A First-Principles Workflow for Automated Thermal Conductivity Computation for Thermal Barrier Coatings

by

Kerui Lai

Department of Mining and Materials Engineering

McGill University, Montreal

June 2022

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Science

Abstract

The application of thermal barrier coatings (TBCs) in gas turbine engines allowed humanity to extend the limit of operating temperature and to achieve better efficiency. The current generation of TBC materials has been used for decades and is showing inadequate potentials for further improvement. Thus it is critical to keep exploring new material candidates with better performance for the next-generation high-performance TBC. One most critical material property for selecting promising TBC material candidates is the thermal conductivity of the top coat, the outmost layer of a TBC. The combination of first-principles calculations, single-mode relaxation-time approximation, and the linearized phonon Boltzmann equation makes the prediction of lattice thermal conductivities of materials possible via computational simulations. However, the computational process involved requires a large number of inter-dependent subtasks. Therefore, it is important to streamline and optimize the process for better computational efficiency. Towards this, this thesis aims to develop workflow toolkit to automate the thermal conductivities computational process. In particular, a first-principles workflow software, Ph3pyWF, has been developed to conduct high-throughput computation of lattice thermal conductivities of TBC materials, able to calculate the lattice thermal conductivities of multiple oxide systems. The calculated results showed good agreement with the experimental results. High degree of automation and robustness was proven throughout the tests, indicative of this workflow to serve as an efficiency booster for high-throughput computational exploration of TBC materials.

Résumé

L'application de revêtements de barrière thermique (*Thermal Barrier Coatings* ou TBC) dans les turbines à gaz a permis à l'humanité d'augmenter leurs limites de température de fonctionnement et d'obtenir une meilleure efficacité. La génération actuelle de matériaux TBC est utilisée depuis plusieurs décennies et présente un potentiel inadéquat d'amélioration future. Il est donc essentiel de continuer à explorer, pour la nouvelle génération, de nouveaux candidats de matériaux TBC possédants de meilleures performances. La conductivité thermique de la couche supérieure, soit la couche externe d'un TBC, est l'une des propriétés les plus importantes pour la sélection de matériaux prometteurs. La combinaison de calculs des premiers principes, d'approximation de temps de relaxation monomode et de l'équation de Boltzmann de phonon linéarisé rend possible la prédiction de conductivités thermiques des réseaux de matériaux par simulations numériques. Toutefois, le processus de calcul en cause nécessite un grand nombre de sous-tâches interdépendants. Il est donc important de rationaliser et d'optimiser le processus pour obtenir une meilleure efficacité de calcul. Pour ce faire, cette thèse vise à développer un outil de gestion de processus de travail pour automatiser le calcul de conductivités thermiques. En particulier, un logiciel de gestion de processus de travail basé sur les premiers principes, Ph3pyWF, a été développé pour effectuer un calcul à haut débit de conductivités thermiques de matériaux TBC, capable de déterminer la conductivité thermique de réseaux de multiples systèmes d'oxydes. Les résultats calculés ont permis d'obtenir des résultats concordants avec les résultats expérimentaux. Un degré élevé d'automatisation et de robustesse a été prouvé tout au long des essais, ce qui indique que ce processus de travail permettra de décupler l'efficacité de l'exploration numérique à haut débit des matériaux TBC.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor Professor Jun Song for giving me this wonderful research opportunity to work in his research group. I appreciate all the precious support and guidance throughout this research. It is pleasant to work under Professor Song's supervision.

I would like to extend my gratitude to all the members of McGill Multiscale Modeling of Materials Group. Special thanks to my senior colleagues Dr. Guoqiang Lan for teaching me how to set up the development environment and the usage of many Python packages involved in this work, and Yuxuan Wang for the tremendously helpful insights regarding the research topic. I'm truly thankful for your patience while answering the rookie questions that I asked.

I would also like to appreciate the generous financial support from National Science and Engineering Research Council of Canada (NSERC) and McGill Engineering Undergraduate Student Masters Award (MEUSMA).

Many thanks to my friends, especially Guanlan Huang, Rui Li, Zhening Zhang, for their emotional support, and for all the precious time we spent together. Thank all the caffeinated drinks I've had for keeping me awake, and all the memes for cheering me up.

