Stability-preserving Lossy Compression for Large-scale Partial Differential Equations

Qian Gong
Oak Ridge National Laboratory (ORNL)
Oak Ridge, USA
gongq@ornl.gov

Mark Ainsworth
Brown University
Providence, USA
mark_ainsworth@brown.edu

Jieyang Chen
Oregon University
Eugene, USA
jieyang@uoregon.edu

Xin Liang
University of Kentucky
Lexington, USA
xliang@uky.edu

Liangji Zhu
Florida University
Gainesville, USA
zhu.liangji@ufl.edu

Ethan Klasky
Florida University
Gainesville, USA
eklasky@ufl.edu

Tushar Athawale
Oak Ridge National Laboratory (ORNL)
Oak Ridge, USA
athawaletm@ornl.gov

Qing Liu
New Jersey Institute of Technology (CSLA)
Newark, USA
qing.liu@njit.edu

Anand Rangarajan
University of Florida
Gainesville, USA
anand@cise.ufl.edu

Sanjay Ranka
University of Florida
Gainesville, USA
ranka@cise.ufl.edu

Scott Klasky
Oak Ridge National Laboratory (ORNL)
Oak Ridge, USA
klasky@ornl.gov

Abstract

Checkpoint/Restart (C/R) strategies are vital for fault tolerance in PDE-based scientific simulations, yet traditional checkpointing incurs significant I/O overhead. Lossy compression offers a scalable solution by reducing checkpoint data size, but conventional methods often lack control over physical invariants (e.g., energy), leading to instability such as oscillations or divergence in Partial Differential Equations (PDE) systems. This paper introduces a stability-preserving compression approach tailored for PDE simulations by explicitly controlling kinetic and potential energy perturbations to ensure stable restarts. Extensive experiments conducted across diverse PDE configurations demonstrate that our method maintains numerical stability with minimal error magnification—even across multiple checkpoint-restart cycles—outperforming state-of-the-art lossy compressors. Parallel evaluations on the Frontier supercomputer show up to 8.4× improvement in checkpoint write performance and 6.3× in read performance, while maintaining relative L^2 errors ~2e-6 throughout continued simulation. These results provide practical guidance for balancing compression accuracy, stability, and computational efficiency in large-scale PDE applications.

SC '25, St Louis, MO, USA

ACM ISBN 979-8-4007-1466-5/25/11
https://doi.org/10.1145/3712285.3759878

CCS Concepts

• **Computer systems organization** → **Dependable and fault-tolerant systems and networks**; • **Mathematics of computing** → **Mathematical analysis**.

Keywords

Checkpoint-restart, lossy compression, stability preservation, large-scale PDEs

ACM Reference Format:
Qian Gong, Mark Ainsworth, Jieyang Chen, Xin Liang, Liangji Zhu, Ethan Klasky, Tushar Athawale, Qing Liu, Anand Rangarajan, Sanjay Ranka, and Scott Klasky. 2025. Stability-preserving Lossy Compression for Large-scale Partial Differential Equations. In *The International Conference for High Performance Computing, Networking, Storage and Analysis (SC '25), November 16–21, 2025, St Louis, MO, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3712285.3759878

1 Introduction

Partial differential equations (PDE) forms the mathematical foundation for a broad range of physical modeling and computational analysis tasks [7, 28]. For complex systems, PDEs are typically solved using finite difference, finite element, and finite volume methods [55], which approximate continuous solutions over discretized space and time domains. Large-scale PDE simulations often span billions of grid points and require continuous computations over days or weeks, making them highly vulnerable to system failures, interruptions, hardware crashes, or power outages.

Checkpoint/restart (C/R) is widely adopted fault-tolerance techniques in HPC systems[19, 66, 67]. By periodically saving the simulation or system state to persistent storage, C/R enables applications to resume and recover execution after interruptions. However, as simulations scale, the volume and the frequency of checkpointing increase significantly. While compute capabilities have outpaced I/O bandwidth growth, this imbalance has made frequent checkpointing a major performance bottleneck—leading to extended runtime, I/O contention, and pressure on storage infrastructure [6, 51, 58]. Advanced techniques such as multi-level checkpointing [32, 50] and burst buffer [47] mitigate some of this overhead, but require complex management of memory/storage hierarchies and careful checkpoint scheduling. In contrast, lossy compression-based checkpointing offers a more scalable and easily integrable solution. By compressing checkpoints prior to writing them to the parallel file system (PFS), it can simultaneously reduce I/O time, storage footprint, and data movement costs.

State-of-the-art lossy compressors—MGARD [1, 2, 13, 23], ZFP [44, 45], SZ3 [17, 43, 63], and SPERR [40]—support strict error control, ensuring that the reconstruction error in norms like L^2 and L^∞ remains within user-defined error tolerance. However, many PDE systems, particularly those governing fluid dynamics or turbulence, are highly sensitive to perturbations due to their chaotic nature [26, 35]. A small compression error in the checkpoint data will propagate and grow after restart. While prior work [8, 9, 21, 57, 59, 64] has shown some success in restarting PDE simulations from lossy checkpoints, fundamental questions—specifically, why such errors grow over time and how to design strategies that prevent instability in continued simulations—remain unresolved.

Second order linear PDEs are generally classified into three major categories [20]: parabolic (e.g., the heat equation), hyperbolic (e.g., the wave equation), and elliptic (the Laplace's equation). This paper focuses on second-order hyperbolic PDEs, specifically the wave equation, to highlight the importance of preserving energy growth during lossy checkpointing. For first-order hyperbolic systems, compression-induced instability is less severe due to lower regularity of solutions. We exclude parabolic PDEs because they naturally dampen high-frequency errors, allowing compression perturbations to dissipate over time. A previous study on lossy compression for checkpointing heat conduction PDEs [8] has shown that compression errors tend to smooth out and attenuate as the simulation progresses. Elliptic PDEs are also omitted, as they describe steady-state phenomena where the solution does not evolve dynamically.

The key novelty of this work lies in introducing a physics-informed approach to lossy checkpointing. Unlike prior studies that treat compression as an application-agnostic data reduction problem, we demonstrate that respecting the governing physics is essential for stability. In particular, for the second-order wave equation, we show that stable restart requires separating temporal iterates into kinetic and potential energy components, each subject to different error constraints. This insight explains, for the first time, why small perturbations can grow uncontrollably in hyperbolic PDEs and provides a concrete mechanism to suppress error amplification. Our contributions can be summarized as follows:

- We analyze the root causes and dynamics of error growth after restart from lossy-compressed checkpoints, establishing the vital need to preserve physics rather than relying solely on error norms such as L^2 or L^∞.
- We provides a concrete application with non-trivial physics—the second-order hyperbolic PDE—that requires a novel compression strategy for checkpointing.
- We design an energy-preserving lossy compression algorithm and demonstrate, both theoretically and experimentally, that it enables stable restart even after multiple consecutive C/R cycles.
- We evaluate the I/O and performance benefit of our method at scale, showing 4.2–6.3× faster reads and 2.9–8.4× faster writes on the Frontier supercomputer while keeping the relative RMSE at ~2e-6 in continued simulation.

2 Related work

2.1 Checkpoint-restart in PDE problems

Checkpointing is a widely adopted fault-tolerance strategy in high-performance computing (HPC), enabling applications to recover from failures and maintain computational resiliency [18, 27, 30, 36]. It works by periodically saving the current simulation state to persistent storage, allowing execution to resume from the most recent checkpoint after a failure or interruption [52]. However, traditional checkpointing introduces substantial I/O overhead and storage pressure, especially for applications running at scale.

To address these challenges, several strategies have been proposed. Diskless checkpointing [29, 56] stores processor states in memory distributed across checkpoint nodes, enabling fast recovery but offering limited fault tolerance. Multilevel checkpointing [4, 32, 47, 50] leverages storage hierarchies to reduce I/O latency, but requires careful scheduling and does not reduce the overall data volume. Compression-based checkpointing [31, 35, 48, 57], reduces both storage footprint and data movement cost by applying lossy or lossless compression before writing data to the PFS. As computational performance—particularly with GPUs—continue to outpace I/O bandwidth growth, compression-based checkpointing has become an increasingly attractive solution for scalable and efficient fault tolerance.

2.2 Lossy data compression

Scientific data compression techniques are broadly classified as lossy and lossless techniques. While lossless compressors [22, 45] preserve full data fidelity, they typically achieve modest compression ratios—often less than 2× for floating-point scientific data [3]. Lossy compression [10, 11], in contrast, offers much higher reduction ratios by allowing bounded degradation in accuracy. Modern lossy compressors implement error control mechanisms to ensure the reconstruction error remains within user-defined thresholds. However, most general-purpose compression tools only guarantee bounds in the L^2 or L^∞ norms for primary data, without explicit control over derived physical quantities or application-specific quantities of interest (QoIs). While some specialized methods [24, 33, 46] support application-specific QoIs, they often rely on conservative error estimates, limiting overall compression efficiency.

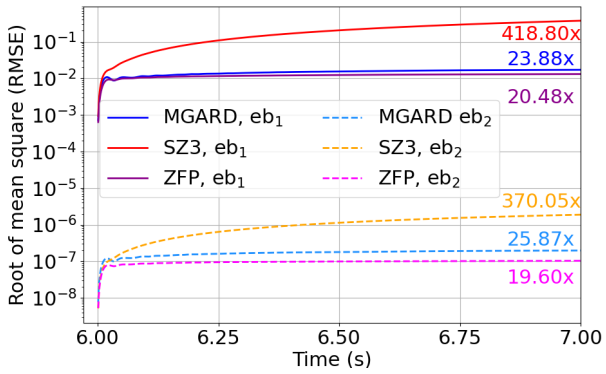


Figure 1: Error propagation in a 2D wave equation starting from checkpoints compressed using three lossy compressors respectively, with a relative eb_1=1e-3 and eb_2=1e-8. Both MGARD and ZFP show an initial spike in the error which quickly settles down, but the error propagation of SZ keeps increasing, even after thousands of time steps ($\Delta t = 1 \times 10^{-3}$).

2.3 Error propagation from lossy state

Despite the central role of PDEs in scientific computing, the stability implications of lossy compression have received limited attention. Compression-induced energy growth—if not properly controlled—can destabilize simulations and lead to divergence. Early work by Calhoun et al. [8, 9] applied lossy compression to checkpointing in Taylor Series, 1D heat, and advection equations. They showed that when the compression tolerance was set much smaller than the numerical truncation error, its impact was masked. However, these studies focused on small-scales, highly dissipative problems with relatively large truncation errors (10^{-4} to 10^{-3}), limiting their broader applicability. Fox et al. [21] analyzed inline ZFP compression in Krylov solvers, concluding that even small floating-point perturbations can amplify across iterations. Similarly, Tao et al. [59] demonstrated that lossy checkpointing can significantly degrade the convergence of iterative linear methods if not carefully managed.

A more directly relevant study by Kukreja et al. [36] examined error propagation in a 3D acoustic wave simulation with lossy checkpointing. They observed a sharp spike in error immediately after restart, followed by partial re-stablization—a behavior consistent with our own observations (Figure 1)—but without addressing the physics-specific mechanisms driving this growth. While some degree of stabilization is expected as no further perturbations are introduced post-restart, the magnitude and unpredictability of error amplification raise concerns about the reliability of so-called "error-controlled" compression. Complementing these findings, Triantafyllides et al. [64] demonstrated that variability across compressors and workflows, underscoring the fragility of approaches that rely solely on abstract error norms..

