LETTER

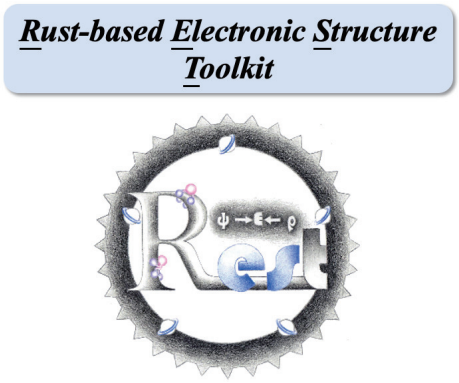
REST: Embracing the Rust Programming Language for Modern Electronic Structure Theory†

Zhiyun Li[a], Tianyi Gao[a], Shirong Wang[a,b], Sheng Bi[a,c], Rulin Feng[a], Zhenyu Zhu[a,b], Yilin Zhao[a,b], Wenjie Yan[a], Lingyue Yu[a,b], Qirui Gao[a], Zihan Lin[a], Jianming Wu[a], Igor Ying Zhang[a,b*], Xin Xu[a,b*]

a. Research Center for Chemical Theory, State Key Laboratory of Porous Materials for Separation and Conversion, Collaborative Innovation Center of Chemistry for Energy Materials, Shanghai Key Laboratory of Molecular Catalysis and Innovative Materials, MOE Key Laboratory of Computational Physical Sciences, Department of Chemistry, Fudan University, Shanghai 200433, China
b. Hefei National Laboratory, Hefei 230088, China
c. Research Center for New Materials Computing, Zhejiang Lab, Hangzhou 311100, China

(Dated: Received on October 10, 2025; Accepted on October 30, 2025)

REST (Rust-based electronic structure toolkit) is a modern open-source electronic structure code entirely written in Rust, combining high performance, memory safety, and expressive concurrency. As a community-driven project, its source code is freely available at https://gitee.com/restgroup, fostering open collaboration and transparent development. It supports a wide range of density functional methods— from local density approximation (LDA), generalized gradient approximation (GGA), meta-GGA, and hybrids to doubly hybrids, as well as machine learning-augmented functionals—enabling high-accuracy simulations with low computational overhead. Its "disk-free" RI-based (RI: resolution-of-the-identity) implementation and efficient shared-memory parallelism (via Rayon) ensure rapid calculations even for challenging systems. REST also offers unique user support through large language model-assisted input generation and developer-friendly tensor libraries for rapid algorithm prototyping.



Key words: Density functional theory, Rust programming language, XYG3-type doubly hybrid (xDH), Fearless concurrency, Open-source software

Density functional theory (DFT) has firmly established itself as the most important and widely used first-principles computational method for investigating microscopic electronic structures across materials science, physics, and chemistry, *etc.* The remarkable success of DFT is deeply intertwined with the development of efficient and robust software platforms that implement its formalism. In the domain of quantum chemistry, several major electronic structure packages—including Gaussian [1], Q-Chem [2], ORCA [3], and PySCF [4], just to name a few—leverage Gaussian-type basis sets to provide researchers with powerful tools for simulating molecular systems. Density functional approximations (DFAs) such as B3LYP [5–7] and PBE [8], formulated within the framework of mean-field formalism and implemented in these platforms, have been pivotal in the rapid expansion of computational research paradigms.

† Part of Virtual Special Issue of "Methods, Algorithms, Codes, and Software for Theoretical and Computational Chemistry".
* Authors to whom correspondence should be addressed.
E-mail: igor_zhangying@fudan.edu.cn, xxchem@fudan.edu.cn

Today, DFT-based simulations have become an indispensable tool for interpreting experimental data and predicting chemical properties.

In recent years, the paradigm of scientific research has gradually shifted from traditional computational approaches toward data-driven artificial intelligence (AI) methodologies [9, 10]. This transition imposes higher demands on both the accuracy of the data produced and the efficiency of data production. However, a persistent dilemma exists in the selection of DFAs: generalized gradient approximation (GGA) functionals such as PBE [8] are computationally efficient but often lack sufficient accuracy required for reliable description of chemical bonding, while more sophisticated approaches, such as the XYG3-type doubly-hybrid functionals (xDH) [11], offer higher accuracy at much increased computational costs. Hybrid functionals such as B3LYP [5–7] remain a practical compromise, not due to excellence, but out of necessity. Hence, the implementation of accurate density functional methods that simultaneously deliver high efficiency is becoming an urgent requirement for the development of electronic structure software in the age of AI.

Another fundamental challenge in designing next-generation electronic structure software lies in the inherent complexity of its ecosystem [12, 13]. Successful electronic structure programs often result from collaborative efforts spanning research groups worldwide. Yet, the integration of contributors with varying levels of programming expertise poses a significant difficulty. Given that highly parallel and fully optimized electronic structure computations frequently involve intricate memory operations, constructing a robust development environment that accommodates diverse coding competencies remains a critical—and still open—question for achieving sustainable and collaborative software development.

