Scalable Parallelization for the Solution of Phonon Boltzmann Transport Equation

Han D. Tran
University of Utah
Salt Lake City, Utah, USA
hantran@cs.utah.edu

Siddharth Saurav
The Ohio State University
Columbus, Ohio, USA
siddharthsaurav.1@buckeyemail.osu.edu

P. Sadayappan
University of Utah
Salt Lake City, Utah, USA
saday@cs.utah.edu

Sandip Mazumder
The Ohio State University
Columbus, Ohio, USA
mazumder.2@osu.edu

Hari Sundar
University of Utah
Salt Lake City, Utah, USA
hari@cs.utah.edu

Abstract

The Boltzmann Transport Equation (BTE) for phonons is often used to predict thermal transport at submicron scales in semiconductors. The BTE is a seven-dimensional nonlinear integro-differential equation, resulting in difficulty in its solution even after linearization under the single relaxation time approximation. Furthermore, parallelization and load balancing are challenging, given the high dimensionality and variability of the linear systems' conditioning. This work presents a 'synthetic' scalable parallelization method for solving the BTE on large-scale systems. The method includes cell-based parallelization, combined band+cell-based parallelization, and batching technique. The essential computational ingredient of cell-based parallelization is a sparse matrix-vector product (SpMV) that can be integrated with an existing linear algebra library like PETSc. The combined approach enhances the cell-based method by further parallelizing the band dimension to take advantage of low inter-band communication costs. For the batched approach, we developed a batched SpMV that enables multiple linear systems to be solved simultaneously, merging many MPI messages to reduce communication costs, thus maintaining scalability when the grain size becomes very small. We present numerical experiments to demonstrate our method's excellent speedups and scalability up to 16384 cores for a problem with 12.6 billion unknowns.

Keywords: parallel computing, large scale, Boltzmann Transportation Equation, phonons, BTE, heat conduction

ICS '23, June 21–23, 2023, Orlando, FL, USA

ACM ISBN 979-8-4007-0056-9/23/06.
https://doi.org/10.1145/3577193.3593723

ACM Reference Format:
Han D. Tran, Siddharth Saurav, P. Sadayappan, Sandip Mazumder, and Hari Sundar. 2023. Scalable Parallelization for the Solution of Phonon Boltzmann Transport Equation. In *2023 International Conference on Supercomputing (ICS '23), June 21–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3577193.3593723

1 Introduction

The study of thermal transport is critical to determining integrated circuits' performance, cost, and reliability, as overheating is one of the most common causes of their failure. Modeling of the underlying mechanisms of thermal transport is essential in developing heat removal strategies. Modern semiconductor devices range from a few tens of nanometers to a few hundreds of nanometers. The mean free path of energy carrying phonons in silicon at room temperature is approximately 300 nm [33] which is comparable to the length scale of the device. Heat conduction in such cases cannot be accurately determined by continuum equations like the Fourier law of heat conduction and warrants the use of the Boltzmann Transport Equation (BTE) due to its validity for non-equilibrium heat conduction over an extensive range of length scales.

The BTE is a seven-dimensional nonlinear integro-differential equation: three spatial coordinates, three wavevector coordinates, and time. Its solution is quite challenging even after linearizing under the single relaxation time approximation. The Monte Carlo method has been used to solve the BTE as it is amenable to including complex physics such as dispersion, polarization, and boundary scattering, as demonstrated in [21]. However, it is prohibitively expensive for practical applications. The unsteady solution of the gray and non-gray BTE can be obtained using deterministic discretization-based methods [5, 22]. The challenges associated with the solution are both from a memory and a computational time standpoint and remain an active research area [20].

Another approach for solving the linear BTE is using iterative solvers within the discrete-ordinates transport sweep method [1]. This method solves the transport equation in

the spatial domain (for a particular direction and energy group) by sweeping over the cells, starting from the cells adjacent to boundaries as incident fluxes. Various parallelization techniques for sweep transport have been developed [6, 10, 11, 30, 35]. Optimized algorithms for parallelization in semi-structured grids in Cartesian domains have been recently developed in [3] (see also [2] for regular grids and [4] for non-contiguous partitions). The Discrete Ordinates Method (DOM), and its more accurate variant, the Finite Angle Method (FAM) [24] has been used extensively for solution of the BTE for photons, or the so-called Radiation Transport Equation (RTE), including its parallelization [13]. Photons travel at the speed of light. Thus, for terrestrial applications, photon transport may be assumed to be instantaneous and it is not necessary to solve the transient RTE. In contrast, phonons travel at the speed of sound and time-dependent solution of the phonon BTE is often warranted in practical applications to understand how fast heat dissipates. This poses an additional challenge since computation of a time-accurate solution is far more expensive than a steady-state solution. Furthermore, the dependency of phonon velocity on frequency (known as dispersion) makes the problem more complex.

Challenges in parallelization and scalability: While the BTE might look like yet another PDE system on the surface, it presents unique challenges towards parallelization. Firstly, the problem size grows rapidly due to the seven-dimensional system, even for modest resolutions. Specifically, based on previous experience [5, 16, 20, 23], a practical device would require $\sim 10^6$ cells in spatial discretization, 400 directions (20 azimuthal and 20 polar angles), and 40 bands in the spectral space (15 bands with two polarizations and 25 bands with one polarization) to adequately resolve the length scales and obtain a spatial and angular grid-independent solution. This typical discretization results in 22k coupled PDEs in space and time. For the nonlinear solution of the temperature within a time step, it often takes 10 - 20 iterations (depending on the time step size) to attain 3 - 4 orders of convergence. Thus, a single time step would require solving ~440k linear systems (each has 10^6 unknowns), and thousands of time steps are generally needed to reach a steady state or enable the extraction of time-dependent variables. Therefore, we are bottle-necked by strong scalability, i.e., reducing the time-to-solution. For this reason, most works are limited to 2D geometries [29, 32, 36] and are limited to under 128 processes. Secondly, choosing which dimension to partition across the processes is not straightforward, as there are varying data interactions in different stages of the overall solution. The most common approach is partitioning across the wavevector coordinates [5, 22, 29, 32], as these dimensions do not have any adaptivity, and the partition is straightforward. This technique also results in process-local linearized systems, thereby avoiding the need for a distributed memory linear solver. Within these, purely band-based parallelization is the easiest to implement, requiring only a global reduction for synchronization. However, band-based parallelism limits the maximum level of parallelism to the total number of bands, typically 40 - 80 bands [29]. The level of available parallelism can be increased by employing direction-based partitioning in addition to band-based partitioning. However, the directional BTEs can be coupled at the boundaries (e.g., adiabatic boundary conditions, specular surfaces). This would require communication and synchronization across directions (data that will now reside on different processes). Such coupling could be across all directions, resulting in global `MPI_Alltoall` communication. At large scales, such global communication can be challenging to scale. Additionally, although the resulting linear systems are process-local, they require different iterations to converge, leading to a *load imbalance* across the processes. Ali *et al.* [5] pursued band and direction-based parallelization strategies in 3D domains, using up to 2000 processors, but their parallel efficiency deteriorated sharply beyond ~400 processors. As this is the most extensive phonon BTE code we are aware of, all comparisons in this work are against the band-based and direction-based parallelism developed in [5]. In this work, we partition across the spatial dimension, resulting in far greater levels of parallelism than is possible with band-based partitioning, as typical geometries have $\sim 10^6$ cells. Such an approach requires a distributed-memory linear solver and results in communication and synchronization for every iteration of the linear solution. However, this can be achieved using purely local point-to-point communications. This approach also requires us to solve many linear systems in parallel, leading to many messages being exchanged (albeit local). The main advantage is that all directional and band information is process-local and can be readily updated without additional communication. Also, we will not consider parallelism in time in this work. Still, it is essential to note that long-time evolution and the need to keep the overall time-to-solution practical usually force us to operate at high levels of parallelism and push the limits on strong scaling. This work primarily focuses on spatial partitioning for BTE problems and presents algorithms and strategies for ensuring excellent strong and weak scalability.

The main **contributions** of this work are:

- A new cell-based parallel phonon BTE solver (with overlapped and local communications) is developed, demonstrating excellent strong and weak scalability, and significant maximum capability speedups compared to the existing band and direction-based approaches. This is presented in section 3.1.
- A new approach combines band and cell-based parallelism to improve the performance given a fixed number of cores. This approach, presented in section 3.2, gives us greater flexibility in scheduling jobs across an arbitrary configuration of nodes/cores.