To illustrate how lossy checkpointing impacts PDE stability in hyperbolic systems, we simulate the motion of a 2D pond surface responding to raindrop impact. The wave equation is solved on a 2048×2048 m^2 domain with space and time discretization satisfying the Courant-Friedrichs-Lewy (CFL) condition [15]. After running the simulation for 6 seconds, we compress the final wave field using MGARD, SZ3, and ZFP at two error tolerance levels, $eb_1 = 10^{-3}$ and $eb_2 = 10^{-8}$, which yield relative L^2 errors of similar magnitude. MGARD is used with L^2 control, SZ3 with L^∞, and ZFP with fixed-accuracy mode, which are the default error-control settings of each compressor. Compression ratios across all compressors were matched by slightly adjusting error bounds, resulting with a ratio of 29× for eb_1 and and 5.5× for eb_2.

Restarting from these compressed checkpoints, we compared each run against a reference restarted from uncompressed data. All three compressors exhibit a spike in error immediately after restart. For MGARD and ZFP, errors plateau, while SZ3 continues to show gradual growth. Overall, root-of-mean-square errors (RMSE) increase by a factor of $20 - 419\times$ over 1000 time steps across all cases, regardless of the error tolerance used. Although some simulation may eventually stabilize (e.g., checkpointing compressed using MGARD and ZFP), such error amplification—especially in simulations involving multiple restarts—can undermine the reliability of scientific conclusions. These findings reinforce prior observations [8, 9, 21, 36, 59, 64], while highlighting that the root cause lies in the neglect of underlying physics—such as kinetic and potential energy conservation—during compression.

3 Problem Overview

The wave equation is widely used to model wave propagation across various physical systems, including acoustics, seismic waves, ocean waves dynamics, and electromagnetic waves transmission [12, 14, 34, 49, 65]. A standard 2D wave equation with periodic boundary conditions over a domain of size $L_x \times L_y$ is given by:

$$\begin{cases} \frac{\partial^2 u}{\partial t^2} - c^2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = S(x,y,t), \\ u(x + L_x, y, t) = u(x,y,t), \\ u(x, y + L_y, t) = u(x,y,t), \end{cases} \tag{1}$$

where $u(x,y,t)$ denotes the wave displacement at spatial coordinates (x,y) and time t, c is the wave speed (potentially varying over space, i.e., $c(x,y)$), and $S(x,y,t)$ is a time-dependent source term. When a source is absent, the initial condition $u(x,y,0) = f(x,y)$ is specified. Periodic boundary conditions conserve linear momentum and energy, ensuring that compression-induced artifacts do not escape the computational domain.

Discretizing Equation 1 using a second-order finite difference scheme with spatial steps Δx, Δy and temporal step Δt, the update formula becomes [37]:

$$\frac{u_{i,j}^{n+1} - 2u_{i,j}^n + u_{i,j}^{n-1}}{\Delta t^2} - c_{i,j}^2\left(\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{\Delta x^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{\Delta y^2}\right) = S_{i,j}^n,$$

where $u_{i,j}^n = u(i\Delta x, j\Delta y, n\Delta t)$. This scheme introduces temporal truncation error of $O(\Delta t^2)$ and spatial truncation error of $O(\Delta x^2 + \Delta y^2)$ for solving $\mathbf{u}^{n+1}$. To ensure numerical stability, the Courant-Friedrichs-Lewy (CFL) condition must be satisfied:

$$c\Delta t \leq 1/\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}}.$$

In this study, we assume the numerical truncation errors are negligible compared to errors introduced by lossy compression. For brevity, we omit the 3D extension of the wave equation.

Let $\mathbf{u}^n$ represents the solution $\{u_{i,j}\}^n$ obtained from an uncompressed checkpoint at time n, and $\tilde{\mathbf{u}}^n$ denote the counterpart reconstructed from a lossy compressed checkpoint. As shown in Equation 3, computing $\mathbf{u}^{n+1}$ requires checkpoint data from two time steps: $\mathbf{u}^n$ and $\mathbf{u}^{n-1}$. Conventional compressors provide bounded L^∞ or L^2 errors in each of these fields individually—i.e., $\mathbf{u}^n - \tilde{\mathbf{u}}^n$ and $\mathbf{u}^{n-1} - \tilde{\mathbf{u}}^{n-1}$. However, the impact of these errors on post-restart simulation depends on multiple hard-to-predict factors, including the type of PDEs being solved, the spatial and temporal error distribution, and the timing of the checkpoint. Empirical error management—such as arbitrarily selecting small error tolerance for compression—is often unreliable. Worse, when lossy checkpoints are used in multiple consecutive restarts, simulations can accumulate instabilities, leading to degraded accuracy or divergence. These observations motivate the need for a more principled approach to compression error control in PDE simulations.

4 Energy-conserved compression for checkpointing

In this section, we demonstrate that conserving energy during C/R is essential for maintaining numerical stability, accuracy, and physical fidelity in PDE simulations.

4.1 Importance of energy conservation for compression

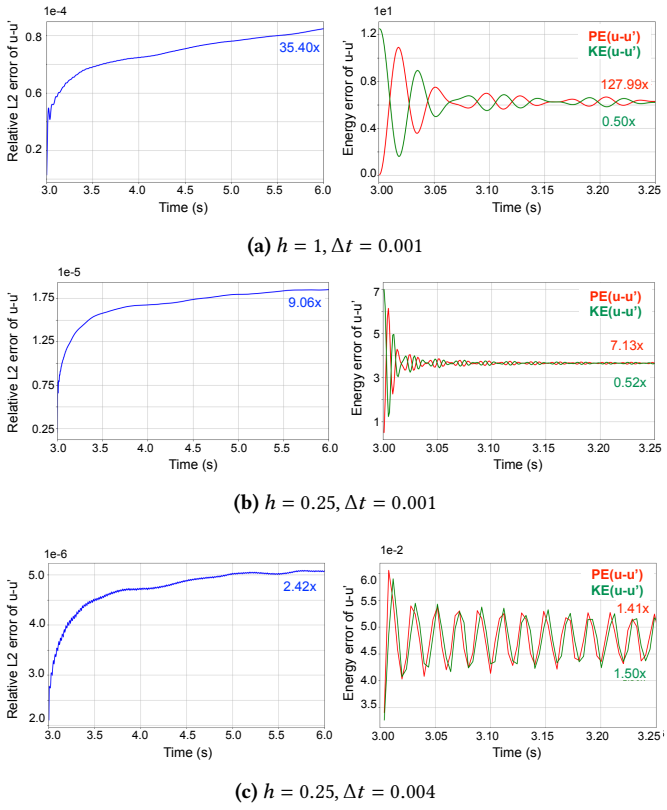
Energy conservation—the principle that total energy in an isolated system remains constant—is a core property of many PDEs. Quantization in lossy compression disrupts this balance by introducing artificial energy loss or gain. To ensure stability after restart, it is critical to understand and manage how compression errors impact system energy.

Conventional approaches compress $\mathbf{u}^n$ and $\mathbf{u}^{n-1}$ independently, typically using L^2 or L^∞ error bounds. However, these norms do not directly control the kinetic energy (**KE**) and potential energy (**PE**), which are essential for PDEs. For hyperbolic PDEs like the wave equation, which propagate forward in time based on past states, it is important to monitor the evolution of energy. We define:

$$\mathrm{PE}(\mathbf{u}^n) = \frac{1}{2}\sum_{i,j}\left[\left(\frac{u^n_{i+1,j}-u^n_{i,j}}{\Delta x}\right)^2 + \left(\frac{u^n_{i,j+1}-u^n_{i,j}}{\Delta y}\right)^2\right]\Delta x\Delta y, \quad (2)$$

$$\mathrm{KE}(\mathbf{u}^n, \mathbf{u}^{n-1}) = \frac{1}{2c^2}\sum_{i,j}\left(\frac{u^n_{i,j}-u^{n-1}_{i,j}}{\Delta t}\right)^2\Delta x\Delta y, \quad (3)$$

The quantity of $\mathrm{PE}(\mathbf{u}^n)$ clearly corresponds to the potential energy at the n^{th} timestep. However, the quantity of $\mathrm{KE}(\mathbf{u}^n, \mathbf{u}^{n-1})$ is more naturally associated with the kinetic energy at the $n-\frac{1}{2}$ timestep. In order to make the KE and PE comparable, we shall consider the modified PE defined by $\mathrm{PE}(\mathbf{u}^{n-\frac{1}{2}})$, where $\mathbf{u}^{n-\frac{1}{2}}$ is defined as $\frac{1}{2}(\mathbf{u}^n + \mathbf{u}^{n-1})$. The perturbation to energy conservation incurred by lossy compression is quantified as the energy in the error field, i.e. $\mathbf{e} = \mathbf{u} - \tilde{\mathbf{u}}$, before and after compression.



(a) $h = 1, \Delta t = 0.001$

(b) $h = 0.25, \Delta t = 0.001$

(c) $h = 0.25, \Delta t = 0.004$

Figure 2: Illustration of the relationship between the energy oscillation and the resulting L^2 error magnification when a simulation restarts from a lossy compressed checkpoint at $t = 3$ sec. A 2D waves equation is simulated within a physical space of 1024×1024 unit size and the wave continuously propagates for another 3 sec after restart. Different spatial (h) and temporal (Δt) discretization are selected to adjust KE and PE's sensitivity to compression-incurred perturbation.

Figure 2 shows an illustrative example of how KE, PE, and the L^2 error evolves in raindrop-on-pond simulation with varying spatial and temporal resolutions. We model a physical domain of size 1024×1024 and a total simulation time of 6 sec. Across different discretizations, checkpoints were compressed at the same simulated time, $t = 3$ sec, using MGARD with L^2 error control and a **relative** RMSE (i.e., $\|\mathbf{u} - \tilde{\mathbf{u}}\|/(\max(\mathbf{u}) - \min(\mathbf{u}))$) of 1e-4. Since the energy of error typically resettles to a steady-state level after the first few hundreds of steps, we plot them only between $3 \le t \le 3.25$ sec. We observe: (1) Lossy checkpoints perturb KE and PE, leading to oscillate untill convergence. (2) Greater imbalance in KE and PE at restart prolongs error convergence period. (3) Even after energy stabilizes, L^2 errors may continue growing slowly, potentially leading to divergence over time.

4.2 Kinetic/Potential energy-aware compression

The GOAL of compression is to replace $\mathbf{u}^n$ and $\mathbf{u}^{n-1}$ using their compressed versions that require less storage, while minimizing

perturbations to KE($\mathbf{u}^n, \mathbf{u}^{n-1}$) and PE($\mathbf{u}^{n-\frac{1}{2}}$). We achieve this by balancing the KE($\mathbf{e}^n, \mathbf{e}^{n-1}$) and PE($\mathbf{e}^{n-\frac{1}{2}}$) during lossy checkpointing, and simultaneously minimize their magnitudes.