Rust-based electronic structure toolkit (REST) is the first open-source electronic structure software developed in Rust, a modern programming language designed for both safety and performance (see FIG. 1). It integrates and extends a range of advanced quantum chemical numerical algorithms. Compared with traditional languages such as C++ and Fortran, which are used in mainstream electronic structure packages like Gaussian [1] and VASP [14–17], Rust provides REST with several foundational advantages as a system language emphasizing safety and efficiency. Rust's unique ownership system and borrow checker prevent many

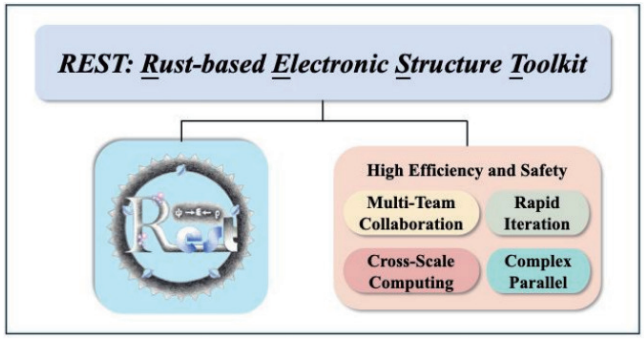


FIG. 1 Schematic of Rust-based electronic structure toolkit (REST) and its logo.

common memory-related errors, such as dangling pointers and data races—at compile time. This memory safety mechanism not only enhances code reliability but also significantly reduces runtime failures, enabling REST to handle complex quantum chemical calculations with greater robustness. Furthermore, Rust was designed with native support for modern multicore architectures. Leveraging this capability, REST achieves "fearless concurrency" in its parallel implementation, substantially accelerating computational performance.

In this letter, we will not revisit the foundational principles of electronic structure theory or the general architecture of computational programs, as these are thoroughly covered in existing reviews and methodological articles [18–21]. Similarly, detailed technical aspects of REST's implementation and usage will not be discussed here; readers who are interested in such information are referred to the official REST program website [22]. Furthermore, following the First REST Workshop held in September 2025, comprehensive installation guides and hands-on tutorial materials have been prepared and are now publicly available on the website (https://gitee.com/restgroup/workshop_for_rest). We encourage the community to access these resources for practical guidance. Instead, the focus of this communication is to highlight the distinctive features of the REST package, particularly in the areas of density functional methods, computational efficiency, and user experience, which distinguish it from conventional electronic structure software.

Versatile DFT capabilities: from mainstream to advanced methods. REST supports a comprehensive range of mainstream density functional methods. By leveraging the Libxc library [23], it incorporates the vast majority of conventional DFAs based on the mean-field Kohn-Sham framework, including local density approximation (LDA), GGA, meta-GGA, and hybrid

DOI: 10.1063/1674-0068/cjcp2510156

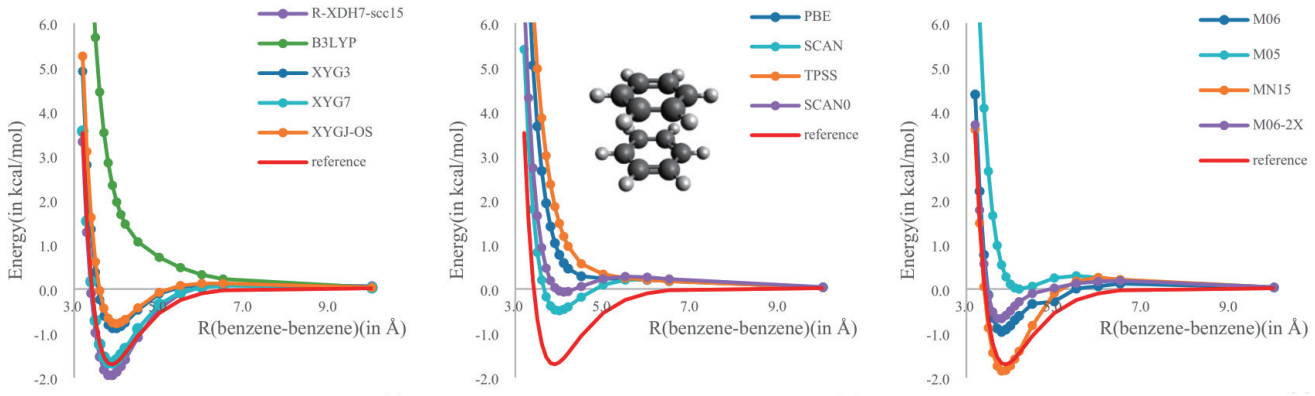


FIG. 2 Assessment of some representative DFAs for benzene dimer dissociation (def2-TZVP/def2-SV(P)-JKFIT). For benchmarking, reference data were prepared using the CCSD(T) method extrapolated to the complete basis set limit (CBS).

functionals. Examples of supported functionals span widely used DFAs such as PBE [8], B3LYP [5–7], X3LYP [24], the empirical Minnesota series (*e.g.*, M05 [25], M06 [26], and MN15 [27]), as well as the non-empirical meta-GGAs of TPSS [28] and SCAN [29] developed by Perdew and collaborators. Moreover, popular empirical dispersion corrections (D3 [30] and D3BJ [31]) is available for these DFAs. At these levels of theory, REST enables both efficient single-point energy calculations and geometry optimizations using analytic forces.