- A new batched cell-based approach (with grouping multiple directions) maintains scalability even when the partitions become very small. Batching does not partition along the directions or bands but coalesces communication to reduce overheads. This is presented in section 3.3.
- We demonstrate maximum capability speedups of 106.2x and 41.3x compared to the existing band and direction-based approaches (see Fig.11 for a realistic problem predicting the temperature transport in semiconductors).

We remark that while the primary purpose of this work pertains to the BTE solution, our development and implementation are potentially and equally applicable to other high-dimensional PDE systems. Also, our approach is applicable to various applications, including radiative transport, thermal and neutron radiation common in nuclear engineering and astrophysics. The paper is organized as follows. The next section summarizes the mathematical model and the discretized phonon BTE equations. We describe the critical components of our methods to parallelize the phonon BTE solution in Section 3. Section 4 presents the numerical experiments demonstrating our developed methods' scalability and the comparisons with existing methods. Concluding remarks are made in section 5.

2 Summary of the mathematical model

In modeling phonon transportation, the Boltzmann Transport Equation (BTE) for the phonon intensity, I, is written as [19, 33]

$$\frac{\partial I}{\partial t} + |\boldsymbol{v}_g|\nabla \cdot (I\hat{\mathbf{s}}) = \frac{1}{\tau}\left(I_0 - I\right), \tag{1}$$

where t denotes time, $\boldsymbol{v}_g$ is the group velocity, $\hat{\mathbf{s}}$ is the direction vector, τ is the overall scattering time-scale of the phonon due to all scattering processes in combination, and I_0 is the equilibrium spectral phonon intensity. The phonon intensity is a function of seven variables, i.e., $I = I(t, \boldsymbol{r}, \hat{\mathbf{s}}, \omega)$, where $\boldsymbol{r}$ is the position vector with three components in the Cartesian coordinate system, ω denotes the angular frequency, and the direction vector $\hat{\mathbf{s}}$ has two components in the polar coordinate system, namely the polar angle θ and the azimuthal angle ϕ.

Equation (1) is often solved numerically using the finite angle method (FAM) [8, 9, 24, 25, 31] and standard finite-volume discretization. In this method, the frequency domain is discretized into a finite number of spectral bins N_{band}, and the solid angle of 4π is discretized into a finite number of directions N_{dir}. Then, equation (1) is specified for each spectral bin $b = 1, ..., N_{band}$ and direction $d = 1, ..., N_{dir}$ as

$$\frac{\partial I_{bd}}{\partial t} + |\boldsymbol{v}_g|_b \nabla \cdot (I_{bd}\hat{\mathbf{s}}_d) = \frac{1}{\tau_b}\left(I_{b0} - I_{bd}\right). \tag{2}$$

Equation (2) represents a set of $N_{band} \times N_{dir}$ PDEs for the phonon intensity I_{bd}, which is a function of time and spatial position, i.e., $I_{bd} = I_{bd}(t, \boldsymbol{r})$. Applying the standard finite-volume scheme in space, and integrating equation (2) over the control volume V_C and the control angle Ω_d (see details in [20]), we have

$$\frac{\partial I_{bdC}}{\partial t} V_C \Omega_d + |\boldsymbol{v}_g|_{bC} \sum_f I_{bdf}\left(\mathbf{S}_d \cdot \hat{\boldsymbol{n}}_f\right) A_f = \frac{1}{\tau_{bC}}\left(I_{b0C} - I_{bdC}\right) V_C \Omega_d. \tag{3}$$

In equation (3), the subscript C indicates that the corresponding quantities are evaluated at the cell center (whose volume is V_C), and the subscript f denotes the face f (whose area is A_f and outward unit normal vector is $\hat{\boldsymbol{n}}_f$). The summation is taken over the bounding faces of cell C. The face-center intensity, I_{bdf}, is approximated in terms of the cell-center intensity using the upwind cell's intensity [8, 9, 25–28, 31], i.e.,

$$I_{bdf} = \begin{cases} I_{bdC} & \text{if } \hat{\mathbf{s}}_d \cdot \hat{\boldsymbol{n}}_f > 0 \\ I_{bdN} & \text{if } \hat{\mathbf{s}}_d \cdot \hat{\boldsymbol{n}}_f < 0 \end{cases}, \tag{4}$$

where the sub-index N denotes the neighbor cell sharing face f with cell C. Also, in equation (3), $\boldsymbol{S}_d$ is defined as

$$\boldsymbol{S}_d = \int_{\Delta\Omega_d} \hat{\mathbf{s}}_d d\Omega_d = \int_{\Delta\phi}\int_{\Delta\theta} \hat{\mathbf{s}}_d \sin\theta d\theta d\phi. \tag{5}$$

Finally, using the implicit backward Euler scheme to approximate the time derivative in (3), we end up with a completely discretized equation to be solved for the phonon intensity at the cell center as

$$\frac{I_{bdC}]^{k+1} - I_{bdC}]^k}{\Delta t} V_C \Omega_d + |\boldsymbol{v}_g|_{bC} \sum_f I_{bdf}\Big]^{k+1} \left(\mathbf{S}_d \cdot \hat{\boldsymbol{n}}_f\right) A_f = \frac{1}{\tau_{bC}}\left(I_{b0C}]^{k+1} - I_{bdC}]^{k+1}\right) V_C \Omega_d, \tag{6}$$

in which the notation $]^k$ indicates that the corresponding quantity is evaluated at time index k, and Δt is the time interval such that $t = k\Delta t$. Equation (6) can be written in the matrix form as

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{7}$$

where $\boldsymbol{x} = [I_{bdC_1}, I_{bdC_2}, ..., I_{bdC_n}]^T$ is the vector of cell-center intensity (at time $k+1$) of cell $C_1, C_2, ..., C_n$, respectively, and n is the total number of cells. The coefficient matrix $\boldsymbol{A}$ and right-hand-side (RHS) vector $\boldsymbol{b}$ depend on the intensity at previous time k and other physical quantities. Given the intensity at time k, there are $N_{band} \times N_{dir}$ linear systems (7) to be solved for the intensity at time $k+1$.

Figure 1 schematically describes the overall process of temperature determination in a typical time step. In each outer iteration, there are loops on bands and directions to obtain the intensity (i.e., the solution to (7)) of all bands and directions. An iterative solver, that requires 'inner' iterations, is used for solving these linear systems. The inner iteration

is the central part of our developed cell-based parallelizations (sections 3.1 and 3.3), and the coupled BTE and energy equation parallelization is our development of the combined band+cell-based method (section 3.2).

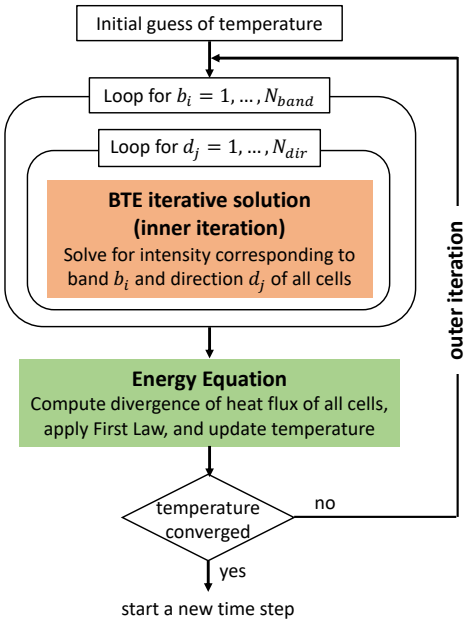


Figure 1. The overall solution process of temperature determination for a typical time step. The BTE solution, i.e., equation (7), contains the dependency of spatial dimension. The energy equation contains the dependency of direction and band dimensions.

3 Methodology

This section presents the main components of the parallelization methods that we developed for the BTE solution, which provide scalability on large-scale systems.

3.1 Cell-based parallelization

The coefficient matrix $\boldsymbol{A}$ in (7) is sparse, i.e., there are maximum n_f of non-zero off-diagonal components per row i, where n_f is the number of cell C_i's bounding faces. The linear system (7) is solved using an iterative solver, in which a sparse matrix-vector product (SpMV) is the core computational kernel. For this reason, we developed a cell-based parallelized SpMV with the critical components described below.

