Accelerating Spectral Elements Method with Extended Precision: A Case Study

Alexandre Hoffmann*, Yves Durand, Jérôme Fereyre
CEA-LIST Institute, University of Grenoble Alpes, CEA List, F-38000 Grenoble, France.

*Corresponding author. Email: alexandre.hoffmann@cea.fr (A.H.)

Manuscript submitted March 20, 2023; revised May 4, 2023; accepted August 7, 2023; published April 16, 2024.
doi: 10.17706/ijapm.2024.14.2.45-58

Abstract: Krylov methods play a major role in solving Partial Differential Equations (PDEs) due to their scalability and low memory requirements. However, for difficult problems, Krylov methods exhibit slow convergence and may even not always converge. Increasing the numerical precision can improve the convergence rate of Krylov methods. In the current work, we evaluate the effect of Variable Precision (VP) on two Krylov-based solvers. Our solvers were applied to a relatively difficult PDE, discretized with the Spectral Element Method (SEM), which produces a set of dense and poorly conditioned system of linear equations. We show that, increasing the numerical precision allows us to both speedup the convergence of the solver and more accurately estimate the residual error, making the solver more reliable.

Keywords: Algebraic solver, extended precision, spectral elements

1. Introduction

The wave equation is ubiquitous in modern physics: it is applied in a variety of applications such as mechanics, acoustics, electromagnetism, fluid dynamics, quantum mechanics, etc.

As an example, it plays a critical role in practical geophysics both in direct and inverse problems. When dealing with large scale Three Dimensional (3D) models, the wave equation is usually solved in the time domain [1–7] and eventually transformed into the frequency domain via a Fast Fourier Transform (FFT) [1].

However, there is still motivation to solve the wave equation in the frequency domain as it is easier to implement a Perfectly Matched Layer (PML) [8, 9] and some physical properties, such as attenuation, in this domain. Moreover, working in the frequency domain drastically reduces the amount of data to manage during inverse problems.

In the frequency domain, the wave equation becomes an indefinite Helmholtz equation, which is challenging to solve with Krylov methods [10]. The aim of the current work is to evaluate the effect of Variable Precision (VP) on two standard Krylov based solvers for the acoustic wave equation.

The remainder of this paper is organized as follows. First, we present the equation to be solved. Then we explain how we discretize it using a Galerkin method and briefly discuss the properties of the discretized equations. Next, we introduce two Krylov methods and explain how finite precision impacts them. Finally, we show how VP affects the convergence of two iterative solvers.

2. Case Study: The Harmonic Wave Equation

2.1. Formulation

In most, if not all, applications, the modeled wavefield u is assumed to propagate in an unbounded domain. A popular method to simulate an infinite domain is the Perfectly Matched Layer (PML): it is an absorbing layer Ω^{pml} inserted at the boundary of the domain Ω so that traveling waves can be properly absorbed before reaching the boundary of the domain $\partial\Omega$.

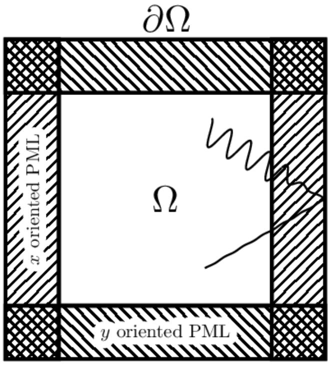


Fig. 1. Illustration of a PML. The truncated computational domain Ω is represented in white.

The absorbing layer Ω^{pml} is hatched with different hatching pattern for the x oriented PMLs and for the y oriented PMLs. The boundary of the whole domain is denoted $\partial\Omega$. We also illustrate the behavior of a waveform traversing the PML. After entering Ω^{pml}, the waveform is almost completely attenuated before reaching $\partial\Omega$ and being reflected back into Ω.

Our absorbing layer is modeled as a complex-valued coordinate transformation which changes oscillating waves into exponentially decaying waves, thus almost nullifying the reflected waves c.f. Fig. 1. The frequency domain acoustic wave equation with a PML can be written as:

$$\begin{gathered} -c^2\nabla.\nabla u - \omega^2 u = f \; over \; \Omega \, , \\ -c^2\tilde{\nabla}.\tilde{\nabla} u - \omega^2 u = f \; over \; \Omega^{\text{pml}}, \\ u = 0 \quad on \; \partial\Omega \end{gathered} \tag{1}$$

where u is the acoustic pressure in, c is the wave velocity in m/s, ω is the angular frequency in rad/s and $\tilde{\nabla}$ is a modified differential operator obtained by a change of coordinate in the absorbing layer:

$$\tilde{\nabla} = \begin{pmatrix} 1/\gamma_x & \partial/\partial x \\ 1/\gamma_y & \partial/\partial y \end{pmatrix}. \tag{2}$$

Here γ_x and γ_y are complex valued functions and their form will determine the effectiveness of our absorbing layer. In the current work, we follow Bermudez's framework [11]. We thus define Ω as$(-a, a) \times (-b, b)$, $\Omega \cup \Omega^{pml}$as $(-a^*, a^*) \times (-b^*, b^*)$, and:

$$\gamma_x(x) = \begin{cases} 1 & if \; |x| < a \\ 1 + \frac{i}{\omega}\,\sigma_x(|x|) & if \; a \leq |x| < a^* \end{cases} \tag{3}$$

$$\gamma_y(y) = \begin{cases} 1 & if \; |y| < b \\ 1 + \dfrac{i}{\omega}\,\sigma_y(|y|) & if \; b \leq |y| < b^* \end{cases}$$

with the following absorbing functions:

$$\sigma_x(x) = \frac{c}{a^* - x}, \quad \sigma_y(y) = \frac{c}{b^* - y}. \tag{4}$$

In order to solve Eq. (1) with a Galerkin method, we first need to formulate it in its equivalent weak form. This is obtained by integrating it against an arbitrary test function $v \in \mathbb{V}$ over $\Omega \cup \Omega^{\mathrm{pml}}$, then by integrating by parts over $\Omega \cup \Omega^{\mathrm{pml}}$ and finally by applying the Dirichlet boundary conditions. This gives:

$$\int_{\Omega} c^2 \nabla u \cdot \nabla v - \omega^2 uv \, dx + \int_{\Omega^{pml}} c^2 \nabla u^T \Gamma \nabla v - \gamma_x \gamma_y u \, v \, dx = \int_{\Omega} fv dx \;\; \forall v \in \mathbb{V}, \tag{5}$$

where:

$$\Gamma = \begin{pmatrix} \gamma_y/\gamma_x & 0 \\ 0 & \gamma_x/\gamma_y \end{pmatrix} \tag{6}$$

Note that, as a result of the integration by parts, the Left Hand Side (LHS) in Eq. (5) is bounded as long as u and v are in H1,0($\Omega \cup \Omega^{\mathrm{pml}}$). At this point, we encounter two difficulties. First, the variational formulation of Eq. (5) is written as the difference of two positive-definite bilinear operators. It thus becomes indefinite as $|\omega/c|$ increases. Second, the operator $\widetilde{\nabla}$ introduces complex values into the bilinear operator, which remains symmetric but is no longer self-adjoint. These difficulties make it impossible to solve the linear system using the usual Conjugate Gradient (CG) method, as CG requires the linear operator to be both positive-definite and self-adjoint.

In the following sections we describe how Eq. (5) is discretized and how the resulting system of linear equations is solved.

