Geophysical Journal International

Geophys. J. Int. (2026) **244,** 1–12
Advance Access publication 2025 December 08
Research Paper

https://doi.org/10.1093/gji/ggaf496

Accelerating three-dimensional seismic wave simulations on ARM using a hybrid half-precision and scalable vector extension approach

Wenqiang Wang [1], Juepeng Zheng,[1,2] Bihe Ren [1], Zekun Yin,[3] Wubing Wan,[4] Yinuo Wang,[4] Lin Gan,[5] Zhenguo Zhang [6], Wei Zhang [6], Haohuan Fu[7] and Xiaofei Chen[6]

[1] *High Performance Computing Department, National Supercomputing Center in Shenzhen, Shenzhen 518055, China. E-mail: zhengjp8@mail.sysu.edu.cn*
[2] *School of Artificial Intelligence, Sun Yat-sen University, Zhuhai 519082, China*
[3] *School of Software, Shandong University, Jinan 250101, China*
[4] *Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*
[5] *Yau Mathematical Sciences Center, Tsinghua University, Beijing 100084, China*
[6] *Department of Earth and Space Sciences, Southern University of Science and Technology, Shenzhen 518055, China*
[7] *Tsinghua Shenzhen International Graduate School, Shenzhen 518055, China*

Accepted 2025 December 3. Received 2025 November 14; in original form 2025 September 30

SUMMARY

Seismic simulation is fundamental for understanding earthquake physics and mitigating seismic hazards, but accurate seismic modelling requires fine computational grids, imposing severe memory and computational challenges. Traditional modelling solvers, relying on single-precision floating-point 32-bit (FP32) and scalar register-based computation, suffer from excessive memory consumption, low-memory access efficiency and limited computational efficiency. Compared with FP32, half-precision floating-point 16-bit (FP16) reduces memory consumption by 50 per cent and improves memory access efficiency; relative to scalar registers, ARM's Scalable Vector Extension (SVE) registers provide vectorized single-instruction multiple-data capabilities, significantly accelerating computation. However, leveraging the advantages of FP16 and SVE involves challenges such as FP16 overflow/underflow, SVE stencil adaptation, and SVE data misalignment from FP16 storage with FP32 arithmetic. Therefore, this study proposes three approaches on the ARM (Advanced RISC Machine) architecture: FP16-based, SVE-accelerated and FP16-SVE hybrid; each is designed to tackle the respective challenges while exploiting FP16 memory efficiency and SVE computational acceleration. Correspondingly, the three solvers are implemented, validated and benchmarked on both hypothetical models and real-world earthquake scenarios. The results of these solvers show near-perfect agreement with the reference solver, confirming their accuracy across diverse seismic scenarios. Moreover, the FP16-SVE hybrid solver halved memory usage and achieved up to 3× computational speedup, delivering more than 2.3× acceleration in the real-world earthquake simulation. The gains in high efficiency of memory and computation highlight the capability of the FP16-SVE hybrid solver to support large-scale, real-time seismic simulations and efficient earthquake hazard assessment on ARM platforms.

Key words: Numerical modelling; Computational seismology; Wave propagation.

1 INTRODUCTION

Simulation is fundamental for understanding earthquake physics, assessing seismic hazards and informing disaster mitigation strategies. Accurate modelling of seismic wave propagation through complex geological environments requires fine computational grids, which, in large-scale 3-D simulations, can lead to high-memory consumption, low-memory access efficiency and limited computational efficiency (e.g. Fu *et al.* 2017; Chen *et al.* 2018; Wan *et al.* 2023).

These challenges motivate the use of efficient solvers like the finite-difference method (FDM). The FDM, particularly in its staggered-grid formulation (Virieux 1986; Graves 1996), has long served as a primary numerical approach for earthquake simulations.

However, traditional FDM face challenges in accurately representing complex geometries, especially rugged surface topography. The Curvilinear Grid Finite-Difference Method (CGFDM) extends the traditional FDM by introducing curvilinear coordinates to better conform to complex geological and topographic structures (Zhang & Chen 2006; Zhang *et al.* 2012). Owing to its capability to accurately model realistic Earth structures, CGFDM has been widely applied in large-scale seismic wave simulations (e.g. Chen *et al.* 2018; Wan *et al.* 2023). In this study, CGFDM is adopted as the core solver. Nevertheless, storing geometric metrics and performing multistage Runge–Kutta time integration (Wan *et al.* 2023; Wang *et al.* 2023) still leads to high-memory consumption, low-memory access efficiency and limited computational efficiency. To address these issues, CGFDM requires memory and computation optimization, which is one task of this work.

The above solvers are implemented using FP32 (single-precision floating-point 32-bit) arithmetic (e.g. Zhang *et al.* 2012; Chen *et al.* 2018; Wang *et al.* 2022; Wan *et al.* 2023), where each data element requires 4 bytes of memory storage. Compared with FP16 (half-precision floating-point 16-bit), FP32 has twice the memory consumption and only 50 per cent the memory access efficiency, which severely limits the simulation scale, increases computational cost and prolongs runtime in large-scale seismic modelling.

In contrast, FP16 stores each data element in only 2 bytes, halving memory footprint and doubling memory access efficiency of FP32, which has driven its adoption in fields such as deep learning and scientific computing (e.g. Clark *et al.* 2010; Courbariaux *et al.* 2014; Micikevicius *et al.* 2018). However, the 2-byte width of FP16 limits its representable range to approximately $6.1 \times 10^{-5} \sim 6.6 \times 10^{4}$ (IEEE 2008), which restricts its applicability in seismic simulations, as the physical quantities in the governing elastic wave equations often exceed this range. Recently, to address this limitation, Wang *et al.* (2023) introduced three dimensionless scaling constants (C_{v}, C_{s} and C_{p}) and derived an FP16-based elastic wave equation that significantly reduced memory usage while preserving FP32-level accuracy in realistic earthquake scenarios. Building on this foundation, subsequent studies optimized FP16 implementations on GPUs (Graphics Processing Unit, Wan *et al.* 2024) and NPUs (Neural Processing Unit, Wang *et al.* 2025), achieving notable speedups. However, these implementations remain confined to GPU and NPU platforms. FP16-based seismic solvers are still unexplored on general-purpose ARM CPUs, which are the dominant architecture in today's supercomputing and computing platforms. In this study, we extend the FP16-based elastic wave equation to the ARM architecture by fully exploiting its features, enabling low-memory consumption and high-memory access efficiency for 3-D seismic simulations.

Even with FP16 improving memory efficiency, traditional seismic solvers, which rely on scalar register-based computation (e.g. Zhang *et al.* 2012; Wang *et al.* 2022, 2023), still suffer from low-computational efficiency on modern ARM processors. This is because scalar registers can process only a single data element per instruction, preventing full exploitation of instruction-level parallelism and thus limiting computational throughput (Hennessy & Patterson 2011). In contrast, ARM's Scalable Vector Extension (SVE) allows each vector register to hold multiple data elements, and process them simultaneously by one single instruction, facilitating higher computational efficiency (Arm 2025). SVE is an ARM single-instruction multiple-data (SIMD) technology designed for high-performance computing, featuring scalable vector lengths from 128 to 2048 bits. Unlike traditional fixed-width SIMD, SVE allows the same code to run efficiently across different ARM platforms without modification, ensuring higher portability and flexibility through its scalable design. Advanced features such as predication, masked operations and gather/scatter instructions further enhance parallel efficiency, making SVE particularly suitable for large-scale scientific and seismic simulations. Therefore, once applied to seismic simulations, SVE can significantly accelerate the computation of seismic solvers. However, a major challenge remains: as each solver (e.g. CGFDM) involves complex diverse computational patterns, and fully exploiting SVE's potential requires careful adaptation of these patterns. To address this, in this work, finite-difference stencils of the CGFDM solver are restructured to fully adapt to SVE's vectorization features, thereby enabling accelerated 3-D seismic simulations on ARM-based platforms.

As introduced above, FP16 reduces memory consumption and improves memory access efficiency in seismic simulations, while SVE enhances computational efficiency. Therefore, combining FP16 with SVE can fully leverage the advantages of both techniques, significantly accelerating seismic simulations. However, the FP16-based seismic solver adopts a mixed-precision strategy, in which data are stored in FP16 format in memory and converted to FP32 during arithmetic computations, to ensure memory efficiency and computational accuracy (Wang *et al.* 2023). This design introduces a challenge: because an SVE register holds twice as many FP16 elements as FP32 ones, converting FP16 data to FP32 within SVE register results in misaligned layouts and significantly complicates vectorized operations. To overcome this limitation, this work will design an FP16–SVE hybrid algorithm to adapt the 'FP16 storage + FP32 arithmetic' scheme to the SVE acceleration framework.

The remainder of this paper is organized as follows: Section 2 introduces the methodology, including FP16-based methods, SVE acceleration, and the FP16-SVE hybrid algorithm. Section 3 verifies the accuracy of the optimized solvers against benchmark models. Section 4 evaluates the memory consumption and computational efficiency of the optimized solvers. Section 5 presents a real-world earthquake simulation, conforming that the FP16-SVE solver satisfies accuracy and computational efficiency for real-world seismic events. Finally, Section 6 concludes this study and highlights FP16-SVE's potential.

2 METHODOLOGY

This section introduces the FP16-based elastic wave equation, design and implementation of SVE algorithm for CGFDM, and the FP16-SVE hybrid algorithm.

