Claimed Versus Verified Capability in Computational Chemistry Software: An Evidence-Tier Review of 31 Programs, With an AI-Native Self-Hosted Platform as a Case Study

Ts. Dr. LEONG YEE ROCK, Forward Deployed Engineer, VYROX AI, Kuala Lumpur, Malaysia

ARTICLE TYPE	Review; documentation-based comparison, no programs executed
SERIES	VYROX AI Research Series, Paper 04
PUBLISHED	13 November 2026
STATUS	Not peer reviewed; independent re-coding of the matrix pending
CITE AS	Leong, Y. R. (2026). Claimed versus verified capability in computational chemistry software: An evidence-tier review of 31 programs, with an AI-native self-hosted platform as a case study (Version 1) [Review]. VYROX AI. https://www.vyrox.com/research-claimed-vs-verified-chemistry-software

ABSTRACT

Background. Computational chemistry programs differ widely in method coverage, verification, performance, licensing and deployment, and new AI-assisted platforms are marketed through capability matrices that are rarely audited. **Methods.** We compared AiMolSim, a proprietary, self-hosted, AI-assisted platform with an in-house C++ and CUDA engine (version 1.10.12), with 30 established programs, using a public 99-capability matrix with every cell linked to official documentation, the platform's 164-method validation register, vendor documentation and peer-reviewed software papers. Programs were scored uniformly (Yes = 1, Partial = 0.5) and the ranking was tested under six scenarios. **Results.** On the published matrix AiMolSim scored 84.5 of 99, ahead of the Schrodinger Suite (73.0), AMS (70.0) and ORCA (64.5), and tied with ORCA, Q-Chem and TURBOMOLE (20.5 of 23) in core electronic structure. Only 37 of its 164 methods (22.6%) reproduce printed literature values, with 4 more checked against exact mathematics; two of ten domains have no check of either kind. Scoring the 28 scientific entries that its own notes mark as unchecked as partial moves AiMolSim to fourth (58.0); under a strict lower bound it ranks thirteenth (39.5). Its in-house engine replaces BLAS, LAPACK, FFT, integral and functional libraries, offers an optional fixed-order mode with bitwise-identical reruns and is licence-independent, but its published matrix-multiplication timings show one shape at 1.13 to 1.20 times Intel MKL speed and other sizes slower, and its GPU kernels are not yet in use. No settings-matched cross-code timing benchmark exists for any of the programs. **Conclusions.** Feature counts measure implemented breadth and the criteria chosen by whoever builds the matrix, not verified accuracy or performance. We propose reporting capability in three evidence tiers (implemented, internally consistent, reproduces published values) and give a use-case selection guide.

Keywords: computational chemistry software; quantum chemistry; software comparison; reproducibility; validation; artificial intelligence; provenance; self-hosted computing

The Paper in Four Sentences

On a sourced 99-capability matrix of 31 programs, the AI-native AiMolSim platform ranks first by implemented breadth and ties ORCA, Q-Chem and TURBOMOLE in core electronic structure. Only 37 of its 164 methods reproduce published values, and when its own unchecked entries are scored as partial its rank falls to fourth, or to thirteenth under a strict lower bound. Established codes lead on verified accuracy, GPU and multi-node performance and scripting, while AiMolSim adds approval before execution, per-result provenance, self-hosted data control and an in-house engine and math kernel that give bitwise-deterministic results but are not yet competitive with vendor libraries in speed. Capability comparisons should therefore report evidence tiers, not feature counts.

1. Introduction

Computational chemistry software has grown over five decades from a handful of Hartree-Fock codes into an ecosystem of general-purpose quantum chemistry packages, periodic electronic structure codes, semiempirical and tight-binding programs, molecular dynamics engines and cheminformatics toolkits [1-27]. Many of these programs were described in a coordinated series of software overviews in The Journal of Chemical Physics in 2020 and 2021 [2-10, 14, 18, 26], and their evolution has been reviewed in relation to the scientific problems they were built to solve [28].

Two developments now change how such software is chosen. First, reproducibility has become a primary concern: even for a well-defined quantity such as the equation of state of elemental solids, agreement between codes had to be established through a dedicated community effort [30], and data infrastructures such as QCArchive [31], AiiDA [32] and the Atomic Simulation Environment [33] were created partly to make calculations traceable and repeatable in line with the FAIR principles [29]. Second, large language models (LLMs) are being coupled to scientific tools, from autonomous experimental planning [35] to tool-augmented chemistry agents [36] and a broad set of materials and chemistry applications [42]. Commercial products are appearing that combine an electronic structure engine with an AI assistant, team workspaces, approval steps and per-result provenance.

These products are usually marketed through capability matrices that place a new platform beside established programs. Such matrices are useful, but they carry three structural risks: the criteria are chosen by the party being compared, a "Yes" can mean "implemented" rather than "verified", and documentation of mature and new programs is not equally deep. No review that we are aware of has examined these risks using a fully sourced, cell-level matrix.

This review has three aims: (i) to compare one AI-native, self-hosted platform, AiMolSim, with 30 established programs on a uniform, documentation-based basis; (ii) to test how sensitive the resulting ranking is to the choice of criteria and to the evidence behind each claim; and (iii) to derive practical guidance for users selecting software and for developers reporting capability.

2. Background: The Program Landscape

We group the 31 programs into seven classes (Table 1). General-purpose molecular quantum chemistry programs, open source (PySCF [3], Psi4 [4], NWChem [6], GAMESS [7], OpenMolcas [11, 51]) and proprietary (ORCA [1, 2], Gaussian 16, Q-Chem [5], TURBOMOLE [9], Molpro [8], AMS/ADF [12], Jaguar [13], Spartan), provide Hartree-Fock, density functional theory (DFT) [34, 38] with empirical dispersion corrections [37], perturbation and coupled cluster theory

and molecular properties. Periodic codes (VASP [16], Quantum ESPRESSO [14], ABINIT [15], CP2K [10]) target solids and surfaces with plane waves or mixed bases. Approximate electronic structure is covered by xtb [17], DFTB+ [18] and MOPAC [27]. Classical simulation is covered by OpenMM [19], GROMACS [20], LAMMPS [21], AmberTools [22], CHARMM [25] and NAMD [26]. Open Babel [23], RDKit and Avogadro 2 [24] provide cheminformatics, file conversion and visualisation. The Schrodinger Suite is an integrated commercial platform that includes Jaguar.

AiMolSim is the subject platform. According to its public documentation, it runs computations only on servers the customer owns (no hosted compute), uses an in-house engine written in C++ and CUDA without third-party numerical libraries, exposes its functions through a browser interface only (no public scripting API or command line), and uses an AI assistant that drafts a computational plan which a person must approve before any job runs. The engine lists 164 methods in ten domains, each tagged Checked (reproduces printed values from the source paper), Partial (code exists; unchecked parts badged), Unverified (built, no printed check) or In build. Cross-engine agreement is reported against PySCF, Psi4, RDKit and Open Babel at a stated tolerance of 1e-8 Eh for energies, with the referee programs run in a separate test environment.

Table 1. The 31 programs reviewed, grouped by class.

Class	Programs
Molecular quantum chemistry, open source	PySCF, Psi4, NWChem, GAMESS, OpenMolcas
Molecular quantum chemistry, proprietary	ORCA, Gaussian 16, Q-Chem, TURBOMOLE, Molpro, AMS (ADF), Jaguar, Spartan
Periodic and materials	VASP, Quantum ESPRESSO, ABINIT, CP2K
Semiempirical and tight binding	xtb, DFTB+, MOPAC
Molecular mechanics and dynamics	OpenMM, GROMACS, LAMMPS, AmberTools, CHARMM, NAMD
Cheminformatics and visualisation	RDKit, Open Babel, Avogadro 2
Integrated platforms	Schrodinger Suite; AiMolSim (subject of review)

3. Methods

3.1 Evidence Sources

The primary evidence was the public capability matrix maintained on the AiMolSim website (aimolsim.com/compare), which compares 31 programs across 99 capabilities in seven categories. Each non-AiMolSim cell carries a status (Yes, Partial, Planned, No, Outside the program, Not confirmed) and a short note, and is linked to the program's official manual, feature list or repository (aimolsim.com/compare/sources). Cells for other programs were dated 26 to 29 October 2026; AiMolSim cells were dated 13 November 2026 and reflect engine 1.10.12. The second source was the platform's method index, validation register and machine-readable summary (aimolsim.com/methods, /validation and /llms.txt), retrieved on 13 November 2026. All pages were archived as retrieved and the parsed matrix is provided as supplementary data.

We did not run any of the programs. This is a documentation-based review: it measures what each program documents, not how well it performs. Section 6 discusses this limitation. To complement the matrix, we searched official release notes, manuals, licence texts and peer-

reviewed software papers for each program (searches run on 13 November 2026) for version, licence, performance and documented limitations, and the AiServa platform documentation for the subject's deployment model [W1, W2]. We also reviewed web, cloud and AI-agent platforms that compete with the subject (Section 4.13) and the literature on numerical reproducibility between programs (Section 4.12).

The 31 programs are those in the published matrix, whose selection was made by the subject's developer. They cover the main program classes, but several widely used codes are not included, among them TeraChem and QUICK (GPU-native quantum chemistry), CFOUR and MRCC (high-order coupled cluster), Dalton and DIRAC (response and relativistic methods), FHI-aims, GPAW, SIESTA and CASTEP (periodic electronic structure), BAGEL, Desmond and BIOVIA Materials Studio. Conclusions about "the field" are therefore limited to the 30 programs compared.

3.2 Scoring

Each cell was scored Yes = 1, Partial = 0.5 and all other statuses = 0, the rule used by the matrix itself. "Outside the program" (a capability reached through an interface to another code) was scored 0, because the comparison concerns what each program provides on its own. Category scores and totals were computed for every program.

3.3 Sensitivity Scenarios

To separate the effect of the criteria and of the evidence behind each claim, the ranking was recomputed under six scenarios. Scenario A uses all 99 published rows. Scenario B removes seven workspace and platform rows for which no other program records a full Yes, except one (compute server management, Yes for the Schrodinger Suite). The rows and the number of other programs recording Partial are: AI planning with human approval (1), per-result provenance record (0), PDF and Word report generation (4), team workspace and user rights (1), AI models and document search (1), compute server management (2, plus one Yes) and job queueing and scheduling (6; AiMolSim itself is Partial). These rows were added to the matrix by the subject's developer. Scenario C keeps only the 80 scientific rows (electronic structure; basis sets, symmetry and SCF; structures, properties and spectra; simulation and sampling). Scenario D keeps the 80 scientific rows and counts an AiMolSim "Yes" as 0.5 whenever the developer's own note for that cell contains "Unverified", "Unchecked" or "no printed" check.

Scenario E repeats Scenario D but scores "Outside the program" as 0.5 for every program, to avoid penalising modular ecosystems in which a capability is reached through an interface. Scenario F is a lower bound for the subject: every AiMolSim scientific Yes is scored 0.5, because no matrix row has yet been mapped to methods that are all at the Checked level, and Section 4.5 shows that a matrix Yes can correspond to Partial methods. Scenario D is asymmetric by design: claims by the other 30 programs were not adjusted, because they describe features of programs whose methods are documented in peer-reviewed software papers and long user histories [1-27], whereas the AiMolSim notes themselves state that the feature has not been checked. Scenario D is therefore still a favourable estimate for AiMolSim, since a non-flagged "Yes" may still map to methods that the method index records as Partial.

3.4 Verification Depth

The method index was used to count, per domain, the methods recorded as Checked, Partial, Unverified and In build. We also extracted the cross-engine agreement figures and the list of capabilities the developer states are not offered.

3.5 Conflict of Interest Handling

AiMolSim is developed by the authors' organisation (see Declarations). To limit bias, (i) every number in this review is taken from a public, dated source that readers can check; (ii) the subject developer's own scoring rule was applied unchanged before any adjustment; (iii) all adjustments in Section 3.3 can only lower, never raise, the subject platform's score; and (iv) internal inconsistencies found in the subject's documentation are reported (Section 4.5).

4. Results

4.1 Overall Coverage

Table 2. Capability scores by category for all 31 programs (Scenario A; Yes = 1, Partial = 0.5).

