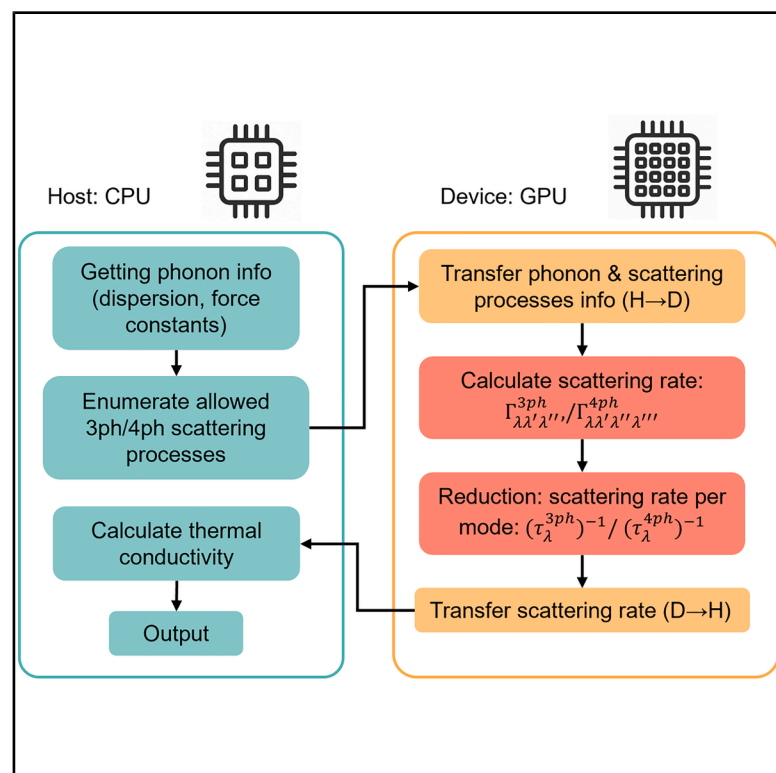


FourPhonon_GPU, a CPU-GPU heterogeneous computing scheme for phonon scattering and thermal conductivity calculations

Graphical abstract



Authors

Ziqi Guo, David McLoughlin, Xiulin Ruan, Guang Lin

Correspondence

ruan@purdue.edu (X.R.),
guanglin@purdue.edu (G.L.)

In brief

Guo et al. develop FourPhonon_GPU, a GPU-accelerated computing framework that offloads phonon scattering rate calculations using OpenACC. Under the relaxation-time approximation, the heterogeneous CPU-GPU scheme eliminates warp divergence, achieving up to 26-fold overall speedup and 400-fold acceleration for the four-phonon scattering-rate step.

Highlights

- A GPU-accelerated heterogeneous computing scheme is developed for ShengBTE
- OpenACC directives enable efficient parallelization with minimal code changes
- Divergent branching is resolved by CPU-side process filtering and packing
- Two parallelization strategies are implemented to balance memory and throughput



Article

FourPhonon_GPU, a CPU-GPU heterogeneous computing scheme for phonon scattering and thermal conductivity calculations

Ziqi Guo,¹ David McLoughlin,² Xiulin Ruan,^{1,*} and Guang Lin^{3,4,*}
¹School of Mechanical Engineering and the Birck Nanotechnology Center, Purdue University, West Lafayette, IN 47907, USA

²School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, USA

³School of Mathematics, Purdue University, West Lafayette, IN 47907, USA

⁴Lead contact

*Correspondence: ruan@purdue.edu (X.R.), guanglin@purdue.edu (G.L.)

<https://doi.org/10.1016/j.xcrp.2026.103491>

SUMMARY

Accurate prediction of phonon scattering is crucial for understanding thermal transport properties. However, the computational cost of such calculations, especially for four-phonon scattering, becomes prohibitive as the number of phonon branches and scattering processes increases. Here, we report **FourPhonon_GPU**, a GPU-accelerated framework for phonon scattering and thermal conductivity calculations. Leveraging OpenACC and a heterogeneous CPU-GPU computing strategy, we offload massive parallelizable tasks to the GPU while using the CPU for process enumeration and control-heavy operations. Under the relaxation-time approximation (RTA), our approach achieves up to $26\times$ acceleration without sacrificing physical accuracy. Benchmarking on various GPU architectures confirms the method's scalability and highlights the importance of aligning parallelization strategies with hardware capabilities. This work provides an efficient and accurate computational tool for phonon transport modeling and opens pathways for accelerated materials discovery.

INTRODUCTION

Thermal conductivity is a fundamental material property that plays a critical role in a wide range of applications, including thermal management in electronic devices^{1,2} and thermoelectric energy conversion.^{3,4} Accurately predicting thermal conductivity is essential for optimizing material performance. The primary heat carriers in dielectrics and semiconductors are phonons. Phonon scattering rates, which limit phonon mean free paths, dictate thermal conductivity.⁵ Accurate prediction of these rates is therefore key to modeling thermal transport behavior.⁶

Recent advances in first-principles calculations of phonon scattering coupled with the Boltzmann transport equation (BTE) have enabled accurate, parameter-free predictions of lattice thermal conductivity. The theoretical foundation for the phonon BTE was first laid by Peierls.⁷ Later, Maradudin et al.⁸ developed a comprehensive framework for three-phonon (3ph) scattering. Broido et al.⁹ further combined *ab initio* force constants with these approaches, leading to robust first-principles predictions of thermal conductivity. It was believed for decades that 3ph scattering is adequate for describing thermal transport except at high temperatures. However, a general theory and computational method for four-phonon (4ph) scattering were lacking, preventing its quantitative evaluation. Feng and Ruan^{10,11} developed the general theory and computational method for 4ph scattering, demonstrating that 4ph processes can significantly impact the

thermal conductivity of many materials at elevated temperatures or even room temperature. Their theoretical predictions for boron arsenide (BAs) were later confirmed experimentally,^{12–14} proving that 4ph scattering is critical for accurate thermal transport modeling. Beyond bulk materials, understanding higher-order anharmonic phonon processes is also critical for dictating thermal transport across interfaces.^{15,16}

Although first-principles calculations have advanced our fundamental understanding of thermal transport, they are computationally prohibitive when including 4ph scattering. The high cost arises from the need to enumerate and compute a large number of phonon scattering processes, which scale as N^3 and N^4 for 3ph and 4ph scattering, respectively, where N is the number of q-points in the Brillouin zone. For instance, a silicon calculation using a $16 \times 16 \times 16$ q-mesh (a discretized grid in reciprocal space) involves approximately 9.0×10^5 3ph and 7.6×10^9 4ph scattering processes, requiring over 7,000 CPU hours.