3.1.1 Domain partition. In distributed-memory cell-based parallelization, the computational domain Ω is partitioned into p non-overlapping subdomains, i.e., $\Omega = \{\omega_i\}_{i=1}^{p}$. Each subdomain is discretized into a finite set of cells, i.e., $\omega_i = \{c_m^i\}_{m=1}^{|\omega_i|}$. As shown in equations (4) and (6), the discretized equation corresponding to cell C has the contribution of the cell-center intensities of the neighboring cells (i.e., I_{bdf}). Thus, each subdomain also consists of ghost/hallo cells along the partition boundary. The cell indices can be *local* (i.e., well-defined only for a specific partition and globally ill-defined) or *global* (i.e., well-defined across all partitions). Figure 2 illustrates a simple example of a 2D domain that is partitioned into four subdomains. It is noted that, while Fig. 2 shows a 2D structured mesh for a mere purpose of illustration, our method is applicable to general unstructured meshes of both 2D and 3D domains. The global mesh Ω, e.g., see the left part of Fig. 2, is composed of the following information [1].

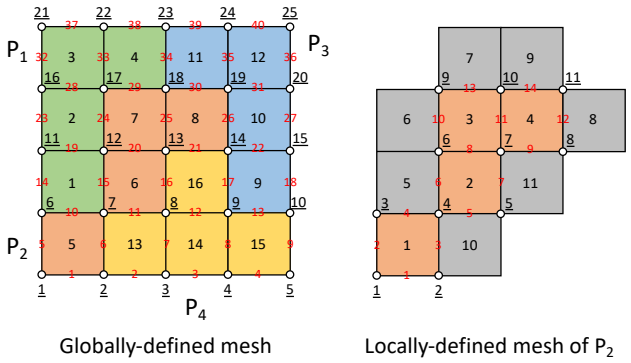


Figure 2. An example to illustrate the globally and locally-defined meshes: the domain Ω is partitioned into four subdomains owned by P_i, $i = \{1, 2, 3, 4\}$, respectively. The color of a cell indicates its owner. Vertex indices are underscored. Face indices are in red. Ghost cells of a locally-defined mesh are in gray.

- Cell data: coordinates of cell center $\boldsymbol{xc}^{gb}$, cell-to-face connectivity $\boldsymbol{lcf}^{gb}$, cell-to-vertex connectivity $\boldsymbol{lcv}^{gb}$, and cell-to-cell connectivity $\boldsymbol{lcc}^{gb}$
- Face data: coordinates of face center $\boldsymbol{xf}^{gb}$, face-to-cell connectivity $\boldsymbol{lfc}^{gb}$, and face-to-vertex connectivity $\boldsymbol{lfv}^{gb}$
- Vertex data: coordinates of vertex $\boldsymbol{xv}^{gb}$

The original global cell indices are first mapped to a sorted array for a more straightforward determination of the correlation between globally-defined and locally-owned quantities. Each process owns a contiguous range of global cell indices, e.g., see the left part of Fig. 2. Next, the local mesh ω_i of the process i is constructed based on the information about the cell's owner. The local mesh defines locally-owned cells, faces, and vertices, e.g., see the right part of Fig. 2. Additionally, ghost cells (i.e., the cells owned by neighboring processes share some faces with ω^i) are specified. Construction of the local mesh requires the following maps to be built in advance.

[1] Superscript 'gb' expresses globally-defined quantities, and the corresponding non-scripted quantities constitute the local mesh that each process owns.

- Local-to-global maps: the maps from owned to global cell index Oc2Gc, local to global face index Lf2Gf, and local to global vertex index Lv2Gv are built using the information about the cell's owner, cell-to-face connectivity, and cell-to-vertex connectivity. Additionally, the total number of locally-owned cells N_{cell}, faces N_{face}, and vertices N_{vertex} are determined.
- Global-to-local maps: the inverse maps from global to owned cell index Gc2Oc, global to local face index Gf2Lf, and global to local vertex index Gv2Lv are straightforwardly constructed using the above local-to-global maps and the locally-owned numbers of cells, faces, and vertices.

Based on the maps built in the above steps, the local mesh is constructed as shown in Algorithm 1. The final step in the pre-processing is to identify ghost cells as described in Algorithm 2. Once the ghost cells are identified, the map from local cell index (i.e., either owned or ghost cell index) to global cell index Lc2Gc is constructed, which is also presented in Algorithm 2. Furthermore, the face-to-cell connectivity $\boldsymbol{lfc}$ and cell-to-cell connectivity $\boldsymbol{lcc}$ are updated to include the local indices of ghost cells.

Algorithm 1 Construction of local mesh

Require: Gc2Oc, Gf2Lf, Gv2Lv, Oc2Gc, Lf2Gf, Lv2Gv, $N_{cell}, N_{face}, N_{vertex}$
Ensure: $\boldsymbol{xc}, \boldsymbol{xf}, \boldsymbol{xv}, \boldsymbol{lcf}, \boldsymbol{lcv}, \boldsymbol{lcc}, \boldsymbol{lfc}, \boldsymbol{lfv}$

for $lc \leftarrow 1$ to N_{cell} **do**
  $\boldsymbol{xc}[lc] \leftarrow \boldsymbol{xc}^{gb}$ [Oc2Gc[lc]]
  $\boldsymbol{lcf}[lc] \leftarrow$ Gf2Lf $\left[\boldsymbol{lcf}^{gb}\right.$ [Oc2Gc[lc]]$\left.\right]$
  $\boldsymbol{lcv}[lc] \leftarrow$ Gv2Lv $\left[\boldsymbol{lcv}^{gb}\right.$ [Oc2Gc[lc]]$\left.\right]$
  $\boldsymbol{lcc}[lc] \leftarrow$ Gc2Oc $\left[\boldsymbol{lcc}^{gb}\right.$ [Oc2Gc[lc]]$\left.\right]$
for $lf \leftarrow 1$ to N_{face} **do**
  $\boldsymbol{xf}[lf] \leftarrow \boldsymbol{xf}^{gb}$ [Lf2Gf[lf]]
  $\boldsymbol{lfc}[lf] \leftarrow$ Gc2Oc $\left[\boldsymbol{lfc}^{gb}\right.$ [Lf2Gf[lf]]$\left.\right]$
  $\boldsymbol{lfv}[lf] \leftarrow$ Gv2Lv $\left[\boldsymbol{lfv}^{gb}\right.$ [Lf2Gf[lf]]$\left.\right]$
for $lv \leftarrow 1$ to N_{vertex} **do**
  $\boldsymbol{xv}[lv] \leftarrow \boldsymbol{xv}^{gb}$ [Lv2Gv[lv]]

3.1.2 Cell-based parallelized SpMV.

In an iterative solution of the linear system (7), the core computational kernel is the SpMV $\boldsymbol{y}^{(m+1)} = \boldsymbol{A}\boldsymbol{y}^{(m)}$, where (m) denotes the m^{th}-iteration before reaching convergence. This operation requires the communication to neighboring processes if a cell locates on the partition boundary. To perform inter-domain communications, we implement two additional data structures [34]:

- Local cell scatter map (LCSM): given partition ω^i, LCSM^i stores the information about the subset of the locally-owned cells that need scattering to neighboring partitions.

Algorithm 2 Determination of ghost cells

Require: $\boldsymbol{lfc}, N_{cell}$
Ensure: N_{ghost}, N_{local}, GhostCells, Lc2Gc

$ng \leftarrow 0$
for $lf \leftarrow 1$ to N_{face} **do**
  if $\boldsymbol{lfc}[lf]$ is not an owned cell **then**
    $ng \leftarrow (ng + 1)$
    GhostCells[ng] $\leftarrow$ global index of $\boldsymbol{lfc}[lf]$
Sort-then-delete-redundants(GhostCells)
$N_{ghost} \leftarrow$ size of GhostCells
$N_{local} \leftarrow N_{cell} + N_{ghost}$
for $lc \leftarrow 1$ to N_{cell} **do**
  Lc2Gc[lc] $\leftarrow$ Oc2Gc[lc]
for $lc \leftarrow 1$ to N_{ghost} **do**
  Lc2Gc[$N_{cell} + lc$] $\leftarrow$ GhostCells[lc]
Update $\boldsymbol{lfc}, \boldsymbol{lcc}$ to include local indices of ghost cells

- Ghost cell gather map (GCGM): given partition ω^i, GCGM^i denotes the ghost cells that need receiving from neighboring partitions.

