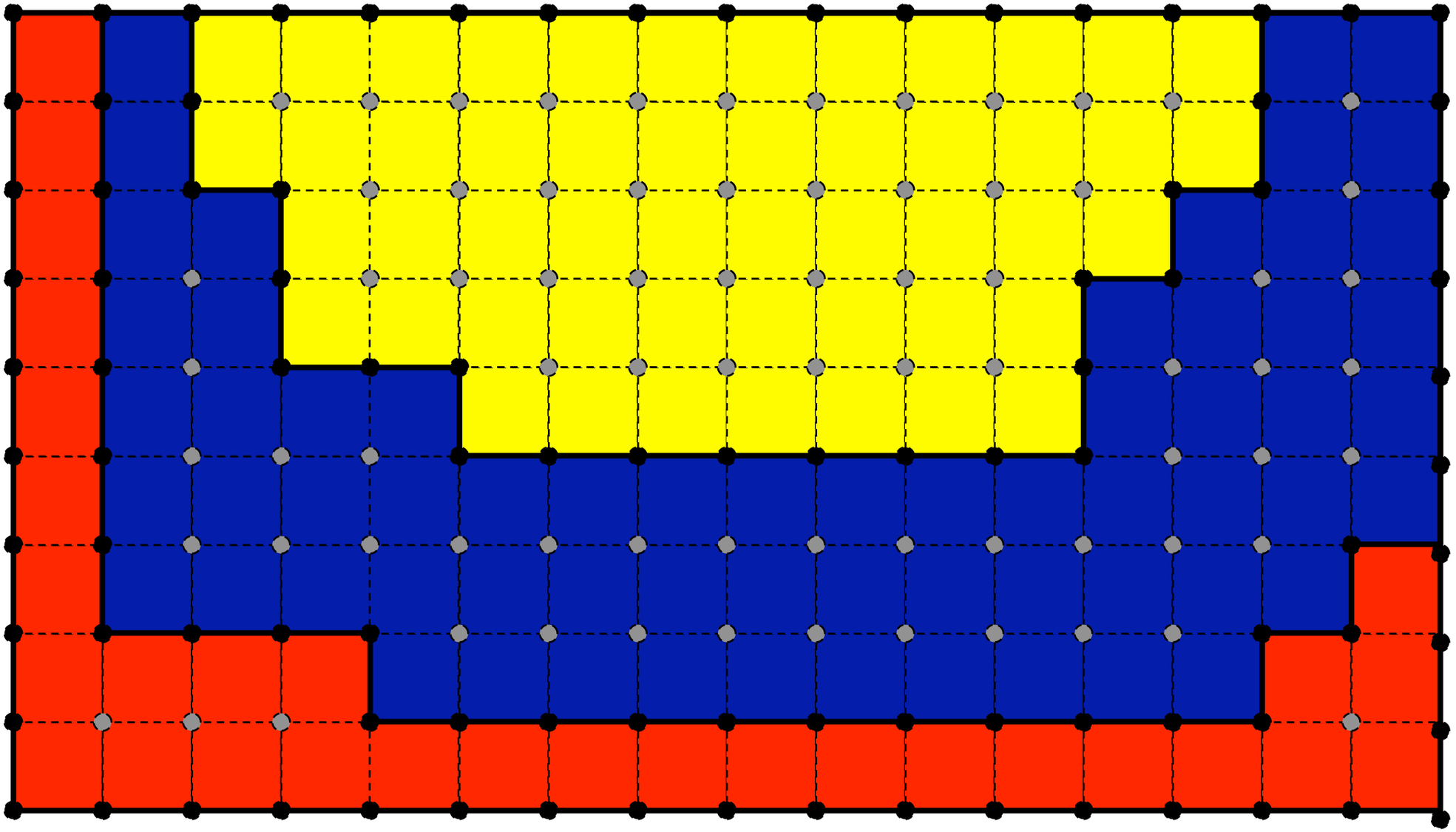
Wavefield storage layout

- Wavefields as matrices: Space x time.
- Large aspect ratios.
