Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Large-scale frequency-domain elastic wave modeling with column-norm-weighted CARP-CG

Mao Rui
Harbin Institute of Technology
Yang Li
yang.li.math@hit.edu.cn
Harbin Institute of Technology
Zhenwu Fu
Harbin Institute of Technology
Guangyu Gao
Harbin Institute of Technology
Yong Chen
Harbin Institute of Technology

Research Article

Keywords:

Posted Date: November 2nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11540866/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Large-scale frequency-domain elastic wave modeling with column-norm-weighted CARP-CG

Rui Mao[a], Yang Li[a,*], Zhenwu Fu[a], Guangyu Gao[a] and Yong Chen[a]

[a]School of Mathematics, Harbin Institute of Technology, No. 92 West Dazhi Street, Harbin, 150001, China

ARTICLE INFO

Keywords:
Frequency-domain elastic wave modeling
Weighted CARP-CG
Row-projection methods
Column-norm weighting
Weighted inner products
Sparse matrix storage

ABSTRACT

Frequency-domain elastic wave modeling requires solving a large-scale sparse linear system. To avoid unaffordable memory costs when direct solvers are used, we resort to iterative methods. An attractive choice is the component-averaged row projections method with conjugate-gradient acceleration (CARP-CG), which has been shown to be robust in complicated physical settings. To avoid its slow convergence, we develop the Weighted CARP-CG (WCARP-CG) method, in which squared column norms of row-scaled local matrices, a positive shift, and an exponent β define fixed positive weights. These weights determine local row updates and component averaging in the W-inner product and induce the M-inner product for CG on original variables. For consistent systems, the CG system and the original equations have the same solutions, and CG converges in exact arithmetic. At $\beta = 0.5$, weighting decreases condition numbers computed from positive eigenvalues by factors of 2.77–3.38 in small-scale tests, relative to equal weights. We verify the accuracy of the WCARP-CG solutions by comparisons with direct solver solutions. Across six three-dimensional tests combining frequencies of 0.5–3 Hz with different grid spacings and two further tests at different Poisson's ratios, the number of iterations decreases by 51.8–65.5% and iteration-loop time decreases by 45.1–61.7% relative to CARP-CG at a relative residual of 10^{-4} for the row-scaled system. We also use an improved CSR storage strategy that reduces coefficient and column-index storage to 5% of the conventional explicit format.

1. Introduction

Frequency-domain elastic modeling solves a large sparse linear system for each selected frequency [1, 2]. The frequency responses can be computed independently, and different sources at the same frequency share the system matrix. These properties make the formulation useful for seismic imaging and full-waveform inversion with selected frequencies [3].

The main computational difficulty is the size and conditioning of these systems. Resolving the shortest shear wavelength requires a fine grid, with three coupled velocity degrees of freedom at each three-dimensional grid point. Material heterogeneity and boundary treatment further affect the indefinite, generally non-Hermitian system and make the system difficult to solve [1, 2]. Efficient modeling therefore requires control of both memory use and solution time.

Sparse direct methods can reuse a matrix factorization for many sources, but fill-in during factorization can make the factors too large to fit in memory in three dimensions [4, 5]. Iterative methods offer a way to solve such systems using matrix–vector products, a set of vectors, and preconditioner data, without storing a global factorization. They are particularly useful when the matrix fits in memory but its factors do not. Their practical cost then depends on how much work each iteration requires and how many iterations are needed.

Developments for the elastic Helmholtz equation include shifted-Laplacian multigrid with smoothers adapted to elastic coupling [6] and a hybrid multigrid/domain-decomposition preconditioner [7]. Yovel and Treister use local Fourier analysis to tune coarse-grid stencils in a matrix-free multigrid method [8]; their later block-acoustic preconditioner reduces elastic block solves to acoustic Helmholtz solves [9]. El-WaveHoltz takes a different approach, filtering time-domain elastic solutions to obtain a Krylov-accelerated iteration for time-harmonic problems with energy-conserving boundaries [10]. Here we investigate a row-projection approach to reducing iteration cost while retaining local sweeps and short recurrences.

*Corresponding author. Postal address: Room 1006, Science Building, School of Mathematics, Harbin Institute of Technology, No. 92 West Dazhi Street, Harbin 150001, China.
Email address: yang.li.math@hit.edu.cn (Y. Li)

We adopt component-averaged row projections with conjugate-gradient acceleration (CARP-CG) for their low storage requirements and independent subdomain sweeps [1, 11]. CARP divides the equation rows among subdomains, each of which applies Kaczmarz row updates in a fixed order using local variable copies [12]. Component averaging combines the copies of shared variables [13]. CG accelerates the forward–backward double sweep [11, 14]. The method requires no global sparse factorization or assembled normal-equation matrix. Its short recurrences keep a fixed number of vectors, and the subdomains perform their sweeps independently between communication steps. Its row operations also allow coefficients to be generated when needed.

A low cost per sweep does not ensure a low total cost. CARP-CG can require many iterations in heterogeneous media and at higher Poisson's ratios [1, 10]. When a double sweep changes some error components only slightly, many repetitions of the row updates, exchanges of shared values, and global reductions may be needed. We seek to reduce the iteration count while preserving the local sweeps, short recurrences, and parallel structure.

Earlier projection methods combine component-dependent weights with parallel updates. Component averaging (CAV) is a simultaneous method: its row corrections use the same current iterate [15]. BICAV applies such updates to successive blocks [16]. Their diagonal weights reflect column sparsity [17], whereas CARP performs ordered row updates within each parallel subdomain before averaging shared copies [11, 13]. Superspace formulations, which collect local variable copies in a larger vector space, also relate projection and averaging operations to Kaczmarz methods and CG acceleration [18]. We retain the CARP sweep structure and construct weights from local matrix coefficients rather than column sparsity. We refer to the resulting algorithm as Weighted CARP-CG.

Classical row projections and arithmetic component averaging use the same Euclidean inner product on the local copies. Arithmetic averaging minimizes the unweighted sum of squared differences between each copy of a shared variable and a common value. It gives these copies equal influence, although they contribute to different sets of local equations. This motivates an averaging objective that accounts for each copy's contribution to its local equations.

Each column of a local elastic matrix records how one velocity variable contributes to the subdomain equations. Its entries depend on density, wave speeds, frequency, and boundary treatment. After row scaling, the squared column norm measures the squared norm of the local residual change caused by a unit change in that variable. We add a positive shift and raise the result to an exponent $\beta \in [0, 1]$. The shift keeps every weight positive, and the exponent controls the differences between weights. Copies of the same variable may have different weights because they belong to different local equation sets. Section 3.1 motivates the column-norm weights through a diagonal approximation of the changes in the local equations caused by averaging.

The same copy weights define the inner product for local row projections and component averaging. Summing the weights of all copies of each original variable defines the inner product used by CG. The local column norms select a coefficient-dependent metric within this construction. We prove that the matrix used by CG is self-adjoint and positive semidefinite in this induced inner product. For consistent systems under the stated sweep conditions, the CG system and the original equations have the same solutions. Section 3 gives the construction and CG convergence analysis.

A compact storage implementation supports large three-dimensional problems. Building on compressed sparse row (CSR) storage, stencil-based encoding, and on-demand coefficient generation [19–21], we combine stencil-offset codes, coefficient regeneration, and reuse of a three-row buffer. We call this representation CSR with stencil encoding and regeneration (CSR-SER). Both solvers use it. CSR-SER preserves the row-update order and reduces persistent coefficient and column-index storage to one-twentieth of that required by the corresponding explicit CSR arrays.

We assess the method using two models of spatially varying subsurface elastic properties: a two-dimensional model derived from Marmousi2 and a modified three-dimensional Overthrust model. Both are discretized with optimized fourth-order staggered-grid finite differences. The tests examine the choice of the weighting exponent β and the accuracy of the computed wavefields. They also measure iteration counts and iteration-loop time as we vary grid spacing, wave frequency, source location, and how the equation rows are divided among parallel processes. The surface-boundary tests compare configurations with and without an absorbing layer at the model surface. Small three-dimensional systems allow us to compute all eigenvalues of the matrices used by CG. The computational cost analysis includes six three-dimensional tests combining frequencies of 0.5–3 Hz with different grid spacings, and two further comparisons at different Poisson's ratios to vary the relative stiffness in volumetric and shear deformation. Across these eight solver comparisons, measured iteration-loop time decreases by 45.1–61.7% relative to CARP-CG. The CSR-SER implementation used by both solvers also handles approximately 70 million complex-valued velocity degrees of freedom (DOFs) on a 32 GiB workstation.

2. Elastic system and classical CARP-CG

2.1. Discrete elastic system

We consider two- and three-dimensional frequency-domain elastic wave propagation in isotropic media. The governing equations are the first-order velocity–stress equations [1, 22]. With Fourier-transform kernel $\exp(-i\omega t)$, their tensor form is

$$\begin{aligned} i\omega\rho\mathbf{v} &= \nabla\cdot\boldsymbol{\tau} + \mathbf{f}, \\ i\omega\boldsymbol{\tau} &= \lambda(\nabla\cdot\mathbf{v})\mathbf{I} + \mu\left[\nabla\mathbf{v} + (\nabla\mathbf{v})^T\right] + \mathbf{S}_\tau. \end{aligned} \tag{1}$$

Here $\mathbf{v}$ is particle velocity, $\boldsymbol{\tau}$ is the symmetric stress tensor, ρ is density, λ and μ are the Lamé parameters, and ω is angular frequency. The vector $\mathbf{f}$ and tensor $\mathbf{S}_\tau$ represent body-force and stress-rate sources. In three dimensions, $\mathbf{v} = (v_x, v_y, v_z)^T$ and the independent stress components are $\tau_{xx}, \tau_{yy}, \tau_{zz}, \tau_{xy}, \tau_{xz}, \tau_{yz}$. The two-dimensional P–SV (compressional and vertically polarized shear) formulation suppresses the y dependence and uses two velocity components, v_x, v_z, and three stress components, $\tau_{xx}, \tau_{zz}, \tau_{xz}$. Attenuation is represented through complex wave speeds controlled by the quality factor Q_{att}; smaller values indicate stronger attenuation, and $Q_{\text{att}} = \infty$ gives the lossless limit. The corresponding complex Lamé parameters are specified in Supplementary material, Section S1.1.

We discretize the stress-divergence and velocity-gradient operators on a staggered grid before eliminating stress. The benchmarks use the optimized fourth-order finite-difference scheme with anti-lumped mass (ALM) weighting [1, 2]. The ALM coefficients distribute the discrete mass term over neighboring velocity samples and are fixed by the discretization. Perfectly matched layers (PMLs) absorb outgoing waves [23, 24]. The two-dimensional free-surface tests impose zero traction at the top; the three-dimensional surface configuration is specified in Supplementary material, Section S1.6. The discretized stress equations express the stresses in terms of velocity and source values. Substituting these expressions into the momentum equations gives the reduced velocity system

$$\mathbf{A}_{\text{FD}}\mathbf{x} = \mathbf{b}_{\text{FD}}, \tag{2}$$

where $\mathbf{x} \in \mathbb{C}^n$ contains the complex-valued particle-velocity DOFs and $\mathbf{b}_{\text{FD}}$ is the source vector. The experiments solve this reduced system, with two velocity DOFs per computational grid point in two dimensions and three in three dimensions, including the PML points. The reduced coefficients retain the staggered-grid placement of the spatially varying material parameters. Supplementary material, Section S1 gives the component equations, grid layout, coefficient and source conventions, and boundary formulas.

Both solvers apply the same diagonal row scaling. For a nonzero row $\mathbf{a}_i^{\text{FD}}$ and prescribed relaxation $0 < \theta_i < 2$, define

$$\mathbf{A} = \mathbf{D}_r\mathbf{A}_{\text{FD}}, \qquad \mathbf{b} = \mathbf{D}_r\mathbf{b}_{\text{FD}}, \qquad (\mathbf{D}_r)_{ii} = \frac{\sqrt{\theta_i}}{\left\|\mathbf{a}_i^{\text{FD}}\right\|_2}, \qquad \mathbf{A}\mathbf{x} = \mathbf{b}. \tag{3}$$

The norm $\|\cdot\|_2$ is Euclidean, and each scaled row has norm $\sqrt{\theta_i}$. Nonsingular row scaling preserves the solution set. From this point onward, $\mathbf{A}$ denotes the row-scaled matrix; its entries, including the prescribed scaling, determine the local squared column norms used to construct the weights.

2.2. Classical CARP-CG

CARP assigns groups of equation rows to subdomains. A finite-difference equation can involve velocity unknowns that also occur in rows assigned to other subdomains. Each subdomain keeps its own copies of these shared unknowns so that it can update its rows independently [11, 13].

A double sweep starts by copying the global variable values to the subdomains. Each subdomain updates its rows in forward order, after which component averaging assigns a common value to every copy of each shared variable. The subdomains then sweep in reverse row order and average again. CG accelerates the iteration defined by this complete double sweep.

To describe these operations, partition the global row indices into disjoint sets $\mathcal{J}_q$, $q = 1, \ldots, N_{\text{sub}}$, where N_{sub} is the number of subdomains. Let $\mathcal{V}_q$ contain the original-variable indices in the fixed stored column pattern of subdomain q. We assume that every original variable belongs to at least one $\mathcal{V}_q$ and every row is nonzero. Set $n_q = |\mathcal{V}_q|$, and let the

restriction $\mathbf{C}_q : \mathbb{C}^n \to \mathbb{C}^{n_q}$ select these variables. The local matrix $\mathbf{A}_q \in \mathbb{C}^{|\mathcal{J}_q| \times n_q}$ and right-hand side $\mathbf{b}_q$ are defined by

$$\mathbf{b}_q = \mathbf{b}[\mathcal{J}_q], \qquad \mathbf{A}[\mathcal{J}_q, :] = \mathbf{A}_q \mathbf{C}_q. \tag{4}$$

The variable sets $\mathcal{V}_q$ may overlap; $\mathcal{I}_j = \{q : j \in \mathcal{V}_q\}$ lists the subdomains that contain variable j. The space of all local copies, called the CARP superspace, is $\mathcal{X} = \prod_{q=1}^{N_{\text{sub}}} \mathbb{C}^{n_q} \simeq \mathbb{C}^{n_c}$, where $n_c = \sum_q n_q$. Stacking the restrictions defines the duplication map $\mathbf{C} : \mathbb{C}^n \to \mathcal{X}$ and the multiplicity matrix $\mathbf{M}_0$:

$$\mathbf{C}\mathbf{x} = (\mathbf{C}_1\mathbf{x}, \dots, \mathbf{C}_{N_{\text{sub}}}\mathbf{x}), \qquad \mathbf{M}_0 = \mathbf{C}^*\mathbf{C}. \tag{5}$$

Here * denotes conjugate transpose and $(\mathbf{M}_0)_{jj} = |\mathcal{I}_j|$ counts the copies of variable j. The consensus subspace $\mathcal{S} = \operatorname{range}(\mathbf{C})$ consists of vectors whose copies of each original variable agree.

Classical component averaging maps local copy values to the original variable space through $\mathbf{G}_0 = \mathbf{M}_0^{-1}\mathbf{C}^*$. The projector $\mathbf{E}_0 = \mathbf{C}\mathbf{G}_0$ assigns the arithmetic mean of each variable's local values to all copies of that variable. For a variable shared by two subdomains, $\mathbf{C}$ creates its two local copies, $\mathbf{G}_0$ returns their mean $(z_{1j} + z_{2j})/2$, and $\mathbf{E}_0$ writes that mean to both copies. The map $\mathbf{E}_0$ is the Euclidean orthogonal projector onto the consensus subspace $\mathcal{S}$; it assigns common copy values that minimize the unweighted sum of squared differences from the input copies. Bold symbols denote matrices or vectors, and $\mathbf{I}$ denotes the identity on the relevant space.