However, conventional electronic structure methods often suffer from significant limitations, such as systematic underestimation of unbounded interactions [30] and transition-state energy barriers [32], as well as inaccurate descriptions of strongly correlated systems such as molecular bond dissociation [33]. To address these issues, it is essential to go beyond the conventional DFAs and incorporate advanced models of electron correlation. Our research group has developed xDH functional series [34], which includes variants such as XYG3 [11] and XYG7 [34] that employ second-order perturbation theory (PT2), the opposite-spin (OS) PT2-based XYGJ-OS [35], and the recently introduced R-xDH7-SCC15 [36], which combines renormalized PT2 (R-PT2) with a machine learning-powered strong correlation correction (SCC15) model. These functionals, implemented in REST, collectively overcome many of the inherent limitations of conventional DFAs and significantly improve accuracy across a wide range of chemical interactions.

As illustrated in FIG. 2 and FIG. 3, which compare interaction energy curves for benzene dimer (nonbounded interactions) and ethane C–C bond dissociation (strong correlation), xDH methods based on (OS)-

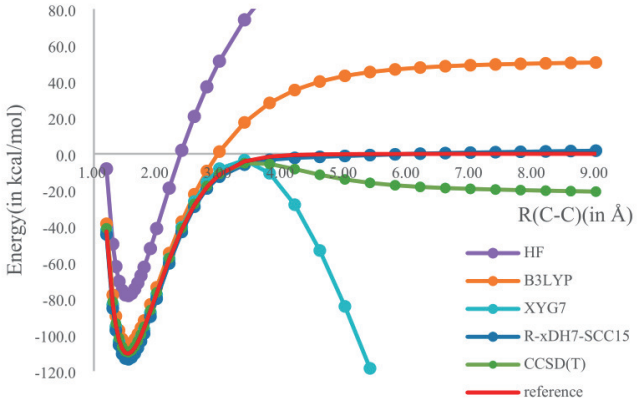


FIG. 3 Assessment of some representative electronic structure methods for the ethane C–C bond dissociation (def2-QZVPPD/def2-SV(P)-JKFIT). For benchmarking, reference data were prepared using the CCSD(T) method extrapolated to CBS for the equilibrium region and the NVEPT2 method for the middle and dissociation limit regions.

PT2—such as XYG3, XYG7, and XYGJ-OS—demonstrate markedly improved performance for nonbounded interactions and energy barriers compared to conventional DFAs. However, these methods remain inadequate for describing bond dissociation processes due to intrinsic limitations of standard perturbation theory [37]. Our recently developed R-xDH7-SCC15 represents the first single-reference framework capable of simultaneously and accurately modeling nonbounded interactions, transition-state barriers, and bond breaking/formation behaviors [36]. All these advanced xDH methods are efficiently implemented in REST for single-point energy calculations and, in its current version, REST also supports geometry optimization using numerical forces.

REST further extends the applicability of its ad-

DOI: 10.1063/1674-0068/cjcp2510156

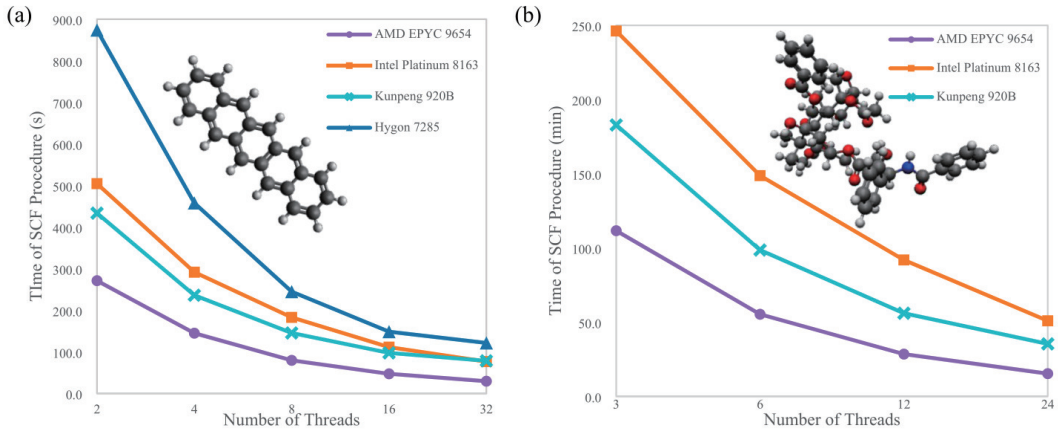


FIG. 4 Parallel scaling of X3LYP performance: (a) pentacene system (36 atoms, def2-TZVP, def2-SV(P)-JKFIT, 766 basis functions, 146 electrons); (b) paclitaxel system (113 atoms, def2-TZVP, def2-SV(P)-JKFIT, 2228 basis functions, 452 electrons).

vanced density functional methods to excited-state calculations through strategies such as ΔSCF [38, 39] and constrained DFT (C-DFT) [40, 41]. To mitigate potential spin contamination issues commonly encountered in excited-state computations, the package also incorporates correction schemes such as the Yamaguchi approach [42] and the restricted open-shell Kohn–Sham (ROKS) framework [43, 44]. Additionally, REST features preliminary implementations of many-body methods widely used in computational physics and materials science for accurate excited-state property prediction, including the GW approximation [45] and the Bethe–Salpeter equation (BSE) [46]. Detailed usage instructions are available on the official REST website. Accuracy evaluations and further methodological developments of these approaches for excited-state properties are currently in progress and will be detailed in forthcoming publications.

