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.

Finite Element Computation of Semiconductor Quantum States: Convergence, Sparse Eigensolvers, and Verification

Wellington Fonseca

fonseca@ufpa.br

Universidade Federal do Pará

Research Article

Keywords:

Posted Date: November 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11570793/v1

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

Additional Declarations: No competing interests reported.

Finite Element Computation of Semiconductor Quantum States: Convergence, Sparse Eigensolvers, and Verification

Wellington da Silva Fonseca ORCID [1*]

[1*]Instituto de Tecnologia, Universidade Federal do Pará, Av. Perimetral, Belém, 66075-110, Pará, Brazil.

Corresponding author(s). E-mail(s): fonseca@ufpa.br;

Abstract

This work presents and analyzes a finite element framework for the stationary Schrödinger eigenvalue problem in one to three spatial dimensions. The weak formulation yields a generalized symmetric eigenproblem $\boldsymbol{A\psi = EM\psi}$ with $\boldsymbol{A = K + M_V}$. Where the discrete form is the exact restriction of the continuous one, the conforming approximation produces eigenvalues that are upper bounds for the exact ones, and under standard regularity assumptions the eigenvalue error converges as $\boldsymbol{\mathcal{O}(h^{2s})}$; the nodal interpolation of the potential used here preserves the bound for convex $\boldsymbol{V}$ on the tensor-product meshes considered, and is verified numerically otherwise. The usual diagonal-shift treatment of Dirichlet data introduces spurious eigenvalues, whereas eliminating the constrained degrees of freedom reproduces the constrained spectrum exactly. Three eigensolution strategies are considered: dense factorization and sparse shift-and-invert Arnoldi iteration are compared on identical matrices, and the serial constraint of the in-framework solver is stated, together with the export route that lifts it. The sparse route is up to $\mathbf{4097}\times$ faster, replaces the $\boldsymbol{\mathcal{O}(n_f^2)}$ dense storage by sparse matrices whose memory is dominated by the factorization fill-in and the Arnoldi basis rather than by the matrices themselves, and makes a problem with $\mathbf{3.96 \times 10^4}$ degrees of freedom (infeasible in dense arithmetic at **51.2** GB) solvable in **0.41** s. Position-dependent mass is supported in the conservative form $\boldsymbol{-\frac{\hbar^2}{2}\nabla\cdot[(1/m^*)\nabla\psi]}$; on an InGaN/GaN well, where the mass contrast is **6.3%**, the levels move by **3.0%**. Verification is fivefold: seven analytic benchmarks, a code-to-code comparison agreeing to $\mathbf{4 \times 10^{-13}}$, measured convergence orders against the $\boldsymbol{\mathcal{O}(h^2)}$ prediction, structural invariants, and the reproduction of two published quantum-confinement benchmarks.

Keywords: Schrödinger equation · Finite element method · Generalized eigenvalue problem · Sparse eigensolvers · Shift-and-invert Arnoldi · Code verification

1 Introduction

Mathematically, quantum mechanics is an eigenvalue problem. One computes the eigenvalues and eigenvectors of Hamiltonians, and the ground state and the first few excited states already reveal much about the physics of a system [1]. The stationary Schrödinger equation is the canonical instance [2, 3]. In dimensionless form it reads

$$-\nabla^2\psi(\mathbf{r}) + V(\mathbf{r})\,\psi(\mathbf{r}) = E\,\psi(\mathbf{r}), \qquad \mathbf{r} \in \Omega \subset \mathbb{R}^d, \quad d = 1, 2, 3, \tag{1}$$

where $\psi = \psi(\mathbf{r})$ is the wavefunction, $V = V(\mathbf{r})$ the confining potential, E the energy of the state and Ω the domain, with $\partial\Omega$ its boundary. Equation (1) is subject to boundary conditions on $\partial\Omega$.

Closed-form solutions exist for only a few separable potentials (the infinite well, the harmonic oscillator, the Coulomb problem), and problems of practical interest require numerical treatment. In semiconductor physics the relevant structures are quantum wells, wires and dots, heterostructures and resonant tunneling diodes, where quantum confinement modifies the electronic structure and hence the optical and transport properties [4, 5]. In these applications the equation is usually written in the effective-mass form

$$-\frac{\hbar^2}{2m^*}\nabla^2\psi + V\psi = E\psi, \tag{2}$$

where $\hbar$ is the reduced Planck constant and m^* the carrier effective mass, both absorbed into the unit coefficient of (1). The relation between this form and (1) is made explicit in Section 2.2, because it controls the interpretation of all reported eigenvalues. Equation (2) presumes a *constant* m^*. Across a heterointerface the operator is not this one but the conservative form $-\frac{\hbar^2}{2}\nabla\cdot[(1/m^*)\nabla\psi]$, which keeps the flux $(1/m^*)\partial_n\psi$ continuous; the distinction and its numerical consequences are taken up in Section 8.4.

Two families of numerical methods dominate the literature. Finite difference discretizations are simple and yield banded or sparse matrices, but represent curved or irregular geometries poorly. Finite element methods handle complex geometries and unstructured meshes naturally, and their convergence theory for eigenvalue problems is well established [6–8]. Discontinuous Galerkin variants have been developed specifically for the Schrödinger equation in device contexts, including multiscale and WKB-based formulations of the stationary Schrödinger–Poisson problem [9, 10]. Finite elements have also been carried into regimes where the kinetic operator is no longer local, a fractional Laplacian replacing $-\nabla^2$ [11]. Finite element approaches to quantum systems have a long history, from low-dimensional models and quantum dots [1, 5] to quantum transport in nanoscale devices [3, 12], where the eigenproblem is typically coupled to a Poisson equation. Device-oriented codes follow the same route: three-dimensional drift-diffusion and ballistic solvers [13], quantum-electronic frameworks built around

a Schrödinger–Poisson core [14], two-dimensional solvers with open boundary conditions for CMOS geometries [15], and spectral element discretizations of the coupled system [16]. What these have in common is that the Schrödinger solve is embedded in a larger device model. The eigenvalue component is therefore exercised as part of a workflow, and the numerical choices it forces, from discretization of the confinement potential, treatment of the constrained degrees of freedom, choice of eigensolver, are usually reported only as settings rather than examined on their own. That is the gap this work addresses, and it is a gap in the reporting of these choices, not a claim that they are absent from the literature. The wider context of this work is the multiphysics simulation of power apparatus, in which coupled electromagnetic, thermal and structural fields are already solved within a single finite element model [17], and of magnetohydrodynamic flows, where the electromagnetic and fluid fields are coupled in one formulation [18]. A quantum-mechanical eigenproblem solver extends that class of models to problems in which carrier confinement must be represented.

Quantum-well structures with an applied field are the setting in which these solvers are most often exercised: intersubband absorption in symmetric InGaN/GaN double wells [19] and self-consistent III-nitride quantum cascade structures [20] both require the confined levels and their field dependence, and both treat the well profile as piecewise constant with a finite barrier.

A finite element discretization of (1) with n unknowns produces sparse matrices, yet many implementations, including the one this work started from, assemble them densely and call a dense LAPACK driver. That is adequate for a few thousand unknowns and hopeless beyond: storage grows as $\mathcal{O}(n^2)$ and time as $\mathcal{O}(n^3)$. Usually, however, only the lowest few eigenpairs are of physical interest, and that is precisely the regime in which iterative Krylov methods with shift-and-invert spectral transformation are effective [21, 22]. Their inner solves rely on sparse direct factorizations [23], whose availability in parallel determines whether the scheme can be distributed. The choice between dense, sparse and distributed solution is therefore not a detail of implementation but a modeling decision, and it is treated as such.

The cross-checking against an independently written implementation is well established in the verification and validation literature [24, 25]. When exact solutions are unavailable, the method of manufactured solutions [26, 27] provides a complementary route. Such cross-checks do appear in the computational physics literature. That route is adopted here, in the code-to-code comparison of Section 7.4.

Contributions

The contributions are the following.

1. A dimension-generic finite element formulation for equation (1) in one, two and three dimensions, supporting arbitrary scalar potentials in one dimension and separable potentials, $V = \sum_k V_k(x_k)$, in higher dimensions (Section 6.3), implemented as a module of the open-source finite element package Elmer [29]. The author is not aware of a previously documented module for this equation in that framework; no claim is made about other packages, nor about priority over unpublished work.

2. Two structural results on the discrete spectrum (Section 4): the conforming eigenvalues are upper bounds for the exact ones, and the eigenvalue error obeys the classical $\mathcal{O}(h^{2s})$ estimate. The predicted order is verified numerically.
3. A quantified comparison of the dense and sparse eigensolution strategies on identical matrices, with a cost model that separates assembly, factorization, iteration and solver overhead, and a measured range over which the sparse route extends the accessible problem size (Sections 5 and 8.3).
4. A five-part verification programme (Section 7) combining analytic benchmarks, code-to-code comparison of two independent implementations, structural consistency checks, convergence-order confirmation, and reproduction of two quantum-confinement benchmarks from the literature (Section 8.4).
5. Documentation of a failure mode that is easy to encounter and hard to diagnose: a tensor-product assembly error that is invisible in one dimension (Remark 2).

The remainder of the paper is organized as follows. Section 2 states the problem, derives the weak form and discusses the unit convention. Section 3 describes the discretization and the treatment of Dirichlet data. Section 4 establishes the properties of the discrete spectrum. Section 5 analyzes the eigensolution strategies. Section 6 describes the two implementations. Section 7 presents the verification program and Section 8 the numerical experiments. Section 10 concludes.

2 Problem setting and notation

2.1 Strong form

Let $\Omega \subset \mathbb{R}^d$ be a bounded, open, connected domain with Lipschitz boundary. A pair (ψ, E) is sought, with $\psi \neq 0$, satisfying the stationary Schrödinger eigenvalue problem [2, 3]:

$$-\nabla^2\psi + V\psi = E\psi \quad \text{in } \Omega, \qquad \psi = 0 \quad \text{on } \partial\Omega. \tag{3}$$

Three problems are conflated in much of the literature, and only the first is the one the discrete equations actually solve.

1. **The bounded-domain problem.** Ω is the box of (3) and the Dirichlet condition is part of the statement. This is the problem the finite element method approximates, and for it $H_0^1(\Omega)$ embeds compactly into $L^2(\Omega)$, so the spectrum is discrete and bounded below once the form is closed and bounded below. Compactness alone does not give that, which is why the formulation assumes throughout that V is real, measurable and in $L^\infty(\Omega)$: under that hypothesis the form domain is $H_0^1(\Omega)$ and the operator is self-adjoint and bounded below, and discreteness follows from compactness. Potentials outside L^∞ are not covered by anything in this paper.
2. **The unbounded problem.** $\Omega = \mathbb{R}^d$ and discreteness must be earned by the potential: it is assumed throughout that V is confining, $V(\mathbf{r}) \to \infty$ as $|\mathbf{r}| \to \infty$, so that the resolvent is again compact and $E_1 \leq E_2 \leq \cdots \to \infty$ with finite multiplicities. This is the physical reference problem, and the analytic values quoted in Table 1 belong to it.
3. **The truncated and discretized problem.** What is computed: problem 1 on a mesh of size h in a box of side L.

The passage from 2 to 1 replaces the unbounded domain by a box and carries a *truncation* error, controlled by L. Discretizing that bounded problem on a mesh of size h carries a *discretization* error. The two are of a different nature, and Section 8.2 separates them numerically: there the mesh is held fixed while L varies, and vice versa. Where a case has a closed-form solution on the half-line rather than in a box, the Morse potential being the example here, the comparison carries the truncation error as well, and it is quoted with that caveat.

2.2 Units and the effective-mass interpretation

Equation (3) places a unit coefficient on the Laplacian. That choice controls how the results are to be interpreted, so its physical content is made explicit.

Equations (2) and (3) differ only by the coefficient $\hbar^2/2m^*$ multiplying the Laplacian [3, 4]. Let L be a characteristic length of the structure [4] and set

$$\mathcal{E} = \frac{\hbar^2}{2m^* L^2}, \qquad \mathbf{r} = L\,\tilde{\mathbf{r}}, \tag{4}$$

so that $\mathcal{E}$ is the energy unit associated with L and $\tilde{\mathbf{r}}$ is the position vector in units of L. The calligraphic symbol is used for the unit deliberately, reserving E_0 for the ground-state energy and E_n for the n-th eigenvalue in increasing order, as in Appendix A. Dividing (2) by $\mathcal{E}$ and using $\nabla^2 = L^{-2}\tilde{\nabla}^2$ gives, after dropping tildes,

$$-\tilde{\nabla}^2\psi + \frac{V}{\mathcal{E}}\psi = \frac{E}{\mathcal{E}}\psi, \tag{5}$$

which is (3) with $\tilde{V} = V/\mathcal{E}$ and $\tilde{E} = E/\mathcal{E}$.

Remark 1 (Physical scaling) Equation (3) is the effective-mass equation in units where $\hbar^2/2m^* = 1$. Eigenvalues reported here are dimensionless and become physical energies upon multiplication by $\mathcal{E}$. For GaAs, where $m^* \approx 0.067\,m_e$, and a nominal length $L = 10\,\mathrm{nm}$, one finds $\mathcal{E} \approx 5.7\,\mathrm{meV}$. Hence the two lowest levels of the harmonic potential $V = x^2$ in (3), namely $E_1 = 1$ and $E_2 = 3$, correspond to 5.7 and 17.1 meV at that length scale. Quadratic confinement of this form is the standard model for a parabolic quantum well [4, 30], and shifts of a few meV in the confined levels are precisely what controls intersubband optical transitions in such structures, as well as the subband structure of modulation-doped wells [31].

Consequently, when the conventional form $-\frac{1}{2}\nabla^2\psi + \frac{1}{2}\omega^2 r^2\psi = E\psi$ (units $\hbar = m = 1$) is used, multiplying by two gives $-\nabla^2\psi + \omega^2 r^2\psi = 2E\psi$; comparison with (3) shows that a potential $V = \alpha r^2$ corresponds to $\omega = \sqrt{\alpha}$, and that the conventional energies are half of those reported here. All analytic references are stated in the convention of (3) (Appendix A) to avoid ambiguity.

