Special Issue Article

Simulation

Simulation: Transactions of the Society for Modeling and Simulation International
2023, Vol. 99(3) 237–244

DOI: 10.1177/00375497211009611
journals.sagepub.com/home/sim
SAGE SCS

Assessment, improvement, and comparison of different computational tools used for the simulation of heat transport in nanostructures

EA Bea[1,3], A Mancardo Viotti[2,3], MF Carusela[2,3], AG Monastra[2,3] and A Soba[1,3]

Abstract

In this work we compare different implementations of two interatomic potential models, one the empirical Tersoff–Brenner and the other the semi-empirical tight-binding, to be used in the thermal transport study of silicon nanosystems. The calculations are based on molecular dynamics simulations. In the case of Tersoff–Brenner potential, two free software packages were used, while for tight-binding potential, an in-house code was developed. Both approaches require an enormous amount of computing effort, so the use of acceleration tools for adequate performance is crucial. We present a detailed study of each computational tool used: efficiency, advantages and disadvantages, and the results of application to the calculation of thermal conductance of structured silicon nanocrystals subjected to a temperature gradient.

Keywords

Heat transfer, nanostructure, thermal conductance, molecular dynamics, high-performance computing, efficiency analysis

1. Introduction

The study of thermal transport properties of nanostructured low-dimensional systems is a highly relevant topic due to the technological implications.[1,2] The study is usually based on numerical simulations, which makes computational tools a key element. Of these, the simulation molecular dynamics (MD) technique is a natural candidate to be used, where the interatomic potential modeling is a crucial issue and classical or quantum mechanical approaches can be implemented. The last approach to a solid material provides accurate results in calculating its physical properties but, to the date, its full implementation is usually very expensive computationally for systems of more than some hundreds of atoms. On the other hand, empirical and semi-empirical potentials can provide satisfactory results for thermal, mechanical, and electronic properties of materials, especially in bulk. Moreover, these approximations are widely used due to the reasonable computing resource requirements. However, in the case of empirical approximations mainly, the price to pay is less transferability for systems with a large surface-to-volume ratio, requiring a critical analysis when it is applied to small systems.[3,4]

In this work, we study the thermal transport of silicon nanosystems, modeling the atomic interactions using: (1) a classic empirical potential of Tersoff–Brenner (Tersoff)[5,6] over two closed software packages[7,8]; and (2) a semi-empirical tight-binding (TB) potential[9,10] over an in-house software. To do that we recreate a non-equilibrium thermodynamical state by means of Langevin's MD simulation for both potential models.

In both approaches, the manipulation of a reasonable number of atoms (which can range from a few hundred to a few thousand) requires a parallel environment and high computing capabilities. In the case of the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) we use the message passing interface (MPI) and OPENMP environments provided by the software and the graphical processors units version also, while the other used package is based solely on a graphics processing unit (GPU). To solve the TB problem, the developed in-house software deals with the resolution of a simple eigensystem in each iteration. However, it is only necessary to obtain

[1]Centro Atómico Constituyentes - CNEA, Argentina
[2]Instituto de Ciencias, Universidad Nacional de Gral. Sarmiento, Argentina
[3]Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina

Corresponding author:
A Soba, CNEA-CAC. Av. Gral. Paz 1499, San Martin, Buenos Aires cp(1650), Argentina.
Email: soba@cnea.gov.ar

slightly more than a half of the eigenvalues with lower absolute values. For this purpose, we apply different strategies using MPI and OPENMP paradigms, measuring their efficiencies.

There are many strategies for this type of problem faced in multiple ways by different scientific teams. In all cases they have one thing in common: simulation jobs require an enormous amount of computing time. In this work we propose three strategies using different tools that offer advantages and disadvantages due to their inherent use and the amount of computing time required by each of them. In particular, the strategies using pure GPUs produce the best computing efficiency. On the other hand, the package that uses TB provides a more precise calculation from the point of view of the physics of the problem, but its performance is much lower. In this way, we will show the analysis carried out on these tools to show three ways of facing the same problem that obtain results of different precision with different degrees of computational performance.

The paper is organized as follow: the next section will describe the physical model and the two interatomic potentials, Tersoff and TB. In Section 3 we present simulation results of some thermal transport properties for the physical model described. Section 4 is devoted to discuss the advantages and disadvantages of each solver used for the TB potential. In Section 5 we present an efficiency analysis of all the tools used. Finally, we include some general conclusions of our work.

2. Atomic interaction models

The physical system of interest on which both simulation approaches will be applied for studying the heat transport is ultra-thin silicon nanocrystals with a central hole of different sizes. These nanocrystals can be the building blocks to create artificial periodic nanostructures that can be used to design and control phonon bands. These are ideal platforms to study the interplay between structural factors and heat transport, where thermal conductivity can be tuned over two or three orders of magnitude by nanostructuring.[11–14]

We model the silicon nanocrystal by a rectangular prism with faces Si{100} and a volume of $49 \times 49 \times 16 \text{Å}^3$, which is structured by a centered square hole of side a in the thinner direction (see Figure 1). In the direction of the longer side, a temperature gradient is applied (see Figure 2).