2.1 FP16-based elastic wave equation

We begin with the velocity-stress form of the 3-D elastic wave equations:

$$\begin{cases} v_{i,t} = \sigma_{ij,j}/\rho, \\ \sigma_{ij,t} = \lambda\, v_{k,k}\, \delta_{ij} + \mu\,(v_{i,j} + v_{j,i}) + s_{ij} \end{cases} \tag{1}$$

Here, v_i and σ_{ij} denote particle velocity and stress, the comma ',' indicates spatial derivatives, $(i, j, k) \in \{x, y, z\}$, δ_{ij} is the Kronecker delta, and λ, μ are the Lamé parameters, defined as

$$\lambda = \rho(\alpha^2 - 2\beta^2), \qquad \mu = \rho\beta^2, \tag{2}$$

where ρ is density, and α, β being the P- and S-wave speeds.

As shown in Table 1, FP16 provides $\approx$3.3 significant decimal digits and a dynamic range of $[6.1 \times 10^{-5} \sim 6.6 \times 10^{4}]$, which cannot safely represent the raw magnitudes of v_i, σ_{ij} and ρ, α, β in

Table 1. IEEE FP16 and FP32 formats and their precision (IEEE 2008)

Format	FP32	FP16
Bit width	32	16
Storage range	$2^{-126} \sim 2^{127}$ $(1.2 \times 10^{-38} \sim 3.4 \times 10^{38})$	$2^{-14} \sim (2^{11} - 2^{0}) \times 2^{5}$ $(6.1 \times 10^{-5} \sim 6.6 \times 10^{4})$
Decimal digits of precision	$\log_{10} 2^{24} \approx 7.2$	$\log_{10} 2^{11} \approx 3.3$

eq. (1). Direct FP16 storage may underflow and/or overflow the wave field v_i and σ_{ij}. To address this, we adopt the scaling framework of (Wang *et al.* 2023), introducing three dimensionless constants C_v, C_s, C_p to define scaled variables

$$\begin{cases} V_i = C_\mathrm{v}\, v_i, \\ \Sigma_{ij} = C_\mathrm{s}\, \sigma_{ij}, \\ P = \dfrac{C_\mathrm{v}}{C_\mathrm{s}\, C_\mathrm{p}\, \rho}, \\ \Lambda = \dfrac{\rho\, C_\mathrm{s}}{C_\mathrm{v}}\,(\alpha^2 - 2\beta^2), \\ M = \dfrac{\rho\, C_\mathrm{s}}{C_\mathrm{v}}\,\beta^2 \end{cases} \tag{3}$$

and $S_{ij} = C_s\, \mathrm{s}_{ij}$ for the source term. Substituting eq. (3) into eq. (1) yields the FP16-based Elastic Wave Equation

$$\begin{cases} V_{i,t} = C_\mathrm{p}\, P\, \Sigma_{ij,j}, \\ \Sigma_{ij,t} = \Lambda\, V_{k,k}\, \delta_{ij} + M\,(V_{i,j} + V_{j,i}) + S_{ij}. \end{cases} \tag{4}$$

By properly selecting $(C_\mathrm{v}, C_\mathrm{s}, C_\mathrm{p})$, Wang *et al.* (2023) provided detailed guidance on how to determine their values. Consequently, the magnitudes of V_i, Σ_{ij}, P, Λ, M remain within the FP16 representable range, while ensuring V_i and Σ_{ij} within similar magnitudes to enhance numerical stability and conditioning. In particular, when $C_\mathrm{v} = C_\mathrm{s} = C_\mathrm{p} = 1$, the FP16-based equation degenerates to the original elastic wave equation.

Following the framework of Wang *et al.* (2023), a mixed-precision strategy is employed: FP16 is used to store wavefields, time-invariant parameters and coefficients, while FP32 is employed for computational operations, such as '$+, -, \times, \div$'. The results of these operations are subsequently converted back to FP16 for storage. This 'FP16 storage + FP32 arithmetic' mixed-precision design combines the memory efficiency of FP16 with FP32-level accuracy.

During data output, wavefield snapshots and synthetic seismograms are recovered in physical units through the inverse scaling of eq. (3), that is, $v_i = V_i / C_\mathrm{v}$ and $\sigma_{ij} = \Sigma_{ij} / C_\mathrm{s}$.

2.2 Design and implementation of SVE algorithm for CGFDM

To accurately model seismic wave propagation in complex geological structures, the CGFDM (Zhang *et al.* 2012) is employed. This method maps irregular physical domains (x, y, z) to a uniform computational domain (ξ, η, ζ) through a curvilinear coordinate mapping:

$$x = x(\xi, \eta, \zeta), \quad y = y(\xi, \eta, \zeta), \quad z = z(\xi, \eta, \zeta). \tag{5}$$

Within this computational domain, the velocity and stress components of the wavefield are organized into a state vector:

$$\begin{aligned} \boldsymbol{W} &= (v_x, v_y, v_z, \sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{xz}, \sigma_{yz})^T \\ \boldsymbol{W} &= (V_x, V_y, V_z, \Sigma_{xx}, \Sigma_{yy}, \Sigma_{zz}, \Sigma_{xy}, \Sigma_{xz}, \Sigma_{yz})^T, \end{aligned} \tag{6}$$

corresponding to eqs (1) and (4), respectively. This formulation enables a unified treatment of all wavefield variables during numerical computation. Spatial derivatives are consistently expressed via the chain rule. For example, the derivative in the x-direction is expanded as

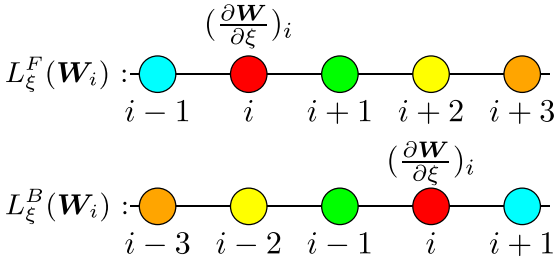

Figure 1. The stencil of the forward and backward difference operators in the MacCormack–Hixon scheme. The grid point i marks the spatial derivative position.

$$\frac{\partial \boldsymbol{W}}{\partial x} = \frac{\partial \boldsymbol{W}}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial \boldsymbol{W}}{\partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial \boldsymbol{W}}{\partial \zeta}\frac{\partial \zeta}{\partial x}, \tag{7}$$

and similar expressions hold for the y- and z-directions. The coefficients $\partial\xi/\partial x$, $\partial\eta/\partial x$, and $\partial\zeta/\partial x$—that is, the entries of the Jacobian inverse matrix $\boldsymbol{J}^{-1} = \partial(\xi, \eta, \zeta)/\partial(x, y, z)$—are computed from the grid coordinates (Zhang *et al.* 2012).

After establishing the coordinate mapping, eq. (1) is discretized in the computational domain. Time integration is performed using the classical 4th-order Runge–Kutta method, ensuring accuracy and long-term stability (Zhang *et al.* 2012). Spatial derivatives in (ξ, η, ζ) are approximated using the high-order MacCormack–Hixon scheme (Hixon 1997, 1998; Zhang *et al.* 2012), which effectively reduces numerical dispersion. Taking the ξ-direction as an example, the spatial derivative $\frac{\partial \boldsymbol{W}}{\partial \xi}$ at point node i, is approximated by forward and backward difference operators, defined respectively as

$$\begin{aligned} L_\xi^F(\boldsymbol{W}_i) &= \frac{1}{\Delta\xi}\sum_{n=-1}^{3} a_n \boldsymbol{W}_{i+n}, \\ L_\xi^B(\boldsymbol{W}_i) &= \frac{1}{\Delta\xi}\sum_{n=-1}^{3} -a_n \boldsymbol{W}_{i-n}, \end{aligned} \tag{8}$$

where the scheme coefficients a_n are provided in Zhang's work (Zhang & Chen 2006; Zhang *et al.* 2012), as illustrated more intuitively by the stencil structures in Fig. 1. In Fig. 1, the forward stencil spans nodes $i-1$ to $i+3$ and the backward stencil spans $i-3$ to $i+1$. In both cases, node i denotes the grid point where the derivative is evaluated.

In conventional implementations, the stencil-based operators in Fig. 1 are computed sequentially with scalar registers, such that each instruction updates at only one grid point. Although this scalar approach is conceptually straightforward, this scalar approach becomes highly inefficient, as computational operation must be repeated for millions or even billions of grid points in a seismic simulation.

To overcome this limitation, this work leverages the SVE of the ARM architecture to accelerate stencil-based computations. SVE

Table 2. Grid points for MacCormack–Hixon stencil to be loaded into each SVE register.