For a local copy vector $\mathbf{z}_q \in \mathbb{C}^{n_q}$ and local row $\iota = (q, r)$, let $\boldsymbol{\ell}_\iota = \mathbf{A}_q[r, :]^*$ and write the row equation as $\boldsymbol{\ell}_\iota^* \mathbf{z}_q = b_\iota$. Here r indexes a local row of $\mathbf{A}_q$; b_ι and θ_ι are the right-hand-side entry and relaxation parameter of its corresponding global row. One relaxed Kaczmarz projection updates the local vector $\mathbf{z}_q$ as

$$\mathcal{K}_\iota(\mathbf{z}_q; b_\iota) = \mathbf{z}_q + \theta_\iota \frac{b_\iota - \boldsymbol{\ell}_\iota^* \mathbf{z}_q}{\boldsymbol{\ell}_\iota^* \boldsymbol{\ell}_\iota} \boldsymbol{\ell}_\iota, \qquad 0 < \theta_\iota < 2. \tag{6}$$

For a fixed inner product, the row-scaling factor cancels between the residual, row direction, and denominator in a normalized Kaczmarz correction. The factor θ_i in equation 6 separately sets the relaxation of that correction. The column norms used to construct the weights in Section 3.1 are computed after the prescribed row scaling.

The double sweep applies a forward sweep, averaging, a backward sweep, and a second averaging. Including the right-hand side gives the affine map $\mathcal{D}_0(\mathbf{x}; \mathbf{b}) = \mathbf{Q}_0\mathbf{x} + \mathbf{c}_0$. A fixed point satisfies $\mathbf{x} = \mathcal{D}_0(\mathbf{x}; \mathbf{b})$, which can be rewritten as a linear system for CG. To identify its matrix, set the row right-hand sides to zero and let $\mathbf{T}_f^{(0)}$ be the block-diagonal forward-sweep operator, with one ordered product of the resulting linear row maps for each subdomain. Reversing the order gives $\mathbf{T}_b^{(0)} = (\mathbf{T}_f^{(0)})^*$. The double-sweep matrix and the fixed-point system are

$$\mathbf{Q}_0 = \mathbf{G}_0\mathbf{T}_b^{(0)}\mathbf{E}_0\mathbf{T}_f^{(0)}\mathbf{C}, \qquad \mathbf{H}_0 = \mathbf{I} - \mathbf{Q}_0, \qquad \mathbf{H}_0\mathbf{x} = \mathbf{c}_0. \tag{7}$$

Here $\mathbf{H}_0 = \mathbf{I} - \mathbf{Q}_0$ is the matrix of the system solved by CG, and $\mathbf{Q}_0$ is the linear map obtained by setting the right-hand side to zero. Every solution of $\mathbf{A}\mathbf{x} = \mathbf{b}$ is a fixed point of $\mathcal{D}_0$.

The duplication map induces the inner product $\langle \mathbf{x}, \mathbf{u} \rangle_{M_0} = \mathbf{x}^* \mathbf{M}_0 \mathbf{u}$. For real relaxations in $(0, 2)$, $\mathbf{H}_0$ is self-adjoint and positive semidefinite in this inner product. Self-adjointness means that the matrix can be moved between the two arguments without changing the inner product; positive semidefiniteness means that its quadratic form is nonnegative. These properties follow from the $\mathbf{W} = \mathbf{I}$ case of Theorem 1. The identity $\langle \mathbf{C}\mathbf{x}, \mathbf{C}\mathbf{u} \rangle_2 = \langle \mathbf{x}, \mathbf{u} \rangle_{M_0}$ links the Euclidean inner product on local copies to the induced inner product on the original variables. Section 5.1 specifies the inner products used in the reported implementations.

The double-sweep construction allows CG acceleration. Its efficiency depends on how much the sweeps and averaging reduce the remaining error. An eigenvalue of $\mathbf{H}_0$ close to zero means that the double sweep with a zero right-hand side changes the corresponding error direction very little. We therefore use local squared column norms to change the row corrections and the averaging of shared copies, and derive the inner product for CG applied to this weighted double sweep.

3. Weight construction and analysis of Weighted CARP-CG

We construct the weights and their associated inner products on local copies and original variables, then define component averaging, row updates, and the complete CG iteration. The subsequent analysis establishes self-adjointness, positive semidefiniteness, solution-set equivalence, and CG convergence.

3.1. Weight design and the induced metric

For every local variable copy, the corresponding column of $\mathbf{A}_q$ records its contribution to all equations in subdomain q. In the three-dimensional elastic equations, for example, v_x contributes to normal stresses through $\lambda + 2\mu$ and λ, and to shear stresses through μ. After stress elimination, these terms couple different velocity components through mixed spatial derivatives. Density, frequency, and boundary treatment also affect these coefficients. After row scaling, we quantify this contribution by the squared norm of the corresponding local column. For $j \in \mathcal{V}_q$, let $\mathbf{e}_{qj}$ be the corresponding coordinate vector and define

$$\begin{aligned} \chi_{qj} &= \left\|\mathbf{A}_q \mathbf{e}_{qj}\right\|_2^2, \qquad w_{qj} = (\chi_{qj} + \varepsilon_s)^\beta, \qquad 0 \le \beta \le 1, \\ \varepsilon_s &= \eta_s \max_{q,j} \chi_{qj}, \qquad \eta_s = 10^{-8}. \end{aligned} \tag{8}$$

Perturbing local copy (q, j) by a scalar δ changes the local matrix-vector product $\mathbf{A}_q\mathbf{z}_q$ by $\delta\mathbf{A}_q\mathbf{e}_{qj}$, whose squared norm is $\chi_{qj}|\delta|^2$. Thus χ_{qj} is the squared norm of the change caused by a unit perturbation of that copy. It is also the corresponding diagonal entry of $\mathbf{A}_q^*\mathbf{A}_q$. For the row-scaled system,

$$\chi_{qj} = \sum_{i \in \mathcal{J}_q} \theta_i \frac{|a_{ij}^{\mathrm{FD}}|^2}{\left\|\mathbf{a}_i^{\mathrm{FD}}\right\|_2^2}, \tag{9}$$

where a_{ij}^{FD} is the coefficient of variable j in row i before scaling. The variable sets retain all stored column indices, including those with zero coefficients. A local column can therefore have zero squared norm; the positive shift assigns its copy a positive weight. The exponent controls the relative sizes of the weights: $\beta = 0$ gives equal weights, $\beta = 1$ uses the shifted squared column norms, and intermediate values reduce the ratios between larger and smaller weights. Because equation 9 sums over the rows assigned to each subdomain, copies of the same velocity variable can have different squared column norms and hence different weights. We compute these weights once and keep them fixed throughout the iterations.

To connect these squared column norms to averaging, let $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_{N_{\mathrm{sub}}})$ collect the local vectors after a sweep, with $z_{qj} = \mathbf{e}_{qj}^* \mathbf{z}_q$. Assigning a common value to each variable replaces $\mathbf{z}_q$ by $\mathbf{C}_q\mathbf{x}$ and changes $\mathbf{A}_q\mathbf{z}_q$ by $\mathbf{A}_q(\mathbf{C}_q\mathbf{x} - \mathbf{z}_q)$. The total squared change is $\sum_q \left\|\mathbf{A}_q(\mathbf{C}_q\mathbf{x} - \mathbf{z}_q)\right\|_2^2$. We retain only the diagonal entries of each $\mathbf{A}_q^*\mathbf{A}_q$ and add $\varepsilon_s\mathbf{I}$. This gives a separable quadratic approximation, which can be minimized one variable at a time:

$$\mathcal{J}_{\mathrm{diag},\varepsilon}(\mathbf{x}; \mathbf{z}) = \sum_j \sum_{q \in \mathcal{I}_j} (\chi_{qj} + \varepsilon_s)|x_j - z_{qj}|^2. \tag{10}$$

For $\beta = 1$, minimizing this objective gives the weighted component average in equation 12. The diagonal approximation therefore motivates the squared column norms as averaging coefficients. We introduce β to control the ratios between weights. For $0 \le \beta < 1$, the average minimizes the same type of quadratic objective with coefficients $w_{qj} = (\chi_{qj} + \varepsilon_s)^\beta$. The iteration tests in Section 5.2.1 and the spectral tests in Section 5.3.2 assess the exponent values.

We collect the copy weights in a block-diagonal matrix, with $\mathbf{W}_q = \mathrm{diag}(w_{qj}) \succ 0$ on subdomain q. The matrices defining the weighted inner products on the superspace $\mathcal{X}$ and the original variable space $\mathbb{C}^n$ are

$$\mathbf{W} = \mathrm{diag}(\mathbf{W}_1, \dots, \mathbf{W}_{N_{\mathrm{sub}}}), \qquad \mathbf{M} = \mathbf{C}^*\mathbf{W}\mathbf{C}, \qquad M_{jj} = \sum_{q \in \mathcal{I}_j} w_{qj}. \tag{11}$$

The matrix $\mathbf{M}$ is diagonal and positive definite. The M-inner product is a weighted dot product on the original unknowns, with one weight M_{jj} per variable. All inner products below are conjugate-linear in their first argument.

On the local-copy space $\mathcal{X}$, we use $\langle \mathbf{z}, \mathbf{y} \rangle_W = \mathbf{z}^* \mathbf{W} \mathbf{y}$. On the original variable space $\mathbb{C}^n$, we use $\langle \mathbf{x}, \mathbf{u} \rangle_M = \mathbf{x}^* \mathbf{M} \mathbf{u}$. Duplication connects the two through $\langle \mathbf{C}\mathbf{x}, \mathbf{C}\mathbf{u} \rangle_W = \langle \mathbf{x}, \mathbf{u} \rangle_M$.

3.2. Weighted component averaging

After independent local sweeps, averaging assigns one common value to the copies of each shared variable. The map $\mathbf{G}_W = \mathbf{M}^{-1}\mathbf{C}^*\mathbf{W}$ computes this value and satisfies $\mathbf{G}_W\mathbf{C} = \mathbf{I}$. The corresponding projection and averaged values are

$$\begin{aligned} \mathbf{E}_W &= \mathbf{C}\mathbf{G}_W = \mathbf{C}(\mathbf{C}^*\mathbf{W}\mathbf{C})^{-1}\mathbf{C}^*\mathbf{W}, \\ \mathbf{E}_W\mathbf{z} &= \mathbf{C}\mathbf{x}^{(W)}, \qquad x_j^{(W)} = \frac{\sum_{q \in \mathcal{I}_j} w_{qj} z_{qj}}{\sum_{q \in \mathcal{I}_j} w_{qj}}. \end{aligned} \tag{12}$$

For each variable, equation 12 assigns one value to all of its copies and minimizes $\sum_{q \in \mathcal{I}_j} w_{qj} |x_j - z_{qj}|^2$. A larger copy weight gives its local value more influence in this average. Thus $\mathbf{G}_W$ returns one averaged value per original variable, and $\mathbf{E}_W$ distributes those values back to all local copies. The row projections below use the same fixed metric.

At $\beta = 0$, $\mathbf{W} = \mathbf{I}$ and $\mathbf{M} = \mathbf{M}_0$, and the row and averaging maps reduce to their classical forms.

Proposition 1 (W-orthogonal projection onto the consensus subspace)**.** The map $\mathbf{E}_W$ is the W-orthogonal projector onto the consensus subspace $\mathcal{S} = \mathrm{range}(\mathbf{C})$. Equivalently,

$$\mathbf{E}_W\mathbf{z} = \arg\min_{\mathbf{y} \in \mathcal{S}} \|\mathbf{z} - \mathbf{y}\|_W^2, \qquad \|\mathbf{z}\|_W^2 = \mathbf{z}^*\mathbf{W}\mathbf{z}. \tag{13}$$

Proof. Writing $\mathbf{y} = \mathbf{C}\mathbf{x}$, the minimization in equation 13 gives the normal equations $\mathbf{M}\mathbf{x} = \mathbf{C}^*\mathbf{W}\mathbf{z}$. Since $\mathbf{M} \succ 0$, the unique minimizer is $\mathbf{x} = \mathbf{G}_W\mathbf{z}$. Moreover,

$$\mathbf{E}_W^2 = \mathbf{E}_W, \qquad \mathbf{E}_W\mathbf{C} = \mathbf{C}, \qquad \mathbf{E}_W^*\mathbf{W} = \mathbf{W}\mathbf{E}_W. \tag{14}$$

Thus $\mathbf{E}_W$ is idempotent, has range $\mathcal{S}$, and is self-adjoint in the W-inner product. These are the defining properties of the W-orthogonal projector onto $\mathcal{S}$. □

The averaging map $\mathbf{G}_W : \mathcal{X} \to \mathbb{C}^n$ is the adjoint of the duplication map $\mathbf{C} : \mathbb{C}^n \to \mathcal{X}$ when the copy space uses the W-inner product and the original-variable space uses the M-inner product. Thus $\langle \mathbf{C}\mathbf{x}, \mathbf{z} \rangle_W = \langle \mathbf{x}, \mathbf{G}_W\mathbf{z} \rangle_M$.

3.3. Weighted row projections

A row update changes the local variables to reduce the row residual. With $\theta_\iota = 1$, the weighted projection satisfies the row equation with the correction of smallest squared W-norm. The correction direction therefore depends on both the row coefficients and the squared column norms of the local matrix. We extend $\boldsymbol{\ell}_\iota = \mathbf{A}_q[r, :]^*$ to the space of all local copies by adding zeros outside its subdomain, and denote the result by $\widetilde{\boldsymbol{\ell}}_\iota \in \mathcal{X}$. Relaxing the projection onto $\widetilde{\boldsymbol{\ell}}_\iota^* \mathbf{y} = b_\iota$ gives

$$\begin{aligned} \mathcal{K}_{\iota,W}(\mathbf{z}) &= \mathbf{R}_{\iota,W}\mathbf{z} + \mathbf{c}_{\iota,W} = \mathbf{z} + \theta_\iota \frac{b_\iota - \widetilde{\boldsymbol{\ell}}_\iota^*\mathbf{z}}{\widetilde{\boldsymbol{\ell}}_\iota^*\mathbf{W}^{-1}\widetilde{\boldsymbol{\ell}}_\iota}\mathbf{W}^{-1}\widetilde{\boldsymbol{\ell}}_\iota, \\ \mathbf{R}_{\iota,W} &= \mathbf{I} - \theta_\iota \frac{\mathbf{W}^{-1}\widetilde{\boldsymbol{\ell}}_\iota\widetilde{\boldsymbol{\ell}}_\iota^*}{\widetilde{\boldsymbol{\ell}}_\iota^*\mathbf{W}^{-1}\widetilde{\boldsymbol{\ell}}_\iota}, \qquad 0 < \theta_\iota < 2, \\ \mathbf{c}_{\iota,W} &= \theta_\iota \frac{b_\iota \mathbf{W}^{-1}\widetilde{\boldsymbol{\ell}}_\iota}{\widetilde{\boldsymbol{\ell}}_\iota^*\mathbf{W}^{-1}\widetilde{\boldsymbol{\ell}}_\iota}. \end{aligned} \tag{15}$$

The inverse weights in $\mathbf{W}^{-1}\widetilde{\boldsymbol{\ell}}_\iota$ rescale the components of the row-correction direction. The denominator normalizes that direction, so the weights redistribute the correction among the variables in the row. Here $\mathbf{R}_{\iota,W}$ is the linear part

obtained with a zero right-hand side, and $\mathbf{c}_{\iota,W}$ is the constant term contributed by b_ι. When $\theta_\iota = 1$, $\mathbf{R}_{\iota,W}$ is the W-orthogonal projector onto $\ker(\widetilde{\boldsymbol{\ell}}_\iota^*)$. For $\mathbf{W} = \mathbf{I}$, equation 15 reduces to equation 6.