- Transposition



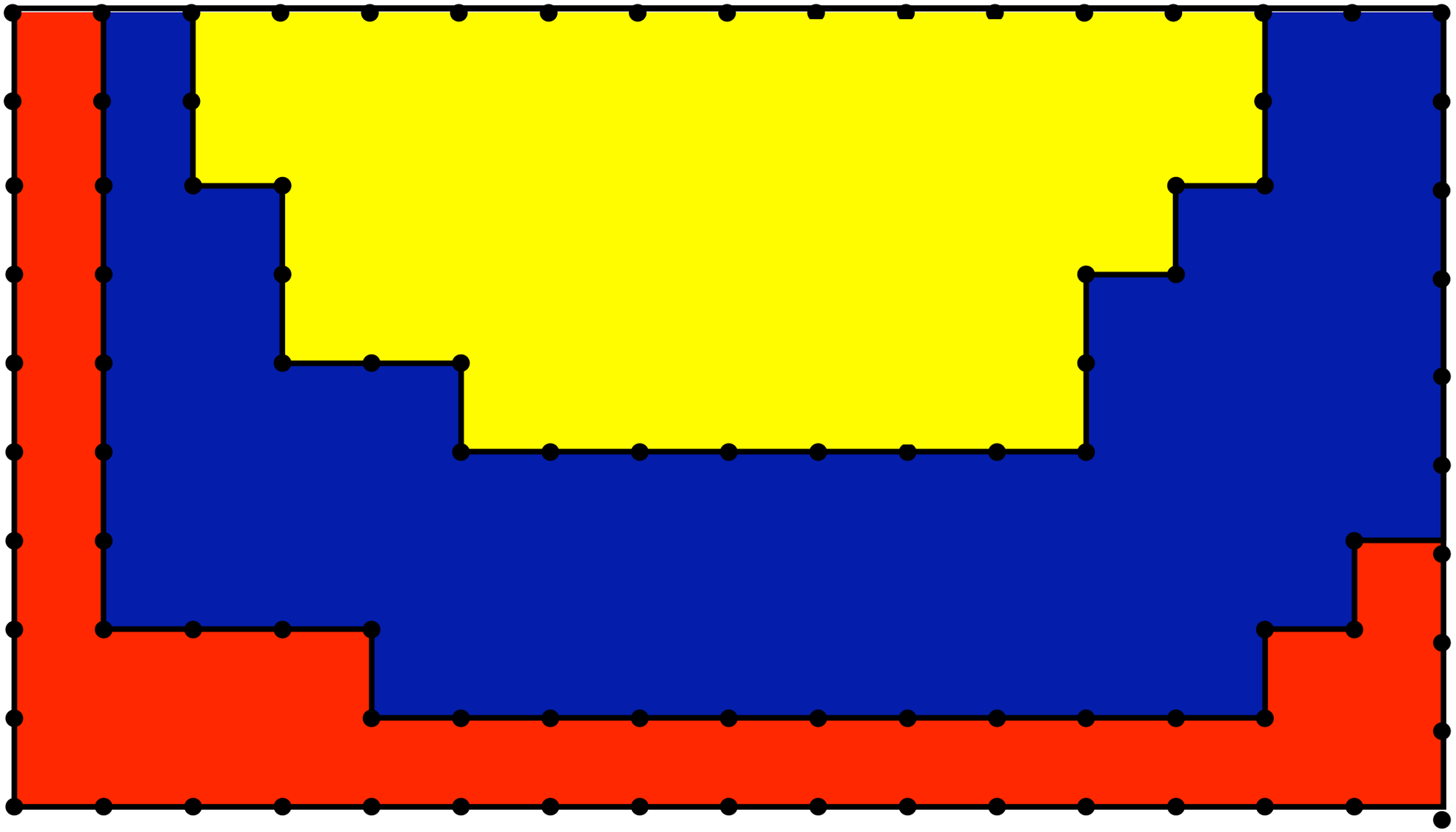
Column size	Row size	
	Small	Large
Small	In-core	Seq. Write
Large	Seq. Read	Split R/W