2.2. The Spectral Element Method

In this work, we discretize Eq. (5) with two different methods: first, we apply the Finite Element Method (FEM), which is a standard method for solving Partial Differential Equations (PDEs). Second, we apply the Spectral Element Method (SEM), which is commonly used for solving the wave equation in the time domain. The first step of both FEM and SEM is to partition the computational domain into non-overlapping elements:

$$\begin{gathered} \Omega \cup \Omega^{\mathrm{pml}} = \bigcup_{e=0}^{N_c} \Omega^e \\ \Omega^{\mathrm{i}} \cup \Omega^{\mathrm{j}} = \emptyset \;\; i \neq j \end{gathered} \tag{7}$$

SEM is typically formulated for quadrangular elements (although a formulation for triangular elements exists [12]. We thus limit the shape of our elements to quadrangles and curved quadrangles.

We then compute the integrals over $\Omega \cup \Omega^{\mathrm{pml}}$ as a sum of integrals over the small elements Ω^e. The general method for computing such integrals is to map each element to a reference element Ω^{ref}, which is typically [−1,1]×[−1,1]. We then define a mapping function $T^e: \Omega^{ref} \rightarrow \Omega^e$, which allows us to compute the integral over all our elements Ω^e with a single quadrature defined on Ω^{ref}. For additional details, the reader is referred to [13, 14].

The mapping from the reference element to an arbitrary element is sketched in Fig. 2.

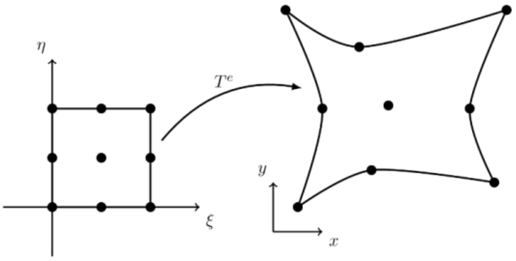


Fig. 2. Left, reference element Ω^{ref}. Right, an arbitrary element Ω^{e}. We use the coordinate system (ξ,η) for the reference element and (x, y) for the elements of our mesh. A transformation $\mathrm{T^e}$ maps Ω^{ref} to Ω^{e}.

Both FEM and SEM then project u and v onto a subspace of $H_0^1(\Omega)$ spanned by a set of interpolating functions $(\phi_n)_{n=1\ldots N_{dof}}$, which are constructed by composing a set of interpolating functions $(l_i)_{i=1,P+1}$, defined on Ω^{ref}, with the inverse of T^e. Assuming we have a mapping $globId$, between the local index *(e,i)* and the global index *n*, our interpolating function can be written as:

$$\phi_{globId(e,i)}(x) = l_i^e(x) \coloneqq (l_i \circ (T^e)^{-1})(x) \tag{8}$$

The reader may refer to [13, 14] for a more in-depth description of how this operation affects the differential operators. Fig. 3 illustrates how FEM and SEM differ in the way they define their interpolating functions. While FEM usually uses linear interpolating functions over Ω^{ref}, SEM uses higher degree Lagrange interpolating functions.

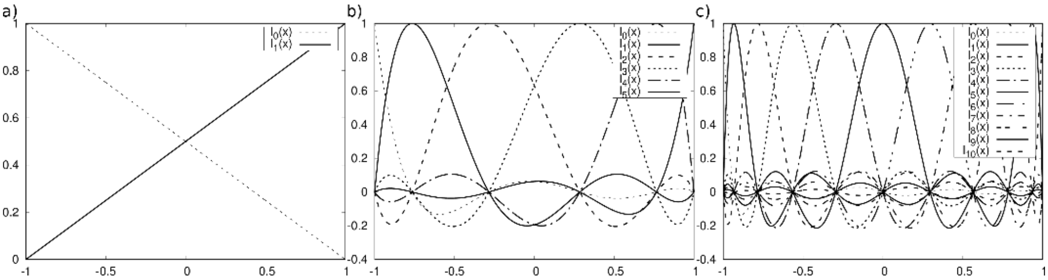


Fig. 3. Comparison of the interpolating functions used by a) FEM, b) SEM with fifth order polynomials, c) SEM with tenth order polynomials.

The interpolation nodes are typically computed using Gauss Lobatto-Legendre (GLL) quadrature points.

The accuracy of SEM is controlled by both the size of an element, *Δx*, and by the degree, *P*, of the interpolating polynomials, l_i^e. As an example, when dealing with the wave equation, if *P* is too small, SEM is not significantly more precise than FEM [15]. On the other hand, if *P* is too large, the discretized operators become denser and their condition number increases [16]. Optimal values for *P* range from 5 to 10 [17]. We study three values of *P* for the SEM interpolating polynomials: *P* = 1 (which is equivalent to the FEM interpolating function), *P* = 5, and *P* = 10. In theory, the wave equation should become increasingly hard to solve as *P* increases. We discuss this assumption in the experimental section, as well as the impact of precision on this computation.

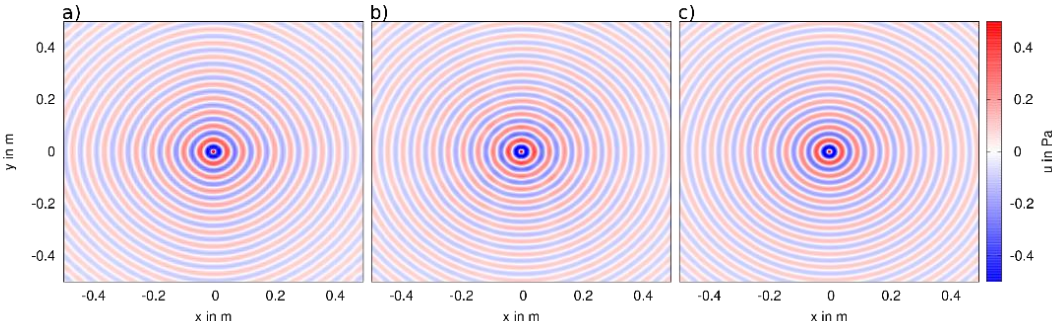


Fig. 4. Spatial representation of the real part of u obtained with a dirac impulse as our source, and discretized with: a) polynomials of order 1, b) polynomials of order 5, c) polynomials of order 10. The solutions have been interpolated on a 1000×1000 regular grid for comparison. We observe that, near the center of the domain, the wavefronts are better approximated by the higher order method c), even though the solution was obtained with the same number of DoFs per wavelength.

In order to ensure the accuracy of our method, we need at least 5 GLL points per wavelength [17–19]. Thus, a mesh for Ω designed for interpolating functions of degree one is comprised of 10 times more elements than a mesh designed for interpolating functions of degree 10. Fig. 4 shows the solution to the wave equation discretized with the three sets of polynomials mentioned above. We observe that, near the center of the domain, the solution obtained with higher order polynomials has a better approximation of the wavefronts, even-though it uses the same number of Degrees of Freedom (DoFs) per wavelength.

The PML is meshed with three elements in each direction regardless of the degree of the interpolating functions.

Table 1 shows the number of elements used in both directions within Ω as well as the total number of DoFs used in Ω and in $\Omega \cup \Omega^{\mathrm{pml}}$.