Rank	Program	Electronic structure (23)	Basis, SCF (7)	Properties, spectra (25)	Simulation, sampling (25)	Workspace, trust (14)	Platform (3)	Licence (2)	Total (99)
1	AiMolSim	20.5	6	23	22.5	10	2.5	0	84.5
2	Schrodinger Suite	15.5	6	20	20	9	2	0.5	73
3	AMS (ADF)	15.5	4.5	22	17.5	9.5	1	0	70
4	ORCA	20.5	6	22	10	5	0	1	64.5
5	TURBOMOLE	20.5	5	21	9.5	7	0	0.5	63.5
6	Q-Chem	20.5	5	22	10	4.5	0	0.5	62.5
7	CP2K	17	5	16.5	14.5	7	0	2	62
8	NWChem	18.5	5	14.5	14	5.5	0	2	59.5
9	GAMESS	19	4.5	15	10.5	4.5	0	1	54.5
10	Gaussian 16	18.5	5.5	18.5	5	5.5	0	0	53
11	Molpro	20	5.5	16	3.5	7	0	0	52
12	PySCF	18	6	12	6	4.5	0	2	48.5
13	Jaguar	14.5	4.5	18.5	4	6.5	0.5	0	48.5
14	OpenMolcas	14.5	5	15.5	4.5	5	0	2	46.5
15	Psi4	18	5	12.5	0.5	4.5	0	2	42.5
16	Spartan	14.5	4	11	4.5	6	1	0	41
17	ABINIT	10.5	3	11	7.5	4.5	0	2	38.5
18	VASP	11	2.5	11	8.5	5	0	0	38
19	Quantum ESPRESSO	9	3	11	6.5	6	0	2	37.5
20	AmberTools	5.5	1.5	7.5	12.5	7.5	0.5	2	37
21	DFTB+	8	0.5	7.5	6.5	5.5	0	2	30
22	CHARMM	1	0.5	7	12.5	6.5	0	1	28.5
23	MOPAC	5.5	1.5	10	4	4.5	0.5	2	28
24	LAMMPS	1	0	5.5	9.5	8	0	2	26

Rank	Program	Electronic structure (23)	Basis, SCF (7)	Properties, spectra (25)	Simulation, sampling (25)	Workspace, trust (14)	Platform (3)	Licence (2)	Total (99)
25	xtb	2	1	8	7	4	0	2	24
26	GROMACS	0.5	0	3.5	9.5	6.5	0	2	22
27	OpenMM	0	0	3	8.5	6	0	2	19.5
28	NAMD	1	0	2.5	8	7	0	1	19.5
29	Open Babel	0	0.5	2	5	4.5	0	2	14
30	Avogadro 2	0	0.5	2.5	4.5	4	0	2	13.5
31	RDKit	0	0	2	4	3.5	0	2	11.5

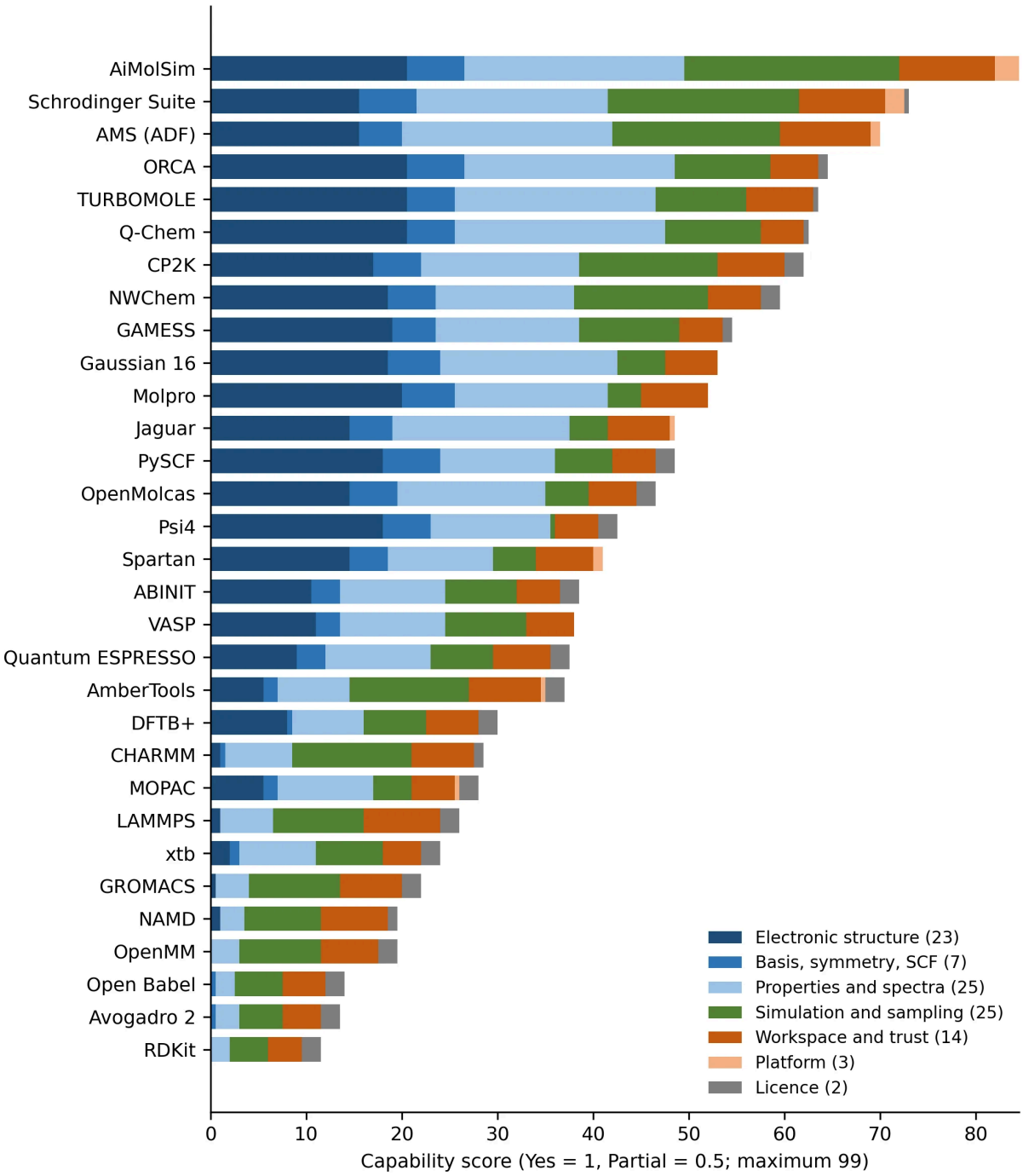


Figure 1. Capability scores of the 31 programs by category (Scenario A). Bars are sorted by total score.

Across all 2,970 non-subject cells, 1,025 (34.5%) were Yes, 408 (13.7%) Partial, 1,297 (43.7%) No and 240 (8.1%) Outside the program, which reflects the specialisation of most programs. General-purpose and integrated programs dominate the top of the ranking, and specialised tools (molecular dynamics engines and cheminformatics toolkits) occupy the lower half. Their low totals reflect scope, not quality: GROMACS, OpenMM and RDKit are leaders in their own fields but were never meant to cover quantum chemistry.

4.2 Category Analysis

Electronic structure. The four leading molecular codes are tied: ORCA, Q-Chem, TURBOMOLE and AiMolSim each scored 20.5 of 23, followed by Molpro (20.0), GAMESS (19.0), NWChem and Gaussian 16 (18.5), and PySCF and Psi4 (18.0). The category is near saturation for mature programs; differences concern restricted open-shell Kohn-Sham, reduced-scaling exchange, embedding, GW and real-time TDDFT. AiMolSim's matrix entry for restricted open-shell methods is No, while ORCA, Gaussian 16, Q-Chem, TURBOMOLE, Molpro and AMS report Yes.

Structures, properties and spectra. AiMolSim (23.0 of 25) leads ahead of AMS, ORCA and Q-Chem (22.0) and TURBOMOLE (21.0). Spectroscopic coverage (NMR, EPR, ECD/VCD/ROA, vibronic spectra, intersystem crossing rates) is broad across the leading proprietary codes.

Simulation and sampling. This is where the subject platform separates from molecular quantum chemistry codes. AiMolSim scored 22.5 of 25, against 20.0 for the Schrodinger Suite, 17.5 for AMS, 14.5 for CP2K and around 10 for ORCA and Q-Chem. The lead comes from combining, in one program, molecular mechanics, periodic MD, enhanced sampling, free energy, docking, machine-learned potentials, grid quantum dynamics and rate theory, capabilities that the ecosystem otherwise spreads across AmberTools, GROMACS, OpenMM, LAMMPS and dedicated cheminformatics tools.

Workspace, platform and licence. AiMolSim led workspace and trust (10.0 of 14) and platform (2.5 of 3), followed by AMS (9.5 and 1.0) and the Schrodinger Suite (9.0 and 2.0). It scored 0 on licence (proprietary; academic terms on request), as did Gaussian 16, VASP, Molpro, AMS and Jaguar, while 17 programs are open source.

4.3 Distinctive and Missing Capabilities

Table 3. Capabilities marked Yes for AiMolSim that two or fewer other programs provide.

Capability	Other programs with Yes	AiMolSim note (developer)
pKa and Redox Potentials	AmberTools, Schrodinger Suite	pKa, redox vs SHE
End-Point Binding Energy (MM/GBSA, MM/PBSA)	AmberTools	MM/GBSA, MM/PBSA
Quantum Dynamics on Grids	None	DVR, wavepackets, tunnelling
Rate Constants and Tunnelling	AMS (ADF), Schrodinger Suite	CVT, SCT, RRKM, microkinetics, lattice KMC (ZGB y1/y2 window checked in the 1.9.2 gate)
AI Planning with Human Approval	None	Plans, waits for approval
Per-Result Provenance Record	None	Version, data checksums, sources
Report Generation (PDF, Word)	None	PDF and Word reports
Team Workspace and User Rights	None	Roles, rights, shared studies
Compute Server Management	Schrodinger Suite	Pair, update, remote; Add Server
AI Models and Document Search	None	Model management, hybrid search

Ten capabilities marked Yes for AiMolSim are offered by two or fewer of the other programs (Table 3). Five are governance and workspace features that no other program in the matrix provides: AI planning with human approval, per-result provenance, report generation, team rights and AI document search. These answer real needs in regulated and team settings (traceability and approval of computational work), and they parallel what workflow systems

such as AiiDA [32] provide for scripted research. Grid quantum dynamics is unique in the matrix; rate constants with tunnelling, pKa and redox potentials, and end-point binding energies are shared with one or two programs.

Table 4. Capabilities where AiMolSim is Partial or No in the matrix.

Capability	AiMolSim status	Developer note
Restricted Open-Shell (ROHF, ROKS)	No	RHF, UHF, RKS, UKS only
Reduced-Scaling DFT (COSX, Fast Multipole)	Partial	COSX, CFMM one level (Unverified)
Subsystem and Quantum Embedding	Partial	Projection (energy identity checked), FDE (Unverified)
GW and BSE	Partial	Molecular G0W0, evGW, BSE
Plane Waves and PAW	Partial	Norm-conserving PW, DFT+U, DFPT, Birch-Murnaghan EOS; no PAW
Density Fitting (RI)	Partial	RI-J (built-in aux), RI-JK (HF) and RI-MP2 with user or auto aux basis
Natural Bond Orbital Analysis	Partial	NHO, NLMO unverified
COSMO-RS and 3D-RISM Thermodynamics	Partial	COSMO-RS H C N O Cl (Unverified); COSMO-SAC core, not in tool
ZFS, Exchange Coupling, Mossbauer	Partial	ZFS (spin-orbit part checked); broken-symmetry J, Mossbauer built, not yet run (Unverified); no magnetizability, MCD
X-Ray Absorption, Core Spectra	Partial	Delta-SCF, CVS, TP; no L edges
Polarisable, Coarse-Grained, Reactive Force Fields	Partial	Thole and Drude polarisation, EAM, Tersoff, SW; no coarse-grained (MARTINI), no ReaxFF, no AMOEBA multipoles
Ab Initio MD (BOMD, CPMD)	Partial	BOMD up to 30 atoms; no Car-Parrinello, no xTB MD
QM/MM	Partial	Guided forms in Engine tools
Machine-Learned Potentials	Partial	Five architectures; own data only
Pharmacophore, Shape and ADMET Filters	Partial	Rules, ESOL; no shipped ADMET models
Python Scripting API	No	GUI only by design
GPU Acceleration	Partial	GPU J/K, off until measured
Multi-Core Execution on One Machine	Partial	Threaded math kernel, thread limit
Multi-Node Parallelism (MPI)	Partial	Own rank mesh to 64 ranks, not MPI; network speed-up unmeasured
Checkpoint and Restart	Partial	MD, sampling, distributed SCF
Linux, macOS, Windows Support	Partial	macOS and Windows; Linux build compiles, 6 regression cases differ, not offered
File Format Import and Export	Partial	Engine reads 11, writes 8 (two unverified); app imports SMILES, MOL, SDF, PDB, XYZ, exports CSV, XYZ, SDF, cube, JSON; no InChI write, Mol2, FCHK
Job Queueing and Scheduling	Partial	Built-in queue, fair share, fill gaps, split runs; not yet on a customer site; no Slurm, PBS, LSF
Open-Source Licence	No	Proprietary
Free for Academic Use	No	Academic terms on request

Table 4 lists 25 capabilities that AiMolSim records as Partial or No. Several are central for professional users: no public Python scripting API (offered by 17 other programs), no Linux release (Linux build compiles, six regression cases differ), parallelism through an in-house rank mesh of up to 64 ranks rather than MPI with network speed-up not yet measured, GPU

acceleration switched off until measured, plane-wave DFT without PAW datasets, and a job scheduler bridge to Slurm, PBS and LSF still in build. The developer also lists capabilities not offered, including PM6, PM7 and GFN-FF, CHARMM36 and ff19SB force fields, W1 and CBS-QB3 composite methods, and InChI writing.