For $\beta > 0$, a copy with a larger squared column norm has a larger weight in the squared W-norm of a correction. To see how a row update affects another equation, let $r_\iota = b_\iota - \widetilde{\boldsymbol{\ell}}_\iota^* \mathbf{z}$ and let r_κ be the residual of another row in the same subdomain. Updating row ι changes that residual to

$$r_\kappa^+ = r_\kappa - \theta_\iota r_\iota \frac{\widetilde{\boldsymbol{\ell}}_\kappa^* \mathbf{W}^{-1} \widetilde{\boldsymbol{\ell}}_\iota}{\widetilde{\boldsymbol{\ell}}_\iota^* \mathbf{W}^{-1} \widetilde{\boldsymbol{\ell}}_\iota}. \tag{16}$$

Equation 16 shows how updating one row changes the residual of another row. By redistributing the correction, weighting changes the effect of successive row updates throughout the sweep, including updates to variables with only one subdomain copy. The same copy weight has two linked roles: its value sets the copy's influence in component averaging, while its reciprocal enters the row-correction direction. Both operations use the same W-norm of changes to local copies. Duplication then induces $\mathbf{M} = \mathbf{C}^* \mathbf{W} \mathbf{C}$ for CG on the original variables.

The change of variables $\widehat{\mathbf{z}} = \mathbf{W}^{1/2}\mathbf{z}$ turns the weighted row maps into relaxed Euclidean Kaczmarz maps for $\mathbf{A}_q \mathbf{W}_q^{-1/2}$. This local column rescaling is related to matrix equilibration [25, 26]. The same change of variables also turns $\mathbf{E}_W$ into the Euclidean orthogonal projector onto $\mathbf{W}^{1/2}\,\mathrm{range}(\mathbf{C})$.

The transformed local column norm is

$$\left\| \mathbf{A}_q \mathbf{W}_q^{-1/2} \mathbf{e}_{qj} \right\|_2 = \frac{\chi_{qj}^{1/2}}{(\chi_{qj} + \varepsilon_s)^{\beta/2}}. \tag{17}$$

For $\chi_{qj} \gg \varepsilon_s$, this norm is approximately $\chi_{qj}^{(1-\beta)/2}$. Consider any two local columns whose squared norms both greatly exceed ε_s. If the ratio of their original column norms is $R \geq 1$, the ratio after rescaling is approximately $R^{1-\beta}$, or $\sqrt{R}$ at $\beta = 0.5$. In this regime, weighting reduces differences between local column norms and changes the correction directions in equation 15. At $\beta = 1$, the transformed local column norms approach equality in this regime; $\beta = 0.5$ only partly reduces their differences. Both choices also change averaging and the induced CG inner product. The complete double sweep therefore depends on the row directions, row order, and coupling through shared variables, as well as on the local column norms. The spectral tests in Section 5.3.2 assess how these changes affect the matrix used by CG.

Remark 1 (Local and original-variable scaling)**.** The local copy weights need not define a single column scaling on the original variables. A positive diagonal matrix $\mathbf{D}$ on the original variables can satisfy $\mathbf{W}^{1/2}\mathbf{C} = \mathbf{C}\mathbf{D}^{1/2}$ if and only if all copies of each variable have equal weights. In that case, D_{jj} is their common weight.

Proposition 2 (W-self-adjointness and nonexpansiveness)**.** The operator $\mathbf{R}_{\iota,W}$ is W-self-adjoint. For any difference $\delta\mathbf{z}$ between two local-copy vectors in $\mathcal{X}$,

$$\left\| \mathbf{R}_{\iota,W} \delta\mathbf{z} \right\|_W^2 = \|\delta\mathbf{z}\|_W^2 - \theta_\iota(2 - \theta_\iota) \frac{|\widetilde{\boldsymbol{\ell}}_\iota^* \delta\mathbf{z}|^2}{\widetilde{\boldsymbol{\ell}}_\iota^* \mathbf{W}^{-1} \widetilde{\boldsymbol{\ell}}_\iota}. \tag{18}$$

Thus a row update does not increase the W-norm of a perturbation; it is nonexpansive in this norm.

Proof. Define the rank-one operator

$$\boldsymbol{\Pi}_{\iota,W} = \frac{\mathbf{W}^{-1} \widetilde{\boldsymbol{\ell}}_\iota \widetilde{\boldsymbol{\ell}}_\iota^*}{\widetilde{\boldsymbol{\ell}}_\iota^* \mathbf{W}^{-1} \widetilde{\boldsymbol{\ell}}_\iota}. \tag{19}$$

It satisfies $\mathbf{\Pi}_{\iota,W}^2 = \mathbf{\Pi}_{\iota,W}$ and $\mathbf{\Pi}_{\iota,W}^* \mathbf{W} = \mathbf{W}\mathbf{\Pi}_{\iota,W}$. Therefore $\mathbf{R}_{\iota,W} = \mathbf{I} - \theta_\iota \mathbf{\Pi}_{\iota,W}$ is W-self-adjoint, and

$$\begin{aligned} \left\|\mathbf{R}_{\iota,W}\delta\mathbf{z}\right\|_W^2 &= \|\delta\mathbf{z}\|_W^2 - \theta_\iota(2-\theta_\iota)\left\|\mathbf{\Pi}_{\iota,W}\delta\mathbf{z}\right\|_W^2, \\ \left\|\mathbf{\Pi}_{\iota,W}\delta\mathbf{z}\right\|_W^2 &= \frac{|\widetilde{\boldsymbol{\mathscr{C}}}_\iota^*\delta\mathbf{z}|^2}{\widetilde{\boldsymbol{\mathscr{C}}}_\iota^*\mathbf{W}^{-1}\widetilde{\boldsymbol{\mathscr{C}}}_\iota}. \end{aligned} \tag{20}$$

This proves equation 18. Since $0 < \theta_\iota < 2$, the norm is unchanged precisely when $\widetilde{\boldsymbol{\mathscr{C}}}_\iota^*\delta\mathbf{z} = 0$. □

3.4. Double sweep and CG iteration

We now combine the row maps and component averaging to form the matrix $\mathbf{H}_W$ used by CG. A weighted CARP double sweep consists of a forward row sweep, weighted averaging, a backward row sweep, and a final weighted average. The backward sweep uses the same row maps and relaxations in reverse order. With a zero right-hand side, the backward-sweep matrix is therefore the W-adjoint of the forward-sweep matrix. A weighted adjoint plays the role of a conjugate transpose in the chosen inner product. For $\mathbf{T} : \mathcal{X} \to \mathcal{X}$ and $\mathbf{L} : \mathbb{C}^n \to \mathbb{C}^n$, the weighted adjoints are $\mathbf{T}^{\star_W} = \mathbf{W}^{-1}\mathbf{T}^*\mathbf{W}$ and $\mathbf{L}^{\star_M} = \mathbf{M}^{-1}\mathbf{L}^*\mathbf{M}$. With a zero right-hand side, each sweep is block diagonal on $\mathcal{X}$, with one ordered product of row maps per subdomain. Including the right-hand side gives the affine sweeps

$$\mathcal{T}_f^{(W)}(\mathbf{z}) = \mathbf{T}_f^{(W)}\mathbf{z} + \mathbf{c}_f^{(W)}, \qquad \mathcal{T}_b^{(W)}(\mathbf{z}) = \mathbf{T}_b^{(W)}\mathbf{z} + \mathbf{c}_b^{(W)}, \tag{21}$$

where $\mathbf{T}_b^{(W)} = (\mathbf{T}_f^{(W)})^{\star_W}$. Here the matrices are the linear parts of the sweeps, and the vectors $\mathbf{c}_f^{(W)}$ and $\mathbf{c}_b^{(W)}$ are their constant terms, determined by $\mathbf{b}$.

Starting from $\mathbf{x}$, the composite map is

$$\begin{aligned} \mathcal{D}_W(\mathbf{x};\mathbf{b}) &= \mathbf{G}_W\mathcal{T}_b^{(W)}\Big(\mathbf{E}_W\mathcal{T}_f^{(W)}(\mathbf{C}\mathbf{x})\Big) = \mathbf{Q}_W\mathbf{x} + \mathbf{c}_W, \\ \mathbf{c}_W &= \mathcal{D}_W(\mathbf{0};\mathbf{b}), \qquad \mathbf{Q}_W = \mathbf{G}_W\mathbf{T}_b^{(W)}\mathbf{E}_W\mathbf{T}_f^{(W)}\mathbf{C}, \\ \mathbf{H}_W &= \mathbf{I} - \mathbf{Q}_W, \qquad \mathbf{H}_W\mathbf{x} = \mathbf{c}_W. \end{aligned} \tag{22}$$

If $\mathbf{A}\mathbf{x}_\star = \mathbf{b}$, the row maps fix $\mathbf{C}\mathbf{x}_\star$ and therefore $\mathcal{D}_W(\mathbf{x}_\star;\mathbf{b}) = \mathbf{x}_\star$ and $\mathbf{H}_W\mathbf{x}_\star = \mathbf{c}_W$.

Algorithm 1 computes products with $\mathbf{H}_W$ through row sweeps, without assembling $\mathbf{H}_W$ or $\mathbf{A}^*\mathbf{A}$. The physical source enters the affine term $\mathbf{c}_W$ and the initial CG residual. To apply $\mathbf{H}_W$ to a search direction, only the linear part $\mathbf{Q}_W$ is needed: $\mathcal{D}_W(\mathbf{p};\mathbf{0}) = \mathbf{Q}_W\mathbf{p}$. Sweeps applied to search directions therefore use a zero right-hand side. The matrix, row order, relaxation parameters, and metric remain fixed during each solve. CG updates the CG residual $\mathbf{r}_k^{\mathrm{CG}} = \mathbf{c}_W - \mathbf{H}_W\mathbf{x}_k$; stopping uses the row-scaled system residual $\mathcal{R}_k$ from equation 32.

Algorithm 1 Weighted CARP-CG

Require: A consistent system $\mathbf{A}\mathbf{x} = \mathbf{b}$ with $\mathbf{b} \neq \mathbf{0}$, initial vector $\mathbf{x}_0$, a fixed row partition, fixed positive weights, real relaxations in $(0, 2)$, complete forward and reverse-order backward sweeps, and tolerance ε.

Setup: Construct local systems and index sets for shared variables; compute χ_{qj}, w_{qj}, and weighted row denominators; store the metric data needed to apply $\mathbf{M}$, $\mathbf{G}_W$, and $\mathbf{E}_W$ implicitly.

$\mathbf{r}_0^{\mathrm{CG}} \leftarrow \mathcal{D}_W(\mathbf{x}_0;\mathbf{b}) - \mathbf{x}_0$, $\gamma_0 \leftarrow \left\langle \mathbf{r}_0^{\mathrm{CG}}, \mathbf{r}_0^{\mathrm{CG}} \right\rangle_M$; if $\gamma_0 = 0$, return $\mathbf{x}_0$.
$\mathbf{p}_0 \leftarrow \mathbf{r}_0^{\mathrm{CG}}$.
for $k = 0, 1, \ldots$ **do**
  $\mathbf{u}_k \leftarrow \mathbf{H}_W\mathbf{p}_k = \mathbf{p}_k - \mathcal{D}_W(\mathbf{p}_k;\mathbf{0})$; $\delta_k \leftarrow \langle \mathbf{p}_k, \mathbf{u}_k \rangle_M$; $\alpha_k \leftarrow \gamma_k/\delta_k$.
  $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k\mathbf{p}_k$; $\mathbf{r}_{k+1}^{\mathrm{CG}} \leftarrow \mathbf{r}_k^{\mathrm{CG}} - \alpha_k\mathbf{u}_k$; $\gamma_{k+1} \leftarrow \left\langle \mathbf{r}_{k+1}^{\mathrm{CG}}, \mathbf{r}_{k+1}^{\mathrm{CG}} \right\rangle_M$.
  At a prescribed check iteration, evaluate $\mathcal{R}(\mathbf{x}_{k+1})$; if $\mathcal{R}(\mathbf{x}_{k+1}) \leq \varepsilon$, return $\mathbf{x}_{k+1}$.
  If $\gamma_{k+1} = 0$, return $\mathbf{x}_{k+1}$.
  $\zeta_k \leftarrow \gamma_{k+1}/\gamma_k$; $\mathbf{p}_{k+1} \leftarrow \mathbf{r}_{k+1}^{\mathrm{CG}} + \zeta_k\mathbf{p}_k$.
end for

3.5. Factorization of the CG operator

We factor the CG operator in rescaled coordinates to establish self-adjointness, positive semidefiniteness, and eigenvalue bounds. To express the two spaces in Euclidean coordinates, rescale original vectors by $\mathbf{M}^{1/2}$ and local-copy vectors by $\mathbf{W}^{1/2}$. The following maps represent duplication, the forward sweep, and averaging in these coordinates:

$$\begin{aligned} \mathbf{U}_W &= \mathbf{W}^{1/2}\mathbf{C}\mathbf{M}^{-1/2}, \qquad \widehat{\mathbf{T}}_W = \mathbf{W}^{1/2}\mathbf{T}_f^{(W)}\mathbf{W}^{-1/2}, \\ \widehat{\mathbf{E}}_W &= \mathbf{W}^{1/2}\mathbf{E}_W\mathbf{W}^{-1/2}, \quad \mathbf{B}_W = \widehat{\mathbf{E}}_W\widehat{\mathbf{T}}_W\mathbf{U}_W. \end{aligned} \tag{23}$$

In these rescaled coordinates, $\mathbf{U}_W : \mathbb{C}^n \to \mathcal{X}$ represents duplication and satisfies $\mathbf{U}_W^*\mathbf{U}_W = \mathbf{I}$, so $\mathbf{U}_W$ preserves Euclidean lengths and inner products. The map $\widehat{\mathbf{T}}_W$ represents the forward sweep, and $\widehat{\mathbf{E}}_W$ is the orthogonal projector onto the transformed consensus subspace. The composed map $\mathbf{B}_W : \mathbb{C}^n \to \mathcal{X}$ duplicates the original-variable vector, applies the forward sweep, and projects the result onto the transformed consensus subspace.

Theorem 1 (M-self-adjointness and positive semidefiniteness)**.** Assume that $\mathbf{C}$ has full column rank, $\mathbf{W} \succ 0$ is a fixed positive diagonal matrix on $\mathcal{X}$, all embedded rows are nonzero, and all relaxations are real and lie in $(0, 2)$. Assume that the backward sweep applies the same row maps and relaxations in exact reverse order. Then

$$\begin{aligned} &\mathbf{Q}_W^{\star_M} = \mathbf{Q}_W, \qquad \mathbf{H}_W^{\star_M} = \mathbf{H}_W, \\ &\mathbf{M}^{1/2}\mathbf{Q}_W\mathbf{M}^{-1/2} = \mathbf{B}_W^*\mathbf{B}_W, \\ &\widehat{\mathbf{H}}_W := \mathbf{M}^{1/2}\mathbf{H}_W\mathbf{M}^{-1/2} = \mathbf{I} - \mathbf{B}_W^*\mathbf{B}_W \succeq 0, \end{aligned} \tag{24}$$

where $\|\mathbf{B}_W\|_2 \le 1$ and $0 \preceq \widehat{\mathbf{H}}_W \preceq \mathbf{I}$. Whenever $\mathbf{A}\mathbf{x} = \mathbf{b}$ is consistent, $\mathbf{H}_W\mathbf{x} = \mathbf{c}_W$ is consistent.