The above maps are constructed once based on the number of owned cells N_{cell}, the number of ghost cells N_{ghost}, and the local-to-global map Lc2Gc. Algorithm 3 presents the SpMV for solving the linear system (7). In this algorithm, we implemented the overlapping of communication and computation. To perform the overlapping, two sets of cells are defined: the independent cells $I(\omega_i)$ whose neighbors are not ghost cells, and the dependent cells $D(\omega_i)$ whose neighbors contain at least one ghost cell.

Algorithm 3 Cell-based parallelized SpMV; process i

Require: LCSM, GCGM, $\boldsymbol{lcc}$, partitioned $\boldsymbol{x}, \boldsymbol{y}$ vectors, partitioned local matrix $\boldsymbol{A}$
Ensure: $\boldsymbol{y} = (\boldsymbol{A}\boldsymbol{x})$

$\boldsymbol{y} \leftarrow \boldsymbol{0}$
local_cell_scatter_begin($\boldsymbol{x}$, LCSM)
ghost_cell_gather_begin($\boldsymbol{x}$, GCGM)
for $c \leftarrow 1$ to $|I(\omega_i)|$ **do** ▷ independent cells
  $y_c \leftarrow A_{cc}x_c$
  for $f \leftarrow 1$ to number of c's faces **do**
    $d \leftarrow \boldsymbol{lcc}[c, f]$ ▷ c's neighbor sharing face f
    $y_c \leftarrow (y_c + A_{cd}x_d)$
local_cell_scatter_end($\boldsymbol{x}$, LCSM)
ghost_cell_gather_end($\boldsymbol{x}$, GCGM)
for $c \leftarrow 1$ to $|D(\omega_i)|$ **do** ▷ dependent cells
  $y_c \leftarrow A_{cc}x_c$
  for $f \leftarrow 1$ to number of c's faces **do**
    $d \leftarrow \boldsymbol{lcc}[c, f]$ ▷ c's neighbor sharing face f
    $y_c \leftarrow y_c + A_{cd}x_d$

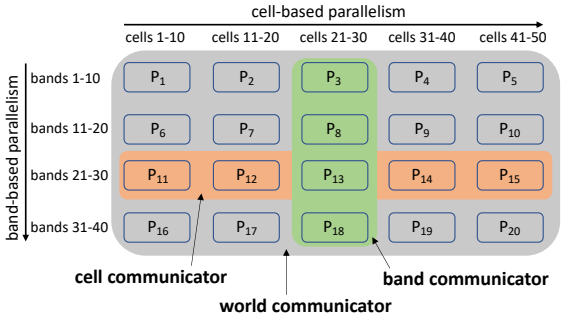
3.2 Combined band+cell-based parallelization

As mentioned, the disadvantage of the band-based approach is the parallelism limitation to the relatively small number of bands used in the BTE computations (e.g., 80 bands [29]) and the load imbalance due to the different convergence rates of the intensity solution across the bands. However, the prominent advantage of the band-based method is the minimum amount of communication and ease of implementation. For single-frequency scattering time-scale expression (which is commonly used [14, 15]), only the reduction of heat flux is needed [5]. Inter-band communication exists for dual-frequency time-scale expression, but it is inconsiderable [29]. For this reason, we developed a combined band+cell-based parallelization to benefit from the minimum inter-band communication cost. Figure 3 schematically illustrates the topology and algorithm of this combined approach. In the 2D Cartesian topology, one dimension is for the band-based parallelism, and the other is for the cell-based parallelism. Heat-flux reductions occur within the band communicators, and ghost exchanges in the SpMV operations (of the inner iterations) are carried out within the cell communicators. The figure uses three background colors (green, orange, gray) to represent the band, cell, and world communicators, respectively.

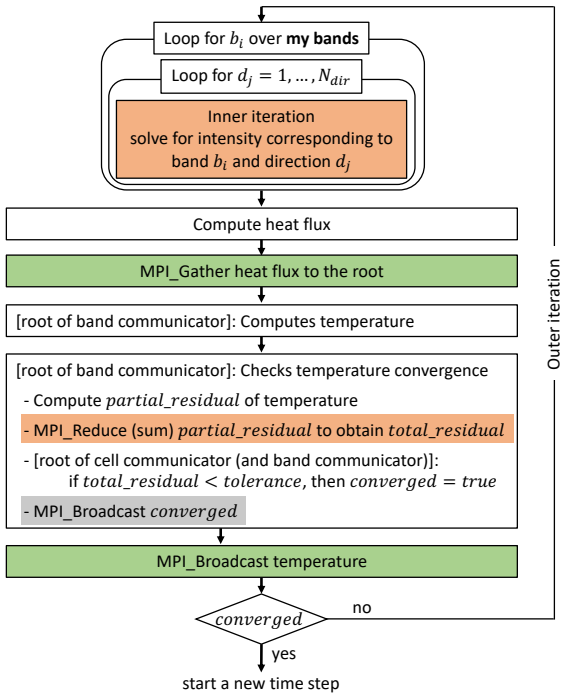
3.3 Batched cell-based parallelization

As shown in Fig. 1, $N_{band} \times N_{dir}$ linear systems (7) are solved within an outer iteration. As the speedup (in strong scalability) reached the plateau, combining multiple linear systems into one 'batched system' coalesces many small MPI messages to reduce the total number of messages that are exchanged. Although the total data transfer in solving the batched system is the same as in solving the original systems separately, we send and receive fewer messages, thus reducing communication overhead. Figure 4 illustrates an example of two independent linear systems that are combined into a batched system. As the global domain Ω is partitioned into p subdomains ω_i, the number of 'extended-owned' cells of the process i in the batched system is $b_c \times |\omega_i|$, where b_c denotes the batch size (i.e., the independent system count). For the batched system formation, the following maps are built to transfer a cell index from an individual system to the batched system.

- `LcSid2BLc`: map from the local cell index of an individual system and the system index (i.e., the identifier of independent linear system in the batch) to the local cell index of the batched system
- `OcSid2BOc`: map from the owned cell index of an individual system and the system index to the owned cell index of the batched system
- `OcSid2BGc`: map from the owned cell index of an individual system and the system index to the global cell index of the batched system



(a) An example showing the topology used in the combined method: the discretization composes of 40 bands and 50 cells, parallelized using 20 processes $P_1 \rightarrow P_{20}$, with four partitions for the band dimension and five partitions for the cell dimension. A particular process, e.g., P_{13} shown in the figure, has the heat-flux reduction carried out in the band communicator, and ghost exchanges in the SpMV operations carried out in the cell communicator.



(b) The chart shows the nonlinear solution of temperature using the combined band+cell-based parallelism within a typical time step.

Figure 3. The Cartesian topology and algorithm of the combined band+cell-based parallelization. The different background colors represent the communicators where corresponding communications occur.

The above first two maps are used in the batched SpMV, and the third one is used to compute the batched RHS vector. Next, the batched system's local cell scatter map (`BLCSM`) and ghost cell gather map (`BGCGM`) are built. Precisely, the information about the subsets of local cells and ghost cells that need to communicate with the neighboring partitions

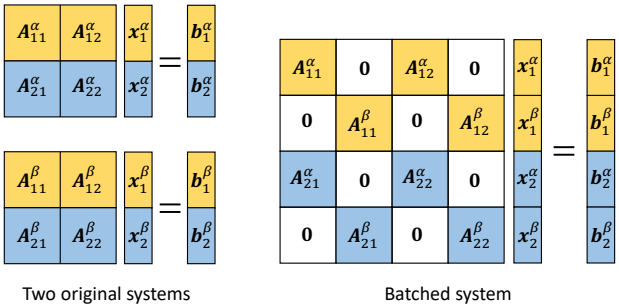


Figure 4. An example to illustrate the batched system: two independent linear systems $\boldsymbol{A}^{\alpha}\boldsymbol{x}^{\alpha} = \boldsymbol{b}^{\alpha}$ and $\boldsymbol{A}^{\beta}\boldsymbol{x}^{\beta} = \boldsymbol{b}^{\beta}$ are grouped into one batched system. The computational domain Ω is partitioned into two subdomains: ω_1 composes of cells with global indices (i.e., row indices of the global matrix) corresponding to the yellow part, and ω_2 composes of cells with global indices corresponding to the blue part.

of the individual system (i.e., LCSM and CGCM) are mapped to the batched system using the maps mentioned above. With all these maps built, the algorithm for the batched SpMV, as shown in Algorithm 4, is implemented to solve the batched system. Similarly to the case of no batching, the overlapping of communication and computation is implemented based on the sets of independent and dependent cells, as mentioned earlier.