4.4 Verification Depth

Table 5. AiMolSim method verification status by domain (method index and engine page, engine 1.10.12, evening of 13 November 2026). Checked (exact) means no paper prints a value and the method is checked against exact mathematics or a brute-force recomputation; the developer counts it separately.

Domain	Checked (printed values)	Checked (exact)	Partial	Unverified	In build	Total	Printed-value share
Quantum chemistry	13	0	33	2	0	48	27%
Excited states and spectroscopy	7	0	16	3	0	26	27%
Thermochemistry and kinetics	6	0	11	1	0	18	33%
Molecular mechanics and dynamics	4	1	10	2	0	17	24%
QM/MM and free energy	0	1	8	0	0	9	0%
Solvation	0	0	12	0	0	12	0%
Relativistic and magnetic	3	0	3	4	0	10	30%
Machine-learned potentials	0	0	2	2	0	4	0%
Cheminformatics	4	0	8	1	0	13	31%
HPC and clusters	0	2	3	1	1	7	0%
Total	37	4	106	16	1	164	23%

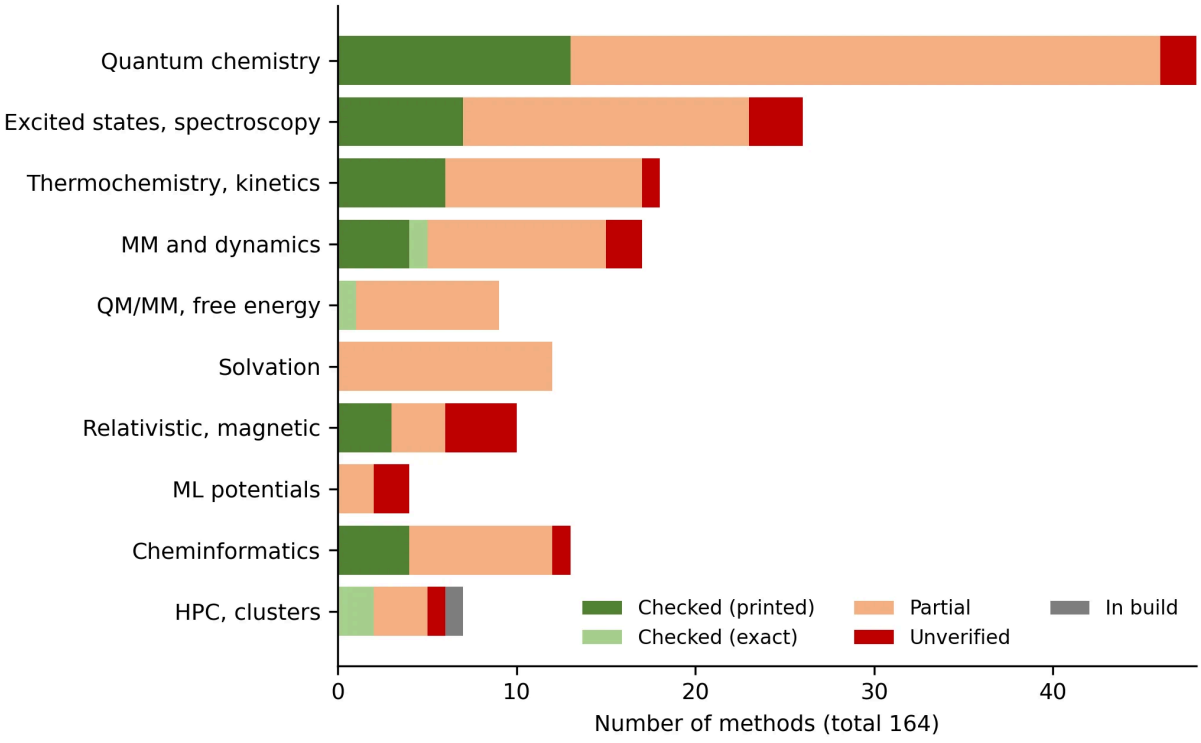


Figure 2. AiMolSim method verification status by domain.

Of 164 methods, 37 (22.6%) are recorded as Checked against printed values, 4 (2.4%) as Checked (exact), 106 (64.6%) as Partial, 16 (9.8%) as Unverified and one as In build (Table 5, Figure 2). The developer introduced the separate exact-check badge on 13 November 2026, moving one quantum chemistry method out of the printed-value category; we report both categories and do not add them together. Verification is deepest in quantum chemistry (13 of 48 checked), where printed values are reproduced, for example UHF-CCSD(T) for the CN radical against Watts, Gauss and Bartlett [41], RHF-CCSD for BeH2 against Purvis and Bartlett [40] and frozen-core MP2 energies against Weigend and Haser [39]. Two domains, solvation (12 methods) and machine-learned potentials (4), contain no checked method of either kind; QM/MM and free energy (9) and HPC and clusters (7) contain exact checks only (one and two methods), for example a distributed SCF whose energy on 2 and 3 ranks equals the one-rank energy within 1e-10 Eh.

Cross-engine checks are more extensive. The validation register reports that 354 of 419 quantum chemistry cases agree with PySCF within tolerance and the other 65 differ for causes the developer has identified (for example, different published STO-3G exponent tables); the corresponding figures are 65 plus 28 explained of 93 against Psi4, 988 plus 55 of 1,043 against RDKit, and 616 plus 14 of 630 against Open Babel, measured with engine 1.10.4 on 11 November 2026. Agreement with an independent program is strong evidence of correct implementation where the methods overlap, but the developer's own summary states that "Those programs cover far more methods today" [W1], so the cross-checks cover only part of the claimed breadth.

4.5 Consistency of the Subject Documentation

Two inconsistencies were found between the matrix and the method index. Restricted open-shell methods are "No" in the matrix but "ROHF and RO-KS [partial]" in the method index; and "pKa and Redox Potentials" is "Yes" in the matrix while both pKa and redox methods are Partial in the method index, and the solvation domain has no checked method. These do not change the conclusions, but they show that matrix entries and method-level evidence can drift apart in a fast-moving product, and that capability claims should be traceable to the method level.

4.6 Sensitivity of the Ranking

Table 6. Ranking of AiMolSim under six scoring scenarios.

Scenario	AiMolSim rank	AiMolSim score	Leading other programs (score)
A Published (99 rows)	1	84.5	Schrodinger Suite (73), AMS (ADF) (70), ORCA (64.5)
B Without seven workspace rows (92)	1	78	Schrodinger Suite (70), AMS (ADF) (68.5), ORCA (64.5)
C Science only (80)	1	72	Schrodinger Suite (61.5), AMS (ADF) (59.5), ORCA (58.5)
D Science, evidence-adjusted (80)	4	58	Schrodinger Suite (61.5), AMS (ADF) (59.5), ORCA (58.5)
E Science, evidence-adjusted, Outside = 0.5 (80)	4	58	Schrodinger Suite (63), ORCA (60), AMS (ADF) (60)
F Science, lower bound: every AiMolSim Yes = 0.5 (80)	13	39.5	Schrodinger Suite (61.5), AMS (ADF) (59.5), ORCA (58.5), Q-Chem (57.5)

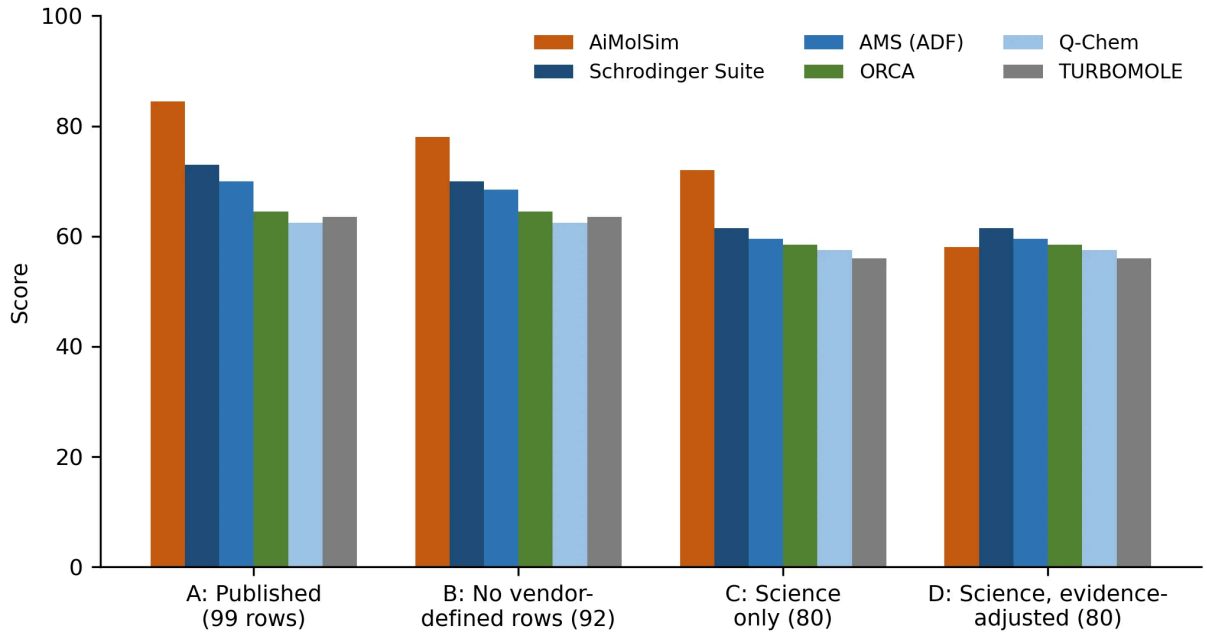


Figure 3. Scores of the six leading programs under Scenarios A to D (Scenarios E and F in Table 6). Only AiMolSim scores change in Scenario D relative to Scenario C.

Removing the seven vendor-defined rows (Scenario B) lowered AiMolSim's score from 84.5 to 78.0 and narrowed its lead over the Schrodinger Suite from 11.5 to 8.0 points; restricting to scientific capability (Scenario C) gave a lead of 10.5 points (72.0 against 61.5). The ranking changed only when evidence was taken into account. Of the 65 AiMolSim scientific "Yes" entries, 28 (43%) carry a note stating that part of the capability is unchecked or unverified. Counting those as Partial (Scenario D) placed AiMolSim fourth with 58.0, behind the Schrodinger Suite (61.5), AMS (59.5) and ORCA (58.5) and just ahead of Q-Chem (57.5) and TURBOMOLE (56.0) (Table 6, Figure 3). Scoring "Outside the program" as 0.5 (Scenario E) keeps AiMolSim fourth (58.0) behind the Schrodinger Suite (63.0), ORCA (60.0) and AMS (60.0). The lower bound (Scenario F), which scores every AiMolSim scientific Yes as 0.5, places it thirteenth with 39.5, close to Jaguar and PySCF. The subject's rank therefore lies between 1 and 13 across the six scenarios. In Scenario D the top six programs are separated by 5.5 points, so the reasonable reading is that the leading general-purpose platforms are broadly comparable in documented scientific scope, and that AiMolSim's first place in the published matrix depends on counting implemented but unverified features.

4.7 Deployment, Interface and Licensing Models

The programs follow three deployment models: open-source libraries and executables driven by input files or Python (PySCF, Psi4, CP2K, GROMACS, OpenMM, RDKit); licensed executables with optional graphical front ends (ORCA, Gaussian 16, Q-Chem, TURBOMOLE, Molpro, VASP); and integrated graphical platforms (Schrodinger Suite, AMS, Spartan, AiMolSim). AiMolSim is distinctive in being GUI-only by design, with no public API, and in running only on customer servers. The first choice lowers the barrier for non-specialists and keeps the AI assistant inside an approval workflow, but it prevents scripted high-throughput use, integration with workflow managers [31-33] and independent automated regression testing by users.

The second choice addresses data sovereignty and confidentiality, a recurring requirement in pharmaceutical and industrial research, but places hardware, operations and scaling on the customer.

4.8 Performance and Hardware Evidence

No peer-reviewed, settings-matched timing benchmark across these programs was found in our search. The nearest community effort, an unreviewed speed test, shows that default DFT integration grids alone can change relative timings between codes [W12]. Published performance evidence is therefore program-specific, and we report it as such (Table 7). Precision verification is better developed than speed: equation-of-state benchmarks first compared 40 solid-state codes and methods [30], and a reference set of 960 all-electron equations of state now verifies nine pseudopotential approaches [44].