Proof. Equation 11 gives $\mathbf{U}_W^*\mathbf{U}_W = \mathbf{I}$. Proposition 1 implies $\widehat{\mathbf{E}}_W^* = \widehat{\mathbf{E}}_W = \widehat{\mathbf{E}}_W^2$, and the reverse sweep satisfies $\mathbf{W}\mathbf{T}_b^{(W)} = (\mathbf{T}_f^{(W)})^*\mathbf{W}$. Substitution into equation 22 yields

$$\begin{aligned} \mathbf{M}^{1/2}\mathbf{Q}_W\mathbf{M}^{-1/2} &= \mathbf{M}^{-1/2}\mathbf{C}^*\mathbf{W}\mathbf{T}_b^{(W)}\mathbf{E}_W\mathbf{T}_f^{(W)}\mathbf{C}\mathbf{M}^{-1/2} \\ &= \mathbf{U}_W^*\widehat{\mathbf{T}}_W^*\widehat{\mathbf{E}}_W\widehat{\mathbf{T}}_W\mathbf{U}_W = \mathbf{B}_W^*\mathbf{B}_W. \end{aligned} \tag{25}$$

By Proposition 2, $\|\widehat{\mathbf{T}}_W\|_2 \le 1$. Together with $\|\widehat{\mathbf{E}}_W\|_2 \le 1$ and $\|\mathbf{U}_W\|_2 = 1$, this gives $\|\mathbf{B}_W\|_2 \le 1$ and $0 \preceq \widehat{\mathbf{H}}_W \preceq \mathbf{I}$. Transforming back gives the stated M-self-adjointness. If $\mathbf{A}\mathbf{x}_\star = \mathbf{b}$, the affine row maps and averaging fix $\mathbf{C}\mathbf{x}_\star$, so $\mathbf{c}_W = \mathbf{H}_W\mathbf{x}_\star$. □

The similarity transform in equation 24 preserves the eigenvalues of $\mathbf{H}_W$. The factorization gives these eigenvalues as $1 - \sigma_j(\mathbf{B}_W)^2$, where $\sigma_j(\mathbf{B}_W)$ denotes a singular value of $\mathbf{B}_W$. Before CG acceleration, the double sweep in the transformed coordinates with a zero right-hand side multiplies the error component along each right singular vector of $\mathbf{B}_W$ by $\sigma_j(\mathbf{B}_W)^2$. A component with $\sigma_j(\mathbf{B}_W)$ close to one is therefore reduced only weakly.

3.6. A quadratic-form identity and solution-set equivalence

Every solution of $\mathbf{A}\mathbf{x} = \mathbf{b}$ is already a fixed point of the double sweep. We now show that the fixed-point system has no additional solutions when the forward sweep visits every local row. To establish this equivalence, we track the norm decrease through the forward sweep and component averaging. For the analysis, list the m row applications across all subdomains as $\iota_1, \dots, \iota_m$ in an order that preserves each local forward sweep. The subdomains still perform their sweeps independently. For an arbitrary test vector $\mathbf{u} \in \mathbb{C}^n$, set $\mathbf{z}_0 = \mathbf{C}\mathbf{u}$ and $\mathbf{z}_j = \mathbf{R}_{\iota_j,W}\mathbf{z}_{j-1}$, so $\mathbf{z}_m = \mathbf{T}_f^{(W)}\mathbf{C}\mathbf{u}$. The factorization and the W-orthogonality of $\mathbf{E}_W$ give

$$\begin{aligned} \langle \mathbf{u}, \mathbf{H}_W\mathbf{u}\rangle_M &= \|\mathbf{z}_0\|_W^2 - \|\mathbf{E}_W\mathbf{z}_m\|_W^2 \\ &= \sum_{j=1}^{m}\left(\|\mathbf{z}_{j-1}\|_W^2 - \|\mathbf{z}_j\|_W^2\right) + \|(\mathbf{I} - \mathbf{E}_W)\mathbf{z}_m\|_W^2 . \end{aligned} \tag{26}$$

Applying equation 18 to each term yields

$$\langle \mathbf{u}, \mathbf{H}_W \mathbf{u} \rangle_M = \sum_{j=1}^{m} \theta_{\iota_j}(2-\theta_{\iota_j}) \frac{|\widetilde{\boldsymbol{\mathscr{C}}}_{\iota_j}^* \mathbf{z}_{j-1}|^2}{\widetilde{\boldsymbol{\mathscr{C}}}_{\iota_j}^* \mathbf{W}^{-1} \widetilde{\boldsymbol{\mathscr{C}}}_{\iota_j}} + \left\| (\mathbf{I} - \mathbf{E}_W)\mathbf{z}_m \right\|_W^2 . \tag{27}$$

The quadratic form $\langle \mathbf{u}, \mathbf{H}_W \mathbf{u} \rangle_M = \mathbf{u}^* \mathbf{M} \mathbf{H}_W \mathbf{u}$ is thus a sum of nonnegative terms. The sum over rows gives the decrease in the squared W-norm during the forward sweep. The final term is the squared W-distance from the resulting local-copy vector to the consensus subspace.

Proposition 3 (Solution-set equivalence)**.** Under the assumptions of Theorem 1, suppose that the forward sweep visits every row of every $\mathbf{A}_q$. Then

$$\ker(\mathbf{H}_W) = \ker(\mathbf{A}). \tag{28}$$

For a consistent system, $\mathbf{H}_W \mathbf{x} = \mathbf{c}_W$ and $\mathbf{A}\mathbf{x} = \mathbf{b}$ have identical solution sets. In particular, $\mathbf{H}_W$ is positive definite in the M-inner product when $\mathbf{A}$ has full column rank.

Proof. If $\mathbf{A}\mathbf{u} = 0$, every homogeneous row map leaves $\mathbf{C}\mathbf{u}$ unchanged, as does component averaging. Thus $\mathbf{H}_W \mathbf{u} = 0$.

Conversely, let $\mathbf{H}_W \mathbf{u} = 0$. Each nonnegative term in equation 27 must vanish. Since $0 < \theta_{\iota_j} < 2$, this implies $\widetilde{\boldsymbol{\mathscr{C}}}_{\iota_j}^* \mathbf{z}_{j-1} = 0$ at every row visit. The first row map therefore leaves $\mathbf{z}_0 = \mathbf{C}\mathbf{u}$ unchanged. Induction gives $\mathbf{z}_j = \mathbf{C}\mathbf{u}$ throughout the sweep, and hence $\widetilde{\boldsymbol{\mathscr{C}}}_{\iota_j}^* \mathbf{C}\mathbf{u} = 0$ for every visited row. Because every row is visited and $\mathbf{A}[\mathcal{J}_q, :] = \mathbf{A}_q \mathbf{C}_q$, we obtain $\mathbf{A}\mathbf{u} = 0$.

For any solution $\mathbf{x}_\star$ of $\mathbf{A}\mathbf{x} = \mathbf{b}$, Theorem 1 gives $\mathbf{c}_W = \mathbf{H}_W \mathbf{x}_\star$. The common kernel then implies

$$\mathbf{H}_W \mathbf{x} = \mathbf{c}_W \iff \mathbf{x} - \mathbf{x}_\star \in \ker(\mathbf{H}_W) \iff \mathbf{A}\mathbf{x} = \mathbf{b}. \tag{29}$$

Full column rank makes this kernel trivial, so positive semidefiniteness becomes positive definiteness. □

3.7. Conjugate-gradient convergence

CG solves $\mathbf{H}_W \mathbf{x} = \mathbf{c}_W$ in the M-inner product. The substitutions $\mathbf{y} = \mathbf{M}^{1/2}\mathbf{x}$ and $\widehat{\mathbf{c}}_W = \mathbf{M}^{1/2}\mathbf{c}_W$ turn it into ordinary Euclidean CG [27] for $\widehat{\mathbf{H}}_W \mathbf{y} = \widehat{\mathbf{c}}_W$. Proposition 3 gives $\operatorname{rank}(\widehat{\mathbf{H}}_W) = \operatorname{rank}(\mathbf{A})$. Under the stated assumptions, CG converges to the unique solution when $\mathbf{A}$ has full column rank. The following theorem applies standard CG theory [28] and also covers rank-deficient systems, where the initial nullspace component determines which solution is obtained.

Theorem 2 (CG convergence and finite termination)**.** Assume the hypotheses of Proposition 3 and that $\mathbf{A}\mathbf{x} = \mathbf{b}$ is consistent. In exact arithmetic, CG with initial vector $\mathbf{x}_0$ converges to the solution $\mathbf{x}_\star$ for which $\mathbf{M}^{1/2}\mathbf{x}_\star$ has the same orthogonal projection onto $\ker(\widehat{\mathbf{H}}_W)$ as $\mathbf{M}^{1/2}\mathbf{x}_0$.

When $\operatorname{rank}(\mathbf{A}) > 0$, let $\lambda_{\min}^+$ and $\lambda_{\max}$ be the smallest and largest positive eigenvalues of $\widehat{\mathbf{H}}_W$, and set $\kappa_+ = \lambda_{\max}/\lambda_{\min}^+$. For $\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}_\star$ and the seminorm defined by $\|\mathbf{e}\|_{H,M}^2 = \mathbf{e}^* \mathbf{M} \mathbf{H}_W \mathbf{e}$, the iterates satisfy

$$\left\| \mathbf{e}_k \right\|_{H,M} \le 2 \left(\frac{\sqrt{\kappa_+} - 1}{\sqrt{\kappa_+} + 1} \right)^k \left\| \mathbf{e}_0 \right\|_{H,M} , \qquad k \ge 1. \tag{30}$$

Termination occurs in at most $d \le \operatorname{rank}(\mathbf{A})$ steps, where d is the number of distinct positive eigenvalues of $\widehat{\mathbf{H}}_W$. If the initial residual is zero, the initial vector is already a solution.

Proof. For any solution $\mathbf{y}_\star$, the initial transformed residual is

$$\widehat{\mathbf{r}}_0 = \widehat{\mathbf{c}}_W - \widehat{\mathbf{H}}_W \mathbf{y}_0 = \widehat{\mathbf{H}}_W (\mathbf{y}_\star - \mathbf{y}_0) \in \operatorname{range}(\widehat{\mathbf{H}}_W). \tag{31}$$

The range of $\widehat{\mathbf{H}}_W$ is invariant under $\widehat{\mathbf{H}}_W$ and orthogonal to its kernel. All CG corrections lie in this range, where $\widehat{\mathbf{H}}_W$ is positive definite. Thus CG preserves the nullspace component of $\mathbf{y}_0$ and is well defined until convergence on the range.

Standard CG theory on this positive-definite range gives the Chebyshev bound in equation 30 and termination in at most the number of distinct positive eigenvalues [28]. The preserved nullspace component identifies the stated solution. □

When $\mathbf{A}$ has full column rank, $\lambda^{+}_{\min}$ is the smallest eigenvalue and the seminorm defined above is a norm.

3.8. Parallel implementation

During setup, a global maximum reduction computes the largest local squared column norm used in the positive shift. We then precompute the weights and weighted row denominators, and each subdomain stores the weights for its local copies. During each CG iteration, row corrections apply the reciprocal copy weights, and component averaging multiplies local values by their weights before combining shared copies. The outer CG vector operations use M_{jj} in their dot products. Precomputation therefore avoids rebuilding the weights but does not remove these weighted operations from the iteration. Component averaging exchanges weighted values between the subdomains sharing each variable and sums their contributions. CG dot products instead combine local scalar sums through global MPI reductions. The communication pattern and the number of stored vectors stay fixed throughout the solve.

4. Memory-efficient matrix storage

In conventional compressed sparse row (CSR) storage, every stored matrix entry has a coefficient and a column index. A separate row-pointer array identifies the entries belonging to each row. Keeping all coefficients and column indices in memory is costly for the large three-dimensional elastic systems. Both 3D solvers therefore use CSR-SER; the 2D tests retain explicit CSR. The storage strategy builds on CSR compression [19], stencil–CSR hybrids [20], and coefficient generation during CARP-CG sweeps [21].

CSR-SER obtains an entry's position and value in two separate steps. The finite-difference stencil specifies which neighboring velocity samples can appear in a row. The large three-dimensional stencil uses 97 distinct relative column offsets, so a one-byte code can identify each entry's offset. The offset table covers all these possible positions; an individual row uses only a subset. A small table on each process translates the code into the relative offset, from which the column index is recovered. The coefficient is computed from the stored material, frequency, PML, and discretization data when the row is needed. Figure 1(a) shows the stored arrays. The scalar example in Fig. 1(c) illustrates the position codes: the same codes select offsets $-1, 0, +1$ in successive rows, while the corresponding column numbers change with the row.

During a sweep, the solver reconstructs only the rows currently being used. At each grid index, it fills temporary coefficient and column-index arrays for the three velocity-component equations, each with at most 51 entries. It applies the three row updates in the prescribed order and writes the updates to the solution vector. The same temporary arrays are then filled with the rows for the next grid index, as shown in Fig. 1(b). The solution vector is stored separately and retains all updates. The backward sweep reverses both the grid traversal and the component order. Coefficients are computed again on each sweep, trading repeated calculation for lower storage.

A double-precision complex coefficient occupies 16 bytes and a column index occupies 4 bytes in this implementation. Conventional explicit CSR therefore uses 20 bytes per stored entry for these two arrays, whereas CSR-SER keeps a one-byte position code. For the same stored entries, the persistent position codes occupy 5% of the space required by the explicit CSR coefficient and column-index arrays. Model data, row pointers, index maps, temporary row arrays, and solver vectors and weights are counted separately. For the large case with PMLs on all boundaries and 3.74×10^9 stored entries, the coefficient and column-index arrays alone require 69.723 GiB in explicit CSR. The CSR-SER codes occupy 3.486 GiB.

With the same input data, generated coefficients, column indices, row order, and normalization, the reconstructed rows define the same algebraic sweeps as explicit CSR in exact arithmetic. Supplementary material, Section S4 gives the equivalence proof and exact storage counts, including the remaining arrays.

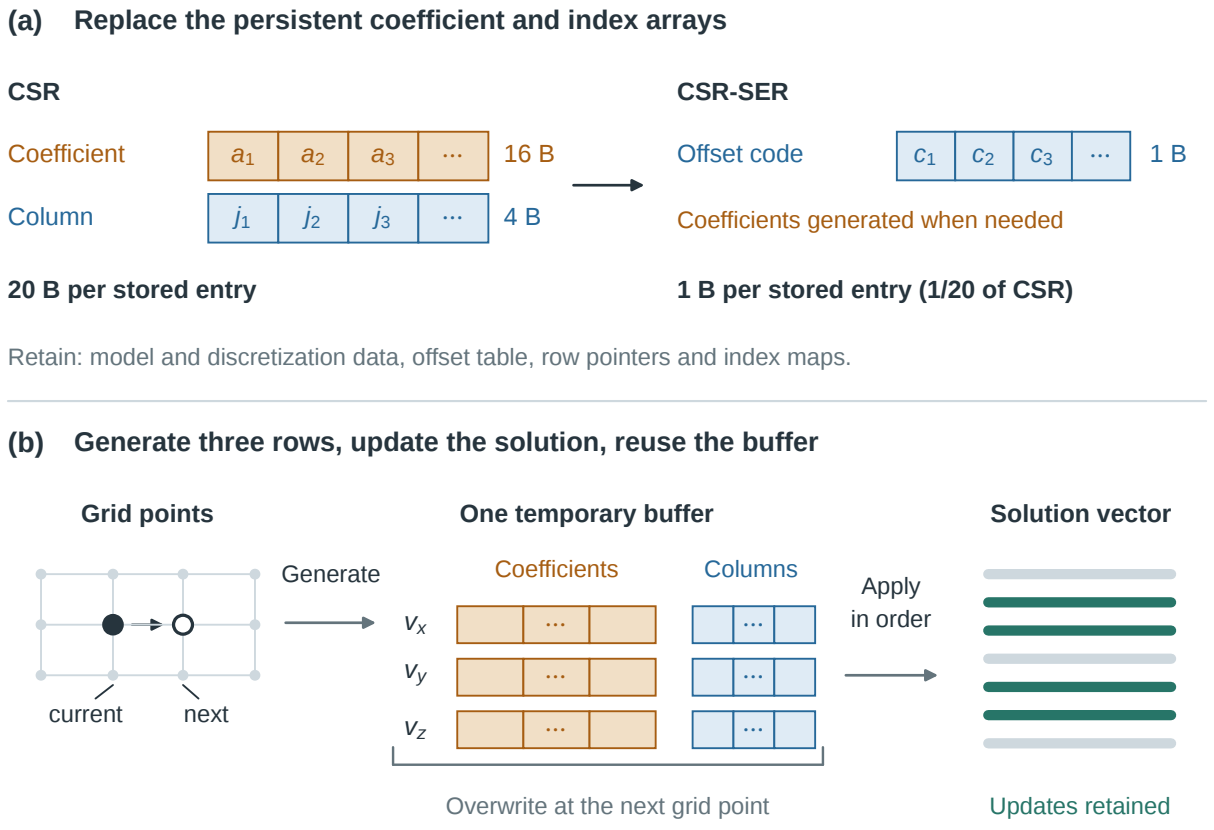


Figure 1: CSR-SER storage used by both 3D solvers. (a) Explicit CSR stores a coefficient a_k and column index j_k for each entry. CSR-SER stores a one-byte code c_k for its relative position and computes the coefficient when needed. (b) Temporary arrays hold the three velocity-component rows. The solver applies the three row updates in order and stores the resulting changes in the separate solution vector before reusing the temporary arrays at the next grid index. (c) A scalar second-difference example: the same codes select offsets $-1, 0, +1$ in each row, giving different column numbers as the row changes. Orange denotes coefficients, blue denotes indices or codes, and green denotes solution updates.