BEM compression



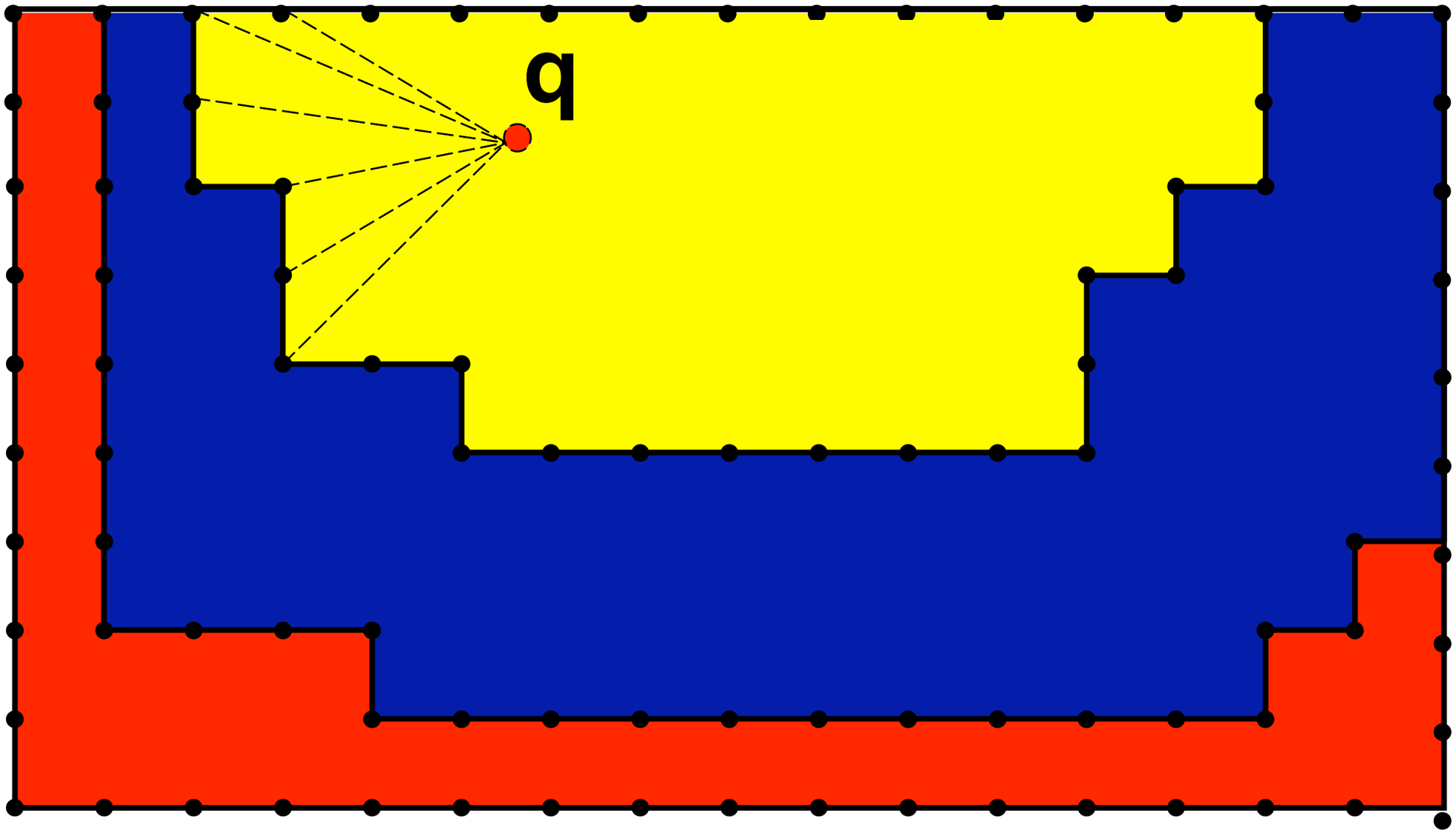
Wavefield domain representation.

BEM compression



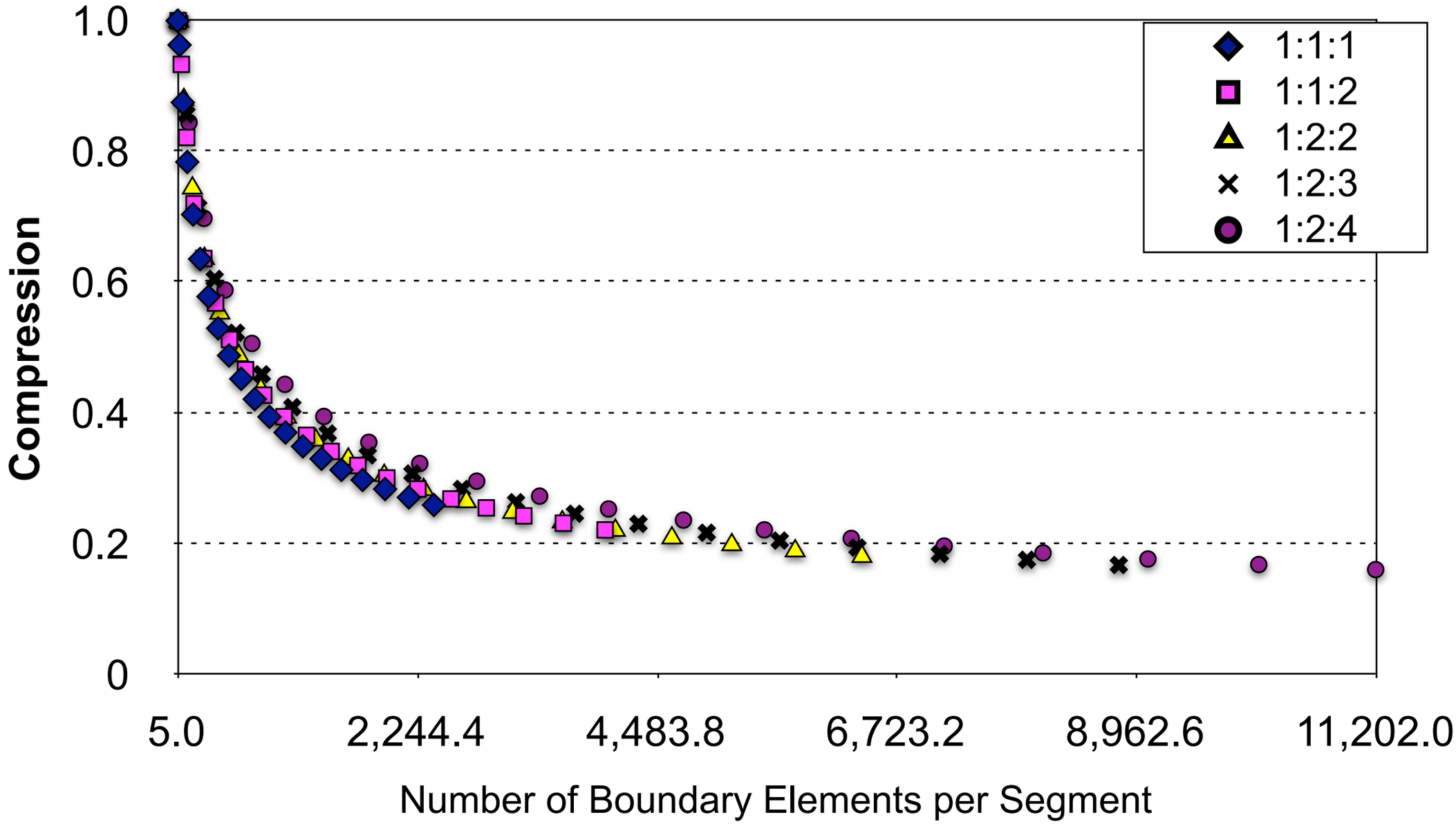
- Wavefield boundary representation.
- Only store values at the boundary.