First step, we introduce **new variables** defined by:

$$\mathbf{u}^D = \frac{1}{2}(\mathbf{u}^n - \mathbf{u}^{n-1}), \text{ and } \mathbf{u}^A = \frac{1}{2}(\mathbf{u}^n + \mathbf{u}^{n-1}). \tag{4}$$

So that the total energy becomes:

$$\text{KE}(\mathbf{u}^n, \mathbf{u}^{n-1}) + \text{PE}(\mathbf{e}^{n-\frac{1}{2}}) = \text{KE}(\mathbf{u}^D) + \text{PE}(\mathbf{u}^A) \tag{5}$$

So the goal of the design is to choose compressed versions of $\mathbf{u}^D \approx \tilde{\mathbf{u}}^D$ and $\mathbf{u}^A \approx \tilde{\mathbf{u}}^A$ while minimizing the perturbation in the energy measured by

$$\text{KE}(\mathbf{u}^D - \tilde{\mathbf{u}}^D) + \text{PE}(\mathbf{u}^A - \tilde{\mathbf{u}}^A). \tag{6}$$

Once $\tilde{\mathbf{u}}^D$ and $\tilde{\mathbf{u}}^A$ are decompressed, we can transform them back to obtain compressed versions of $\tilde{\mathbf{u}}^n$ and $\tilde{\mathbf{u}}^{n-1}$ given by the formula of

$$\tilde{\mathbf{u}}^n = \tilde{\mathbf{u}}^A + \tilde{\mathbf{u}}^D, \text{ and } \tilde{\mathbf{u}}^{n-1} = \tilde{\mathbf{u}}^A - \tilde{\mathbf{u}}^D. \tag{7}$$

4.3 Error control via s-norm

The problem now becomes to device a compression technique which minimizes the quantity defined in Equation 5. This is a non-standard compression problem which requires to simultaneously control L^2 norm in the first term and a derivative-based norm in the second term. Fortunately, the MGARD compression technique has the capability to control both of these quantities defined by:

$$\epsilon_{PE} = \frac{1}{h^2}\|\mathbf{u}^A - \tilde{\mathbf{u}}^A\|_1, \text{ and } \epsilon_{KE} = \frac{1}{c^2\Delta t^2}\|\mathbf{u}^D - \tilde{\mathbf{u}}^D\|_0. \tag{8}$$

4.3.1 S-norms of MGARD. MGARD is an error-controlled compressor based on the theory of multigrid. Given an input array $\mathbf{u}$ defined on a grid N_L in $\mathbb{R}^d$, MGARD decomposes data into multilevel coefficients $\texttt{u\_mc[x]}^l$ residing on subgrids $N_{L-1}, ..., N_0$ downsampled from N_L. The compression error is incurred by quantizing $\texttt{u\_mc[x]}^l$ to $\texttt{ũ\_mc[x]}^l$ for $x \in N_L$. Based on the previous work in [2], MGARD defines a **s-norm** and uses a family $\{\|.\|_s : s \in \mathbb{R}\}$ to measure the loss in $\mathbf{u}-\tilde{\mathbf{u}}$. These norms have been encapsulated into a specialized quantizers used for adaptively reducing $\texttt{u\_mc[x]}^l$ across different subgrid N_l. Under appropriate conditions, MGARD's s-norm are equivalent [5, 54] to the usual Sobolev norms $\{\|.\|_{H^s} : |s| < \frac{3}{2}\}$. In particular, we observe that the s-norm $\|.\|_s$ with $s = 0$ is identical to the L^2 norm $\|.\|_{L^2}$, and $\|.\|_s$ with $s = 1$ is equivalent to the norm of potential energy. Previous work has leverage MGARD's L^2 error control [1, 23, 25, 41, 42], but our study is the first to explicitly exploit the connection between PE, L^2, and the $\|.\|_1$ norm of MGARD.

First, we analyze the distribution of errors in data compressed using different s-norms of MGARD. Figure 3 presents the results of reducing $\mathbf{u}$, which is a wave field vector, by a compression ratio of 16× and taking s to be either 0 or 1. The simulation models the evolution of a Gaussian pulse signal within layered material space (*Flat Velocity* in Figure 6). We computed both $L^2(\mathbf{u}-\tilde{\mathbf{u}})$ (i.e., middle row in Figure 3) and PE($\mathbf{u}-\tilde{\mathbf{u}}$) (i.e., bottom row in Figure 3).The results show that using MGARD with $s = 0$ minimizes L^2 errors (i.e., $\|\mathbf{u}-\tilde{\mathbf{u}}\|_0$ norm), while $s = 1$ minimizes the PE($\mathbf{u}-\tilde{\mathbf{u}}$) error ($\|\mathbf{u}-\tilde{\mathbf{u}}\|_1$ norm), under equal compression ratios. This results demonstrate

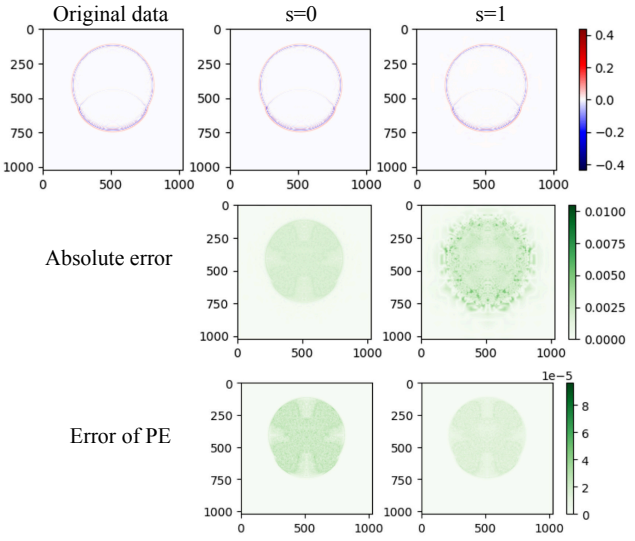


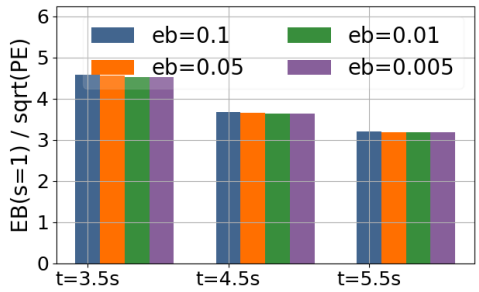
Figure 3: Illustration of the error preservation using MGARD $s = 0$ and $s = 1$. The top row shows the original wave field (left) and the reconstructed wave fields (middle and right). The middle row shows the absolute errors ($\|\mathbf{u} - \tilde{\mathbf{u}}\|$), and the bottom row shows PE of compression errors ($\nabla(\mathbf{u} - \tilde{\mathbf{u}})$) resulting from the compression using $s = 0$ and $s = 1$.

an unique feature of MGARD's compression under different norms, which has not been seen in other state-of-the-art compressors.

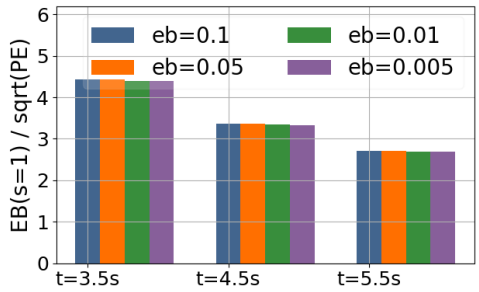
4.3.2 Energy-conserved compression. Based on the MGARD foundations and its relation to Sobolev norms, we observe the potential energy of errors achieved using MGARD $s = 1$ is upper bounded by a constant, C_{PE}, such that $\text{PE}(\mathbf{u}) = \|\mathbf{u}-\tilde{\mathbf{u}}\|_1 \leq \tau_{PE}/C_{\text{PE}}$, where C_{PE} depends on the shape of the domain, but is independent of $\mathbf{u}$ and $\tilde{\mathbf{u}}$, the domain size and the number of grid points. Figure 4 presents the ratio of τ_{PE} to the corresponding PE($\mathbf{u}-\tilde{\mathbf{u}}$) using MGARD $s = 1$. We executed the simulation using two initial conditions—rain drops landing on pond surface and a Gaussian pulse signal traversing a 2D space initialized using the velocity map *Curve Fault* in Figure 6. We then reduce the solutions obtained at different checkpointing time using $\tau_{\text{PE}} = 0.1$, 0.05, 0.01, and 0.005. Across different error tolerance and simulation configurations, C_{PE} remains within a relatively constant range. Although a new C_{PE} may need to be measured when switching physical problems, results in Figure 4 demonstrate that the same C_{PE} can be applied to different configurations, initial conditions, and checkpoint time.

As we discussed in Section 4.2, the new checkpoints variables, $\mathbf{u}^A$ and $\mathbf{u}^D$, should be compressed using different error controls, with L^2 for $\mathbf{u}^D$ and potential energy for $\mathbf{u}^A$, which correspond to the $s = 0$ and $s = 1$ norms of MGARD. Here, we denote tolerance used for $\mathbf{u}^D$ and $\mathbf{u}^A$ as τ_{KE} and τ_{PE}, with

$$\|\mathbf{u}^D - \tilde{\mathbf{u}}^D\|_0 \leq \tau_{\text{KE}} \text{ and } \|\mathbf{u}^A - \tilde{\mathbf{u}}^A\|_1 \leq \tau_{\text{PE}}/C_{\text{PE}}.$$



(a) Rain drops in a 2048×2048 space (uniform velocity)



(b) Pulse source in a 4096×4096 space (Curve Fault)

Figure 4: Demonstration of MGARD's error control on potential energy (PE). eb refers to the error tolerance using MGARD $s = 1$ norm, and PE measures the potential of compression error, i.e., PE(u − ũ). The wave fields are compressed at t =3.5s, 4.5s, and 5.5s in each simulation. The ratio of τ_{PE} to PE(e) remains constant across different error tolerance, domain sizes, and simulation initial conditions.

To balance KE error and PE error defined in Equation 8, we derive the following **relation**:

$$\frac{\tau_{KE}}{\tau_{PE}} = \frac{c^2 \Delta t^2 C_{\text{PE}}}{h^2}. \tag{9}$$

4.3.3 Relation between PE and L^2 errors. So far, we have only discussed controlling the error in energy. In fact, as we shall now show, this also provides indirect control of the errors in L^2. The corresponding L^2 errors in $\mathbf{u}^n$ is given by:

$$\begin{aligned} \|\mathbf{u}^n - \tilde{\mathbf{u}}^n\|_{l^2} &= \|(\mathbf{u}^A - \tilde{\mathbf{u}}^A) + (\mathbf{u}^D - \tilde{\mathbf{u}}^D)\|_{l^2} \\ &\leq \tau_{KE} + \|\mathbf{u}^A - \tilde{\mathbf{u}}^A\|_{l^2}. \end{aligned} \tag{10}$$

Below, we prove there exhibits a constant γ, such that $\|\mathbf{u}^A - \tilde{\mathbf{u}}^A\|_{l^2} \leq \gamma \|\mathbf{u}^A - \tilde{\mathbf{u}}^A\|_1$.