5. Numerical results and discussion

The experiments assess wavefield accuracy, iteration counts, and iteration-loop time. The two-dimensional tests vary the weighting exponent, source location, frequency, attenuation, surface boundary condition, and assignment of equation rows to parallel processes. Small three-dimensional systems allow us to compute every eigenvalue of the CG matrices. Larger tests vary frequency, grid spacing, Poisson's ratio, source depth, and surface boundary condition.

5.1. Comparison criteria and common settings

CARP-CG and Weighted CARP-CG use the same matrix, right-hand side, boundary condition, row partition and order, relaxation, and zero initial vector in each comparison. All solver comparisons use isotropic explosive sources, with positions specified for each study. Convergence histories and stopping decisions use the explicitly evaluated relative 2-norm residual of the row-scaled system:

$$\mathcal{R}(\mathbf{x}) = \frac{\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2}{\|\mathbf{b}\|_2}, \qquad \mathcal{R}_k = \mathcal{R}(\mathbf{x}_k), \qquad \mathbf{b} \neq \mathbf{0}. \tag{32}$$

Here $\mathbf{x}_k$ is the kth CG iterate. The residual is recomputed from $\mathbf{A}\mathbf{x}_k$, independently of the recursive CG residual. Unless stated otherwise, a run stops at the first scheduled check satisfying $\mathcal{R}_k \leq 10^{-4}$. In the tables, Weighted denotes Weighted CARP-CG.

The CARP-CG baseline uses the Euclidean inner product $\mathbf{x}^*\mathbf{u}$, whereas Weighted CARP-CG uses $\mathbf{x}^*\mathbf{M}\mathbf{u}$. Unless varied, the weights use $\beta = 0.5$ and $\eta_s = 10^{-8}$ in equation 8. The uniform row relaxation is 1.2 in two dimensions and 1.0 in three dimensions.

All computations use double precision on an Intel Core i9-13900K workstation (24 physical cores, 32 logical CPUs, 32 GiB memory). Parallel runs use single-threaded Message Passing Interface (MPI) processes, with process counts specified for each study. Iteration counts refer to outer CG updates. Iteration-loop time includes the sweeps, communication, CG vector operations, residual checks, and logging inside the loop; it excludes setup and final wavefield output. Timings are single-run measurements. Supplementary material, Sections S3.1 and S3.2 give additional iteration-loop times.

5.2. 2D Marmousi2 experiments

The two-dimensional study covers exponent selection, wavefield accuracy, iteration cost, and surface-boundary effects. Unless stated otherwise, these tests use 16 single-threaded MPI processes and evaluate the residual at every iteration.

5.2.1. Model and weight-exponent sensitivity

The experiments use a 2×2 km Marmousi2-derived elastic submodel with a spatially varying Poisson's ratio [29]. The reference grid has 101×101 physical points at $h = 20$ m. The default PML is 200 m thick on each absorbing side. The central source is at $(x, z) = (1000, 1000)$ m; the shallow source at $(460, 160)$ m lies in a near-surface low-V_S region outside the PML, and the deep source at $(900, 1260)$ m lies in a locally smooth interior region. Fig. 2 shows the elastic parameter models and source locations.

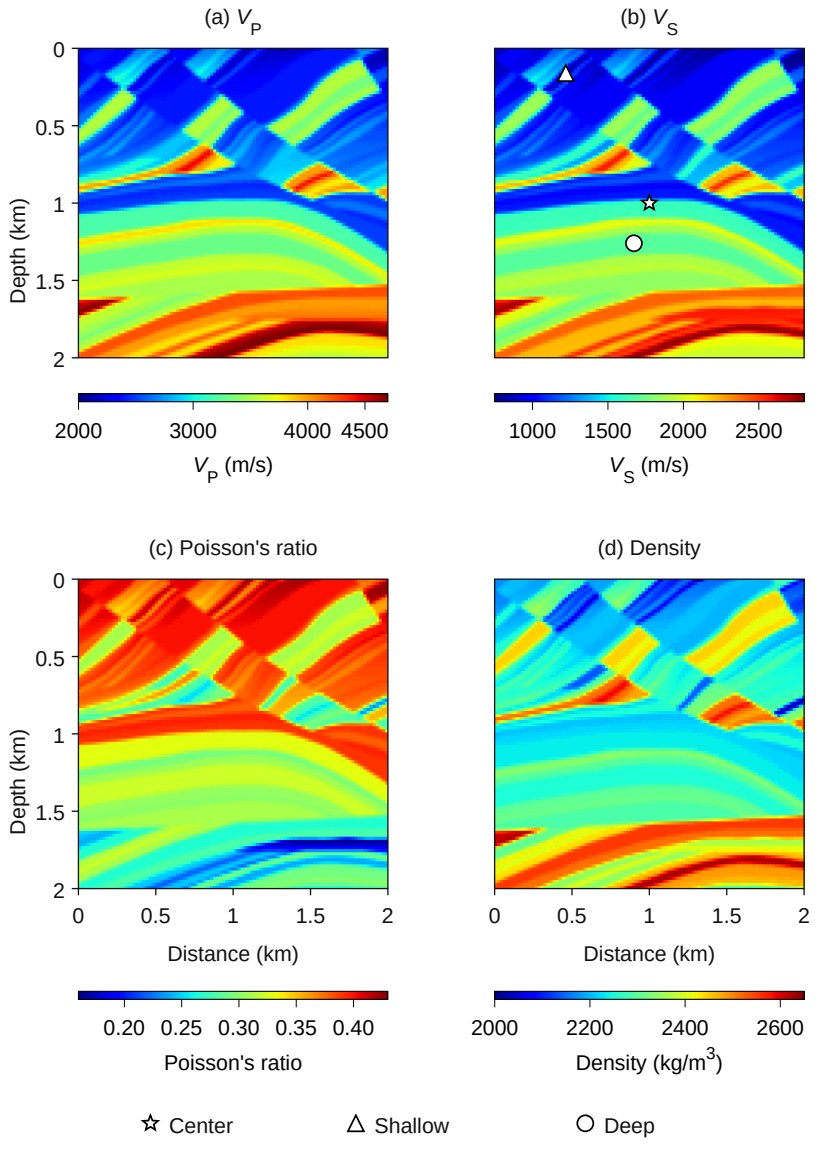


Figure 2: Elastic parameter models over the 2×2 km Marmousi2-derived domain: (a) V_P, (b) V_S, (c) Poisson's ratio, and (d) density. Each panel has its own color scale. In panel (b), the star, triangle, and circle mark the central source, the shallow source in the low-V_S region, and the deep source, respectively.

Spatial sampling is quantified by the minimum number of grid points per S-wave wavelength, $N_{\lambda,S} = V_{S,\min}/(fh)$, where $f = \omega/(2\pi)$, h is the grid spacing, and $V_{S,\min}$ is the minimum shear-wave speed in the physical model. We adopt $N_{\lambda,S} \geq 4$ as the dispersion-based sampling criterion for the optimal fourth-order staggered-grid scheme [1, 2].

To select β, we compare the iteration counts for different exponents in equation 8 on the same 5 Hz shallow-source PML system, using a common residual target. The original Euclidean CARP-CG recurrence provides the baseline for the complete weighted method. Within this study the matrix, source, row partition, and stopping criterion are fixed.

The test uses 16 row blocks, an attenuation quality factor of $Q_{\text{att}} = 1000$ (smaller values mean stronger attenuation), and the row-scaled residual target 10^{-4}. CARP-CG requires 4703 iterations. Table 1 reports the iteration counts for the five positive exponents and their reductions relative to this baseline.

Table 1
Exponent sensitivity on the 5 Hz shallow-source Marmousi2 PML system at $\mathcal{R} \leq 10^{-4}$.

β	Iterations	Reduction
0.10	3170	32.60%
0.25	2973	36.79%
0.50	2920	37.91%
0.75	3016	35.87%
1.00	3665	22.07%

On the PML system, $\beta = 0.5$ requires 2920 iterations, compared with 4703 for the baseline, a reduction of 37.9%. All five positive exponents converge, including $\beta = 1$ at 3665 iterations. The fewest iterations occur at $\beta = 0.5$ among these values. The stronger local column rescaling at $\beta = 1$ therefore does not give the lowest iteration count in this case. Section 5.3.2 examines the effect on the complete double-sweep spectra.

5.2.2. Algebraic accuracy, grid refinement, and PML sensitivity

We now switch from the shallow-source exponent study to the central-source system. Comparing the iterative solutions with a sparse direct solution measures algebraic solution error on this fixed grid. Direct-solution comparisons across grids and PML settings then assess the discretization used in the subsequent solver tests.

The central-source comparison uses 5 Hz, $h = 20$ m, $Q_{\mathrm{att}} = 1000$, and a 200 m PML. The direct solver is SuperLU, accessed through SciPy 1.13.1 with column approximate minimum degree (COLAMD) ordering [30–32]; its relative residual is below 5×10^{-15}. Let $\mathbf{v}$ collect both complex velocity components at their staggered coordinates within the physical domain and define $E_{\mathrm{alg}} = \left\|\mathbf{v}^{\mathrm{iter}} - \mathbf{v}^{\mathrm{dir}}\right\|_2 / \left\|\mathbf{v}^{\mathrm{dir}}\right\|_2$. At $\mathcal{R} \leq 10^{-4}$, CARP-CG and Weighted CARP-CG give errors of 3.429% and 3.350%, respectively, in 5651 and 3376 iterations. Thus Weighted CARP-CG reaches comparable combined-field accuracy with 40.26% fewer iterations. The relation between equation residual and field error also depends on the conditioning of the discrete system [28]. Fig. 3 illustrates the horizontal component and its difference from the direct reference. Supplementary material, Section S3.1 defines the field samples and single-component statistics.

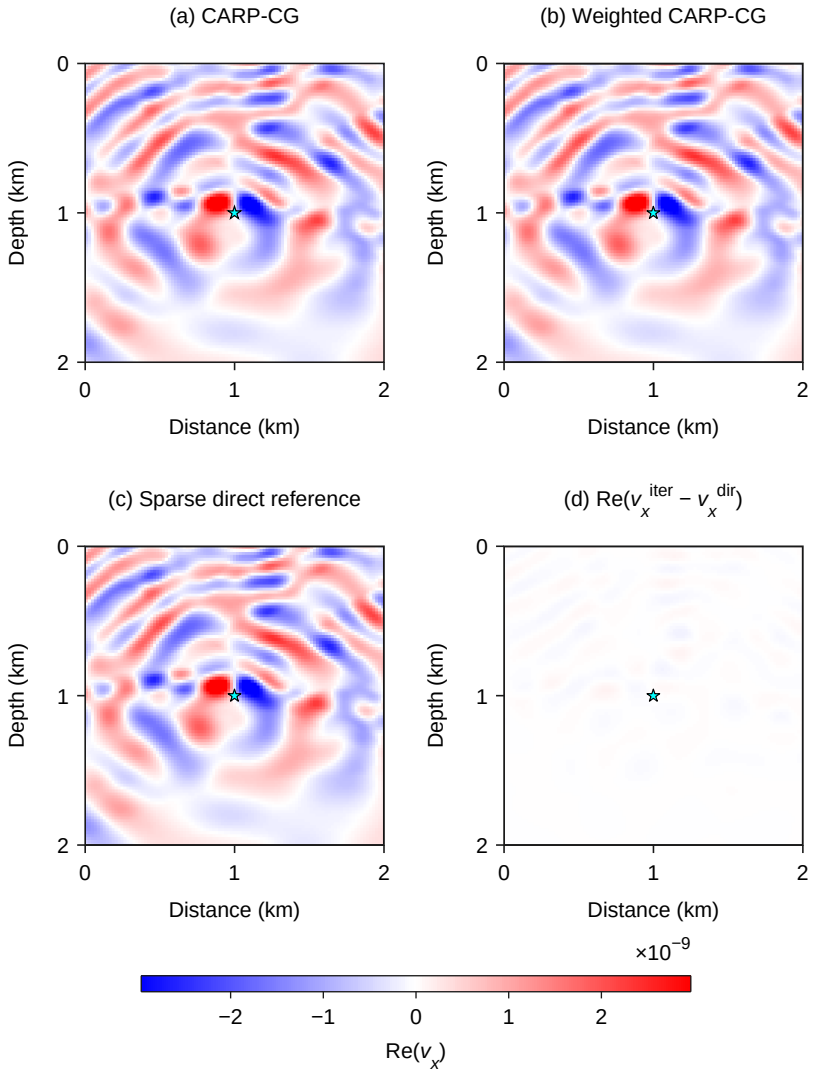


Figure 3: Real horizontal particle velocity for the same 5 Hz Marmousi2 finite-difference system with PMLs: (a) CARP-CG, (b) Weighted CARP-CG, and (c) the sparse direct reference. Panel (d) shows the signed difference $\mathrm{Re}(v_x^{\mathrm{iter}} - v_x^{\mathrm{dir}})$ between Weighted CARP-CG and the direct reference. The relative complex-field error for the horizontal-component samples shown is 3.69%. Both iterative solutions satisfy $\mathcal{R} \leq 10^{-4}$. The panels show the 101×101 horizontal-component subarrays after removal of ten PML index layers on each side; Supplementary material, Section S3.1 defines the sampling and associated correlation and error. Axes show grid coordinates before the $h/2$ stagger of v_x in x is added. The star marks the source. All panels use the source normalization in Supplementary material, Section S1.3 and the same symmetric amplitude scale.

To assess grid sensitivity, a separate direct-solution comparison at 8 Hz uses $h = 20$, 12.5, and 6.25 m while keeping the physical domain, source normalization, and 200 m PML thickness fixed. The $h = 6.25$ m solution is the finest-grid reference. Relative to this reference, the combined complex-velocity differences are 6.96% and 1.93% for the two coarser grids. This measures the change in the discrete wavefield under refinement. Supplementary material, Section S3.1 tabulates the grids and results and specifies the common receiver grid and interpolation procedure.

To assess PML sensitivity, we change the layer thickness or damping amplitude at 5 Hz and $h = 20$ m. We compare the fields with an expanded-domain numerical reference obtained by a sparse direct solve at the same grid spacing. Supplementary material, Section S3.1 describes the reference-domain construction and its extension-sensitivity check. Every comparison uses the original physical source and observation coordinates. Table 2 gives the relative differences from the reference, measured over the original physical domain.

Table 2
PML sensitivity at 5 Hz and $h = 20$ m, measured by relative 2-norm differences in the combined complex velocity components from the expanded-domain numerical reference.

PML thickness (m)	d_{PML} (s^{-1})	Relative field difference in the physical domain
200	100	0.438%
200	200	0.439%
200	400	5.047%
400	200	0.146%

Doubling PML thickness reduces the physical-domain difference from 0.439% to 0.146%, while doubling the damping amplitude increases it to 5.047%. Increasing thickness is thus more effective than increasing damping for this configuration.

5.2.3. Effects of row partitioning, source location, attenuation, and frequency

With the discretization fixed, we next examine iteration cost under changes in the row partition, source, attenuation, and frequency. The row partition determines which local equations contribute to each variable copy's squared column norm and weight. To examine how the row partition affects convergence, Table 3 compares 8, 16, and 32 contiguous row blocks on the same 5 Hz central-source PML system, with one block per MPI process. The shared-copy percentage counts local copies whose original variable appears in more than one subdomain, divided by the total number of local copies. The maximum number of copies of any original variable is $m_{\max} = \max_j |\mathcal{I}_j|$.