To address this high computational cost, several approaches have been explored, including machine-learning surrogates^{17,18} and statistical sampling methods.¹⁹ These techniques have been applied to estimate thermal and radiative properties of solids.^{20–30} While these methods reduce computational cost, they rely on approximations that sacrifice accuracy. Consequently, they are limited to applications where a certain margin of error is acceptable and are inadequate for scenarios requiring rigorous, fully resolved calculations of every scattering process.



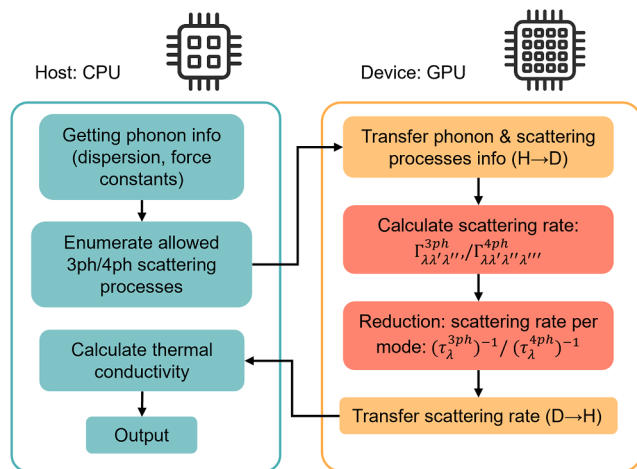


Figure 1. Schematic of the heterogeneous CPU-GPU computing framework in FourPhonon_GPU

The CPU handles control-heavy process enumeration and filtering, while the GPU executes the regular, computationally intensive scattering-rate calculations in parallel.

An alternative path is leveraging hardware acceleration through GPUs. Initially developed for rendering graphics, GPUs have evolved into powerful tools for scientific computing,³¹ machine learning,^{32–37} and high-performance simulations.³⁸ Their massively parallel architectures, featuring thousands of cores capable of running millions of concurrent threads, are well suited for independent mathematical operations.³⁹ This advantage has been demonstrated in atomistic simulations. Packages such as Abinit⁴⁰ and VASP^{41,42} support GPU acceleration, yielding orders-of-magnitude speedups over CPU-only implementations. For phonon transport, Wei et al.⁴³ offloaded scattering-rate calculations in ShengBTE⁴⁴ onto GPUs. Zhang et al.⁴⁵ subsequently developed GPU_PBT, using a two-kernel strategy to improve throughput. Under the relaxation-time approximation (RTA), each scattering process is decoupled and can be evaluated independently, making the calculation highly parallelizable.

Here, we develop a GPU-accelerated framework for both 3ph and 4ph phonon scattering calculations by combining CPU-based preprocessing with GPU parallel computing. Building on the open-source FourPhonon⁴⁶ package, we use OpenACC to enable GPU acceleration with minimal code restructuring. However, direct GPU offloading suffers from performance bottlenecks due to divergent branching. To address this, we propose a heterogeneous CPU-GPU scheme where the CPU filters and enumerates momentum- and energy-conserving processes, and the GPU evaluates the scattering rates in parallel (Figure 1). We also implement memory coalescing, loop collapse, and function inlining to maximize hardware utilization. Systematic benchmarks demonstrate that this method preserves the accuracy of the original solver while achieving up to 26× acceleration under the RTA. This framework provides an efficient pathway for rigorous phonon transport calculations on modern high-performance computing systems.

RESULTS AND DISCUSSION

Experimental setup and accuracy validation

We evaluate the performance of our GPU-accelerated framework using silicon and MgO as benchmarks. Calculations are conducted at 300 K under the RTA for both 3ph and 3ph + 4ph processes. We use broadening factors of 1.0 for 3ph and 0.1 for 3ph + 4ph calculations, consistent with previous studies.^{17,19,46} All simulations were performed on the Gilbreth cluster at Purdue University. The software and hardware configurations are summarized in Table 1.

To evaluate the performance gain, we compare the GPU-accelerated framework against a multi-core CPU baseline. The CPU baseline is measured on 32 cores and reported as total CPU time (core-hours), which serves as a reliable proxy for single-core execution time because single-core runs on dense q-meshes are computationally impractical. For the GPU-accelerated runs, we use one CPU core and one GPU and report the sum of CPU-based preprocessing and GPU scattering-rate calculations. This enables a direct comparison of the total computational cost. The calculated thermal conductivities from the CPU and GPU runs agree within 0.1%, with the minor differences arising from compiler-specific floating-point optimizations. We also compare the mode-resolved scattering rates and cumulative thermal conductivities in Figures S1–S3, which show negligible discrepancies. These results confirm that GPU acceleration does not compromise the physical accuracy of the original models.

Performance speedup of heterogeneous computing

We first analyze the speedups obtained via our heterogeneous CPU-GPU scheme. Figures 2 and 3 compare the total computational cost of the CPU baseline and the CPU-GPU heterogeneous version across different q-mesh densities for Si and MgO, respectively. Across all meshes, the heterogeneous implementation achieves up to 26× overall speedup. Specifically, we achieve approximately 13× and 14× overall speedups for Si (3ph and 3ph + 4ph, respectively), while MgO exhibits larger speedups of 14× and 26×. This greater speedup in MgO is driven by its denser scattering matrix: MgO has approximately 40% more allowed 4ph processes than Si, which increases the arithmetic intensity per q-point and leads to higher GPU thread occupancy and SIMT efficiency. Note that these total times include harmonic-properties calculation, phase-space evaluation, and post-processing, which remain CPU bound. Isolating the computational cost of the scattering-rate calculation step reveals even more substantial speedups of up to 400× for 4ph scattering-rate calculations, as shown in the insets of Figures 2 and 3. We expect even higher acceleration when the broadening factor for 3ph + 4ph calculations is increased. We also tested performance at a higher temperature (500 K) for Si, which yields similar speedups (see Figure S4). Benchmarks verify the speedup and highlight the importance of matching parallelization strategies to hardware limits.