Definition 4.1. Suppose we have a grid with nodes located at $x_j = jh$, $j = 0, 1, ..., N$. For each $m \in \{1, 2, ..., N-1\}$, we define a vector of sinusoidal functions $\theta^{(m)} = (\theta_0^{(m)}, \ldots, \theta_N^{(m)})$, where $\theta_j^{(m)} = \sin \frac{\pi jm}{N}$ for $\forall j \in \{0, \ldots, N\}$. This vector has has the following properties:

(1) P1: $\theta_0^{(m)} = \theta_N^{(m)} = 0$ for all m.
(2) P2: Since $\theta^{(m)}$ and $\theta^{(n)}$ are orthogonal when $m \neq n$:

$$\theta^{(m)} \cdot \theta^{(n)} = \sum_{j=0}^{N} \theta_j^{(m)} \theta_j^{(n)} = \begin{cases} 0, & \text{with } m \neq n \\ \frac{1}{2}N, & \text{with } m = n \end{cases}$$

(3) P3: $-\theta_{j+1}^{(m)} + 2\theta_j^{(m)} - \theta_{j-1}^{(m)} = 4 \sin^2 \frac{\pi m}{2N} \theta_j^{(m)}$

Proof. Let $L_{p3} = -\theta_{j+1}^{(m)} + 2\theta_j^{(m)} - \theta_{j-1}^{(m)}$

$$\begin{aligned} L_{P3} &= -\sin(\frac{\pi jm}{N} + \frac{\pi m}{N}) + 2\sin(\frac{\pi jm}{N}) - \sin(\frac{\pi jm}{N} - \frac{\pi m}{N}) \\ &= 2\sin(\frac{\pi jm}{N})(-\cos(\frac{\pi m}{N}) + 1) \\ &= 4 \sin^2 \frac{\pi m}{2N} \theta_j^{(m)}, \end{aligned}$$

□

(4) P4: $\sum_{j=0}^{N-1} (\theta_{j+1}^{(m)} - \theta_j^{(m)})^2 = 2N \sin^2 \frac{\pi m}{2N}$.

Proof. Let $L_{P4} = \sum_{j=0}^{N-1} (\theta_{j+1}^{(m)} - \theta_j^{(m)})^2$,

$$\begin{aligned} L_{P4} &= \sum_{j=0}^{N-1} \theta_{j+1}^{(m)} (\theta_{j+1}^{(m)} - \theta_j^{(m)}) - \sum_{j=0}^{N-1} \theta_j^{(m)} (\theta_{j+1}^{(m)} - \theta_j^{(m)}) \\ &= \sum_{j=1}^{N-1} \theta_j^{(m)} (-\theta_{j+1}^{(m)} + 2\theta_j^{(m)} - \theta_{j-1}^{(m)}) \\ &= \sum_{j=1}^{N-1} 4 \sin^2 \frac{\pi m}{2N} (\theta_j^{(m)})^2 \\ &= 2N \sin^2 \frac{\pi m}{2N} \quad \texttt{(use P3, P2)} \end{aligned}$$

□

With above, we prove the L^2 norm can be controlled by PE through the following theorem.

Theorem 4.2. *Let $\{u_j\}_{j=0}^N \in \mathbb{R}$: $u_0 = u_N = 0$, then*

$$\sum_{j=0}^{N} u_j^2 \leq \frac{N^2 h^2}{\pi^2} \{1 + O(N^{-2})\} PE(\boldsymbol{u})$$

Proof. Since $\{\theta^{(m)}\}$ are linearly independent, we could write $\{u_j\}$ using $\{\theta^{(m)}\}$ as the bases according to P1 and P2:

$$u_j = \sum_{n=1}^{N-1} \hat{u}_n \theta_j^{(n)}, \forall j \in \{0, \ldots, N\}$$

where $\hat{u}_n = \frac{2}{N} \sum_{j=0}^{N-1} u_j \theta_j^{(m)}$, for $n \in \{1, ..., N-1\}$. Then we have:

$$\sum_{j=1}^{N-1} u_j^2 = \sum_{m,n=1}^{N-1} \hat{u_m} \hat{u_n} \sum_{j=1}^{N-1} \theta_j^{(m)} \theta_j^{(n)} = \frac{N}{2} \sum_{m=1}^{N-1} \hat{u}_m^2 \quad \texttt{(use P2)}$$

As $u_{j+1} - u_j = \sum_{m=1}^{N-1} \hat{u}_m(\theta_{j+1}^{(m)} - \theta_j^{(m)})$, we further have:

$$\begin{aligned}
\sum_{j=0}^{N-1}(u_{j+1}-u_j)^2 &= \sum_{m=1}^{N-1}\hat{u}_m^2 \sum_{j=0}^{N-1}(\theta_{j+1}^{(m)}-\theta_j^{(m)})^2 \qquad \texttt{(use P2)} \\
&= \sum_{m=1}^{N-1}\hat{u}_m^2 \cdot 2N\sin^2\frac{m\pi}{2N} \\
&\geq 2N\sin^2\frac{\pi}{2N}\sum_{m=1}^{N-1}\hat{u}_m^2 \qquad \texttt{(relaxation on } m\texttt{)} \\
&= 4\sin^2\frac{\pi}{2N}\sum_{n=1}^{N-1}u_n^2 \quad \texttt{(use previous equation)} \\
&= 2(1-\cos\frac{\pi}{N})\sum_{n=1}^{N-1}u_n^2 \\
&= \frac{\pi^2}{N^2}(1+O(N^{-2}))\sum_{n=1}^{N-1}u_n^2 \quad \texttt{(Taylor series)}
\end{aligned}$$

As $\mathrm{PE}(\mathbf{u}) = \sum_{k=0}^{N-1}(\frac{u_{k+1}-u_k}{h})^2$, we have the following relationship:

$$\begin{aligned}
\sum_{n=1}^{N-1}u_n^2 &\leq \frac{N^2}{\pi^2}(1+O(N^{-2}))\sum_{k=0}^{N-1}(u_{k+1}-u_k)^2 \\
&= \frac{N^2h^2}{\pi^2}(1+O(N^{-2}))\sum_{k=0}^{N-1}(\frac{u_{k+1}-u_k}{h})^2 \\
&= \frac{N^2h^2}{\pi^2}(1+O(N^{-2}))\mathrm{PE}(\mathbf{u})
\end{aligned}$$

□

This theorem shows also holds for grid functions defined in $\mathbb{R}^d, d \in N$. We omit the proof here for brevity.

4.4 Design overview

The implementation of the lossy checkpointing-restart pipeline is conceptually straightforward and consists of the following steps:

(1) **Variable transformation:** Convert the checkpoint variables $\mathbf{u}^n$ and $\mathbf{u}^{n-1}$ into a pair of new variables $\mathbf{u}^D$ and $\mathbf{u}^A$, to enable separate controls over $\mathrm{PE}(\mathbf{u}^{n-\frac{1}{2}})$ and $\mathrm{KE}(\mathbf{u}^n - \mathbf{u}^{n-1})$.
(2) **Error balancing:** Determine and balance the error tolerance τ_{KE} and τ_{PE} based on the relations defined in Equation 9 and 4.2.
(3) **Kinetic energy compression:** Compress $\mathbf{u}^D$ to $\tilde{\mathbf{u}}^D$ using MGARD with $s = 0$ (i.e., $\|.\|_0$) and the tolerance τ_{KE}
(4) **Potential energy compression:** Compress $\mathbf{u}^A$ to $\tilde{\mathbf{u}}^A$ using MGARD with $s = 1$ (i.e., $\|.\|_1$) and τ_{PE}.

During reconstruction, $\tilde{\mathbf{u}}^D$ and $\tilde{\mathbf{u}}^A$ are first decompressed, then recombined to recover $\tilde{\mathbf{u}}^n$ and $\tilde{\mathbf{u}}^{n-1}$

Figure 5 illustrates the full pipeline for stability-preserving lossy checkpointing and restart. Compression parallelism is achieved through data chunking, a feature available in modern parallel I/O libraries such as HDF5 [39] and ADIOS [38]. The pipeline consists of two stages:

- **Offline stage:** Executes a coarse-grained simulation of the target problem to estimate C_{PE}, a constant required for controlling PE error via MGARD with $s = 1$. The coarse simulation is computationally efficient and performed at lower resolution. As shown in Figure 4, the ratio $\tau_{\mathrm{PE}}/\mathrm{PE}(\mathbf{u} - \tilde{\mathbf{u}})$ is used to estimate C_{PE}. Once estimated, C_{PE} remains stable across different initial conditions, discretizations, domain sizes, and error tolerances.
- **Online stage:** Performs lossy compression at runtime, A key component of the online stage is the *error tolerance estimator*, which translates the user-provided error bound (in either L^2 norm, KE, or PE) into the corresponding values of τ_{KE} and τ_{PE} for compressing $\mathbf{u}^A$ and $\mathbf{u}^D$. using Equations 10 and 4.2. For PDEs in heterogeneous media, the wave speed vector $\mathbf{c}$ is replaced with an averaged value $\bar{c}$ over regions where the solution is significant.

Notably, this pipeline introduces negligible computational overhead. Instead of compressing $\mathbf{u}^n$ and $\mathbf{u}^{n-1}$ directly, it compresses $\mathbf{u}^D$ and $\mathbf{u}^A$—a simple transformation with minimal cost.

Algorithm 1: Error tolerance assignment

Input: Requested eb_{TYPE} and τ_n, C_{PE}, grid points N, physical domain size M, averaged velocity $\bar{c}$

if $eb_{TYPE} \in L^2$ **then**
  $\tau_{\mathrm{PE}} \leftarrow \tau_n \Big/ \left(\frac{M^2}{\pi^2} + \frac{\bar{c}^2\Delta t^2 C_{\mathrm{PE}}}{h^2}\right)$;
  $\tau_{\mathrm{KE}} \leftarrow \tau_{\mathrm{PE}}\frac{\bar{c}^2\Delta t^2 C_{\mathrm{PE}}}{h^2}$;
else if $eb_{TYPE} \in \tau_{KE}$ **then**
  $\tau_{\mathrm{PE}} \leftarrow \tau_{\mathrm{n}}\frac{h^2}{\bar{c}^2\Delta t^2 C_{\mathrm{PE}}}$;
  $\tau_{\mathrm{L2}} \leftarrow \frac{M^2}{\pi^2}\tau_{PE} + \tau_{\mathrm{n}}$;
else if $eb_{TYPE} \in \tau_{PE}$ **then**
  $\tau_{\mathrm{KE}} \leftarrow \tau_{\mathrm{n}}\frac{\bar{c}^2\Delta t^2 C_{\mathrm{PE}}}{h^2}$;
  $\tau_{\mathrm{L2}} \leftarrow \frac{M^2}{\pi^2}\tau_n + \tau_{\mathrm{KE}}$;
else
  Input error type not supported ;
return τ_{PE}, τ_{KE}, τ_{L2};

5 Evaluation

We evaluate the proposed pipeline in terms of the ability to (1) minimize compression-incurred perturbations in PDE checkpointing and restart, (2) generalize across diverse initial conditions and simulation configurations, and (3) maintain stability across multiple C/R cycles. Additionally, we assess its integration into a parallel I/O pipeline. Given an error tolerance specified on final solution accuracy (i.e., post-error magnification), we demonstrate that our method delivers superior I/O performance compared to both uncompressed and standard lossy checkpointing schemes. We refer to our method as *MGARD-energy*, and to the baseline MGARD implementation with conventional L^2 error control as *MGARD-l2*.