2.3 Weak form and variational characterization

Let $H_0^1(\Omega) = \{v \in H^1(\Omega) : v|_{\partial\Omega} = 0\}$. Multiplying (3) by $v \in H_0^1(\Omega)$ and integrating by parts [6, 8],

$$\underbrace{\int_\Omega \nabla\psi \cdot \nabla v \, \mathrm{d}\Omega}_{a(\psi,v)} + \underbrace{\int_\Omega V\psi v \, \mathrm{d}\Omega}_{b_V(\psi,v)} = E \underbrace{\int_\Omega \psi v \, \mathrm{d}\Omega}_{m(\psi,v)}. \tag{6}$$

The forms a, b_V and m are symmetric, and m is positive definite on $H_0^1(\Omega)$. The form a is H_0^1-elliptic, but b_V is *not* elliptic when V takes negative values, as it does in several of the benchmark cases below ($V_{\min} = -10$ for Morse and $-1/2$ for the sinusoidal potential). What is required is a hypothesis on the full energy form

$$q(u,v) = a(u,v) + b_V(u,v) = \int_\Omega \big(\nabla u \cdot \nabla v + V\, uv\big) \, \mathrm{d}\Omega. \tag{7}$$

It is assumed throughout that $V \in L^\infty(\Omega)$ is bounded below, so that b_V is bounded and q is bounded below on $H_0^1(\Omega)$. By the theory of closed semibounded forms [2, 8], the form $q + c\,m$ is continuous and coercive for c large enough, so it defines a unique self-adjoint operator H with $\langle Hv, v\rangle = q(v,v)$ on its domain; the shift by c is a bounded perturbation and does not affect discreteness. Because Ω is bounded, $H_0^1(\Omega)$ embeds compactly into $L^2(\Omega)$, so the resolvent of H is compact. Its spectrum is therefore discrete and bounded below, and consists of an increasing sequence $E_1 \le E_2 \le \cdots \to \infty$ of finite multiplicity. *The eigenvalues need not be positive*: for the Morse potential all four reported levels are negative (Table 6). The ordering is by value, not by sign. The min–max characterization below uses the energy form q of (7) rather than the stiffness alone, because a is positive definite but b_V is not when V changes sign. Defining the Rayleigh quotient [2]

$$R(v) = \frac{a(v,v) + b_V(v,v)}{m(v,v)}, \qquad v \in H_0^1(\Omega) \setminus \{0\}, \tag{8}$$

the eigenvalues admit the min–max characterization [2, 7, 8]

$$E_n = \min_{\substack{S_n \subset H_0^1(\Omega) \\ \dim S_n = n}} \; \max_{0 \neq v \in S_n} R(v). \tag{9}$$

This characterization is the basis of the results in Section 4.

3 Finite element discretization

3.1 Multilinear elements and quadrature

Let $\mathcal{T}_h$ be a conforming partition of Ω into d-linear elements (intervals, quadrilaterals or hexahedra), with mesh size h. Let $\tau_e \in \mathcal{T}_h$ denote a generic element. The discrete space is

$$V_h = \{v_h \in H_0^1(\Omega) : v_h|_{\tau_e} \in \mathcal{Q}_1(\tau_e) \;\; \forall \tau_e \in \mathcal{T}_h\} \subset H_0^1(\Omega), \tag{10}$$

where $\mathcal{Q}_1$ denotes multilinear polynomials [6]. The inclusion $V_h \subset H_0^1(\Omega)$ makes the method conforming and is used in Theorem 1. On each element the approximation is $\psi_h|_{\tau_e} = \sum_{i=1}^{2^d} N_i^{(e)}(\boldsymbol{\xi})\, \psi_i^{(e)}$.

Volume integrals are evaluated with 2^d-point Gauss–Legendre quadrature per element [6], and for each element the code computes

$$K_{ij}^{(e)} = \int_{\tau_e} \nabla N_i \cdot \nabla N_j \, \mathrm{d}\Omega, \quad M_{ij}^{(e)} = \int_{\tau_e} N_i N_j \, \mathrm{d}\Omega, \quad M_{V,ij}^{(e)} = \int_{\tau_e} N_i V N_j \, \mathrm{d}\Omega. \tag{11}$$

Remark 2 (The tensor-product trap) On a tensor-product mesh the multidimensional matrices are *sums* of Kronecker products in which exactly one factor is a 1D stiffness matrix and the remaining factors are 1D mass matrices. In two dimensions,

$$K = K_1 \otimes M_1 + M_1 \otimes K_1, \qquad M = M_1 \otimes M_1, \tag{12}$$

and in three dimensions K is the sum of the three cyclic terms $K_1 \otimes M_1 \otimes M_1 + M_1 \otimes K_1 \otimes M_1 + M_1 \otimes M_1 \otimes K_1$, with $M = M_1 \otimes M_1 \otimes M_1$. A tempting shortcut is to use a single Kronecker product of 1D stiffness matrices, $K_1 \otimes K_1$ in two dimensions. That is *incorrect*: it omits the cross terms in which the derivative acts on only one of the two coordinates, and it is not even dimensionally consistent, since K_1 scales as h^{-1} while the multidimensional stiffness scales as h^{d-2}. The same remark applies to the potential term, which for a separable V is again a sum of terms with one weighted mass matrix $\int N_i V_k N_j$ in the corresponding slot. During development this error produced 2D and 3D eigenvalues roughly 20× too small while leaving the 1D results correct. That signature made the defect hard to attribute, because the one-dimensional verification passed. Assembling (11) by direct quadrature for every dimension removes the error. It is reported because it is easy to introduce in dimension-generic codes, and because it produces results that look plausible in the dimension where they are first tested.

3.2 Assembly and the generalized eigenproblem

Assembling (11) over $\mathcal{T}_h$ gives global matrices $K, M, M_V \in \mathbb{R}^{n\times n}$, symmetric, with K and M_V sparse and

$$A = K + M_V, \qquad A\,\boldsymbol{\psi} = E\,M\,\boldsymbol{\psi}. \tag{13}$$

This is a generalized symmetric eigenproblem with M symmetric positive definite [6, 22]. Its eigenvectors are M-orthonormal: $\boldsymbol{\psi}_i^{\top} M \boldsymbol{\psi}_j = \delta_{ij}$, which provides an implementation-independent check used in Section 7.5.

3.3 Dirichlet boundary conditions

The treatment of constrained degrees of freedom deserves attention. The most common approach for boundary value problems is incorrect here, and the way in which it fails depends on how the mass matrix is treated.

For a source problem $Au = f$, Dirichlet data are frequently imposed by zeroing the row and column of A and placing 1 on the diagonal. Two variants of this recipe appear in practice when it is carried over to (13).

1. $A_{ii} = 1$ *and* $M_{ii} = 1$ for a constrained node i. The pair (A_{ii}, M_{ii}) then defines a one-dimensional generalized problem whose Rayleigh quotient is exactly 1, so the

discrete spectrum contains $E = 1$ with an eigenvector supported on the constrained nodes. With n_c constrained nodes the value 1 appears with multiplicity up to n_c, independently of the physical problem.

2. $A_{ii} = 1$ with M_{ii} left at its assembled value. The diagonal quotient is then $1/M_{ii} \neq 1$, and no eigenvalue is pinned at 1; but the constrained rows no longer represent the variational problem, and the spectrum is perturbed in a way that is largest for the low-lying states of interest.

Both variants were verified numerically on the 1D harmonic case ($L = 20$, 40 elements). With $A_{ii} = M_{ii} = 1$ the two lowest computed eigenvalues are 1.000000 and 1.000000, against the exact 1 and 3: the low-frequency spectrum is dominated by two spurious modes. With $A_{ii} = 1$ and M_{ii} unchanged the spurious value disappears, but the two lowest eigenvalues become 1.057052 and 3.117594, which are also wrong. In either case the constrained rows must be *removed*, not modified.

The correct procedure eliminates the constrained unknowns [6]. The following statement makes the equivalence precise.

Proposition 1 (Elimination) *Let W_h be the finite element space of multilinear functions on the mesh,* before *any boundary condition is imposed, and let $V_h = W_h \cap H_0^1(\Omega)$ be the conforming subspace obtained by requiring that the nodal values on $\partial\Omega$ vanish. Write V_h^0 for the further subspace of V_h in which the constrained nodal values are set to zero, so that V_h^0 is the discrete counterpart of H_0^1 once the degrees of freedom attached to the constrained nodes have been removed. (In the mesh used here the constrained nodes are exactly those on $\partial\Omega$, so $V_h^0 = V_h$ as sets of functions; the distinction matters when constrained nodes are interior, for instance on an interface.) Then the variational eigenproblem on V_h^0,*

$$q(\psi_h, v_h) = E_h\, m(\psi_h, v_h) \qquad \forall v_h \in V_h^0, \tag{14}$$

is equivalent to the reduced algebraic problem

$$A_{ff}\,\boldsymbol{\psi}_f = E\, M_{ff}\,\boldsymbol{\psi}_f : \tag{15}$$

the two have the same eigenvalues, and the nodal coefficients of an eigenfunction of (14) are $(\boldsymbol{\psi}_f, \mathbf{0})$. Moreover A_{ff} and M_{ff} are symmetric and M_{ff} is positive definite.

Proof The basis functions φ_j with $j \in f$ span V_h^0, since the coefficients of the constrained basis functions vanish there. Writing $v_h = \sum_{j\in f} v_j \varphi_j$ and testing (14) with φ_i, $i \in f$, gives

$$\sum_{j\in f} (A_{ij} - E_h M_{ij}) v_j = 0 \qquad \forall i \in f, \tag{16}$$

which is precisely (15) with $A_{ff} = (A_{ij})_{i,j\in f}$ and $M_{ff} = (M_{ij})_{i,j\in f}$. Conversely, every solution of (15), extended by zero on the constrained nodes, satisfies (14), because the φ_j with $j \in f$ span V_h^0. The block rows indexed by c are *not* imposed, and are in general not satisfied: eliminating a degree of freedom means discarding its equation, not merely setting its coefficient to zero (Remark 3). Symmetry of A_{ff} and M_{ff} follows by restriction. For positive definiteness, let $\boldsymbol{\psi}_f \neq \mathbf{0}$; then $\boldsymbol{\psi}_f^\top M_{ff} \boldsymbol{\psi}_f = (\boldsymbol{\psi}_f, \mathbf{0})^\top M (\boldsymbol{\psi}_f, \mathbf{0}) > 0$, because M is positive definite and $(\boldsymbol{\psi}_f, \mathbf{0}) \neq \mathbf{0}$. □

Remark 3 (The discarded rows are not satisfied) It is tempting to describe the elimination as imposing $\boldsymbol{\psi}_c = \mathbf{0}$ in the block form of (13) and then observing that the first block row reduces to (15). The second block row, however, becomes $A_{cf}\boldsymbol{\psi}_f = EM_{cf}\boldsymbol{\psi}_f$, which is an additional constraint and holds only by accident. This was checked numerically on two benchmark cases. For the 1D harmonic oscillator the residual of the discarded row is 8×10^{-15}, which is round-off, but only because the exact eigenfunction has already decayed to $\sim 10^{-17}$ at the first interior node. For the infinite well, where the eigenfunction is $\mathcal{O}(h)$ near the wall, the same residual is 1.8×10^{-2} against a peak amplitude of 2.2×10^{-1}, that is, 8%. The distinction is therefore not pedantic: it separates a proof that holds in general from one that holds only for the subclass of problems whose eigenfunctions decay fast enough at the boundary.

Both implementations perform this elimination, and the number of free degrees of freedom n_f is reported for every case, so that the effective problem size is unambiguous.

4 Convergence and properties of the discrete spectrum

Two properties of the conforming approximation. They serve both as theoretical justification and as practical checks.

4.1 The discrete eigenvalues are upper bounds

Theorem 1 (Upper bound) *Let q be the energy form (7) and suppose $V \in L^\infty(\Omega)$ is bounded below, so that q is bounded below and the continuous problem (6) has a discrete spectrum $E_1 \le E_2 \le \cdots$. Let $E_{h,1} \le E_{h,2} \le \cdots \le E_{h,n_h}$ be the eigenvalues of the discrete problem*

$$q(\psi_h, v_h) = E_h\, m(\psi_h, v_h) \qquad \forall v_h \in V_h, \tag{17}$$

with $n_h = \dim V_h$. If $V_h \subset H_0^1(\Omega)$ and the discrete form is the restriction of q to $V_h \times V_h$, then

$$E_{h,n} \ge E_n \qquad \text{for all } n \le n_h. \tag{18}$$

Proof By (9), $E_n = \min_{\dim S_n = n} \max_{0 \ne v \in S_n} R(v)$ with $S_n \subset H_0^1(\Omega)$, whereas the discrete eigenvalues satisfy the same formula with S_n restricted to subspaces of V_h [2, 7, 8]. Since $V_h \subset H_0^1(\Omega)$, the set of admissible subspaces in the discrete case is a subset of the admissible set in the continuous case, and minimizing over a smaller set cannot decrease the minimum. Hence $E_{h,n} \ge E_n$. □

Remark 4 (Scope of the upper bound) The hypothesis that the discrete form be the restriction of q is essential, and the implementations used here do *not* satisfy it exactly: they replace V by its nodal interpolant $\mathcal{I}_h V$ when assembling the potential term (Remark 8). The consequence depends on the sign of $V - \mathcal{I}_h V$. If V is convex, as for the harmonic potential $V = x^2$ on the tensor-product meshes used here, the multilinear interpolant is a convex combination of the vertex values and therefore satisfies $\mathcal{I}_h V \ge V$ pointwise; hence $q_h(v, v) \ge q(v, v)$ for every $v \in V_h$ and the upper-bound property survives. The argument uses the tensor-product structure and does not extend automatically to arbitrary element geometries or to non-convex potentials. For a general, non-convex V no such inequality holds, and the computed eigenvalues need not be upper bounds for those of the continuous problem. The property is

therefore asserted for the exactly integrated problem, and *verified numerically* for the cases at hand (Section 7.5) rather than assumed.

Remark 5 Subject to that scope, Theorem 1 has a practical consequence: computed eigenvalues approach the exact ones from above. A mesh refinement that moves an eigenvalue in the wrong direction therefore signals an implementation error, not a discretization effect. This serves as a consistency check in Section 7.5, while noting that monotonicity alone is not a proof of correctness.

4.2 A priori eigenvalue error estimate

Theorem 2 (Convergence) *Let $\{\mathcal{T}_h\}_h$ be a shape-regular family of partitions of Ω into multilinear elements of degree p, and let the discrete form be the restriction of q to $V_h \times V_h$. Suppose that the eigenvalues $E_n, \dots, E_{n+m-1}$ form a group of multiplicity m separated from the rest of the spectrum, and that the associated spectral subspace $\mathcal{E} = \operatorname{span}\{u_n, \dots, u_{n+m-1}\}$ satisfies $\mathcal{E} \subset H^{1+s}(\Omega)$ for some $s \in (0, p]$. Then there exist constants C and C', depending on the group, on the separation distance and on the shape regularity of $\mathcal{T}_h$ but not on h, such that*

$$|E_{h,\ell} - E_\ell| \le C\, h^{2s} \qquad \ell = n, \dots, n+m-1, \tag{19}$$

and, for a suitable orthonormal basis $\{\psi_{h,\ell}\}$ of the discrete spectral subspace, $\|\psi_{h,\ell} - u_\ell\|_{H^1} \le C' h^s$.

The estimate (19) is the classical result of Babuška and Osborn [7, 8], stated here for a spectral group rather than for a simple eigenvalue; the simple case is $m = 1$. Two features deserve emphasis.