The batched system's convergence test is based on the convergence test of the individual systems in the batch. Convergence is detected at iteration k if

$$\|\boldsymbol{r}_k^{(i)}\| < \max\left(RTOL \times \|\boldsymbol{b}^{(i)}\|, ATOL\right) \forall i \in \text{batch}, \quad (8)$$

where $\boldsymbol{r}_k^{(i)} = \boldsymbol{b}^{(i)} - \boldsymbol{A}^{(i)}\boldsymbol{x}_i^{(i)}$ is the residual of the system i at iteration k, RTOL and ATOL are the tolerances for relative and absolute size of the residual norm, respectively.

Algorithm 4 Batched SpMV; process i

Require: BLCSM, BGCGM, LcSid2BLc, OcSid2BOc, $\boldsymbol{lcc}$, partitioned $\boldsymbol{x}, \boldsymbol{y}$ vectors, partitioned local matrix $\boldsymbol{A}$
Ensure: $\boldsymbol{y} = (\boldsymbol{Ax})$

```
y ← 0
local_cell_scatter_begin(x, BLCSM)
ghost_cell_gather_begin(x, BGCGM)
for b ← 1 to b_c do
    for c ← 1 to |I(ω_i)| do                ▷ independent cells
        m ← LcSid2BLc[c, b]
        n ← OcSid2BOc[c, b]
        y_m ← A_nn x_m
        for f ← 1 to number of c's faces do
            d ← lcc[c, f]   ▷ c's neighbor sharing face f
            l ← LcSid2BLc[d, b]
            y_m ← y_m + A_nl x_l
local_cell_scatter_end(x, BLCSM)
ghost_cell_gather_end(x, BGCGM)
for b ← 1 to b_c do
    for c ← 1 to |D(ω_i)| do                ▷ dependent cells
        m ← LcSid2BLc[c, b]
        n ← OcSid2BOc[c, b]
        y_m ← A_nn x_m
        for f ← 1 to number of c's faces do
            d ← lcc[c, f]   ▷ c's neighbor sharing face f
            l ← LcSid2BLc[d, b]
            y_m ← y_m + A_nl x_l
```

4 Results

In this section, we present the numerical experiments demonstrating the scalability of our developed methods and the achieved speedups compared with previously existing methods.

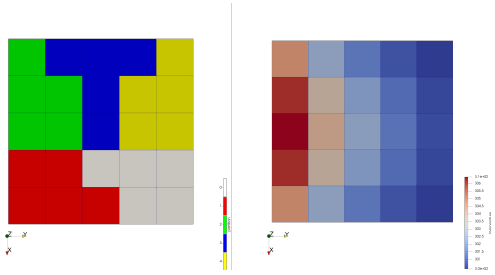
4.1 Numerical experiment setup

The scalability experiments presented in this paper are performed on the Frontera supercomputer. Frontera is an Intel supercomputer at the Texas Advanced Computing Center (TACC) with a total of 8,008 nodes, each consisting of a Xeon Platinum 8280 ("Cascade Lake") processor with 56 cores per node. Each node has 192 GB of memory. The interconnect is based on Mellanox HDR technology with full HDR (200 Gb/s) connectivity between the switches and HDR100 (100 Gb/s) connectivity to the compute nodes. Our SpMV is incorporated with PETSc [7] via the MatShell interface. For the iterative solution of the intensity in all experiments, we used the generalized minimal residual method (GMRES) and the symmetric successive over-relaxation (SOR) preconditioner with a relaxation factor of 1.25. A relative tolerance of 10^{-6} is used for both outer iteration and inner iteration, and an absolute tolerance of 10^{-15} is used for the inner iteration. The baseline for comparison of our results is the code written by [5], which can solve the BTE equations using either band-based or direction-based parallelization. The solver used by [5] is GMRES with the incomplete factorization (ILU) preconditioner. We observed that the convergence rates using SOR and ILU preconditioners for the BTE solution are comparable. For instance, for the problem shown in Fig. 6 used in the scalability experiments, the average number of iterations when solving 22000 linear systems of the first outer iteration is 33.5 for SOR and 36.3 for ILU.

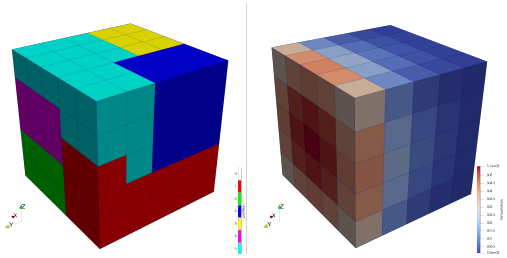
4.2 Correctness verification

We first examine the problems of simple 2D and 3D domains, as shown in Fig. 5, to verify the implementation's correctness. In both problems, one face of the domain is a hot isothermal wall (310 K), while other faces are at ambient temperature (300 K). The spectral space is discretized into 40 bands (15

bands with two polarizations and 25 bands with one polarization). The angular space is discretized into 25 directions (for the 2D problem) and 400 directions (for the 3D problem). In these verification tests, we used a relative tolerance of 10^{-9} for both outer and inner iterations. We tested with five different schemes: sequential, cell-based, combined band+cell-based, batched cell-based, and combined band+batched-cell-based parallelizations. The computed temperature after 10 time steps of every cell is compared with the reference [5]. The relative differences between our computed temperature and the referenced temperature, $|T - T_{ref}|/T_{ref}$ (where T is our computed result, and T_{ref} is the reference result), are less than 10^{-14} at every cell in all five schemes, which is below the tolerance value.



(a) 2D domain of 1 $\mu m \times 1$ μm, discretized into 5×5 cells, 5 partitions for the spatial domain and 4 partitions in the band dimension. A batch size of 5 is used in the tests of batched approach.



(b) 3D domain of 1 $\mu m \times 1$ $\mu m \times 1$ μm, discretized into $5 \times 5 \times 5$ cells, 7 partitions for the spatial domain and 5 partitions in the band dimension. A batch size of 10 is used in the tests of batched approach.

Figure 5. Verification tests for simple 2D and 3D domains: the spatial partition (distinguished by different colors) is on the left, and the temperature after 10 time steps is on the right. Five schemes are tested: sequential, cell-based, combined band+cell-based, batched cell-based, and combined band+batched-cell-based. The computed temperature is compared with the reference [5] by computing the relative error of the temperature, $|T - T_{ref}|/T_{ref}$ (where T is our computed result, and T_{ref} is the reference result). In all five schemes, the relative error at every cell is less than 10^{-14}, which is well below the tolerance.

Secondly, we verify the correctness with a more realistic problem imitating a transistor, as shown in Fig. 6, which requires a much larger scale of computation. The overall dimension of the device is 4 $\mu m \times 4$ $\mu m \times 1$ μm. A heat source generates heat on one surface with a nominal temperature of 150 K. The rectangular cooling channel temperature is set to 100 K. An adiabatic condition (i.e., zero heat flux) is applied to other boundary faces of the device. The discretization composes of 40 bands (two polarizations for the bands 1⟶15 and one polarization for the remaining bands), 400 directions, and 571k cells. For this test, we used 40 spatial partitions as shown in the left part of Fig. 6. A relative tolerance of 10^{-6} for both outer and inner iterations and an initial guess of 100 K for the temperature are used. We tested with four schemes: cell-based (40 cell partitions), combined band+cell-based (5 band partitions, 40 cell partitions), batched cell-based (40 cell partitions, batch of 5), and combined band+batched-cell-based (5 band partitions, 40 cell partitions, batch of 5), and ran the test until the completion of first time, which required 43 outer iterations for the temperature convergence. The right part of Fig. 6 shows the half-section view of the device displaying the temperature distribution in the interior of the device. We computed the relative error of the temperature, $|T - T_{ref}|/T_{ref}$ (where T is our computed result, and T_{ref} is the reference result [5]), at all cells. In all tests, the maximum error is less than 10^{-6}, which is the tolerance used in the experiments.