Finally, I cannot express enough thanks to my mother Weiqun Guo, and father Zhenshan Lai, for their everlasting love and unconditional support. Since my childhood, they have always been so encouraging no matter what nerdy hobbies I wanted to do. Thank you for bringing me to this wonderful world.

Contribution of Authors

The author of this thesis, Kerui Lai, is the primary author of all the chapters presented herein. This thesis was also made possible with the supervision and advice of Professor Jun Song and collaboration with Dr. Guoqiang Lan.

Professor Jun Song helped define the concept, gave feedback on the computation results, and reviewed and edited the writing of this thesis.

Dr. Guoqiang Lan helped set up the development environment and gave feedback on the programming aspect and computation results.

Table of Contents

List of Figures

List of Tables

Chapter 1: General Overview

1.1. Introduction

Gas turbine engine, a type of continuous internal combustion engine with gas as the working fluid, has profound impact on human society. With higher power-to-weight ratio than traditional intermittent-combustion engine, it is widely adopted in various of industries including aerospace and power generation. Since the birth of the first gas turbine engine, scientists and engineers have been striving to increase the operating temperature to achieve higher efficiency and power-to-weight ratio. Studies have shown that the work output increases by approximately 10% for every 50 K increase in operating temperature, resulting in a 1.5% gain in thermal efficiency [1,2]. To push the performance of gas turbine engines beyond the limitation of metallic materials, particularly through elevating the operating temperature, ceramic thermal barrier coatings (TBCs) were developed.

A typical TBC system consists of three layers: the ceramic topcoat for thermal protection, the bond coat for connection between superalloy substrate and topcoat, and the thermally grown oxide between bond coat and topcoat. The ceramic materials for the topcoat are designed to have thermal conductivity (less than 3 W/m-K at 1000°C) magnitudes less than that of the superalloy substrate (~30±5 W/m-K at 1000°C). Modern TBC systems provide significant reduction in the surface temperature (up to a few hundred degrees Celsius) of superalloy substrate, allowing higher operating temperature. Besides of application in gas turbine engines, TBCs can also be employed in automotive industry to improve fuel efficiency and reduce exhaust pollution [3,4].

Traditionally, new material systems are discovered mainly by conducting large number of trial-and-error experiments, which are time consuming and require specialized equipment for characterization of material properties. Accompanied with the development of modern quantum

mechanics and the exponentially growing computational capability, a new field of materials science has emerged, known as computational materials design. It allows the prediction of materials properties without conducting experiments, and thus drastically reduces the cost of materials and labor work. Electronic structure methods such as Hartree-Fock (HF) and density function theorem (DFT) enable first-principles (or "ab-initio") simulation of material systems, which as the name implies, uses minimal to no empirical knowledge of the systems. Currently, computational methods have been adopted in the exploration of many categories of new functional materials including catalysts [5–7], thermoelectrics [8–11], and battery materials [12–14]. In the exploration of new TBC materials, computational materials design has also been proven as a powerful tool for screening promising candidates from huge pool of material systems.

Oftentimes when conducting computational study of materials, researchers find themselves doing repetitive works. Typical procedure of computational studies involves preparation of input files, submission of jobs, retrieval and parsing of output files. In the case where large number of computational jobs are conducted in parallel, manual operation is inefficient and error prone. For example, computational approach for lattice thermal conductivity often leads to hundreds to thousands of first-principles calculations. Additionally, error identification and recovery for failed subprocesses is usually complicated and frustrating. All the aforementioned issues associated with manual running of computational studies raise significant demand for automation of computational workflow.

To accelerate the process of computational materials exploration, efforts have been made to develop new software infrastructures that combine automated computational workflows and database storage solutions. Such infrastructures do not only improve the simplicity and efficiency

of computational studies, but also ensure better reproducibility of results and reusability of computational codes.

1.2. Objectives

As described above, workflow is essential to accelerate the exploration of new TBC top coat materials. As such, this work aims to develop a highly automated workflow toolkit to conduct first-principles computation of lattice thermal conductivities of TBC materials. The overall goal of this thesis work to design a software that can manage and execute large number of inter-dependent subtasks with minimal user intervention while reliably predicting lattice thermal conductivities. Specifically, the following objectives are targeted, listed below:

- Design and develop a computational workflow software to predict lattice thermal conductivity using first-principles methods.
- Validate this workflow by calculating lattice thermal conductivities of multiple oxide systems using this workflow and comparing the calculated results with experimental values.
- Evaluate the performance of the workflow and identify directions for future improvement.