Register	ξ-direction	η-direction	ζ-direction
		Forward difference (F)	
R_{-1}	$(i-1 \sim i+14, j, k)$	$(i \sim i+15, j-1, k)$	$(i \sim i+15, j, k-1)$
R_0	$(i \sim i+15, j, k)$	$(i \sim i+15, j, k)$	$(i \sim i+15, j, k)$
R_1	$(i+1 \sim i+16, j, k)$	$(i \sim i+15, j+1, k)$	$(i \sim i+15, j, k+1)$
R_2	$(i+2 \sim i+17, j, k)$	$(i \sim i+15, j+2, k)$	$(i \sim i+15, j, k+2)$
R_3	$(i+3 \sim i+18, j, k)$	$(i \sim i+15, j+3, k)$	$(i \sim i+15, j, k+3)$
		Backward difference (B)	
R_{-3}	$(i-3 \sim i+12, j, k)$	$(i \sim i+15, j-3, k)$	$(i \sim i+15, j, k-3)$
R_{-2}	$(i-2 \sim i+13, j, k)$	$(i \sim i+15, j-2, k)$	$(i \sim i+15, j, k-2)$
R_{-1}	$(i-1 \sim i+14, j, k)$	$(i \sim i+15, j-1, k)$	$(i \sim i+15, j, k-1)$
R_0	$(i \sim i+15, j, k)$	$(i \sim i+15, j, k)$	$(i \sim i+15, j, k)$
R_1	$(i+1 \sim i+16, j, k)$	$(i \sim i+15, j+1, k)$	$(i \sim i+15, j, k+1)$

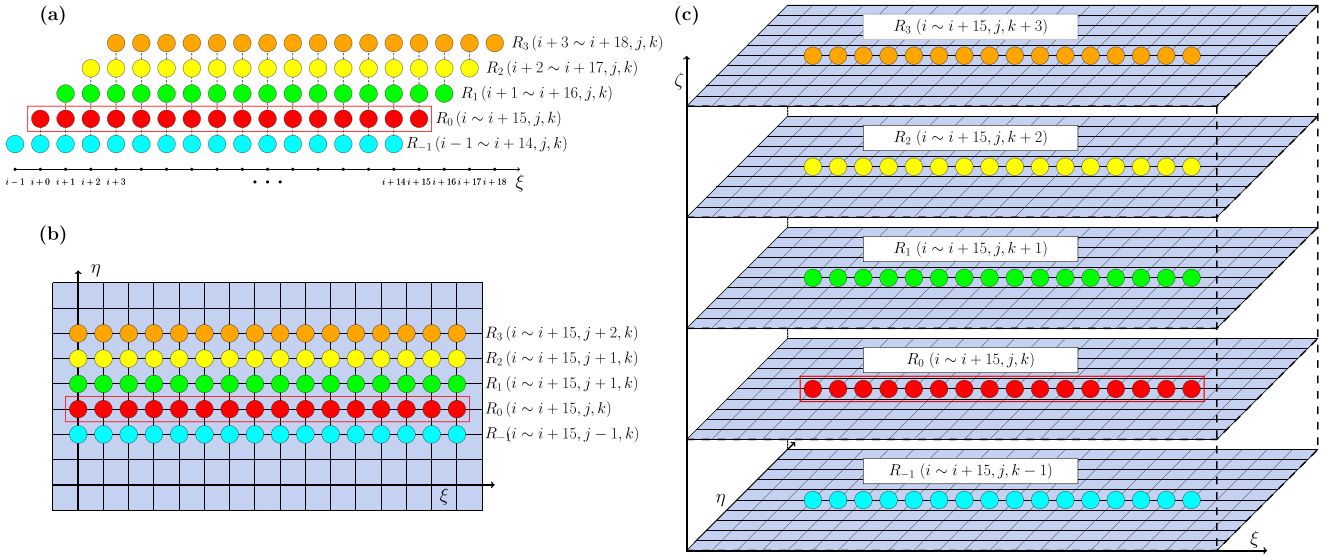

Figure 2. Grid points for the forward difference stencil to be loaded into the SVE registers along the (a) ξ, (b) η and (c) ζ directions for computing spatial derivatives. The grid points $(i \sim i+15, j, k)$ denote where derivatives are evaluated. The the stencil points are consistent with those used in Fig. 1.

registers are configurable in width—256, 512, or 1024 bits—and can hold multiple floating-point elements depending on their width (e.g. a 512-bit register can store 16 FP32) (Arm 2025). In this study, 512-bit registers are used as an illustrative example, but the proposed method is fully general and applicable to other vector lengths. Each core is equipped with SVE vector registers, enabling the computation of spatial derivatives at 16 consecutive grid points $(i \sim i+15)$ within a single vector instruction, rather than updating one grid point (i) at a time as in scalar operations.

In the SVE-based implementation of the MacCormack–Hixon scheme, the 5-point stencil is mapped onto five vector registers, allowing derivatives at 16 consecutive grid points $(i \sim i+15)$ to be computed simultaneously. For forward differences, the registers are denoted as R_{-1}, R_0, R_1, R_2, R_3, while for backward differences, they are R_{-3}, R_{-2}, R_{-1}, R_0, R_1. Each register holds 16 FP32 values, allowing the derivative at all 16 grid points to be evaluated simultaneously in a single vector instruction.

Specially, we consider computing the derivative at 16 consecutive grid points $(i \sim i+15, j, k)$, that is, $\left(\frac{\partial W}{\partial \xi}\right)_{(i\sim i+15, j, k)}$, where W denotes any component of the wavefield $\boldsymbol{W}$, and the indices (i, j, k) represent grid points along the ξ, η and ζ directions, respectively. In memory, the wavefield data is stored in a ξ-major order, followed by η and ζ. To ensure contiguous memory access for vectorized computation, the stencil data is loaded along the ξ direction into the SVE registers.

Following the stencil configuration in eq. (8), the forward and backward difference operators in the ξ-direction are defined as:

$$\begin{aligned}
\left(\frac{\partial W}{\partial \xi}\right)^{F}_{(i\sim i+15, j, k)} &= \frac{1}{\Delta \xi}\left(a_{-1}R_{-1} + a_0 R_0\right. \\
&\left. + a_1 R_1 + a_2 R_2 + a_3 R_3\right) \\
\left(\frac{\partial W}{\partial \xi}\right)^{B}_{(i\sim i+15, j, k)} &= \frac{1}{\Delta \xi}\left(-a_3 R_{-3} - a_2 R_{-2} - a_1 R_{-1}\right. \\
&\left. - a_0 R_0 - a_{-1} R_1\right)
\end{aligned} \tag{9}$$

The derivatives in the η- and ζ-directions are computed analogously using the corresponding stencil mappings along those directions.

Table 2 lists the positions of all grid points for the MacCormack–Hixon forward and backward difference stencil that need to be loaded for each SVE register. Taking the forward difference as an example, we detail how the corresponding stencil is loaded into SVE registers (shown in Fig. 2). In the ξ-direction (Fig. 2a), the registers R_{-1}, R_0, R_1, R_2, R_3 load the grid point values along the same row of the k-th layer and the j-th line, covering nodes from $i-3$ to $i+18$. Each register holds 16 consecutive values, and some of the values are shared among adjacent registers due to the

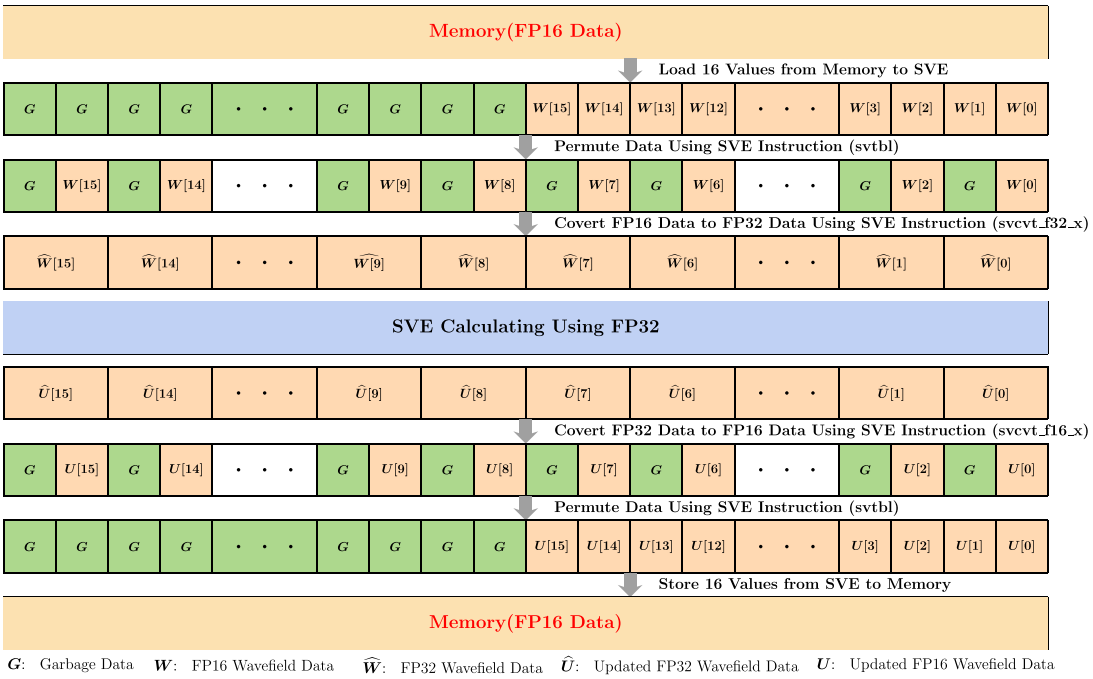

Figure 3. Illustration of the FP16–SVE hybrid algorithm. Data are loaded in FP16 format, rearranged and converted to FP32 for computation, then converted back to FP16 and reordered before being written to memory.

stencil overlap. In the η-direction (Fig. 2b), the registers still load values along the ξ-direction for nodes $i \sim i+15$, but now they correspond to multiple rows from $j-1$ to $j+3$ in the k-th layer. In the ζ-direction (Fig. 2c), the registers load values along the ξ-direction for nodes $i \sim i+15$ in the j-th row, but across adjacent layers from $k-1$ to $k+3$.

The backward difference is implemented analogously to the forward difference. The correspondence between grid points and the registers R_{-3}, R_{-2}, R_{-1}, R_0, R_1 is summarized in Table 2.