Comparison of all-modes and mode-by-mode strategies

Next, we compare the mode-by-mode and all-modes parallelization strategies (Figure 4A). For 4ph scattering, a smaller

Table 1. Experimental hardware and software configurations

Item	Description
CPU	AMD EPYC 7543 32-Core Processor
GPU	primary: NVIDIA A100 (80GB) GPU. Tests: NVIDIA A10, A30 GPUs
Compiler (GPU)	NVIDIA HPC Compiler (nvc 23.5-0)
CUDA version	CUDA 12.6.0
Compiler (CPU)	Intel OneAPI Compilers 2024.2.1
Arithmetic precision	double precision (FP64)
MPI Version	Intel MPI 2021.13

q-mesh is used here to keep the CPU reference runs tractable; this trend is expected to hold for larger mesh sizes. All-modes parallelization yields an additional 21% speedup for 3ph ($32 \times 32 \times 32$ q-mesh) and 9% for 3ph + 4ph ($10 \times 10 \times 10$ q-mesh). This gain is driven by (1) wider parallel concurrency over all processes and (2) reduced host-device memory traffic by avoiding repeated transfers between modes. However, all-modes execution requires significantly more memory (Figure 4B). For dense meshes, this requirement can exceed physical memory capacity. Indeed, the 3ph + 4ph calculation on a $16 \times 16 \times 16$ mesh exceeded 80 GB, preventing execution under the all-modes strategy. To address this, we fall back to mode-by-mode parallelization for the $16 \times 16 \times 16$ mesh. Although less efficient, it still achieves a $7 \times$ speedup compared to the CPU baseline (Figure 4C). This trade-off emphasizes the need to select parallelization schemes based on problem size and memory constraints. In practice, both strategies are implemented in our open-source code as distinct compilation options. We recommend that users initially employ the all-modes strategy to maximize throughput. For systems with complex unit cells or high mesh densities, such as our $16 \times 16 \times 16$ 3ph + 4ph benchmark, the mode-by-mode strategy serves as a robust fallback to prevent out-of-memory errors. A detailed comparison of these two strategies across various mesh sizes for both Si and MgO is provided in Figure S5.

Comparison with GPU-only implementation

We also compare the heterogeneous CPU-GPU scheme against a GPU-only implementation (Figure 5). For 3ph scattering ($32 \times 32 \times 32$ q-mesh), the heterogeneous approach is $5 \times$ faster than the GPU-only version. For 3ph + 4ph scattering ($10 \times 10 \times 10$ q-mesh), the GPU-only code is twice as slow as the CPU baseline. This performance degradation stems from the sparsity of the 4ph scattering matrix, which causes severe warp divergence and degrades execution efficiency on the GPU. These results highlight the benefits of our heterogeneous strategy: offloading regular, highly parallel workloads to the GPU while retaining irregular, branch-heavy operations on the CPU.

Performance across different GPU architectures

Finally, we evaluate performance across different GPU architectures. Because FourPhonon uses double-precision (FP64) arithmetic, performance depends heavily on the GPU's FP64 throughput. We tested three NVIDIA GPUs (A100, A30, and A10), whose specifications are summarized in Table 2. To isolate hardware differences, we compare only the GPU kernel execution times, excluding CPU-side preprocessing. Kernel performance scales as $A100 > A30 > A10$ (Figure 6), matching their FP64 performance rankings. The A10, optimized for single-precision workloads with low FP64 throughput, is less suited for double-precision scientific calculations and exhibits a substantial performance drop in our benchmarks. The larger memory capacity of the A100 also provides an advantage for dense q-mesh calculations. For instance, a $14 \times 14 \times 14$ q-mesh under the all-modes strategy requires approximately 63 GB of memory. This exceeds the 24-GB capacity of the A30 and A10, forcing them to fall back to mode-by-mode execution, which reduces performance. These results highlight that the choice of GPU hardware must align with the floating-point precision and memory demands of the target physical system.

Future work will focus on several optimizations. First, while the current solver is limited to the RTA, extending this

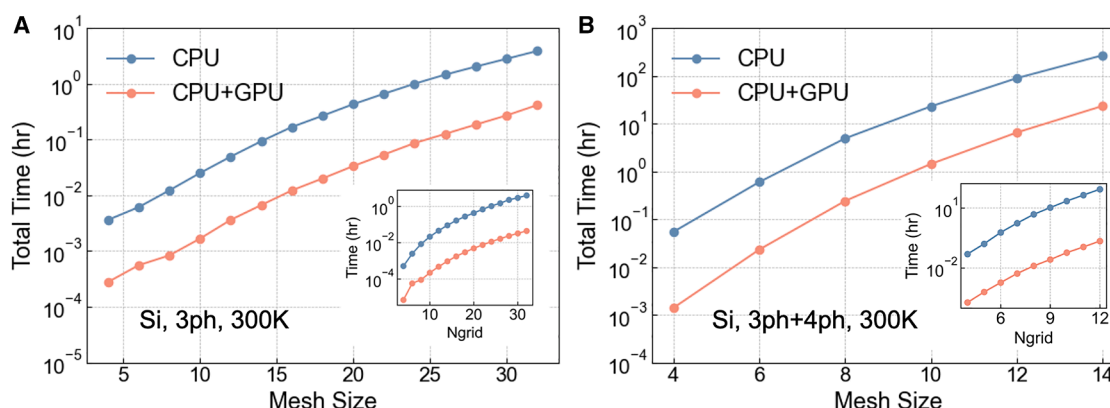


Figure 2. Comparison of total computational cost of Si between CPU-only and CPU-GPU hybrid implementations across different q-mesh sizes

(A and B) (A) 3ph scattering and (B) 3ph + 4ph scattering. The insets show the isolated computational cost of the 3ph and 4ph scattering step alone.