Table 3
Row-partition sensitivity for the 5 Hz Marmousi2 PML case with a central source at $\mathcal{R} \le 10^{-4}$.

			Iterations		
N_{sub}	Shared local copies (%)	$m_{\max}$	CARP-CG	Weighted	Reduction
8	51.53%	2	4625	3238	29.99%
16	85.31%	2	5651	3376	40.26%
32	99.49%	3	5710	3638	36.29%

Both methods require more iterations as the number of subdomains increases, and Weighted CARP-CG retains a lower count in every partition. Although the shared-copy fraction increases with the number of subdomains, the reduction in iteration count varies nonmonotonically across the tested partitions.

With $\beta = 0.5$ fixed, we next vary source position, attenuation, or frequency on the same $h = 20$ m grid. The base PML configuration uses the central source, 5 Hz, and $Q_{\mathrm{att}} = 1000$. The other PML tests change attenuation to $Q_{\mathrm{att}} = 50$, source position, or frequency. The 8 Hz case has 4.80 points per minimum S-wave wavelength, compared with 7.68 at 5 Hz. Section 5.2.2 assesses the wavefield's grid sensitivity at 8 Hz; the present comparison measures iteration cost on the fixed grid. Table 4 also includes the free-surface (FS) configuration examined in the next subsection.

Table 4
Iteration counts for the Marmousi2 systems on the 101×101 physical grid satisfying $N_{\lambda,\mathrm{S}} \geq 4$, at $\mathcal{R} \leq 10^{-4}$.

Case	$N_{\lambda,\mathrm{S}}$	CARP-CG	Weighted	Reduction
PML, center, 5 Hz	7.68	5651	3376	40.26%
PML, center, 5 Hz ($Q_{\mathrm{att}} = 50$)	7.68	4242	2727	35.71%
PML, shallow low-V_{S}, 5 Hz	7.68	4703	2920	37.91%
PML, deep, 5 Hz	7.68	5124	3036	40.75%
PML, center, 8 Hz	4.80	3792	2513	33.73%
FS, center, 5 Hz	7.68	9238	4844	47.56%

Weighted CARP-CG reaches the target in fewer iterations for all five PML configurations in Table 4. The reductions range from 33.73% to 40.75%, with counts of 2513–3376 for Weighted CARP-CG and 3792–5651 for CARP-CG. The reduction is smaller at 8 Hz than at 5 Hz on the fixed grid. Fig. 4 shows comparable initial decreases followed by faster late-stage residual decay for Weighted CARP-CG.

Supplementary material, Section S3.1 includes a 12 Hz residual-convergence test on a grid with fewer than four points per minimum S-wave wavelength and comparisons with unpreconditioned GMRES(50) [33], CGNR, and BiCGSTAB [34]. Each recorded run on the three representative PML systems allows at most 8000 iterations. None of the three unpreconditioned Krylov methods reaches the prescribed residual target in these runs.

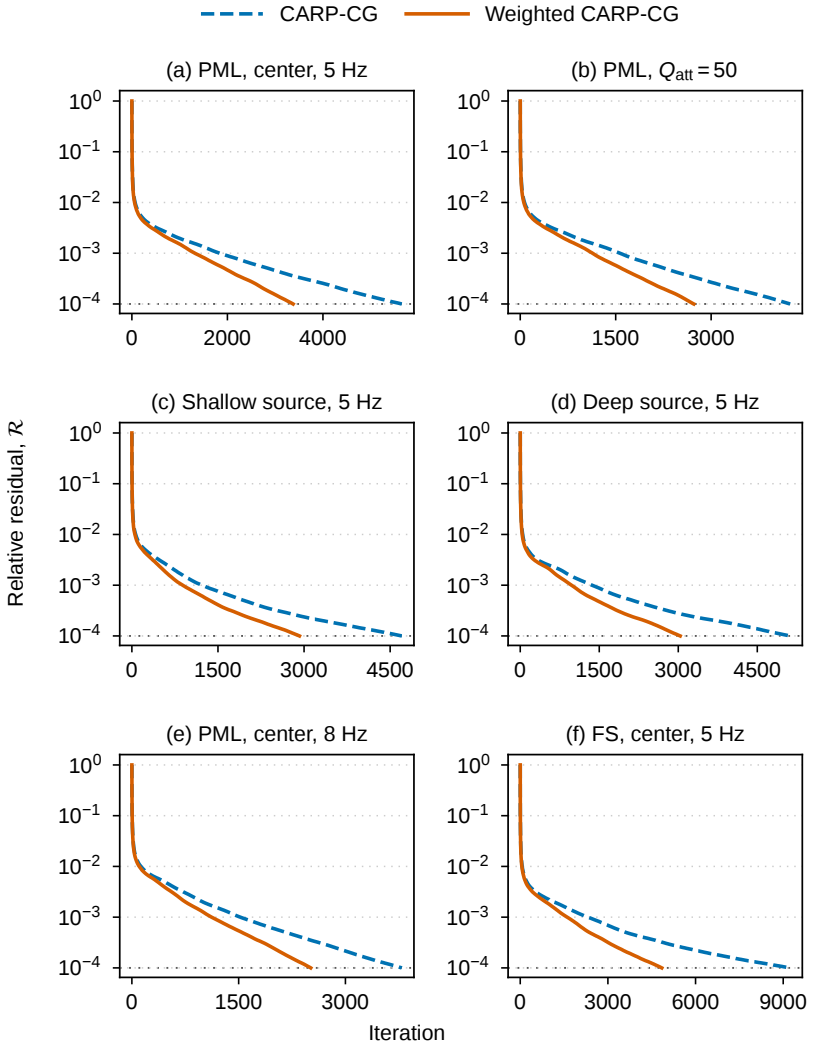


Figure 4: Histories of the relative residual $\mathcal{R}$ defined in equation 32 for the 2D Marmousi2 tests satisfying $N_{\lambda,S} \geq 4$: (a) central source at 5 Hz, (b) increased attenuation, (c) shallow source in the low-V_S region, (d) deep source, (e) central source at 8 Hz, and (f) the central-source free-surface case at 5 Hz. Panels (a)–(e) use PMLs on all sides; panel (f) retains PMLs on the sides and bottom. The line styles distinguish CARP-CG from Weighted CARP-CG.

5.2.4. Surface-boundary and Poisson's ratio effects

We first replace the PML at the model surface in the heterogeneous model by a traction-free boundary, then use fixed Poisson's ratios to separate boundary effects from changes in volumetric stiffness.

The FS row of Table 4 uses the central 5 Hz PML configuration with only the top boundary replaced by the fourth-order/ALM traction-free treatment in Supplementary material, Section S1. Material, source, attenuation, partition, and solver settings are unchanged. Removing the PML at the model surface changes the computational grid from 121×121 to 121×111, giving 26 862 complex-valued velocity DOFs on the same 101×101 physical grid.

For the free-surface case, CARP-CG requires 9238 iterations and Weighted CARP-CG requires 4844, a reduction of 47.56%. Both counts exceed the corresponding central-source PML counts of 5651 and 3376. Fig. 4(f) shows the two residual histories.

We next examine high Poisson's ratios with and without a traction-free top. Four controlled variants of the central 5 Hz, $h = 20$ m configuration combine $\nu = 0.25$ or 0.49 with a PML at the model surface or a free surface. The heterogeneous V_S and density fields, $Q_{att} = 1000$, source, and solver settings remain fixed. We replace the original P-wave speed by $V_P = V_S\sqrt{2(1-\nu)/(1-2\nu)}$. Thus $\lambda/\mu = 2\nu/(1-2\nu)$ increases from 1 to 49 while the minimum number of grid points per S-wave wavelength remains 7.68. This isolates the change in the elastic coefficients from a change in shear-wave resolution.

Table 5
Controlled Poisson's ratio and surface-boundary comparisons at 5 Hz, $h = 20$ m, $\beta = 0.5$, and $\mathcal{R} \le 10^{-4}$.

		Iterations			Iteration-loop time	
ν	Top	CARP-CG	Weighted	Reduction	CARP-CG	Weighted
0.25	PML	4787	3008	37.16%	1.439 s	1.642 s
0.25	FS	7587	4080	46.22%	2.108 s	2.091 s
0.49	PML	28 278	8391	70.33%	8.529 s	4.608 s
0.49	FS	32 781	8876	72.92%	9.068 s	4.522 s

All eight solves in Table 5 reach $\mathcal{R} \le 10^{-4}$. Increasing ν raises the counts for both methods and surface boundary conditions. At $\nu = 0.49$, Weighted CARP-CG reduces iterations by 70.33% with PML and 72.92% with FS; iteration-loop time decreases by 45.98% and 50.13%. At $\nu = 0.25$, iteration reductions of 37.16% and 46.22% give little or no time saving: PML iteration-loop time increases from 1.439 s to 1.642 s, while FS iteration-loop time changes from 2.108 s to 2.091 s. The larger iteration reductions at $\nu = 0.49$ are accompanied by larger savings in iteration-loop time than at $\nu = 0.25$.

For the heterogeneous free-surface case, the iteration-loop times are 2.559 s for CARP-CG and 2.462 s for Weighted CARP-CG. The 47.56% iteration reduction therefore gives a 3.79% reduction in iteration-loop time. In the $\nu = 0.25$ PML case, the 37.16% iteration reduction instead accompanies a 14.1% increase in iteration-loop time. The weights remain fixed after setup but still enter every row update, component average, and CG dot product. On these small systems, communication and synchronization over 16 MPI processes may also account for a substantial fraction of the iteration time.

Supplementary material, Sections S1.4 and S1.5 give the two-dimensional free-surface discretization checks; Section S3.1 provides additional residuals and timings.

5.3. 3D experiments on the modified Overthrust model

The three-dimensional study first uses small systems to examine the eigenvalues of the matrices used by CG. It then measures iteration counts and iteration-loop time as frequency, grid spacing, and Poisson's ratio change. The final tests use the full model to compare source depths and surface boundary conditions and to report memory use.

5.3.1. Model and study design

The benchmark is a modified elastic model constructed from a spatial subregion of the SEG/EAGE Overthrust model [35], with $350\times350\times160$ physical samples, $h = 25$ m, and a 450 m low-velocity elastic layer at the top. Below that layer, V_S is estimated from V_P using the empirical relation of Brocher [36], with the spatially varying Poisson's ratio constrained to $0.20 \le \nu \le 0.40$. Density is estimated from V_P using the Gardner relation [37]. The resulting

ranges are $V_P = 1500$–6000 m/s, $V_S = 612.37$–3549.40 m/s, $\rho = 1926$–2723 kg/m^3, and $\nu = 0.215$–0.400. Fig. 5 shows the elastic parameter models; "Dip" and "Cross" denote the x and y directions.

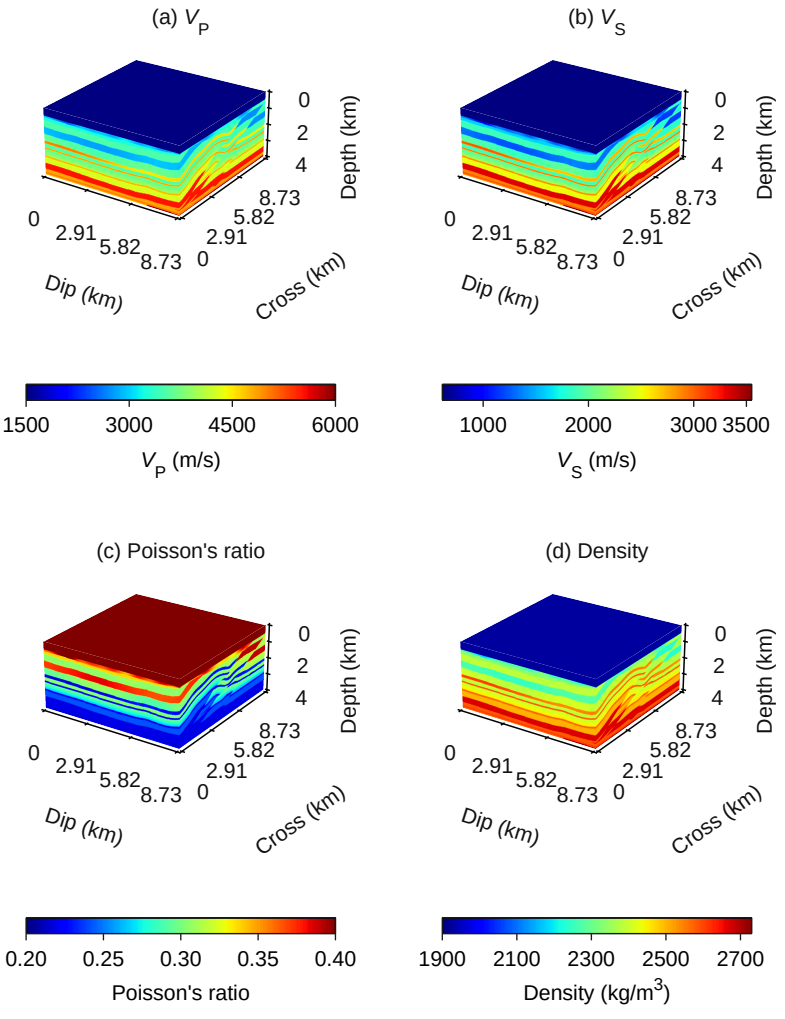


Figure 5: Elastic parameter models derived from the modified Overthrust-derived model: (a) V_P, (b) V_S, (c) Poisson's ratio, and (d) density. The panels depict the same physical volume with a separate color scale for each field. "Dip" and "Cross" correspond to x and y, respectively. Density is expressed in kg/m^3.

The three-dimensional cost studies use lossless material, $\beta = 0.5$, and $\mathcal{R} \leq 10^{-4}$.

5.3.2. Spectra of the matrices used by CG

The CG bound in Theorem 2 depends on κ_+, the ratio of the largest to the smallest positive eigenvalue of $\widehat{\mathbf{H}}_W$. Small systems allow us to compute all eigenvalues and examine how the weighting exponent changes this ratio. These tests complement the iteration counts in Table 1.

The systems come from a $5\times5\times5$ spatial subregion around the source in the $h = 300$ m modified Overthrust model. Two PML layers on each side give 9^3 computational points and 2187 complex-valued velocity DOFs. We partition the rows into $2\times2\times2$ subdomains and use unit row relaxation. The eight row blocks are processed serially in the spectral checks. Two tests retain the spatially varying Poisson's ratio at 1 and 1.5 Hz. Two more tests impose constant $\nu = 0.25$ and 0.49 at 1 Hz while keeping the same V_S and ρ fields. Within each case, the matrix, row order, row partition, and duplication map $\mathbf{C}$ stay fixed as β takes the values 0, 0.5, and 1.

We assemble the complete forward–average–backward–average map and compute every eigenvalue of $\widehat{\mathbf{H}}_W = \mathbf{M}^{1/2}\mathbf{H}_W\mathbf{M}^{-1/2}$. Equal weights give the CARP double sweep with its natural metric $\mathbf{M}_0$. Supplementary material, Section S3.3 provides the inputs and numerical checks.

Fig. 6(a,b) shows that weighting raises the smallest eigenvalues and lowers the condition number of the positive spectrum. The full-spectrum calculations summarized in Supplementary material, Section S3.3 give largest eigenvalues close to one. At $\beta = 0.5$, the condition numbers fall by factors of 2.77–3.38 across the four tests. For $\nu = 0.25$, the smallest eigenvalue rises from 9.62×10^{-5} to 2.69×10^{-4}; for $\nu = 0.49$, it rises from 1.99×10^{-7} to 6.73×10^{-7}. At each tested exponent, the $\nu = 0.49$ case has a smaller minimum eigenvalue than the $\nu = 0.25$ case; at both Poisson's ratios, $\beta = 0.5$ raises this minimum relative to equal weights.

In all four tests, $\beta = 1$ gives a larger condition number than $\beta = 0.5$. Among the three tested exponents, these spectra therefore favor $\beta = 0.5$. The same exponent also requires the fewest iterations in Table 1.