Table 1. Number of elements along the x and y axes and the resulting number of DoFs for the domain of interest (Ω), the absorbing layer (Ω^{pml}), and the whole computational domain ($\Omega \cup \Omega^{\mathrm{pml}}$). The number of elements and DoFs are given for interpolating functions of degree P = 1, P = 5, and P = 10. Note that the number of DoFs for the whole domain grows as the polynomial degree increases, because there are more DoFs in the absorbing layer

P	Ω			Ω^{pml}		$\Omega \cup \Omega^{pml}$
	N_e for x	N_e for y	N_{dof}	N_e for x	N_e for y	N_{dof}
1	100	100	10,201	3	3	11,449
5	20	20	10,201	3	3	17,161
10	10	10	10,201	3	3	25,921

Note that, because the number of elements used to produce the mesh for our PML is independent of *P*, we have significantly more DoFs for interpolating functions of degree 10 than for interpolating functions of degree 1. We could scale the number of elements in the PML with the polynomial degree (*P*), but such scaling is not required because three elements are sufficient to fully absorb the incoming wave.

3. Materials and Methods / Arbitrary Precision

3.1. Krylov Solvers

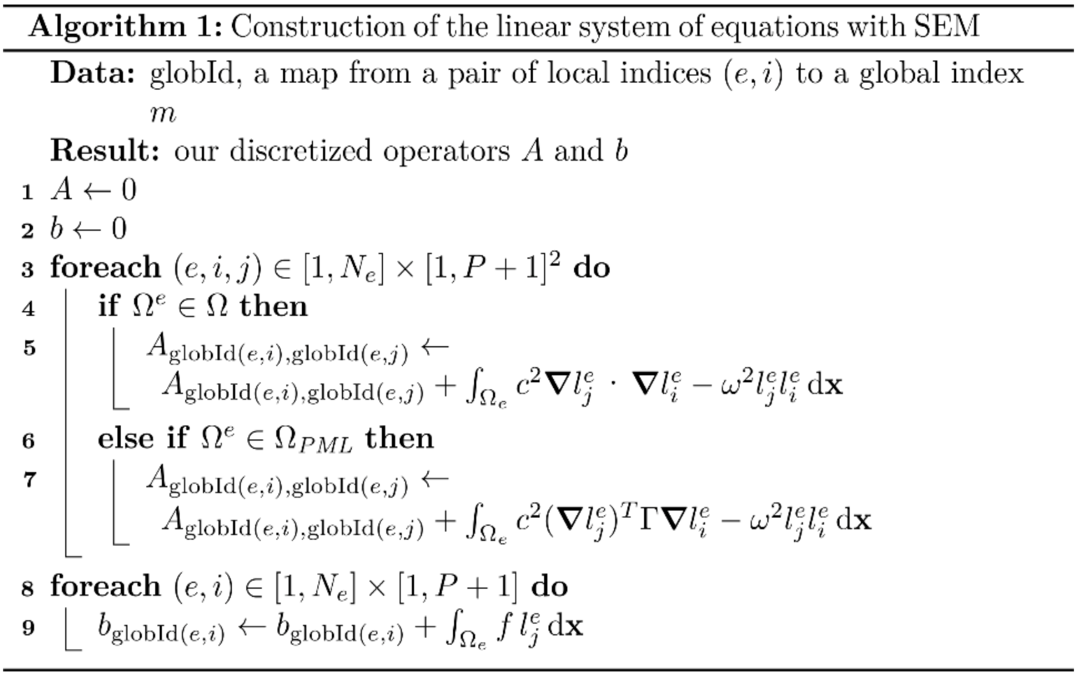
Discretizing the weak form of our wave Eq. (5) results in a linear system of equations:

$$Ax = b, \tag{9}$$

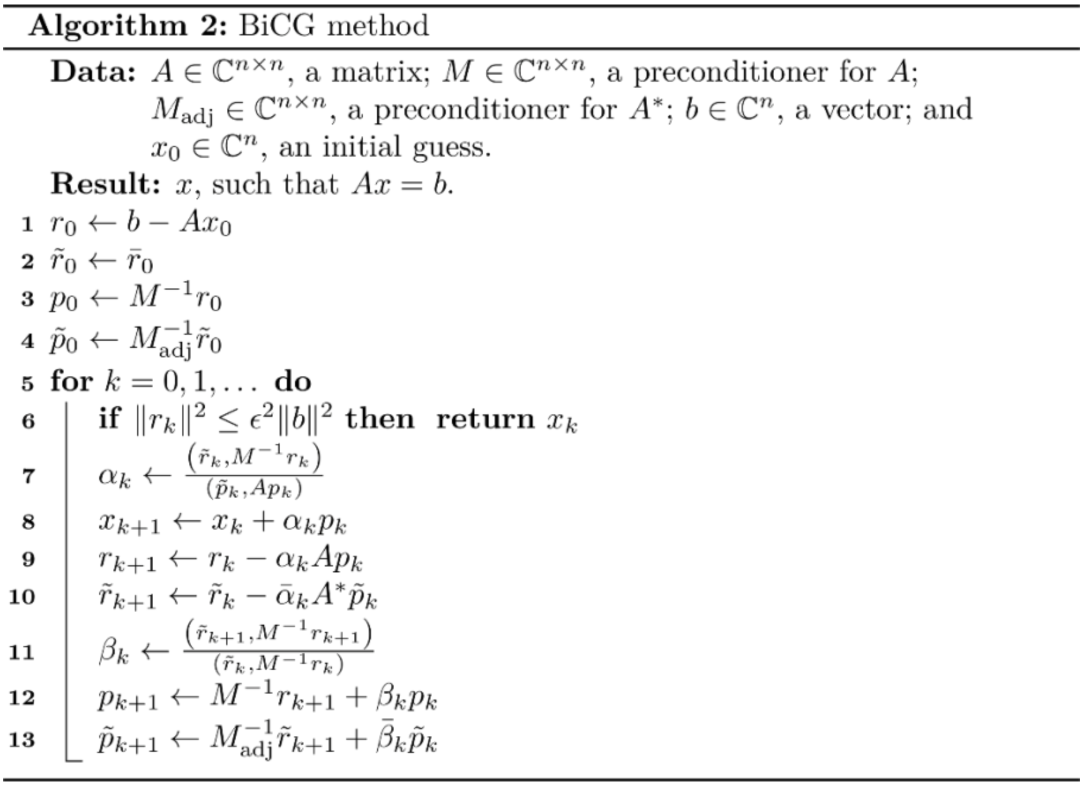
where *A* and *b* correspond to the discretized operators of Eq. (5) and where *x* is a vector that stores the value of the solution at the interpolating nodes. For more details concerning the construction of *A* and *b*, see Algorithm 1.

Note that our discretized operator is symmetric but non-Hermitian. While some attempts have been made to use sophisticated massively parallel direct solvers for Eq. (9) [20, 21], their memory requirements, as well as their lower scalability compared to iterative solvers and time-domain solvers, make them unfit for large-scale 3D problems. This is especially true for inverse problems such as Full Waveform Inversion (FWI) [22, 23], where the wave equation needs to be solved for various parameters and for, potentially, several thousands of Right Hand Sides (RHS). For such problems, solver efficiency is highly important.