Remark 6 (From the theorem to the implemented scheme) Theorem 2 applies to a discrete form that is the exact restriction of q. The implementations used here are not that: they replace V by its nodal interpolant $\mathcal{I}_h V$ and evaluate the integrals by Gauss quadrature (Remark 8). The scheme they actually solve is therefore a perturbation of the one the theorem describes, and the two must be related explicitly rather than by assertion. Writing $q_h = q + (q_h - q)$, the consistency error $q_h - q$ is a bilinear form with kernel $\int (\mathcal{I}_h V - V) vw$ over $V_h \times V_h$. For the smooth potentials considered in Sections 7–8, $\|V - \mathcal{I}_h V\|_{L^\infty} = \mathcal{O}(h^2)$, so this perturbation is $\mathcal{O}(h^2)$ in the energy norm. The estimate is *not* claimed for a potential with a jump: there $\|V - \mathcal{I}_h V\|_{L^\infty}$ is $\mathcal{O}(h^0)$ and the argument below does not apply. The step potential of Case B is a separate consistency study, reported in Section 8.4, and its rates are not a consequence of Theorem 2. A standard perturbation argument [7, 8] then transfers (19) to the perturbed problem at the cost of an additive $\mathcal{O}(h^2)$ term, which for $s = 1$ is the same order as the estimate itself. This is why the observed orders in Tables 2–8 are unchanged by the interpolation, and it is also why no stronger claim is made: the argument bounds the extra term, it does not remove it, and the constant in it is not computed here.

First, the exponent s is a property of the *problem*, not of the method: it is limited by the smoothness of $\partial\Omega$, by the regularity of V and by the boundary conditions. On a convex polygon with $V \in L^\infty$ one has $u_n \in H^2$ and $s = 1$; on a domain with reentrant corners, or for a potential with a discontinuity, $s < 1$ and the asymptotic rate

is correspondingly reduced. The estimate also presumes a *shape-regular* mesh family and a consistent approximation of the same variational problem.

Second, the estimate concerns the *spectral subspace*, not individual eigenfunctions: within a group of multiplicity $m > 1$ the individual eigenvectors are not determined by the differential operator, so only the subspace, equivalently the set of eigenvalues, is a well-posed object. This is the reason the degeneracies of Section 7.5 are a meaningful test, whereas a comparison of individual eigenvectors within a degenerate group is not.

Corollary 3 *Let Ω be a bounded convex polygon and let $V \in L^\infty(\Omega)$ with $\mathcal{E} \subset H^2(\Omega)$. Then for multilinear elements ($p = 1$, $s = 1$)*

$$|E_{h,\ell} - E_\ell| = \mathcal{O}(h^2). \tag{20}$$

The hypothesis $\mathcal{E} \subset H^2$ is what makes $s = 1$ available. It holds for a convex polygon in two dimensions by the classical elliptic regularity theory [7, 8], and for $V \in L^\infty$ it is not lost, since the potential term is a lower-order perturbation. In three dimensions the statement needs care: H^2 regularity on a general Lipschitz polyhedron is *false*: the solution develops edge and vertex singularities even for smooth data, so Corollary 3 should not be read as a claim about arbitrary hexahedral meshes on a general polyhedron. The boxes used here are rectangular parallelepipeds, for which the solution is as regular as the data permits, and the observed order in Table 8 is 1.98–1.99 in 3D, consistent with $s = 1$. Where the domain is not convex the exponent drops below 1 and the observed order falls accordingly.

Remark 7 (Theoretical versus observed order) The order reported in Tables 2 and 8 is an *observed* quantity, obtained as $\log(e_i/e_{i+1})/\log(h_i/h_{i+1})$ from the relative errors of two consecutive refinements. It is therefore a local estimate that inherits the noise of both, and it should be read as evidence consistent with Corollary 3, not as a measurement of the exponent s. The tables report the relative error $|E_{h,1} - E_1|/|E_1|$; the corresponding absolute errors differ by the factor $|E_1|$, which is why the Morse case, with $|E_1| \approx 8.5$, appears to have larger absolute deviations than the infinite well at the same resolution. Convergence of eigenfunctions is a separate question: the H^1 error is only $\mathcal{O}(h^s)$, and a comparison of eigenfunctions additionally requires a normalization convention and a choice of sign, both of which are arbitrary.

5 Eigensolution strategies

Only the lowest few eigenpairs of (15) are normally of interest. Three strategies are considered and their cost analyzed. The adjacent problem of locating bound states in a device that carries contacts leads to a nonlinear eigenvalue problem, for which Krylov methods of the same family are used [32].

5.1 Dense direct solution

The classical approach reduces (15) to a standard eigenproblem via a Cholesky factorization $M_{ff} = LL^\top$ and calls the LAPACK driver `DSYGV` [23, 33], which computes

all eigenpairs. It is robust and requires no tuning. Its cost is the obstacle. Storing the two matrices A_{ff} and M_{ff} in double precision requires at least

$$\mathrm{mem}_{\mathrm{dense}} \;\geq\; 16\, n_f^2 \text{ bytes}, \tag{21}$$

and the factorization costs $\mathcal{O}(n_f^3)$ operations. Equation (21) is the irreducible minimum for holding the two reduced matrices alone. The implementations used here do something larger, and it is worth stating exactly what, because the footprint is not three matrices of size n_f but four. The stiffness and the potential-weighted mass are accumulated *into* A during assembly, so A costs one matrix, not two; the global arrays are therefore A and B, each of order n_{dof}, where $n_{\mathrm{dof}} = n_f + (\text{boundary nodes})$. The elimination then allocates *copies* of the free blocks, A_{ff} and B_{ff}, while A and B are still alive, and all four are resident during `DSYGV`. The peak is

$$\mathrm{mem}_{\mathrm{peak}} = 16\,(n_{\mathrm{dof}}^2 + n_f^2) \text{ bytes}, \tag{22}$$

For the square grids used here $n_{\mathrm{dof}} = (n+1)^2$ and $n_f = (n-1)^2$, so at $n_f = 39\,601$ the peak is 51.2 GB, against 25.1 GB for the two-reduced-matrix minimum of (21) and 37.6 GB for the three-matrix estimate that a naive count gives. Throughout, GB denotes 10^9 bytes and not 2^{30}. Holding four arrays rather than two is a choice of the present implementation and not a requirement, and it is what the memory column of Table 9 reports.

5.2 Sparse shift-and-invert Arnoldi iteration

The alternative exploits sparsity and the fact that only a few eigenpairs are wanted. Implicitly restarted Arnoldi iteration [21] computes the eigenvalues of largest magnitude of a matrix. To obtain the *smallest* eigenvalues of (15) the shift-and-invert transformation is used [34], which converts the problem into one whose dominant eigenvalues correspond to those nearest a prescribed shift σ:

$$(A_{ff} - \sigma M_{ff})^{-1} M_{ff}\, \mathbf{v} = \mu\, \mathbf{v}, \qquad E = \sigma + \frac{1}{\mu}. \tag{23}$$

Here $\mathbf{v}$ is the eigenvector of the transformed problem and the eigenvalues μ of largest magnitude correspond to the eigenvalues E of (15) nearest σ. The factorization of $(A_{ff} - \sigma M_{ff})$ is performed *once*, in sparse form, before the iteration [23], giving where n_{cv} is the dimension of the Arnoldi basis, that is the size of the Krylov subspace actually stored, and is in general larger than the number k of requested eigenpairs: the solver keeps a working subspace from which the k Ritz pairs are extracted. In the experiments below n_{cv} was left at the library default, so no numerical value is asserted for it here.

$$\mathrm{mem}_{\mathrm{sparse}} = \underbrace{\mathcal{O}(\mathrm{nnz})}_{\text{input matrices}} + \underbrace{\mathcal{O}(\mathrm{nnz}_L)}_{\text{factors}} + \underbrace{\mathcal{O}(n_f\, n_{\mathrm{cv}})}_{\text{Arnoldi basis}}, \tag{24}$$

where nnz denotes the number of nonzeros of $(A_{ff} - \sigma M_{ff})$, nnz_L that of its factors, and k the number of requested eigenpairs. The middle term is the one most easily

overlooked. Shift-and-invert factorizes the matrix once and keeps the factors for the whole iteration, and fill-in makes nnz_L substantially larger than nnz; its size depends on the ordering, on the elimination tree and on the spatial dimension, so it is not linear in n_f with a constant independent of the mesh. The two are therefore not interchangeable, and a figure that quoted only the first would understate the cost of the method. This is the strategy implemented in the Python reference code and is available to the Elmer module through matrix export.

5.3 Distributed solution

The same transformation can be distributed: PETSc [35] provides the parallel sparse matrix infrastructure and SLEPc [36] the Krylov–Schur eigensolver with shift-and-invert. Both rely on a *parallel* sparse direct factorization for the inner solve; without one the shift-and-invert step cannot be distributed and the driver is confined to a single process. This is a deployment constraint rather than an algorithmic one, and what was measured is reported in Section 8.3.

5.4 Cost model

For a quasi-uniform mesh in d dimensions each node couples to $\mathcal{O}(1)$ neighbors, so the number of nonzeros of A_{ff} and M_{ff} grows linearly, $\mathrm{nnz} = \mathcal{O}(n_f)$, and the storage estimate (24) holds. The *time*, however, is not captured by a single term, and the contributions are separated rather than stated as a universal law.

A shift-and-invert Arnoldi solve consists of:

1. the assembly of A_{ff} and M_{ff}, of cost $\mathcal{O}(\mathrm{nnz})$;
2. one sparse direct factorization of $(A_{ff} - \sigma M_{ff})$, of cost $\mathcal{O}(\mathrm{nnz}^{\alpha})$, where α depends on the dimension, the reordering and the elimination tree and is *not* bounded by 3/2 in general: nested dissection grows like $\mathcal{O}(n^{3/2})$ in two dimensions but can approach $\mathcal{O}(n^2)$ in three, so no single exponent is asserted here for either case [23];
3. n_{it} Arnoldi iterations, each requiring one back-substitution with the stored factors, of cost $\mathcal{O}(\mathrm{nnz}_L)$ where nnz_L is the number of nonzeros of the factor;
4. the eigensolver overhead proper, of order $n_f\, n_{\mathrm{cv}}^2$ for a basis of n_{cv} vectors of length n_f, the reorthogonalisation being what makes this quadratic in the basis size rather than linear.

The factorization cost is therefore *not* linear in n_f in general, and in the regime studied here it is the second term that dominates. Writing the four contributions separately,

$$T_{\mathrm{sparse}} \sim \underbrace{c_1\,\mathrm{nnz}}_{\text{assembly}} + \underbrace{c_2\,\mathrm{nnz}^{\alpha}}_{\text{factorization}} + \underbrace{c_3\, n_{\mathrm{it}}\,\mathrm{nnz}_L}_{\text{iterations}} + \underbrace{c_4\, n_f\, n_{\mathrm{cv}}^2}_{\text{overhead}}, \tag{25}$$

while the dense route remains $T_{\mathrm{dense}} \sim n_f^3$ with storage $\mathcal{O}(n_f^2)$. Because the factorization term dominates in the range of n_f considered, the measured ratio $T_{\mathrm{dense}}/T_{\mathrm{sparse}}$ grows faster than n_f, consistently with the n_f^2 ratio that a purely linear sparse cost would imply. What the measurements in Section 8.3 establish is the *behavior*, not the

exponent: the dense cost is observed to grow as n_f^3, while the sparse cost stays below a second across the two orders of magnitude in n_f.

A reproducible cost report should distinguish at least the assembly time, the factorization time and memory, the number of Arnoldi iterations, the eigensolver tolerance and the residual attained. Of these, the eigensolution time is reported, the dense-memory requirement and one residual; the assembly time and the Arnoldi iteration count are not instrumented in the present study, and the factorization happens inside the external sparse solver, which does not expose its own timing. That gap is worth closing in a software paper that claims reproducibility.

6 Implementation

To enable the code-to-code verification of Section 7, the formulation of Sections 2–3 was implemented *twice*, independently, in different languages and on different infrastructure.

6.1 Elmer solver module

Elmer [29] is an open-source finite element suite developed at CSC – IT Center for Science (Finland) and used principally for multiphysics problems. Its solver interface allows the addition of user modules compiled against the Elmer libraries. The module `SchrodingerSolver` was implemented following it, in about 460 lines of Fortran 90, and contains the discretization, the elimination and the call to the eigensolver. Mesh handling, assembly primitives, the constrained-node queries and the field output come from the host suite and are not part of the module.

Meshes for all cases were generated with Gmsh [37] and converted to the Elmer format with ElmerGrid. The module reads the mesh through the standard element interface and identifies constrained nodes from the boundary elements, giving the free/constrained partition of Proposition 1. It reads the potential as described in Section 6.3, assembles K, M and M_V, and solves the dense generalized symmetric problem with `DSYGV`. It writes the ground-state eigenfunction back to the Elmer field database, so that standard output modules produce VTK files, and it optionally exports A_{ff} and M_{ff} in ASCII format for the external sparse route.

The documentation follows the structure of the official Elmer manuals (model manual, solver manual and per-case tutorials), to lower the barrier for external users.

6.2 Python reference implementation

The second implementation is a dimension-generic Python code that shares no source code with the Elmer module. It comprises a mesh generator and assembler for structured 1D, 2D and 3D grids using direct Gauss quadrature as in (11); the sparse shift-and-invert Arnoldi solver of Section 5, built on ARPACK through SciPy, with the same interface as the dense solver so that the two can be interchanged; and a PETSc/SLEPc driver implementing distributed Krylov–Schur with shift-and-invert, together with conversion utilities for the assembled matrices. The Python code is the

one used for the performance measurements, because it allows the dense and sparse strategies to be exercised on identical matrices.

6.3 Potential specification in a general framework

Specifying a multidimensional potential in a general finite element code requires care, and a practical obstacle is documented. The expression parser of Elmer does not expose the individual coordinate components in a form that allows a two- or three-dimensional expression to be written directly for the potential field. The syntaxes tested, a coordinate list, per-component indexing, and separate named variables, all failed with parser errors.

A component-wise convention was therefore implemented *inside the solver*: the potential is read as a sum of independent nodal fields, one per coordinate,

$$V(x_1, \ldots, x_d) = \sum_{k=1}^{d} V_k(x_k), \tag{26}$$

each obtained by an independent database query, with the precedence order: (i) a directly specified nodal field `Potential` (arbitrary V, used in 1D); (ii) component fields `Potential X`, `Potential Y`, `Potential Z`, summed by the solver (used in 2D and 3D); (iii) $V \equiv 0$ if none is present. The restriction to separable potentials is mild for the verification suite considered here. Removing it would require extending the expression interface rather than the solver core; this is revisited in Section 9.2.

7 Verification

Verification, the process of establishing that the code solves the equations it is intended to solve, is treated here as a first-class component. It differs from validation, which establishes agreement with experiment; for an experimental validation study in a comparable electromagnetic setting, see [38]. Five independent checks are applied here, following the philosophy of the verification literature [24, 25]: analytic benchmarks for seven potentials (Section 7), measured convergence orders (Section 7.3), code-to-code comparison of two independently written implementations (Section 7.4), and structural invariants (Section 7.5). The fifth, the reproduction of two published benchmarks, is reported separately in Section 8.4 because its role is different: the first four verify the code against mathematics, the fifth tests it against the literature of this journal.

7.1 Analytic benchmarks

Closed-form solutions are available for the potentials in Table 1 (Appendix A), and they are classical [2, 4]. The infinite well is a genuine test of the discretization: the eigenvalue problem is satisfied in the interior with homogeneous boundary data, and the eigenvalues are known exactly.