The thermal transport simulation is performed by using a MD approach describing atomic interactions with different levels of approximation and, therefore, with different computational performance in the treatment of systems with a large number of atoms in the nanometric scale.[3,4] We consider for the atomic interactions the two following models: (a) a classical Tersoff potential; and (b) an adapted TB potential for MD simulations.

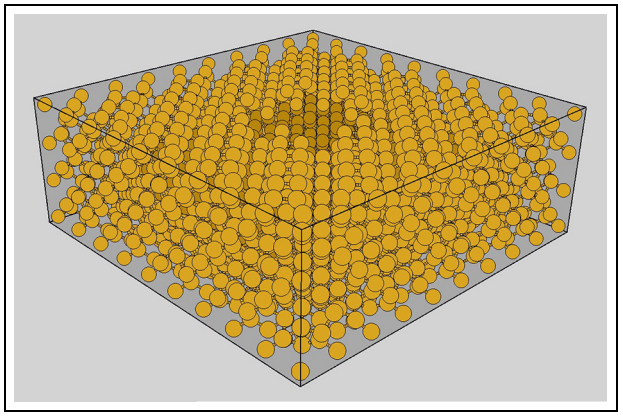

Figure 1. Perspective views of the structural model of an ultra-thin silicon nanocrystal with a square hole centered on the face (001).

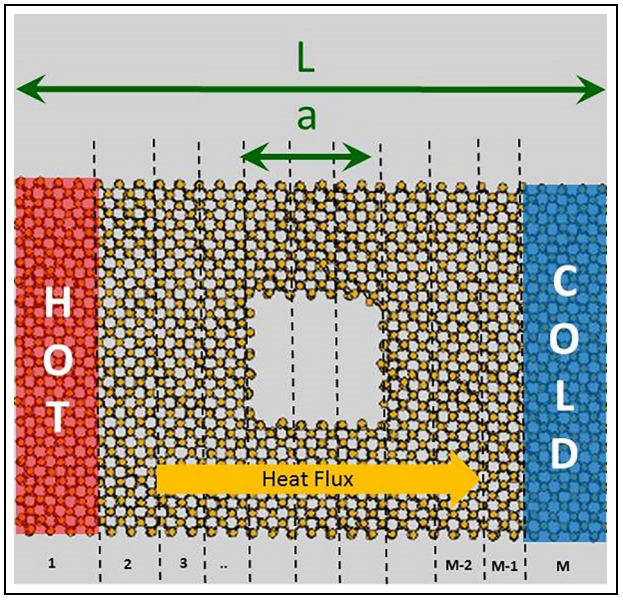

Figure 2. Schematic illustration of the Si nanopatterned setup used in molecular dynamics simulations. A nanoribbon of length L is connected to a hot (H) and cold (C) reservoir at its ends. Heat flows from H to C. It Free boundary conditions are applied in the transverse direction of the heat flux. A square hole of side a equal to N units cells is patterned. N goes from 0 to 5. The sheet is divided in M equally spaced blocks (indicated by vertical dashed lines). Block 1 corresponds to the source of heat, while block M corresponds to the sink.

There are important aspects to consider when choosing potentials for simulations. Accuracy should be considered, on the one hand, to reproduce the properties of interest as best as possible. On the other hand, the transferability allows them to be used to study properties for which they have not been adjusted. Finally, essential aspects when performing numerical simulations are computation time and computational performance.

The Tersoff-type potentials[5,6] take into account the electronic wave functions in an effective way, considering the nuclei of the atoms in fixed positions of the crystalline structure. These potentials can be expanded in terms of many-body interactions, which depend on the relative distances of atoms and the number of and angles between bonds, and produce reasonably accurate potential energy landscapes for materials with covalent and metallic bonds. From the computational point of view, there are many software packages that implement this kind of approximation with a very good performance. For MD simulations with this potential type, two free packages were used: (a) free software LAMMPS[8]; and (b) the Graphics Processing Units Molecular Dynamics (GPUMD) package.[7]

We implement a semi-empirical TB potential adapted to a MD simulation scheme.[9,10] This model uses as expansion basis s and p valence electronic orbitals of Si atoms to describe the electronic fundamental state. Hopping integrals of the Hamiltonian matrix and repulsive ion pair potentials are parameterized by using short-range pair functions, depending on the distance between neighboring ions. Numerically, this approach requires solving the diagonalization of the Hamiltonian matrix for a sufficiently large number of eigenvalues and eigenvectors that allows the expansion of all valence electrons into non-bonding and bonding states. For more details about this potential and also the study of thermal conductance in comparison with the Tersoff approach, see Mancardo Viotti et al.[15] and Bea et al.[16]