Table 7. Published performance and hardware evidence by program (no settings-matched cross-code benchmark exists).

Program	Performance and hardware evidence	Source
AiMolSim	Runs on customer macOS and Windows machines with NVIDIA GPU support; Linux build compiles but six regression cases differ, so it is not offered. Own GEMM ran 1.13 to 1.20 times Intel MKL speed for one 256 by 4096 by 4096 product on 16 threads, and 0.63 to 0.74 of MKL for a 2048 square product. GPU Coulomb and exchange builds are off until measured faster than the CPU. Multi-GPU and fast-LAN HPC are "design only"; split frequency jobs across servers recombine within 2.4e-4 cm-1.	Vendor documentation [W1]
PySCF	GPU4PySCF adds GPU integrals, SCF, DFT, gradients, Hessians and solvation; Hartree-Fock about two orders of magnitude faster than multithreaded CPU PySCF on one A100, and comparable to GAMESS and QUICK on the same GPU.	[45, 46]
ORCA	MPI parallel; RIJCOSX and DLPNO approximations reduce the cost of hybrid DFT and coupled cluster; version 6 released 2024 to 2025.	[1, 2, 43]
Molpro	MPI parallel; GPUs not supported or used (HPC site documentation).	[8, W7]
Quantum ESPRESSO	GPU port described in the 2020 software paper, aimed at exascale systems.	[14]
VASP	OpenACC GPU port on NVIDIA since 6.2; experimental OpenMP offload for AMD and Intel in 6.6; one MPI rank per GPU; RPA, GW, linear response and BSE remain on CPU; on-the-fly ML force fields.	[16, 47, W29]
Amber (pmemd.cuda)	GPU explicit-solvent PME molecular dynamics at microsecond scale; strongest on single GPUs.	[48]
OpenMM 8	134,099-atom protein-ligand system at 105 ns/day with only the 53-atom ligand on the ANI-2x potential (hybrid ML/MM).	[49]
GROMACS 2025 and 2026	GPU-direct communication on by default (re-enabled in 2025.1 after crashes in 2025.0); NVSHMEM halo exchange; backends CUDA, HIP, SYCL and deprecated OpenCL; AMD HIP backend limited to nonbonded kernels in 2025.	[W8, W18]
NAMD 3.0	GPU-resident mode more than doubles single-GPU speed over GPU-offload; limited to one node per replica.	[26, W9]
TURBOMOLE	GPU offload of XC, seminumerical exchange, RI-K, GW and BSE in several modules; vendor RI-DFT timing tables not compared with other codes.	[9, W6]
Gaussian 16	GPU acceleration on NVIDIA GPUs under Linux for HF and DFT only; A100 support added in Rev. C.02.	[W5]
DFTB+	Single- and multi-GPU diagonalisation and density-matrix construction through MAGMA.	[58]
Quantum ESPRESSO 7.x	GPU port (OpenACC and CUDA Fortran) extended to phonons, HP and CP codes; AMD and Intel offload planned.	[59]
LAMMPS	GPU package (CUDA, OpenCL, HIP) and KOKKOS (CUDA, HIP, SYCL, OpenMP).	[21]
NAMD 3.0 (scale)	224 million atoms (STMV tiling) demonstrated on Frontera and Summit in GPU-offload mode.	[W31]

Program	Performance and hardware evidence	Source
GAMESS	Fragmentation runs on 128 to 512 Frontier nodes for 935 water molecules (exascale project work).	[7]

For AiMolSim the evidence is notably candid but thin. The developer publishes one measured kernel win and states that the kernel is slower than MKL elsewhere, that GPU acceleration stays off until it is proven faster, and that distributed computation has been measured only with local processes on one machine. By contrast, the established MD engines and several electronic structure codes have years of published GPU and multi-node scaling. Writing all dense linear algebra in-house, without MKL, OpenBLAS or cuBLAS, removes third-party dependencies but gives up decades of vendor tuning; on current evidence, AiMolSim should not be chosen for throughput on large systems.

4.9 Strengths and Weaknesses of Each Program

Table 8 summarises the principal strengths and documented weaknesses of each program. Weaknesses are drawn from the capability matrix (cells recorded as No), from official documentation and from licence texts; they describe scope and conditions of use, not code quality.

Table 8. Strengths and weaknesses of the 31 programs.

Program	Principal strengths	Documented weaknesses and limitations
AiMolSim	Broadest single-program scope in the matrix; AI planning with human approval; per-result provenance; reports; team rights; self-hosted, data stays on site; published per-method verification status	Only 22.6% of methods checked against printed values (25.0% including exact checks); no public API or command line; no Linux release; MPI, multi-GPU and scheduler bridge not released; GPU off by default; proprietary, no free academic tier; no restricted open-shell in the matrix; no PM6, PM7, CHARMM36 or ff19SB
ORCA	Free for academic use; DLPNO coupled cluster; RIJCOSX; spectroscopy and magnetic properties; version 6.1.1 [W3]	Closed source; no periodic boundary conditions; no GPU support in the 6.1 changelog; some DLPNO paths single core; defaults and keywords change between versions [W16]
Gaussian 16	Broad, standardised molecular methods; GaussView; long user history	Paid licence with restrictive terms, including clauses against use that assists competitors [W5]; no source licence for some countries; GPU only for HF and DFT on NVIDIA under Linux; B3LYP differs from VWN5 codes [52]
Q-Chem	Excited states (EOM, ADC, TDDFT); energy decomposition; wide functional range; version 7.0 [W4]	Commercial; GPU acceleration through separately licensed BrianQC; former SG-1 default grid inadequate for meta-GGAs [54, W17]
TURBOMOLE	Efficient RI-DFT, RI-MP2, CC2 and GW; molecular and periodic DFT; TmoleX GUI	Proprietary; free teaching licence not usable for published results; vendor benchmarks only
Molpro	High-accuracy correlated, multireference and F12 methods	No GPU support; paid licence; no periodic systems
AMS (ADF)	Slater-type orbitals; ZORA relativistic; BAND, DFTB and ReaxFF in one GUI; machine-learned potentials (MACE, Orb, MatterSim, FairChem) in AMS2025 and later [W10]	Commercial; not free for academics; partial GPU
Jaguar / Schrodinger Suite	Integrated drug discovery platform; Maestro GUI; Python; ML force fields since release 2025-4 [W11]	High cost; closed source; Jaguar narrower than dedicated QC codes; partial GPU
Spartan	Teaching and desktop GUI; property database	Small method scope; no API, GPU or multi-node; Spartan'26 single licence (up to 16 cores) from USD 1,800 academic [W13]
PySCF	Python library; broad molecular and periodic Gaussian methods; GPU4PySCF	No GUI; limited MPI; Python overhead

Program	Principal strengths	Documented weaknesses and limitations
Psi4	SAPT; density-fitted correlated methods; Python and QCArchive integration	No periodic systems; no multi-node MPI; no GUI
NWChem	MPI scalability; coupled cluster; plane-wave DFT; QM/MM	Complex build; development shifting to NWChemEx; no GUI
GAMESS	FMO and EFP; many MCSCF and CC variants; no cost	No-cost site licence but not open source; registration required; older input format
OpenMolcas	Multiconfigurational methods (CASSCF, CASPT2, MC-PDFT); spin-orbit	Some Molcas modules not included in the open-source release; molecules only; steep learning curve
CP2K	Condensed-phase DFT MD; linear scaling; QM/MM; GPU and multi-node	Hartree-Fock exchange too expensive with MOLOPT basis sets unless ADMM is used [W28]; large input tree; hard build
Quantum ESPRESSO	Plane-wave DFT; DFPT phonons; GPU port; large ecosystem	Plane-wave hybrid functionals expensive beyond a few dozen atoms per cell; GPU port NVIDIA only, AMD and Intel planned [59]
ABINIT	DFPT; GW and BSE; electron-phonon coupling; PAW	Periodic focus; partial GPU; no GUI
VASP	PAW datasets; hybrid functionals; GW and RPA; GPU; ML force fields	Licence per research group, centrally registered, not free; RPA, GW, linear response and BSE not ported to GPU [W29]; closed source
xtb	GFN methods; fast conformer and screening workflows	Accuracy limited by parameterisation span, for example failures for transition-metal hydride pKa [56]; minimal basis; no multi-node; no GUI
DFTB+	Molecules and solids; electron transport; MPI	Element coverage limited by Slater-Koster sets (3OB covers 15 elements); GPU through MAGMA only for diagonalisation and density matrix [58]
MOPAC	PM6 and PM7; MOZYME linear scaling; now open source	PM7 over-binds van der Waals complexes of large molecules (developer caveats) [W27]; barrier heights in enzyme models off by about 10 to 15 kcal/mol [57]; no GPU or multi-node
OpenMM	Python-first; custom forces; GPU; ML potentials through OpenMM-ML	No multi-node MPI; single precision shows systematic energy drift; double precision slow on some GPUs [W19]
GROMACS	Leading biomolecular MD performance on GPUs and clusters; free energy	Command line only; trajectories not bitwise reproducible [W18]; distributed GPU PME pending validation [W18]; steep learning curve
LAMMPS	Very broad materials and soft-matter models; KOKKOS portability; ML interatomic potentials	Biomolecular setup less convenient; build complexity with ML packages
AmberTools / Amber	Biomolecular force fields; pmemd.cuda GPU MD; MM/GBSA	pmemd licence for commercial users; multi-GPU gains mainly for replica methods
CHARMM	Reference CHARMM force fields; wide methods including QM/MM	Old scripting language; steep learning curve; not open source
NAMD	Scalable MD; GPU-resident mode; VMD integration	GPU-resident limited to one node per replica; QM only through external programs
RDKit	Standard open cheminformatics toolkit; Python and C++	No QM or MD; default embedding gives low-quality conformers for macrocyclic peptides (preprint) [W30]
Open Babel	Conversion among more than 100 file formats	Infrequent releases; no QM
Avogadro 2	Free GUI builder and input generator for QM codes	Not a compute engine; platform-specific bugs in recent releases [W14]

4.10 AI and Machine Learning Across the Field

Machine learning is now present in most major programs, but in two different roles. The first is machine-learned potentials that replace or accelerate electronic structure: VASP trains force fields on the fly [47], OpenMM 8 runs ML potentials inside MD [49], AMS2025 integrates

MACE, Orb, MatterSim and FairChem [W10], and Schrodinger releases include ML force fields [W11]. The second is language-model agents that plan and run computations: Coscientist planned and executed experiments [35], ChemCrow connected an LLM to 18 chemistry tools [36], and El Agente builds quantum chemistry workflows from natural-language requests, with high task success in its evaluation but a narrow set of calculation types [50]. AiMolSim belongs to the second group, with one difference from these research agents: its assistant can only draft a plan, and a person must approve it before anything runs. That makes it a governed product feature rather than an autonomous agent. Its machine-learned potentials are restricted to the user's own training data and have no checked method.

The AiMolSim assistant documents further safeguards: it creates a study, molecule and draft plan but queues nothing until a person presses Start, with an extra cost confirmation for large estimates; it quotes only stored results with numbered citations; suggested literature must match author, year and title in Crossref or is dropped; AI-proposed structures are validated by the engine and compared with PubChem, including stereo and isotope agreement; outside content is treated as data, never as instructions; suggestion chips are limited to an allowlist; and behaviour and safety cases run in an offline test gate before each release [W1]. Reports are blocked when a number cannot be traced to a stored result. On security, the developer states two open limits: several organisations sharing one GPU are not isolated, so this is not offered, and a two-factor policy re-check on organisation switch is pending [W32]. The developer also lists, as not planned by design, a user scripting API or command line, hosted compute, models trained on other groups' data and parameters without a published source [W1].

4.11 Current Versions, Licences and Prices

Table 9 lists the current release, licence and published price of every program, checked on official release pages, repositories and price pages on 13 November 2026 [W15]. Seventeen programs are free and open source. Of the proprietary programs, ORCA is free for academic and personal use, GAMESS is distributed at no cost under a site licence that is not open source, and TURBOMOLE offers a free teaching licence that may not be used for published results. Gaussian, VASP, Q-Chem, AMS, Jaguar and the Schrodinger Suite are sold by quote or regional price list. AiMolSim publishes no prices and offers academic terms on request.

Table 9. Current versions, licences and prices (official pages, 13 November 2026).