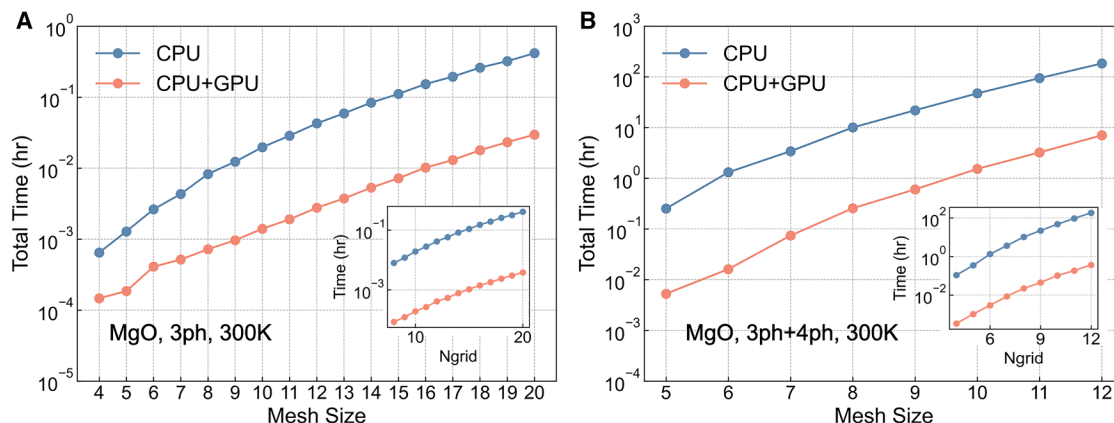


Figure 3. Comparison of total computational cost of MgO between CPU-only and CPU-GPU hybrid implementations across different q-mesh sizes

(A and B) (A) 3ph scattering and (B) 3ph + 4ph scattering. The insets show the isolated computational cost of the 3ph and 4ph scattering step alone.

framework to support full iterative BTE solvers is a high priority. Additionally, combining GPU acceleration with statistical sampling methods¹⁹ would further reduce computation time and memory footprint. Furthermore, overlapping CPU-side process enumeration and GPU-side execution asynchronously could reduce overall wall-clock times. Moreover, while the current framework is designed for single-GPU execution, implementing multi-GPU support would enable handling larger material systems by distributing the computational load and mitigating the memory bottleneck on a single device. A potential implementation path involves using Message Passing Interface (MPI) to distribute scattering processes across multiple GPUs. Lastly, while our implementation focuses on double-precision arithmetic for accuracy, many modern GPUs, particularly those optimized for machine-learning workloads, offer significantly higher single-precision throughput. Exploring mixed-precision

arithmetic could yield additional performance gains without compromising physical accuracy.

In summary, we developed `FourPhonon_GPU`, a heterogeneous CPU-GPU solver for 3ph and 4ph scattering calculations. By offloading computationally intensive rates calculations to the GPU while keeping branch-heavy process filtering on the CPU, we achieve overall speedups of up to 26 $\times$ acceleration under RTA. Comprehensive benchmarks demonstrate the effectiveness of this framework and underscore the importance of aligning algorithm design with hardware capabilities. While the current work focuses on calculations within the RTA to establish the efficiency of the heterogeneous scheme, future work will aim to support iterative BTE solvers and explore mixed-precision execution to further enhance performance. This framework offers a high-efficiency computational tool for modeling phonon transport and accelerating thermal materials design.

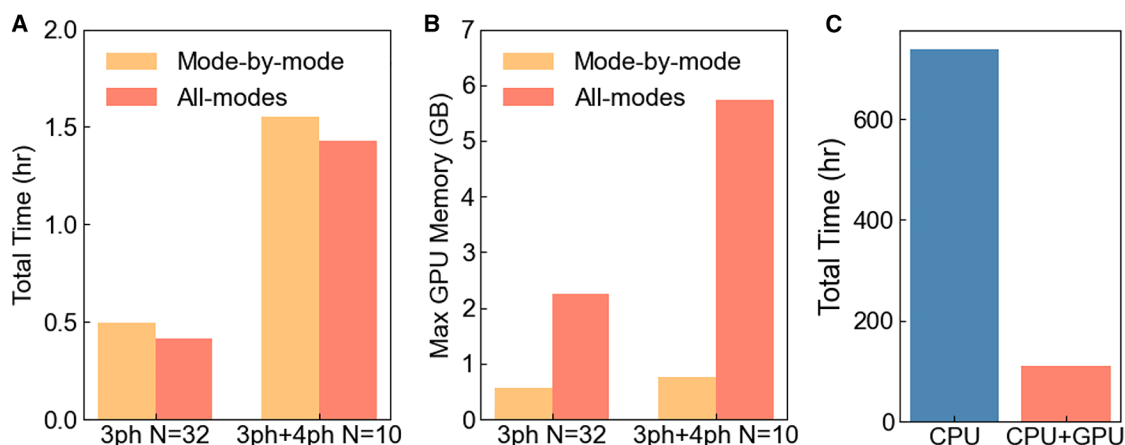


Figure 4. Comparison between all-modes and mode-by-mode parallelization strategies

(A and B) (A) Computational cost and (B) GPU memory usage for 3ph and 3ph + 4ph scattering calculations using a q-mesh of $32 \times 32 \times 32$ and $10 \times 10 \times 10$, respectively.

(C) Computational cost for 3ph + 4ph scattering with a $16 \times 16 \times 16$ q-mesh, comparing CPU-only and CPU-GPU with mode-by-mode parallelization implementations.

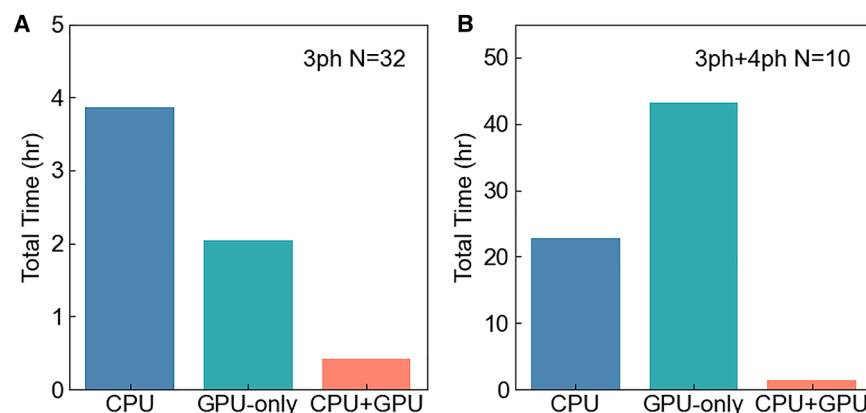


Figure 5. Comparison of computational cost between the CPU-GPU heterogeneous implementation and the GPU-only implementation

(A) 3ph scattering for Si on a $32 \times 32 \times 32$ q-mesh. (B) 3ph + 4ph scattering for Si on a $10 \times 10 \times 10$ q-mesh.