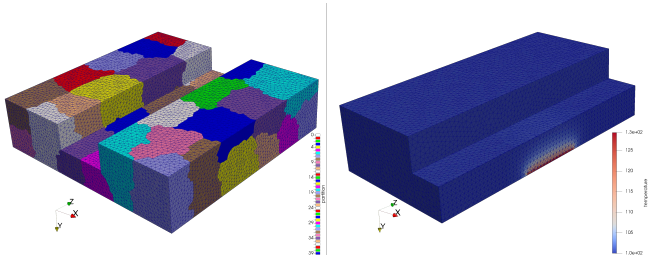


Figure 6. Verification test for a device-like structure discretized into 571k cells: the spatial partition (40 partitions for this test) is on the left, and the half-section view of temperature distributed in the device's interior after the first time step is on the right. Four schemes are tested: cell-based, combined band+cell-based, batched cell-based, and combined band+batched-cell-based. The maximum relative error of temperature compared with the reference [5] in every cell is less than 10^{-6}, which is the tolerance value used for this test.

4.3 Scalability of the cell-based parallelization

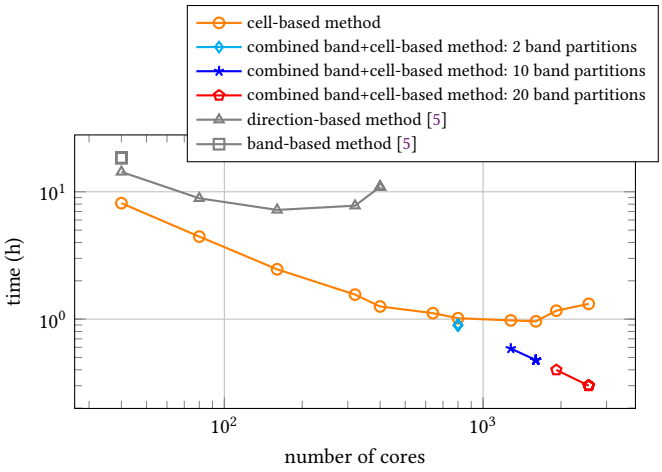
To demonstrate the scalability of our developed methods, we examined the large-scale computation of the device-like structure shown in Fig. 6. There are 12.6 billion unknowns for this problem, which are the intensity at the cell centers of all bands and directions. We used METIS library [17] incorporated in Gmsh [12] to partition the domain.

The overall BTE simulation generally requires thousands of time steps. For illustrating the strong scalability, we measured the time to solution of the first time step, which is sufficient to relatively compare the performance of our developed methods with the existing methods. Figure 7 shows the strong scalability of the cell-based parallelization (orange points) and the comparison with the results using the code of [5] (gray points). With the same number of cores, our cell-based method achieved a speedup of 2.3x compared with the band-based method, and an average speed up of 4.1x compared with the direction-based method. In terms of maximum achievable capability, our cell-based method obtained a speedup of 19.3x compared with the band-based method, and 7.5x compared with the direction-based method. In addition to the time to solution, we computed the corresponding parallel efficiency (normalized by the 40-core case), as shown in Fig. 7b. Given the same number of cores, the cell-based parallel efficiencies are higher than that of the direction-based parallel [5].

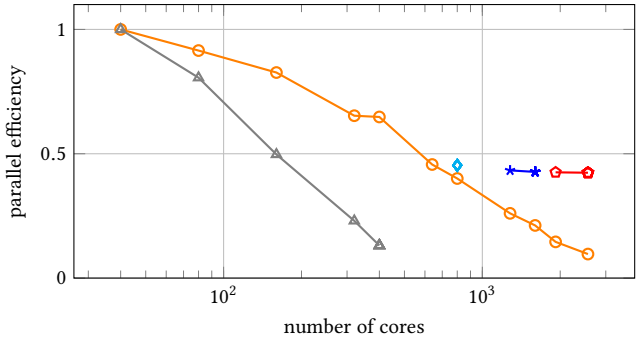
Figure 8 presents the weak scalability experiments for the same problem described above. In these experiments, we measured the time of a single outer iteration of the first time step, which is mainly the time to solve for the intensity of all bands and directions (i.e., the solutions of 22000 linear systems (7) within a single outer iteration) using the cell-based parallel technique. The device is discretized into 1.9k, 3.8k, 8k, 17k, 39k, 169k, 571k, and 929k cells, respectively. We experimented with three grain sizes of 475, 950, and 1900, respectively. It is seen that the larger grain size has better weak scalability than the smaller grain size, which is reasonable due to the larger ratio of computation to communication in the cell-based parallelization method. It should be noted that the increasing computational times in these weak scaling experiments also come from the increasing amount of work when the meshes are refined, e.g., the averaged number of linear iterations (of 22000 linear systems) is 9.6 for 1.9k mesh and 41.1 for 929k mesh.

4.4 Scalability of the combined band+cell-based approach

As mentioned in section 3.2, the combined band+cell-based method benefits from the low cost of inter-band communications. To demonstrate, we re-examined the experiments presented in section 4.3 for the cases closed to and after saturation using the combined band+cell-based approach. Figure 9 shows the computational time of the first time step running with various combinations of band and cell partitions. Compared with the pure cell-based method, the combined method's performance is improved from 1.1x (800 cores) to 3.7x (2560 cores). We also see the beneficial trend of having more and more band partitions in the combined approach only lasts for a while. This is due to the load imbalance resulting from different convergence rates (band to band) of the linear solution of (7). This imbalance is more apparent when



(a) Time to solution of the first time step in the temperature solution.



(b) Parallel efficiency normalized by the 40-core run.

Figure 7. Strong scalability of the cell-based method (section 4.3) and combined band+cell-based method (section 4.4) for the problem shown in Fig. 6. With the same number of cores, the cell-based parallelization achieved speedups of 2.3x and 4.1x compared with the band-based and direction-based methods [5], respectively. In terms of the maximum capability, the cell-based method achieved speedups of 19.3x and 7.5x compared with the methods mentioned above, respectively. For the combined band+cell-based method, heuristic choices of the number of band partitions to obtain the maximum scalability are presented. In terms of maximum capability, the combined approach achieved a speedup of 61.7x compared with the band-based method, and 24.0x compared with the direction-based method [5].

the band-partition grain size becomes smaller [18], i.e., the 'averaging effect' of load imbalance when grouping. Thus, the combined method's performance improves only when the benefit from reducing inter-domain communication is higher than the load-imbalance suffering.

Figure 7 illustrates a heuristic choice for the number of band partitions to obtain the maximum scalability. We present these results of the combined band+cell-based method on the same figure of the cell-based method (discussed in section 4.3)

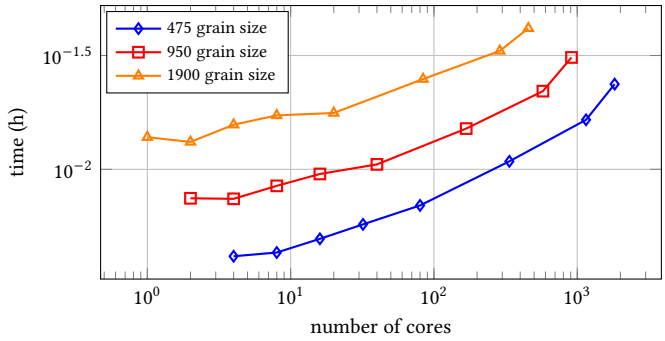


Figure 8. Weak scalability of cell-based parallelization for the problem shown in Fig. 6. The domain discretization composes of 1.9k to 929k cells. Three grain sizes are experimented.

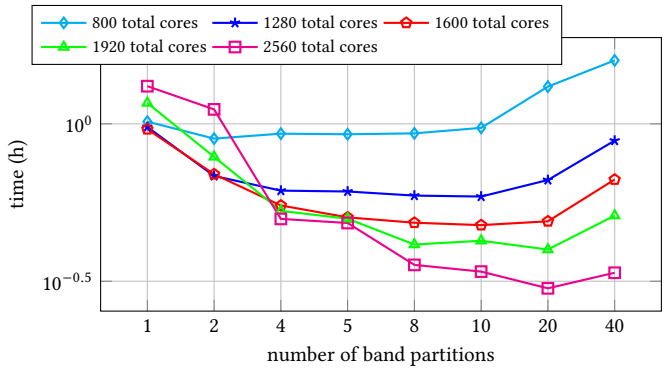


Figure 9. Time to solution of the first time step when solving the problem shown Fig. 6, using various combinations of band and cell partitions in the combined approach. The combined method's performance exhibits an improvement from 1.1x (800 cores) up to 3.7x (2560 cores) compared with the pure cell-based approach.

to illustrate the improvement of the combined method's performance. As shown in Fig. 7a, in terms of the maximum capability, the combined approach achieved a speedup of 61.7x compared with the band-based approach, and 24.0x compared with the direction-based approach. We additionally computed the corresponding parallel efficiency (normalized by the 40-core case), as shown in Fig. 7b, to demonstrate the improvement of parallel efficiency using the combined approach.