BEM compression



- Wave data computation with query-time microsolver.
- Convolution u(ξ) * G(ξ,q).

Potential volume to boundary reduction



- Compression factors between 1.5X and 3X.

BEM compression steps

- Model segmentation: find homogeneous regions.
- Generate a boundary meshes.
- Extract wavefield data for boundary mesh nodes.

- Frequency domain representation.
- Transpose from space-time to time-space.
- Transform to the frequency domain.

Query-time BEM microsolver

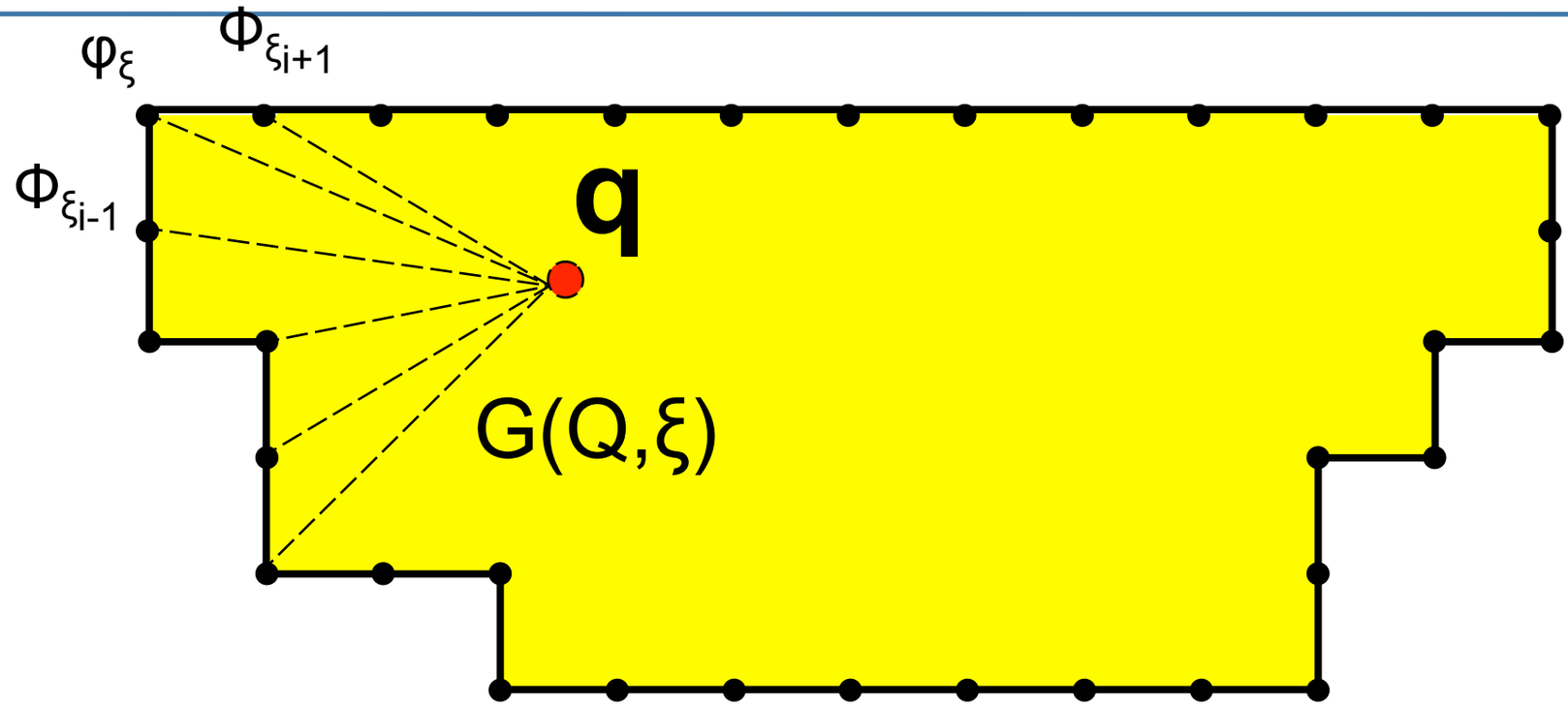
1. Spatial lookup: Find region containing q.
2. Compute phi values for boundary elements.
3. Compute q's data from phi values.