METHODS

Programming model and parallelization strategy

The original ShengBTE⁴⁴ and FourPhonon⁴⁶ codes are written in Fortran and parallelized with MPI for CPU clusters. To minimize code restructuring and maximize cross-platform portability, we use OpenACC, a directive-based programming model for heterogeneous computing. Because individual phonon scattering events are decoupled under the RTA, the calculations lack loop-carried dependencies. This maps naturally to the single-instruction, multiple-threads (SIMT) model of GPUs, where independent scattering channels are evaluated concurrently by lightweight threads.

To illustrate our GPU acceleration strategy, we use the 3ph absorption process as an example, with detailed pseudocode provided in the supplemental methods. In the original CPU implementation (see Algorithm S1), the code loops over all phonon modes, calling a subroutine to evaluate the scattering rate of each mode. This subroutine runs nested loops over all combinations of phonon wavevectors and branches. Energy-conservation conditions filter out forbidden processes before computing the weighted phase space (WP) and scattering rate (Γ). Because of the vast number of combinations, this step is the primary computational bottleneck. We first implement a direct GPU-offloading strategy (Algorithm S2). To minimize runtime host-device communication, we preload all static data onto the GPU, returning only the final scattering rates to host memory. We parallelize over all possible scattering processes and, concurrently, across multiple phonon modes to enhance throughput. This all-modes parallelization approach achieves wider parallel concurrency and higher speedup compared to the mode-by-mode strategy

used in prior GPU implementations^{43,45} (see Algorithm S3), at the expense of larger GPU memory consumption. Both of these methods are implemented and a comparison is provided in the results and discussion section.

GPU execution optimization techniques

Directly applying OpenACC directives to the original CPU code is, however, insufficient to achieve high performance. To optimize GPU execution, we implement four key modifications. First, we flatten nested loops using the `collapse` clause to expose more parallel threads. Second, we reorder loops to enable memory coalescing, ensuring consecutive threads access contiguous memory addresses. Third, we inline calculations for the broadening factor (σ) and matrix elements (V_p) to avoid function call overhead on the GPU. Finally, we accumulate scattering rates using the OpenACC `reduction` clause to prevent race conditions without the overhead of atomic operations.

Although the direct GPU-offloading approach preserves the original CPU workflow, it suffers from performance bottlenecks due to divergent branching. Specifically, conditional checks for energy conservation introduce warp divergence, undermining SIMT execution efficiency. In this scenario, GPU threads assigned to forbidden scattering processes idle while active threads execute, leading to serialization and reduced throughput. This inefficiency stems from the extreme sparsity of the scattering matrix. In our silicon benchmarks using a $16 \times 16 \times 16$ q-mesh, the fraction of allowable scattering processes is only approximately 7% for 3ph and drops to 0.3% for 4ph calculations. Consequently, more than 99% of threads in a warp remain inactive during 4ph rate evaluation (Figure 7, upper panel), consistent with observations in prior work.⁴⁵ Furthermore, symmetry considerations restrict the 4ph iteration space to a triangular domain (Figure 8). Because the OpenACC `collapse` directive requires rectangular iteration spaces, the GPU-only approach cannot exploit these symmetry conditions, further increasing computational overhead.

Heterogeneous CPU-GPU computing scheme

To resolve these limitations, we propose a heterogeneous CPU-GPU computing framework, partitioning the workload between the host and device (Figure 7, lower panel). The CPU enumerates all symmetry-allowed, energy-conserving scattering processes

Table 2. Specifications of selected NVIDIA GPUs used in this work

GPU model	FP32 performance (TFLOPS)	FP64 performance (TFLOPS)	Memory (GB)
A100	19.5	9.7	80
A30	10.3	5.2	24
A10	31.2	0.39	24

TFLOPS, tera floating-point operations per second

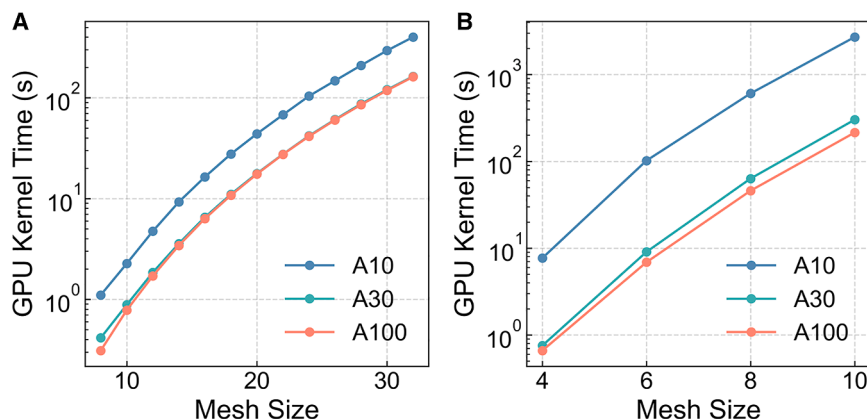


Figure 6. Comparison of GPU kernel execution times across three different NVIDIA GPU architectures

(A) 3ph scattering for Si on a $32 \times 32 \times 32$ q-mesh. (B) 3ph+4ph scattering for Si on a $10 \times 10 \times 10$ q-mesh. Preprocessing on the CPU is excluded.

(Algorithm S4). It then transfers these filtered indices to the GPU, which evaluates the scattering rates in parallel (Algorithm S5). Evaluating only these valid events eliminates warp divergence and maximizes GPU thread occupancy. Although host-side enumeration introduces CPU overhead, it significantly reduces the cost of scattering-rate evaluations and improves overall efficiency. Section results and discussion compares the CPU-only, GPU-only, and heterogeneous schemes. This heterogeneous

approach is the default workflow in our open-source implementation.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Guang Lin (guanglin@purdue.edu).