4.5 Scalability of the batched cell-based approach

We re-examined the experiments presented in section 4.3 for the cases of 800 cores and more using the batched approach, as shown in Fig. 10. In general, a large batch size improves performance, and larger batch sizes are needed for smaller grain sizes to maintain scalability. However, increasing batch size also adds unnecessary computations to the iterative solution of a batched system. This is because, as shown in equation (8), the iteration of a batched system depends on the slowest one in the batch. The convergence rate of a system is generally different from the others in the batch. The larger the batch size, the more unnecessary computations are generated because there is a higher chance of a more significant difference in convergence rates. The convergence rates of individual systems are not known beforehand. Thus, a strategy for batching systems with similar convergence rates is generally an ad hoc solution.

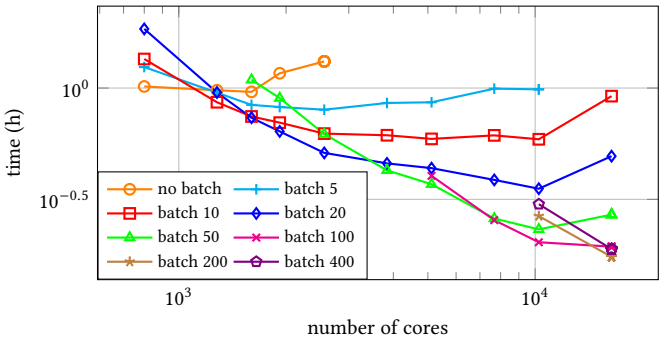
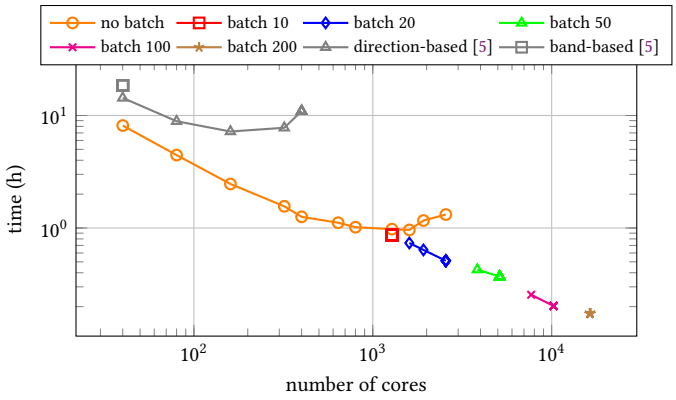


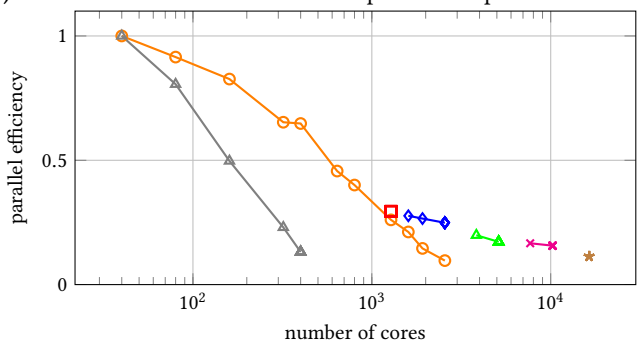
Figure 10. Time to solution of the first time step when solving the problem shown in Fig. 6, using various batch sizes in the cell-based approach. Larger batch sizes are required to maintain scalability when the grain sizes become smaller. The no-batch results are repeated for an easy comparison.

Figure 11 shows a heuristic choice for the batch size to obtain the maximum scalability. It is observed that batch size needs to increase approximately 2-3 times as the grain size decreases by half. As shown in Fig. 11a, in terms of the maximum achievable capability, the batched approach achieved a speedup of 106.2x compared with the band-based approach, and 41.3x compared with the direction-based approach. We also computed the corresponding parallel efficiencies (normalized by the 40-core case), as shown in Fig. 11b, to demonstrate the improvement of parallel efficiency when using the batched approach. These results show that the batched method reduces the communication overhead, particularly when the grain sizes become very small in the cell-based method.

Finally, the combination of band+cell-based with batching can further improve the performance of batched method when it starts losing scalability. Figure 12 illustrates an example using this approach to solve the problem shown in Fig. 6 but with a coarser mesh of 169k cells. The figure shows the time to solution using a heuristic choice for the band



(a) Time to solution of the first time step in the temperature solution.



(b) Parallel efficiency normalized by the 40-core run.

Figure 11. Strong scalability of the batched cell-based parallelization for the problem shown in Fig. 6: heuristic choice of the batch sizes to obtain the maximum scalability. In terms of maximum capability, the batched approach achieved a speedup of 106.2x compared with the band-based method, and 41.3x compared with the direction-based method. The results of no-batch cell-based approach are repeated for an easy comparison.

partitions and batch sizes as discussed above. It is seen that, the batched method lost its scalability at 16384 cores, but not the combined band+batched-cell-based method. At this point, using 8 band partitions and batch size 50, we obtained a speedup of 2.5x compared with the batched method using a batch size 400.

5 Conclusion

This paper presents a synthetic approach, including cell-based, combined band+cell-based, and batched cell-based parallelizations for the phonon BTE solution to simulate heat transfer in semiconductors. The cell-based approach employs spatial domain partition, enabling the method to use many more cores than the existing band-based and direction-based approaches. The core computational kernel of our cell-based method development is the SpMV incorporated with PETSc

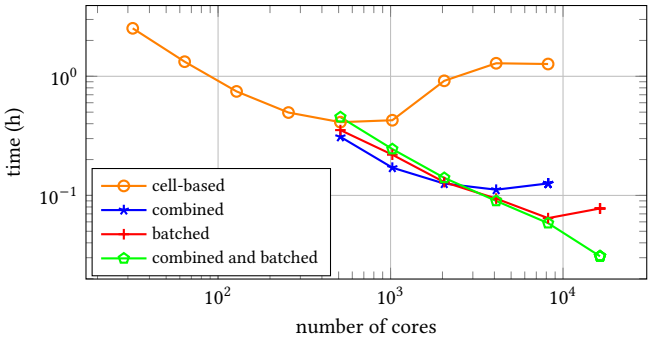


Figure 12. An example illustrates the combined band+batched-cell-based method can maintain scalability after batching reaches its limitation. The figure shows the time to solution of the first time step when solving the problem shown in Fig. 6 using a mesh of 169k cells. The number of band partitions and batch sizes are chosen heuristically to have maximum scalability. At 16384 cores, the combined band+batched-cell-based approach obtained a speedup of 2.5x compared with the batched method.

via the MatShell interface. Additionally, we developed a combined band+cell-based parallelization employing both band-based and cell-based parallelism to improve the performance, given a fixed number of cores. This method benefits from the inconsiderable inter-band communications. We further developed a batched cell-based parallelization that groups multiple linear systems to be solved simultaneously. The batched approach merges many MPI messages exchanged, thus reducing communication overhead and maintaining scalability even when the grain sizes become very small. Our developed methods achieved excellent scalability (up to 16384 cores for the problem with 12.6 billion unknowns) and speedups (106.2x and 41.3x compared with existing band-based and direction-based methods, respectively, using the batched method). Based on the experiments' results, the combined band+batched-cell-based approach is preferred for a significantly fine mesh and unlimited computing resources. However, with limited computing resources (e.g., less than 1000 cores), the cell-based method is recommended for a fine mesh, and the combined band+cell-based approach for coarser meshes.

Acknowledgments

This research was funded in part by the National Science Foundation award numbers 2004236 and 2003747. The computing resources on Frontera were through an allocation by the Texas Advanced Computing Center PHY22019.