2.3 FP16-SVE hybrid algorithm

As discussed in Section 2.1, this study adopts a mixed-precision strategy of 'FP16 storage + FP32 arithmetic'. According to the theoretical analysis in Section 2.1, arithmetic operations must be performed in FP32, even though the data are stored in FP16. This design introduces a technical challenge for SVE acceleration: FP16-stored wavefields do not match FP32 vector lanes, making direct vectorized computation impossible without reshaping or padding. To address this limitation, we design an FP16–SVE hybrid algorithm that adapts the 'FP16 storage + FP32 arithmetic' scheme to the SVE acceleration framework. The procedure is illustrated in Fig. 3 and described as follows.

Referring to Section 2.2, using 512-bit registers as an example, each SVE register can hold 16 FP32 or 32 FP16 elements. Under 'FP16 storage + FP32 arithmetic' strategy, only 16 FP16 values from memory are initially loaded into one SVE register, which are then converted to FP32 for computation. In this process, the lower half of the register (right-hand side of Fig. 3) is filled with FP16 wavefield data W, while the upper half (left-hand side) remains unused and contains invalid garbage values G.

Direct conversion of these FP16 values to FP32 is not supported for arbitrary register positions. Instead, SVE provides conversion only for FP16 elements located at even/odd indices. To overcome this limitation, the FP16 data are first rearranged using the `svtbl` instruction, which permutes the register contents such that all valid FP16 wavefield elements are placed at even positions. Subsequently, the `svcvt_f32_x` instruction converts these FP16 elements into FP32 format, fully slots of the SVE register with FP32 wavefield values $\widehat{W}$.

Once converted, the FP32 wavefield data $\widehat{W}$ are fed into the SVE arithmetic pipeline, where stencil operations (e.g. spatial derivatives) are executed in parallel over 16 grid points per vector instruction, as detailed in Section 2.2. After the computation, the updated FP32 results $\widehat{U}$ are converted back to FP16 using the `svcvt_f16_x` instruction. As in the loading phase, this produces FP16 values only at even register positions. Therefore, another rearrangement operation via `svtbl` is applied to restore the correct ordering of the FP16 data within the register. Finally, the reordered FP16 values U are stored back to memory, completing one full cycle of load—compute—store in the FP16–SVE hybrid algorithm.

2.4 Boundary conditions

For the free-surface boundary, the traction image method (Zhang & Chen 2006; Zhang *et al.* 2012) is employed to enforce the vanishing traction condition (Aki & Richards 2002) in this study. To suppress artificial reflections at the computational boundaries, the complex-frequency-shifted perfectly matched layer using auxiliary differential equation (ADE-CFS-PML) technique (Zhang & Shen 2010) is adopted as an absorbing boundary condition. Both approaches are directly compatible with the present mixed-precision and SVE-accelerated framework. Wang *et al.* (2023) demonstrated that FP16 optimization of ADE-CFS-PML requires introducing dozens of additional physical variables and associated dimensionless constants, rendering parameter tuning highly complex and impractical. Hence, PML is not suitable for FP16 optimization. Following their strategy, it is used in this

Table 3. The 3-D half-space multilayered media model used for the verification test, following Subsection 3.1.

Thickness (km)	Density ρ(kg m^{-3})	S-wave velocity β (m s^{-1})	P-wave velocity α (m s^{-1})
5	2250	2500	5000
5	2500	3000	6000
5	2750	3500	7000
5.1	3000	4000	8000

study only as an absorbing boundary without further optimization.

3 NUMERICAL VERIFICATIONS

Following the methodology presented in Section 2, three optimized solvers were developed based on the ARM architecture: the FP16 solver (hereafter FP16-CGFDM), the FP32-based SVE-accelerated solver (hereafter FP32-SVE) and the hybrid FP16-SVE solver (hereafter FP16-SVE). For verification, the FP32-based solver CGFDM3D-EQR (Wang *et al.* 2022) (hereafter FP32-CGFDM) is employed as the reference. The FP16-CGFDM, FP32-SVE and FP16-SVE solvers are evaluated for their accuracy in capturing sharp contrasts in multilayered media and their capability to handle complex geometries, using two benchmark cases: seismic wave propagation in a half-space multilayered media model and within Gaussian-shaped topography Model.

For the two numerical experiments in this section, we follow the approach of Wang *et al.* (2023), setting the three dimensionless constants $C_s = 1 \times 10^{-6}$, $C_v = 5 \times 10^3$ and $C_p = 1 \times 10^6$. An explosive point source is adopted, with the derivative of the Ricker wavelet as the source time function (Wang *et al.* 2023). The parameters are set as the centre frequency $f_c = 1.5$ Hz and the time delay $t_0 = 1.2/f_c s$, and the seismic moment $M_0 = 10^{16}$ N · m.

3.1 Half-space multilayered media model

To verify the accuracy in capturing sharp contrasts in multilayered media, we first consider seismic wave propagation in a classical 3-D half-space multilayered model, with parameters listed in Table 3. The computational domain spans 20.1 km × 20.1 km × 20.1 km and is discretized into 201 × 201 × 201 grid points, corresponding to a spatial resolution of 100 m. The simulation runs for 16 s with a time step of 0.008 s, resulting in 2000 iterations. The parallel decomposition adopts a 1 × 8 × 8 MPI(Message Passing Interface) layout on 64 physical cores of the ARM platform. The hypocentre is located at (0, 0, −2) km, as marked by the red star in Fig. 4. Five virtual seismic stations are deployed on the free surface along the x-axis: A (0,0,0), B (2, 0, 0) km, C (4, 0, 0) km, D (6, 0, 0) km and E (8, 0, 0) km (green inverted triangles in Fig. 4).

Fig. 4 presents the vertical velocity snapshots at $t = 6.4$ s. The wavefield snapshots from FP16-SVE and FP32-CGFDM are indistinguishable, demonstrating near-perfect consistency in both amplitude, phase and wavefront. The relative error (Fig. 4c) remains below 0.64 per cent across the entire wavefield. This level of agreement demonstrates that FP16-SVE maintains excellent accuracy in reproducing the reference solution, even in the presence of very sharp multilayer interfaces.

Fig. 5 shows synthetic seismograms at stations A–E generated by the optimized solvers (FP16-CGFDM, FP32-SVE and FP16-SVE). The produced waveforms are indistinguishable from FP32-CGFDM in arrival times, phases and amplitudes. The zoomed view at station E reveals that the difference between FP16-SVE and FP32-CGFDM only becomes apparent after a 100-fold amplification. These results provide compelling evidence that our optimized solvers achieve the same level of accuracy as the reference FP32-CGFDM solver in the multilayered model.

To assess time–frequency misfits between the FP16-SVE and FP32-CGFDM3D results, the envelope misfit (TFEM) and phase misfit (TFPM) of the vertical component (v_z) seismogram at station E (Fig. 5) are calculated following the criteria of Kristeková *et al.* (2006, 2009). The TFEM (Fig. 6a) is mainly concentrated around the principal seismic phase between 2 and 6 s, with dominant frequencies of 2–10 Hz. The TFPM (Fig. 6b) appears mainly in the high-frequency range, but remains negligible. Overall, both TFEM and TFPM are negligible and within an acceptable range, indicating that FP16-SVE maintains sufficient accuracy in multilayered media.

3.2 Gaussian-shaped topography model

To further challenge the optimized solvers with complex geometry, seismic wave propagation is simulated in a medium overlain by a Gaussian-shaped topography. The surface elevation is defined as

$$z(x, y) = h \exp\left(-\frac{x^2 + y^2}{a^2}\right), \tag{10}$$

with $h = 2000$ m and $a = 1000$ m. The medium parameters are $V_s = 3464$ m s^{-1}, $V_p = 6000\, m\, s^{-1}$ and $\rho = 2670$ kg m^3. The computational domain spans 30.1 km × 30.1 km with a minimum depth

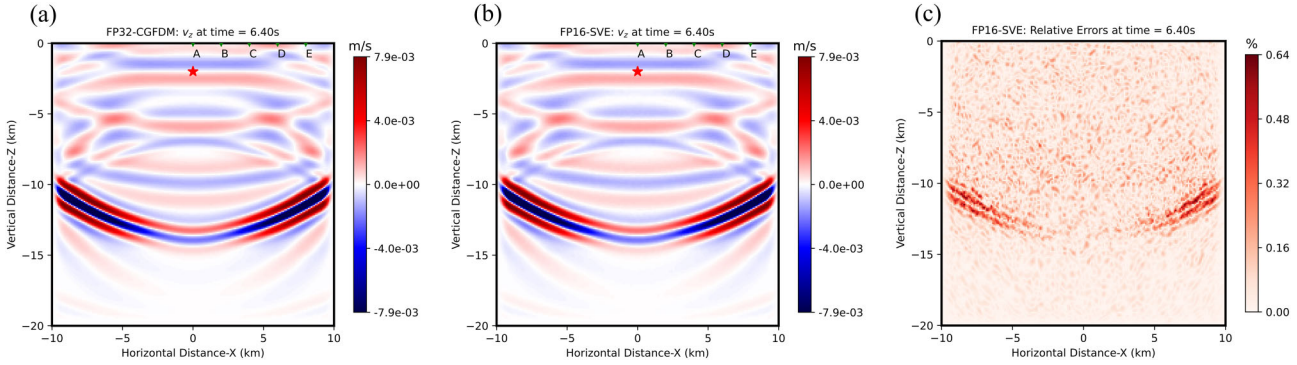

Figure 4. Vertical velocity snapshots (v_z) at $t = 6.4$ s for the 3-D half-space multilayered media model: (a) FP32-CGFDM, (b) FP16-SVE and (c) relative errors of FP16-SVE with respect to FP32-CGFDM. The inverted triangles denote the five virtual stations (A–E), and the star marks the seismic hypocentre.