Chapter 2: Literature Review

2.1. Overview of Thermal Barrier Coatings (TBC)

2.1.1. History of TBCs

During World War II, turbojet powered aircrafts were developed and adopted by the military of multiple nations [15]. The demand of higher engine operating temperature has emerged since then. The first TBC applied in aerospace industry was developed in the 1950s by the National Advisory Committee for Aeronautics (NACA) and the National Bureau of Standards (NBS) [16]. Frit enamels were used as the topcoat of this TBC system, and were applied to the substrate by spraying followed by drying and firing [17]. In the 1960s, Pratt & Whitney introduced TBC system with innovative flame-sprayed Ni-Al bond coat, which marked the starting point of TBC systems with multi-layered structure [18]. During this stage, the topcoat material was mainly magnesia-stabilized zirconia (MSZ), which was susceptible to destabilization and has relatively low limit of temperature. In the 1980s, TBC performance was significantly improved by replacing the topcoat materials with yttria-stabilized zirconia (YSZ), which has much better durability and thermal resistance [18]. Specifically, the partially stabilized zirconia with ~7 wt.% yttria (7YSZ), with high resistance against thermal cycling than the fully stabilized counterpart, has become the industry standard for the following decades [19].

2.1.2. Structure of TBCs

Most commonly, a modern TBC system consists of three layers of different materials on the surface of substrate: bond coat, thermally grown oxide (TGO), and topcoat. Notably, the substrate is sometimes considered as part of a TBC system since its properties affect the selection of the other layers materials. The bond coat serves two major purposes: to strengthen the adhesion between ceramic topcoat and metallic substrate (by reducing the thermal expansion mismatch

between these layers), and to provide resistance against oxidation of the substrate [20]. The typical material system of the bond coat is MCrAlY, where M can be Ni and/or Co. The presence of Ni and/or Co ensures the high-temperature mechanical properties, similar to that in superalloys. Addition of Co can further improve hot corrosion resistance [21]. The presence of Al and Cr mainly serves as a reservoir that replenishes the TGO scale [22]. The presence of Y improves the mechanical strength of TGO scale at high-temperature, as well as reduces the growth rate of TGO scale [23]. During operation, oxidation of the bond coat results in formation of TGO, which typically presents in the form of α-Al_2O_3 scale, between the bond coat and the topcoat. The growth of TGO leads to several detrimental effects including crack initiation, increase in induced strain, and spallation of the ceramic topcoat [24,25]. However, with carefully engineered bond coat, the dense and defect-free α-Al_2O_3 formed at slow growth rate can act as a oxygen diffusion barrier due to its extremely low oxygen ionic diffusivity, thus provide further protection to the substrate [26]. The outermost layer of a TBC system is the topcoat. The topcoat provides excellent thermal insulation and significantly reduces the surface temperature of the substrate. The material systems of the topcoat must have minimal thermal conductivity and optimal thermal expansion coefficient. Usually, the topcoat is manufactured to be porous to further improve its thermal insulation performance.

2.1.3. Manufacturing Techniques for TBCs

The microstructure of coatings is directly influenced by manufacturing technique, which has a substantial impact on TBC performance. Highly porous microstructure can lead to more than 60% reduction in effective thermal conductivity of 7 wt.% yttria stabilized zirconia [27]. TBCs can be deposited on substrates using various techniques including atmospheric plasma spray, electron beam physical vapor deposition, and high-velocity oxygen fuel. Below several common techniques are described in detail.

a. Atmospheric Plasma Spray (APS)

APS is one of the most common deposition methods of TBCs. In essence, the coating material is melted while passing through a high-temperature plasma and is accelerated to the substrate surface at high velocity. The plasma is generated between a tungsten cathode rod and a water-cooled copper anode nozzle with flowing inert gas. High-frequency AC discharge is used to start the high intensity electric arc, which heats the inert gas to plasma temperature. DC power is then applied to maintain the following operation. The plasma expands and accelerates to form a jet exiting the nozzle at high velocity. Coating material powder particles are then injected into the plasma and heated to molten state. These molten powder particles are accelerated through the plasma, and strike the substrate, forming "splats" on the substrate surface. Eventually these splats solidify rapidly and form the coating. [20,25]