References

[1] Marvin L. Adams and Edward W. Larsen. 2002. Fast iterative methods for discrete-ordinates particle transport calculations. *Progress in Nuclear Energy* 40 (2002), 3–159.

[2] Michael P. Adams, Marvin L. Adams, W. Darel Hawkins, Timmie Smith, Lawrence Rauchwerger, Nancy M. Amato, Teresa S. Bailey, and Robert D. Falgout. 2013. Provably optimal parallel transport sweeps on regular grids. *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering* 4 (2013), 19 pages.

[3] Michael P. Adams, Marvin L. Adams, W. Daryl Hawkins, Timmie Smith, Lawrence Rauchwerger, Nancy M. Amato, Teresa S. Bailey, Robert D. Falgout, Adam Kunen, and Peter Brown. 2020. Provably optimal parallel transport sweeps on semi-structured grids. *J. Comput. Phys.* 407 (2020), 109234.

[4] Michael P. Adams, Marvin L. Adams, Carolyn N. McGraw, Andrew T. Till, Teresa S. Bailey, and R.D. Falgout. 2015. Provably optimal parallel transport sweeps with non-contiguous partitions. *Joint International Conference on Mathematics and Computation, Supercomputing in Nuclear Applications and the Monte Carlo Method* (2015), 1–19.

[5] Syed Ashraf Ali, Gautham Kollu, Sandip Mazumder, P. Sadayappan, and Arpit Mittal. 2014. Large-scale parallel computation of the phonon Boltzmann Transport Equation. *International Journal of Thermal Sciences* 86 (2014), 341–351.

[6] Randal S. Baker and Kenneth R. Koch. 1998. An Sn algorithm for the massively parallel CM-200 computer. *Nuclear Science and Engineering* 128 (1998), 312–320.

[7] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Steven Benson, Jed Brown, Peter Brune, Kris Buschelman, Emil M. Constantinescu, Lisandro Dalcin, Alp Dener, Victor Eijkhout, Jacob Faibussowitsch, William D. Gropp, Václav Hapla, Tobin Isaac, Pierre Jolivet, Dmitry Karpeev, Dinesh Kaushik, Matthew G. Knepley, Fande Kong, Scott Kruger, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Lawrence Mitchell, Todd Munson, Jose E. Roman, Karl Rupp, Patrick Sanan, Jason Sarich, Barry F. Smith, Stefano Zampini, Hong Zhang, Hong Zhang, and Junchao Zhang. 2023. PETSc Web page. https://petsc.org/

[8] John C. Chai, HaeOk S. Lee, and Suhas V. Patankar. 1993. Ray effect and false scattering in the discrete ordinates method. *Numerical Heat Transfer, Part B: Fundamentals* 24 (1993), 373–389.

[9] John C. Chai, HaeOk S. Lee, and Suhas V. Patankar. 1994. Finite volume method for radiation heat transfer. *Journal of Thermophysics and Heat Transfer* 8 (1994), 419–425.

[10] Milo R. Dorr and Charles H. Still. 1996. Concurrent source iteration in the solution of three-dimensional, multigroup discrete ordinates neutron transport equations. *Nuclear Science and Engineering* 122 (1996), 287–308.

[11] Thomas M. Evans, Alissa S. Stafford, Rachel N. Slaybaugh, and Kevin T. Clarno. 2010. Denovo: a new three-dimensional parallel discrete ordinates code in scale. *Nuclear Technology* 171 (2010), 171–200.

[12] Christophe Geuzaine and Jean-François Remacle. 2009. Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering* 79, 11 (2009), 1309–1331.

[13] J. Gonçalves and P.J. Coelho. 1997. Parallelization of the discrete ordinates method. *Numerical Heat Transfer – Part B: Fundamentals* 32 (1997), 151–173.

[14] M.G. Holland. 1963. Analysis of lattice thermal conductivity. *Physical Review* 132 (1963), 2461–2471. Issue 6.

[15] M.G. Holland. 1964. Phonon scattering in semiconductors from thermal conductivity studies. *Physical Review* 134, 2A (1964), A471–A480.

[16] Yue Hu, Yongxing Shen, and Hua Bao. 2022. Optimized phonon band discretization scheme for efficiently solving the nongray Boltzmann transport equation. *Journal of Heat Transfer* 144 (2022), 072501.

[17] George Karypis and Vipin Kumar. 1998. *METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices.*

[18] Gautham Kollu. 2014. *Large-scale Parallel Computation of the Phonon Boltzmann Transport Equation.* Master's thesis. The Ohio State University.

[19] A. Majumdar. 1993. Microscale heat conduction in dielectric thin films. *Journal of Heat Transfer* 115 (1993), 7–16.

[20] Sandip Mazumder. 2022. Boltzmann transport equation based modeling of phonon heat conduction: progress and challenges. In *Annual Review of Heat Transfer.* Vol. 24. Begell House, Chapter 3, 71–130.

[21] Sandip Mazumder and Arunava Majumdar. 2001. Monte Carlo study of phonon transport in solid thin films including dispersion and polarization. *Journal of Heat Transfer* 123 (2001), 749–759.

[22] Arpit Mittal and Sandip Mazumder. 2010. Monte Carlo study of phonon heat conduction in silicon thin films including contributions of optical phonons. *Journal of Heat Transfer* 132 (2010), 052402.

[23] Arpit Mittal and Sandip Mazumder. 2011. Hybrid discrete ordinates-spherical harmonics solution to the Boltzmann transport equation for phonons for non-equilibrium heat conduction. *J. Comput. Phys.* 230 (2011), 6977–7001.

[24] Michael F. Modest and Sandip Mazumder. 2021. *Radiative Heat Transfer* (4th ed.). Academic Press.

[25] J. Y. Murthy and S. R. Mathur. 1998. Finite volume method for radiative heat transfer using unstructured mesh. *Journal of Thermophysics and Heat Transfer* 12, 3 (1998), 313–321.

[26] J. Y. Murthy and S. R. Mathur. 2002. Computation of sub-micron thermal transport using an unstructured finite volume method. *Journal of Heat Transfer* 124, 6 (2002), 1176–1181.

[27] S. V. J. Narumanchi. 2003. *Simulation of Heat Transport in Sub-micron Conduction.* Ph. D. Dissertation. Carnegie Mellon University.

[28] Sreekant V. J. Narumanchi, Jayathi Y. Murthy, and Cristina H. Amon. 2004. Submicron heat transport model in silicon accounting for phonon dispersion and polarization. *Journal of Heat Transfer* 126, 6 (2004), 946–955.

[29] Chunjian Ni and Jayathi Y. Murthy. 2009. Parallel computation of the phonon Boltzmann transport equation. *Numerical Heat Transfer, Part B: Fundamentals* 55, 6 (2009), 435–456.

[30] Shawn D Pautz. 2002. An algorithm for parallel Sn sweeps on unstructured meshes. *Nuclear Science and Engineering* 140 (2002), 111–136.

[31] G. D. Raithby and E. H. Chui. 1990. A finite-volume method for predicting a radiant heat transfer in enclosures with participating media. *Journal of Heat Transfer* 112, 2 (1990), 415–423.

[32] S Srinivasan, RS Miller, and E Marotta. 2004. Parallel computation of the Boltzmann transport equation for microscale heat transfer in multilayered thin films. *Numerical Heat Transfer, Part B: Fundamentals* 46, 1 (2004), 31–58.

[33] Chang-Lin Tien, Arunava Majumdar, and Frank M. Gerner. 1998. *Microscale Energy Transport.* Taylor & Francis.

[34] Han D. Tran, Milinda Fernando, Kumar Saurabh, Baskar Ganapathysubramanian, Robert M. Kirby, and Hari Sundar. 2022. A scalable adaptive-matrix SPMV for heterogeneous architectures. In *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS).* 13–24. https://doi.org/10.1109/IPDPS53621.2022.00011

[35] Joseph R. Zerr and Yousry Y. Azmy. 2011. Solution of the within-group multidimensional discrete ordinates transport equations on massively parallel architectures. *Transactions of the American Nuclear Society* 105 (2011), 429–431.

[36] Chuang Zhang, Songze Chen, Zhaoli Guo, and Lei Wu. 2021. A fast synthetic iterative scheme for the stationary phonon Boltzmann transport equation. *International Journal of Heat and Mass Transfer* 174 (2021), 121308.