3. Thermal transport simulation

Firstly, a thermalization of the system to 300 K is carried out. For modeling the initial atomic structure, atom positions are set like in the ideal Si crystal. Ideal nanocrystals are used to initialize the LAMMPS simulations. Instead for TB and GPUMD simulations, nanocrystals with reconstructed faces are modeled. The first atomic plane on ideal surface is arranged in rows of dimers with (2x1) periodicity (see Figure 1),[17] as described in a previous work.[16] Once the system is thermalized, the structure is coupled in its ends to two thermal reservoirs at different temperatures. Therefore, the evolution is under a non-equilibrium state, establishing a temperature gradient along the system.[3] The coupling to the thermal reservoirs is through three atomic layers in each longitudinal end connected to a Langevin thermal bath at $T_H = 350$ K (hot bath) and $T_C = 250$ K (cold bath), respectively (see Figure 2).

We integrate the equations of motion for a time long enough to guarantee that a stationary state is reached. Once achieved, we compute the energy E injected (absorbed) by (from) the hot (cold) bath. In this way we calculate the heat current J as the rate dE/dt at which energy is injected into (taken out of) the system per unit time. So according to Fourier's law for heat conduction,

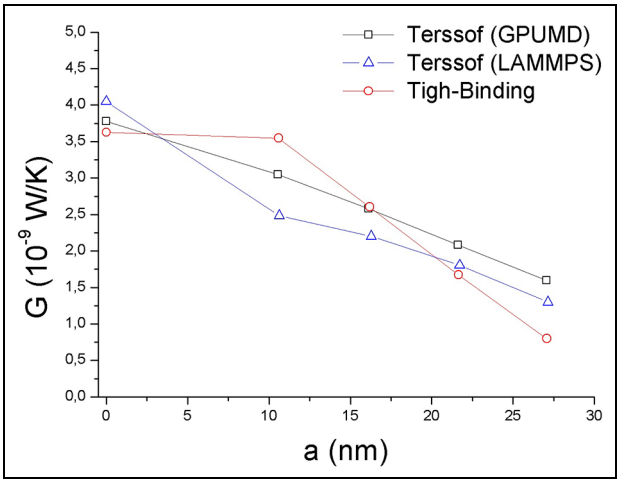

Figure 3. Thermal conductance G versus hole size a, obtained with the three codes used in this work. The hole size range from cases with no hole ($a = 0$) to a 5×5 hole. All curves descend with the increase of a. The data are calculated within a $\pm 5\%$ error interval (error bars are not shown). GPUMD: Graphics Processing Units Molecular Dynamics; LAMMPS: Large-scale Atomic/Molecular Massively Parallel Simulator.

the thermal conductance of the system G can be defined through the following relation:

$$J = G\,\Delta T\,, \tag{1}$$

where $\Delta T = T_H - T_C = 100$ K is the temperature difference along the system. The thermal conductance is a measure of the capability of the system to transfer heat. It is not an intrinsic property of the material but of the system as a whole.

In Figure 3 we show the thermal conductance as a function of hole size as predicted by TB and Tersoff potential models, using the three codes analyzed here. The data simulated by TB and GPUMD codes were extracted from our recently published previous work.[16] For the Tersoff potential model, the conductance displays a monotonic decrease with a. GPUMD and LAMMPS (both version used) implementations display a reasonably good agreement within the error intervals, considering that on the faces of the nanocrystal one simulates an ideal structure while the other is a structure with reconstructed surfaces. For $a > 10$ nm, the TB potential model also presents a monotonic decrease but with a larger rate. Nevertheless, the predictions of the TB model deviates from those of the Tersoff's. This discrepancy is enhanced with the size of the hole, suggesting a significant role of the surface effects, which is not accurately considered by Tersoff models due to their low transferability conditions.

From eigenvalues and eigenvectors of the TB Hamiltonian matrix, the local density of states (LDOS) projected onto a single atom or group of atoms can be

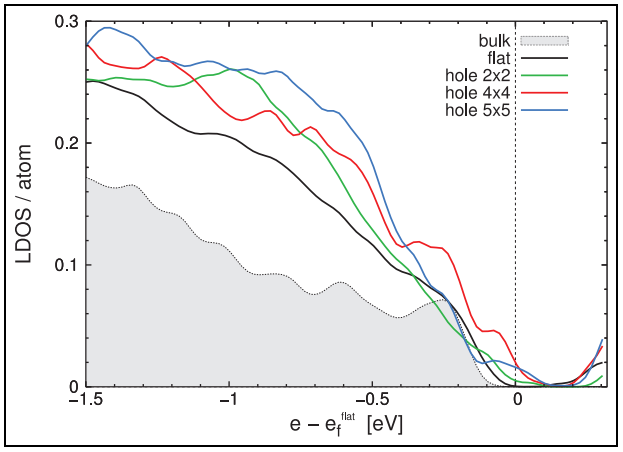

Figure 4. Time-averaged local density of states (LDOS) projected onto a group of atoms contained within a slice of width a, enclosing the hole, for nanocrystals with no hole (flat), a 2×2 hole, a 4×4 hole, and a 5×5 hole. The DOSs were calculated from eigenvalues and eigenvectors of the tight-binding Hamiltonian matrix. The DOS projected onto four bulk atoms of flat nanocrystal is also shown (curve with gray shading). Energies are set to zero by taking as reference the Fermi level of a flat nanocrystal.