5.1 Experiment setup

5.1.1 Experimental environment. All experiments were conducted on Frontier [53], a leadership class supercomputer at Oak Ridge

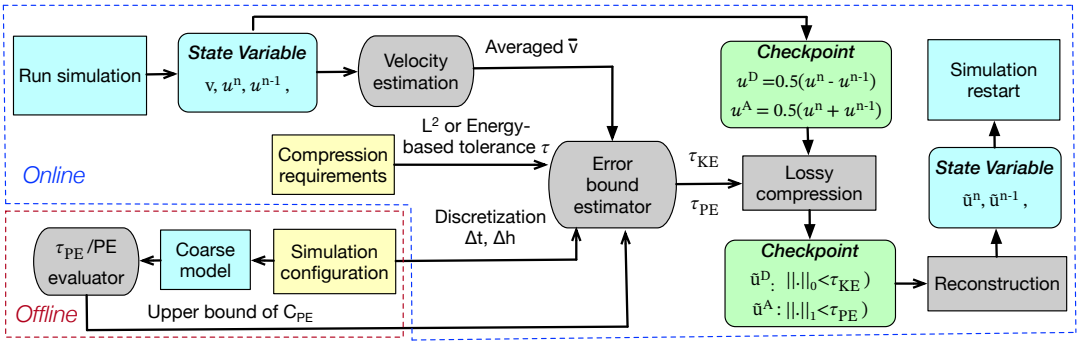


Figure 5: Overview of stability-preserving lossy compression pipeline for PDEs.

Leadership Computing Facility (OLCF). Frontier consists of 9,408 nodes, each equipped with 4 AMD Instinct MI250X GPUs (128 GB memory each) and a 64-core AMD EPYC CPU (512 GB memory). The file system is Lustre-based with a peak I/O bandwidth of 9.4 TB/s.

5.1.2 Reference problem. We implemented a 2D and 3D wave propagation simulation based on the finite difference scheme described in Equation 3 with periodic boundary conditions. Velocity models were initialized using velocity maps from the *Vel* and *Fault* families of the OPENFWI dataset [16], a multi-structural benchmark suite for full waveform inversion (FWI). We modified the *CurveFault* map by adding non-absorbing obstacles to simulate wave scattering. The original map size (70×70) was upsampled via nearest-neighbor interpolation and rescaled to the velocity range $[0, 275]$ m/s. For 3D experiments, the upsampled 2D slices were duplicated along the z-axis. Dataset dimensions are detailed in Table 1. Simulations were discretized spatially using $h = 1$, with time steps $\Delta t = 5 \times 10^{-4}$ sec (2D experiments) and $\Delta t = 1 \times 10^{-3}$ sec (3D experiments).

For velocity models from OPENFWI, we introduced an explosive source function, $s(t)$, defined as the derivative of a Gaussian pulse:

$$s(t) = -2\alpha(t - t_0)e^{-\alpha(t-t_0)^2}.$$

where $t_0 = 0.1$ sec. The source was placed in the center of the domain and deactivated after 0.25 sec. We also modeled a 3D Gaussian spherical wave propagating through a uniform medium. Figure 6 shows fours velocity maps and their corresponding wave fields at checkpointing time in single C/R tests. As presented in Table 1, for all 2D cases, simulations ran until $t = 4.5$ sec, then resumed from checkpoint and continued for 2.5 sec. In 3D cases, simulations ran for 4 sec total, with a checkpoint saved at $t = 2$ sec and $t = 3$ sec for *CurveFault* and *Uniform* models, respectively.

5.1.3 Compressors and error metrics. We compare our method against standard lossy checkpointing approaches that compress $\mathbf{u}^n$ and $\mathbf{u}^{n-1}$ separately. Specifically, we use:

- **SZ3**: L^∞ and the default `interpolation+lorenzo` configuration
- **ZFP**: Fixed-accuracy mode (equivalent to L^∞ control).
- **MGARD**: Standard L^2 error control mode.

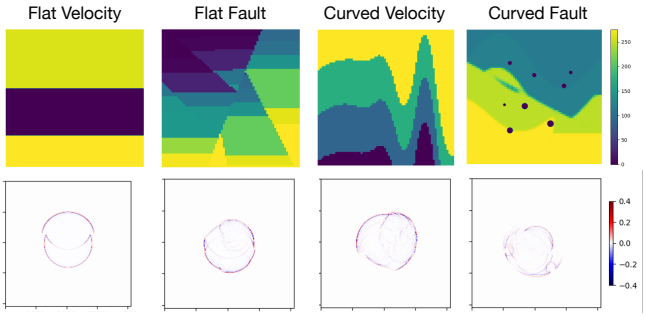


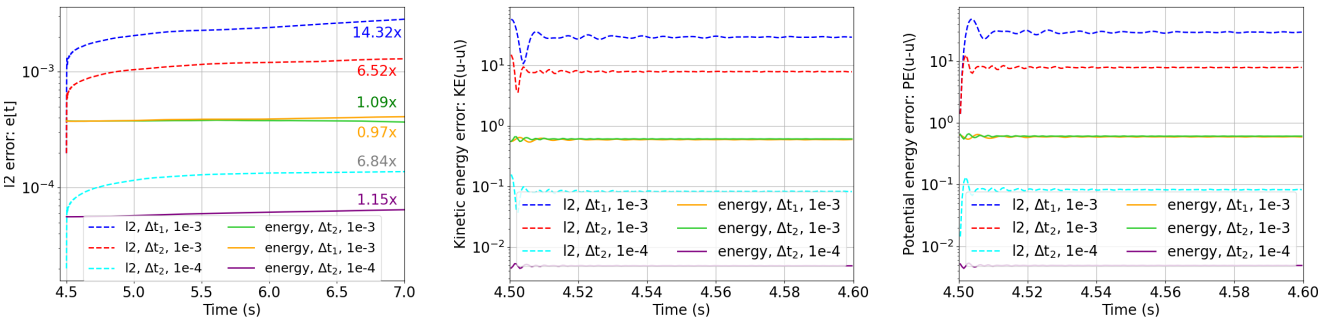
Figure 6: Velocity maps (top row) used in simulation and the wave fields (bottom row) in 2D at the lossy checkpointing.

Velocity	2D	3D
Uniform		$900 \times 900 \times 900$, 3k/1k ts
FlatVel	4096×4096, 9k/5k ts	
FlatFault	4096×4096, 9k/5k ts	
CurvedVel	4096×4096, 9k/5k ts	
CurvedFault	4096×4096, 9k/5k ts	$512 \times 512 \times 512$, 2k/2k ts

Table 1: Velocity maps, domain size, and the number of timesteps (ts) running before and post C/R.

We use the CPU implementation of SZ3 [60], ZFP [61], and MGARD [62]. For lossy checkpointing and I/O throughput evaluation, we use the GPU implement of MGARD, the only one among the three compressors portable to AMD GPUs. Because MGARD employs a portable implementation, the compression ratios and achieved errors are similar between CPUs and GPUs.

We assess the difference between the reference field $\mathbf{u}^n$ and the decompressed field $\tilde{\mathbf{u}}^n$, using three metrics: root-of-mean-square error (RMSE), KE($\mathbf{u}^n - \tilde{\mathbf{u}}^n$), and PE($\mathbf{u}^n - \tilde{\mathbf{u}}^n$). To measure long-term numerical stability, we define **Error Magnification (EM)** as the ratio between the final error after N simulation steps and the initial error at restart. An EM close to 1 indicates that the compression-induced perturbation remains bounded, suggesting minimal disruption to the physical system.



(a) RMSE **(b) KE($\mathbf{u}-\tilde{\mathbf{u}}$)** **(c) PE($\mathbf{u}-\tilde{\mathbf{u}}$)**

Figure 7: Illustration of error propagation when restarted simulation from checkpoints compressed using different error tolerances under two numerical discretization ($\Delta t_1 = 5\times10^{-4}$ and $\Delta t_2 = 1\times10^{-3}$).

5.2 Error propagation after a single C/R

First, we demonstrate that our method enables stable restart regardless of compression error tolerance or numerical discretization. We use the 2D *CurveVel* velocity model with three configurations: (1) $\Delta t_1 = 5\times10^{-4}$ and a RSME tolerance of 1×10^{-3}; (2) $\Delta t_2 = 1\times10^{-3}$ and a RSME tolerance of 1×10^{-3}; (3) $\Delta t_3 = 1\times10^{-3}$ and a RSME tolerance of 1×10^{-4}. The associated error tolerances follow the relations $\sqrt{\tau_{\text{KE1}}} = \frac{1}{2}\sqrt{\tau_{\text{KE2}}} = 5\sqrt{\tau_{\text{KE3}}}$ and $\sqrt{\tau_{\text{PE1}}} = \sqrt{\tau_{\text{PE2}}} = 10\sqrt{\tau_{\text{PE3}}}$. We adjusted the MGARD-l2 tolerance to achieve comparable compression ratios: 52× for case (1) and (2), and 34× for case (3). Figure 7 shows that restarts from MGARD-energy-compressed checkpoints exhibit minimal disturbance, with $EM \approx 1$ across RSME, KE, and PE. We plot KE and PE errors only in the first 0.1 sec after restart, as they stabelize quickly. For case (1) and (2), MGARD-energy yield similar errors, obustness against discretization changes. In contrast, RMSE with MGARD-l2 increases by 14.3× and 6.5× for case (1) and (2), respectively. Furthermore, RSME magnification with MGARD-l2 remains largely unchanged between cases (2) and (3), suggesting no clear correlation between error tolerance and downstream error propagation.

We also evaluate other compressors using a 2D simulation initialized with *CurveFault* velocity model. Similar to the procedure above, we adjust the error tolerance of SZ3, ZFP, MGARD-l2, and MGARD-energy to achieve comparable compression ratios, which is ~ 70×. As shown in Figure 8, MGARD-energy shows the best stability ($EM \approx 1$), while ZFP and MGARD-l2 introduce 11 – 14× RMSE magnification, and SZ3 reaches a *EM* of 2894×. While KE and PE stabilize across all methods within 0.1 sec, the RMSE in checkpoints compressed by SZ3 continues to grow, indicating long-term instability. This difference in error behavior is likely due to the compressors' decorrelation strategies—SZ3 uses dynamic predictors, whereas ZFP and MGARD-l2 rely on global transforms. The final error values (relative to MGARD-energy) are as follows: (1) **RMSE**: SZ3=1292.3×, ZFP=8×, MGARD-l2=7.4×; (2) **KE**($\mathbf{u}-\tilde{\mathbf{u}}$): SZ3=58.6×, ZFP=73.4×, MGARD-l2=56.2×; (3) **PE**($\mathbf{u}-\tilde{\mathbf{u}}$): SZ3=59.6×, ZFP=73.4×, MGARD-l2=56.5×.

We repeat the experiment across additional 2D and 3D (Table 1). Figure 9 shows EM results over 5000 (2D), 2000 (*CurvedFault-3D*), and 1000 (*Uniform-3D*) time steps after restarts. As KE($\mathbf{u}-\tilde{\mathbf{u}}$) magnification remains relatively constant (~ 0.5 for SZ3/ZFP/MGARD-l2 and 1 for MGARD-energy), we omit its plot. Across all five cases, MGARD-energy consistently achieves $EM \approx 1$ for RMSE, KE($\mathbf{u}-\tilde{\mathbf{u}}$), and PE($\mathbf{u}-\tilde{\mathbf{u}}$), while other methods yield varying and often unpredictable magnifications, especially in 3D. **Final RMSE** values relatives to MGARD-energy are summarized as follows. (1) *FlatVel-2D*: SZ3= 280.6×, ZFP=6×, MGARD-l2=6.2×; (2) *FlatFault-2D*: SZ3=1207.6×, ZFP=8.1×, MGARD-l2=6.3×; (3) *CurveVel-2D*: SZ3=307.5×, ZFP=3.7×, and MGARD-l2=6.9×; (4) *Uniform-3D*: SZ3= 69.4×, ZFP=0.7×, and MGARD-l2=15.6×; (5) *CurveFault-3D*: SZ3= 1248.7×, ZFP=3.3×, and MGARD-l2=11.7×.