Since APS technique uses high intensity electric arc as the heat source, it is able to reach extremely high temperature (above 10,000K) and allows deposition of many materials and better composition control. Plasma spraying process can also achieve high deposition rate at relatively low cost [25]. The microstructure of coating deposited by APS contains many inter-splat pores, which are oriented in parallel with to the substrate surface. These inter-splat pores provide effective thermal impedance, which results in lower through-thickness thermal conductivity of APS TBCs [28]. However, they also act as crack initiators and lead to segmentation at elevated temperature. Eventually, this characteristic results in a poor thermal cycle/shock resistance [29–31]. In addition, coating deposited by APS is subjected to oxidation due to operation under atmospheric condition. This issue can be neglected when the coating material is oxides (e.g. ceramic topcoat) or when oxide contamination is tolerable [32].

APS is commonly used to apply top coat on the components that work under less harsh mechanical conditions, such as combustion chamber, fuel vaporizer, and other static components [32]. In addition, bond coat can be applied using plasma spraying technique under low air pressure (LPPS) or vacuum (VPS) [24].

b. Electron Beam Physical Vapor Deposition (EB-PVD)

In EB-PVD, the coating material is vaporized by a high-intensity electron beam and solidifies upon reaching the substrate surface. One or more electron guns generate the high-intensity electron beam, direct and focus electron beam to the ingot of coating material. In some situation, the ingot is held by a moving device and can be rotated and fed to maintain the position of vaporizing surface. The deposition chamber must maintain high vacuum condition to ensure free passage for the electron beam and gaseous coating material. [33]

The coating deposited by EB-PVD exhibits single-crystal columnar microstructure. Intercolumnar pores are also present in the coating, aligned perpendicular to the substrate surface. This property renders EB-PVD TBCs high strain tolerance, and reduced thermal expansion mismatch, thus improved resistance to spallation. Additionally, EB-PVD TBCs are reported to have strong mechanical adhesion, high surface smoothness, and excellent thermal shock resistance during thermal cycling [25,34]. However, this columnar morphology characteristic also results in significantly higher thermal conductivity. Top coats deposited using EB-PVD have approximately 80% higher effective thermal conductivity than those deposited using APS [28].

EB-PVD technique demands high equipment cost and operation cost due to the complexity of electron guns and the requirement of high-vacuum in deposition chamber. Also, EB-PVD has relatively low evaporation efficiency and low deposition rate since long source-to-substrate distance is required to protect substrate from overheating [34].

EB-PVD is often used to apply topcoat on the critical components such as high-pressure turbine blades, where sufficient strain tolerance is more important than low thermal conductivity [18]. Improved EB-PVD variants have been developed to overcome the aforementioned disadvantages. Hass et al. developed directed vapor deposition (DVD) which uses a jet of inert carrier gas to guide vaporized coating materials towards the substrate [34]. The electron gun used was modified to work under low-vacuum condition (10^{-3} – 10 Torr) which is much easier and cheaper to maintain than the high-vacuum condition (10^{-4} – 10^{-8} Torr) required by conventional EB-PVD [35,36]. EB-DVD technique demonstrated remarkably greater materials utilization efficiency and deposition rate [36]. To further reduce the thermal conductivity of topcoat while maintaining sufficient strain tolerance, a new technique using EB-DVD was later developed to deposit coatings with zig-zag shaped columnar morphology [37]. The zig-zag shaped pores reduce the thermal conductivity to a similar level as coatings deposited by APS [38].

c. High-Velocity Oxygen Fuel (HVOF)

High-velocity oxygen fuel, usually referred to as HVOF, is a relatively new thermal spray coating technique. It shares many similarities with APS. In HVOF, oxygen and fuel are continuously fed into the combustion chamber in a water-cooled nozzle. The oxygen-fuel mixture combusts to produce high-velocity and high-temperature (up to 3000°C) stream. Coating material powders are fed into the combustion chamber by inert carrier gas, heated to molten or semi-molten state and sprayed to substrate surface at supersonic velocity. [25]