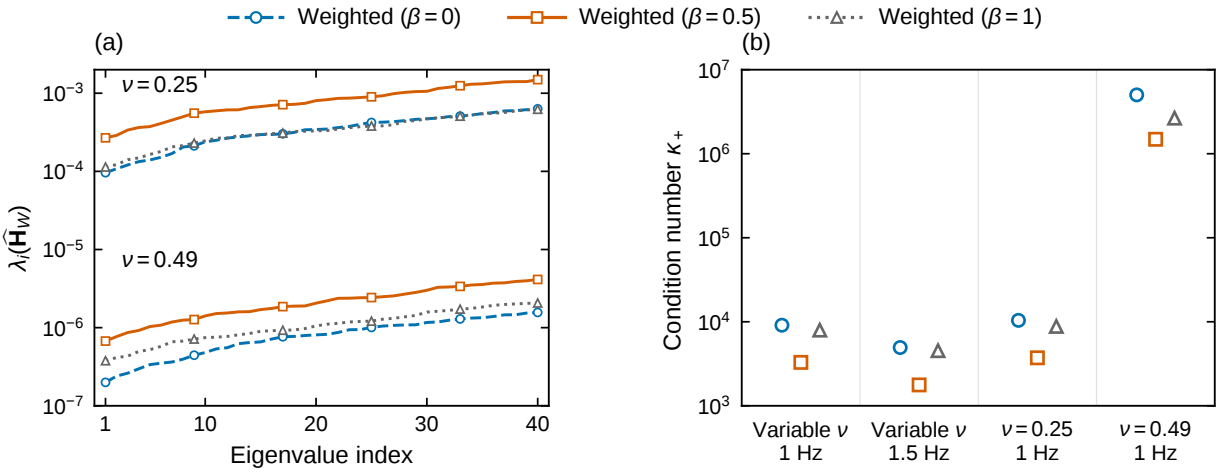


Figure 6: Effect of the weight exponent on the spectra of $\widehat{\mathbf{H}}_W$ for the 2187-unknown systems. (a) The smallest 40 eigenvalues for the two tests with fixed Poisson's ratios. (b) Condition numbers computed from positive eigenvalues for all four tests.

5.3.3. Frequency and grid-size effects on iteration cost

Having examined the small-system spectra, we now measure computational cost on larger systems. We distinguish increasing frequency at an approximately fixed number of grid points per S-wave wavelength from refining the grid at a fixed frequency. Table 6 includes both comparisons: the tests with $N_{\lambda,\mathrm{S}} \approx 4.08$ increase frequency while refining the grid, and the three 1.5 Hz tests use progressively finer grids at fixed frequency. Here n_x, n_y, n_z count the physical-domain grid points. The total number of complex-valued velocity DOFs, N_u, includes the PML and equals the linear-system dimension n.

Each case adds ten PML cells on every side and divides the computational domain into 4×4×2 subdomains over 32 single-threaded MPI processes. The physical PML thickness therefore changes with h, and the material coefficients are sampled on each grid. The source stays at (4375, 4375, 2000) m. When this location falls between grid nodes, trilinear weights distribute the source to the normal-stress equations before the staggered divergence is applied. The reported iteration count is that at the first residual check that meets the target; checks are performed every 50 iterations. Both reduction columns give the percentage decrease relative to CARP-CG.

The five-frequency sequence from 0.5 to 6 Hz keeps approximately 4.08 points per minimum S-wave wavelength. As frequency increases, the grid becomes finer and the system grows from 255 000 to 73 926 000 complex-valued velocity DOFs. Fig. 7 shows the residual histories and plots the iteration counts against $N_g = (n_x n_y n_z)^{1/3}$, the geometric mean of the physical-domain grid-point counts. The dotted line is proportional to N_g and serves as a visual

Table 6
Frequency and grid-spacing effects on three-dimensional iteration cost at $\mathcal{R} \leq 10^{-4}$ and $\beta = 0.5$.

					Iterations			Iteration-loop time		
f (Hz)	h (m)	$n_x \times n_y \times n_z$	N_u	$N_{\lambda,\mathrm{S}}$	CARP-CG	Weighted	Reduction	CARP-CG	Weighted	Reduction
0.5	300	$30 \times 30 \times 14$	255 000	4.08	2800	1350	51.79%	26.43 s	14.52 s	45.08%
0.75	200	$45 \times 45 \times 21$	519 675	4.08	3000	1300	56.67%	57.84 s	28.78 s	50.25%
1.5	100	$88 \times 88 \times 41$	2 134 512	4.08	6300	2250	64.29%	506.69 s	203.11 s	59.91%
1.5	75	$117 \times 117 \times 54$	4 166 718	5.44	9900	3650	63.13%	1504.79 s	638.68 s	57.56%
1.5	50	$176 \times 176 \times 81$	11 640 048	8.16	12 900	4650	63.95%	6802.54 s	2607.24 s	61.67%
3	50	$176 \times 176 \times 81$	11 640 048	4.08	11 000	3900	64.55%	5409.52 s	2179.53 s	59.71%
6	25	$350 \times 350 \times 160$	73 926 000	4.08	11 900	4250	64.29%	36 555.72 s	14 084.84 s	61.47%

reference for this frequency and grid sequence. The largest case, with a 6 Hz central source and PML boundaries, is also the reference configuration for Section 5.3.5.

To measure iteration cost under grid refinement, we hold the frequency at 1.5 Hz and use $h = 100, 75$, and 50 m with ten PML cells on each side. Both methods require more iterations and more iteration-loop time as the grid is refined. Across these grids, Weighted CARP-CG reduces iterations by 63.1% to 64.3% and iteration-loop time by 57.6% to 61.7%. The $h = 50$ m grid also allows a frequency comparison without changing the grid: increasing f from 1.5 to 3 Hz lowers the counts from 12 900 to 11 000 for CARP-CG and from 4650 to 3900 for Weighted CARP-CG. On this fixed grid, both solvers require fewer iterations at the higher frequency. Along the sequence with approximately 4.08 points per minimum S-wave wavelength, frequency and grid size change together. The increases in iteration count and iteration-loop time along that sequence therefore also reflect grid refinement.

Across the six configurations combining frequencies of 0.5–3 Hz with different grid spacings, Weighted CARP-CG reduces iterations by 51.8–64.5% and iteration-loop time by 45.1–61.7%. At 6 Hz, the corresponding reductions are 64.3% and 61.5%. These savings extend from the coarsest system to the 73 926 000-unknown case. Weighted CARP-CG also requires fewer iterations as the residual decreases from its first sampled value below 10^{-3} to the stopping target of 10^{-4}. At 3 Hz and $h = 50$ m, moving from the first sampled residual below 10^{-3} to 10^{-4} takes a further 7900 iterations for CARP-CG and 2750 for Weighted CARP-CG; the corresponding 0.5 Hz counts are 1400 and 650.

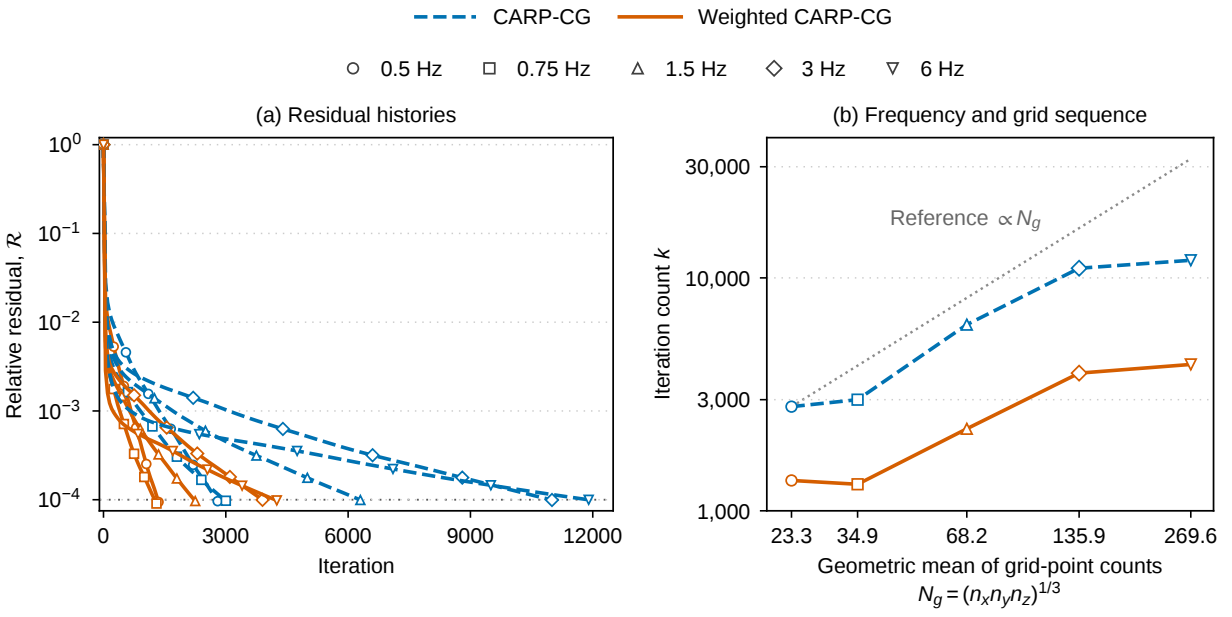


Figure 7: Frequency sequence with approximately 4.08 points per minimum S-wave wavelength: $(f, h) = (0.5\,\mathrm{Hz}, 300\,\mathrm{m})$, $(0.75\,\mathrm{Hz}, 200\,\mathrm{m})$, $(1.5\,\mathrm{Hz}, 100\,\mathrm{m})$, $(3\,\mathrm{Hz}, 50\,\mathrm{m})$, and $(6\,\mathrm{Hz}, 25\,\mathrm{m})$. (a) Relative residuals sampled every 50 iterations. (b) Iteration counts at $\mathcal{R} \le 10^{-4}$ versus $N_g = (n_x n_y n_z)^{1/3}$, the geometric mean of the physical-domain grid-point counts, excluding PML. Colors and line styles identify the methods; marker shapes identify the frequencies. The dotted line is proportional to N_g and anchored at the first CARP-CG point as a visual reference.

5.3.4. Controlled Poisson's ratio comparison

To study changes in material coupling on a fixed grid, we use a 41^3 submodel extracted from the interior of the modified Overthrust model. We retain its spatially varying V_S and ρ fields and set $V_P = V_S\sqrt{2(1-\nu)/(1-2\nu)}$ for constant $\nu = 0.25$ and 0.49. The original model has spatially varying ν; these two tests replace it by a constant value while keeping the shear-wave resolution fixed. Both submodels remain heterogeneous. The shear modulus $\mu = \rho V_S^2$ is unchanged, while λ/μ increases from 1 to 49.

Both tests use 6 Hz, $h = 62.5$ m, a central isotropic source, eight PML layers, and a $2 \times 2 \times 2$ decomposition over eight single-threaded MPI processes. The computational grid is 57^3, with 555 579 complex-valued velocity DOFs and $N_{\lambda,S} = 4.736$. Residuals are checked every 25 iterations.

Table 7
Effect of the constant Poisson's ratio on three-dimensional iteration cost at 6 Hz, $h = 62.5$ m, $\mathcal{R} \le 10^{-4}$ and $\beta = 0.5$.

	Iterations		Iteration-loop time	
ν	CARP-CG	Weighted	CARP-CG	Weighted
0.25	2100	725	93.56 s	38.90 s
0.49	3575	1250	160.01 s	67.33 s

In Table 7, increasing ν raises the iteration count by 70.2% for CARP-CG and 72.4% for Weighted CARP-CG. The grid, shear modulus, and S-wave resolution are unchanged, so this increase accompanies the greater volumetric stiffness. Changing ν also changes the coefficients that couple the velocity components, and hence the weights computed from the row-scaled column norms in equation 9. At both Poisson's ratios, Weighted CARP-CG uses approximately 65% fewer iterations and 58% less iteration-loop time. Its relative savings remain similar even though both methods take longer. Supplementary material, Section S3.2 compares the fields obtained at the stopping tolerance.

5.3.5. Full-model source and surface-boundary tests

Table 8 compares two source depths on the full model at 6 Hz, using a PML at the model surface or the three-dimensional FS construction specified in Supplementary material, Section S1.6. The central-source PML row reuses the 6 Hz reference case in Table 6. Both sources have $x = y = 4.375$ km, with depths of 2.000 and 0.950 km. The shallower source lies in a low-V_S interval below the top layer. Within each boundary configuration, changing the source depth changes the right-hand side while preserving the matrix and metric.

The PML computational grid is 370×370×180, with 73 926 000 complex-valued velocity DOFs and 3 743 245 440 stored entries. The FS computational grid is 370 × 370 × 170, with 69 819 000 complex-valued velocity DOFs and 3 534 528 120 stored entries. All four configurations use a 4×4×2 decomposition over 32 single-threaded MPI processes and residual checks every 50 iterations. The PML thickness is 250 m on every side in the PML configurations and on the lateral and bottom sides in the FS configurations.

Memory use. We measure process memory use by resident set size (RSS), the size of a process's memory pages currently held in physical RAM. For Weighted CARP-CG, Table 8 reports the mean of the peak RSS values measured separately for each MPI process.

Table 8
Large-system source and surface-boundary configurations at 6 Hz, $h = 25$ m. Resident memory use is reported as the mean per-process peak RSS for Weighted CARP-CG.

		Iterations		Iteration-loop time		
Boundary	Source	CARP-CG	Weighted	CARP-CG	Weighted	Mean per-process peak RSS (MiB)
PML	Center	11 900	4250	36 555.72 s	14 084.84 s	945.7
FS	Center	11 750	6050	34 881.24 s	19 676.84 s	921.5
PML	Shallow	7350	3800	22 715.15 s	12 743.18 s	981.2
FS	Shallow	14 250	7500	41 477.98 s	23 656.36 s	896.6

Moving the source closer to the surface reduces the PML iteration counts from 11 900 to 7350 for CARP-CG and from 4250 to 3800 for Weighted CARP-CG. Under FS, the corresponding counts increase from 11 750 to 14 250 and from 6050 to 7500. The FS iteration reductions are 48.51% and 47.37%, with iteration-loop time reductions of 43.59% and 42.97%. Fig. 8 shows all four pairs of residual histories. For additional sources with the same matrix and decomposition, the solver can reuse the weights, row denominators, and shared-variable index sets.

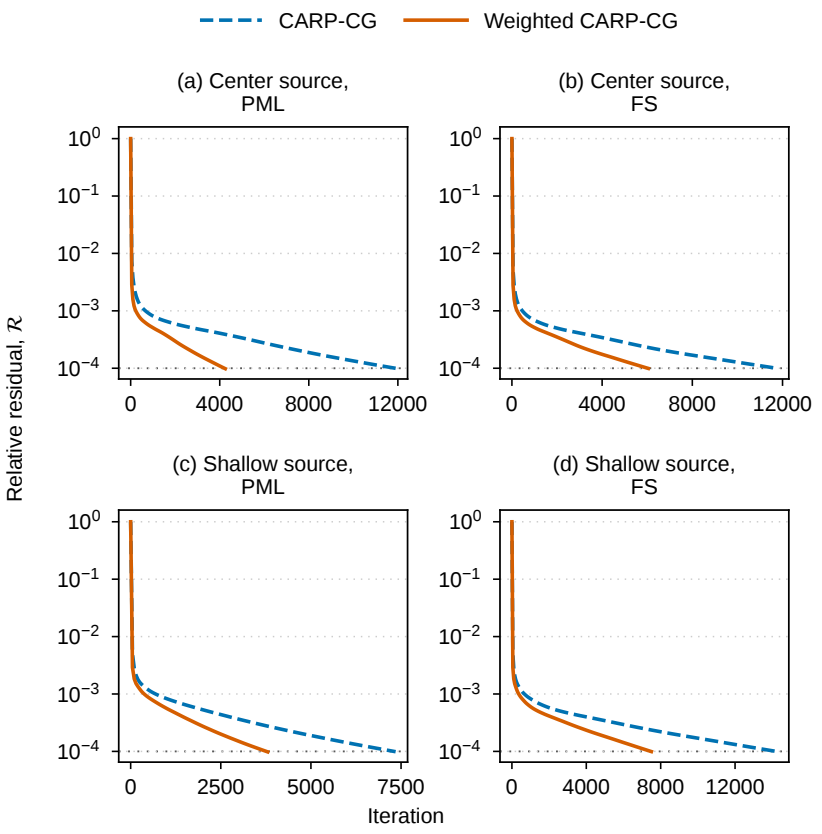


Figure 8: Histories of the relative residual $\mathcal{R}$ defined in equation 32 for the large-system 6 Hz source and surface-boundary configurations: (a) PML central source, (b) FS central source, (c) PML shallow source in the low-V_S region, and (d) FS shallow source in the low-V_S region. The relative residual $\mathcal{R}$ is evaluated every 50 iterations.