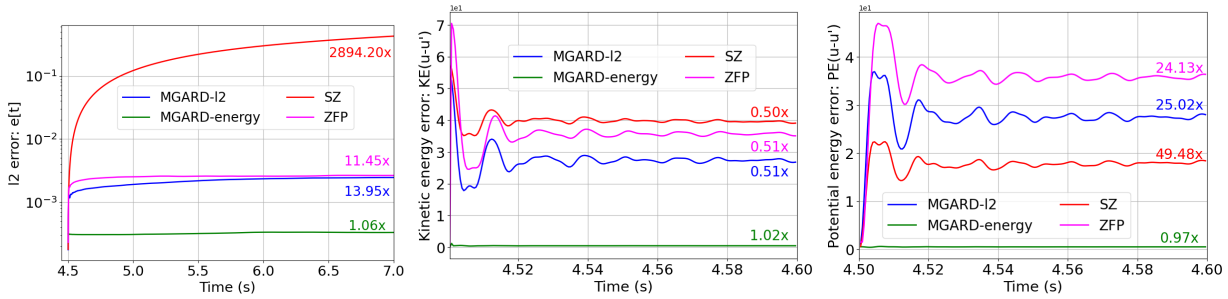
5.3 Error propagation post multiple C/R

We now evaluate error propagation under multiple sequential checkpoint/restarts. Using the same 2D *FlatFault* setup, we ran the simulation for 3000 steps and saved checkpoint cr_1. We then resumed from cr_1 for 4000 more steps, saved cr_2, and repeated this process four times, reaching $t = 10.5$ sec. Each checkpoint was compressed using the same τ_{KE} and τ_{PE} (equivalent to RMSE=1×10^{-3}) by MGARD-energy. For comparison, we repeated the process with MGARD-l2 using the same compression ratios achieved above.

Figure 10 plots RMSE over the full trajectory, across 4 consecutive C/Rs. MGARD-energy maintains stable error throughout each restart, though each new compression introducing additional errors. In contrast, MGARD-l2 exhibits cumulative RMSE upon the magnification of errors introduced by each checkpointing. Assuming $EM \approx 1$ with MGARD-energy and that new checkpoints will be reduced after a system resettled from the previous restart, the total error after p restart cycles can be estimated as

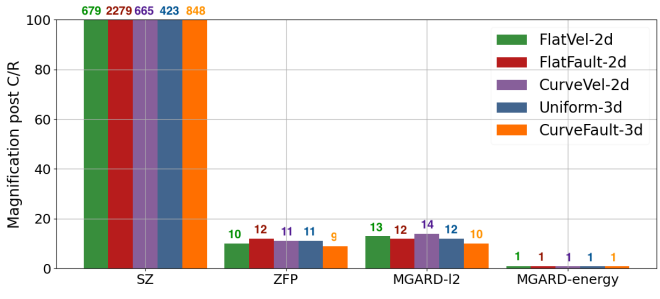
$$\epsilon_p \approx p\epsilon_0 \leq p\tau,$$

where ϵ_0 is the real error and τ is the prescribed error tolerance used for compression. This relation allows users to select an appropriate checkpoint frequency p and per-checkpoint error tolerance τ to meet a global error threshold ϵ_p.

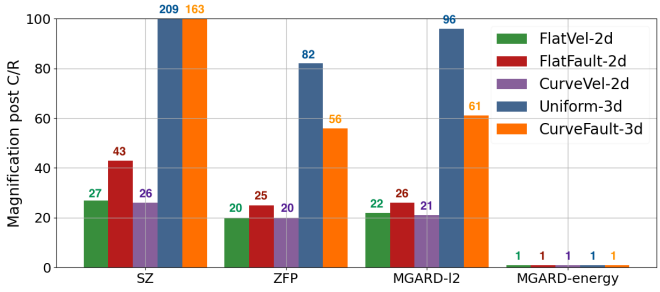


(a) RMSE (b) KE(u − ũ) (c) PE(u − ũ)

Figure 8: Illustration of error propagation in the *CurveFault-2D* case with checkpoints compressed using different compressors. KE and PE errors after $t = 4.6$ sec are omitted from the visualization because the oscillation stabilizes afterwards.



(a) RMSE magnification



(b) PE error magnification

Figure 9: Illustration of error magnification incurred by lossy checkpointing across different initial conditions. The magnification is measured over 5000 steps for 2D experiments, 2000 steps for CurveFault-3d, and 1000 steps for Uniform-3d.

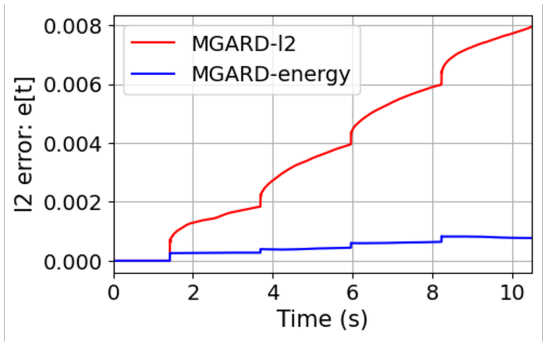


Figure 10: Illustration of RSME error magnification after multiple lossy checkpointing and restart. Each restart runs for 4000 steps (i.e., 2 sec) and a tolerance on RMSE of 1×10^{-3} is prescribed for checkpoint compression.

5.4 I/O acceleration using lossy checkpointing

Finally, we demonstrate the I/O performance benefits of MGARD-energy. We perform weak scaling tests across 1,024 Frontier nodes using data derived from two velocity maps—*CurveFault-3D* and *Uniform-3D*, loading 4 GB and 17 GB data per GPU, respectively. Across all cases, relative RMSE errors remains below 2×10^{-6}. Compression and decompression are performed using GPU-accelerated MGARD. Prior work [13] showed MGARD achieving 103 TB/s throughput under the same setting but with different compression ratios. Below, we measure:

- *Read time:* Time to read compressed checkpoint and decompress.
- *Write time:* Time to compress data and write to PFS.

Figure 11 shows the I/O results for checkpointing at various simulation ($t = 2$ sec, 2.75 sec, and 3.5 sec for *CurveFault-3D*; $t = 3$ sec, 3.5 sec and 3.8 sec for *Uniform-3D*) using 1024 nodes. To ensure fair comparison, MGARD-l2 error bounds were tuned to match final RMSE levels post-restart (i.e., after error magnification). Experiments show that the I/O of C/R using MGARD-energy is $4.2 - 6.3\times$ faster on read, and $2.9 - 8.4\times$ faster on write relative to the I/O of uncompressed data. Comparing to I/O of C/R made through MGARD-l2, MGARD-energy achieves $1.2 - 1.8\times$ reads and $1.5 - 2.2\times$ writes. Specifically, the compression ratios achieved are: (1) *Uniform-3D*: MGARD-energy=117.3×, 80.5×, 65.4× for t=3s, 3.5s, 3.8s; MGARD-l2= 64×, 41.8×, and 34.3× for t=3s, 3.5s, 3.8s. (2) *CurveFault-3D*: MGARD-energy=238×, 89.3×, and 42.2× for t=3s, 3.5s, 3.8s; MGARD-l2=144×, 54.3×, 25.8× for t=3s, 3.5s, 3.8s. Overall, MGARD-energy provides a $63 - 91\%$ higher compression ratio than MGARD-l2, translating to substantial I/O speedups during checkpointing.

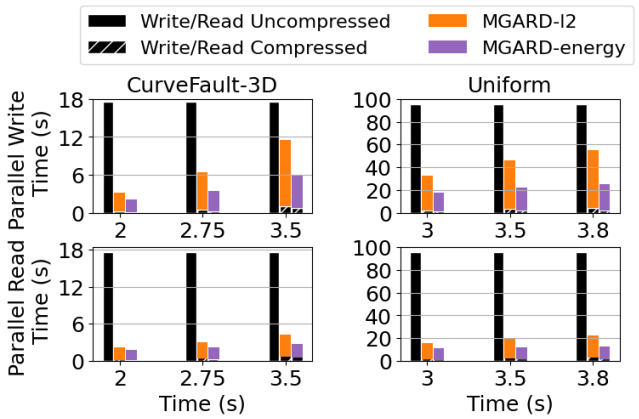


Figure 11: Parallel I/O with lossy checkpointing across 1024 nodes on Frontier at the OLCF.

6 Conclusion and Future work

This work introduces a physics-informed compression approach, addressing a fundamental challenge in checkpointing for the wave equation such that compression-induced errors do not amplify over time. Unlike prior efforts that treated lossy checkpointing as an application-agnostic data reduction task, our method explicitly incorporates the physics of the second-order wave equation by separating temporal iterates into kinetic and potential energy components, each preserved with distinct error constraints through MGARD s-norms. This physics-informed strategy prevents unbounded error growth, maintains stability across multiple restart cycles, and significantly accelerates checkpoint I/O. Compared with checkpoints compressed using standard error norms, our approach obtains minimal error amplification and improved stability, underscoring the importance of incorporating physical constraints—such as energy conservation—into lossy compression for scientific simulations. These contributions provide a practical and scalable path to enhance the reliability and resilience of large-scale PDE simulations.

Looking ahead, this work opens several directions for advancing physics-informed compression. Extending the methodology beyond the second-order wave equation to nonlinear, mixed-type, and multiphysics PDEs will be essential for broader adoption. Many of these systems may require preserving additional invariants and different compression strategies to ensure long-term reliability. Integration with advanced checkpointing frameworks also presents opportunities for deployment in production environments, where coupling physics-aware compression with hierarchical storage management, burst-buffer optimizations, and adaptive checkpoint scheduling can deliver application-aware trade-offs between speed, accuracy, and resilience. Beyond fault tolerance, physics-preserving compression holds promise for stability in situ analysis, reduced-order modeling, and machine-learning–based surrogates, where long-term consistency of reduced data is critical.

Acknowledgments

The research is supported in part by the U.S. Department of Energy (DOE) RAPIDS-2 SciDAC and Sirius2 projects under contract number DE-AC05-00OR22725, and National Science Foundation (NSF) under the grants DMS-2324364, OAC-2313122, OAC-2311756, OAC-2311757 and OAC-2144403. This research used resources of the Oak Ridge Leadership Computing Facility (OLCF), which is a DOE Office of Science User Facility.

References

[1] Mark Ainsworth, Ozan Tugluk, Ben Whitney, and Scott Klasky. 2019. Multilevel techniques for compression and reduction of scientific data—the multivariate case. *SIAM Journal on Scientific Computing* 41, 2 (2019), A1278–A1303.

[2] Mark Ainsworth, Ozan Tugluk, Ben Whitney, and Scott Klasky. 2019. Multilevel techniques for compression and reduction of scientific data-quantitative control of accuracy in derived quantities. *SIAM Journal on Scientific Computing* 41, 4 (2019), A2146–A2171.

[3] Noushin Azami, Alex Fallin, and Martin Burtscher. 2025. Efficient Lossless Compression of Scientific Floating-Point Data on CPUs and GPUs. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*. 395–409.

[4] Leonardo Bautista-Gomez, Seiji Tsuboi, Dimitri Komatitsch, Franck Cappello, Naoya Maruyama, and Satoshi Matsuoka. 2011. FTI: High performance fault tolerance interface for hybrid systems. In *Proceedings of 2011 international conference for high performance computing, networking, storage and analysis*. 1–32.

[5] Folkmar Bornemann and Harry Yserentant. 1993. A basic norm equivalence for the theory of multilevel methods. *Numer. Math.* 64 (1993), 455–476.