Program	Current version (date)	Licence	Price	Platforms
AiMolSim	Engine 1.10.12 (2026-11)	Proprietary	Not published; academic terms on request	macOS, Windows (Linux not offered)
PySCF	2.15.0 (2026-11-05)	Apache-2.0	Free	Cross-platform Python
Psi4	1.11 (2026-06-29)	LGPL-3.0	Free	Cross-platform
NWChem	7.3.1 (2025-11-06)	ECL-2.0	Free	Cross-platform
GAMESS	2024 R2 (latest confirmed)	No-cost site licence, not open source	Free after registration	Cross-platform
OpenMolcas	26.10 (2026-10-15)	LGPL-2.1	Free	Cross-platform
CP2K	2026.2 (2026-07-15)	GPL-2.0	Free	Linux mainly
Quantum ESPRESSO	7.6 (GitHub mirror, 2026-07-09)	GPL-2.0	Free, registration to download	Cross-platform

Program	Current version (date)	Licence	Price	Platforms
ABINIT	10.8.3 (2026-08-07)	GPL-3.0	Free	Cross-platform
xtb	6.8.0 (2026-11-05)	LGPL-3.0	Free	Cross-platform
DFTB+	25.1 (2025-12-18)	LGPL	Free	Cross-platform
MOPAC	23.2.5 (2026-05-03)	Apache-2.0	Free	Cross-platform
OpenMM	8.6.1 (2026-09-11)	MIT and LGPL	Free	Cross-platform
GROMACS	2026.4 (2026-10-02)	LGPL-2.1	Free	Cross-platform
LAMMPS	Stable 30 Sep 2026, update 1	GPL-2.0	Free	Cross-platform
AmberTools / Amber	AmberTools26 / Amber26 (2026)	AmberTools mostly GPL; Amber separate licence	Amber free for non-commercial academic use; industrial licence paid	Linux, macOS
RDKit	2026.09.2 (2026-11-05)	BSD-3-Clause	Free	Cross-platform
Open Babel	3.2.1 (2026-07-11)	GPL-2.0	Free	Windows, macOS, Linux
Avogadro 2	2.1.0 (2026-10-23)	BSD-3-Clause	Free	Windows, macOS, Linux
ORCA	6.1.1 (2025-12)	Proprietary EULA	Free academic and personal; commercial by quote (FACCTs)	Windows, macOS, Linux
Gaussian 16	Rev. C.02	Proprietary	Regional price lists; no source licence for some countries	Linux, Windows, macOS
Q-Chem	7.0 (2026)	Proprietary seat, cluster or unlimited	Quote; free trial	Linux, macOS, Windows
VASP	6.6.x (2026)	Proprietary, centrally registered	Quote (academic VASP GmbH, commercial Materials Design)	Linux mainly
TURBOMOLE	8.0.1	Proprietary perpetual	Store prices; free teaching licence not for published results	Linux, Windows, macOS
Molpro	2026.1 (2026-06)	Proprietary	Group licence (published academic price EUR 1,700 first year, EUR 1,500 per year after; may be outdated)	All modern platforms
AMS (SCM)	AMS2026	Proprietary	Quote; 30-day trial	Windows, Linux, macOS Apple silicon
Jaguar / Schrodinger Suite	Release 2026-4	Proprietary	Quote	Linux, Windows, macOS ARM64
Spartan	Spartan'26	Proprietary	Single licence up to 16 cores: USD 1,800 academic, 3,600 government, 5,400 commercial	Windows, macOS, Linux
CHARMM	Yearly releases	Free academic CHARMM; commercial CHARMm via BIOVIA	Free academic; commercial by quote	Linux, macOS
NAMD	3.0.3 (2026-07)	NAMD licence, source available	Free non-commercial	Many platforms

4.12 Numerical Reproducibility Across Programs

Several documented pitfalls mean that two programs running the "same" method do not necessarily compute the same number, and they bear directly on any comparison. First, density functional definitions differ: B3LYP uses the VWN5 local correlation in ORCA and TURBOMOLE but VWN3 in Gaussian, which changes total energies [52]; ORCA offers a separate keyword for the Gaussian form [W16]. Second, integration grids matter: default grids

caused errors of several kcal/mol for meta-GGA reaction energies with the M06 suite [53], Q-Chem now recommends larger grids than its former SG-1 default for meta-GGAs [W17], and pruned grids are not rotationally invariant, so that rotating a transition state changed a predicted selectivity from 62:38 to 99:1 in one study . Third, approximations have their own errors: ORCA documents separate RI and COSX grid errors for RIJCOSX [W16]. Fourth, in molecular dynamics, GROMACS, AMBER, LAMMPS, DESMOND and CHARMM agreed to better than 0.1% in energies only after settings were reconciled, with different Coulomb constants among the largest discrepancies [54]; GROMACS states that trajectories are in principle not bitwise reproducible across runs [W18], and OpenMM documents a systematic energy drift in single precision (3.98 kJ/mol/ns against 0.217 for mixed and 0.001 for double precision) [W19]. For periodic DFT, a 960-structure all-electron reference set now allows precision to be verified across implementations [44].

These pitfalls matter for AiMolSim in two ways. Its cross-engine agreement figures are meaningful because they state the tolerance and trace every explained difference to a cause, such as differing published STO-3G exponent tables. Its developer also reports that an atomic size adjustment measured worse on its Coarse and Normal grids and is not applied there, which shows the same grid sensitivity documented for other codes.

4.13 Cloud, Web and AI Platforms

AiMolSim also competes with web and cloud platforms and with research AI agents (Table 10). Rowan is a browser platform with hosted, credit-metered compute (1 credit per CPU minute, 3 per GPU minute and 7 per H100 or H200 minute, USD 0.04 per extra credit, a free weekly allowance), a Python API on every tier and a single-tenant enterprise option [W20]. WebMO is a self-hosted web front end to existing engines such as Gaussian, ORCA, NWChem, Q-Chem, VASP and xTB, sold as one-time licences (Pro USD 1,200 academic or 3,000 commercial; Enterprise USD 2,400 or 6,000; a free Basic edition; cloud from USD 25 per month) [W21]. Schrodinger LiveDesign is a collaborative design platform whose managed compute runs in the customer's cloud account or on Schrodinger-hosted systems [W22]. Materials Square offers cloud materials and chemistry simulation [W23]. SAMSON is a desktop design platform with an AI help assistant, Python SDK and credit-based remote jobs [W24]. AiiDA [32] and Atomate2 [55] provide self-hosted, scriptable workflows with provenance as a core design goal. Among research agents, El Agente builds and runs quantum chemistry workflows from natural language and exports action traces [50], ChemCrow connects an LLM to chemistry tools and a cloud robotic platform [36], and Coscientist planned and executed coupling reactions [35]. ChemCrow and Coscientist discuss human oversight and safety measures [35, 36], but in our search we found no documentation that any of these agents requires approval of each computation before it runs. Among the platforms in Table 10, AiMolSim is the only one for which we found documentation of all four of AI planning, mandatory human approval, per-result provenance and self-hosted compute; because several competitor cells are "not documented", this is a statement about documentation, not proof of absence. AiMolSim is also the only platform in the table without any scripting API.

Table 10. Web, cloud and AI platforms compared with AiMolSim (official pages and papers, 13 November 2026).

Platform	Compute model	AI planning	Human approval	Provenance	Scripting API	Public price
AiMolSim	Self-hosted only	Yes	Mandatory	Per result, with report gate	None by design	No

Platform	Compute model	AI planning	Human approval	Provenance	Scripting API	Public price
Rowan	Cloud; single-tenant option	Agentic offering	Not documented	Not documented	Python	Credit rates
WebMO	Self-hosted or cloud	No	Not applicable	Not documented	Not documented	Yes
Schrodinger LiveDesign	Customer cloud or vendor-hosted	Not documented	Not documented	Access controls	Not documented	No
Materials Square	Cloud	No	Not applicable	Not documented	Not documented	No
SAMSON	Desktop plus cloud credits	Help assistant	Not documented	Not documented	Python SDK	No
AiiDA, Atomate2	Self-hosted, open source	No	Not applicable	Yes (AiiDA core design)	Python	Free
El Agente, ChemCrow, Coscientist	Research code	Yes	Not documented	Action traces (El Agente)	Research code	Not applicable

Machine-learned potentials are also being packaged as deployable services. NVIDIA ALCHEMI provides a containerised GPU engine for batched geometry relaxation with MACE, AIMNet2 and TensorNet models [W25], and Meta released the OMol25 data set of more than 100 million quantum chemistry calculations together with the UMA models under research licences [W26]. Such models compete with the fast tiers of electronic structure programs, and AiMolSim deliberately does not ship models trained on other groups' data.

4.14 The AiMolSim Engine and Math Kernel Compared With Other Engines

Capability matrices compare what programs offer; they say nothing about how the numbers are produced. Because the subject platform makes an unusual architectural choice, writing its own numerical foundation instead of linking established libraries, we compare the internal numerical stack of the VYROX Chemistry Engine (version 1.10.12) and its VYROX Math Kernel with those of other engines, using the developer's engine and math-kernel documentation [W1, W33] and the build documentation and software papers of the other programs [W34].

Architecture. The AiMolSim engine is organised in three layers that call only downwards: an engine layer of 185 tools in 19 modules with input checks, explicit refusals and a provenance record on every result; a core layer (data, geometry, graph, chemistry, force fields, quantum chemistry, exchange-correlation, dispersion, machine-learned potentials, docking); and a kernel layer (base, linear algebra, FFT, grids, optimisation, integrals, GPU). It is written in C++ and CUDA and, according to its developer, links no third-party numerical library: no MKL, OpenBLAS, BLIS, Eigen, Accelerate or cuBLAS, no FFTW, and integral, exchange-correlation and dispersion code written from the published papers, which are cited in the source [W33]. The engine rejects unknown input keys instead of ignoring them, and records the math-kernel mode, instruction set and thread count with each result.

How other engines are built. Most established programs assemble their numerical stack from shared, separately published libraries (Table 11). Psi4 requires BLAS and LAPACK (searching for MKL first), Libint2 for integrals, gau2grid for grid collocation, Libxc for functionals and s-dftd3 for dispersion. PySCF evaluates integrals with libcint [64] and relies on NumPy and SciPy for linear algebra. CP2K requires the DBCSR sparse-matrix library [79] and ScaLAPACK, with optional ELPA [62], COSMA [63], Libint2 and Libxc [34]. Quantum ESPRESSO

and VASP require BLAS, LAPACK, ScaLAPACK and FFTW [71] or vendor equivalents. NWChem can build OpenBLAS, ScaLAPACK and ELPA and link Libxc. ORCA, a closed-source code, generates and processes integrals with its own SHARK library [65], into which Libint was integrated from version 5, and uses Libxc for functionals. GPU-native codes write their own integral kernels: TeraChem builds on the GPU algorithms of Ufimtsev and Martinez [66-68], QUICK on GPU electron repulsion integrals and DFT [69, 70], and GPU4PySCF accelerates PySCF on NVIDIA GPUs [45, 46]. Among the engines whose numerical stacks are publicly documented (Table 11), only the GPU-native integral engines and ORCA's SHARK own their integral code as the AiMolSim engine does, and none also replaces BLAS, LAPACK and FFT libraries with in-house code; TeraChem is closed source and does not disclose its linked libraries.

Table 11. Internal numerical stacks of the VYROX Chemistry Engine and selected engines (developer documentation, build guides and software papers; 13 November 2026). Gaussian 16, Q-Chem, TURBOMOLE, Molpro, GAMESS and TeraChem do not document their linked libraries publicly and are omitted.