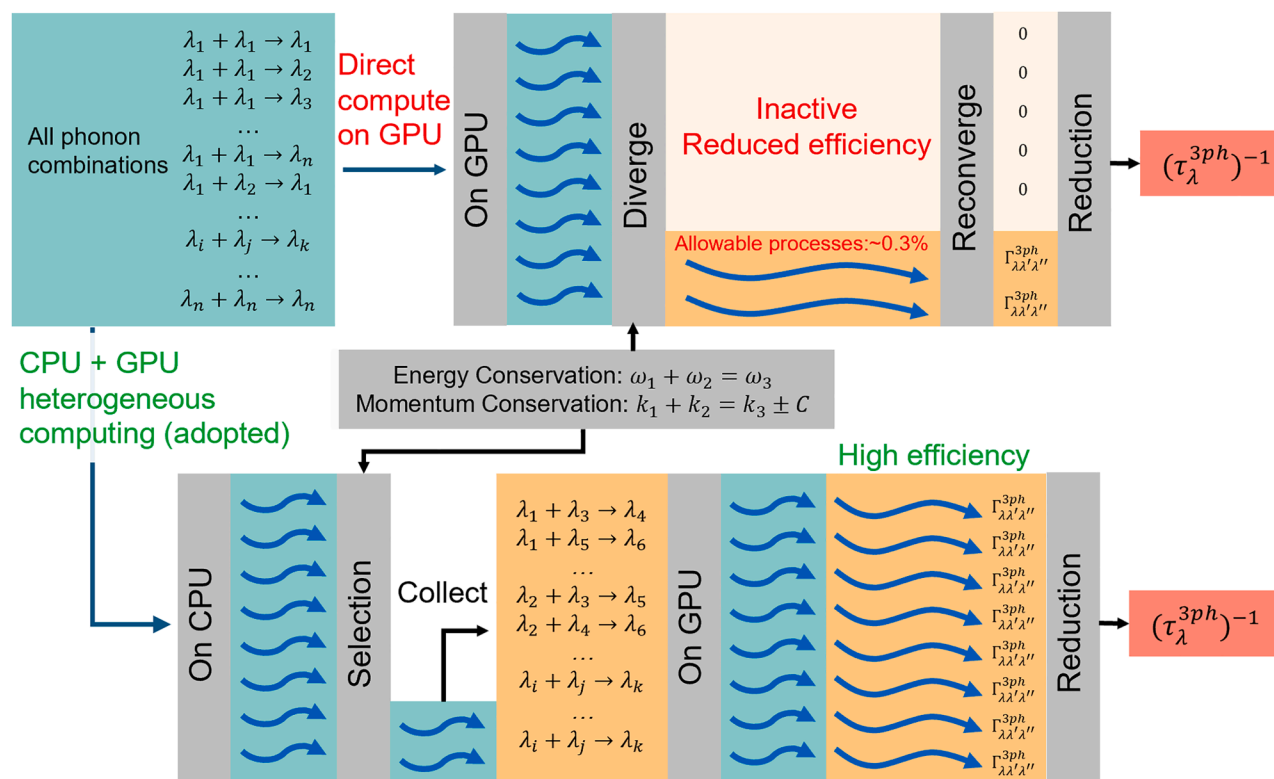


Figure 7. Illustration of thread occupancy and serialization on the GPU due to divergent branching in 3ph and 4ph scattering-rate calculations

For the GPU-only approach (upper panel), the extreme sparsity of allowed scattering processes causes severe warp divergence, where most threads idle while waiting for the few active threads. In our CPU-GPU heterogeneous scheme (lower panel), the CPU filters and packs valid processes, enabling high occupancy and efficient parallel execution on the GPU.

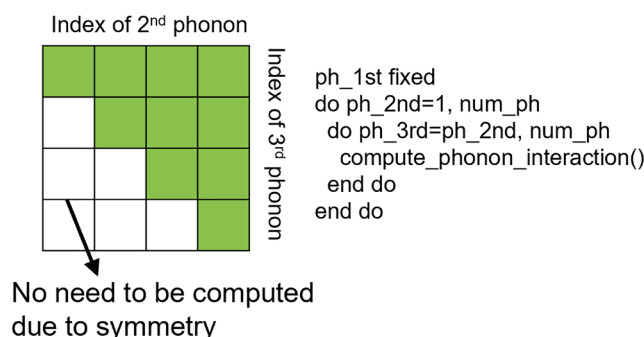


Figure 8. Schematic of the triangular iteration space domain for 4ph scattering under symmetry constraints
The CPU pre-computes indices only for the unique, symmetry-allowed processes, avoiding redundant evaluations on the GPU.

Materials availability

This study did not generate new unique materials.

Data and code availability

- All data reported in this paper will be shared by the [lead contact](#) upon request.
- The open-source code for FourPhonon_GPU is publicly available on GitHub at <https://github.com/FourPhonon/FourPhonon>. The version of the code associated with this paper is archived on Zenodo and is available at <https://doi.org/10.5281/zenodo.21255384>.

ACKNOWLEDGMENTS

Z.G. and X.R. acknowledge partial support from NSF awards 2311848 and 2321301. Z.G. is partly supported by the System Fellows 2024 Doctoral Fellowship provided by the Purdue Systems Collaboratory at Purdue University. Simulations were performed at the Rosen Center for Advanced Computing (RCAC) of Purdue University. G.L. acknowledges the National Science Foundation under grants DMS-2053746, DMS-2134209, ECCS-2328241, CBET-2347401, and OAC-2311848. The US Department of Energy also supports this work through the Office of Science Advanced Scientific Computing Research program (DE-SC0023161) and the Office of Fusion Energy Sciences (DE-SC0024583).

AUTHOR CONTRIBUTIONS

G.L., X.R., and Z.G. conceived the study. Z.G. designed and implemented the software, did the simulations, analyzed the results, and wrote the manuscript. G.L. and X.R. supervised the project. All authors contributed to discussions and revisions of the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at <https://doi.org/10.1016/j.xcrp.2026.103491>.