[6] Greg Bronevetsky, Daniel Marques, Keshav Pingali, and Paul Stodghill. 2003. Automated application-level checkpointing of MPI programs. In *Proceedings of the ninth ACM SIGPLAN symposium on Principles and practice of parallel programming*. 84–94.

[7] Oscar P Bruno. 2003. Fast, high-order, high-frequency integral methods for computational acoustics and electromagnetics. In *Topics in computational wave propagation: direct and inverse problems*. Springer, 43–82.

[8] Jon Calhoun, Franck Cappello, Luke N Olson, Marc Snir, and William D Gropp. 2019. Exploring the feasibility of lossy compression for pde simulations. *The International Journal of High Performance Computing Applications* 33, 2 (2019), 397–410.

[9] Jon Calhoun, Luke Olson, and Marc Snir. 2016. Application of Numerical Accuracy to the Selection of Lossy Compression Error Tolerances. In *The International Conference for High Performance Computing, Networking, Storage and Analysis*.

[10] Franck Cappello, Allison Baker, Ebru Bozda, Martin Burtscher, Kyle Chard, Sheng Di, Paul Christopher O Grady, Peng Jiang, Shaomeng Li, Erik Lindahl, et al. 2025. Lossy Compression of Scientific Data: Applications Constrains and Requirements. *arXiv preprint arXiv:2503.20031* (2025).

[11] Franck Cappello, Sheng Di, Sihuan Li, Xin Liang, Ali Murat Gok, Dingwen Tao, Chun Hong Yoon, Xin-Chuan Wu, Yuri Alexeev, and Frederic T Chong. 2019. Use cases of lossy compression for floating-point data in scientific data sets. *The International Journal of High Performance Computing Applications* 33, 6 (2019), 1201–1220.

[12] Aaron Charous and Pierre FJ Lermusiaux. 2021. Dynamically orthogonal differential equations for stochastic and deterministic reduced-order modeling of ocean acoustic wave propagation. In *OCEANS 2021: San Diego–Porto*. IEEE, 1–7.

[13] Jieyang Chen, Qian Gong, Yanliang Li, Xin Liang, Lipeng Wan, Qing Liu, Norbert Podhorszki, and Scott Klasky. 2025. HPDR: High-Performance Portable Scientific Data Reduction Framework. *arXiv preprint arXiv:2503.06322* (2025).

[14] Pedro CS Costa, Stephen Jordan, and Aaron Ostrander. 2019. Quantum algorithm for simulating the wave equation. *Physical Review A* 99, 1 (2019), 012323.

[15] Carlos A De Moura and Carlos S Kubrusly. 2013. The courant–friedrichs–lewy (cfl) condition. *AMC* 10, 12 (2013), 45–90.

[16] Chengyuan Deng, Shihang Feng, Hanchen Wang, Xitong Zhang, Peng Jin, Yinan Feng, Qili Zeng, Yinpeng Chen, and Youzuo Lin. 2022. OpenFWI: Large-scale multi-structural benchmark datasets for full waveform inversion. *Advances in Neural Information Processing Systems* 35 (2022), 6007–6020.

[17] Sheng Di and Franck Cappello. 2016. Fast error-bounded lossy HPC data compression with SZ. In *2016 ieee international parallel and distributed processing symposium (ipdps)*. IEEE, 730–739.

[18] Daiane I Dolci, James R Maddison, David A Ham, Guillaume Pallez, and Julien Herrmann. 2024. checkpoint_schedules: schedules for incremental checkpointing of adjoint simulations. *Journal of Open Source Software* 9 (2024).

[19] Ifeanyi P Egwutuoha, David Levy, Bran Selic, and Shiping Chen. 2013. A survey of fault tolerance mechanisms and checkpoint/restart implementations for high performance computing systems. *The Journal of Supercomputing* 65 (2013), 1302–1326.

[20] Gwynne Evans, Jonathan Blackledge, and Peter Yardley. 2012. *Numerical methods for partial differential equations*. Springer Science & Business Media.

[21] Alyson Fox, James Diffenderfer, Jeffrey Hittinger, Geoffrey Sanders, and Peter Lindstrom. 2020. Stability analysis of inline ZFP compression for floating-point data in iterative methods. *SIAM Journal on Scientific Computing* 42, 5 (2020), A2701–A2730.

[22] Jean-loup Gailly and Mark Adler. 2004. Zlib compression library. (2004).

[23] Qian Gong, Jieyang Chen, Ben Whitney, Xin Liang, Viktor Reshniak, Tania Banerjee, Jaemoon Lee, Anand Rangarajan, Lipeng Wan, Nicolas Vidal, et al. 2023. MGARD: A multigrid framework for high-performance, error-controlled data compression and refactoring. *SoftwareX* 24 (2023), 101590.

[24] Qian Gong, Xin Liang, Ben Whitney, Jong Youl Choi, Jieyang Chen, Lipeng Wan, Stéphane Ethier, Seung-Hoe Ku, R Michael Churchill, C-S Chang, et al. 2021. Maintaining trust in reduction: Preserving the accuracy of quantities of interest for lossy compression. In *Smoky Mountains Computational Sciences and Engineering Conference*. Springer, 22–39.

[25] Qian Gong, Chengzhu Zhang, Xin Liang, Viktor Reshniak, Jieyang Chen, Anand Rangarajan, Sanjay Ranka, Nicolas Vidal, Lipeng Wan, Paul Ullrich, et al. 2023. Spatiotemporally adaptive compression for scientific dataset with feature preservation–a case study on simulation data with extreme climate events analysis. In *2023 IEEE 19th International Conference on e-Science (e-Science)*. IEEE, 1–10.

[26] Sebastian Götschel and Martin Weiser. 2019. Compression challenges in large scale partial differential equation solvers. *Algorithms* 12, 9 (2019), 197.

[27] Sebastian Götschel and Martin Weiser. 2019. Lossy Compression for Large Scale PDE Problems. (2019).

[28] Graham Griffiths and William E Schiesser. 2010. *Traveling wave analysis of partial differential equations: numerical and analytical methods with MATLAB and Maple*. Academic Press.

[29] Doug Hakkarinen and Zizhong Chen. 2012. Multilevel diskless checkpointing. *IEEE Trans. Comput.* 62, 4 (2012), 772–783.

[30] Vincent Heuveline and Andrea Walther. 2006. Online checkpointing for parallel adjoint computation in PDEs: Application to goal-oriented adaptivity and flow control. In *European Conference on Parallel Processing*. Springer, 689–699.

[31] Dewan Ibtesham, Kurt B Ferreira, and Dorian Arnold. 2015. A checkpoint compression study for high-performance computing systems. *The International Journal of High Performance Computing Applications* 29, 4 (2015), 387–402.

[32] Masoud Jami. 2024. *Optimizing Checkpoint/Restart and Input/Output for Large Scale Applications*. Ph. D. Dissertation. Mathematisch-Naturwissenschaftliche Fakultät.

[33] Pu Jiao, Sheng Di, Hanqi Guo, Kai Zhao, Jiannan Tian, Dingwen Tao, Xin Liang, and Franck Cappello. 2022. Toward quantity-of-interest preserving lossy compression for scientific data. *Proceedings of the VLDB Endowment* 16, 4 (2022), 697–710.

[34] Dimitri Komatitsch, Jean-Pierre Vilotte, Rossana Vai, José M Castillo-Covarrubias, and Francisco J Sánchez-Sesma. 1999. The spectral element method for elastic wave equations—application to 2-D and 3-D seismic problems. *International Journal for numerical methods in engineering* 45, 9 (1999), 1139–1164.

[35] Navjot Kukreja, Jan Hückelheim, Mathias Louboutin, Paul Hovland, and Gerard Gorman. 2019. Combining checkpointing and data compression to accelerate adjoint-based optimization problems. In *Euro-Par 2019: Parallel Processing: 25th International Conference on Parallel and Distributed Computing, Göttingen, Germany, August 26–30, 2019, Proceedings 25*. Springer, 87–100.

[36] Navjot Kukreja, Jan Hückelheim, Mathias Louboutin, John Washbourne, Paul HJ Kelly, and Gerard J Gorman. 2020. Lossy checkpoint compression in full waveform inversion. *Geoscientific Model Development Discussions* 2020 (2020), 1–26.

[37] Hans Petter Langtangen. 2016. Finite difference methods for wave motion. *Center for Biomedical Computing, Simula Research Laboratory 2Department of Informatics, University of Oslo* (2016), 5–9.

[38] Michael Laufer and Erick Fredj. 2022. High performance parallel I/O and in-situ analysis in the WRF model with ADIOS2. *arXiv preprint arXiv:2201.08228* (2022).

[39] Sunwoo Lee, Kai-yuan Hou, Kewei Wang, Saba Sehrish, Marc Paterno, James Kowalkowski, Quincey Koziol, Robert B Ross, Ankit Agrawal, Alok Choudhary, et al. 2022. A case study on parallel HDF5 dataset concatenation for high energy physics data analysis. *Parallel Comput.* 110 (2022), 102877.

[40] Shaomeng Li, Peter Lindstrom, and John Clyne. 2023. Lossy scientific data compression with sperr. In *2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 1007–1017.

[41] Xin Liang, Qian Gong, Jieyang Chen, Ben Whitney, Lipeng Wan, Qing Liu, David Pugmire, Rick Archibald, Norbert Podhorszki, and Scott Klasky. 2021. Error-controlled, progressive, and adaptable retrieval of scientific data with multilevel decomposition. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–13.

[42] Xin Liang, Ben Whitney, Jieyang Chen, Lipeng Wan, Qing Liu, Dingwen Tao, James Kress, David Pugmire, Matthew Wolf, Norbert Podhorszki, et al. 2021. Mgard+: Optimizing multilevel methods for error-bounded scientific data reduction. *IEEE Trans. Comput.* 71, 7 (2021), 1522–1536.

[43] Xin Liang, Kai Zhao, Sheng Di, Sihuan Li, Robert Underwood, Ali M Gok, Jiannan Tian, Junjing Deng, Jon C Calhoun, Dingwen Tao, et al. 2022. Sz3: A modular framework for composing prediction-based error-bounded lossy compressors. *IEEE Transactions on Big Data* 9, 2 (2022), 485–498.

[44] Peter Lindstrom, Jeffrey Hittinger, James Diffenderfer, Alyson Fox, Daniel Osei-Kuffuor, and Jeffrey Banks. 2025. ZFP: A compressed array representation for numerical computations. *The International Journal of High Performance Computing Applications* 39, 1 (2025), 104–122.

[45] Peter G Lindstrom. 2017. *Fpzip*. Technical Report. Lawrence Livermore National Laboratory (LLNL), Livermore, CA (United States).

[46] Jinyang Liu, Pu Jiao, Kai Zhao, Xin Liang, Sheng Di, and Franck Cappello. 2024. QPET: A Versatile and Portable Quantity-of-Interest-preservation Framework for Error-Bounded Lossy Compression. *arXiv preprint arXiv:2412.02799* (2024).

[47] Ning Liu, Jason Cope, Philip Carns, Christopher Carothers, Robert Ross, Gary Grider, Adam Crume, and Carlos Maltzahn. 2012. On the role of burst buffers in leadership-class storage systems. In *2012 IEEE 28th Symposium on Mass Storage Systems and Technologies (MSST)*. IEEE, 1–11.