Engine	Language	Dense linear algebra	Distributed	Integrals	Exchange-correlation	FFT	GPU
VYROX Chemistry Engine (AiMolSim)	C++, CUDA	Own (VYROX Math Kernel)	Own rank mesh, up to 64 ranks; no MPI, no ScaLAPACK	Own (Obara-Saika route as cross-check; Haser-Ahlrichs screening)	Own	Own (radix-2, mixed radix, Bluestein; CUDA Stockham)	Own CUDA J and K, off by default
Psi4	C++, Python	External (MKL preferred)	None	Libint2	Libxc	Not applicable	Limited
PySCF / GPU4PySCF	Python, C	External via NumPy/SciPy	Limited (mpi4pyscf)	libcint	Libxc or XCFun	Not applicable	GPU4PySCF CUDA kernels
ORCA	Closed source	Not disclosed	MPI	Own SHARK with Libint integrated	Libxc	Not applicable	No GPU support documented (6.1)
CP2K	Fortran, C	External	DBCSR, ScaLAPACK, optional ELPA and COSMA	Libint2 (optional) plus own GPW	Libxc (optional)	FFTW	CUDA and HIP backends
NWChem	Fortran, C	External (can build OpenBLAS)	Global Arrays, ScaLAPACK, ELPA	Own	Own or Libxc	Plane-wave module	CCSD(T) kernels
Quantum ESPRESSO	Fortran	External (vendor recommended)	ScaLAPACK, ELPA	Plane waves	Own or Libxc	FFTW	OpenACC and CUDA Fortran
VASP	Fortran	External (mandatory)	ScaLAPACK (mandatory)	Plane waves	Own	FFTW or vendor (mandatory)	OpenACC; OpenMP offload
DFTB+	Fortran, C	External	ScaLAPACK, ELSI	Not applicable	Not applicable	Not applicable	MAGMA
QUICK	Fortran, C++, CUDA	External	MPI	Own GPU ERI engine	Own GPU quadrature	Not applicable	CUDA multi-GPU
GROMACS	C++	Bundled or external BLAS and LAPACK	MPI domain decomposition	Not applicable	Not applicable	FFTW, MKL or bundled	CUDA, HIP, SYCL
LAMMPS	C++	Not required for core	MPI	Not applicable	Not applicable	FFTW3, MKL, KISS; cuFFT, hipFFT with KOKKOS	GPU and KOKKOS packages

Dense linear algebra. The AiMolSim GEMM uses packed, cache-blocked matrix multiplication with register-tiled micro-kernels selected at run time for Apple SME, ARM NEON or a scalar path [W33]. This is the same design that underlies GotoBLAS, OpenBLAS and BLIS, in which operands are packed into contiguous panels sized for each cache level and a small assembly or intrinsic micro-kernel does the arithmetic [60, 61]. What distinguishes production libraries is the per-architecture tuning of block sizes and micro-kernels. The single published measurement shows both sides: a 256 by 4096 by 4096 double-precision product on 16 threads ran 1.13 to 1.20 times as fast as Intel MKL on the same machine, while a 2048 by 2048 product ran at 0.63 to 0.74 of MKL speed, and the developer claims no parity [W33]. The GEMM micro-kernels are documented for Apple SME, ARM NEON and a scalar fallback; no x86 (AVX2 or AVX-512) GEMM micro-kernel is listed, although the FFT has an AVX2 stage. The hardware used for the published MKL comparison is not stated; because MKL targets x86 processors, the documentation should state which micro-kernel produced that result and which path x86 servers use. One preprint reports that just-in-time SME kernels on Apple M4 processors exceed vendor BLAS in many tested configurations (see the note on unreviewed reports).

Eigensolver. The AiMolSim symmetric eigensolver performs blocked Householder tridiagonalisation [72, 80] followed by the implicit QL algorithm [73], with an optional deterministic order that gives identical bits on reruns. The developer's HPC page also cites the divide-and-conquer methods of Cuppen [74] and Gu and Eisenstat [75], but the math-kernel page describes only QL. Production libraries offer faster alternatives for the eigenvectors that SCF needs: LAPACK provides divide-and-conquer (dsyevd) and relatively robust representation (dsyevr) solvers, and ELPA's two-stage reduction is designed for massively parallel electronic structure work [62]. QL with eigenvector accumulation scales as O(n^3) with a larger constant than divide-and-conquer or MRRR; the cost is small next to integral evaluation for the few hundred to few thousand basis functions of typical molecular work, but it becomes a disadvantage for large systems, and the developer publishes no eigensolver timings.

Integrals and GPU. The developer documents direct, screened Coulomb and exchange builds on the GPU up to d functions and GPU RI-J three-centre integrals, following Ufimtsev and Martinez [66, 67] and Haser and Ahlrichs screening [76], with Obara-Saika recursion [77] as a cross-check route. These kernels are badged Unverified and stay switched off until measured faster than the CPU [W33]. By comparison, GPU integral engines in TeraChem, QUICK and GPU4PySCF support higher angular momentum (QUICK reached f functions and beyond for ERIs and derivatives [70]) and have published speed-ups, for example Hartree-Fock speed-ups of about two orders of magnitude on one A100 GPU relative to multithreaded CPU PySCF, as reported by its developers for their test systems [45]. The CPU integral code, not the GPU, therefore carries AiMolSim's production work today.

Fourier transforms and distributed work. The engine implements its own FFT (radix-2, mixed radix and Bluestein, with an AVX2 stage on the CPU and a Stockham CUDA kernel) instead of FFTW [71], whose planner adapts algorithms to the hardware; the developer states that its CUDA FFT is not claimed faster than the CPU [W1]. Distributed SCF runs over an in-house rank mesh of up to 64 ranks, with energies on 2 and 3 ranks equal to the single-rank energy within 1e-10 Eh, but without MPI, ScaLAPACK, ELPA or measured network speed-up [W1]. Established codes reach thousands of nodes through MPI and distributed libraries such as ScaLAPACK, ELPA, DBCSR, COSMA and MAGMA [62, 63, 78, 79], as the GAMESS and NAMD scale results in Table 7 show.

Reproducibility. The engine offers an optional mode with a fixed summation order in which one and four threads, and repeated runs with the same number of ranks, give bitwise identical results, and every result records the kernel mode and instruction set [W33]. Most engines inherit floating-point summation order from their BLAS and threading libraries; Intel MKL offers an optional conditional numerical reproducibility mode for this reason [W35], and GROMACS states that its trajectories are not bitwise reproducible by design [W18]. Bitwise reproducibility on fixed hardware does not imply agreement across processors or with other programs, which is why the cross-engine checks in Section 4.4 remain necessary.

Table 12. Trade-offs of writing the numerical foundation in-house versus linking established libraries.

Dimension	VYROX engine and math kernel (own code)	Engines built on established libraries
Auditability and licensing	Every routine is in-house and cited to its source paper; no third-party binary or licence in the numerical path	Libraries such as BLAS, LAPACK, Libint, libcint, Libxc and FFTW are open, peer-reviewed and audited by many users; proprietary vendor libraries are closed
Correctness assurance	One team tests every kernel (micro-kernel versus triple loop, threads versus single thread, residuals); exposure is limited to one user base	Libraries are exercised by thousands of users and codes, and errors are found and fixed across the community
Performance	One measured GEMM shape faster than MKL; a 2048 square product 0.63 to 0.74 of MKL and other sizes slower; GPU J and K off; x86 GEMM micro-kernel not documented	Vendor BLAS and cuBLAS run close to hardware peak and are tuned per processor generation; mature GPU integral engines publish large speed-ups
Scalability	Own rank mesh to 64 ranks, no MPI; split frequency jobs across servers	MPI with ScaLAPACK, ELPA, DBCSR and COSMA to thousands of nodes
Reproducibility	Optional fixed-order mode with bitwise-identical reruns on the same hardware; kernel mode recorded per result	Depends on library settings; optional reproducibility modes (for example MKL CNR); MD engines generally non-deterministic on GPUs
Portability	Same source on macOS and Windows, CPU and CUDA; Linux build has six differing regression cases	Linux is the primary platform for most HPC codes; broad CPU and GPU vendor coverage
Maintenance burden	The developer must re-implement and re-tune for each new processor feature	Improvements in shared libraries reach every dependent code

Section 4.14 relies entirely on vendor self-reports; we made no measurements. Taken together, the AiMolSim engine and math kernel are a documented in-house design. They buy auditability, licence independence, deterministic results and a single code base for macOS and Windows, at the cost of performance, scalability and the community testing that shared libraries receive (Table 12). On current published evidence the kernel does not match vendor libraries outside one matrix size, and its GPU path is not yet in production use. Independent benchmarking of the engine against PySCF, Psi4 and ORCA on common molecules, hardware and settings, which the developer has not yet published, is the most important missing evidence.

4.15 Gap Analysis

Table 13. Gap analysis.

Gap	Findings
Gaps AiMolSim fills in the market	Approval before execution, provenance per result and reports printed from stored rows in one product; molecule-scale QM, MM, sampling and cheminformatics in one self-hosted workspace; plain publication of unchecked status

Gap	Findings
Gaps in AiMolSim relative to the market	Verification depth (75% of methods not checked against printed values or exact results); no scripting API, so no high-throughput or workflow-manager use; no Linux, MPI, multi-GPU or scheduler release; unproven performance; no free academic tier; missing standard methods (ROKS in the matrix, PM6, PM7, PAW, CHARMM36, ff19SB, W1, CBS-QB3)
Gaps in established programs	Little built-in governance (approval, audit, team rights); provenance usually needs external workflow systems [31-33]; steep learning curves and command-line interfaces; restrictive licences for several commercial codes
Gaps in the field	No settings-matched, peer-reviewed cross-code timing benchmark; capability matrices without evidence tiers; no shared standard for reporting what AI assistants may run without human approval

5. Discussion

5.1 Feature Counts Measure Breadth, Not Accuracy

The central finding is that a uniform, fully sourced matrix still ranks programs mainly by how much they implement and by which criteria are included. For mature programs, documented features rest on peer-reviewed implementations and years of external use [1-27]; for a new engine, the same "Yes" may describe code that has not yet reproduced a published number. The 22.6% printed-value share of AiMolSim methods is not a criticism of transparency: the developer publishes its own unchecked and unverified status, which made this analysis possible. It is a reminder that a capability matrix without evidence tiers cannot separate the two situations.

5.2 A Three-Tier Reporting Proposal

We propose that capability comparisons report each cell in three evidence tiers: (T1) implemented and documented; (T2) internally consistent, meaning the program satisfies exact identities or agrees with an independent program within a stated tolerance; and (T3) reproduces published reference values from the method's source or a community benchmark, with the source and tolerance cited. AiMolSim's Checked status corresponds to T3 and its cross-engine agreement to T2. For established programs, T2 and T3 evidence exists in benchmark literature and community projects [30, 31] but is rarely linked to feature lists. Reporting tiers would let users weigh breadth against verification and would reward the published-value checking that the reproducibility literature calls for [29, 30].

5.3 Where AI-Native Platforms Add Value

The governance features that only AiMolSim offers in the matrix (approval before execution, provenance per result, reports printed from stored rows, team rights) address risks that grow as LLM agents are connected to scientific tools [35, 36, 42]: unreviewed execution, untraceable defaults and unreproducible reports. A human-approval step is a practical safeguard that matches the precautions discussed for autonomous chemical agents [35, 36]. These features are, however, orthogonal to scientific capability, and established programs reach similar traceability through external workflow systems [31-33]. In selection decisions they should be weighed as platform features, separately from method coverage.

5.4 Selection Guidance

Table 14. Use-case selection guide derived from the capability evidence.

Primary need	Programs indicated by the documented capability evidence
High-accuracy molecular wavefunction methods (CCSD(T), local correlation, F12, multireference)	ORCA, Molpro, Q-Chem, TURBOMOLE, OpenMolcas
Periodic solids, surfaces and materials	VASP, Quantum ESPRESSO, ABINIT, CP2K
Biomolecular and large-scale molecular dynamics	GROMACS, OpenMM, AmberTools, NAMD, CHARMM, LAMMPS
Integrated commercial drug discovery workflows	Schrodinger Suite
Fast screening of large molecular sets	xtb, DFTB+, MOPAC
Method development, scripting and teaching through code	PySCF, Psi4
Cheminformatics, file conversion and visualisation	RDKit, Open Babel, Avogadro 2
GUI-only, self-hosted use with AI planning, approval and provenance, where data must stay on site	AiMolSim, after confirming that the specific methods required are at the Checked level

Table 14 summarises which programs the documented evidence indicates for common needs. For any program, and especially for a fast-developing one, users should confirm that the specific methods they need are verified at the level their work requires, ideally by reproducing a published value for a system close to their own.

6. Limitations

First, this is a documentation-based review: no program was executed, the engine and math-kernel comparison in Section 4.14 rests on the developer's own published measurements, and no timing, scaling or accuracy benchmark was performed. Second, the matrix was compiled by the subject developer; although each cell links to official documentation and we applied adjustments that could only lower the subject's score, the choice of the 99 rows remains the developer's, and rows on which established programs excel (for example, breadth of basis sets, high-order coupled cluster or scalability on thousands of cores) may be under-represented. Third, documentation is uneven: established programs may under-document features that exist, while a new program may document features before they are mature. Fourth, only the subject platform's claims were evidence-adjusted, because only it publishes method-level verification status. Fifth, the matrix was coded by the subject's developer and has not yet been re-coded by independent raters; a blind re-scoring of a stratified random sample of cells by at least two coders unaffiliated with any of the programs, with agreement reported as Cohen's kappa, is required before the ranking can be treated as independent, and the criteria should be extended with rows on which established programs excel (for example high-order coupled cluster, basis set library breadth, periodic hybrid DFT at scale and scripting for high-throughput use). Sixth, both the programs and the matrix change quickly; the results describe the state on 26 October to 13 November 2026. An independent re-scoring of a sample of cells by investigators unconnected with any of the programs, and execution-based benchmarks on a common test set, are the natural next steps.