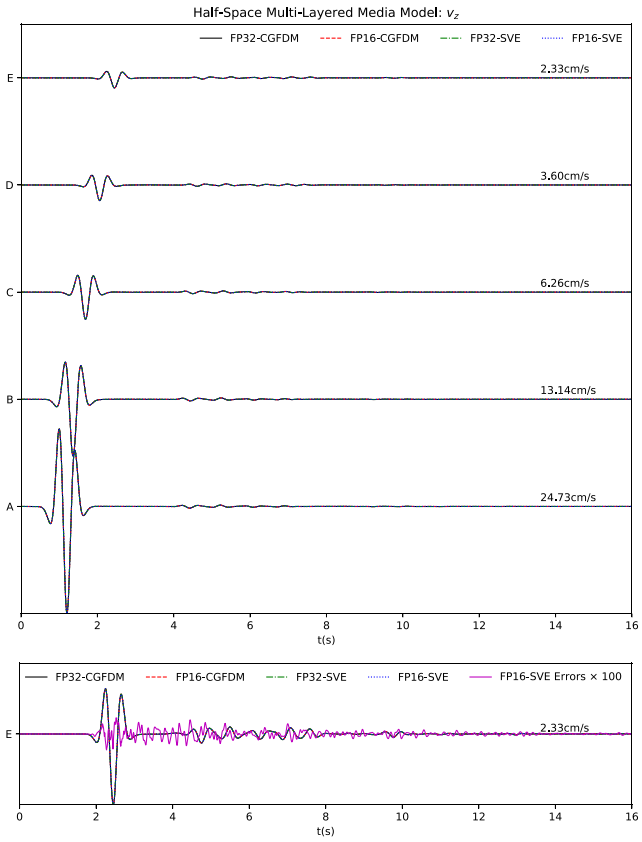

Figure 5. Synthetic vertical velocity (v_z) seismograms at virtual stations A–E (in Fig. 4). Different line styles indicate the corresponding solvers: FP32-CGFDM (solid), FP16-CGFDM (dashed), FP32-SVE (dash–dot), and FP16-SVE (dot). The lower panel presents a magnified view of station E, with the FP16-SVE error (FP16-SVE minus FP32-CGFDM) amplified 100-fold.

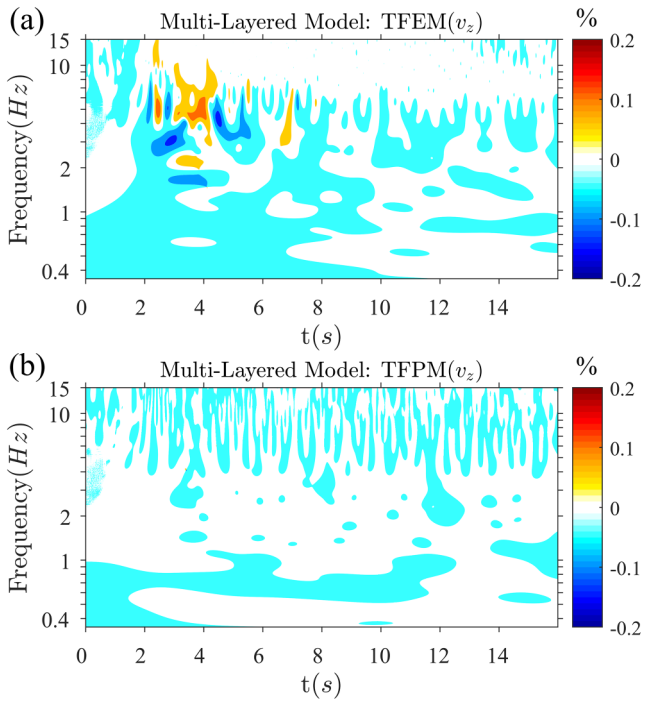

Figure 6. TFEM (a) and TFPM (b) of v_z at station E, between FP16-SVE and FP32-CGFDM within the half-space multilayered model.

of 15 km, discretized into 301 × 301 × 151 grid points. The simulation runs for 15 s with a 0.005 s time step, giving 3000 iterations. The parallel decomposition adopts 1 × 8 × 8 MPI processes (64 ARM cores). The hypocentre is located at (−5, 0, −1) km (red star in Fig. 7), and five virtual stations (A–E) are positioned along the x-axis across the Gaussian peak (green inverted triangles in Fig. 7): A (−8, 0, 0) km, B (−2, 0, 1.256) km, C (1, 0, 2.425) km, D (3, 0, 0.42) km and E (8, 0, 0) km.

Fig. 7 shows the vertical velocity snapshots at $t = 3.0$ s. FP16-SVE delivers results that are indistinguishable from those of FP32-CGFDM, confirming its full numerical fidelity. The wavefront shapes, reflection patterns around the topography and amplitude distributions are perfectly consistent. The maximum relative error is less than 0.75 per cent. Notably, no spurious reflections or artificial artifacts are introduced by the hybrid algorithm (FP16-SVE), confirming its robustness under highly irregular geometries.

The seismograms in Fig. 8 reinforce this conclusion. Across stations A–E, all optimized solvers reproduce the reference waveforms with indistinguishable arrival times and amplitudes. Even under the most challenging conditions (station E, located farthest from the seismic hypocentre shown in Fig. 7), the FP16-SVE error, when amplified by a factor of 100, remains small compared to the signal amplitude. This highlights the numerical stability and accuracy preservation of the FP16-SVE solver when applied to complex geometrical domains.

As with the multilayered model, the TFEM and TFPM of vertical component (v_z) seismogram at station E is calculated for the Gaussian-shaped model (Fig. 9). The TFEM (Fig. 9a) is concentrated around the main phase between 3 and 8 s, with dominant frequencies of 2–10 Hz. The TFPM (Fig. 9b) appears mainly at higher frequencies but remains negligible. Overall, both misfits are minimal and within acceptable limits, demonstrating that FP16-SVE provides sufficient accuracy for seismic wave propagation over complex irregular topography.

In summary, both the half-space multilayered media and Gaussian-shaped topography tests demonstrate that our optimized solvers, especially FP16-SVE, preserve the full accuracy of the FP32-CGFDM reference. The near-perfect agreement observed in wavefields, seismograms, error distributions and time-frequency misfits confirms that FP16-SVE is a robust and reliable tool for large-scale seismic simulations of complex models.

4 PERFORMANCE EVALUATION

In this section, the memory consumption and computational efficiency of the three optimized solvers are evaluated, with FP32-CGFDM serving as the reference benchmark. As discussed in Section 2.4, PML is not suitable for optimization. Therefore, it remains unoptimized and is excluded from the performance comparison.

4.1 Memory consumption analysis

In the CGFDM framework for solving the elastic wave equation, the quantities stored at each grid point include:

(i) nine wavefield components:

$$\boldsymbol{W} = (v_x, v_y, v_z, \sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{xz}, \sigma_{yz})^T$$

or

$$\boldsymbol{W} = (V_x, V_y, V_z, \Sigma_{xx}, \Sigma_{yy}, \Sigma_{zz}, \Sigma_{xy}, \Sigma_{xz}, \Sigma_{yz})^T$$

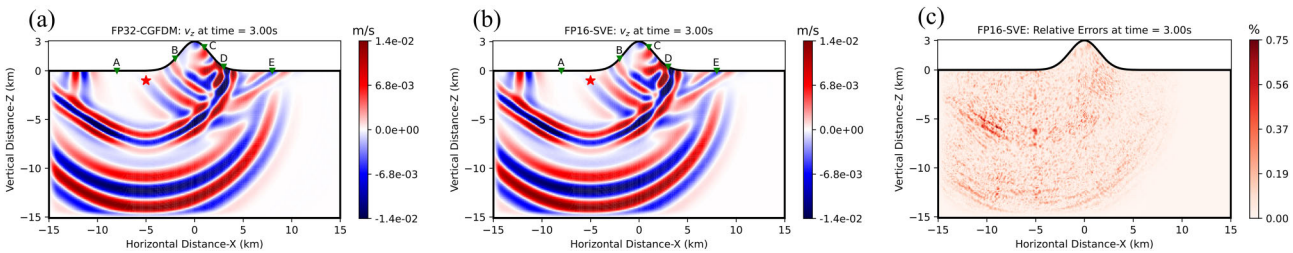

Figure 7. Vertical velocity snapshots (v_z) at $t = 3.0$ s for the Gaussian-shaped topography model: (a) FP32-CGFDM, (b) FP16-SVE and (c) relative errors of FP16-SVE relative to FP32-CGFDM. The inverted triangles indicate the five virtual stations (A–E), and the star denotes the seismic hypocentre.