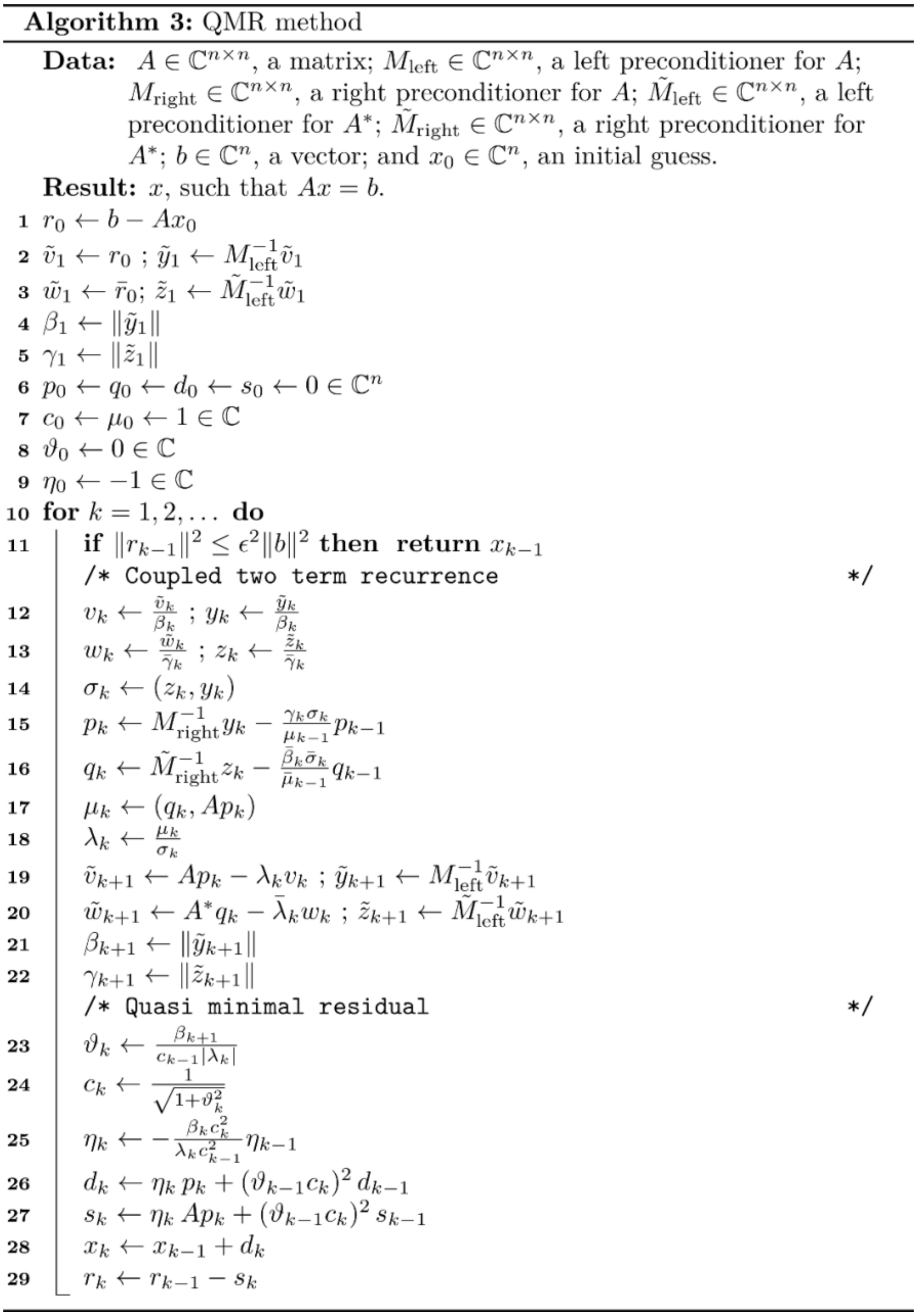
Methods such as BiConjugate Gradient (BiCG) [24] and Quasi Minimal Residual (QMR) [25] address both the memory and scalability issues of direct solvers. Indeed, both BiCG and QMR are based on short recurrences, and thus only require the storage of a small number of vectors. Moreover, they only perform Matrix-vector products using the unchanged, sparse, source matrices (see Algorithms 2 and 3 for more details). Additionally, if the same equation needs to be solved for different RHSs, several solvers can be run in parallel with no additional communication cost. It is worth mentioning that BiCG typically exhibits a very erratic convergence behavior and that there is no formal guarantee on its convergence. However, in this study, we observe that both BiCG and QMR converge to a solution after a similar number of iterations.



```
Algorithm 1: Construction of the linear system of equations with SEM
  Data: globId, a map from a pair of local indices $(e, i)$ to a global index $m$
  Result: our discretized operators $A$ and $b$
$A \leftarrow 0$
$b \leftarrow 0$
foreach $(e, i, j) \in [1, N_e] \times [1, P+1]^2$ do
    if $\Omega^e \in \Omega$ then
        $A_{\text{globId}(e,i),\text{globId}(e,j)} \leftarrow A_{\text{globId}(e,i),\text{globId}(e,j)} + \int_{\Omega_e} c^2 \nabla l_j^e \cdot \nabla l_i^e - \omega^2 l_j^e l_i^e \, \mathrm{d}\mathbf{x}$
    else if $\Omega^e \in \Omega_{PML}$ then
        $A_{\text{globId}(e,i),\text{globId}(e,j)} \leftarrow A_{\text{globId}(e,i),\text{globId}(e,j)} + \int_{\Omega_e} c^2 (\nabla l_j^e)^T \Gamma \nabla l_i^e - \omega^2 l_j^e l_i^e \, \mathrm{d}\mathbf{x}$
foreach $(e, i) \in [1, N_e] \times [1, P+1]$ do
    $b_{\text{globId}(e,i)} \leftarrow b_{\text{globId}(e,i)} + \int_{\Omega_e} f \, l_j^e \, \mathrm{d}\mathbf{x}$
```



```
Algorithm 2: BiCG method
  Data: $A \in \mathbb{C}^{n \times n}$, a matrix; $M \in \mathbb{C}^{n \times n}$, a preconditioner for $A$;
        $M_{\text{adj}} \in \mathbb{C}^{n \times n}$, a preconditioner for $A^*$; $b \in \mathbb{C}^n$, a vector; and
        $x_0 \in \mathbb{C}^n$, an initial guess.
  Result: $x$, such that $Ax = b$.
$r_0 \leftarrow b - Ax_0$
$\tilde{r}_0 \leftarrow \bar{r}_0$
$p_0 \leftarrow M^{-1} r_0$
$\tilde{p}_0 \leftarrow M_{\text{adj}}^{-1} \tilde{r}_0$
for $k = 0, 1, \ldots$ do
    if $\|r_k\|^2 \leq \epsilon^2 \|b\|^2$ then return $x_k$
    $\alpha_k \leftarrow \frac{(\tilde{r}_k, M^{-1} r_k)}{(\tilde{p}_k, A p_k)}$
    $x_{k+1} \leftarrow x_k + \alpha_k p_k$
    $r_{k+1} \leftarrow r_k - \alpha_k A p_k$
    $\tilde{r}_{k+1} \leftarrow \tilde{r}_k - \bar{\alpha}_k A^* \tilde{p}_k$
    $\beta_k \leftarrow \frac{(\tilde{r}_{k+1}, M^{-1} r_{k+1})}{(\tilde{r}_k, M^{-1} r_k)}$
    $p_{k+1} \leftarrow M^{-1} r_{k+1} + \beta_k p_k$
    $\tilde{p}_{k+1} \leftarrow M_{\text{adj}}^{-1} \tilde{r}_{k+1} + \bar{\beta}_k \tilde{p}_k$
```