Received: March 6, 2026

Revised: June 11, 2026

Accepted: July 21, 2026

Published: August 12, 2026

REFERENCES

- Moore, A.L., and Shi, L. (2014). "Emerging challenges and materials for thermal management of electronics". *Mater. Today* 17, 163–174. <https://doi.org/10.1016/j.mattod.2014.04.003>.
- He, S., Ma, Z., Deng, W., Zhang, Z., Guo, Z., Liu, W., and Liu, Z. (2022). "Experimental investigation on the start-up performance of a novel flat loop heat pipe with dual evaporators". *Energy Rep.* 8, 7500–7507. <https://doi.org/10.1016/j.egy.2022.05.248>.
- Dresselhaus, M.S., Chen, G., Tang, M.Y., Yang, R., Lee, H., Wang, D., Ren, Z., Fleurial, J.-P., and Gogna, P. (2007). "New directions for low-dimensional thermoelectric materials". *Advanced materials* 19, 1043–1053.
- Chen, G., Dresselhaus, M.S., Dresselhaus, G., Fleurial, J.-P., and Caillat, T. (2003). "Recent developments in thermoelectric materials". *Int. Mater. Rev.* 48, 45–66. <https://doi.org/10.1179/095066003225010182>.
- Ziman, J. (2001). *Electrons and Phonons: The Theory of Transport Phenomena in Solids* (Oxford University Press).
- Xu, Y., Fan, H., and Zhou, Y. (2023). "Quantifying spectral thermal transport properties in framework of molecular dynamics simulations: a comprehensive review". *Rare Met.* 42, 3914–3944. <https://doi.org/10.1007/s12598-023-02483-x>.
- Peierls, R. (1929). Zur kinetischen theorie der wärmeleitung in kristallen. *Ann. Phys.* 395, 1055–1101. <https://doi.org/10.1002/andp.19293950803>.
- Maradudin, A.A., and Fein, A.E. (1962). Scattering of neutrons by an anharmonic crystal. *Phys. Rev.* 128, 2589–2608. <https://doi.org/10.1103/physrev.128.2589>.
- Broido, D.A., Malorny, M., Birner, G., Mingo, N., and Stewart, D.A. (2007). "Intrinsic lattice thermal conductivity of semiconductors from first principles". *Appl. Phys. Lett.* 91, 231922. <https://doi.org/10.1063/1.2822891>.
- Feng, T., and Ruan, X. (2016). "Quantum mechanical prediction of four-phonon scattering rates and reduced thermal conductivity of solids". *Phys. Rev. B* 93, 045202. <https://doi.org/10.1103/physrevb.93.045202>.
- Feng, T., Lindsay, L., and Ruan, X. (2017). "Four-phonon scattering significantly reduces intrinsic thermal conductivity of solids". *Phys. Rev. B* 96, 161201. <https://doi.org/10.1103/physrevb.96.161201>.
- Kang, J.S., Li, M., Wu, H., Nguyen, H., and Hu, Y. (2018). "Experimental observation of high thermal conductivity in boron arsenide". *Science* 361, 575–578. <https://doi.org/10.1126/science.aat5522>.
- Tian, F., Song, B., Chen, X., Ravichandran, N.K., Lv, Y., Chen, K., Sullivan, S., Kim, J., Zhou, Y., Liu, T.-H., et al. (2018). "Unusual high thermal conductivity in boron arsenide bulk crystals". *Science* 361, 582–585. <https://doi.org/10.1126/science.aat7932>.
- Li, S., Zheng, Q., Lv, Y., Liu, X., Wang, X., Huang, P.Y., Cahill, D.G., and Lv, B. (2018). "High thermal conductivity in cubic boron arsenide crystals". *Science* 361, 579–581. <https://doi.org/10.1126/science.aat8982>.