Table 1 Analytic reference solutions used for verification, in the unit convention of (3).

potential	notation	eigenvalues	eigenfunctions
infinite well (1D)	$V = 0$, $\psi(\pm L/2) = 0$	$E_n = (n\pi/L)^2$	trigonometric
infinite box (dD)	$V = 0$, $\psi\|_{\partial} = 0$	$E = \pi^2 \sum_k n_k^2/L^2$	products
harmonic, $V = \alpha r^2$	isotropic	$E_n = (2n + d)\sqrt{\alpha}$	$H_n\times$ Gaussian
Morse	$D_e(1 - e^{-a(x-x_0)})^2 - D_e$	closed form	—

7.2 Normalization of the analytic ground state

Because the Gaussian of (A2) is used as a reference in the tests above, its prefactor was checked numerically against $\int |\psi_0|^2 \mathrm{d}^d r = 1$ for several values of α. Writing the prefactor as $c(\alpha)^d$ with $c = (\sqrt{\alpha}/\pi)^{1/4}$, the product form of the integral gives $c^{2d}(\pi/\sqrt{\alpha})^{d/2} = 1$ for every α. Using the incorrect $(\alpha/\pi)^{d/4}$ instead yields a total norm of 0.500 for $\alpha = 1/4$ in 2D and 0.354 in 3D, and the two prefactors coincide only at $\alpha = 1$. This is why the 1D benchmark case, which uses $\alpha = 1$, does not detect the error, whereas the 2D and 3D cases with $\alpha = 1/4$ do.

7.3 Convergence order

Corollary 3 predicts $\mathcal{O}(h^2)$ for the first eigenvalue on convex domains with multilinear elements. This was measured directly. For the 1D infinite well with $L = 40$, refining from $N = 25$ to $N = 400$ elements, the relative error in E_1 falls from 1.317×10^{-3} to 5.140×10^{-6}, giving a fitted order of 2.00 over four refinements. In 2D, refining a 10×10 to an 80×80 mesh for the square well reduces the error from 8.251×10^{-3} to 1.285×10^{-4}, also order 2.00. Table 2 gives both sequences, and Figure 1(a) shows them against the reference slope. The measured orders agree with (19), which turns the observed convergence from an empirical observation into a predicted-and-confirmed property.

Throughout this section h denotes the diameter of an element, so that for the uniform structured meshes used here $h = L/n$ in one dimension and $h = L/n$ in each coordinate direction in higher dimensions, with n the number of elements per side. Each refinement halves h exactly, so the fitted order $\log(e_i/e_{i+1})/\log 2$ is exact rather than approximate; here e_i is the relative error $|E_{h,1} - E_1|/|E_1|$ against the analytic value of Table 1. All meshes are uniform, so the shape-regularity hypothesis of Theorem 2 is satisfied with a constant that does not degrade under refinement.

Table 2 concerns the eigenvalues. The eigenfunctions converge more slowly, and Table 3 reports their error for the same 1D well, measured against the exact $\psi(x) = \sqrt{2/L}\cos(\pi x/L)$ in the L^2 and H^1 norms (sign aligned). The H^1 error decreases at order 1, the L^2 error at order 2. These are expected from the usual Aubin–Nitsche duality argument applied on top of Theorem 2, which states the energy-norm estimate and the eigenvalue estimate directly: the energy-norm error is $\mathcal{O}(h^s)$ with $s = 1$, one order worse than the eigenvalue error, and the additional order in L^2 is the usual Aubin–Nitsche gain. This is the practical content of the doubling of the order mentioned after Theorem 2.

Table 2 Measured convergence of the first eigenvalue. The fitted order is obtained from consecutive refinements and compared with the theoretical value $2s = 2$ of Corollary 3.

discretization	h	relative error in E_1	fitted order
1D well, $N = 25$	1.600	1.317×10^{-3}	—
1D well, $N = 50$	0.800	3.290×10^{-4}	2.00
1D well, $N = 100$	0.400	8.225×10^{-5}	2.00
1D well, $N = 200$	0.200	2.056×10^{-5}	2.00
1D well, $N = 400$	0.100	5.140×10^{-6}	2.00
2D box, $n = 10$	1.000	8.251×10^{-3}	—
2D box, $n = 20$	0.500	2.058×10^{-3}	2.00
2D box, $n = 40$	0.250	5.141×10^{-4}	2.00
2D box, $n = 80$	0.125	1.285×10^{-4}	1.99

Table 3 Error of the first *eigenfunction* for the 1D infinite well ($L = 40$), against the exact solution, after aligning the sign. The orders are computed as in Remark 7.

n	h	$\|\psi_h - \psi\|_{L^2}$	order	$\|\psi_h - \psi\|_{H^1}$	order
25	1.600	5.90×10^{-4}	—	2.91×10^{-3}	—
50	0.800	1.47×10^{-4}	2.00	1.43×10^{-3}	1.02
100	0.400	3.68×10^{-5}	2.00	7.13×10^{-4}	1.01
200	0.200	9.20×10^{-6}	2.00	3.56×10^{-4}	1.00
400	0.100	2.30×10^{-6}	2.00	1.78×10^{-4}	1.00

7.4 Code-to-code verification

The strongest evidence in this study comes from comparing the two independent implementations. Using identical meshes and potentials, and requesting the lowest eigenvalue, the Elmer module and the Python code agree to within 4×10^{-13} in relative terms across all three spatial dimensions (Table 4). The last column of that table gives the normalized residual $\|A_{ff}\boldsymbol{\psi} - EM_{ff}\boldsymbol{\psi}\|/\|A_{ff}\boldsymbol{\psi}\|$ of the computed eigenpair. Both norms are Euclidean, and the eigenvector is normalized by $\boldsymbol{\psi}^\top M_{ff}\boldsymbol{\psi} = 1$. Dividing by $\|A_{ff}\boldsymbol{\psi}\|$ makes the quantity dimensionless and independent of the scaling of A, which is what allows rows whose eigenvalues differ greatly in magnitude to be compared.

The level of agreement should be read with the conditioning of the problem in mind. For the 1D well the first eigenvalue, 6.2×10^{-3}, is three orders of magnitude smaller than $\|A_{ff}\| \approx 1/h = 6.25$, so the relative accuracy attainable in E_1 is limited to roughly $\varepsilon\,\|A_{ff}\|/E_1 \approx 10^{-12}$ in double precision, whatever algorithm is used. The residual of 3.2×10^{-11} in that row reflects the same conditioning. For the 2D and 3D cases the eigenvector is well conditioned and both the eigenvalue difference and the residual are at the 10^{-13} and 10^{-14} levels respectively. Three independent solution paths for the 1D case were checked (the in-framework dense solve, the exported sparse solve, and an independent dense implementation written for this verification) differ among themselves by up to 5×10^{-12} in relative terms, the same round-off scale.

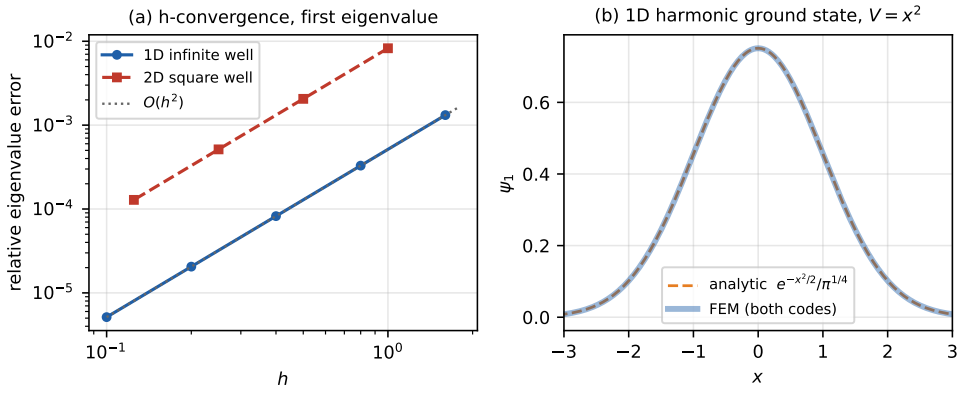


Fig. 1 (a) Relative error of the first eigenvalue against mesh size h for the 1D infinite well and the 2D square well, with the $\mathcal{O}(h^2)$ reference slope predicted by Corollary 3. (b) Ground state of the 1D harmonic oscillator, $V = x^2$: both implementations produce the analytic Gaussian to within the line width.

Table 4 Code-to-code verification: two independently written implementations on identical meshes and potentials.

case	Elmer E_1	Python E_1	relative difference	$\|A\psi - EM\psi\|/\|A\psi\|$
1D infinite well	$6.168583925347 \times 10^{-3}$	$6.168583925345 \times 10^{-3}$	3.6×10^{-13}	3.2×10^{-11}
2D harmonic	1.028585384356000	1.028585384356285	2.8×10^{-13}	4.8×10^{-14}
3D harmonic	1.625507320678000	1.625507320677697	1.9×10^{-13}	1.1×10^{-14}

Because the two codes share no source code, use different mesh containers, different assembly loops and different eigensolver libraries, agreement at this level strongly constrains a broad class of implementation errors that comparison against analytic solutions alone would not detect. The approach follows the spirit of earlier cross-validation of independent codes in computational physics [28] and is, in the author's assessment, the single most informative result of the verification program, available precisely because two implementations were developed.

A related internal check is that solving the *exported* matrices with the sparse pipeline reproduces the dense in-framework solution (Table 5). The twelve digits shown coincide, so the export introduces no error at the level of the reported precision and the sparse route is a drop-in replacement for the dense one. This is a statement about the two solution paths agreeing with each other, not about either being exact: both solve the same discrete problem, and both inherit whatever discretization error that problem carries.

Table 5 The exported-matrix sparse route reproduces the in-framework dense solution to all displayed digits.

route	E_1	E_2	E_3	E_4
in-framework (dense `DSYGV`)	1.113644656498	2.172208383659	2.172208383659	3.230772110820
exported + sparse (ARPACK)	1.113644656498	2.172208383659	2.172208383659	3.230772110820

7.5 Structural consistency checks

Three checks require no external reference.

Mass-orthonormality. The computed eigenvectors must satisfy $\boldsymbol{\psi}_i^\top M \boldsymbol{\psi}_j = \delta_{ij}$. This was verified numerically. In the 2D case the largest deviation from the identity is 3.1×10^{-15}, at the level of double-precision round-off.

Degeneracies. Symmetry of the domain and potential implies degeneracies of known multiplicity. The 2D isotropic harmonic oscillator on a square domain must have a doubly degenerate first excited state; the computed values are $E_2 = E_3 = 2.043966018888$ to all displayed digits. In 3D the first excited state is triply degenerate; the computed values are $E_2 = E_3 = E_4 = 2.669307913349$. Degeneracies are a strong test because they are destroyed if the assembly breaks the symmetry, for instance through inconsistent element ordering or a dimension-dependent quadrature error.

Virial theorem. For a harmonic potential the virial theorem [2] requires $\langle T \rangle = \langle V \rangle = E/2$. Multiplying (3) by ψ and integrating by parts gives $\langle T \rangle = \int |\nabla\psi|^2 \mathrm{d}\Omega$ and $\langle V \rangle = \int V|\psi|^2 \mathrm{d}\Omega$, so that $E = \langle T \rangle + \langle V \rangle$ holds identically. Discretely, the natural definitions are the quadratic forms of the assembled matrices, $\langle T \rangle_h = \boldsymbol{\psi}^\top K \boldsymbol{\psi}$ and $\langle V \rangle_h = \boldsymbol{\psi}^\top M_V \boldsymbol{\psi}$ with $\boldsymbol{\psi}$ normalized by $\boldsymbol{\psi}^\top M \boldsymbol{\psi} = 1$; then $E_h = \langle T \rangle_h + \langle V \rangle_h$ holds exactly by construction, and only the ratio is a test. For the 1D harmonic ground state the values obtained are

n	$\langle T \rangle_h$	$\langle V \rangle_h$	$\langle T \rangle_h / \langle V \rangle_h$
100	0.5000031	0.5091573	0.98202
200	0.5000002	0.5022911	0.99544
400	0.5000000	0.5005729	0.99886
800	0.5000000	0.5001432	0.99971
1600	0.5000000	0.5000358	0.99993

The ratio approaches the exact value 1 as $\mathcal{O}(h^2)$, while $\langle T \rangle_h$ is already accurate to 10^{-8} at $n = 400$. The check is useful because it relates the eigenvalue to the eigenfunction without invoking the analytic form of the latter.

Two qualifications keep this honest. First, the identity $E = \langle T \rangle + \langle V \rangle$ is *not* a discrete theorem proved here: it holds by the construction of $A = K + M_V$ and by the normalisation, so it verifies the assembly of the three matrices and nothing beyond it. Second, the equality $\langle T \rangle = \langle V \rangle$ is a property of the *continuous* eigenfunction. What the table measures is how far the discrete ratio is from one, and it approaches one at

second order in h; this is reported as an observed rate. No discrete virial identity is derived here, and the check should not be read as one.

Two aspects of the finite box are separated. The identity $E = \langle T\rangle + \langle V\rangle$ is an integration by parts whose boundary term is $\psi\,\partial_n\psi$ on $\partial\Omega$, which vanishes because $\psi = 0$ there. It therefore holds in a bounded box with no correction, and the table confirms it ($E_h - \langle T\rangle_h - \langle V\rangle_h$ is at round-off, $\sim 10^{-14}$). The ratio $\langle T\rangle_h/\langle V\rangle_h$ is a different statement, derived from the scaling $x \mapsto \lambda x$, which in a bounded box is not a symmetry of the problem. It was checked numerically that it is nevertheless independent of the box: at fixed $h = 0.25$ and $\alpha = 1$ the ratio equals 0.97221 for every L from 12 to 28, so the deviation from 1 is entirely the $\mathcal{O}(h^2)$ discretization error above and not a truncation effect. The first row of the table, 0.98202 at $n = 100$, and the value 0.97221 at $h = 0.25$ are consistent with this reading.

Monotonicity. By Theorem 1 the computed eigenvalues must approach the exact ones from above under refinement. All sequences in Table 2 satisfy this.

8 Numerical experiments

8.1 Benchmark suite

The eigensolver parameters are those of the libraries, with the defaults left in place except where stated. LAPACK `DSYGV` uses its default workspace. ARPACK is called through SciPy with `sigma` $= 0$ (the smallest eigenvalues are wanted), `which` $=$ `'LM'` on the transformed problem, `tol` $= 0$ so that its stopping criterion is machine precision, and `ncv` and `maxiter` at library defaults; the number of requested eigenpairs is $k = 4$ in every timing reported below. The Elmer module uses the default of its own solver interface when it factorizes densely. The remaining measurements in this section were taken on the same workstation; no distributed run is reported. Timings are the wall-clock time of the eigensolution call alone, excluding assembly; the sparse figures are the minimum of three repetitions, whereas the dense figures are *single-run* timings, taken once because of their cost. The sparse and dense columns are therefore not statistically comparable, and the speed-ups derived from them are indicative rather than estimates. Absolute times are properties of the hardware and are not transferable; the algorithmic comparison is.