7. Conclusions

On a 99-capability, documentation-based comparison of 31 programs, the AI-native, self-hosted AiMolSim platform ranks first by implemented breadth, ties with ORCA, Q-Chem and TURBOMOLE in core electronic structure, and leads, on the published matrix, in simulation, sampling and governance features. Only 22.6% of its methods are recorded as reproducing published values (25.0% including exact checks), two domains have no checked method of any kind, and when unverified claims are scored as partial it falls to fourth, within a narrow band of leading general-purpose platforms. Capability matrices should therefore report evidence tiers alongside feature status. AI-assisted platforms provide approval, provenance and usability features not otherwise combined in one program, but their scientific case rests on verification depth, which users should check method by method. Its in-house engine and math kernel trade vendor-tuned performance, MPI scalability and community testing for auditability, licence independence and deterministic results; its performance on large systems is not yet demonstrated, and the field as a whole still lacks fair, settings-matched timing benchmarks.

Declarations

Article type: review, published 13 November 2026 as an institutional publication intended for journal submission; no peer review is claimed. **Author:** Ts. Dr. LEONG YEE ROCK, VYROX AI, Kuala Lumpur, Malaysia. **Competing interests:** AiMolSim, the case-study platform, is a commercial product of VYROX INTERNATIONAL SDN BHD, the author's organisation, which also compiled the capability matrix and validation register analysed here. The steps taken to limit bias are described in Section 3.5 and the remaining limits in Section 6; an independent blind re-coding of the matrix has not yet been done. **Funding and ethics:** none asserted; no human participants. **Citation policy:** the numbered reference list is restricted to journals indexed in the Web of Science Core Collection; one preprint and one non-indexed software paper are listed separately and flagged, and software documentation is cited as dated online sources.

AI assistance: an AI assistant supported literature and documentation search, reference verification against Crossref, data parsing, scoring, figure preparation, drafting, an independent fact-check and web and PDF preparation; it does not replace author or independent review. **Data availability:** the parsed 99 by 31 capability matrix, derived scores, scenario scripts and archived source pages (retrieved 13 November 2026) are available from the author on request and will be deposited in a public archive with a persistent identifier on journal acceptance.

References

Neese F. Software update: The ORCA program system, version 5.0. WIREs Comput Mol Sci. 2022;12:e1606. doi:10.1002/wcms.1606

Neese F, Wennmohs F, Becker U, Riplinger C. The ORCA quantum chemistry program package. J Chem Phys. 2020;152:224108. doi:10.1063/5.0004608

Sun Q, Zhang X, Banerjee S, et al. Recent developments in the PySCF program package. J Chem Phys. 2020;153:024109. doi:10.1063/5.0006074

Smith DGA, Burns LA, Simmonett AC, et al. Psi4 1.4: Open-source software for high-throughput quantum chemistry. J Chem Phys. 2020;152:184108. doi:10.1063/5.0006002

Epifanovsky E, Gilbert ATB, Feng X, et al. Software for the frontiers of quantum chemistry: An overview of developments in the Q-Chem 5 package. J Chem Phys. 2021;155:084801. doi:10.1063/5.0055522

Apra E, Bylaska EJ, de Jong WA, et al. NWChem: Past, present, and future. J Chem Phys. 2020;152:184102. doi:10.1063/5.0004997

Barca GMJ, Bertoni C, Carrington L, et al. Recent developments in the general atomic and molecular electronic structure system. J Chem Phys. 2020;152:154102. doi:10.1063/5.0005188

Werner HJ, Knowles PJ, Manby FR, et al. The Molpro quantum chemistry package. J Chem Phys. 2020;152:144107. doi:10.1063/5.0005081

Balasubramani SG, Chen GP, Coriani S, et al. TURBOMOLE: Modular program suite for ab initio quantum-chemical and condensed-matter simulations. J Chem Phys. 2020;152:184107. doi:10.1063/5.0004635

Kuhne TD, Iannuzzi M, Del Ben M, et al. CP2K: An electronic structure and molecular dynamics software package. Quickstep: Efficient and accurate electronic structure calculations. J Chem Phys. 2020;152:194103. doi:10.1063/5.0007045

Fdez. Galvan I, Vacher M, Alavi A, et al. OpenMolcas: From source code to insight. J Chem Theory Comput. 2019;15:5925-5964. doi:10.1021/acs.jctc.9b00532

te Velde G, Bickelhaupt FM, Baerends EJ, et al. Chemistry with ADF. J Comput Chem. 2001;22:931-967. doi:10.1002/jcc.1056

Bochevarov AD, Harder E, Hughes TF, et al. Jaguar: A high-performance quantum chemistry software program with strengths in life and materials sciences. Int J Quantum Chem. 2013;113:2110-2142. doi:10.1002/qua.24481

Giannozzi P, Baseggio O, Bonfa P, et al. Quantum ESPRESSO toward the exascale. J Chem Phys. 2020;152:154105. doi:10.1063/5.0005082

Gonze X, Amadon B, Antonius G, et al. The Abinit project: Impact, environment and recent developments. Comput Phys Commun. 2020;248:107042. doi:10.1016/j.cpc.2019.107042

Kresse G, Furthmuller J. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. Phys Rev B. 1996;54:11169-11186. doi:10.1103/PhysRevB.54.11169

Bannwarth C, Caldeweyher E, Ehlert S, et al. Extended tight-binding quantum chemistry methods. WIREs Comput Mol Sci. 2021;11:e1493. doi:10.1002/wcms.1493

Hourahine B, Aradi B, Blum V, et al. DFTB+, a software package for efficient approximate density functional theory based atomistic simulations. J Chem Phys. 2020;152:124101. doi:10.1063/1.5143190

Eastman P, Swails J, Chodera JD, et al. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. PLoS Comput Biol. 2017;13:e1005659. doi:10.1371/journal.pcbi.1005659

Abraham MJ, Murtola T, Schulz R, et al. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX. 2015;1-2:19-25. doi:10.1016/j.softx.2015.06.001

Thompson AP, Aktulga HM, Berger R, et al. LAMMPS: A flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. Comput Phys Commun. 2022;271:108171. doi:10.1016/j.cpc.2021.108171

Case DA, Aktulga HM, Belfon K, et al. AmberTools. J Chem Inf Model. 2023;63:6183-6191. doi:10.1021/acs.jcim.3c01153

O'Boyle NM, Banck M, James CA, et al. Open Babel: An open chemical toolbox. J Cheminform. 2011;3:33. doi:10.1186/1758-2946-3-33

Hanwell MD, Curtis DE, Lonie DC, et al. Avogadro: An advanced semantic chemical editor, visualization, and analysis platform. J Cheminform. 2012;4:17. doi:10.1186/1758-2946-4-17

Brooks BR, Brooks CL 3rd, Mackerell AD Jr, et al. CHARMM: The biomolecular simulation program. J Comput Chem. 2009;30:1545-1614. doi:10.1002/jcc.21287

Phillips JC, Hardy DJ, Maia JDC, et al. Scalable molecular dynamics on CPU and GPU architectures with NAMD. J Chem Phys. 2020;153:044130. doi:10.1063/5.0014475

Stewart JJP. Optimization of parameters for semiempirical methods V: Modification of NDDO approximations and application to 70 elements. J Mol Model. 2007;13:1173-1213. doi:10.1007/s00894-007-0233-4

Krylov A, Windus TL, Barnes T, et al. Perspective: Computational chemistry software and its advancement as illustrated through three grand challenge cases for molecular science. J Chem Phys. 2018;149:180901. doi:10.1063/1.5052551

Wilkinson MD, Dumontier M, Aalbersberg IJ, et al. The FAIR Guiding Principles for scientific data management and stewardship. Sci Data. 2016;3:160018. doi:10.1038/sdata.2016.18

Lejaeghere K, Bihlmayer G, Bjorkman T, et al. Reproducibility in density functional theory calculations of solids. Science. 2016;351:aad3000. doi:10.1126/science.aad3000

Smith DGA, Altarawy D, Burns LA, et al. The MolSSI QCArchive project: An open-source platform to compute, organize, and share quantum chemistry data. WIREs Comput Mol Sci. 2021;11:e1491. doi:10.1002/wcms.1491

Huber SP, Zoupanos S, Uhrin M, et al. AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance. Sci Data. 2020;7:300. doi:10.1038/s41597-020-00638-4

Hjorth Larsen A, Mortensen JJ, Blomqvist J, et al. The atomic simulation environment: A Python library for working with atoms. J Phys Condens Matter. 2017;29:273002. doi:10.1088/1361-648X/aa680e

Lehtola S, Steigemann C, Oliveira MJT, Marques MAL. Recent developments in libxc: A comprehensive library of functionals for density functional theory. SoftwareX. 2018;7:1-5. doi:10.1016/j.softx.2017.11.002

Boiko DA, MacKnight R, Kline B, Gomes G. Autonomous chemical research with large language models. Nature. 2023;624:570-578. doi:10.1038/s41586-023-06792-0

M. Bran A, Cox S, Schilter O, et al. Augmenting large language models with chemistry tools. Nat Mach Intell. 2024;6:525-535. doi:10.1038/s42256-024-00832-8

Grimme S, Antony J, Ehrlich S, Krieg H. A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu. J Chem Phys. 2010;132:154104. doi:10.1063/1.3382344

Mardirossian N, Head-Gordon M. Thirty years of density functional theory in computational chemistry: An overview and extensive assessment of 200 density functionals. Mol Phys. 2017;115:2315-2372. doi:10.1080/00268976.2017.1333644

Weigend F, Haser M. RI-MP2: First derivatives and global consistency. Theor Chem Acc. 1997;97:331-340. doi:10.1007/s002140050269

Purvis GD, Bartlett RJ. A full coupled-cluster singles and doubles model: The inclusion of disconnected triples. J Chem Phys. 1982;76:1910-1918. doi:10.1063/1.443164

Watts JD, Gauss J, Bartlett RJ. Coupled-cluster methods with noniterative triple excitations for restricted open-shell Hartree-Fock and other general single determinant reference functions. Energies and analytical gradients. J Chem Phys. 1993;98:8718-8733. doi:10.1063/1.464480

Jablonka KM, Ai Q, Al-Feghali A, et al. 14 examples of how LLMs can transform materials science and chemistry: A reflection on a large language model hackathon. Digital Discovery. 2023;2:1233-1250. doi:10.1039/D3DD00113J

Neese F. Software update: The ORCA program system, version 6.0. WIREs Comput Mol Sci. 2025;15:e70019. doi:10.1002/wcms.70019

Bosoni E, Beal L, Bercx M, et al. How to verify the precision of density-functional-theory implementations via reproducible and universal workflows. Nat Rev Phys. 2024;6:45-58. doi:10.1038/s42254-023-00655-3

Li R, Sun Q, Zhang X, Chan GKL. Introducing GPU acceleration into the Python-based simulations of chemistry framework. J Phys Chem A. 2025;129:1459-1468. doi:10.1021/acs.jpca.4c05876

Wu X, Sun Q, Pu Z, et al. Enhancing GPU-acceleration in the Python-based simulations of chemistry frameworks. WIREs Comput Mol Sci. 2025;15:e70008. doi:10.1002/wcms.70008

Jinnouchi R, Lahnsteiner J, Karsai F, Kresse G, Bokdam M. Phase transitions of hybrid perovskites simulated by machine-learning force fields trained on the fly with Bayesian inference. Phys Rev Lett. 2019;122:225701. doi:10.1103/PhysRevLett.122.225701

Salomon-Ferrer R, Gotz AW, Poole D, Le Grand S, Walker RC. Routine microsecond molecular dynamics simulations with AMBER on GPUs. 2. Explicit solvent particle mesh Ewald. J Chem Theory Comput. 2013;9:3878-3888. doi:10.1021/ct400314y

Eastman P, Galvelis R, Pelaez RP, et al. OpenMM 8: Molecular dynamics simulation with machine learning potentials. J Phys Chem B. 2024;128:109-116. doi:10.1021/acs.jpcb.3c06662

Zou Y, Cheng AH, Aldossary A, et al. El Agente: An autonomous agent for quantum chemistry. Matter. 2025;8:102263. doi:10.1016/j.matt.2025.102263

Li Manni G, Fdez. Galvan I, Alavi A, et al. The OpenMolcas Web: A community-driven approach to advancing computational chemistry. J Chem Theory Comput. 2023;19:6933-6991. doi:10.1021/acs.jctc.3c00182

Hertwig RH, Koch W. On the parameterization of the local correlation functional. What is Becke-3-LYP? Chem Phys Lett. 1997;268:345-351. doi:10.1016/S0009-2614(97)00207-8

Wheeler SE, Houk KN. Integration grid errors for meta-GGA-predicted reaction energies: Origin of grid errors for the M06 suite of functionals. J Chem Theory Comput. 2010;6:395-404. doi:10.1021/ct900639j

Shirts MR, Klein C, Swails JM, et al. Lessons learned from comparing molecular dynamics engines on the SAMPL5 dataset. J Comput Aided Mol Des. 2017;31:147-161. doi:10.1007/s10822-016-9977-1