For all frequencies (ω).

Phi-computation: once per region.

$$ui(p,\omega) = \sum_{k=0}^{n-1} \int_{\xi_k} \phi_\omega(\xi) G_i(p,\xi,\omega) d\xi$$

BEM microsolver query computation



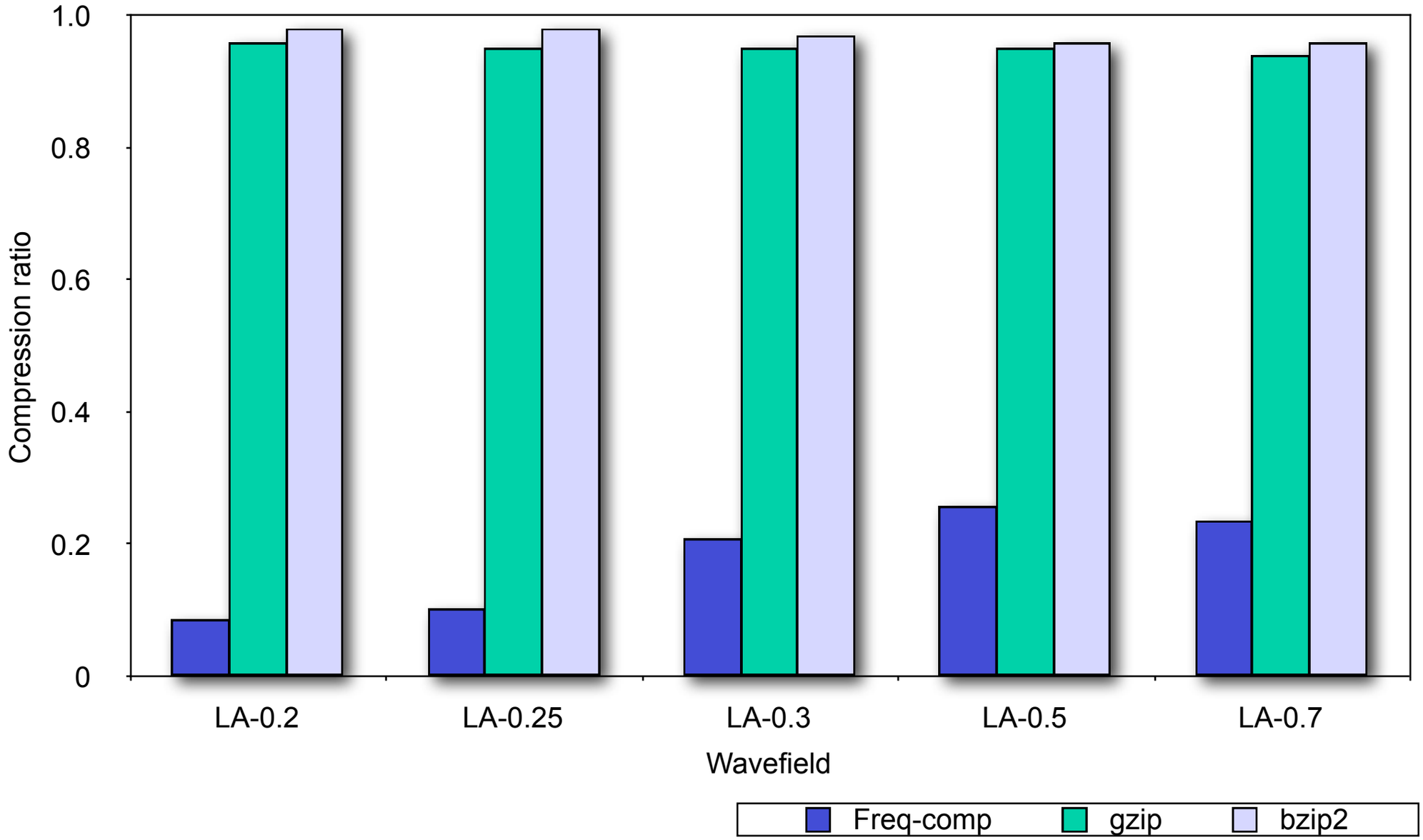
- Apply the integral equation again.
- Phi values are now known.
- Iterate over all BEs and add contribution.