Seven cases were run to completion with the Elmer module, covering smooth and non-smooth potentials, bound and quasi-bound states, and one- and multidimensional geometries. Table 6 lists the four lowest eigenvalues of each, together with the number of free degrees of freedom. Figure 2 shows the potentials and the computed levels.

The double well. With a barrier separating two symmetric minima, tunneling splits the two lowest states; the computed value is $E_2 - E_1 = 0.376$, a splitting that vanishes if the barrier is removed. A *finite* splitting is a sensitive indicator that the barrier region is resolved and that the two wells are correctly coupled through it. A mesh too coarse in the barrier overestimates it, and a symmetry-broken discretization can suppress it artificially. Double-well structures of this kind are the standard model for coupled quantum wells and for the tunneling phenomena exploited in resonant tunneling devices [9].

Table 6 Four lowest eigenvalues of each case in the unit convention of (3), with the analytic values and the free degrees of freedom n_f after elimination.

case	E_1	E_2	E_3	E_4	n_f
1D infinite well	6.168584×10^{-3}	2.467531×10^{-2}	5.552310×10^{-2}	9.871683×10^{-2}	249
1D harmonic	1.000572892	3.001197697	5.002447062	7.004320695	399
1D Morse	−8.481033787	−5.818565432	−3.656064809	−1.991029948	399
1D double well	0.224110514	0.600028005	1.274687373	2.059803060	399
1D sinusoidal	−0.213200584	−0.209442228	0.285067468	0.343455929	399
2D harmonic	1.028585384	2.043966019	2.043966019	3.059346653	1521
3D harmonic	1.625507321	2.669307913	2.669307913	2.669307913	2197

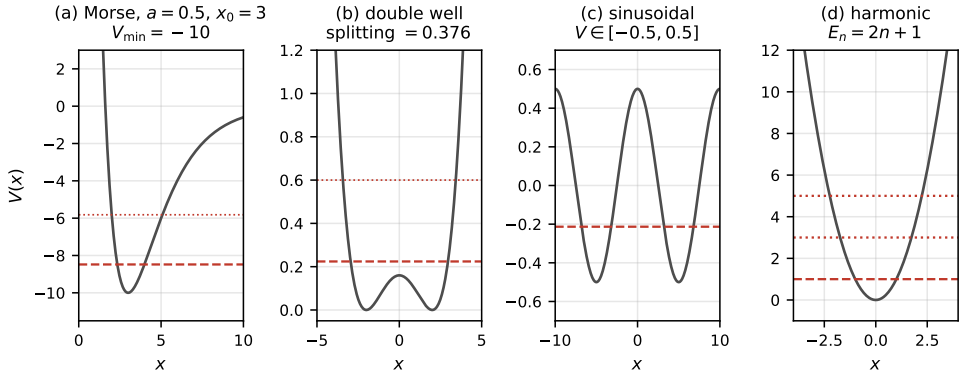


Fig. 2 Potentials of the 1D benchmark cases with the computed levels (dashed and dotted lines): (a) Morse, $V_{\min} = -10$; (b) double well, splitting $E_2 - E_1 = 0.376$; (c) sinusoidal, $V \in [-0.5, 0.5]$; (d) harmonic, $E_n = 2n + 1$.

The periodic potential. The lowest two eigenvalues, -0.213200584 and -0.209442228, lie just above the potential minimum $V_{\min} = -0.5$ and are separated by only 3.8×10^{-3}. That near-degeneracy is a property of this bounded interval and not, by itself, a band structure: the computation imposes Dirichlet conditions and defines no quasi-momentum, so the levels are those of a finite segment of a periodic potential rather than of a crystal. Establishing a band structure would require Bloch conditions or a superlattice with many periods, and is outside the present scope. Potentials of this type model superlattices and miniband structures [4]; for the computation of genuine band structures, where the eigenproblem is again of the shift-and-invert type, see [39].

8.2 Discretization error versus box truncation

The harmonic cases deserve care, because two distinct error sources are easily conflated there. The analytic eigenvalues $E_n = n + d/2$ in the convention of (3) refer to an unbounded domain, whereas the computation imposes $\psi = 0$ on the boundary of

a finite box of side L. Both the finite box and the finite mesh raise the computed eigenvalue, but by different amounts; in the cases considered here the mesh dominates.

The truncation effect is negligible at these box sizes. By (A2) the analytic ground state is $\exp(-\sqrt{\alpha}\, r^2/2)$, so writing it as $\exp(-r^2/2\sigma^2)$ the characteristic length is $\sigma = \alpha^{-1/4}$, which for $\alpha = 1/4$ is $\sqrt{2}$ and *not* $\sqrt{\alpha} = 1/2$. For a centered box of side L the amplitude at a face is $\exp(-\sqrt{\alpha}\, L^2/8)$; for $\alpha = 1/4$ and $L = 12$ that is $\exp(-9) \approx 1.2 \times 10^{-4}$ of the peak, so the exact solution very nearly satisfies the Dirichlet condition already. *If* one assumes the eigenvalue perturbation to behave like the square of that amplitude, the estimate is 10^{-8} or less; this is an order-of-magnitude heuristic and not a demonstrated bound. The direct evidence is in Table 7, where the box is varied at fixed mesh size.

That the discrepancy is instead a mesh effect is confirmed by refinement. Table 8 refines the 2D and 3D harmonic cases at fixed box size. The errors decrease as $\mathcal{O}(h^2)$, in agreement with Corollary 3, and the sequences converge to the *unbounded* values 1.0 and 1.5 rather than to any larger box eigenvalue. Refining the mesh at fixed L therefore recovers the analytic result, which would not occur if the box were the controlling error.

The discrepancies quoted in Table 6, namely 2.9% in 2D at $h = 0.5$ and 8.4% in 3D at $h = 0.857$, are thus discretization error at a deliberately coarse mesh. The distinction is not merely semantic: reducing them requires refining the mesh, not enlarging the box. Higher excited states oscillate faster and are therefore affected more: at these mesh sizes the relative error in the fourth eigenvalue is 2.0% in 2D and 6.8% in 3D, against 2.9% and 8.4% in the first. Box dimensions and mesh sizes are therefore reported explicitly (Appendix B).

A second, independent check separates the two effects by varying the box at *fixed* mesh size. Table 7 reports the first eigenvalue of the 2D and 3D harmonic cases for a sequence of boxes, with h held at 0.5 and 1.0 respectively. In 2D the value is constant to twelve significant digits from $L = 16$ onwards. In 3D, where the fixed mesh is much coarser, it is constant to nine significant digits from $L = 12$ onwards; the coarser plateau reflects the larger discretization error and not a larger truncation effect, since the box is already several times the characteristic length of the ground state. In both cases the offset from the unbounded analytic value (1 and 1.5) is a pure discretization error at the fixed mesh size: 2.9×10^{-2} at $h = 0.5$ in 2D and 1.7×10^{-1} at $h = 1.0$ in 3D. It does not shrink as the box grows. Over the range tested, truncation is therefore below the resolution of the computation. That is the direct evidence that the discrepancies of Table 6 must be attributed to the mesh. The statement is limited to the boxes and mesh sizes reported: a box smaller than $L = 12$ in 3D does show a truncation error, as the first row of Table 7 makes plain.

Remark 8 (Quadrature of the potential term) The error constant depends on how the potential enters the weighted mass matrix (11). Both implementations assemble $\int N_i\, (\mathcal{I}_h V)\, N_j$, that is, they use the *interpolant* $\mathcal{I}_h V$ of V in the finite element space rather than evaluating V exactly at the quadrature points, which is the natural convention in a general finite element code, where the potential is supplied as a nodal field. For the quadratic potentials considered here this introduces an additional $\mathcal{O}(h^2)$ contribution. It does not change the order of convergence, but it does increase the constant. Recomputing the 1D harmonic case ($L = 20$, 400

Table 7 Domain-size study at *fixed* mesh size: the first eigenvalue stops changing once the box is large enough.

2D harmonic, $h = 0.5$, $\alpha = 1/4$, unbounded $E_1 = 1$				
L	12	16	28	40
n_f	529	961	3 025	6 241
E_1	1.028585446	1.028585384	1.028585384	1.028585384
3D harmonic, $h = 1.0$, $\alpha = 1/4$, unbounded $E_1 = 1.5$				
L	6	9	12	18
n_f	125	512	1 331	4 913
E_1	1.737955701	1.670595303	1.670466988	1.670466985
ΔE_1	—	6.7×10^{-2}	1.3×10^{-4}	3×10^{-9}

Table 8 Refinement of the harmonic cases at fixed box size. Both sequences converge to the *unbounded* analytic values, so the leading error is discretization and not box truncation.

case	n	h	E_1	relative error	observed order
2D, $L = 20$, $\alpha = 1/4$	20	1.000	1.113645	1.14×10^{-1}	—
	40	0.500	1.028585	2.86×10^{-2}	1.99
	60	0.333	1.012719	1.27×10^{-2}	2.00
	80	0.250	1.007158	7.16×10^{-3}	2.00
3D, $L = 12$, $\alpha = 1/4$	10	1.200	1.744630	1.63×10^{-1}	—
	14	0.857	1.625507	8.37×10^{-2}	1.98
	18	0.667	1.576105	5.07×10^{-2}	1.99
	20	0.600	1.561688	4.11×10^{-2}	1.99

elements) with V evaluated exactly at the quadrature points raises the accuracy of the first eigenvalue from 1.000572892 to 1.000156243, against the exact value 1. The interpolant convention is retained, and the resulting difference is reported here so that its size is on record: 1.0% at the coarsest mesh of that sequence.

8.3 Eigensolution performance

Table 9 and Figure 3 compare the dense and sparse strategies on the 2D harmonic case, holding the mesh and the number of requested eigenpairs ($k = 4$) fixed. The discretization is identical in all rows; only the solver changes. The first column gives the dimension of the reduced matrices, that is, the number of free degrees of freedom after the elimination of Proposition 1; for a structured mesh of $n \times n$ elements on a square with Dirichlet data on the whole boundary this is $(n-1)^2$. The measurements were taken on a workstation with 31 GB of memory. The *algorithmic* behaviour was reproduced on a cluster node, in the sense that the same sequences of residual reduction and the same eigenvalue orders were observed there; the absolute times were not, and differ between the two machines by roughly a factor of three, as reported below.

Three details of the table need stating in the text rather than in the caption. The last row is not merely slow in dense arithmetic: by (22) the four resident arrays alone

Table 9 Eigensolution time for the 2D harmonic case, $k = 4$, against the number of free degrees of freedom n_f. Identical matrices in both columns.

n_f	dense `DSYGV` (s)	sparse ARPACK (s)	speed-up	dense memory
361	0.01	< 0.01	—	5.2 MB
1 521	0.58	0.01	58×	82.2 MB
6 241	41.06	0.05	821×	1.31 GB
14 161	491.60	0.12	4097×	6.64 GB
39 601	infeasible (51.2 GB)	0.41	—	51.2 GB

would need 51.2 GB, against the 31 GB of the workstation used. On a large-memory node it would fit, but the sparse route makes the question moot. In the first row the sparse time is below the resolution of the timer, so no speed-up is quoted there and none is annotated in Figure 3.

Repeating the sparse measurements on the same machine in a later session gave times a factor of three larger, with the same growth: 0.03, 0.15, 0.38 and 1.23 s at $n_f = 1521$, 6241, 14161 and 39601. The dense column moves by a similar factor. The ratios between the two routes therefore reproduce, while the absolute times do not, which is the point made earlier and the reason the table reports a single session rather than an average over several.

The measured speed-up grows faster than linearly. A factor 70 for a factor 9.3 in problem size would be consistent with an $\mathcal{O}(n_f^2)$ ratio, but four sizes and three ratios do not fix an asymptotic exponent. What the data establish is narrower than a complexity class: over the sizes tested, up to 39 601 free degrees of freedom, the sparse solve stayed below one second while the dense route exceeded the available memory at the largest case. The mechanism is understood: the sparse route never forms an $n_f \times n_f$ dense matrix, and it is what makes the largest case accessible at all. The comparison at the largest case is not a timing comparison at all: the dense route cannot be run. This is the practical argument for the sparse strategy: it is not merely faster, it extends the accessible problem class.

Thread-level parallelism does not substitute for the algorithmic change. Taking the single-thread sparse solve at $n_f = 39601$ as the reference, enabling multithreaded BLAS multiplies the time by 1.00, 1.02, 1.28 and 1.91 with one, two, four and eight threads respectively (minimum of three repetitions). Threading therefore gives no speed-up here: it degrades the result, because the extra workers contend for the same memory bandwidth while the sparse factorization itself remains sequential. The reference run is itself about twice as slow as the corresponding row of Table 9, because the two were taken in different sessions; this is why the argument rests on the ratios within one session rather than on absolute times. It reinforces the point of Section 5: the gain must come from the algorithm, not from the hardware.

8.4 Reproduction of published benchmarks

The strongest evidence that a solver is usable by a device community is that it reproduces results that community already trusts. Two benchmarks are therefore taken from

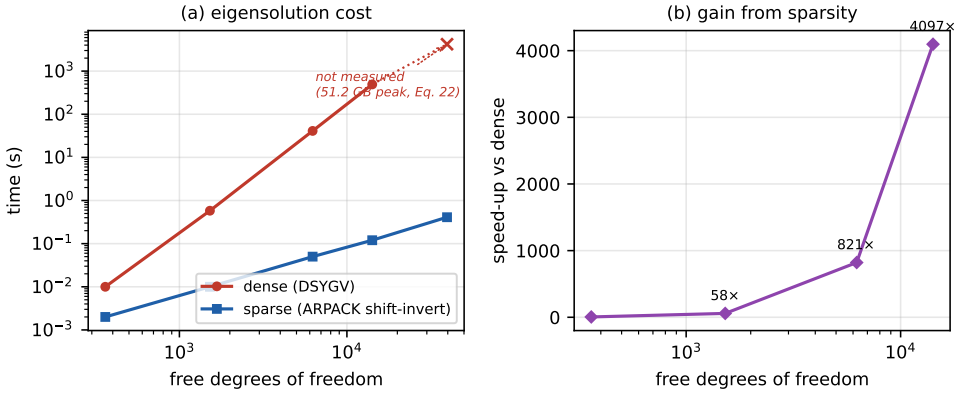


Fig. 3 (a) Eigensolution time against problem size for the dense and sparse strategies, on identical matrices. (b) Speed-up of the sparse strategy over the dense one, with measured values annotated.

recent issues of this journal and recompute them with the module, without adjusting any parameter to fit the published numbers.

Case A: the two-dimensional harmonic oscillator, a mathematical benchmark.