Efficient and scalable performance for modern computational challenges. To position itself as a next-generation electronic structure code in the age of AI, REST not only implements advanced density functional methods with high accuracy, but also prioritizes efficient and high-performance program design. To balance computational cost with system size, REST adopts a default "disk-free" implementation and utilizes the Resolution-of-Identity (RI) approximation [47] for four-center electron repulsion integrals, which significantly reduce memory footprint and I/O overhead. On the REST platform, we are actively exploring the application of lower-scaling algorithms and tensor low-rank decomposition techniques. These efforts aim to enable routine xDH calculations for systems with hundreds of atoms using hardware commonly available to researchers (*e.g.*, a single node with ≤100 cores and ≤500 GB memory). We expect to report these advances in a forthcoming publication.

Benefiting from Rust's unique ownership system, all algorithms in REST can seamlessly and safely leverage fine-grained parallelism within a single node (shared memory) through Rayon, a library that enables "fearless concurrency" analogous to OpenMP, while maintaining excellent parallel performance. As illustrated in FIG. 4, the parallel scaling of the hybrid functional X3LYP was evaluated on an x86 machine (AMD EPYC 9654 and Intel Platinum 8163) and domestic machines (Kunpeng 920B and Hygon 7285) for two test systems: a pentacene system (36 atoms, 766 basis functions) and a paclitaxel molecule (113 atoms, 2228 basis functions). As demonstrated in FIG. 4(a) for the pentacene system, scaling from 2 to 32 computing cores achieves a speedup factor of approximately 7–9 for AMD EPYC 9654 and Intel Platinum 8163 processors, and 5–7 for Hygon 7285 processors. The parallel efficiency improves for larger systems, as demonstrated by the results for the larger paclitaxel molecule in FIG. 4(b). Benchmarking on an AMD EPYC 9654 processor shows that the execution time decreases significantly from 111 min to 55, 28, and 15 min as the number of threads increases from 3 to 6, 12, and 24. This near-linear scaling demonstrates highly efficient parallelism. Furthermore, REST has been successfully ported to ARM-based architectures, such as Huawei's Kunpeng processors (model: Kunpeng 920B). REST exhibits similarly strong parallel scaling on this domestic hardware, although the total execution time is approximately 2–3 times longer than on

DOI: 10.1063/1674-0068/cjcp2510156

TABLE I Time consumption for several sizable molecules (Intel(R) Xeon(R) Gold 6150 CPU @ 2.70GHz, 36 cores). Relevant timing data are collected from Ref.[48]. N_{atom}: number of atoms in the system; N_{occ}: number of occupied molecular orbitals; N_{vir}: number of virtual molecular orbitals. $t_{per\ iter}$: time per SCF iteration for the B3LYP hybrid functional; t_{PT2}: time for PT2 calculation in the XYG7 doubly hybrid functional; t_{rPT2}: time for renormalized PT2 calculation; t_{SCC15}: time for the SCC15 model.

Species	N_{atom}	N_{occ}	N_{vir}	$t_{per\ iter}$	t_{PT2}	t_{rPT2}	t_{SCC15}
ED08	72	162	3202	~488	~282	~343	~464
PR08	74	163	3261	~490	~305	~354	~478
ED09	70	155	3095	~420	~245	~303	~386
PR09	72	156	3154	~460	~456	~314	~409
PR24	120	205	2360	~310	~392	~430	~506
PR25	117	210	2370	~300	~431	~453	~564
ED41	74	130	2943	~420	~258	~272	~279
PR41	74	130	2943	~370	~261	~220	~279

AMD CPUs.

During the development of REST, particular emphasis has been placed on optimizing the performance of advanced density functional methods, especially the xDH family of doubly hybrid functionals. As shown in Table I, for transition-metal complexes containing 70–120 atoms and approximately 3000 basis functions [48] comprising both main-group and transition-metal elements, the computational cost of evaluating advanced correlation energy models such as PT2 and opposite-spin variant (OS-PT2), renormalized PT2 (rPT2), and the strong-correlation correction model (SCC15) in REST is comparable to that of a single self-consistent field (SCF) iteration in a hybrid functional calculation like B3LYP. This remarkable efficiency implies that, by using REST, the accuracy of electronic structure calculations can be elevated from the hybrid functional level to that of high-accuracy xDHs at only a modest additional cost, approximately equivalent to one extra SCF iteration. Such a low overhead significantly enhances the practicality of high-level methods for generating precise and reliable data in production-scale studies.

Due to its "disk-free" strategy and use of Rayon for shared-memory parallelization, REST is currently constrained by the memory limits of individual compute nodes. In the tests described above, the largest systems required approximately 1 TB of memory. To overcome this constraint, a hybrid MPI/Rayon parallelization scheme has been preliminarily implemented, extending REST's capabilities to multi-node distributed-memory environments. Initial benchmarks show that this hybrid approach achieves excellent parallel efficiency for conventional density functional methods. However, for advanced functionals that involve intensive communication such as xDH methods, the parallel scalability remains suboptimal. To mitigate this issue, we are actively developing strategies that leverage the sparsity of integrals by improved RI techniques and incorporate tensor low-rank decomposition methods. These algorithmic improvements aim to significantly reduce memory and communication demands, thereby enabling larger systems to be treated efficiently within a single node. Promising progress has been made in this direction, and detailed results will be reported in a forthcoming publication.