Seismic wavefields

Wavefield	Mesh		Size (GB)
	elements	nodes	
LA-0.2 Hz	0.4 M	0.5 M	6.3
LA-0.3 Hz	1.6 M	1.8 M	25
LA-0.5 Hz	8 M	8.6 M	116
LA-0.7 Hz	18 M	19.4 M	260
LA-1.0 Hz	64.1 M	66.5 M	893

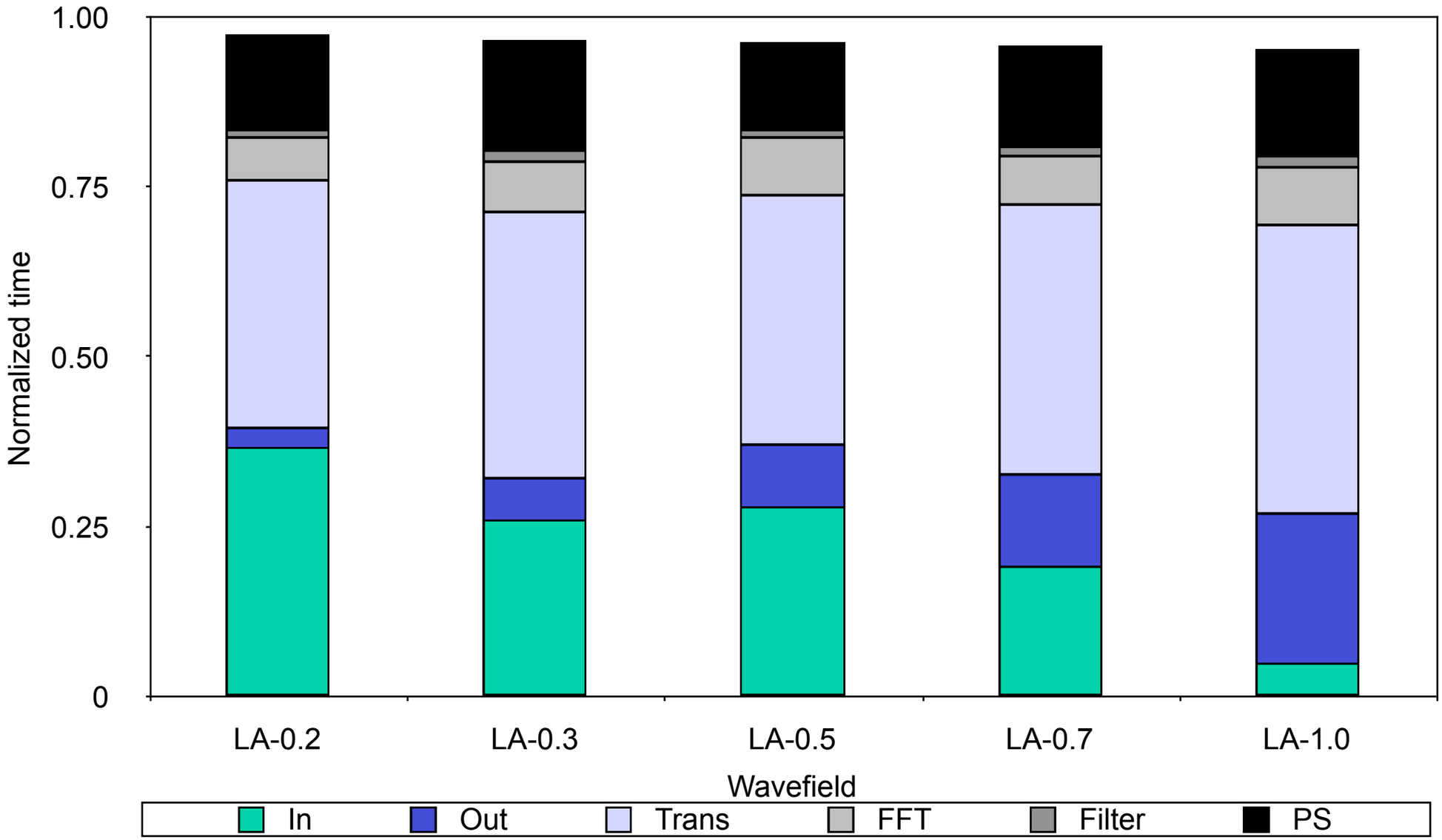
- Min Vs = 500m/s, PPWL = 10
- Pre-segmented LA-basin 0.2Hz model.
- Simulated time = 60s, dt = 0.01s.
- Output sampling rate = 10 Hz, 600 output steps.