Salehi and Granpayeh [41] use the isotropic oscillator as the analytic test of a polar finite-difference time-domain code. Their Hamiltonian is $H = -\hbar^2\nabla^2/(2m) + \frac{1}{2}m\omega_0^2 r^2$ and their levels are $E_n = (n+1)\hbar\omega_0$. Ours is $H^* = -\nabla^2 + \alpha r^2$ with eigenvalues $\lambda_N = (2N+d)\sqrt{\alpha}$. The two are related by a single rescaling, and it is worth writing it out because the factor is easy to lose. Multiplying $H\psi = E\psi$ by $2m/\hbar^2$,

$$-\nabla^2\psi + \left(\frac{m\omega_0}{\hbar}\right)^2 r^2\psi = \frac{2mE}{\hbar^2}\,\psi, \tag{27}$$

so

$$\alpha = \left(\frac{m\omega_0}{\hbar}\right)^2, \qquad \lambda_N = \frac{2m}{\hbar^2}E_N. \tag{28}$$

With $m = \hbar = 1$ this reads $\alpha = \omega_0^2$ and $E_N = \lambda_N/2$, and the physical levels are compared against $(N+1)\omega_0$. Using $\alpha = (\omega_0/2)^2$ instead would silently describe an oscillator of frequency $\omega_0/2$, which is why the rescaling is stated rather than implied. Figure 4(a) places the computed levels on the analytic lines for $\omega_0 = 1$ and $\omega_0 = 2$, and Figure 4(b) shows the relative error falling as h^2. What the comparison adds beyond an eigenvalue check is the *multiplicity*: the levels carry 1, 2 and 3 states for $N = 0, 1, 2$, which is the signature of the isotropic two-dimensional oscillator and would not appear by accident. Figure 5 displays the corresponding densities, in the style of the eigenstate plots of the same reference. The two $N = 1$ panels there are not two different levels: they are an arbitrary orthonormal basis of the same two-dimensional eigenspace, and which basis the solver returns is a property of the eigensolver, not of the physics.

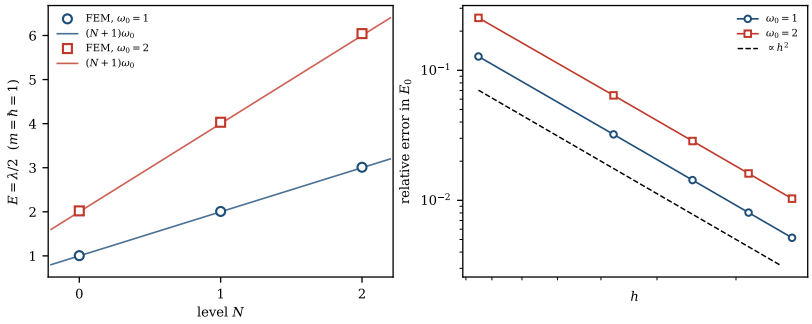


Fig. 4 Case A, after Salehi and Granpayeh [41]. (a) Computed physical levels $E_N = \lambda_N/2$ (markers) against the analytic $E_N = (N+1)\omega_0$ (lines) for $\omega_0 = 1$ and $\omega_0 = 2$, with $\alpha = \omega_0^2$ as in (28). (b) Relative error in the ground state for both frequencies, with the $\mathcal{O}(h^2)$ reference. Mesh 200×200 on $[-15, 15]^2$, $m = \hbar = 1$.

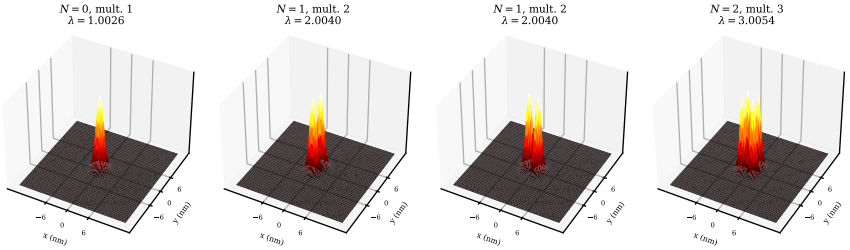


Fig. 5 Case A: $|\psi|^2$ for the first four eigenstates of $V = \alpha r^2$, $\alpha = 1/4$, $n = 200$, $h = 0.15$. The labels give the dimensionless eigenvalue λ of the scaled problem, *not* the physical energy $E = \lambda/2$; for $\alpha = 1/4$ the exact values are $\lambda = 1, 2, 3$. The two $N = 1$ panels span the same two-dimensional eigenspace.

Case B: the InGaN/GaN double quantum well.

En-nadir et al. [19] study a symmetric InGaN/GaN double well with a finite barrier under an applied field. The same structure is built: two 3 nm wells separated by a 3 nm barrier, with $E_g(\mathrm{InN}) = 0.78\,\mathrm{eV}$, $E_g(\mathrm{GaN}) = 3.42\,\mathrm{eV}$, bowing $b = 3.8\,\mathrm{eV}$ and $x_{\mathrm{In}} = 0.15$, which gives $E_g(\mathrm{InGaN}) = 2.540\,\mathrm{eV}$. A 70/30 band offset puts the conduction-band barrier at $V_0 = 0.616\,\mathrm{eV}$. Figure 6 shows the potential and the two lowest eigenfunctions: ψ_1 is symmetric, ψ_2 antisymmetric, and the pair is the bonding–antibonding doublet of the coupled wells.

The reference uses a position-dependent mass, Eq. (4) of En-nadir et al. [19]:

$$m^*(z) = \begin{cases} x\, m^*_{\mathrm{InN}} + (1-x)\, m^*_{\mathrm{GaN}} & \text{wells,} \\ m^*_{\mathrm{GaN}} & \text{barriers,} \end{cases} \tag{29}$$

with $m^*_{\mathrm{GaN}} = 0.19\, m_0$, $m^*_{\mathrm{InN}} = 0.11\, m_0$ and $x = 0.15$. This gives $0.178\, m_0$ in the wells against $0.190\, m_0$ in the barriers, a contrast of only 6.3%; the relative permittivity varies by 3.0% (9.97 against 9.68).

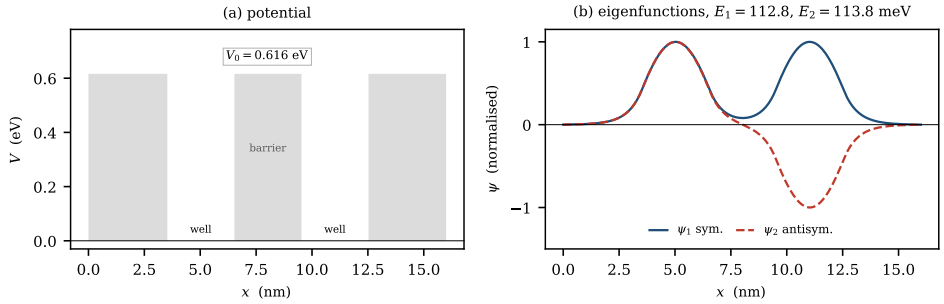


Fig. 6 Case B, after En-nadir et al. [19]. (a) potential of the symmetric InGaN/GaN double well, $V_0 = 0.616\,\mathrm{eV}$, zero at the well bottom; (b) the two lowest eigenfunctions, normalised to unit maximum. The levels quoted are those of the *constant*-mass solution, $m^* = 0.190\,m_0$; with the two-valued mass of (29) they become 116.2 and 117.2 meV, cf. Table 10.

Table 10 Case B with constant and position-dependent effective mass, in meV. Both codes use $n = 2560$ elements and the same potential; all levels are given to three decimals. The last column is $\delta = |E^{\mathrm{Elmer}} - E^{\mathrm{Python}}|/E^{\mathrm{Elmer}}$, the relative difference between the two implementations at each level.

	level (meV)				
mass model	E_1	E_2	E_3	shift	δ
constant, $0.190\,m_0$ (Elmer)	112.838	113.786	414.684	—	7×10^{-5}
variable, Eq. (29) (Elmer)	116.192	117.216	427.171	+3.0%	3×10^{-4}
constant (Python)	112.830	113.780	414.660	—	
variable (Python)	116.190	117.200	427.060	+3.0%	

The conservative form is solved $-\frac{\hbar^2}{2}\nabla\cdot[(1/m^*)\nabla\psi] + V\psi = E\psi$, so the stiffness matrix carries the factor $1/m^*$ and the natural energy unit is the bare-mass one,

$$\mathcal{E}_0 = \frac{\hbar^2}{2m_0L^2} = 0.0381\ \mathrm{eV} \quad (L = 1\ \mathrm{nm}), \tag{30}$$

a dimensionless level λ corresponding to $E = \lambda\mathcal{E}_0$. The barrier of 0.616 eV enters as $V_0/\mathcal{E}_0 = 16.17$, and a field F in $\mathrm{eV\,nm^{-1}}$ as $FL/\mathcal{E}_0$. The principal physical effects omitted, after En-nadir et al. [19], are the donor impurity and the dielectric mismatch, together with the self-polarization term that the reference also neglects; what is reproduced is the structure, the spectrum and the field dependence, not the absolute optical coefficients of the reference.

Table 10 reports both mass treatments. The 6.3% contrast moves the levels by 3.0%, which is two orders of magnitude above the discretization error of this case and therefore a real physical effect that a constant-mass model cannot capture. It is reported as a small but measurable difference rather than as a large one.

This is described as a simplified numerical cross-check rather than a reproduction, because the subset of physics it shares with the reference is partial. Figure 7 sweeps the

field angle, as the reference does in its Fig. 3, but the comparison required a correction of the reading of that figure adopted at first. The quantity En-nadir et al. [19] plot is not the gap between the two members of a doublet; their text and the surrounding equations give ΔE_{fi} as the $1S \to 2P$ *intersubband* separation of a donor, quoted as $0.493 - 0.109 = 0.384\,\mathrm{eV}$, and it varies by only 2 meV when the field is applied. The present structure has no donor, so the analogue is the separation between the two lowest well subbands, $\Delta E_{31} = E_3 - E_1 = 302\,\mathrm{meV}$, panel (a) of the figure. It too has a minimum at $\theta = \pi/2$, as the reference's quantity does at its critical angle, and it varies by 4.9 meV, or 1.6%, against 0.5% in the reference.

The doublet splitting $\Delta E_{21} = 0.95\,\mathrm{meV}$, panel (b), is a different quantity: it is large in absolute terms only under a longitudinal field, where the Stark shift separates the two wells' levels, and it says nothing about the intersubband transition the reference computes. Both are reported because an intuition about "the gap of a double well" suggests ΔE_{21}, while the reference's Fig. 3 is about ΔE_{31}, and conflating the two would overstate the agreement: the angular trends match, the magnitudes are not comparable, and the difference is the donor impurity.

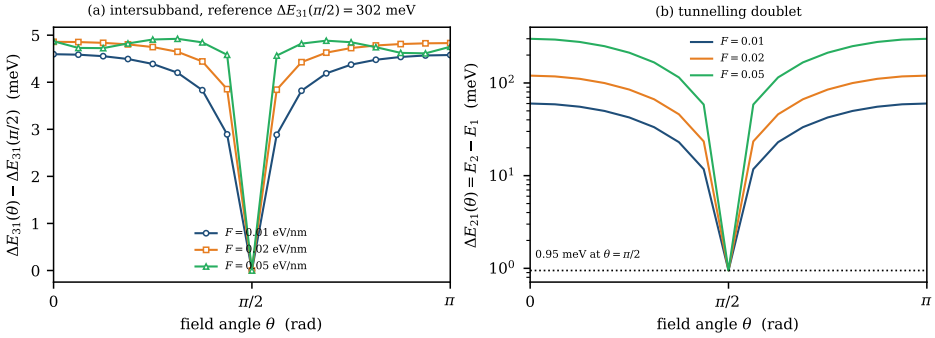


Fig. 7 Case B. The field makes an angle θ with the growth direction z, so the longitudinal component is $F\cos\theta$ and vanishes at $\theta = \pi/2$. *(a)* intersubband separation $\Delta E_{31} = E_3 - E_1$, plotted as its change from the value at $\theta = \pi/2$, which is 301.8 meV; the vertical axis is therefore a difference of differences, not the separation itself. *(b)* tunnelling doublet $\Delta E_{21} = E_2 - E_1$ on a logarithmic scale; the dotted line marks the value at $\theta = \pi/2$, 0.95 meV, and the three curves rise at $\theta = 0$ to 60, 120 and 300 meV for $F = 0.01$, 0.02 and $0.05\,\mathrm{eV\,nm^{-1}}$ respectively. The quantity in (b) is not the one plotted by the reference; see the text.

Scope.

Table 11 sets the solver against the two references.

Position-dependent mass.

The Hamiltonian of a heterostructure is not $-\hbar^2\nabla^2/(2m^*)\psi$ with a variable m^*; the hermitian form is $-\frac{\hbar^2}{2}\nabla\cdot[(1/m^*)\nabla\psi]$, with continuity of ψ and of $(1/m^*)\partial\psi/\partial n$ across an interface. The module supports this. An optional body-force keyword supplies m^* as a nodal field, in the same form as the potential, and the element loop multiplies the

Table 11 Feature coverage of the present solver and scope of the two reference benchmarks. "yes" means implemented *and* exercised here; "implemented, not exercised" means the code path exists without a demonstration in this paper; "—" means the feature is absent from that reference.

feature	this work	[19]	[41]
band-profile potential	yes	yes	—
spatially varying m^* (nodal)	yes	yes	—
dielectric mismatch	**no**	yes	—
donor impurity	**no**	yes	—
applied electric field	yes	yes	—
bound states and splittings	yes	yes	yes
optical absorption	no	yes	—
self-consistent Poisson	**no**	—	—
unstructured meshes	implemented, not exercised here	—	no
eigenvalue error estimate	yes	no	no
sparse eigensolver comparison	yes	no	no

stiffness by $1/m^*$ at the Gauss point. The default is $m^* = 1$, so every case of Table B1 is unchanged bit for bit, and Table 10 is the verification of the new path against the independent Python implementation. The keyword and its syntax are documented in the archived release rather than here.

The interface condition needs a precise statement, because two different things are easily confused. The weak form represents the flux condition $(1/m^*)\partial_n\psi$ continuous *naturally*: no penalty or constraint is imposed, and none is needed, because the jump of the coefficient is what the variational statement absorbs. What the code does not do is resolve that jump exactly. The coefficient is supplied as a nodal field and interpolated, so the discontinuity is replaced by a ramp over one element, and the code solves a *regularized* problem rather than the piecewise-constant one. The distinction matters for the claim being made: what Table 10 verifies is the implementation of the conservative operator, not the accuracy of the interface representation. For the 6.3% contrast of Case B the regularization is below the effect reported here; for a strong mass contrast the two errors would have to be separated, and the remedy would be to align the mesh with the interface and integrate each material separately, or to quantify the regularization by refinement.

A numerical consequence.

These benchmarks exposed a point that a step potential makes visible, and it is *not* a loss of regularity of the continuous solution. For a bounded piecewise-constant V in one dimension the eigenfunction satisfies $\psi \in H^2$: ψ and ψ' are continuous across the jump and $\psi'' = (V - E)\psi \in L^2$, so there is no loss of H^2 regularity to appeal to, and the $s = 1/2$ branch of Theorem 2 does not apply here. Something else does. The potential is integrated as its nodal interpolant $\mathcal{I}_h V$, so the jump is not resolved exactly by subdomain quadrature; it is represented through the nodal values, and how faithfully it is represented depends on where the interface falls relative to the nodes.