Due to high spraying velocity, HVOF coating exhibits high density and decent adhesion. It is also reported that HVOF coating has fewer microcracks and thus provides better passivation against corrosion [39]. Such properties make HVOF a suitable technique for bond coat deposition. Since the maximum temperature in HVOF is lower than that in APS due to the fuel-oxygen

combustion limitation, coating material powders must be ultra-fine and have narrow size distribution. Coarse powders lead to deposition of insufficiently melted particles, which is detrimental to the surface finish and structural properties of the coating layer. Particles finer than the desirable size range may degrade before reaching the substrate surface. [40]

Additionally, HVOF inflict thermal stress on the substrate due to high temperature gradient during spraying. This effect limits the application of HVOF. However, it is less detrimental for applying bond coat on superalloy substrate. [41]

2.1.4. TBC Design Criteria

TBCs are complex systems that operate under extreme environments. To guarantee that the TBC system achieves the expected performance, a variety of factors must be considered while selecting materials as TBC components. Below several key factors considered in the design of TBCs are elaborated.

a. Thermal Conductivity of Top Coat

The most obvious purpose of TBC materials is to provide heat insulation to protect the metallic substrate from the hot gas steam which has temperature may exceed the melting point of most superalloys. Therefore, a low thermal conductivity is a necessity for any candidate TBC material to provide adequate reduction in the substrate surface temperature.

At microscopic scale, to reduce the thermal conductivity of TBC system, the intrinsic thermal conductivity of the topcoat material must be reduced. In general, this can be achieved by reducing the phonon group velocity and increasing anharmonicity of the interatomic bonds.

The group velocity of acoustic phonon is approximately $\sqrt{E/\rho}$, where E is the bulk elastic modulus which reflects the strength of interatomic bonds, and ρ is the density of the material [42]. This equation provides a selection guideline that low intrinsic thermal conductivity can be found

in materials with smaller elastic modulus and larger density. However, materials with weak interatomic bonds tend to have low melting point, which is unfavorable for TBC application. One approach to resolve this deadly issue is to increase the mass contrast of compositional elements and the complexity of the primitive cells, instead of directly weakening the interatomic bonds [43].

To increase anharmonicity of the interatomic bonds, we may look for material systems with open-frameworks in crystal structures. Some example materials include Si-Ge clathrates, filled skutterudites, and β-pyrochlore oxides [44–46]. Such material systems have very weak interatomic bonds between host atoms and guest atoms inside of the open-framework, and thus exhibit high anharmonicity.

Moreover, the thermal conductivity of a TBC system can be further reduced by introducing point defects in the lattice structure and refining grains to increase grain boundary density [47]. Both approaches provide extra phonon scattering processes, and hence reduce thermal conductivity. Macroscopically, porosity can be introduced by altering the deposition techniques, as described in the manufacturing techniques section, and will also reduce the thermal conductivity of the TBC system.

b. Thermal Expansion Mismatch

The difference in the coefficient of thermal expansion (CTE) between different coating layers is a critical factor to consider when designing TBC systems. Usually, the ceramic topcoat has much lower CTE than the metallic substrate. In a typical TBC system, the CTE of 7YSZ topcoat, NiCoCrAlY bond coat, and IN738 substrate are $10.7\times10^{-6}K^{-1}$, $17.5\times10^{-6}K^{-1}$, and $16\times10^{-6}K^{-1}$ respectively [48]. Mismatch of CTE leads to thermal stress near the interface between layers, which in long term initiates cracks and accelerate crack propagation.

The failure behavior due to CTE mismatch of TBC is usually studied by conducting thermal shock test, also known as burner rig test [49]. During burner rig test, TBC is tested with higher heating and cooling rate, but with shorter holding time in the high-temperature condition. Such test condition results in negligible TGO growth, thus allows more focused investigation on the failure due to CTE mismatch.