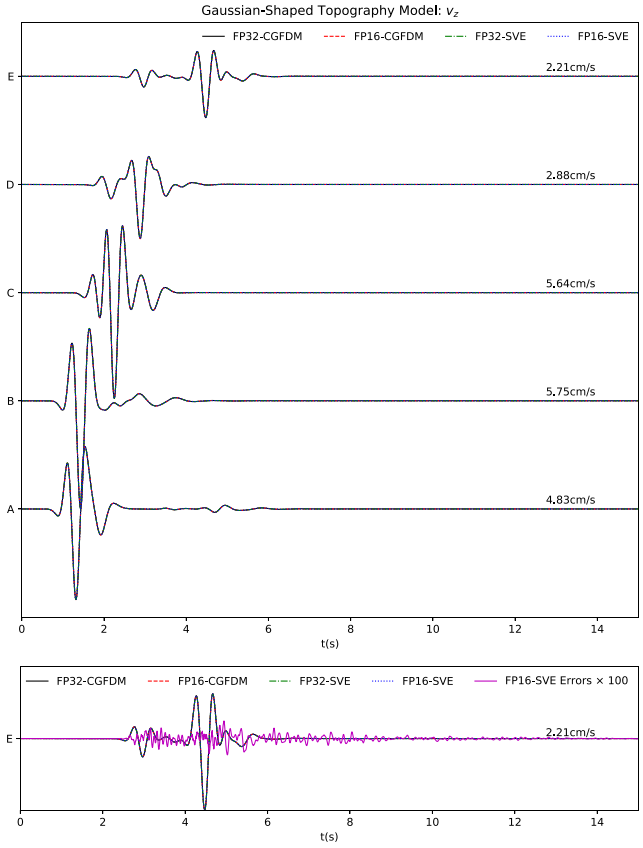

Figure 8. Synthetic vertical velocity (v_z) seismograms at virtual stations A–E shown in Fig. 7. Line styles follow the same convention as in Fig. 5. The lower panel shows a magnified view of station E, which depicts the FP16-SVE error (FP16-SVE minus FP32-CGFDM) amplified by a factor of 100.

(ii) three elastic parameters: λ, μ, ρ or Λ, M, P.
(iii) nine geometric coefficients:

$$\frac{\partial \xi}{\partial x}, \frac{\partial \xi}{\partial y}, \frac{\partial \xi}{\partial z}; \quad \frac{\partial \eta}{\partial x}, \frac{\partial \eta}{\partial y}, \frac{\partial \eta}{\partial z}; \quad \frac{\partial \zeta}{\partial x}, \frac{\partial \zeta}{\partial y}, \frac{\partial \zeta}{\partial z}.$$

(iv) one Jacobian determinant: J.

During time integration, all Runge–Kutta substeps of the nine wavefield components must be stored. Therefore, the FP32-CGFDM solver and the three optimized solvers, which employ a fourth-order Runge–Kutta scheme, require storage equivalent to four times the nine wavefield components, that is, 36 components. In addition, the other variables remain constant in time, totalling 13 components. Consequently, each grid point requires storage for 49 components in total.

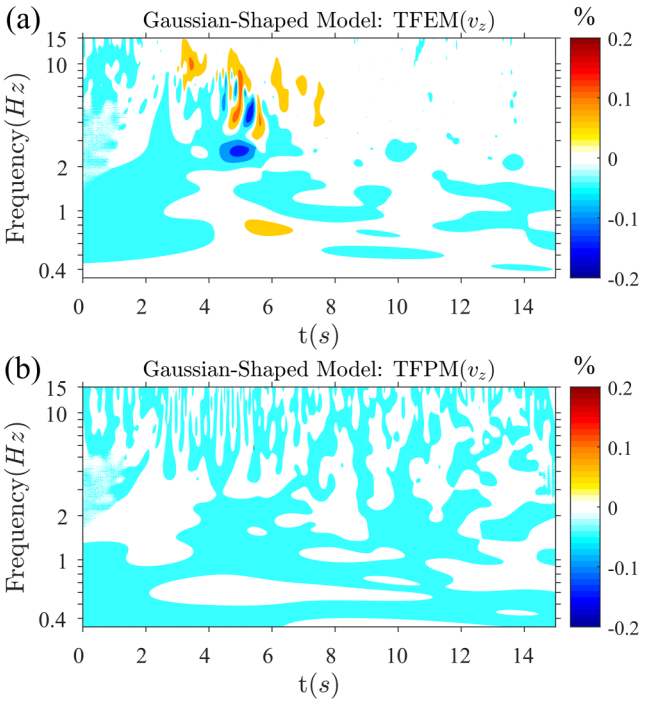

Figure 9. TFEM (a) and TFPM (b) of v_z at station E, between FP16-SVE and FP32-CGFDM within the Gaussian-shaped topography model.

For a computational grid of size $NX \times NY \times NZ$, the total memory requirement is approximately

$$\text{Memory (FP32)} = \frac{4 \times 49 \times NX \times NY \times NZ}{1024^3} \text{ GB},$$
$$\text{Memory (FP16)} = \frac{2 \times 49 \times NX \times NY \times NZ}{1024^3} \text{ GB}.$$

Using FP32-CGFDM as the benchmark with 100 per cent memory usage, the memory footprint of FP16-CGFDM is reduced to approximately 50 per cent, while FP32-SVE maintains the same memory as FP32-CGFDM. FP16-SVE reduces memory consumption to around 50 per cent of the benchmark. This reduced memory requirement demonstrates that FP16-SVE not only enables simulations of larger-scale earthquakes within the same memory capacity but also lowers computational costs.

4.2 Computational efficiency analysis

To analyse the computational efficiency of the proposed solvers and compare their performance, five benchmark models with point sources were selected. The models use different grid sizes (Fig. 10),

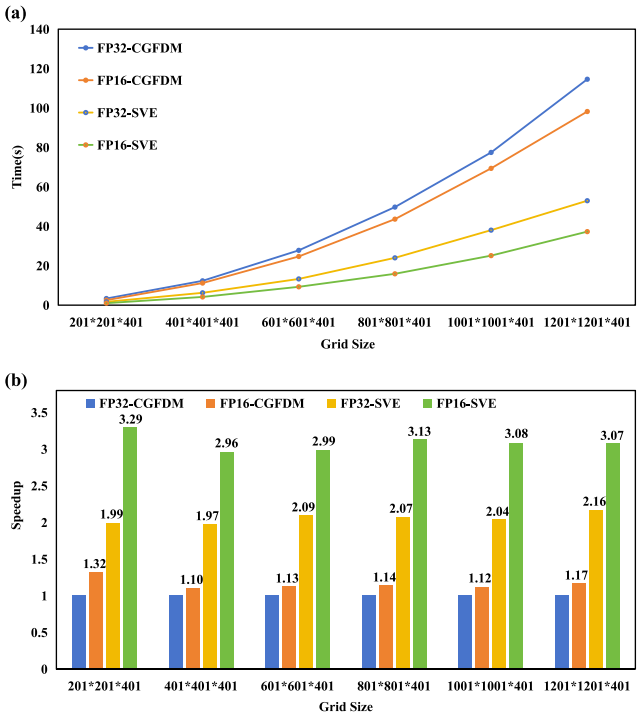

Figure 10. Computational efficiency of the FP32 solver versus three optimized solvers across different grid sizes: (a) computation time and (b) speedup factors.

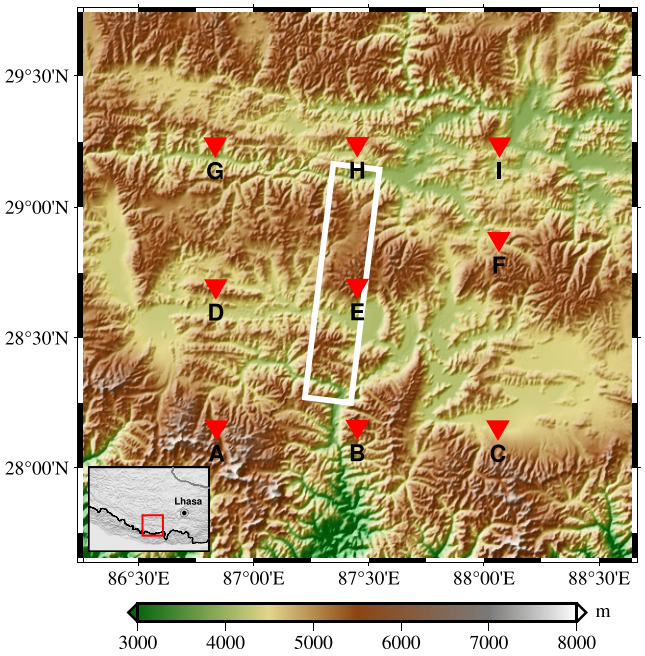

Figure 11. Geographic extent of the 2025 Tibet earthquake simulation. The rectangle indicates the surface projection of the finite-fault, and inverted triangles mark nine virtual stations (A–I).

with an MPI configuration of $1 \times 8 \times 8$ and 200 time steps. Computation times averaged over 10 steps for all models are presented in Fig. 10(a), and these times were converted into speedup factors, as shown in Fig. 10(b). It should be noted that the performance results depend on the underlying hardware platform, and all analyses reported here are based on our ARM processor with 512-bit SVE support.

Fig. 10(b) presents the measured speedups of different solvers (FP16-CGFDM, FP32-SVE and FP16-SVE) relative to the baseline FP32-CGFDM, achieving 1.16×, 2.05× and 3.09×, respectively, and FP16-SVE further attains a 1.51× speedup over FP32-SVE.

FP16-CGFDM halves the memory traffic compared to FP32-CGFDM. However, the actual speedup is only about 1.16×, far below the theoretical 2× bandwidth-driven limit. This indicates that, under scalar execution, both FP32-CGFDM and FP16-CGFDM remain compute-bound, meaning that reducing memory access alone cannot significantly improve overall performance. This motivates the need to exploit hardware compute capability through vectorization.