Enhancing usability with large language models. The success of computational software often hinges on comprehensive user documentation and responsive technical support, resources that are typically beyond the capacity of medium-sized academic research groups. Coinciding with the development of REST, rapid advances in large language models (LLMs), such as ChatGPT, DeepSeek, and Tongyi, have created new opportunities to improve user experience with minimal manual intervention.

To leverage these tools, we have systematically restructured the core architecture of REST, its input structure, and key parameters into a well-organized Markdown document (the README.md file in the REST repository). This structured format allows LLMs to effectively parse and reason over the software's interface. For instance, by uploading this document to platforms like DeepSeek and providing task descriptions in natural language, users can automatically generate ready-to-run input files for REST. Notably, thanks to the strong multilingual translation and comprehension abilities of modern LLMs, the fact that the REST README.md file is written primarily in Chinese (the developers' native language) poses no barrier to users who wish to inquire about keyword usage and input card syntax, or to generate task-specific input files in other languages, such as English. This cross-lingual flexibility significantly lowers the accessibility threshold for international users.

FIG. 5 illustrates the workflow of employing an LLM to generate and execute REST input files. As shown in the example of calculating the energy of a CH_4 molecule with the XYG7 method, the LLM successfully trans-

DOI: 10.1063/1674-0068/cjcp2510156

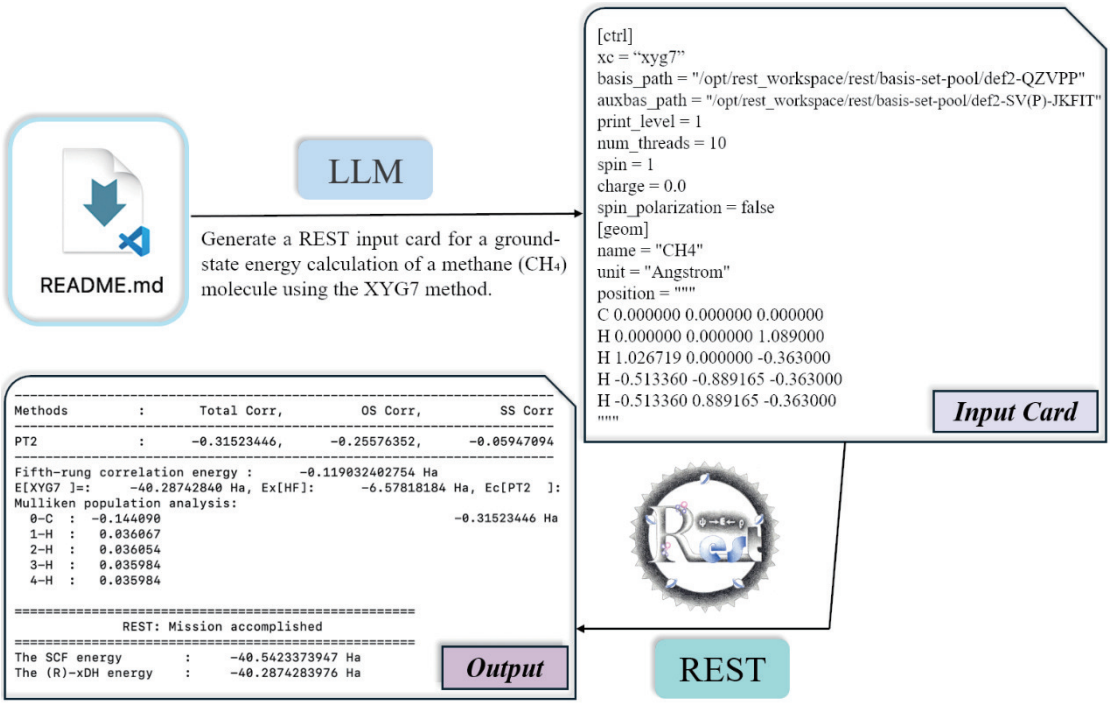


FIG. 5 Workflow for generating and executing REST input with LLM assistance.

lates a user prompt written in Chinese into a syntactically correct and physically meaningful input file, without requiring the user to have prior expertise in electronic structure theory. The generated input is then directly executed by REST to perform the computation, yielding the desired molecular energy. This example highlights how LLMs can bridge both knowledge and language gaps, making advanced quantum chemical calculations accessible to a broader range of researchers with minimal learning overhead.

Developer-friendly features for method implementation. Electronic structure calculations rely heavily on complex matrix and tensor operations. Although Rust is a relatively young language in the domain of scientific computing and lacks mature libraries for such tasks, its strong memory safety guarantees and powerful support for generic programming have enabled us to develop dedicated matrix and tensor libraries—*rest_tensors* and *rstsr*—tailored to the needs for quantum chemical applications. These libraries provide a stable and high-performance foundation for implementing cutting-edge electronic structure algorithms within REST. For detailed documentation, users are referred to the respective resources for *rest_tensors* [49] and *rstsr* [50].

The *rest_tensors* library is specifically designed for two-dimensional matrix operations and numerical computations, which are ubiquitous in quantum chemistry. However, Rust's inherent emphasis on memory safety imposes strict constraints on pointer manimpulation, making it difficult to perform the low-level, "white-box" style of direct memory access commonly available in traditional languages like C or Fortran. To address this issue, *rest_tensors* is carefully architected to support a "white-box" operational paradigm, allowing users to directly inspect and manipulate matrix data, while fully adhering to Rust's safety principles. A key aspect of this design is its adherence to the column-major memory layout, identical to that used in Fortran. This ensures optimal data locality and cache efficiency for many linear algebra operations, and enables seamless compatibility with established numerical libraries (*e.g.*, BLAS, LAPACK) that expect this standard memory ordering.