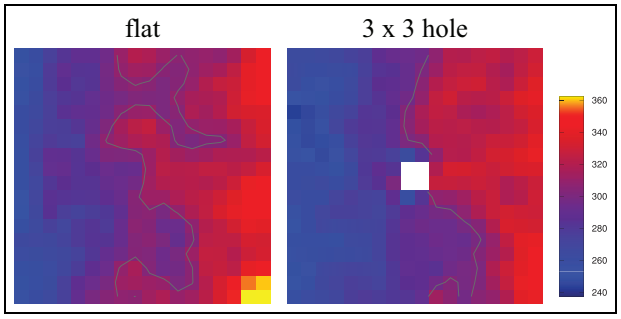

Figure 5. Two-dimensional temperature map averaged in time and space along the z-axis direction (perpendicular to the plane of heat propagation). Two cases corresponding to nanocrystals with no hole (flat) and a 3 × 3 hole are shown. Each plot represents the top view of an $18 \times 18 \times 1$ grid subdividing the structure volume, in which pixel colors indicate the average temperature of atoms within the subvolumes, according to the color scale shown on the right-hand side.

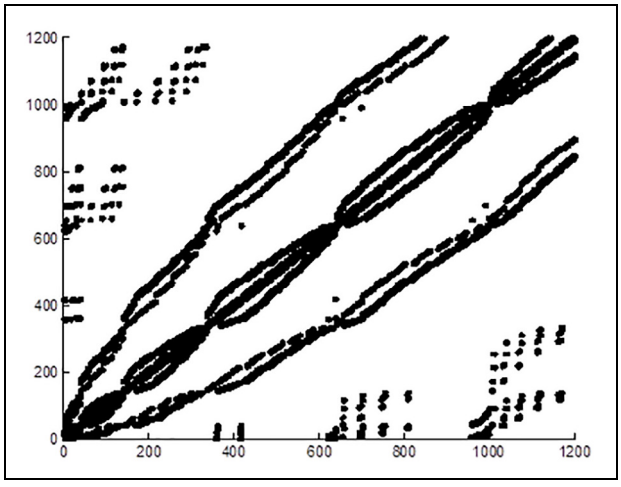

Figure 6. Example of a tight-binding matrix structure showing no nulls distribution.

computed (see Figure 4). So we take advantage of all the electronic information provided by the TB potential for highlighting somehow the surface effect when the surface-to-volume ratio increases. The LDOS is projected onto atoms being within an imaginary slice having the same width a of the hole (hole slice), which covers the entire cross-section through which heat flows. Close to the Fermi level, the DOS of the nanocrystal with a 2×2 hole resembles that of the one with no hole, while for nanocrystals with 4×4 and 5×5 holes, the DOS differs from that of the flat nanocrystal and exhibits a greater number of non-bonding states, which are located on the surface.

The numerical computations also allow one an insight into the thermal energy distribution along the system. We show in Figure 5 a bidimensional temperature map for the flat and for the 3×3 hole structure, calculated with the TB model. Local temperatures are calculated with a time average of the kinetic energy and a spatial average in the out-of-plane direction. This map evidences how the hole affects the local temperature distribution in a non-trivial way. This effect is enhanced as long the surface size hole to surface size nanocrystal ratio increases.

4. Tight-binding performance improvements

In order to solve the problem involved in the potential TB, once the Hamiltonian of the system has been built, it is necessary to obtain from the raised system a sufficient number of eigenvalues to characterize the linked states and the first ones above the zero line. In this way, basically the problem is reduced to obtains a number of eigenvalues and eigenstates of $[K][\Phi] = \lambda[\Phi]$. For systems of N atoms, the dimensions of the matrices will be $4N \times 4N$ and have the generic form shown in Figure 6. The number of required eigenvalues is in the range $[2N + 4N, 2N + 4N^* 0.1]$. There are tools to solve this type of matrix problem when dealing with positive defined matrices[18] or semi-defined ones.[19] If the system does not have these properties, it is possible to use a direct solver such as the one provided by the EisPACK libraries.[20] When the matrices are scattered or their dimensions large enough, an iterative method can be passed that generally provides a few eigenvalues efficiently.[21,22]

Table 1. Comparison of computation times between the different solvers used and for different sizes of matrices. The number of calculated eigenvalues for each size of matrix is indicated.