Frequency-domain compression ratio



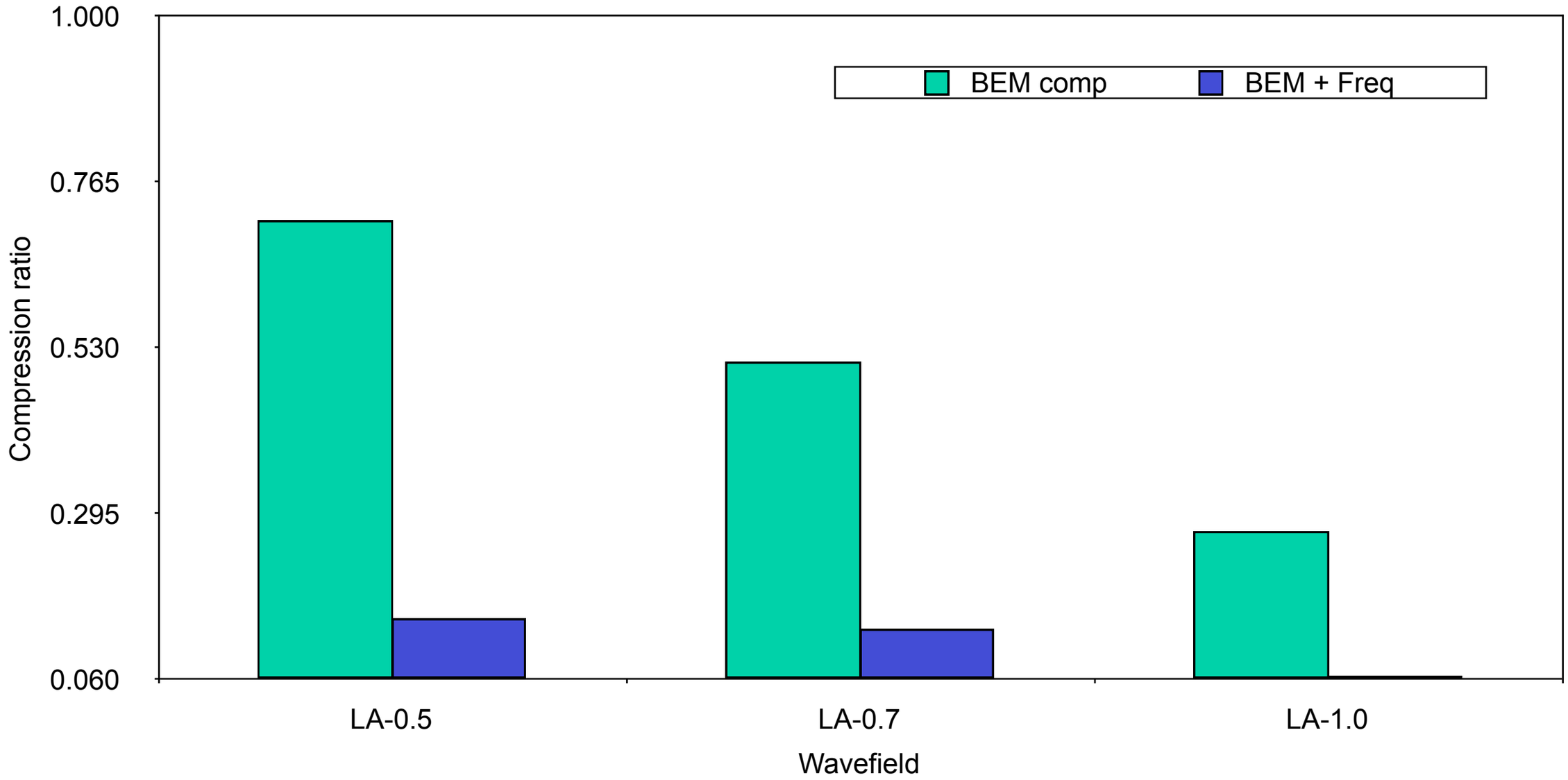
- Compression factor: 5X – 10X.