The full-model Weighted CARP-CG runs also quantify resident memory use with CSR-SER, the storage representation used by both solvers. Across these configurations, the mean per-process peak RSS is 896.6–981.2 MiB. For the central-source PML case, the sum of the individual process peaks is 31.73 GB, an upper bound on the sum of process RSS values at any one time.

To show the computed wavefield at this scale, Fig. 9 pairs material sections with the saved central-source velocity field from the full-grid PML case.

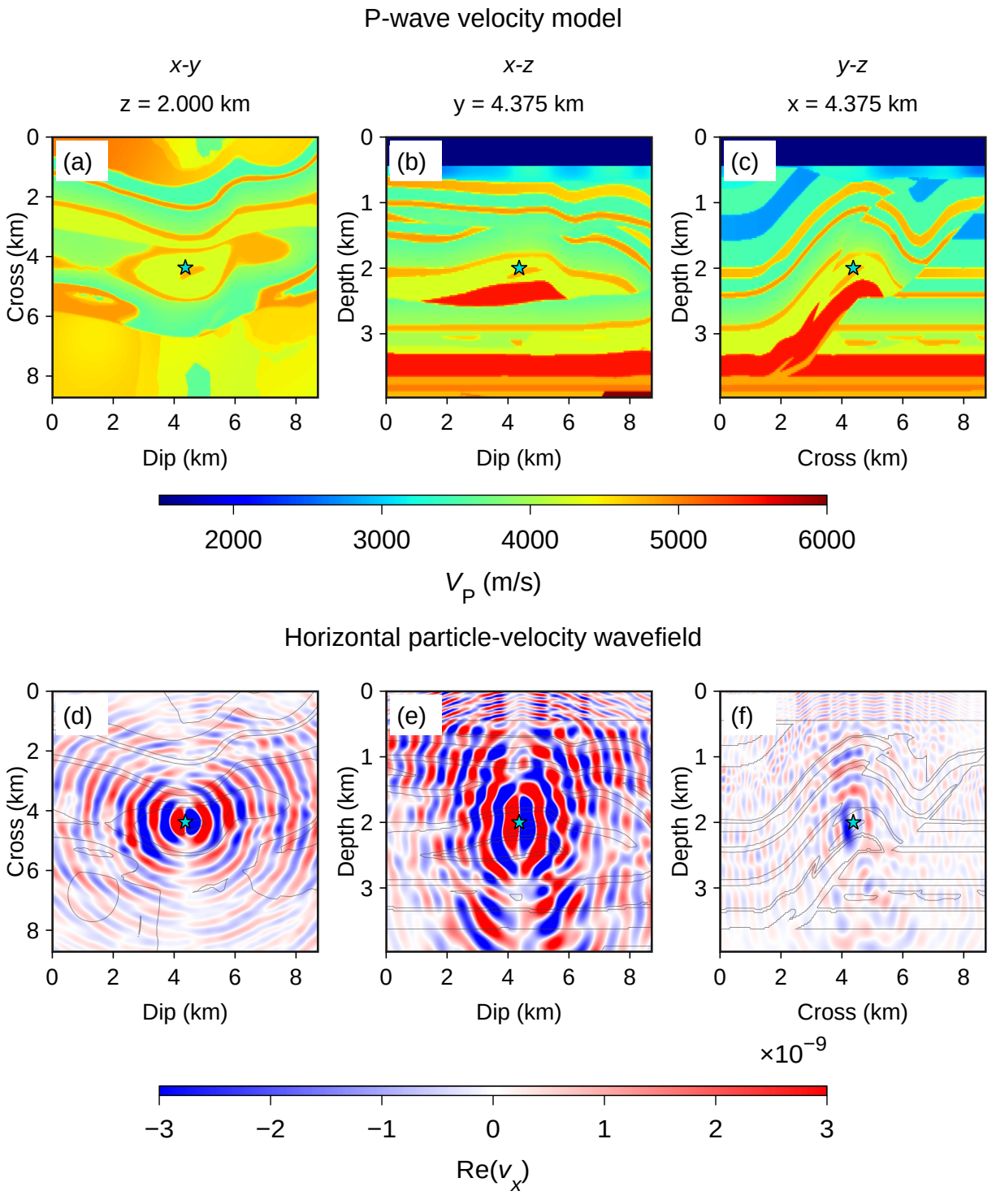


Figure 9: Material and Weighted CARP-CG wavefield sections at the central source indices for the large 6 Hz PML case. Panels (a)–(c) show V_P; panels (d)–(f) show $\mathrm{Re}(v_x)$. Columns show sections at the grid coordinates $z = 2.000$ km, $y = 4.375$ km, and $x = 4.375$ km, respectively. Velocity samples retain their $h/2$ stagger in x relative to the material nodes. Each row shares a color scale. Stars mark the source, and model contours are overlaid on the wavefield. Amplitudes use the source normalization in Supplementary material, Section S1.3. "Dip" and "Cross" denote x and y.

6. Conclusions

Weighted CARP-CG uses squared column norms of the row-scaled local matrices to construct fixed positive weights for each local variable copy. The same weights govern row corrections and component averaging in the W-inner product and induce the M-inner product for CG on the original variables. For consistent systems under the stated sweep conditions, the analysis establishes self-adjointness, positive semidefiniteness, solution-set equivalence, and CG convergence in exact arithmetic. The construction retains ordered local sweeps, short recurrences, and the exchange pattern for shared variables.

In the two-dimensional tests with absorbing boundaries, CARP-CG and Weighted CARP-CG give comparable field errors against the same direct solutions at the common residual target. Among the tested exponents, $\beta = 0.5$ requires the fewest iterations in the two-dimensional exponent study and gives the smallest condition numbers in the four small-system spectral tests. Across six three-dimensional tests combining frequencies of 0.5–3 Hz with different grid spacings and two further tests at different Poisson's ratios, iteration counts decrease by 51.8–65.5% and iteration-loop time by 45.1–61.7% at the row-scaled residual target of 10^{-4}. For the 73 926 000-unknown central-source PML case at 6 Hz, iteration-loop time decreases from 36 555.72 s to 14 084.84 s, a reduction of 61.47%. The time savings depend on the cost of the weighted updates: the two-dimensional free-surface comparison uses 47.56% fewer iterations but only 3.79% less iteration-loop time, and the PML comparison at low Poisson's ratio takes more iteration-loop time despite fewer iterations.

CSR-SER supports large systems by storing stencil-offset codes, regenerating coefficients, and reusing a three-row buffer. Shared by both solvers, it preserves the ordered sweeps and reduces persistent coefficient and column-index storage to 5% of the corresponding explicit CSR arrays. The implementation handles approximately 70 million complex-valued velocity DOFs on a 32 GiB workstation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Numerical data generated for the tables and figures are available from the corresponding author upon reasonable request. The source model datasets are described in [29, 35] and remain subject to their original licensing terms.

Supplementary material

The supplementary material contains the discretization and boundary checks (S1), weight-scaling identities and additive regularization (S2), additional numerical comparisons (S3), and CSR-SER equivalence and storage counts (S4).

Acknowledgements

This work was supported through grants to Guangyu Gao from the National Natural Science Foundation of China [grant number 12501585]; the China Postdoctoral Science Foundation [grant number 2025M774293]; the "Hong Kong Scholar" Program [grant number XJ202536]; and the Fundamental Research Funds for the Central Universities.

References

[1] Y. Li, L. Métivier, R. Brossier, B. Han, J. Virieux, 2D and 3D frequency-domain elastic wave modeling in complex media with a parallel iterative solver, Geophysics 80 (3) (2015) T101–T118. doi:10.1190/geo2014-0480.1.

[2] Y. Li, B. Han, L. Métivier, R. Brossier, Optimal fourth-order staggered-grid finite-difference scheme for 3D frequency-domain viscoelastic wave modeling, Journal of Computational Physics 321 (2016) 1055–1078. doi:10.1016/j.jcp.2016.06.018.

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

[4] S. Wang, M. V. de Hoop, J. Xia, X. S. Li, Massively parallel structured multifrontal solver for time-harmonic elastic waves in 3-D anisotropic media, Geophysical Journal International 191 (2012) 346–366. doi:10.1111/j.1365-246X.2012.05634.x.

[5] C. Weisbecker, P. Amestoy, O. Boiteau, R. Brossier, A. Buttari, J.-Y. L'Excellent, S. Operto, J. Virieux, 3D frequency-domain seismic modeling with a block low-rank algebraic multifrontal direct solver, in: 83rd Annual International Meeting, SEG, Expanded Abstracts, 2013, pp. 3411–3416. doi:10.1190/segam2013-0603.1.

[6] G. Rizzuti, W. A. Mulder, Multigrid-based 'shifted-Laplacian' preconditioning for the time-harmonic elastic wave equation, Journal of Computational Physics 317 (2016) 47–65. doi:10.1016/j.jcp.2016.04.049.

[7] E. Treister, R. Yovel, A hybrid shifted Laplacian multigrid and domain decomposition preconditioner for the elastic Helmholtz equations, Journal of Computational Physics 497 (2024) 112622. doi:10.1016/j.jcp.2023.112622.

[8] R. Yovel, E. Treister, LFA-Tuned Matrix-Free Multigrid Method for the Elastic Helmholtz Equation, SIAM Journal on Scientific Computing 46 (5) (2024) S1–S21. doi:10.1137/23M1583466.

[9] R. Yovel, E. Treister, A Block-Acoustic Preconditioner for the Elastic Helmholtz Equation, SIAM Journal on Scientific Computing 48 (4) (2026) A1898–A1924. doi:10.1137/24M1712709.

[10] D. Appelö, F. Garcia, A. Alvarez Loya, O. Runborg, El-WaveHoltz: A time-domain iterative solver for time-harmonic elastic waves, Computer Methods in Applied Mechanics and Engineering 401 (2022) 115603. doi:10.1016/j.cma.2022.115603.

[11] D. Gordon, R. Gordon, CARP-CG: A robust and efficient parallel solver for linear systems, applied to strongly convection dominated PDEs, Parallel Computing 36 (2010) 495–515. doi:10.1016/j.parco.2010.05.004.

[12] S. Kaczmarz, Angenäherte Auflösung von Systemen linearer Gleichungen, Bulletin International de l'Académie Polonaise des Sciences et des Lettres 35 (1937) 355–357.

[13] D. Gordon, R. Gordon, Component-averaged row projections: A robust, block-parallel scheme for sparse linear systems, SIAM Journal on Scientific Computing 27 (2005) 1092–1117. doi:10.1137/040609458.

[14] Å. Björck, T. Elfving, Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations, BIT 19 (2) (1979) 145–163. doi:10.1007/BF01930845.

[15] Y. Censor, D. Gordon, R. Gordon, Component averaging: An efficient iterative parallel algorithm for large and sparse unstructured problems, Parallel Computing 27 (6) (2001) 777–808. doi:10.1016/S0167-8191(00)00100-9.

[16] Y. Censor, D. Gordon, R. Gordon, BICAV: A block-iterative parallel algorithm for sparse systems with pixel-related weighting, IEEE Transactions on Medical Imaging 20 (10) (2001) 1050–1060. doi:10.1109/42.959302.

[17] Y. Censor, T. Elfving, Block-iterative algorithms with diagonally scaled oblique projections for the linear feasibility problem, SIAM Journal on Matrix Analysis and Applications 24 (2002) 40–58. doi:10.1137/S089547980138705X.

[18] D. Gordon, The Cimmino–Kaczmarz equivalence and related results, Applied Analysis & Optimization 2 (2018) 253–270 [cited 22 September 2026].
URL https://cs.haifa.ac.il/~gordon/cke.pdf

[19] K. Kourtis, G. Goumas, N. Koziris, Exploiting compression opportunities to improve SpMxV performance on shared memory systems, ACM Transactions on Architecture and Code Optimization 7 (3) (2010) 16:1–16:31. doi:10.1145/1880037.1880041.

[20] S. Boukhris, A. Napov, Y. Notay, Algebraic multigrid using a stencil–CSR hybrid format on GPUs, SIAM Journal on Scientific Computing 45 (3) (2023) C154–C178. doi:10.1137/22M1480938.

[21] T. van Leeuwen, F. J. Herrmann, 3D frequency-domain seismic inversion with controlled sloppiness, SIAM Journal on Scientific Computing 36 (5) (2014) S192–S217. doi:10.1137/130918629.

[22] J. Virieux, P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method, Geophysics 51 (1986) 889–901. doi:10.1190/1.1442147.

[23] W. C. Chew, Q. H. Liu, Perfectly matched layers for elastodynamics: A new absorbing boundary condition, Journal of Computational Acoustics 4 (4) (1996) 341–359. doi:10.1142/S0218396X96000118.

[24] D. Komatitsch, J. Tromp, A perfectly matched layer absorbing boundary condition for the second-order seismic wave equation, Geophysical Journal International 154 (2003) 146–153. doi:10.1046/j.1365-246X.2003.01950.x.

[25] P. A. Knight, D. Ruiz, B. Uçar, A symmetry preserving algorithm for matrix scaling, SIAM Journal on Matrix Analysis and Applications 35 (2014) 931–955. doi:10.1137/110825753.

[26] S. Diamond, S. Boyd, Stochastic matrix-free equilibration, Journal of Optimization Theory and Applications 172 (2017) 436–454. doi:10.1007/s10957-016-0990-2.

[27] M. R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, Journal of Research of the National Bureau of Standards 49 (6) (1952) 409–436. doi:10.6028/jres.049.044.

[28] Y. Saad, Iterative methods for sparse linear systems, 2nd Edition, Society for Industrial and Applied Mathematics, Philadelphia, 2003. doi:10.1137/1.9780898718003.

[29] G. S. Martin, R. Wiley, K. J. Marfurt, Marmousi2: An elastic upgrade for Marmousi, The Leading Edge 25 (2006) 156–166. doi:10.1190/1.2172306.

[30] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental algorithms for scientific computing in Python, Nature Methods 17 (2020) 261–272. doi:10.1038/s41592-019-0686-2.

[31] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, J. W. H. Liu, A supernodal approach to sparse partial pivoting, SIAM Journal on Matrix Analysis and Applications 20 (1999) 720–755. doi:10.1137/S0895479895291765.

[32] T. A. Davis, J. R. Gilbert, S. I. Larimore, E. G. Ng, A column approximate minimum degree ordering algorithm, ACM Transactions on Mathematical Software 30 (2004) 353–376. doi:10.1145/1024074.1024079.

[33] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 7 (1986) 856–869. doi:10.1137/0907058.

[34] H. A. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 13 (1992) 631–644. doi:10.1137/0913035.

[35] F. Aminzadeh, J. Brac, T. Kunz, 3-D salt and Overthrust models, no. 1 in SEG/EAGE 3-D Modeling Series, Society of Exploration Geophysicists, 1997.

[36] T. M. Brocher, Empirical relations between elastic wavespeeds and density in the Earth's crust, Bulletin of the Seismological Society of America 95 (2005) 2081–2092. doi:10.1785/0120050077.

[37] G. H. F. Gardner, L. W. Gardner, A. R. Gregory, Formation velocity and density—the diagnostic basics for stratigraphic traps, Geophysics 39 (1974) 770–780. doi:10.1190/1.1440465.