Solver	$N = 4 \times 300$		$N = 4 \times 1200$		$N = 4 \times 5080$	
LAPACK	1200 eig.	3.29 s	4800 eig.	252.41 s	20,324 eig.	17,264.15 s
ARPACK	600 eig.	13.90 s	2500 eig.	1335.85 s	10,300 eig.	–
JD	620 eig.	639.76 s	2500 eig.	43682.80 s	10,300 eig.	–
ScaLAPACK/pDSYEVD	1200 eig.	2.60 s	4800 eig.	82.17 s	20,324 eig.	8345.76 s
ScaLAPACK/pDSYEVX	640 eig.	4.53 s	2500 eig.	71.78 s	10,200 eig.	2564.49 s

JD: Jacobi–Davidson.

In the cases analyzed in this work the matrices were clearly sparse with a ratio of no nulls of approximately 0.034. Although in this scenario it would seem wise to address the solution mode using solvers that exploit this particularity, we see that this is not efficient. The relatively small size of the matrices involved, together with the large number of required eigenvalues, make the problem an interesting case of analysis. Examples of this type of system that pose a compromise between the size of the matrices and the number of necessary eigenvalues can be found in other calculation packages, such as the one used in SIESTA,[17] where a linear numerical combination of an orbital base is solved over a uniform space grid. In this case, a dispersed matrix system of high dimensions, on which an appreciable number of eigenvalues (usually 1/3) is required, is resolved using the method of Jacobi–Davidson (JD), in this case including deflation, extracting converged eigenvalues from the solution subspace.[17,23]

For our particular case we have tested three lines of solvers. (1) Firstly, we use DSYEVX provided by the LAPACK library,[18] based on the bisection method with inverse iterations that works with dense, symmetrical matrices. The solver provides as a solution all the eigenvalues of the matrix and their corresponding eigenvectors. This solver is fast and efficient for a relatively small number of atoms (< 1000). However, the amount of memory required for large-scale matrices, as well as their dispersion, seemed limiting factors that suggested the possibility of exploring other models. Note that for a number of 5080 atoms, the dimensions of the matrices reach 20,320 $\times$ 20,320, and about 12,200 eigenvalues are required. In this case we expand the problem to a solver parallelized using the ScaLAPACK version (pDSYEVX) of this same solver.[24] We also tested the pDSYEVD solver, based on the divide and conquer algorithm that has faster results on some occasions. On the other hand, and to take advantage of the dispersion of the matrices, we focus on using scattered iterative systems such as those provided by the ARPACK library,[19] which uses a modified Arnoldi system to obtain the requested eigenvalues and with the option to calculate eigenvectors.[25] Along the same lines we use a method based on the JD algorithm, in this case including deflation, extracting converged eigenvalues from the solution subspace. While the efficiency of the dispersed algorithms mentioned, both Arnoldi and JD, is well known, it is also true that their degree of efficiency increases when (a) the matrix is sufficiently large and (b) the required eigenvalues are relatively few, although the mentioned methods allow obtaining portions limiting eigenvalues in well-located regions of the spectrum.

5. Efficiency analysis

5.1. Distributed and shared memory approximation

All our calculations were carried out on two systems. We use the TUPAC supercomputer,[26] a 48 TFLOPS machine with 4352 AMD Opteron cores distributed among 68 computing nodes. Thirty-two Tesla 2090 Nvidia boards can be used for the GPU. Five separate networks support the interconnection of nodes: three separated Ethernet networks for monitoring and administration, and two QDR Infiniband networks with very low latency designed for message interchange during computing. In addition, we use a Server Intel Xeon 2xCPU E5-2620 v3, 2.4 GHz, with x86-64 architecture.

The performances obtained for the three lines of diagonalization solvers yielded dramatically better figures in the LAPACK/ScaLAPACK cases, with computation times as detailed in Table 1. The relative times between both solvers, although comparable in order of magnitude, differ by one factor, especially in cases with the highest number of atoms.

Figure 7 shows a scalability curve obtained using ScaLAPACK (pDSYEVX) for the different sizes of the problem considered. We verified a strong scalability, obtaining better performance in the cases of greater systems. Somehow the optimal times provided by the algorithm come from the division obtained with the minimum value of processors involved. Increasing the number of processors does not seem to compensate for the inherent communications, except in the case of $N = 5081$ where the configuration of 64 processors yields the minimum resolution time (see Figure 8). On the other hand, it is interesting