Algorithm 3: QMR method

```
Data: $A \in \mathbb{C}^{n\times n}$, a matrix; $M_{\text{left}} \in \mathbb{C}^{n\times n}$, a left preconditioner for $A$;
      $M_{\text{right}} \in \mathbb{C}^{n\times n}$, a right preconditioner for $A$; $\tilde{M}_{\text{left}} \in \mathbb{C}^{n\times n}$, a left
      preconditioner for $A^*$; $\tilde{M}_{\text{right}} \in \mathbb{C}^{n\times n}$, a right preconditioner for
      $A^*$; $b \in \mathbb{C}^n$, a vector; and $x_0 \in \mathbb{C}^n$, an initial guess.
Result: $x$, such that $Ax = b$.
$r_0 \leftarrow b - Ax_0$
$\tilde{v}_1 \leftarrow r_0$ ; $\tilde{y}_1 \leftarrow M_{\text{left}}^{-1}\tilde{v}_1$
$\tilde{w}_1 \leftarrow \bar{r}_0$; $\tilde{z}_1 \leftarrow \tilde{M}_{\text{left}}^{-1}\tilde{w}_1$
$\beta_1 \leftarrow \|\tilde{y}_1\|$
$\gamma_1 \leftarrow \|\tilde{z}_1\|$
$p_0 \leftarrow q_0 \leftarrow d_0 \leftarrow s_0 \leftarrow 0 \in \mathbb{C}^n$
$c_0 \leftarrow \mu_0 \leftarrow 1 \in \mathbb{C}$
$\vartheta_0 \leftarrow 0 \in \mathbb{C}$
$\eta_0 \leftarrow -1 \in \mathbb{C}$
for $k = 1, 2, \ldots$ do
    if $\|r_{k-1}\|^2 \le \epsilon^2\|b\|^2$ then return $x_{k-1}$
    /* Coupled two term recurrence                                  */
    $v_k \leftarrow \frac{\tilde{v}_k}{\beta_k}$ ; $y_k \leftarrow \frac{\tilde{y}_k}{\beta_k}$
    $w_k \leftarrow \frac{\tilde{w}_k}{\bar{\gamma}_k}$ ; $z_k \leftarrow \frac{\tilde{z}_k}{\bar{\gamma}_k}$
    $\sigma_k \leftarrow (z_k, y_k)$
    $p_k \leftarrow M_{\text{right}}^{-1} y_k - \frac{\gamma_k \sigma_k}{\mu_{k-1}} p_{k-1}$
    $q_k \leftarrow \tilde{M}_{\text{right}}^{-1} z_k - \frac{\bar{\beta}_k \bar{\sigma}_k}{\bar{\mu}_{k-1}} q_{k-1}$
    $\mu_k \leftarrow (q_k, Ap_k)$
    $\lambda_k \leftarrow \frac{\mu_k}{\sigma_k}$
    $\tilde{v}_{k+1} \leftarrow Ap_k - \lambda_k v_k$ ; $\tilde{y}_{k+1} \leftarrow M_{\text{left}}^{-1}\tilde{v}_{k+1}$
    $\tilde{w}_{k+1} \leftarrow A^* q_k - \bar{\lambda}_k w_k$ ; $\tilde{z}_{k+1} \leftarrow \tilde{M}_{\text{left}}^{-1}\tilde{w}_{k+1}$
    $\beta_{k+1} \leftarrow \|\tilde{y}_{k+1}\|$
    $\gamma_{k+1} \leftarrow \|\tilde{z}_{k+1}\|$
    /* Quasi minimal residual                                       */
    $\vartheta_k \leftarrow \frac{\beta_{k+1}}{c_{k-1}|\lambda_k|}$
    $c_k \leftarrow \frac{1}{\sqrt{1+\vartheta_k^2}}$
    $\eta_k \leftarrow -\frac{\beta_k c_k^2}{\lambda_k c_{k-1}^2}\eta_{k-1}$
    $d_k \leftarrow \eta_k\, p_k + (\vartheta_{k-1} c_k)^2\, d_{k-1}$
    $s_k \leftarrow \eta_k\, Ap_k + (\vartheta_{k-1} c_k)^2\, s_{k-1}$
    $x_k \leftarrow x_{k-1} + d_k$
    $r_k \leftarrow r_{k-1} - s_k$
```

Both BiCG and QMR are based on Lanczos iteration. More precisely, after k iterations, both methods construct a set of bi-orthogonal vectors W_k and V_k. Then, at the k^{th} iteration, the k^{th} iterate satisfies the following relation:

$$x_k = x_0 + V_k z_k \tag{10}$$

and the corresponding residual can be expressed as:

$$r_k = V_{k+1}(\|r_0\| e_1 - \bar{T}_k z_k) \tag{11}$$

where $e_1 \in \mathbb{C}^{k+1}$ is the unit vector whose first component is one and $\bar{T}_k \in \mathbb{C}^{(k+1)\times k}$ is tridiagonal. Both BiCG and QMR attempt to find z_k as an approximate solution to the overdetermined system:

$$\bar{T}_k z = \|r_0\| e_1. \tag{12}$$

BiCG chooses z_k as the solution of the k first rows of Eq. (12), neglecting the last one, while QMR chooses z_k as the least-square solution to Eq. (12) [26]. QMR typically exhibits a much smoother convergence behavior than BiCG [25]. This characteristic motivates us to repeat our experiments with both methods, to compare the impact of precision in each case.

3.2. Stability Issues, and the Impact of Precision for BiCG and QMR

The Lanczos process underlying both BiCG and QMR allows for a lower memory requirement of $\mathcal{O}(n)$ scalars versus the $\mathcal{O}(n^2)$ needed by direct solvers and by the Generalized Minimal RESidual (GMRES) method. However, it may introduce a number of numerical instabilities, which may affect the convergence behavior of both methods. Both BiCG and QMR can encounter so called *near-breakdowns*, which happen when some quantities are inappropriately evaluated as null. The first near-breakdown happens when the underlying Lanczos iteration suffers from a near-breakdown. It happens when:

$$\begin{gathered}(\tilde{r}_k, M^{-1} r_k) \approx 0 \text{ in Algorithm 2} \\ \sigma_k \approx 0 \text{ in Algorithm 3}\end{gathered} \tag{13}$$

The second case of near-breakdown happens when a diagonal element of $\bar{T}_k$ becomes very small, or when roundoff creates a huge error on $\|x_k\|$, making it impossible for both methods to approximately solve [27]. This corresponds to:

$$\begin{gathered}(\tilde{p}_k, A p_k) \approx 0 \text{ in Algorithm 2} \\ \mu_k \approx 0 \text{ in Algorithm 3}\end{gathered} \tag{14}$$

Increasing the working precision (*i.e.* reducing the *epsilon machine* ε_m) makes it possible to represent these values, and to accurately perform operations involving them, even when they become extremely small, and thus it avoids these cases of *near-breakdown*.