Next, comparing FP32-CGFDM (scalar version) with FP32-SVE, vectorization delivers a 2.05× measured speed-up. Nonetheless, with a 512-bit SVE vector capable of processing 16 elements per instruction, the theoretical upper bound is 16×. The large gap indicates that SVE compute throughput cannot be fully utilized due to insufficient memory bandwidth, meaning that under vectorized execution, FP32-SVE becomes memory-bound. Therefore, reducing memory traffic becomes essential in the SVE regime.

To address this, we combine FP16 with SVE and obtain the FP16-SVE version, which reaches 3.09× speedup over the baseline FP32-CGFDM. Relative to FP32-SVE, switching to FP16 halves total memory traffic, whose theoretical limit is a 2× speedup. The measured gain is 1.51×, confirming the effectiveness of FP16. This may stem from the additional precision-conversion and data-rearrangement operations (as illustrated in Fig. 3), which introduce extra computational overhead.

Overall, these results indicate that the combined use of FP16 storage and SVE vectorization effectively alleviates both computation and memory bottlenecks, producing a balanced and practical performance profile. Moreover, FP16-SVE combines both enhancements, achieving a speedup exceeding 3×, and demonstrating high efficiency for large-scale, high-resolution seismic simulations.

5 REAL-WORLD EARTHQUAKE SIMULATION

To further assess the accuracy, robustness and efficiency of the FP16-SVE solver in extremely complex geological environments, we conduct a real-world simulation of the 2025 Tibet earthquake. Following Wang *et al.* (2023), the dimensionless constants are set as $C_s = 1 \times 10^{-6}$, $C_v = 5 \times 10^3$, and $C_p = 1 \times 10^6$. For this earthquake simulation, the complexity of the finite-fault rupture, topography and 3-D structural heterogeneity is fully incorporated. A realistic finite-fault rupture source model released by USGS, the SRTM90 data set (Reuter *et al.* 2007), and the CSRM-1.0 model (Xiao *et al.* 2024) are used to account for their effects on seismic wave propagation. Fig. 11 illustrates the geographic extent of the simulation region and the surface projection of the finite fault (white box). Together, these data sets serve as inputs, ensuring that the seismic wave propagation is simulated under conditions that closely reflect the real-world scenarios.

The computational domain, shown in Fig. 11, spans 240 km × 240 km × 240 km, discretized into $601 \times 601 \times 151$ grid points. The parallel decomposition employs $4 \times 8 \times 2$ MPI processes. The total simulation duration is 80 s with a time step of 0.025 s, resulting in 3200 integration steps.

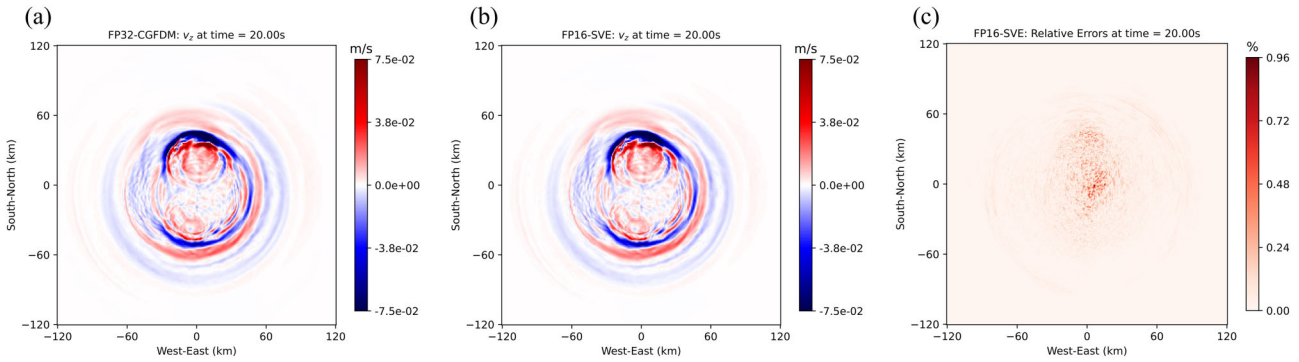

Figure 12. Ground motions of vertical velocity (v_z) snapshots at $t = 20$ s for the 2025 Tibet earthquake, showing (a) FP32-CGFDM, (b) FP16-SVE and (c) relative errors of FP16-SVE with respect to FP32-CGFDM.

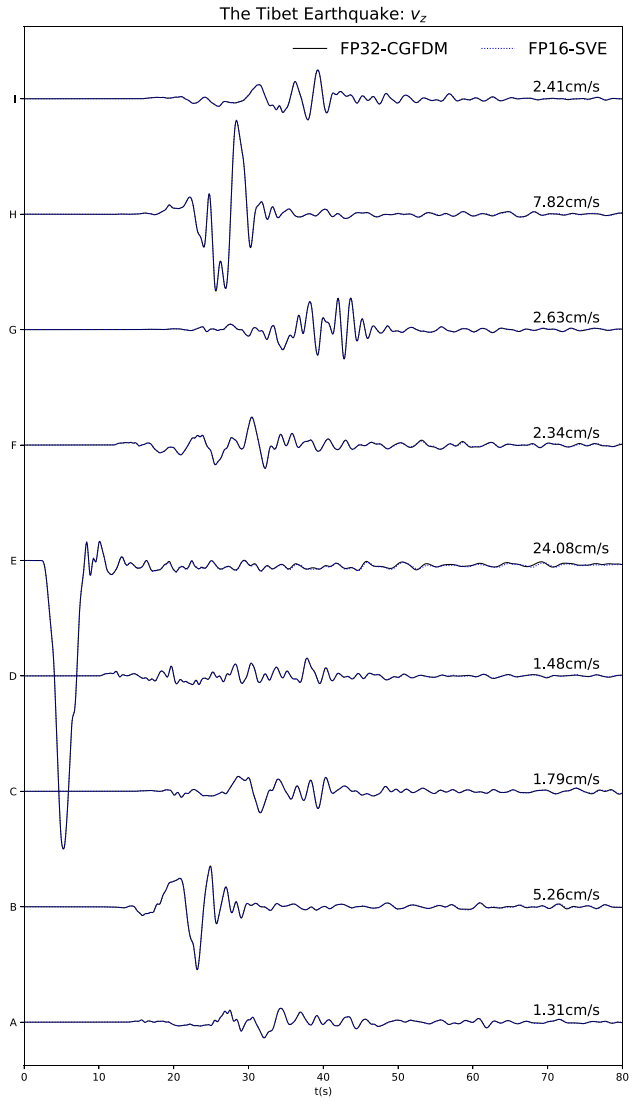

Figure 13. Synthetic seismograms of vertical velocity (v_z) for the 2025 Tibet earthquake, recorded at nine virtual stations (A–I). Solid lines correspond to FP32-CGFDM, dashed lines represent FP16-SVE, and the values at the end of each trace indicate the maximum absolute value of v_z at the corresponding station.

In realistic earthquake simulations, PML is necessary to prevent artificial reflections from affecting the true wavefield. Two sets of tests were conducted to evaluate performance. FP16-SVE achieved significant speedups over FP32-CGFDM: with PML, 27.329 min versus 62.951 min (2.3× speedup); without PML, 19.037 min versus 57.121 min (3× speedup). These results demonstrate that, even with the additional computational cost of PML, FP16-SVE maintains high efficiency and robustness, highlighting its suitability for realistic large-scale earthquake simulations. The slightly lower speedup observed with PML is mainly due to load imbalance: PML computations are performed only at the boundaries, that is, on boundary MPI processes, which can create uneven workloads (Wan *et al.* 2023). As noted in Section 2.4, PML is not well suited for FP16 optimization, and therefore its optimization is not attempted in this study. A comprehensive investigation of PML optimization lies beyond the present scope and may be addressed in future research.

Fig. 12 presents the simulated ground motions of the vertical velocity (v_z) snapshots at $t = 20$ s for FP32-CGFDM (a) and FP16-SVE (b), with the relative error distribution shown in panel (c). The two wavefields exhibit almost identical radiation patterns and amplitude distributions, with relative errors below 1 per cent across the entire domain. This confirms that the FP16-SVE solver can robustly capture complex wavefield features, even under realistic earthquake source and media conditions.

Fig. 13 shows synthetic seismograms of the vertical velocity (v_z) recorded at nine virtual stations, A–I, indicated by red inverted triangles in Fig. 11. The comparison between FP32-CGFDM and FP16-SVE demonstrates excellent consistency in both arrival times and amplitude evolution. The maximum deviations remain negligible relative to the signal amplitude, further supporting the accuracy of the hybrid solver.

Peak ground velocity maps for the horizontal component are displayed in Fig. 14. Subplots (a) and (b) correspond to FP32-CGFDM and FP16-SVE, respectively. The spatial patterns of ground motion are highly consistent, with both methods producing comparable PGV distributions. No systematic bias or artificial anomalies are observed, which highlights the reliability of FP16-SVE in seismic hazard assessment.

Overall, the Tibet earthquake case demonstrates that the FP16-SVE solver achieves FP32-level accuracy even in highly complex geological environments, making it suitable for real-world seismic simulations and earthquake hazard studies. Moreover, it delivers a more than 2.3× speedup over FP32-CGFDM, highlighting its potential for operational rapid-response simulations and long-term seismic hazard assessments.