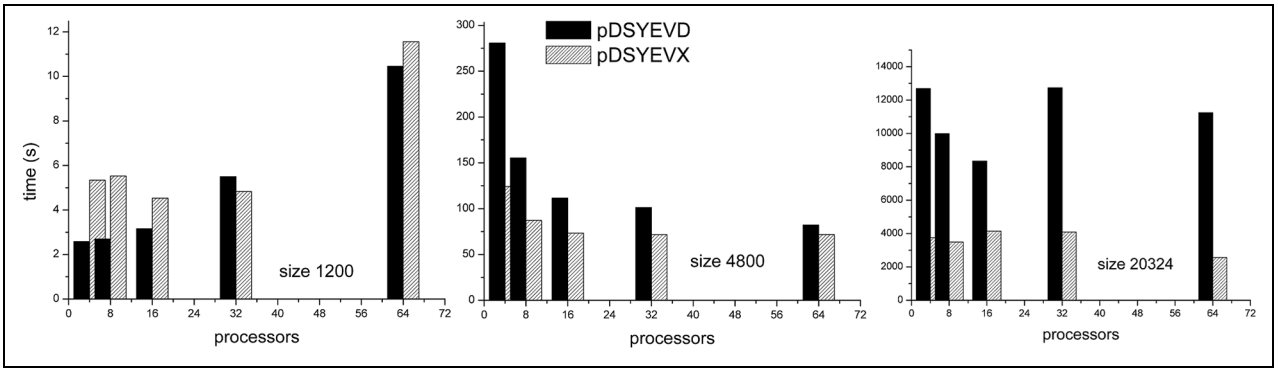

Figure 7. Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) and ScaLAPACK (pDSYEVX) speedup. In the LAMMPS case, the simulations were made over the environment used to obtain the conductance values (2200 atoms approximately). ScaLAPACK was tested over a fictitious arrays of atoms to analyze weak and strong scalability.

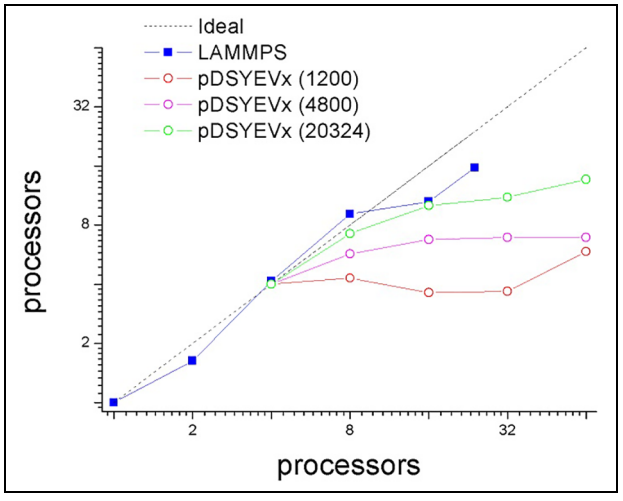

Figure 8. Computation times of the two solvers ScaLAPACK used in three array sizes.

to mention that the central processing unit (CPU) time consumed by pDSYEVD solver, based on the algorithm of dividing and conquering, is better for low matrix dimensions, but when we increase the number of atoms this advantage is lost, resulting in pDSYEVX always being the better the option.

It should be clarified the LAPACK serial code times are still advantageous compared to the parallel ones for small systems (less than 1000 atoms). However, for larger systems and a number of processors greater than four, parallel times several orders of magnitude smaller can be obtained.

It is also noteworthy that the loss of scalability or the dependence of the computer time frame on the resolution grid used by the library is noticeable, with anti-intuitive fluctuations being observed practically in all the related tests. However, as the matrix of our problem is non-variable in size and shape, it is possible to determine the optimal mesh at the beginning based on the casuistry developed and evolve the solution in the optimal configuration of processors to obtain the shortest time of possible computing.

On the other hand, the empirical potentials used in LAMMPS have a good scalability in a small number of processors, losing the ideal behavior for a number greater than 32, a situation that is known when running with MPI across multi-core nodes, where communication bottlenecks are often suffered.[27] We present a plot of this behavior in Figure 7 for a number of atoms close to 2200. Even though LAMMPS and TB use different algorithms, we apply them to analyze the same physical problem with different degree of transferability, but with the emphasis on the unavoidable considerations about the computational performance.

5.2. GPU environment

We compare the use of LAMMPS-GPU and GPUMD, two similar packages based on CUDA and GPUs. As we are not developers of these two packages, the objective of improving their speedup is outside the scope of this work. We run the same problem over a Nvidia GM107GL (Quadro K2200) board for systems with approximately 2200 atoms with different behavior for each package.

In the case of GPUMD is necessary to explain several points. Firstly, it was written in CUDA C++ and requires a CUDA-enabled Nvidia GPU of computing capability of no less than 3.5. Secondly, it was developed for use with GPUs with the aim to accelerate classical MD simulations. That establishes a difference with LAMMPS, which uses tools to accelerate preexisting solvers. This package is highly efficient for doing MD simulations with many-body potentials, such as the Tersoff potential using a single GPU, which can run 100 MD steps for a one-million-atom system within 1 second,[7] resulting in it being particularly good for heat transport applications. Thirdly, just a few classical and semiclassical potentials have been included until now, and there is much less progress on the

acceleration of force evaluations for many-body potentials compared to pairwise ones. With these points in mind and for the Tersoff many-body potential, the double precision performance of GPUMD is equivalent to that of LAMMPS running with about 100 CPU cores.[7] In our case the times involved to solve the same reference problems run with LAMMPS (2200 atoms) are in the range of 3600.0–4500.0 seconds for the entire calculus.