This is achieved by first leveraging Rust's powerful generics to abstract common matrix characteristics into a *BasicMatrix* trait, which unifies the representation of matrix dimensions and provides standardized data access operations. Furthermore, the library makes extensive use of Rust's iterator system to encapsulate and implement zero-overhead, efficient indexing for various matrix element access patterns, such as single or multiple rows/columns, diagonal elements, and even arbitrary submatrices. FIG. 6 shows the corresponding code for matrix data access and operations utilizing the *rest_tensors* library. These features facilitate the conve-

DOI: 10.1063/1674-0068/cjcp2510156

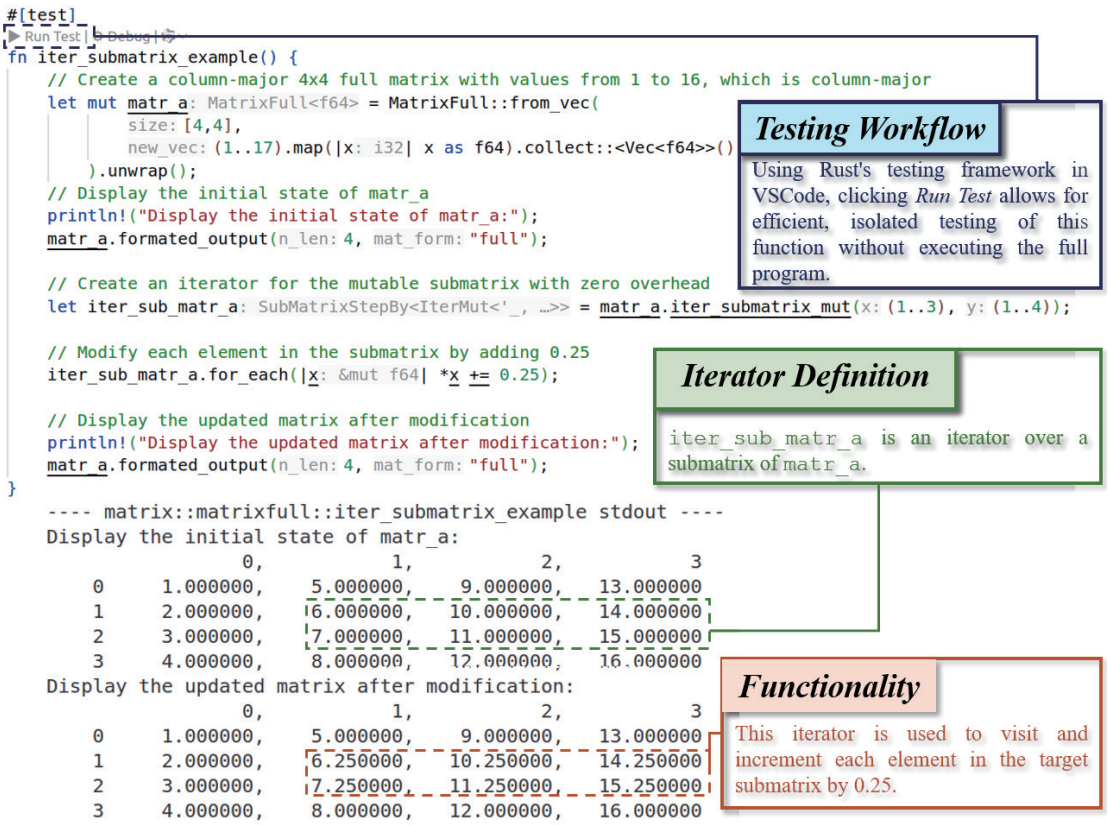


FIG. 6 Code paradigm for matrix access and manipulation via iterators in rest_tensors.

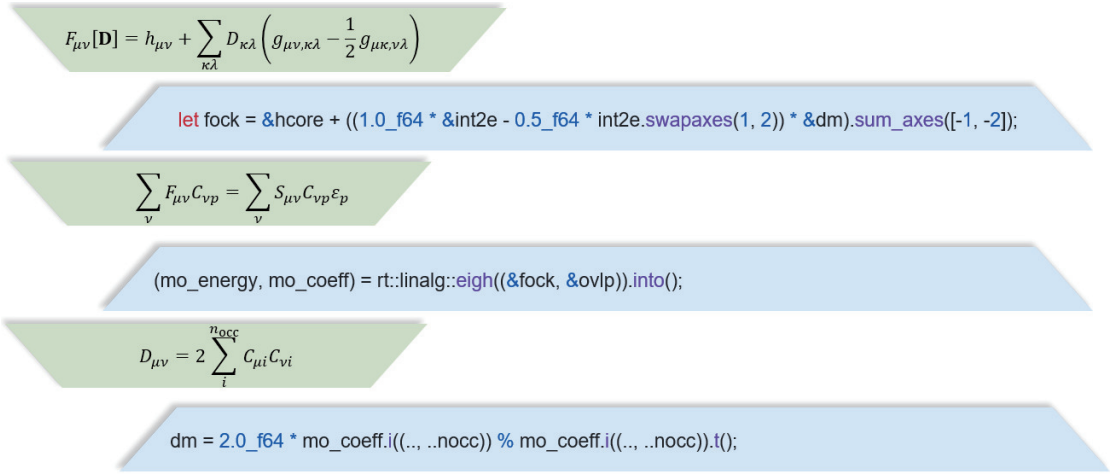


FIG. 7 Implementation of SCF iteration kernel using rstsr.

nient construction of matrix types common in quantum chemistry, such as the general-purpose "*struct MatrixFull*", and its specialized variants, such as triangular and constant matrices. In this regard, *rest_tensors* adheres to a lightweight and open design philosophy, welcoming developers to customize matrix structures and develop algorithms according to their computational needs.