The two alignments must be described in terms of the nodal values, because a conforming Q_1 field on a conforming mesh is continuous: there is one value per node

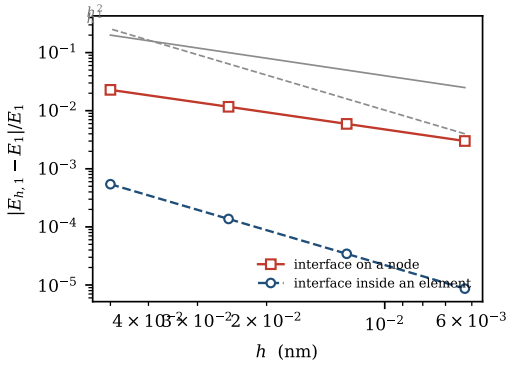


Fig. 8 Case B: eigenvalue error against h with the well interface on a mesh node (squares) and between nodes (circles). Orders one and two; the same limit.

and both adjacent elements use it. Consider the left interface of a well, with $V = V_0$ to its left and $V = 0$ to its right.

Interface on a node. The node at x_j receives the barrier value V_0. The element to its left is then constant at V_0, while the element to its right interpolates from V_0 at x_j to 0 at x_{j+1}, so the drop is spread over a full element on one side only. The representation is one-sided: exact at the node and wrong by V_0 at the next node to the right.

Interface inside an element. The element $[x_j, x_{j+1}]$ contains the jump at $x_j + \theta h$ with $0 < \theta < 1$. Both of its nodes carry V_0 and 0 respectively, so the interpolant ramps linearly over the whole element, from V_0 at x_j to 0 at x_{j+1}. The ramp is not symmetric about the physical position of the junction unless $\theta = 1/2$; what distinguishes this configuration from the previous one is not symmetry but the fact that both neighbouring elements are involved, each on its own side, rather than one element being constant.

What the data show. Figure 8 reports both sequences on the same structure. With the interface on a node the consecutive differences fall by a factor 1.94 and 1.97 per refinement, an order of 0.97; with the interface inside an element they fall by 3.96 and 3.98, an order of 1.99. The two limits agree to 3.4×10^{-5} in relative terms, so neither configuration produces a spurious value; only the rate differs.

What the data do not establish. The mechanism is stated here as an empirical attribution, not as a proved error expansion. The one-sided representation above is a description of the interpolant, and the natural conjecture is that its consistency error enters the Rayleigh quotient at leading order in the on-node case while the symmetric ramp cancels it in the other; a proof would require the expansion of $\int (\mathcal{I}_h V - V)\psi_h^2$ for a piecewise-constant V, which is not attempted here. A reader should treat the two rates as measured facts about this discretization and this coefficient representation, and not as a consequence of Theorem 2.

For a piecewise-constant potential the interface should therefore be kept off the nodes, or the potential integrated exactly by subdomains if the geometry permits. The same caution applies to $1/m^*$, with the additional caveat of the preceding paragraph.

9 Discussion

9.1 Two routes and their crossover

The results suggest a division of labor rather than a single recommended strategy:

In-framework route (dense, serial). Up to roughly 10^4 free degrees of freedom, which covers a wide range of one- and two-dimensional quantum-well and quantum-dot calculations at engineering accuracy, the dense in-framework solver is the simpler choice. It requires no matrix export and keeps the entire workflow inside the finite element framework, including post-processing. The $n_f = 6241$ case solves in 41 s, acceptable for interactive use.

Export route (sparse, parallel-capable). Beyond that size, and whenever several eigenpairs are needed or the mesh must be refined, the matrices are exported and solved externally. The route reproduces the dense result exactly (Table 5) and reduces storage from the $\mathcal{O}(n_f^2)$ of the dense route to $\mathcal{O}(\mathrm{nnz})$ for the matrices plus $\mathcal{O}(\mathrm{nnz}_L)$ for the factors of the shifted matrix and $\mathcal{O}(n_f\, n_{\mathrm{cv}})$ for the Arnoldi basis, as derived in Section 5.4; given a parallel sparse direct factorization it is potentially parallel-capable; no parallel scaling is measured here. The exported matrices are the *same* matrices: the route is a change of linear algebra, not of discretization, so the two solve the same discrete problem and agree to the reported precision (Table 5). They are not required to agree bit for bit, since two different algorithms on the same matrices need not round identically. The choice between them is one of computational resources.

The crossover follows from the storage model, and which model is used matters. Three figures are often conflated, and they differ by up to a factor of two.

The irreducible minimum, (21): the two reduced matrices alone, $16\, n_f^2$ bytes. The dense route becomes impractical there at $n_f \sim 10^4$ for a 1.6 GB budget and at $n_f \sim 4.0 \times 10^4$ for 25 GB. These two budgets are not physical memory sizes but working-set allowances: 1.6 GB is roughly what remains of a 2 GB container after the operating system and the runtime, and 25 GB is the corresponding allowance on a 32 GB node once the rest of the workflow is resident. They are meant as scenarios, not as hardware specifications.

What this implementation allocates, (22): four arrays, $16\,(n_{\mathrm{dof}}^2 + n_f^2)$ bytes. With $n_{\mathrm{dof}} \approx n_f$ this is close to $32\, n_f^2$, and the same budgets are reached at $n_f \sim 7.1 \times 10^3$ and $n_f \sim 2.8 \times 10^4$.

The three-matrix count used in an earlier version of this work, $24\, n_f^2$, sits between the two and corresponds to neither: it overstates the minimum and understates what the code does. It is retained nowhere in the paper except as a point of comparison in Section 8.3.

The thresholds quoted above are $\sqrt{\text{budget/coefficient}}$ from the storage model alone. They take no account of the workspace the eigensolver allocates, which only makes the dense route less attractive, and they are sensitive to the ratio n_{dof}/n_f, which for the square grids used here is $((n+1)/(n-1))^2$ and approaches one from above.

9.2 Limitations

Separable potentials. The component-wise convention of Section 6.3 restricts the module to potentials of the form $\sum_k V_k(x_k)$. Non-separable potentials, a Coulomb term, or the $1/r$ inter-electronic interaction used in quantum-dot models [5], would require extending the expression interface, not the solver core. Coupling to a self-consistent Poisson equation, as in the Schrödinger–Poisson problem [9, 40], is likewise outside the present scope, as is the treatment of open boundary conditions and polar geometries [41]. For time-dependent and nonlinear formulations, which lie outside the scope considered here, the reader is referred to [42, 43].

No parallel in-framework solver. The module is serial, and parallelism is obtained through the export route. The scaling of that route is limited by the availability of a parallel sparse direct factorization in the deployed PETSc build. On the installation used here, PETSc was configured without a parallel LU, so the shift-and-invert step could not be distributed and the driver was confined to one process. The driver is complete and verified at one process; only the linear-algebra backend is missing. This is regarded as a deployment issue rather than a methodological limitation, but it does mean that no parallel scaling numbers can be reported, and this is stated rather than omitted.

Coarse meshes for the confined cases. As shown in Section 8.2, the discrepancies in the harmonic cases are discretization error, not box truncation: the boxes are large enough that the truncation contribution is several orders of magnitude smaller. Reducing those discrepancies therefore requires mesh refinement, and the meshes used for the 2D and 3D harmonic cases are deliberately coarse. For potentials that decay slowly, or for boxes chosen too small, truncation would become the leading error and would then have to be separated from the discretization error.

Only the ground state is written to file. The module returns all requested eigenvalues but writes only the first eigenfunction to the field database, so the standard output modules produce a single mode per run.

10 Conclusions

This work presents a finite element framework for the stationary Schrödinger eigenvalue problem, implemented independently in two codes and verified in five ways. The main findings are the following.

1. The generalized symmetric eigenproblem $A\psi = EM\psi$ with $A = K + M_V$ is solved accurately by multilinear finite elements in one, two and three dimensions. Where the discrete form is the exact restriction of the continuous one, the conforming approximation yields eigenvalues that are upper bounds for the exact ones (Theorem 1); the implementation's nodal interpolation of the potential preserves that property for convex V on the tensor-product meshes used here, and is checked numerically for the others (Remark 4). The eigenvalue error obeys the classical $\mathcal{O}(h^{2s})$ estimate of Theorem 2; the predicted $\mathcal{O}(h^2)$ order is confirmed numerically over four refinements in 1D and three in 2D.

2. Elimination of the constrained degrees of freedom reproduces the constrained spectrum exactly (Proposition 1), whereas the common diagonal-shift treatment introduces spurious eigenvalues at $E = 1$.
3. Two independently written implementations, a Fortran module for Elmer and a Python reference code, agree to approximately 4×10^{-13} on identical problems. The agreement complements the analytic benchmarks: two codes that disagree would expose an implementation error, whereas two codes that agree cannot exclude an error shared by both.
4. The choice of eigensolution strategy dominates the computational cost. Replacing a dense solve by sparse shift-and-invert Arnoldi iteration yields speed-ups from 58× to 4097× over the sizes tested and replaces the $\mathcal{O}(n_f^2)$ storage of the dense matrices with $\mathcal{O}(\mathrm{nnz})$ for those matrices plus the fill-in of the factorization, extending the accessible problem size from roughly 10^4 to at least 4×10^4 degrees of freedom on the same hardware, in agreement with the cost model (25).
5. The in-framework solver is serial in its current form. Exporting the matrices and solving them with a sparse distributed library is the route to parallelism; whether it is faster on a given problem depends on the availability of a parallel sparse direct solver, which was not available and was therefore not measured. No parallel scaling is claimed.
6. Structural checks requiring no external reference, expected degeneracies, mass-orthonormality, the virial theorem, and monotonicity of the eigenvalues under refinement, are inexpensive and detect errors that analytic comparisons at a single resolution can miss.
7. The solver reproduces the two-dimensional oscillator benchmark of Salehi and Granpayeh [41] to 0.4%, and performs a partial cross-check of the InGaN/GaN double well of En-nadir et al. [19]: structure, spectrum and angular trend, without the donor impurity or the dielectric mismatch. The second required adding position-dependent mass in the conservative form $-\frac{\hbar^2}{2}\nabla \cdot [(1/m^*)\nabla\psi]$ with m^* a nodal field; on that structure, where the mass contrast is 6.3%, the levels move by 3.0%, two orders of magnitude above the discretization error. What the comparison does *not* establish is quantitative agreement with the reference, which includes a donor impurity and a dielectric mismatch that are not modelled. The point being made is about the boundary of the verification, not about the host package.

Acknowledgments. The author thanks CSC – IT Center for Science for developing and maintaining Elmer as open-source software. Computational resources were provided by the high-performance computing cluster of the CEAMAZON - Federal University of Pará.

Declarations

- Funding: The Article Processing Charge (APC) of this publication is funded by the Coordination for the Improvement of Higher Education Personnel, CAPES (ROR identifier: 00x0ma614). For the purpose of open access, the author has applied a Creative Commons CC BY licence to any accepted version of the article. This

is the acknowledgement text required by Art. 9, item X of CAPES Ordinance no. 224/2026.

- Conflict of interest: The author declares that he has no conflict of interest.
- Ethics approval: Not applicable.
- Consent to participate: Not applicable.
- Consent for publication: Not applicable.
- Data availability: The archived release (https://doi.org/10.5281/zenodo.23366621, version DOI https://doi.org/10.5281/zenodo.23821744) contains the Gmsh source and Elmer-format mesh for each case, the solver input files for the seven benchmarks of Table B1 and for the two benchmarks of Table B2, and the Python scripts of `verification/` that reproduce Tables 1–5.
- Materials availability: Not applicable.
- Code availability: Both implementations are open source, together with the case definitions, the meshes and the container definition. The version used for every result in this paper is release `v1.1.0`, commit `738a103`, archived under the concept DOI https://doi.org/10.5281/zenodo.23366621, which always resolves to the most recent release; the release itself is frozen at that tag and commit, and the archived snapshot carries its own version DOI, https://doi.org/10.5281/zenodo.23821744. Cite the concept DOI to refer to the software in general and the version DOI to refer to exactly the state that produced these results. The software is distributed under GPL-3.0-or-later and is developed at https://github.com/wellfonseca/schrodinger-fem [44].
- Author contribution: The author is the sole contributor to this work: formulation, implementation, verification, and writing of the manuscript.

Appendix A Analytic reference solutions

All expressions are stated in the unit convention of (3), that is, with a unit coefficient on the Laplacian. The relation to physical units is given in Section 2.2.

Infinite well, one dimension. For $V = 0$ on $[-L/2, L/2]$ with $\psi(\pm L/2) = 0$, $E_n = (n\pi/L)^2$ for $n = 1, 2, \ldots$

Infinite box, d dimensions. For a d-dimensional box of side L with homogeneous Dirichlet conditions on all faces, $E_{n_1 \ldots n_d} = (\pi^2/L^2) \sum_{k=1}^{d} n_k^2$.

Isotropic harmonic oscillator. For $V = \alpha r^2$ with $r^2 = \sum_k x_k^2$ the problem separates and

$$E_N = (2N + d)\sqrt{\alpha}, \qquad N = 0, 1, 2, \ldots, \tag{A1}$$

where N is the *total* excitation number and $E_0 = d\sqrt{\alpha}$ is the ground-state energy. The normalized ground state is

$$\psi_0(\mathbf{r}) = \left(\frac{\sqrt{\alpha}}{\pi}\right)^{d/4} \exp\left(-\frac{\sqrt{\alpha}}{2}\, r^2\right), \tag{A2}$$

which follows from $\int_{\mathbb{R}} \exp(-ax^2)\mathrm{d}x = \sqrt{\pi/a}$ with $a = \sqrt{\alpha}$ applied in each coordinate. The prefactor is $(\sqrt{\alpha}/\pi)^{d/4}$ and *not* $(\alpha/\pi)^{d/4}$; the two coincide only at $\alpha = 1$, which

is why the case plotted in Figure 2 does not distinguish them, whereas the 2D and 3D benchmark cases use $\alpha = 1/4$ and do.

The multiplicity of the level with total excitation N is

$$g_N = \binom{N + d - 1}{d - 1}, \tag{A3}$$

so the first excited level, $N = 1$ and $E = d + 2$ for $\alpha = 1$, has multiplicity d, doubly degenerate in 2D and triply degenerate in 3D. This is the property used as a symmetry check in Section 7.5, and it is confirmed by the 2.669307913 eigenvalue appearing three times in the 3D row of Table 6. The multiplicity $d(d + 1)/2$ belongs to the *second* excited level, $N = 2$, $E = d + 4$, and not to the first.