The risk of CTE mismatch may be mitigated via either intrinsic or extrinsic methods. Intrinsically, designing topcoat ceramic materials with higher CTE that matches the bond coat and substrate properties would definitely be effective, but is challenging due to the nature of ceramic materials. Extrinsically, and more commonly, fabrication techniques can be controlled to increase the porosity of the topcoat layer, reducing the global stiffness of the topcoat and hence improve the strain tolerance.

c. Top coat Sintering

Under elevated temperature below the melting point of the topcoat, sintering occurs in the topcoat layer. It is a microstructure transformation with closing of pores and micro-cracks. Sintering process results in densification of coating layer and increase in thermal conductivity. The densification effect leads to increase in the global elastic modulus (stiffness) of the topcoat, thus reduces the strain tolerance and makes the system more susceptible to failure due to CTE mismatch. The increase in thermal conductivity is clearly an unfavorable effect, as it indicates the deterioration of the most critical functionality of the topcoat. With increased thermal conductivity, the topcoat provides insufficient thermal shielding for the metallic substrate, which could lead to severe deformation of the entire component.

d. CMAS Infiltration

During operation of aerospace gas turbines, it is almost impossible to avoid intake of silicate particles existing in suspended dust, sand, and even volcanic ash cloud. These silicate particles mainly consist of calcium-magnesium-aluminosilicate (CMAS). At elevated temperature above 1200°C, molten CMAS wets the topcoat surface upon contact, and infiltrates into the porosity of the topcoat layer [50]. The topcoat with columnar as-fabricate porosities is more susceptible to CMAS penetration along the perpendicular pores. Mechanically, such infiltration stiffens the topcoat layer, which along with CTE mismatch between CMAS and topcoat material, leads to reduction in strain tolerance and coat delamination through operations. CMAS infiltration also densifies the topcoat layer, causes increase in thermal conductivity and degradation of TBC insulation effectiveness. If CAMS infiltration reaches the TGO layer, chemical interaction between CMAS and TGO may promote creep cavitation in the bond coat and lead to crack propagation in the metallic substrate.

In particular, when 7YSZ topcoat is in contact with CMAS under temperature above the CMAS melting point, the tetragonal prime (t') phase dissolves in molten CMAS, and reprecipitate in form of either yttria-enriched or yttria-depleted phases [51]. The yttria-depleted phase undergoes destructive phase transformation upon cooling.

Current methods to mitigate CMAS infiltration issue mainly focus on chemical modification of the topcoat system. To obtain better resistance against CMAS, the topcoat need to be designed to react with CMAS upon contact, to form a stable crystalline layer which impedes further infiltration, while the reaction must be kinetically more competitive than the infiltration process [50].

e. Oxidation of Bond coat

During prolonged operation under elevated temperature, oxygen atoms diffuse through the topcoat and oxidize the metallic bond coat. The oxidation reaction occurs at the interface between the topcoat and the bond coat, yields continuous layer of oxide scale, and leads to the growth of TGO layer. As TGO grows, the transformation from metal to oxide causes significant density reduction and volume expansion, and magnifies the effect of CTE mismatch between the TGO and the topcoat. The volume expansion and CTE mismatch consequently leads to residual stress build-up in the TGO layer. Once the TGO growth reaches the critical thickness, spallation begins to occur in the TGO layer, causing the breakaway of both the TGO and topcoat layers.

Before TGO growth reaches the critical thickness, failure may occur due to chemical reactions [52]. During exposure to elevated temperature, the aluminum content in the bond coat drops. Depletion of aluminum in the bond coat causes oxidation of the Ni, Cr, Co elements in the bond coat by reacting with existing alumina scale in the TGO. The Ni, Cr, Co enriched oxide forms a porous layer of with increasing pore size with time, replacing the protective alumina scale. Consequently, this aluminum depleted porous oxide has poor mechanical strength and is susceptible to spallation. In addition, such layer is inadequate as an oxygen barrier, allowing oxygen atoms diffuse into the substrate. This process is also known as chemical failure.

f. Corrosion by Impurities

Vanadium exists in the fuel used by gas turbines as a detrimental impurity. During combustion, vanadium reacts with oxygen to form vanadium pentoxide (V_2O_5). Presence of vanadium is especially detrimental for TBC systems using YSZ as the topcoat material [53]. Upon contact, vanadium pentoxide rapidly reacts with yttria in the YSZ system to produce yttrium vanadate through the following reaction:

$$V_2O_5 + Y_2O_3 \rightarrow 2YVO_4 \quad (1)$$

This reaction causes depletion of yttria in the YSZ system, and consequently leads to destabilization and destructive tetragonal-to-monoclinic phase transformation upon cooling.

A more uncommon TBC system using calcium silicate as the topcoat material has been proven to have better resistance against vanadium corrosion. However it is susceptible to corrosion caused by sulfur, which is another common impurity existing in industrial fuels [53].