Complementing this, the *rstsr* tensor library— initially developed by Dr. Zhu in our group— provides a general-purpose framework for multi-dimensional arrays. It adopts a highly abstract design philosophy, representing a "black-box" interface to users that enables NumPy-like vectorized operations in syntax. This approach allows for a clear separation between the user-facing API and backend optimizations. As illustrated in FIG. 7, the self-consistent field (SCF) iteration kernel can be implemented in as few as three lines

DOI: 10.1063/1674-0068/cjcp2510156

of code using the *rstsr* tensor library. This conciseness demonstrates that REST is not only an efficient simulation tool for production-level calculations, but also a powerful and agile platform for methodological development. By combining code safety with high-level expressiveness, REST allows developers to rapidly prototype, test, and optimize new algorithms, ensuring that novel theoretical approaches can be translated into high-performance applications. This capability significantly accelerates research cycles and solidifies REST's role as a versatile tool for tackling frontier scientific challenges.

Concluding remarks. In summary, REST represents a distinctive advancement in electronic structure software by leveraging features of memory safety, high performance, and built-in concurrency of the Rust programming language. It integrates a wide spectrum of density functional methods from mainstream DFAs to advanced doubly hybrids, while offering efficient, scalable, and user-friendly computational tools tailored for both production-level simulations and methodological innovation. To further expand its versatility and applicability, we are actively developing several key extensions, including the implementation of a polarizable continuum model (PCM) for implicit solvation effects [51], the exact two-component (X2C) approach for relativistic corrections [52, 53], analytic gradients for xDHs to enable efficient geometry optimizations, as well as spectroscopic simulation capabilities such as infrared (IR), Raman [54, 55], and nuclear magnetic resonance (NMR) spectroscopy [56]. These ongoing efforts aim to broaden REST's scope across a wide range of chemical environments and physical properties, reinforcing its role as a next-generation, open-ended platform for cutting-edge computational research.

AUTHOR CONTRIBUTIONS

The REST project was initiated under the leadership of Prof. Xin Xu and is directed by Prof. Igor Ying Zhang, who designed its overall architecture and implemented the core functionalities independently. The computational data and figures presented in this work were produced and organized by Zhiyun Li. The author list is ordered primarily according to the sequence of their joining the project. For a detailed breakdown of individual contributions, please visit: https://gitee.com/restgroup.

NOTES

There are no conflicts to declare.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Nos. 22125301, 22393911, 22393912, 22321003, 22233002), the Innovation Program for Quantum Science and Technology (2021ZD0303305), and the robotic AI-Scientist platform of the Chinese Academy of Science. Part of the numerical calculations were performed on the Hefei Advanced Computing Center.

[1] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J. Fox, *Gaussian 16, Revision A.03*, Wallingford, CT: Gaussian, Inc., (2016).
[2] Y. Shao, Z. Gan, E. Epifanovsky, A. T. Gilbert, M. Wormit, J. Kussmann, A. W. Lange, A. Behn, J. Deng, and X. Feng, Mol. Phys. **113**, 184 (2015).
[3] F. Neese, WIREs Comput. Mol. Sci. **2**, 73 (2012).
[4] Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. D. McClain, E. R. Sayfutyarova, and S. Sharma, WIREs Comput. Mol. Sci. **8**, e1340 (2018).
[5] A. D. Becke, Phys. Rev. A **38**, 3098 (1988).
[6] A. D. Becke, J. Chem. Phys. **98**, 5648 (1993).
[7] C. Lee, W. Yang, and R. G. Parr, Phys. Rev. B **37**, 785 (1988).
[8] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).
[9] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Rev. Mod. Phys. **91**, 045002 (2019).