Frequency compression time



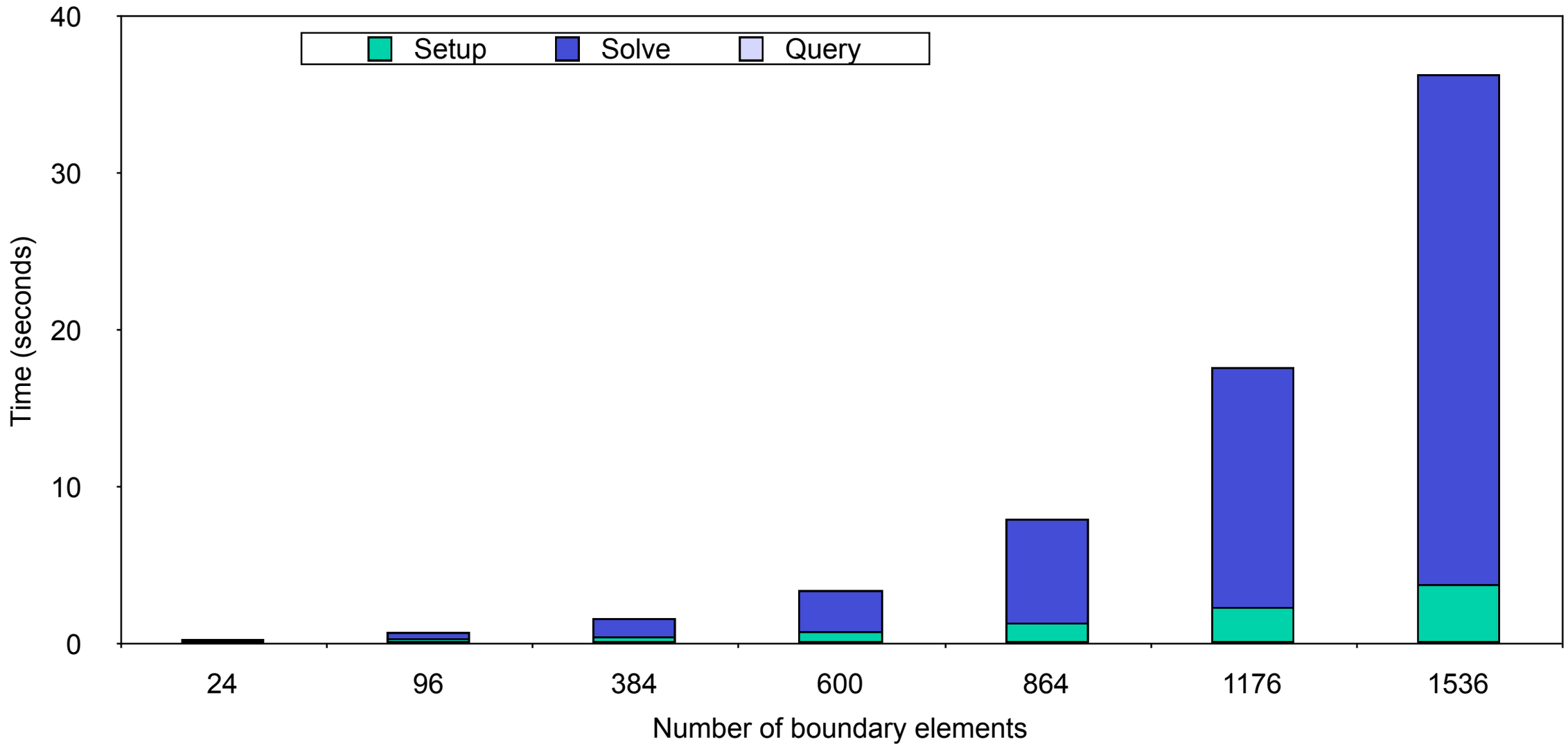
- Compression time < cp time.

BEM compression



- BEM compression factor: 1.4X - 3X.
- BEM + freq. compression factor: 7X – 10X.

BEM query time computation cost



- Query O(n), Setup O(n2), Solve O(n3)

Parallel BEM computation

- Partitioning across homogeneous regions.
- Partitioning phi computation across frequencies.
- Solving system of equations in parallel.
- Partitioning across query points.

Conclusions

- New computational wavefield database approach.
  - Spatially indexed compressed fields.
  - Microsolvers: query-time computation.
- Data-intensive to compute-intensive tasks.

- Indexed compressed wavefield representations:
  - Frequency-domain compression (4X – 5X).
  - BEM compression (5X – 10X) .
- SMT out-of-core efficient transposition (copy time).

Yes … but

- Requires application transformations: Hard!

- What other approaches can we take?

- What about dealing with failures?