Additionally, round-off errors degrade the bi-orthogonality of the set of vectors W_k and V_k computed by the Lanczos process underlying both BiCG and QMR. This phenomenon is exacerbated when dealing with a linear operator A with clustered eigenvalues [28, 29]. Eq. (10) illustrates how a loss of bi-orthogonality between W_k and V_k can affect the convergence behavior of both BiCG and QMR as it introduces errors in V_k. This in turn affects how well we can represent our estimated solution x_k. This can be explained by considering the equivalent asymmetric Lanczos process, using the correspondence explicitly pointed out in [30]. The analysis of Bai [31] establishes the reciprocal relation between the convergence of the Lanczos process and the loss of bi-orthogonality. Moreover, the bounds given by Bai suggests that, when dealing with an operator A with clustered eigenvalues, increasing the numerical precision can facilitate the convergence of the considered solvers [32].

Previous SEM studies have shown, on a small example, that the eigenvalues of operators tend to become clustered when discretized with higher order SEMs [14]. Computing the eigenvalues of our operators confirms this phenomenon (Fig. 5). We can thus expect a substantial improvement in the rate and quality of convergence when using extended precision, in our particular case.

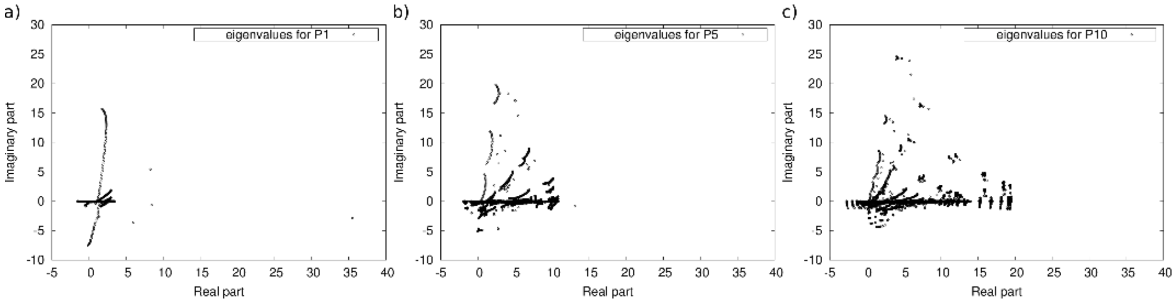


Fig. 5. Comparison of the eigenvalues of the wave equation discretized with a) FEM , b) SEM with fifth order polynomials, c) SEM with tenth order polynomials. The eigenvalues are displayed in the complex plane, with their real parts given by the x-coordinate and their imaginary parts by their y-coordinates.

3.3. Software Realization

Our software stack consists of four layers, as represented in Fig. 6, which mainly rely on three libraries. The two uppermost layers are problem specific and are written in C++.

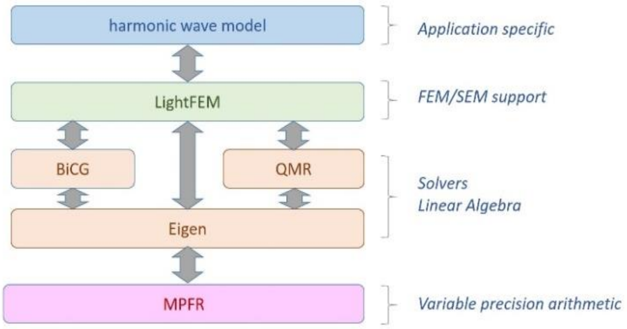


Fig. 6. Software layers involved in the C++ implementation.

The SEM part was handled by the LightFEM software library [33] which supports Two Dimensional (2D) SEM for arbitrary degree polynomials.

We implemented our own version of both BiCG and QMR. The linear algebra part was handled by the Eigen software library [34], which is a fully templated linear algebra software with support for dense and sparse linear algebra. The bottom layer relies on the GNU Multiple Precision Floating-Point Reliable (MPFR) software library [35] for the representation of arbitrary precision real numbers, and for the associated arithmetics. A wrapper for MPFR [36] was used in conjunction with Eigen, allowing us to perform all the linear algebra operations required by our solver at the desired precision.

MPFR is a widespread library written in C and based on the GNU Multi-precision Library. However, its execution speed is slow, like any other software emulation of extended precision, and this is the limiting factor in the size of our experiments. There exist very few emerging hardware accelerators that support variable width high-precision arithmetic, one example is our previous realization [37]. We expect to deal with this issue with our upcoming hardware accelerator Variable eXtended Precision (VXP) (formerly VRP) [32] when the next silicon prototype is available.

4. Experimental Results

In this section, we demonstrate the impact of increased numerical precision on both BiCG and QMR methods. We evaluate two aspects of the question:

1. The impact of numerical precision on the convergence behavior of both methods.
2. The impact on the *accuracy* of the estimated residual r_k (Algorithms 2 and 3) compared to the actual residuals $b - Ax_k$.

We modelled the propagation of a wave with a speed of 1 m/s, and a frequency of 20 Hz (40π rad/s) within a (−0.5 m, 0.5 m) box. We used a Dirac impulse as our source term. We discretized the wave equation with FEM and with SEM with polynomials of order 5 (referred to as P5) and 10 (referred to as P10). We then solved the resulting systems with both BiCG and QMR with a numerical precision for the mantissa of 53 bits (equivalent to double precision), 203, 353 and 503 bits. A simple Jacobi left-preconditioner was used for both BiCG and QMR.

Fig. 7 illustrates the impact of increased numerical precision on the convergence behavior of BiCG.

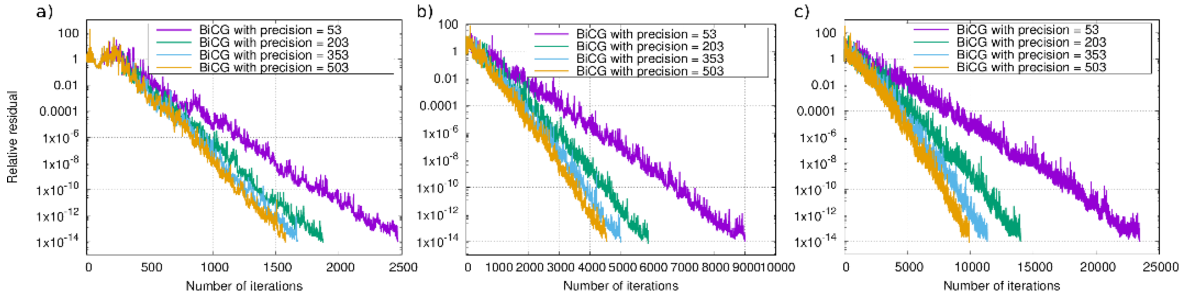


Fig. 7. Comparison of the convergence behavior of BiCG with precisions 53, 203, 353 and 503 bits. The system solved is the wave equation discretized with: a) FEM b) P5 SEM c) P10 SEM.

For each of the discretization schemes, we observe an improvement in the convergence rate of BiCG as the numerical precision increases. Additionally, we notice the effectiveness of increasing the precision increases with the order of the method. More precisely, for FEM, we observe a 1.56× improvement of convergence with a precision 503 bits, a 1.99× improvement with P5 SEM and a 2.36× improvement with P10 SEM (Table 2 for the exact number of iterations).