In the case of LAMMPS-GPU, there are some limitations in the implementations reported by the developers that make it difficult to know the optimal number of CPUs per GPU to be used. In general is necessary use a parameter sweep to find optimal settings for the different packages. In general, reported measures only show the best results for each package. In contrast, with the GPUMD software, LAMMPS uses an accelerator that takes advantage of the hardware features, but it is done in different ways and acceleration is not always guaranteed. As a consequence, for a particular simulation on specific hardware, one package may be faster than the other. In our test, we use double precision computations and the CUDA package. Moreover, the number of atoms is not within the recommended parameters for LAMMPS. Apparently, in a hybrid MPI/GPU parallelism, the speedup versus CPU will be significant with more than 32,000 particles per node (i.e., per GPU). This relation is not accomplished in our calculation and LAMMPS makes several load balances in a dynamical way. In our reference case of 2200 atoms we obtain computing times in the GPU similar to that involved in four MPI task calculations. When the size of the hole increase, the times using the GPU present a small decreasing, showing the mentioned dynamical load balance action. The use of the hybrid execution MPI-GPU does not help, showing a performance degradation. More information related to the two GPU packages can be founded in Zheyong et al.,[7] Moore,[27] and Nguyen et al.[28]

6. Conclusions

In this work, we analyzed different computational schemes for simulating the thermal transport of silicon nanosystems. We considered different interatomic models and implemented MD simulations using different codes. These approaches were applied to the study of the thermal transport of an ultra-thin silicon nanocrystal.

The thermal conductance obtained by LAMMPS, using the Tersoff potential, is similar to the one calculated by GPUMD, being of the order of 10^{-9} W/K, which is associated with a thermal conductivity of around 0.5 W/(m K), depending on the hole size. These values imply a thermal conductivity for a system of this size that is two orders of magnitude less than the bulk silicon conductivity of 148 W/(m K). This is a phenomenon already observed in ultra-thin silicon membranes.[29]

The Tersoff potential is an empirical bond-order potential fitted with experimental data. However, it is not capable of describing properly the effect of the surface and the low dimensionality of the system that have some implications in the transport of energy. Nevertheless, the efficiency from the computational point of view is in general better, even in the MPI version of LAMMPS. On the other hand, the lesser times involved in the simulations using the GPU (both in case of the LAMMPS-GPU and GPU) convert those options into the most effective from the computational point of view.

Using a semi-empirical TB model, which is more reliable due to its higher transferability, is adequate for nanostructured low-dimensional systems with a high surface-to-volume ratio. As a counterpoint, the computational costs increase considerably because this method involves diagonalization of a matrix at each time step, as a numerical performance analysis of different eigenvalue resolution methods was performed.

We optimized as far as we can the TB in-house solver. The results obtained by different diagonalization methods gave better figures for the solver included in the LAPACK/ScaLAPACK libraries. Although the computation times obtained with ScaLAPACK are comparable in order of magnitude to those obtained with LAPACK, they can be reduced by some factors in cases with a higher number of atoms.

Acknowledgment

The authors wish to express their gratitude to the TUPAC cluster, where the calculations for this paper were carried out.

Funding

This work has been partially carried out with resources provided by the CYTED cofunded Thematic Network RICAP (517RT0529) and the funding project PIO-CONICET 14420140100013CO.

ORCID iD

A Soba https://orcid.org/0000-0003-1147-508X

References

1. Xie XP, Liang MH, Choo ZM, et al. A comparative simulation study of silicon (001) surface reconstruction using different interatomic potentials. *Surf Rev Lett* 2001; 8: 5.
2. Bui DH and Yarmohammadi M. Direction-dependent electronic thermal conductivity and thermopower of single-layer black phosphorus in the presence of bias voltage and dilute charged impurity. *Phys E Low Dim Syst Nanostruct* 2018; 103: 76–80.
3. Neogi S, Reparaz S, Pereira LFC, et al. Tuning thermal transport in ultrathin silicon membranes by surface nanoscale engineering. *ACS Nano* 2015; 9: 3820–3828.
4. Srivastava GP and Tütüncü HM. Theoretical modelling of surface phonons. *Cent Eur J Phys* 2009; 7: 209–219.