Ganose AM, Sahasrabuddhe H, Asta M, et al. Atomate2: Modular workflows for materials science. Digital Discovery. 2025;4:1944-1973. doi:10.1039/D5DD00019J

Sinha V, Laan JJ, Pidko EA. Accurate and rapid prediction of pKa of transition metal complexes: Semiempirical quantum chemistry with a data-augmented approach. Phys Chem Chem Phys. 2021;23:2557-2567. doi:10.1039/D0CP05281G

Ryan H, Carter M, Stenmark P, Stewart JJP, Braun-Sand SB. A comparison of X-ray and calculated structures of the enzyme MTH1. J Mol Model. 2016;22:168. doi:10.1007/s00894-016-3025-x

Vuong VQ, Cevallos C, Hourahine B, et al. Accelerating the density-functional tight-binding method using graphical processing units. J Chem Phys. 2023;158:084802. doi:10.1063/5.0130797

Carnimeo I, Affinito F, Baroni S, et al. Quantum ESPRESSO: One further step toward the exascale. J Chem Theory Comput. 2023;19:6992-7006. doi:10.1021/acs.jctc.3c00249

Goto K, van de Geijn RA. Anatomy of high-performance matrix multiplication. ACM Trans Math Softw. 2008;34:12. doi:10.1145/1356052.1356053

Van Zee FG, van de Geijn RA. BLIS: A framework for rapidly instantiating BLAS functionality. ACM Trans Math Softw. 2015;41:14. doi:10.1145/2764454

Marek A, Blum V, Johanni R, et al. The ELPA library: Scalable parallel eigenvalue solutions for electronic structure theory and computational science. J Phys Condens Matter. 2014;26:213201. doi:10.1088/0953-8984/26/21/213201

Kwasniewski G, Kabic M, Besta M, VandeVondele J, Solca R, Hoefler T. Red-blue pebbling revisited: Near optimal parallel matrix-matrix multiplication. In: Proceedings of SC19, the International Conference for High Performance Computing, Networking, Storage and Analysis. ACM; 2019:1-22. doi:10.1145/3295500.3356181

Sun Q. Libcint: An efficient general integral library for Gaussian basis functions. J Comput Chem. 2015;36:1664-1671. doi:10.1002/jcc.23981

Neese F. The SHARK integral generation and digestion system. J Comput Chem. 2023;44:381-396. doi:10.1002/jcc.26942

Ufimtsev IS, Martinez TJ. Quantum chemistry on graphical processing units. 1. Strategies for two-electron integral evaluation. J Chem Theory Comput. 2008;4:222-231. doi:10.1021/ct700268q

Ufimtsev IS, Martinez TJ. Quantum chemistry on graphical processing units. 2. Direct self-consistent-field implementation. J Chem Theory Comput. 2009;5:1004-1015. doi:10.1021/ct800526s

Ufimtsev IS, Martinez TJ. Quantum chemistry on graphical processing units. 3. Analytical energy gradients, geometry optimization, and first principles molecular dynamics. J Chem Theory Comput. 2009;5:2619-2628. doi:10.1021/ct9003004

Manathunga M, Miao Y, Mu D, Gotz AW, Merz KM Jr. Parallel implementation of density functional theory methods in the Quantum Interaction Computational Kernel program. J Chem Theory Comput. 2020;16:4315-4326. doi:10.1021/acs.jctc.0c00290

Miao Y, Merz KM Jr. Acceleration of high angular momentum electron repulsion integrals and integral derivatives on graphics processing units. J Chem Theory Comput. 2015;11:1449-1462. doi:10.1021/ct500984t

Frigo M, Johnson SG. The design and implementation of FFTW3. Proc IEEE. 2005;93:216-231. doi:10.1109/JPROC.2004.840301

Householder AS. Unitary triangularization of a nonsymmetric matrix. J ACM. 1958;5:339-342. doi:10.1145/320941.320947

Bowdler H, Martin RS, Reinsch C, Wilkinson JH. The QR and QL algorithms for symmetric matrices. Numer Math. 1968;11:293-306. doi:10.1007/BF02166681

Cuppen JJM. A divide and conquer method for the symmetric tridiagonal eigenproblem. Numer Math. 1981;36:177-195. doi:10.1007/BF01396757

Gu M, Eisenstat SC. A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem. SIAM J Matrix Anal Appl. 1995;16:172-191. doi:10.1137/S0895479892241287

Haser M, Ahlrichs R. Improvements on the direct SCF method. J Comput Chem. 1989;10:104-111. doi:10.1002/jcc.540100111

Obara S, Saika A. Efficient recursive computation of molecular integrals over Cartesian Gaussian functions. J Chem Phys. 1986;84:3963-3974. doi:10.1063/1.450106

Tomov S, Dongarra J, Baboulin M. Towards dense linear algebra for hybrid GPU accelerated manycore systems. Parallel Comput. 2010;36:232-240. doi:10.1016/j.parco.2009.12.005

Borstnik U, VandeVondele J, Weber V, Hutter J. Sparse matrix multiplication: The distributed block-compressed sparse row library. Parallel Comput. 2014;40:47-58. doi:10.1016/j.parco.2014.03.012

Martin RS, Reinsch C, Wilkinson JH. Householder's tridiagonalization of a symmetric matrix. Numer Math. 1968;11:181-195. doi:10.1007/BF02161841

Unreviewed and non-indexed reports (cited in the text but outside the Web of Science reference list): Bootsma AN, Wheeler SE. Popular integration grids can result in large errors in DFT-computed free energies. ChemRxiv preprint, 2019, doi:10.26434/chemrxiv.8864204.v5. Moussa JE, Stewart JJP. MOPAC: An open-source semiempirical molecular orbital program. J Open Source Softw. 2026;11:8025, doi:10.21105/joss.08025. Remke S, Breuer A. Hello SME! Generating fast matrix multiplication kernels using the Scalable Matrix Extension. arXiv:2409.18779, 2024 (preprint). Preprints cited through online sources: arXiv:2305.19800 [W30].

Software without a journal article: Gaussian 16, Revision C.01, Gaussian, Inc., Wallingford CT; Schrodinger Suite and Spartan, vendor documentation; RDKit: Open-source cheminformatics, https://www.rdkit.org; AiMolSim documentation, https://aimolsim.com (accessed 13 November 2026).

Online Sources

W1 AiMolSim documentation: engine, methods, validation, deployment, math kernel and compare pages, https://aimolsim.com (accessed 13 November 2026).

W2 AiServa platform documentation, https://aiserva.com (accessed 13 November 2026).

W3 ORCA 6.1 release, Max Planck Institute for Coal Research, https://www.kofo.mpg.de/1018457/2025-06-17_Orca-6_1 (accessed 13 November 2026).

W4 Q-Chem, https://www.q-chem.com (Q-Chem 7.0) and purchase page https://www.q-chem.com/purchase (accessed 13 November 2026).

W5 Gaussian, Inc.: release notes and pricing pages, https://gaussian.com/relnotes and https://gaussian.com/pricing; licence terms as reproduced in the TACC Gaussian usage agreement, https://docs.tacc.utexas.edu/taccdocs/TACC_GAUSSIAN_Usage_Agreement.pdf (accessed 13 November 2026).

W6 TURBOMOLE benchmarks, Dassault Systemes BIOVIA, https://www.3ds.com/fileadmin/PRODUCTS-SERVICES/BIOVIA/PDF/TURBOMOLE-Benchmarks.pdf (accessed 13 November 2026).

W7 Molpro, https://www.molpro.net, and MetaCentrum Molpro documentation (no GPU use), https://docs.metacentrum.cz/en/docs/software/sw-list/molpro (accessed 13 November 2026).

W8 GROMACS 2025 release notes, https://manual.gromacs.org/2025.3/release-notes/2025/major/highlights.html (accessed 13 November 2026).

W9 NAMD 3.0 features, https://tcbg.illinois.edu/Research/namd/3.0/features.html (accessed 13 November 2026).

W10 AMS2025 release notes, Software for Chemistry and Materials, https://www.scm.com/ams2025 (accessed 13 November 2026).

W11 Schrodinger release 2025-4 notes, https://www.schrodinger.com/life-science/download/release-notes/release-2025-4/ (accessed 13 November 2026).

W12 qmspeedtest community benchmark, https://github.com/PedroJSilva/qmspeedtest (accessed 13 November 2026).

W13 Spartan pricing, Wavefunction Inc., https://wavefun.com/spartan-pricing (accessed 13 November 2026).

W14 Avogadro 1.101 release notes, https://avogadro.cc/install/versions/v1101.html (accessed 13 November 2026).

W15 Release and licence pages, accessed 13 November 2026: github.com/pyscf/pyscf/releases; github.com/psi4/psi4/releases; github.com/nwchemgit/nwchem/releases; msg.chem.iastate.edu/gamess/download.html; gitlab.com/Molcas/OpenMolcas; github.com/cp2k/cp2k/releases; github.com/QEF/q-e; github.com/abinit/abinit/releases; github.com/grimme-lab/xtb/releases; github.com/dftbplus/dftbplus/releases; github.com/openmopac/mopac/releases; github.com/openmm/openmm/releases; manual.gromacs.org/documentation; github.com/lammps/lammps/releases; ambermd.org/GetAmber.php; github.com/rdkit/rdkit/releases; github.com/openbabel/openbabel/releases; github.com/OpenChemistry/avogadrolibs/releases; faccts.de/orca; vasp.at/info/faq/purchase_vasp; turbomole.org/turbomole/order-turbomole; molpro.net; scm.com/support/downloads; schrodinger.com/life-science/download/release-notes; wavefun.com; academiccharmm.org/program; ks.uiuc.edu/Research/namd.

W16 ORCA manual: numerical integration, RI approximations, density functionals and FAQ, https://orca-manual.mpi-muelheim.mpg.de (accessed 13 November 2026).

W17 Q-Chem 6.0 manual, standard quadrature grids, https://manual.q-chem.com/6.0/sect_standard-quad.html (accessed 13 November 2026).

W18 GROMACS user guide (managing simulations; mdrun performance), https://manual.gromacs.org (accessed 13 November 2026).

W19 OpenMM user guide, testing and validation, https://docs.openmm.org/latest/userguide/library/07_testing_validation.html (accessed 13 November 2026).

W20 Rowan pricing, https://www.rowansci.com/pricing (accessed 13 November 2026).

W21 WebMO pricing, https://www.webmo.net/get/pricing.html (accessed 13 November 2026).

W22 Schrodinger cloud computing, https://www.schrodinger.com/cloudcomputing (accessed 13 November 2026).

W23 Materials Square, https://www.materialssquare.com (accessed 13 November 2026).

W24 SAMSON, OneAngstrom, https://www.samson-connect.net (accessed 13 November 2026).

W25 NVIDIA ALCHEMI batched geometry relaxation documentation, https://docs.nvidia.com/nim/alchemi/alchemi-bgr/latest/overview.html (accessed 13 November 2026).

W26 Meta FAIR science releases (OMol25, UMA), https://ai.meta.com/blog/meta-fair-science-new-open-source-releases/ (accessed 13 November 2026).

W27 MOPAC PM7 caveats, https://openmopac.net/PM7_and_PM6-D3H4_accuracy/PM7_caveats.html (accessed 13 November 2026).

W28 CP2K Hartree-Fock exchange exercise, https://cp2k.org/exercises:2015_pitt:hfx (accessed 13 November 2026).

W29 VASP wiki, GPU ports of VASP, https://www.vasp.at/wiki (accessed 13 November 2026).

W30 Macrocyclic peptide conformer generation, arXiv:2305.19800 (preprint, accessed 13 November 2026).

W31 NAMD performance and benchmarks, https://www.ks.uiuc.edu/Research/namd/benchmarks/ (accessed 13 November 2026).

W32 AiMolSim security page, https://aimolsim.com/security/ (accessed 13 November 2026).

W33 VYROX Chemistry Engine and VYROX Math Kernel pages, https://aimolsim.com/engine/ and https://aimolsim.com/math-kernel/ (accessed 13 November 2026).

W34 Build and library documentation, accessed 13 November 2026: psicode.org/psi4manual/master/build_planning; manual.cp2k.org/trunk/technologies/libraries.html; nwchemgit.github.io/Compiling-NWChem.html; quantum-espresso.org/Doc/user_guide; vasp.at/wiki/Linking_to_libraries; abinit.github.io/abinit_web/reqpkg.html; github.com/dftbplus/dftbplus/blob/main/INSTALL.rst; manual.gromacs.org/current/install-guide; docs.lammps.org/stable/Build_settings.html; faccts.de/docs/orca/6.0/manual/contents/detailed/shark.html; quick-docs.readthedocs.io.

W35 Intel oneAPI Math Kernel Library, Conditional Numerical Reproducibility, Intel developer documentation (accessed 13 November 2026).