DOI: 10.1063/1674-0068/cjcp2510156

[10] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, Nature **559**, 547 (2018).
[11] Y. Zhang, X. Xu, and W. A. Goddard III, Proc. Natl. Acad. Sci. USA **106**, 4963 (2009).
[12] G. Wilson, D. A. Aruliah, C. T. Brown, N. P. Chue Hong, M. Davis, R. T. Guy, S. H. Haddock, K. D. Huff, I. M. Mitchell, and M. D. Plumbley, PLoS Biol. **12**, e1001745 (2014).
[13] A. Krylov, T. L. Windus, T. Barnes, E. Marin-Rimoldi, J. A. Nash, B. Pritchard, D. G. Smith, D. Altarawy, P. Saxe, and C. Clementi, J. Chem. Phys. **149**, 180901 (2018).
[14] G. Kresse and J. Furthmüller, Comput. Mater. Sci. **6**, 15 (1996).
[15] G. Kresse and J. Furthmüller, Phys. Rev. B **54**, 11169 (1996).
[16] G. Kresse and J. Hafner, Phys. Rev. B **47**, 558 (1993).
[17] G. Kresse and D. Joubert, Phys. Rev. B **59**, 1758 (1999).
[18] R. M. Martin, *Electronic Structure: Basic Theory and Practical Methods*, Cambridge university press, (2020).
[19] A. D. Becke, J. Chem. Phys. **140**, 18A301 (2014).
[20] J. P. Perdew and K. Schmidt, *Jacob's Ladder of Density Functional Approximations for the Exchange-correlation Energy*, American Institute of Physics, (2001).
[21] W. Kohn, A. D. Becke, and R. G. Parr, J. Phys. Chem. **100**, 12974 (1996).
[22]]REST_Group, (2025). https://gitee.com/restgroup
[23] S. Lehtola, C. Steigemann, M. J. Oliveira, and M. A. Marques, SoftwareX **7**, 1 (2018).
[24] X. Xu and W. A. Goddard III, Proc. Natl. Acad. Sci. USA **101**, 2673 (2004).
[25] Y. Zhao, N. E. Schultz, and D. G. Truhlar, J. Chem. Phys. **123**, 161103 (2005).
[26] Y. Zhao and D. G. Truhlar, Theor. Chem. Acc. **120**, 215 (2008).
[27] S. Y. Haoyu, X. He, S. L. Li, and D. G. Truhlar, Chem. Sci. **7**, 5032 (2016).
[28] J. Tao, J. P. Perdew, V. N. Staroverov, and G. E. Scuseria, Phys. Rev. Lett. **91**, 146401 (2003).
[29] J. Sun, A. Ruzsinszky, and J. P. Perdew, Phys. Rev. Lett. **115**, 036402 (2015).
[30] S. Grimme, J. Antony, S. Ehrlich, and H. Krieg, J. Chem. Phys. **132**, 154104 (2010).
[31] S. Grimme, S. Ehrlich, and L. Goerigk, J. Comput. Chem. **32**, 1456 (2011).
[32] Y. Zhao and D. G. Truhlar, J. Chem. Phys. **125**, 194101 (2006).
[33] A. J. Cohen, P. Mori-Sánchez, and W. Yang, Science **321**, 792 (2008).
[34] I. Y. Zhang and X. Xu, J. Phys. Chem. Lett. **12**, 2638 (2021).
[35] I. Y. Zhang, X. Xu, Y. Jung, and W. A. Goddard III, Proc. Natl. Acad. Sci. USA **108**, 19896 (2011).
[36] Y. Wang, Z. Lin, R. Ouyang, B. Jiang, I. Y. Zhang, and X. Xu, JACS Au **4**, 3205 (2024).
[37] R. J. Bartlett and M. Musiał, Rev. Mod. Phys. **79**, 291 (2007).
[38] T. Ziegler, E. J. Baerends, J. G. Snijders, W. Ravenek, and V. Tschinke, J. Phys. Chem. **93**, 3050 (1989).
[39] E. Vandaele, M. Mališ, and S. Luber, J. Chem. Phys. **156**, 130901 (2022).
[40] B. Kaduk, T. Kowalczyk, and T. Van Voorhis, Chem. Rev. **112**, 321 (2012).
[41] Q. Wu and T. Van Voorhis, J. Chem. Theory Comput. **2**, 765 (2006).
[42] K. Yamaguchi, F. Jensen, A. Dorigo, and K. Houk, Chem. Phys. Lett. **149**, 537 (1988).
[43] C. W. Murray, N. C. Handy, and R. D. Amos, J. Chem. Phys. **98**, 7145 (1993).
[44] T. V. Russo, R. L. Martin, and P. J. Hay, J. Chem. Phys. **101**, 7729 (1994).
[45] M. S. Hybertsen and S. G. Louie, Phys. Rev. B **34**, 5390 (1986).
[46] M. Rohlfing and S. G. Louie, Phys. Rev. B **62**, 4927 (2000).
[47] X. Ren, P. Rinke, V. Blum, J. Wieferink, A. Tkatchenko, A. Sanfilippo, K. Reuter, and M. Scheffler, New J. Phys. **14**, 053020 (2012).
[48] Y. Wang, I. Y. Zhang, and X. Xu, J. Comput. Chem. **45**, 2878 (2024).
[49] REST_Group, *rest_tensors*, (2025). https://gitee.com/restgroup/rest_tensors
[50] REST_Group, *rstsr*, (2025). https://gitee.com/restgroup/rstsr
[51] J. Tomasi, B. Mennucci, and R. Cammi, Chem. Rev. **105**, 2999 (2005).
[52] M. Iliaš and T. Saue, J. Chem. Phys. **126**, 064102 (2007).
[53] W. Liu and D. Peng, J. Chem. Phys. **131**, 031104 (2009).
[54] E. B. Wilson, J. C. Decius, and P. C. Cross, *Molecular Vibrations: the Theory of Infrared and Raman Vibrational Spectra*, Courier Corporation, (1980).
[55] C. J. Cramer, *Essentials of Computational Chemistry: Theories and Models*, John Wiley & Sons, (2013).
[56] J. R. Cheeseman, G. W. Trucks, T. A. Keith, and M. J. Frisch, J. Chem. Phys. **104**, 5497 (1996).

DOI: 10.1063/1674-0068/cjcp2510156