Table 2. Number of iterations of both BiCG and QMR required for convergence when solving the wave equation discretized with FEM, P5 SEM and P10 SEM with different numerical precision. The norm of the residual produced by QMR with precision 53 bits stagnates around $1e^{-14}$ and does not satisfy its convergence conditions within the $2N_{dof}$ iterations, which is set as the maximum number of iterations permitted. N/C indicates the algorithm has not converged

Prec (bits)	FEM		P5 SEM		P10 SEM	
	BiCG	QMR	BiCG	QMR	BiCG	QMR
53	2,467	N/C	9,006	N/C	23,437	N/C
203	1,876	1,843	5,868	5,867	13,968	13,944
353	1,671	1,666	4,982	4,999	11,341	11,137
503	1,575	1,569	4,535	4,536	9,928	9,860

We then carry out the same experiment with QMR. The results are shown in Fig. 8.

First, we see that, with a precision of 53 bits, and for all discretization schemes, the residual computed by QMR stagnates around $1e^{-14}$, which is consistent with the convergence behavior found by Freund & Nachtigal [25]. We do not see such stagnation with higher precision. However, it is likely that increasing numerical precision delays the stagnation rather than suppresses it. The stagnation of the residual for a precision of 53 bits makes it hard to properly evaluate the speed-up obtained by increasing the numerical precision. However, for all other levels of precision, QMR converges roughly as fast as BiCG (Table 2). We can thus expect the same order of speed-up for the two methods, as long as the residual computed by QMR does not stagnate.

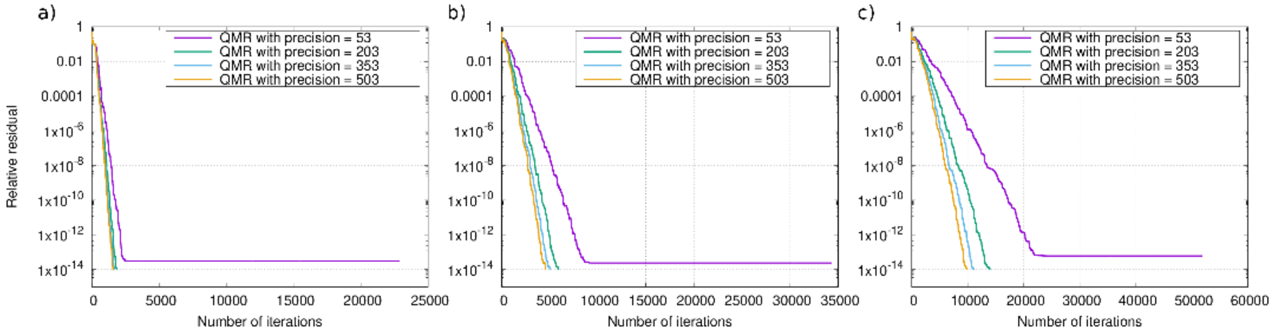


Fig. 8. Comparison of the convergence behavior of QMR with precision 53, 203, 353, or 503 bits. The system solved is the wave equation discretized with: a) FEM b) P5 SEM c) P10 SEM.

As mentioned in the previous section, having found similar convergence behavior for QMR and BiCG, we conclude, with a reasonable degree of certainty, that the speed-up obtained on BiCG was not accidental and was solely due to VP.

These experiments reinforce our hypotheses that: 1) clustered eigenvalues tend to slow the Lanczos process underlying both BiCG and QMR, thus slowing down the convergence of the methods and 2) increasing the numerical precision helps mitigate this phenomenon.

Let us now study how increasing the numerical precision impacts the accuracy of the estimated residual. An accurate estimation of the residual is essential to ensure the accuracy of the Krylov methods, as the norm of the estimated residual is used in their convergence criterion. Overestimating the norm of the residual may delay the convergence of the Krylov method. Underestimating the norm of the residual may lead to the algorithm stopping before the approximate solution x_k has attained the desired precision. We show in Table 3 the ratio between the actual residual and the estimated residual for all discretizations, iterative solvers, and values of precision.

Table 3. Ratio between the norm of the actual residual, $b{-}Ax_k$, and the estimated residual r_k for both BiCG and QMR. Both BiCG and QMR were applied on the wave equation discretized with FEM, P5 SEM, and P10 SEM with different numerical precision

Prec (bits)	FEM		P5 SEM		P10 SEM	
	BiCG	QMR	BiCG	QMR	BiCG	QMR
53	19.41	6.16	10.46	5.01	66.00	10.80
203	1.00	1.00	1.00	1.00	1.00	1.00
353	1.00	1.00	1.00	1.00	1.00	1.00
503	1.00	1.00	1.00	1.00	1.00	1.00

We note that, at precisions of 203 bits and higher, the estimated residual becomes indistinguishable from the actual residual. This means that, at least for this problem, a drastic increase in the numerical precision is not needed in order to ensure an accurate estimated solution. Secondly, we note that, at a precision of 53 bits, QMR tends to have a much better estimation of the actual residual than BiCG. This might be due to the coupled two term recurrence in Algorithm 3, which has been noted to improve the numerical stability of the algorithm 14. This better estimation of the actual residual by QMR might also explain why BiCG was able to reach the convergence criterion while QMR was not, since the norm of the residual estimated by BiCG is less than a tenth of the value of the norm of the actual residual.

5. Conclusions and Perspectives

This study demonstrates the benefit of increasing numerical precision for the resolution of a common yet problematic PDE. Our case study focuses on the harmonic wave equation combined with a PML which leads to bilinear forms with multiple difficulties: complex-values, indefiniteness, and non-hermiticity. The equation was discretized using a high-order SEM, which is known to lead to linear systems that are difficult to solve because of the dense, poorly-conditioned operators with clustered eigenvalues.

The effect of increasing the numerical precision has been studied in the case of real, positive, symmetric matrices. Our current work shows similar benefits for non-hermitian, indefinite matrices. More precisely, we obtain a reduction in the number of iterations required for the convergence of BiCG. We find the number of iterations is reduced by factors of 0.5 and 0.4 for the systems discretized with SEM, using fifth order and tenth order polynomials respectively; and by a factor of 0.6 for the system discretized with FEM (Table 2). Additionally, we observe a better approximation of the estimated error, especially when the system is discretized with high order SEM, where the actual residuals are 66 times larger than the estimated residual for BiCG and ~11 times larger for QMR with 53 bit precision (Table 2). We note that the estimated residual quickly converges to the actual residual when we increase the numerical precision. This leads us to believe that increasing the numerical precision is essential to speed up the convergence of Krylov solvers and to ensure the correct error criterion has been met upon convergence.

Our experiments have been run with software emulated high precision, which has limited run-time performance. We expect to use this example to validate our in-house developed hardware VXP accelerator as soon as it is available.

Conflict of Interest

The authors declare no conflict of interest

Author Contributions

Alexandre Hoffmann and Yves Durand conducted the research, Alexandre Hoffmann and Jérôme Fereyre developed the software, Alexandre Hoffmann and Yves Durand analyzed the results. All authors wrote the paper and all authors had approved the final version.

Funding

This work was partially funded by the French Agence nationale de la recherche (ANR) for project “Improving Predictability of Numerical Computations (ImPreNum)” under ref. ANR-18-CE46-0011, and by the European Union’s Horizon 2020 research and innovation programme for project “European Processor Initiative (EPI)” under grant agreement No. 826 647.

References

[1] Sirgue, L., Etgen, J. T., & Albertin, U. (2008). 3D frequency domain waveform inversion using time domain finite difference methods. *Proceedings of the 70th EAGE Conference and Exhibition Incorporating SPE EUROPEC 2008*.

[2] Sirgue, L., Barkved, O. I., Dellinger, J., Etgen, J., Albertin, U., & Kommedal, J. H. (2010). Thematic set: Full waveform inversion: The next leap forward in imaging at Valhall. *First Break*, 28(4).