15. Xu, Y., Fan, H., Li, Z., and Zhou, Y. (2023). "Signatures of anharmonic phonon transport in ultrahigh thermal conductance across atomically sharp metal/semiconductor interface". *Int. J. Heat Mass Tran.* 207, 123628. <https://doi.org/10.1016/j.ijheatmasstransfer.2022.123628>.
16. Xu, Y., and Zhou, Y. (2024). "Metric for quantifying elastic and inelastic thermal transport at interfaces". *Phys. Rev. B* 110, 115305. <https://doi.org/10.1103/physrevb.110.115305>.
17. Guo, Z., Roy Chowdhury, P., Han, Z., Sun, Y., Feng, D., Lin, G., and Ruan, X. (2023). "Fast and accurate machine learning prediction of phonon scattering rates and lattice thermal conductivity". *npj Comput. Mater.* 9, 95. <https://doi.org/10.1038/s41524-023-01020-9>.
18. Srivastava, Y., and Jain, A. (2024). "Accelerating thermal conductivity prediction through machine-learning: Two orders of magnitude reduction in phonon-phonon scattering rates calculation". *Mater. Today Phys.* 41, 101345. <https://doi.org/10.1016/j.mtphys.2024.101345>.
19. Guo, Z., Han, Z., Feng, D., Lin, G., and Ruan, X. (2024). "Sampling-accelerated prediction of phonon scattering rates for converged thermal conductivity and radiative properties". *npj Comput. Mater.* 10, 31. <https://doi.org/10.1038/s41524-024-01215-8>.
20. Guo, Z., Han, Z., Alkandari, A., Khot, K., and Ruan, X. (2024). "First-principles prediction of thermal conductivity of bulk hexagonal boron nitride". *Appl. Phys. Lett.* 124, 163906. <https://doi.org/10.1063/5.0210935>.
21. Zhang, X., Shao, C., and Bao, H. (2024). "Cryogenic thermal transport properties from accelerated first-principles calculations: Role of boundary and isotope scattering". *Phys. Rev. B* 110, 224301. <https://doi.org/10.1103/physrevb.110.224301>.
22. Guo, Z., Sokalski, P., Han, Z., Cheng, Y., Shi, L., Taniguchi, T., Watanabe, K., and Ruan, X. (2024). "First-principles prediction of zone-center optical phonon linewidths and ir spectra of hexagonal boron nitride". *Appl. Phys. Lett.* 125, 231106. <https://doi.org/10.1063/5.0223650>.
23. Tang, Z., Wang, X., He, C., Li, J., Chen, M., Tang, C., and Ouyang, T. (2024). "Effects of thermal expansion and four-phonon interactions on the lattice thermal conductivity of the negative thermal expansion material scf 3". *Phys. Rev. B* 110, 134320. <https://doi.org/10.1103/physrevb.110.134320>.
24. Alkandari, A., Han, Z., Guo, Z., Beechem, T.E., and Ruan, X. (2025). "Anisotropic anharmonicity dictates the thermal conductivity of β -ga 2 o 3". *Phys. Rev. B* 111, 094308. <https://doi.org/10.1103/physrevb.111.094308>.
25. Wei, J., Xia, Z., Xia, Y., and He, J. (2024). "Hierarchy of exchange-correlation functionals in computing lattice thermal conductivities of rocksalt and zinc-blende semiconductors". *Phys. Rev. B* 110, 035205. <https://doi.org/10.1103/physrevb.110.035205>.
26. Guo, Z., Katsamba, I., Carne, D., Feng, D., Moss, K., Barber, E., Fang, Z., Felicelli, A., and Ruan, X. (2025). "Electronic and phononic characteristics of high-performance radiative cooling pigments h-bn: A comparative study to baso4". *Mater. Today Phys.* 54, 101721. <https://doi.org/10.1016/j.mtphys.2025.101721>.
27. Khot, K., Xiao, B., Han, Z., Guo, Z., Xiong, Z., and Ruan, X. (2025). "Phonon local non-equilibrium at al/si interface from machine learning molecular dynamics". *J. Appl. Phys.* 137, 115301. <https://doi.org/10.1063/5.0243641>.
28. Felicelli, A., Katsamba, I., Barrios, F., Zhang, Y., Guo, Z., Peoples, J., Chiu, G., and Ruan, X. (2022). "Thin layer lightweight and ultrawhite hexagonal boron nitride nanoporous paints for daytime radiative cooling". *Cell Rep. Phys. Sci.* 3, 101058. <https://doi.org/10.1016/j.xcrp.2022.101058>.
29. Carne, D., Peoples, J., Guo, Z., Feng, D., Han, Z., Liu, X., and Ruan, X. (2025). "Fos: A fully integrated open-source program for fast optical spectrum calculations of nanoparticle media". *Comput. Phys. Commun.* 307, 109393. <https://doi.org/10.1016/j.cpc.2024.109393>.
30. Guo, Z., Feng, D., Carne, D., Lin, G., and Ruan, X. (2026). "Machine learning inverse design reveals a double narrow-band absorption approach for effective colored radiative cooling paints". *Nano Lett.* 26, 8359–8367. <https://doi.org/10.1021/acs.nanolett.6c01941>.
31. Owens, J.D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A.E., and Purcell, T.J. (2007). "A survey of general-purpose computation on graphics hardware". *Comput. Graph. Forum* 26, 80–113. <https://doi.org/10.1111/j.1467-8659.2007.01012.x>.
32. Raina, R., Madhavan, A., and Ng, A.Y. (2009). "Large-scale deep unsupervised learning using graphics processors". *Icml* 9, 873–880.
33. Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012). "Imagenet classification with deep convolutional neural networks". *Adv. Neural Inf. Process. Syst.* 25.
34. Carne, D., Guo, Z., and Ruan, X. (2025). "Overcoming the Curse of Dimensionality: Enabling Multi-Layer Photon Transport with Recurrent Neural Network". Preprint at arXiv. <https://doi.org/10.48550/arXiv.2509.22890>.
35. Guo, Z., Carne, D., Khot, K., Feng, D., Lin, G., and Ruan, X. (2025). "A review of artificial intelligence-driven approaches for nanoscale heat conduction and radiation". *J. Comput. Inf. Sci. Eng.* 25, 120805. <https://doi.org/10.1115/1.4070202>.
36. Li, S., Guo, Z., Ioffe, J.B., Hu, Y., Zhen, Y., and Zhou, X. (2021). "Text mining of gene-phenotype associations reveals new phenotypic profiles of autism-associated genes". *Sci. Rep.* 11, 15269. <https://doi.org/10.1038/s41598-021-94742-z>.
37. Qin, F., Zhang, Y., Guo, Z., Park, T.J., Park, H., Kim, C.S., Park, J., Fu, X., No, K., Song, H.W., et al. (2024). "High on/off ratio sio 2-based memristors for neuromorphic computing: understanding the switching mechanisms through theoretical and electrochemical aspects". *Mater. Adv.* 5, 4209–4220. <https://doi.org/10.1039/d3ma01142a>.
38. Stone, J.E., Phillips, J.C., Freddolino, L., Hardy, D.J., Trabuco, L.G., and Schulten, K. (2007). "Accelerating molecular modeling applications with graphics processors". *J. Comput. Chem.* 28, 2618–2640. <https://doi.org/10.1002/jcc.20829>.
39. Kirk, D.B., and Wen-Mei, W.H. (2016). *Programming Massively Parallel Processors: A Hands-On Approach* (Morgan kaufmann).
40. Gonze, X., Jollet, F., Abreu Araujo, F., Adams, D., Amadon, B., Applencourt, T., Audouze, C., Beuken, J.-M., Bieder, J., Bokhanchuk, A., et al. (2016). "Recent developments in the abinit software package". *Comput. Phys. Commun.* 205, 106–131. <https://doi.org/10.1016/j.cpc.2016.04.003>.
41. Hacene, M., Anciaux-Sedrakian, A., Rozanska, X., Klahr, D., Guignon, T., and Fleurat-Lessard, P. (2012). "Accelerating vasp electronic structure calculations using graphic processing units". *J. Comput. Chem.* 33, 2581–2589. <https://doi.org/10.1002/jcc.23096>.
42. Hutchinson, M., and Widom, M. (2012). "Vasp on a gpu: Application to exact-exchange calculations of the stability of elemental boron". *Comput. Phys. Commun.* 183, 1422–1426. <https://doi.org/10.1016/j.cpc.2012.02.017>.
43. Wei, Y., You, X., Yang, H., Luan, Z., and Qian, D. (2020). Towards gpu acceleration of phonon computation with shengbte. In *Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region, HPCAsia2020* (Association for Computing Machinery), pp. 32–42.
44. Li, W., Carrete, J., A Katcho, N., and Mingo, N. (2014). "Shengbte: A solver of the boltzmann transport equation for phonons". *Comput. Phys. Commun.* 185, 1747–1758. <https://doi.org/10.1016/j.cpc.2014.02.015>.
45. Zhang, B., Fan, Z., Zhao, C.Y., and Gu, X. (2021). "Gpu_pbte: an efficient solver for three and four phonon scattering rates on graphics processing units". *J. Phys. Condens. Matter* 33, 495901. <https://doi.org/10.1088/1361-648x/ac268d>.
46. Han, Z., Yang, X., Li, W., Feng, T., and Ruan, X. (2022). "Fourphonon: An extension module to shengbte for computing four-phonon scattering rates and thermal conductivity". *Comput. Phys. Commun.* 270, 108179. <https://doi.org/10.1016/j.cpc.2021.108179>.