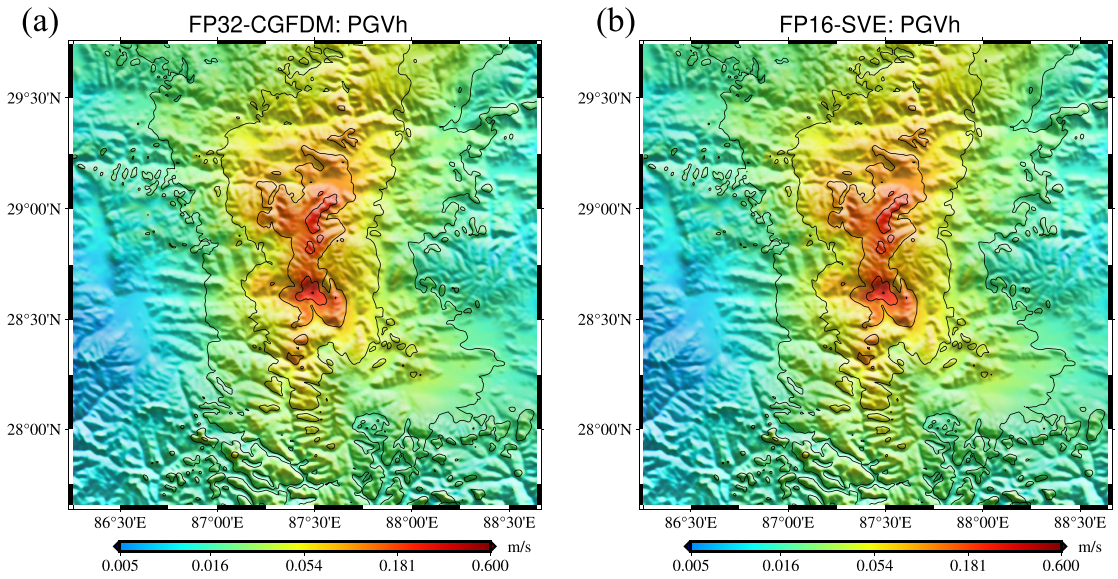

Figure 14. Peak ground velocity (PGV) distribution maps for the horizontal component of the 2025 Tibet earthquake: (a) FP32-CGFDM and (b) FP16-SVE.

6 CONCLUSIONS

This study developed three seismic simulation solvers on the ARM architecture: FP16-CGFDM, FP32-SVE and FP16-SVE. In particular, the proposed FP16-SVE hybrid solver reduces memory consumption, improves memory access efficiency and accelerates computation, without compromising accuracy.

Extensive numerical experiments on half-space multilayered media and Gaussian-shaped topography models demonstrated that the FP16-SVE solver achieves FP32-level accuracy, with relative errors below 1 per cent across complex geological conditions. Performance benchmarks confirmed that FP16-SVE halves memory consumption and simultaneously achieves speedups exceeding 3× compared to FP32-CGFDM, enabling larger-scale simulations while reducing computational costs.

The 2025 Tibet earthquake case study further validated the FP16-SVE solver's applicability in real-world earthquake simulations. Even under extremely complex geological environments, the FP16-SVE solver achieved FP32–level accuracy in waveforms, ground motion snapshots and final PGV distributions, demonstrating a more than 2.3× speedup. This highlights its potential for rapid-response earthquake simulations and long-term seismic hazard assessment.

Overall, the FP16-SVE hybrid solver provides a robust and efficient framework for high-resolution, large-scale seismic simulations on ARM platforms, accurately capturing extreme geological structures. To promote reproducibility and facilitate further research, the codes and numerical examples developed in this study are open-sourced (shown in Section 7). This enables practical applications in both rapid-response earthquake analysis and reliable seismic hazard assessment.

ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to the editor, Dr. Gaetano Festa, and the reviewers, Dr. Jozef Kristek and Dr. Etienne Bachmann, for their time and valuable comments which have significantly improved the quality of this manuscript. This work was supported by Shenzhen Science and Technology Program (Grant No. JCY20241202125759001), the National Natural Science Foundation of China (Grant No. T2125006), the China Postdoctoral Science Foundation (Grant No. 2024M760678), Shenzhen Science and Technology Program (Grant Nos. KCXFZ20240903093759004 and KJZD20230923115106012) and Jiangsu Innovation Capacity Building Program (Grant No. BM2022028).

DATA AVAILABILITY

All source codes of our proposed solvers (FP16-CGFDM, FP32-SVE, FP16-SVE) in this work, has been open-sourced on GitHub: https://github.com/wenqiangwangsustech/FP16-SVE. In addition, the CGFDM3D-EQR code developed by Wang *et al.* (2022) has been released as open source on GitHub at https://github.com/wenqiangwangsustech/CGFDM3D-EQR. The time–frequency envelope misfit package TF_MISFIT_GOF_CRITERIA (Kristeková *et al.* 2006, 2009) can be accessed at https://www.nuquake.eu/Computer_Codes/.

REFERENCES

Aki, K. & Richards, P.G., 2002. *Quantitative Seismology,* (2nd edn), University Science Books.

Arm, 2025. *Arm Architecture Reference Manual for A-profile architecture,* Arm Community

Chen, B. *et al.*, 2018. Simulating the Wenchuan earthquake with accurate surface topography on Sunway TaihuLight, *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis,* IEEE, pp.517–528.

Clark, M., Babich, R., Barros, K., Brower, R. & Rebbi, C., 2010. Solving lattice QCD systems of equations using mixed precision solvers on GPUs, *Comput. Phys. Commun.*, **181**(9), 1517–1528.

Courbariaux, M., Bengio, Y. & David, J.P., 2014. Training deep neural networks with low precision multiplications, *arXiv preprint*. arXiv:1412.7024

Fu, H. *et al.*, 2017. 18.9-Pflops nonlinear earthquake simulation on Sunway TaihuLight: enabling depiction of 18-Hz and 8-meter scenarios, *SC '17: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis,* Association for Computing Machinery, New Yourk, USA, pp.1–12.

Graves, R.W., 1996. Simulating seismic wave propagation in 3D elastic media using staggered-grid finite differences, *Bull. seism. Soc. Am.,* **86**(4), 1091–1106.

Hennessy, J.L. & Patterson, D.A., 2011. *Computer Architecture: A Quantitative Approach,* Elsevier.

Hixon, R., 1997. On increasing the accuracy of MacCormack schemes for aeroacoustic applications, *3rd AIAA/CEAS Aeroacoustics Conference,* National Aeronautics and Space Administration, pp. 1–28

Hixon, R., 1998. Evaluation of a high-accuracy MacCormack-type scheme using benchmark problems, *J. Comput. Acoust.,* **06**(03), 291–305.

IEEE, 2008. IEEE Standard for Floating-Point Arithmetic, *IEEE Std 754-2008,* 1–70.

Kristeková, M., Kristek, J., Moczo, P. & Day, S.M., 2006. Misfit criteria for quantitative comparison of seismograms, *Bull. seism. Soc. Am.,* **96**(5), 1836–1850.

Kristeková, M., Kristek, J. & Moczo, P., 2009. Time-frequency misfit and goodness-of-fit criteria for quantitative comparison of time signals, *Geophys. J. Int.,* **178**(2), 813–825.

Micikevicius, P. *et al.*, 2018. Mixed Precision Training. *International Conference on Learning Representations*. Vancouver Convention Center, Vancouver Canada.

Reuter, H.I., Nelson, A. & Jarvis, A., 2007. An evaluation of void-filling interpolation methods for SRTM data, *Int. J. Geograph. Inf. Sci.,* **21**(9), 983–1008.

Virieux, J., 1986. P-SV wave propagation in heterogeneous media: velocity-stress finite-difference method, *Geophysics,* **51**(4), 889–901.

Wan, J., Wang, W. & Zhang, Z., 2024. Enhancing computational efficiency in 3-D seismic modelling with half-precision floating-point numbers based on the curvilinear grid finite-difference method, *Geophys. J. Int.,* **238**(3), 1595–1611..

Wan, W. *et al.*, 2023. 69.7-pflops extreme scale earthquake simulation with crossing multi-faults and topography on sunway, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis,* SC'23, Association for Computing Machinery, New York, USA.

Wang, W., Zhang, Z., Zhang, W., Yu, H., Liu, Q., Zhang, W. & Chen, X., 2022. CGFDM3D-EQR: A platform for rapid response to earthquake disasters in 3D complex media, *Seismol. Res. Lett.,* **93**(4), 2320–2334.

Wang, W., Zhang, Z., Zhang, W. & Liu, Q., 2023. Implementation of efficient low-storage techniques for 3-d seismic simulation using the curved grid finite-difference method, *Geophys. J. Int.,* **234**(3), 2214–2230.

Wang, Y. *et al.*, 2025. Accelerating half-precision seismic simulation on neural processing unit, *IEEE Trans. Parallel Distrib. Syst.,* **36**(10), 1998–2013.

Xiao, X. *et al.*, 2024. Csrm-1.0: A china seismological reference model, *J. Geophys. Res.: Solid Earth,* **129**(9), e2024JB029520.

Zhang, W. & Chen, X., 2006. Traction image method for irregular free surface boundaries in finite difference seismic wave simulation, *Geophys. J. Int.,* **167**(1), 337–353.

Zhang, W. & Shen, Y., 2010. Unsplit complex frequency-shifted PML implementation using auxiliary differential equations for seismic wave modeling, *Geophysics,* **75**(4), T141–T154.

Zhang, W., Zhang, Z. & Chen, X., 2012. Three-dimensional elastic wave numerical modelling in the presence of surface topography by a collocated-grid finite-difference method on curvilinear grids, *Geophys. J. Int.,* **190**(1), 358–378.