[3] Hu, G., Etienne, V., Castellanos, C., Operto, S., Brossier, R., & Virieux, J. (2012). Assessment of 3D acoustic isotropic full waveform inversion of wide-azimuth OBC data from Valhall. *Proceedings of the SEG International Exposition and Annual Meeting*.

[4] Warner, M., Ratcliffe, A., Nangoo, T., Morgan, J., Umpleby, A., Shah, N., Vinje, V., Štekl, I., Guasch, L., Win, C., Conroy, G., & Bertrand, A. (2013). Anisotropic 3d full-waveform inversion. *Geophysics*, *78(2)*, R59–R80.

[5] Kamath, N., Brossier, R., Métivier, L., Pladys, A., & Yang, P. (2021). Multiparameter full-waveform inversion of 3d ocean-bottom cable data from the valhall Field. *Geophysics, 86(1)*, B15–B35.

[6] Lucka, F., Pérez-Liva, M., Treeby, B. E., & Cox., B. T. (2021). High resolution 3d ultrasonic breast imaging by time-domain full waveform inversion. *Inverse Problems*, *38(2)*, 025008.

[7] Huot, F., Chen, Y.-F., Clapp, R., Boneti, C., & Anderson J. (2019). High-resolution imaging on TPUs. arXiv preprint, arXiv:1912.08063.

[8] Berenger, J.-P. (1994). A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics, 114(2)*, 185–200.

[9] Hastings, F. D., Schneider, J. B., & Broschat, S. L. (1996). Application of the Perfectly Matched Layer (PML) absorbing boundary condition to elastic wave propagation. *The Journal of the Acoustical Society of America, 100(5)*, 3061–3069.

[10] Ernst, O. G., & Gander, M. J. (2012). Why it is difficult to solve Helmholtz problems with classical iterative methods. *Numerical Analysis of Multiscale Problems*, 325–363.

[11] Bermudez, A., Hervella-Nieto, L., Prieto, A., & Rodriguez, R. (2006). An optimal finite-element/pml method for the simulation of acoustic wave propagation phenomena. *Variational Formulations in Mechanics: Theory and Applications, 01*.

[12] Pasquetti, R., & Rapetti, F. (2004). Spectral element methods on triangles and quadrilaterals: Comparisons and applications. *Journal of Computational Physics, 198*.

[13] Komatitsch, D., & Vilotte. J.-P. (1998). The spectral element method: An efficient tool to simulate the seismic response of 2D and 3D geological structures. *Bulletin of the Seismological Society of America*, *88(2)*, 368–392.

[14] Deville, M. O., Fischer, P. F., & Mund, E. H. (2002). *High-Order Methods for Incompressible Fluid Flow*. Cambridge University Press, Cambridge.

[15] Marfurt, K. J. (1984). Seismic modeling: A frequency-domain/finite-element approach. *SEG Technical Program Expanded Abstracts*, 633–634.

[16] Eisenträger, S., Atroshchenko, E., & Makvandi, R. (2019). On the condition number of high order finite element methods: Influence of p-refinement and mesh distortion. *Computers & Mathematics with Applications, 80(11)*, 2289–2339.

[17] Seriani, G., & Priolo, E. (1994). Spectral element method for acoustic wave simulation in heterogeneous media. *Finite Elements in Analysis and Design*, *16(3)*, 337–348.

[18] Faccioli, E., Maggio, F., Paolucci, R., & Quarteroni, A. (1997). 2D and 3d elastic wave propagation by a pseudo-spectral domain decomposition method. *Journal of Seismology, 1(3)*, 237–251.

[19] Komatitsch, D., & Tromp, J. (1999). Introduction to the spectral element method for three-dimensional seismic wave propagation. *Geophysical Journal International*, *139(3)*, 806–822.

[20] Operto, S., Miniussi, A., Brossier, R., Combe, L., Haller, N., Kjos, E., Métivier, L., Milne, R., Ribodetti, A., Song, Z., Virieux, J., & Zheng, Y. (2015). Efficient 3d frequency-domain full-waveform inversion of ocean-bottom cable data-application to Valhall in the Visco-ac. *Proceedings of the 77th EAGE Conference and Exhibition 2015. European Association of Geoscientists & Engineers*.

[21] Aghamiry, H. S., Gholami, A., Combe, L., & Operto, S. (2022). Accurate 3d frequency-domain seismic wave modeling with the wavelength-adaptive 27-point finite-difference stencil: A tool for full-waveform inversion. *Geophysics, 87(3)*, R305–R324.

[22] Lailly, P. (1983). The seismic problem as a sequence of before-stack migrations. *Proceedings of the Conference on Inverse Scattering: Theory and Applications*.

[23] Tarantola, A. (1984). Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, *49(8)*, 1259–1266.

[24] Fletcher, R. (1976). Conjugate gradient methods for indefinite systems. *Proceedings of the Dundee Conference on Numerical Analysis* (pp. 73–89).

[25] Freund, R. W., & Nachtigal, N. M. (1991). QMR: A quasi-minimal residual method for non-hermitian linear systems. *Numerische Mathematik, 60(1)*, 315–339.

[26] Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems,* 2nd ed. Society for Industrial and Applied Mathematics.

[27] Greenbaum, A. (1997). Estimating the attainable accuracy of recursively computed residual methods. *SIAM Journal on Matrix Analysis and Applications, 18*, 535–551.

[28] Ye, Q. (1991). A convergence analysis for nonsymmetric Lanczos algorithms. *Mathematics of Computation*, *56*, 677–691.

[29] Zemke, J.-P. M. (2001). How orthogonality is lost in Krylov methods. In *Symbolic Algebraic Methods and Verification Methods* (pp. 255–266). Vienna: Springer.

[30] Tichý, P., & Zítko, J. (1998). Derivation of BICG from the conditions defining Lanczos' method for solving a system of linear equations. *Applications of Mathematics*, *43*, 381–388.

[31] Bai, Z. (1994). Error analysis of the Lanczos algorithm for the nonsymmetric eigenvalue problem. *Mathematics of Computation, 62*, 209–226.

[32] Durand, Y., Guthmuller, E., Fuguet, C., Fereyre, J., Bocco, A., & Alidori, R. (2022). Accelerating variants of the conjugate gradient with the variable precision processor. *Proceedings of the 2022 IEEE 29th Symposium on Computer Arithmetic (ARITH)* (pp. 51–57).

[33] Hoffmann, A. (2022). Lightfem. Retrieved from https://github.com/alexandrehoffmann/LightFEM

[34] Guennebaud, G., Jacob, B., *et al*. (2010). Eigen v3. Retrieved from http://eigen.tuxfamily.org

[35] Fousse, L., Hanrot, G., Lefèvre, V., Pélissier, P., & Zimmermann, P. (2007). MPFR: A multiple-precision binary floating-point library with correct rounding. *ACM Trans. Math. Softw., 33(2)*, 13.

[36] Schneider, C. (2012). MPFR: Real (v0.0.9-alpha). Retrieved from http://chschneider.eu/programming/mpfr_real/

[37] Bocco, A., Durand, Y., & Dinechin, F. (2019). Smurf: Scalar multiple-precision unum Risc-v floating-point accelerator for scientific computing. *Proceedings of the Conference for Next Generation Arithmetic 2019, CoNGA'19*.