[48] Andreas-Stefanos I Margetis, Evangelos M Papoutsis-Kiachagias, and Kyriakos C Giannakoglou. 2023. Reducing memory requirements of unsteady adjoint by synergistically using check-pointing and compression. *International Journal for Numerical Methods in Fluids* 95, 1 (2023), 23–43.

[49] Ravish Mehra, Nikunj Raghuvanshi, Lauri Savioja, Ming C Lin, and Dinesh Manocha. 2012. An efficient GPU-based time domain solver for the acoustic wave equation. *Applied Acoustics* 73, 2 (2012), 83–94.

[50] Kathryn Mohror, Adam Moody, Greg Bronevetsky, and Bronis R De Supinski. 2013. Detailed modeling and evaluation of a scalable multilevel checkpointing system. *IEEE Transactions on Parallel and Distributed Systems* 25, 9 (2013), 2255–2263.

[51] Adam Moody, Greg Bronevetsky, Kathryn Mohror, and Bronis R De Supinski. 2010. Design, modeling, and evaluation of a scalable multi-level checkpointing system. In *SC'10: Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 1–11.

[52] Marina Morán, Javier Balladini, Dolores Rexachs, and Emilio Luque. 2019. Prediction of energy consumption by checkpoint/restart in HPC. *IEEE Access* 7 (2019), 71791–71803.

[53] ORNL. [n. d.]. *Frontier Supercomputer*. https://www.olcf.ornl.gov/frontier/

[54] Peter Oswald. 2013. *Multilevel finite element approximation: Theory and applications*. Springer-Verlag.

[55] Joaquim Peiró and Spencer Sherwin. 2005. Finite difference, finite element and finite volume methods for partial differential equations. In *Handbook of Materials Modeling: Methods*. Springer, 2415–2446.

[56] James S Plank, Kai Li, and Michael A Puening. 1998. Diskless checkpointing. *IEEE Transactions on parallel and Distributed Systems* 9, 10 (1998), 972–986.

[57] Naoto Sasaki, Kento Sato, Toshio Endo, and Satoshi Matsuoka. 2015. Exploration of lossy compression for application-level checkpoint/restart. In *2015 IEEE international parallel and distributed processing symposium*. IEEE, 914–922.

[58] Faisal Shahzad, Markus Wittmann, Thomas Zeiser, Georg Hager, and Gerhard Wellein. 2013. An evaluation of different I/O techniques for checkpoint/restart. In *2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum*. IEEE, 1708–1716.

[59] Dingwen Tao, Sheng Di, Xin Liang, Zizhong Chen, and Franck Cappello. 2018. Improving performance of iterative methods by lossy checkponting. In *Proceedings of the 27th international symposium on high-performance parallel and distributed computing*. 52–65.

[60] ANL teams. 2025. SZ code repository. https://github.com/szcompressor/SZ. Accessed: 2025-08-27.

[61] LLNL teams. 2025. ZFP code repository. https://github.com/LLNL/zfp. Accessed: 2025-08-27.

[62] ORNL teams. 2025. MGARD code repository. https://github.com/CODARcode/MGARD. Accessed: 2025-08-27.

[63] Jiannan Tian, Sheng Di, Kai Zhao, Cody Rivera, Megan Hickman Fulp, Robert Underwood, Sian Jin, Xin Liang, Jon Calhoun, Dingwen Tao, et al. 2020. Cusz: An efficient gpu-based error-bounded lossy compression framework for scientific data. In *Proceedings of the ACM International Conference on Parallel Architectures and Compilation Techniques*. 3–15.

[64] Pavlo Triantafyllides, Tasmia Reza, and Jon C Calhoun. 2019. Analyzing the impact of lossy compressor variability on checkpointing scientific simulations. In *2019 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 1–5.

[65] Jean-Luc Vay. 2001. An extended FDTD scheme for the wave equation: application to multiscale electromagnetic simulation. *J. Comput. Phys.* 167, 1 (2001), 72–98.

[66] John Paul Walters and Vipin Chaudhary. 2006. Application-level checkpointing techniques for parallel programs. In *International Conference on Distributed Computing and Internet Technology*. Springer, 221–234.

[67] Yi-Min Wang, Yennun Huang, Kiem-Phong Vo, Pe-Yu Chung, and Chandra Kintala. 1995. Checkpointing and its applications. In *Twenty-fifth International Symposium on fault-tolerant Computing. Digest of papers*. IEEE, 22–31.

Appendix: Artifact Description/Artifact Evaluation

Artifact Description (AD)

A Overview of Contributions and Artifacts

A.1 Paper's Main Contributions

Lossy compression offers a scalable solution by reducing checkpoint data size; however, conventional methods often fail to preserve physical invariants (e.g., energy), leading to instability such as oscillations or divergence in Partial Differential Equations (PDE) systems. This paper introduces a stability-preserving compression approach specifically tailored for PDE simulations by explicitly controlling kinetic and potential energy perturbations to ensure stable restarts.

- **C_1** We analyze the root causes and dynamics of error growth following restarts from lossy-compressed PDE checkpoints.
- **C_2** We develop an energy-preserving lossy compression method that ensures stable restarts even after multiple consecutive checkpoint-restart (C/R) cycles.
- **C_3** We evaluate the I/O benefits. Parallel experiments on Frontier supercomputer show our method achieves $4.2 - 6.3\times$ faster reads and $2.9 - 8.4\times$ faster writes compared to uncompressed checkpointing, while maintaining a low relative RMSE ($\sim$ 2e-6) in continued simulation.

A.2 Computational Artifacts

A_1 https://github.com/gqian-coder/waveEquation (DOI: https://doi.org/10.5281/zenodo.15873900)

Artifact ID	Contributions Supported	Related Paper Elements
A_1	C_1, C_2, C_3	Tables 1 Figure 1-2, 7-11

B Artifact Identification

B.1 Computational Artifact A_1

Relation To Contributions

Artifact A_1 includes code for wave equation simulation, the proposed stability-preserving compression framework, and evaluation metrics such as kinetic/potential energy loss and root-of-mean-square data error (RMSE). The wave equation is implemented using finite difference method and supports multiple velocity and source configurations. The MGARD-energy framework leverages a special MGARD feature (snorm) for potential energy preservation. Energy-balancing and error-control logic is fully implemented in A_1.

Expected Results

- For C_1: Reproducing experiments from Figure 1 should reveal that L^∞ or L^2-based compressors (e.g., MGARD-l2, SZ3, ZFP) lead to error growth over time regardless of error tolerance. Figure 2 demonstrates the role of imbalanced KE and PE in driving such divergence.
- For C_2: Re-running experiments from Figures 7–9 will show that MGARD-energy controls post-restart errors effectively–unlike other methods, its RMSE, KE, and PE errors remain stable across time steps. Figure 10 shows stability across multiple C/R cycles.
- For C_3: As shown in Figure 11, MGARD-energy achieves the fastest I/O throughput by achieving $63 - 91\%$ higher compression ratios than MGARD-l2 and uncompressed data.

Expected Reproduction Time (in Minutes)

Compression/decompression and energy evaluations take seconds per dataset. Wave simulations vary from minutes to several hours depending on grid size, time steps, and number of nodes/GPUs used.

Artifact Setup (incl. Inputs)

Hardware. The proposed algorithms and softwares can be implemented on general CPUs, NVIDIA GPUs and AMD GPUs. All experiments in this work were conducted on Frontier, a leadership class supercomputer at Oak Ridge Leadership Computing Facility (OLCF).

Software.

- The wave equations were simulated using the artifact A_1
- The I/O was implemented through ADIOS v2.10 https://github.com/ornladios/ADIOS2
- Three lossy compressors were used in evaluation:
  (1) MGARD v1.5.2 https://github.com/CODARcode/MGARD/tree/master
  (2) ZFP v1.0.1 https://github.com/LLNL/zfp
  (3) SZ3 v3.2.1 https://github.com/szcompressor/SZ3

Datasets / Inputs. Velocity maps are initialized using Vel and Fault families from the OPENFWI dataset, a multi-structural benchmark suite for full waveform inversion: https://openfwi-lanl.github.io/docs/data.html

Installation and Deployment.

- A_1 can be built and installed using GCC 11+ and the following script with slight modification for the testing system: ./build_script.sh
- The compression libraries (e.g., MGARD, ZFP, SZ3) used in evaluation require to be built and installed separately and linked with A_1. The I/O throughput evaluation requires CUDA 12+ (NVIDIA GPUs) or ROCm 6+ (AMD GPUs).

Artifact Analysis (incl. Outputs)

For all contributions—$C_1 - C_2$, results are generated by the *waveEnergy* executable. The I/O throughput measures in C_3 include compression/decompression and write/read duration.

Artifact Evaluation (AE)

C.1 Computational Artifact A_1

Artifact Setup (incl. Inputs)

Installation and compilation:

(1) git clone https://github.com/gqian-coder/waveEquation.git
(2) Donwload and install all dependency libraries and velocity templates following the instructions in artifacts setup of artifact A_1
(3) For non-uniform velocities, reshape masks from OPENFWI to the dimensions listed in Table 1. The suggested procedure is through the python library *cv2*:
cv2.resize(image_array, (new_width, new_height), interpolation=cv2.INTER_LINEAR)
Scaled values to 0–275 (2D) and 0–68 (3D).
(4) Update build_script.sh and CMakeLists.txt with software installation paths.
(5) Run: ./build_script.sh

Artifact Execution

The experiments need to be done with following the task order:

- Install artifact A_1
- Use the four scripts: *mgard_energy_eval.sh*, *mgard_l2_eval.sh*, *sz3_eval.sh*, *zfp_eval.sh* for C_1 and C_2. Each script executes the followings for MGARD-energy, MGARD-l2, sz3, and zfp-based lossy checkpoint restart:
  (1) Run original wave simulation to time N, saving the wave field data every time step—calling it u—as the reference.
  (2) Compress the checkpoint data at time M.
  (3) Restart from compressed checkpoint data and run to time N—calling it $\tilde{u}$.
  (4) Compute error amplification (i.e., $(u^{t_N} - \tilde{u}^{t_N})/(u^{t_M} - \tilde{u}^{t_M})$) in terms of RMSE, PE, and KE.
- Repeat steps 2–3, with each new restart loaded from the previously compressed checkpoint data for multi-C/R cycle evaluation.
- For C_3, tune error tolerances of MGARD-energy and MGARD-l2 to match final RMSE after N steps. Suggested tolerance:
  - CurveFault-3D: τ_{KE} = 8e−5 and τ_{PE} = 0.325 for MGARD-energy, and τ_{l2} = 8.5e−6 for MGARD-l2. The final error was approximately 8e−5 for both MGARD-energy and MGARD-l2.
  - Uniform-3D: τ_{KE} = 5.9e−6 and τ_{PE} = 0.058 for MGARD-energy, and τ_{l2} = 6.3e−7 for MGARD-l2. The final error was approximately 9e−6 for both MGARD-energy and MGARD-l2.

Artifact Analysis (incl. Outputs)

- In each of the xx_eval.sh scripts, there are two cases – 2D and 3D. Running *2D case* scripts in each script reproduces Figure 8. Running *3D case* reproduces the *CurveFault-3D* case in Figure 9. MGARD-l2, sz3 and zfp cases show growing error; MGARD-energy shows stable error after lossy restart.
- Figure 11 is reproduced by scaling experiments to multiple nodes (e.g., 1024 Frontier nodes used). Since it was a weak scaling test, smaller node counts still yield similar trends.