5. Tersoff J. New empirical approach for the structure and energy of covalent systems. *Phys Rev B* 1988; 37: 6991.
6. Brenner DW. Empirical potential for hydrocarbons for use in simulating the chemical vapor deposition of diamond films. *Phys Rev B* 1990; 42; 9458.
7. Zheyong F, Wei Ch, Ville V, et al. Efficient molecular dynamics simulations with many-body potentials on graphics processing units. *Comput Phys Commun* 2017; 218: 10-16.
8. LAMMPS. http://lammps.sandia.gov (accessed October 2020).
9. Wang CZ, Chan CT and Ho KM. Empirical tight-binding force model for molecular-dynamics simulation of Si. *Phys Rev B* 1989; 39: 8586.
10. Kwon I, Biswas R, Wang CZ, et al. Transferable tight-binding models for silicon. *Phys Rev B* 1994; 49: 7242.
11. Voltz S, Shiomi J, Nomura M, et al. Heat conduction in nanostructured materials. *J Therm Sci Technol* 2016; 11: 1.
12. Houssa M, Dimoulas A and Molle A. *2D materials for nanoelectronics*. Boca Raton: Taylor & Francis, 2016.
13. Aspelmeyer M, Kippenberg TJ and Marquardt F. Cavity optomechanics. *Rev Mod Phys* 2014; 86: 1391.
14. Balandin AA Thermal properties of graphene and nanostructured carbon materials. *Nat Mater* 2011; 10: 569.
15. Mancardo Viotti AM, Bea EA, Carusela MF, et al. Simulación del transporte de calor en nanoestructuras de silicio. *Mecánica Computacional* 2018; 36: 2179–2187.
16. Bea EA, Carusela MF, Soba A, et al. Thermal conductance of structured silicon nanocrystals. *Model Simul Mater Sci Eng* 2020; 28: 075004.
17. Sanz-Navarro CF, Grima R, García A, et al. An efficient implementation of a QM–MM method in SIESTA. *Theor Chem Acc Theor Computat Model (Theoretica Chimica Acta)* 2010; 128: 825–833.
18. LAPACK. http://www.netlib.org/lapack/ (accessed October 2020).
19. ARPACK. http://www.caam.rice.edu/software/ARPACK/ (accessed October 2020).
20. EISPACK. http://www.netlib.org/eispack/ (accessed October 2020).
21. Jang H-J. Preconditioned conjugate gradient method for large generalized eigenproblems. In: Jang HJ (ed) *Trends in mathematics*. Vol. 4. ICMS, 2001, pp.103–109.
22. Kresse G and Furthmüller J. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys Rev B* 1996; 54: 11169.
23. Eloy RA. *Parallel implementation of Davidson-type methods for large-scale eigenvalue problems*. PhD Thesis, Universitat Politecnica de Valencia, 2012.
24. ScaLAPACK. www.netlib.org/ScaLAPACK/ (accessed April 2021).
25. Saad Y. *Numerical methods for large eigenvalue problems*. Minneapolis: Manchester University Press, 1992.
26. Vinazza D, Otero A, Soba A, et al. Initial experiences from TUPAC supercomputer. *Commun Comput Inform Sci* 2017; 796: 3–20.
27. Moore S. Accelerating LAMMPS performance. In: *2017 LAMMPS workshop and symposium* (ed S Moore), Albuquerque, New Mexico, 1–3 August 2017.
28. Nguyen TD, Carrillo JY and Brown MW. Experience with GPU acceleration for large-scale molecular dynamics simulations using LAMMPS. In: *workshop on accelerated high-performance computing in computational sciences* (eds TD Nguyen, JY Carrillo and WM Brown) Trieste, Italy, 25 May–5 June 2015. Trieste: ICTP.
29. Neogi S, Sebastian Reparaz J, Pereira LF, et al. Tuning thermal transport in ultrathin silicon membranes by surface nanoscale engineering. *ACS Nano* 2015; 9: 3820-3828.

Author biographies

EA Bea holds a BSc in Physics, MSc in Materials Science & Technology, and PhD in Exact Sciences (Chemistry). He is a researcher member at the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina, since 2011. He works in the Nuclear Fuel Management Department at the Comisión Nacional de Energía Atómica (CNEA), Argentina.

A Mancardo Viotti is a PhD student at the Institute of Sciences of the National University of General Sarmiento (UNGS), Buenos Aires, Argentina.

MF Carusela is full professor at the Institute of Sciences of the National University of General Sarmiento (UNGS), Buenos Aires, Argentina; and member of the National Scientific and Technical Research Council (Conicet), Argentina.

AG Monastra holds a PhD in Solid Physics from Université Paris XI (2001) and postdocs from the Weizmann Institute of Sciences, Israel (2002), and the Technische Universität Dresden, Germany (2003–2006), and is a researcher at Comisión Nacional de Energía Atómica (2006–2012) and the Comisión Nacional de Investigaciones Científicas y Técnicas, Argentina (since 2006). AG Monastra is a Full Professor at Universidad Nacional de General Sarmiento, Argentina (since 2012). Present research interests and expertise include non-equilibrium thermodynamics, heat transport, MD, and biophysics.

A Soba is an independent researcher in CONICET working in the Code and Models Section of the Nuclear Fuel Management department, National Atomic Energy Agency (CNEA). He is professor at the Beninson Institute, UNSAM.