Morse potential. For $V(x) = D_e(1 - e^{-a(x-x_0)})^2 - D_e$ the spectrum consists of a finite number of bound states, with

$$E_\nu = 2a\sqrt{D_e}\,(\nu + \tfrac{1}{2}) - a^2(\nu + \tfrac{1}{2})^2 - D_e, \qquad \nu = 0, 1, \ldots \tag{A4}$$

for the full line. The case used here is $D_e = 10$, $a = 0.5$ and $x_0 = 3$, on the truncated domain $[0, 10]$; at $x = 0$ the potential has already reached 111.2, so the left wall is effectively infinite, while at $x = 10$ it stands at -0.595. This serves as a quantitative check (Section 7). Because (A4) holds on the full line and the computation is performed on $[0, 10]$, the comparison carries a truncation contribution as well as a discretization one. On this mesh and box the computed levels lie above the closed-form values by 3.3×10^{-4} to 3.5×10^{-3} in absolute terms. *Relative* to each level the deviations are 3.9×10^{-5}, 8.9×10^{-5}, 2.0×10^{-4} and 1.7×10^{-3} for $\nu = 0, 1, 2, 3$: the range is 4×10^{-5} to 1.7×10^{-3}, the largest value arising in the $\nu = 3$ level because it is the smallest in magnitude, not because it is the worst resolved. A Richardson extrapolation of the mesh sequence agrees with the closed-form values to 10^{-8}, so the bulk of that difference is the discretization at $n = 400$ and the truncation is the smaller part.

Appendix B Benchmark case parameters

Table B1 Parameters of the benchmark cases. All meshes are structured and were generated with Gmsh.

case	dimension	domain	potential	elements
1D infinite well	1	$[-20, 20]$	$V = 0$	250
1D harmonic	1	$[-10, 10]$	$V = x^2$	400
1D Morse	1	$[0, 10]$	$V = 10(e^{-2a(x-x_0)} - 2e^{-a(x-x_0)})$, $a = 0.5$, $x_0 = 3$	400
1D double well	1	$[-10, 10]$	$V = 0.01(x^2 - 4)^2$	400
1D sinusoidal	1	$[-10, 10]$	$V = 0.5\cos(2\pi x/10)$	400
2D harmonic	2	$[-10, 10]^2$	$V = 0.25(x^2 + y^2)$	40×40
3D harmonic	3	$[-6, 6]^3$	$V = 0.25(x^2 + y^2 + z^2)$	14^3

In all cases homogeneous Dirichlet conditions are imposed on the whole boundary. The potentials of the 2D and 3D harmonic cases are specified through the component-wise convention of Section 6.3.

Table B2 Parameters of the two reproduced benchmarks of Section 8.4, specified here so that the reported numbers can be regenerated without consulting the code. Case B is run with the mass constant at $0.190\,m_0$ and with the two-valued profile of the last-but-one row.

quantity	Case A	Case B
equation	$-\nabla^2\psi + \alpha r^2\psi = \lambda\psi$	see the two operator rows below
dimension	2	1
domain	$[-15, 15]^2$	$[0, 16]$ nm
potential	$\alpha = \omega_0^2$, $\omega_0 \in \{1, 2\}$	$V_0 = 0.616$ eV on $[0, 3.53]$, $[6.53, 9.53]$, $[12.53, 16]$; 0 in the wells
α / V_0	$\alpha = 1$ and 4	$V_0/\mathcal{E}_0 = 16.17$
mesh	200×200 elements, $h = 0.15$	2560 elements, $h = 6.25 \times 10^{-3}$ nm
mass	$m = \hbar = 1$	$0.190\,m_0$ everywhere, or $0.178\,m_0$ in the wells
operator, constant	—	$-\psi'' + \tilde{V}(x)\psi = \lambda\psi$, with $\tilde{V} = V/\mathcal{E}_0$
operator, variable	—	$-\frac{\mathrm{d}}{\mathrm{d}x}\left[\frac{m_0}{m^*(x)}\frac{\mathrm{d}\psi}{\mathrm{d}x}\right] + \tilde{V}\psi = \lambda\psi$
energy scale	$E = \lambda/2$	$\mathcal{E}_0 = 0.0381$ eV
reference	[41]	[19]

Two energy units appear in this paper and they must not be confused, because the same physical barrier has different numerical values in each.

The bare-mass unit. With the stiffness written as $\int (1/m^*)\nabla N_i \cdot \nabla N_j$, the natural unit is $\mathcal{E}_0 = \hbar^2/(2m_0L^2) = 0.0381\,\mathrm{eV}$ and the barrier enters as $V_0/\mathcal{E}_0 = 16.17$. This is the convention of Table B2 and of the case files.

The effective-mass unit. If the whole equation is divided by $1/m^*$ for a constant $m^* = 0.19\,m_0$, as the unit-mass Python implementation does, the stiffness becomes $\int \nabla N_i \cdot \nabla N_j$ and the unit is $\mathcal{E} = \hbar^2/(2m^*L^2) = \mathcal{E}_0/0.19 = 0.2005\,\mathrm{eV}$, with $V_0/\mathcal{E} = 3.07$. This is the convention of Table 10 and of the figures.

The two are the same physics, since $16.17 \times 0.0381 = 3.07 \times 0.2005 = 0.616\,\mathrm{eV}$, but a numerical value of the barrier means nothing without the convention it belongs to. The field of Figure 7 enters as $FL/\mathcal{E}$ with F in $\mathrm{eV\,nm^{-1}}$ and $\mathcal{E}$ the unit of the code that is running.

The constant-mass choice and the absence of the donor impurity are the two physical simplifications with respect to [19].

References

[1] Malvezzi, A.L.: An introduction to numerical methods in low-dimensional quantum systems. Brazilian Journal of Physics **33**(1), 55–72 (2003)

[2] Courant, R., Hilbert, D.: Methods of Mathematical Physics, Vol. I. Interscience Publishers, New York (1953)

[3] Datta, S.: Quantum Transport: Atom to Transistor. Cambridge University Press, Cambridge (2005)

[4] Harrison, P.: Quantum Wells, Wires and Dots: Theoretical and Computational Physics of Semiconductor Nanostructures, 2nd edn. John Wiley & Sons, Chichester (2005)

[5] Caruso, F., Oguri, V., Silveira, F.: Numerical solutions for a two-dimensional quantum dot model. Brazilian Journal of Physics **49**(4), 432–437 (2019) https://doi.org/10.1007/s13538-019-00656-7

[6] Strang, G., Fix, G.J.: An Analysis of the Finite Element Method. Prentice-Hall, Englewood Cliffs, NJ (1973)

[7] Babuška, I., Osborn, J.: Eigenvalue problems. In: Ciarlet, P.G., Lions, J.L. (eds.) Handbook of Numerical Analysis, Vol. II: Finite Element Methods (Part 1), pp. 641–787. North-Holland, Amsterdam (1991)

[8] Boffi, D.: Finite element approximation of eigenvalue problems. Acta Numerica **19**, 1–120 (2010) https://doi.org/10.1017/S0962492910000012

[9] Wang, W., Shu, C.-W.: The WKB local discontinuous Galerkin method for the simulation of Schrödinger equation in a resonant tunneling diode. Journal of Scientific Computing **40**(1-3), 360–374 (2009) https://doi.org/10.1007/s10915-008-9237-1

[10] Negulescu, C.: Numerical analysis of a multiscale finite element scheme for the resolution of the stationary Schrödinger equation. Numerische Mathematik **108**(4), 625–652 (2008) https://doi.org/10.1007/s00211-007-0132-8

[11] Sek, L., Zaiz, I., Miloudi, A.: Finite-element modeling of fractional quantum transport and optical response in coupled quantum dot–ring nanostructures with donor impurities. Journal of Computational Electronics **25** (2026) https://doi.org/10.1007/s10825-026-02533-4

[12] Ram-Mohan, L.R.: Finite Element and Boundary Element Applications in Quantum Mechanics. Oxford University Press, Oxford (2002)

[13] Fiori, G., Iannaccone, G.: Code for the 3d simulation of nanoscale semiconductor devices, including drift-diffusion and ballistic transport in 1d and 2d subbands, and 3d tunneling. Journal of Computational Electronics **4**, 63–66 (2005) https://doi.org/10.1007/s10825-005-7108-7

[14] Baumgartner, O., Stanojevic, Z., Schnass, K., Karner, M., Kosina, H.: Vsp—a quantum-electronic simulation framework. Journal of Computational Electronics **12**, 701–721 (2013) https://doi.org/10.1007/s10825-013-0535-y

[15] Pourghaderi, M.A., Magnus, W., Sorée, B., De Meyer, K., Meuris, M., Heyns, M.: General 2d schrödinger-poisson solver with open boundary conditions for nanoscale cmos transistors. Journal of Computational Electronics **7**, 475–484 (2008) https://doi.org/10.1007/s10825-008-0257-8

[16] Cheng, C., Liu, Q.H., Lee, J.-H., Massoud, H.Z.: Spectral element method for the schrödinger-poisson system. Journal of Computational Electronics **3**, 417–421 (2004) https://doi.org/10.1007/s10825-004-7088-z

[17] Sousa, A.R.M.d., Nunes, M.V.A., Fonseca, W.d.S., Araujo, R.C.F., Lima, D.d.S.: Magneto-thermo-structural analysis of power transformers under inrush and short circuit conditions. Energies **14**(11), 3266 (2021) https://doi.org/10.3390/en14113266

[18] Fonseca, W.d.S., Araújo, R.C.F., Silva, M.d.O.e., Cruz, D.O.d.A.: Analysis of the magnetohydrodynamic behavior of the fully developed flow of conducting fluid. Energies **14**(9), 2463 (2021) https://doi.org/10.3390/en14092463

[19] En-nadir, R., El Ghazi, H., Jorio, A., Zorkani, I., Kiliç, H.Ş.: Intersubband optical absorption in (in,ga)n/gan double quantum wells considering applied electric field effects. Journal of Computational Electronics **21**, 111–118 (2022) https://doi.org/10.1007/s10825-021-01830-4

[20] Saha, S., Kumar, J.: Fully self-consistent analysis of iii-nitride quantum cascade structures. Journal of Computational Electronics **15**, 1531–1540 (2016) https://doi.org/10.1007/s10825-016-0911-5

[21] Lehoucq, R.B., Sorensen, D.C., Yang, C.: ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods. Society for Industrial and Applied Mathematics, Philadelphia, PA (1998)

[22] Saad, Y.: Numerical Methods for Large Eigenvalue Problems, Revised edn. Society for Industrial and Applied Mathematics, Philadelphia, PA (2011)

[23] Davis, T.A.: Direct Methods for Sparse Linear Systems. Society for Industrial and Applied Mathematics, Philadelphia, PA (2006)

[24] Oberkampf, W.L., Roy, C.J.: Verification and Validation in Scientific Computing. Cambridge University Press, Cambridge (2010)

[25] Roache, P.J.: Verification and Validation in Computational Science and Engineering. Hermosa Publishers, Albuquerque, NM (1998)

[26] Salari, K., Knupp, P.: Code verification by the method of manufactured solutions. Technical Report SAND2000-1444, Sandia National Laboratories (2000). https://doi.org/10.2172/759450

[27] Gfrerer, M.H., Schanz, M.: Code verification examples based on the method of manufactured solutions for kirchhoff–love and reissner–mindlin shell analysis. Engineering with Computers **34**(4), 775–785 (2017) https://doi.org/10.1007/s00366-017-0572-4

[28] Paes, A.C.J., Abe, N.M., Serrão, V.A., Passaro, A.: Simulations of plasmas with electrostatic PIC models using the finite element method. Brazilian Journal of Physics **33**(2), 411–417 (2003) https://doi.org/10.1590/s0103-97332003000200046

[29] CSC – IT Center for Science: Elmer Finite Element Solver, version 9.0. https://github.com/ElmerCSC/elmerfem (2024)

[30] Seraide, R.M., Hai, G.-Q.: Low-temperature electron mobility in parabolic quantum wells. Brazilian Journal of Physics **32**(2a), 344–346 (2002) https://doi.org/10.1590/s0103-97332002000200026

[31] Freire, H.J.P., Egues, J.C.: Subband structure of ii-vi modulation-doped magnetic quantum wells. Brazilian Journal of Physics **33**(1) (2003) https://doi.org/10.1590/s0103-97332003000100016

[32] Vandenberghe, W.G., Fischetti, M.V., Van Beeumen, R., Meerbergen, K., Michiels, W., Effenberger, C.: Determining bound states in a semiconductor device with contacts using a nonlinear eigenvalue solver. Journal of Computational Electronics **13**, 753–762 (2014) https://doi.org/10.1007/s10825-014-0597-5

[33] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D.: LAPACK Users' Guide, 3rd edn. Society for Industrial and Applied Mathematics, Philadelphia, PA (1999)

[34] Ericsson, T., Ruhe, A.: The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems. Mathematics of Computation **35**(152), 1251–1268 (1980) https://doi.org/10.2307/2006390

[35] Balay, S., Abhyankar, S., Adams, M.F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E., Dalcin, L., Dener, A., et al.: PETSc/TAO users manual. Technical Report ANL-21/11, Revision 3.19, Argonne National Laboratory (2023). https://doi.org/10.2172/1968587

[36] Hernandez, V., Roman, J.E., Vidal, V.: SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. ACM Transactions on Mathematical Software **31**(3), 351–362 (2005) https://doi.org/10.1145/1089014.1089019

[37] Geuzaine, C., Remacle, J.-F.: Gmsh: A three-dimensional finite element mesh

generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering **79**(11), 1309–1331 (2009) https://doi.org/10.1002/nme.2579

[38] Tabora, J.M.n., Tshoombe, B.K., Fonseca, W.d.S., Tostes, M.E.d.L., Matos, E.O.d., Bezerra, U.H., Silva, M.d.O.e.: Virtual modeling and experimental validation of the line-start permanent magnet motor in the presence of harmonics. Energies **15**(22), 8603 (2022) https://doi.org/10.3390/en15228603

[39] Huang, T.-M., Li, T., Lin, J.-W., Lin, W.-W., Tian, H.: Structure-preserving methods for computing complex band structures of three dimensional photonic crystals. Journal of Scientific Computing **83**(2) (2020) https://doi.org/10.1007/s10915-020-01220-1

[40] Claus, M., Mothes, S., Blawid, S., Schröter, M.: Coos: a wave-function based schrödinger–poisson solver for ballistic nanotube transistors. Journal of Computational Electronics **13**(3), 689–700 (2014) https://doi.org/10.1007/s10825-014-0588-6

[41] Salehi, M., Granpayeh, N.: Numerical solution of the schrödinger equation in polar coordinates using the finite-difference time-domain method. Journal of Computational Electronics **19**(1), 91–102 (2020) https://doi.org/10.1007/s10825-020-01446-0

[42] Li, D., Li, X., Sun, H.-w.: Optimal error estimates of sav crank–nicolson finite element method for the coupled nonlinear schrödinger equation. Journal of Scientific Computing **97**(3) (2023) https://doi.org/10.1007/s10915-023-02384-2

[43] Liu, C.-S.: A positivity preserving iterative method for finding the ground states of saturable nonlinear schrödinger equations. Journal of Scientific Computing **84**(3) (2020) https://doi.org/10.1007/s10915-020-01297-8

[44] Fonseca, W.d.S.: schrodinger-fem: a finite element solver for the stationary Schrödinger eigenvalue problem in Elmer, version 1.1.0. https://github.com/wellfonseca/schrodinger-fem, tag `v1.1.0`, commit `738a103`. GPL-3.0-or-later. Concept DOI https://doi.org/10.5281/zenodo.23366621; version DOI https://doi.org/10.5281/zenodo.23821744 (2026)