2.2. Materials Systems for TBC Application

2.2.1. 7 wt.% Yttria stabilized zirconia (7YSZ)

Partially stabilized zirconia with ~7 wt.% yttria (7YSZ), ever since its adoption in the 1980s, has become the industrial standard for TBCs. Pure zirconia can exist in three phases: monoclinic phase stable at temperature lower than 1170°C, tetragonal phase stable from 1170°C to 2370°C, and cubic phase from 2370°C to the melting point (2680°C) [54]. During cooling from melting temperature, pure zirconia is susceptible to cracking induced by the tetragonal-to-monoclinic phase transformation which involves a 3-5% volume expansion [55,56]. This property makes pure zirconia extremely difficult to fabricate, and also not practical for TBC applications where the material must withstand numerous of thermal cycle.

To improve the thermal cycle resistance of zirconia, additives such as calcia (CaO), magnesia (MgO), and yttria (Y_2O_3) were added in attempt to stabilize the high-temperature phases. However, zirconia-calcia and zirconia-magnesia systems are proven to be not viable. Calcia and magnesia were added for stabilizing the high-temperature cubic phase, but the effect was limited. The cubic phase becomes unstable below about 1140°C for zirconia-calcia, and below 1400°C for zirconia magnesia [57]. Zirconia-yttria system on the other hand, showed excellent performance superior to zirconia-calcia and zirconia-magnesia systems. While stabilizing high-temperature cubic phase, yttria additive also helps forming a non-transformable tetragonal prime (t') phase [58]. Initially, the material system was designed to contain 12-20 wt.% of yttria to fully stabilize the cubic phase. It was later proven experimentally by Stecura that zirconia partially stabilized by 6-8 wt.% of yttria achieves optimal resistance to thermal cycle [59]. Such zirconia system partially stabilized with 6-8 wt.% of yttria was then named 7YSZ and was widely used since.

In order to achieve lower thermal conductivity with YSZ systems, research was conducted to study the effect of doping elements, commonly rare-earth (RE) elements. By adding 1 mol% of Yb_2O_3 and 1 mol% of Gd_2O_3 to the 7YSZ system, replacing part of the yttria composition, the result coating showed significant improvement, with thermal conductivity reduced from 1.45 W/m-K to 1.2 W/m-K at 1316°C. The doped YSZ also showed better sintering resistance, with ratio of thermal conductivity increment after 20 hours of operation reduced from 52% to 37%. These results shows improved microstructural and chemical stability of the YSZ doped with RE elements. [60]

Compared with the previous generations of TBC materials, 7YSZ has lower thermal conductivity, and higher thermal expansion coefficient which reduces thermal expansion mismatch [61]. It also has better thermo-chemical compatibility with TGO. In addition, the t' phase existing in 7YSZ provides decent fracture toughness of about 2 MPa$\sqrt{m}$ due to the ferroelastic toughening mechanism [62,63].

Drawbacks of 7YSZ as TBC materials include limited operation temperature, high sintering rate, susceptibility to CMAS degradation, and inadequate thermal conductivity. At temperature above 1200°C, the metastable t' phase begins to decompose into yttria enriched cubic phases and yttria depleted tetragonal phases. During cooling to room temperature, while the yttria enriched phases remain in cubic or transform to t' phases, the yttria depleted phases, similar to pure zirconia, are subject to destructive tetragonal-to-monoclinic phase transformation [64]. The sintering of 7YSZ starts at 800-1000°C, and reaches peak rate at 1150-1280°C [65]. Such sintering process occurs under operating condition, and results in reduction in porosity and ultimately increase in thermal conductivity. Upon contact with CMAS during operation, 7YSZ can be easily infiltrated by CMAS, leading to phase decomposition and delamination during thermal cycling [66].

Finally, despite having relatively low thermal conductivity at the beginning of its era, 7YSZ is currently showing inadequately low thermal conductivity while the industry is pursuing higher operating temperature.

2.2.2. Pyrochlore Ceramic Oxides

One promising candidate material system to replace 7YSZ for applications operate at above 1200°C is material with pyrochlore structure $A_2B_2O_7$ where A and B are 3+ and 4+ ions respectively. Specifically, rare-earth zirconate pyrochlores, with chemical formula $RE_2Zr_2O_7$, are the most widely studied. In material systems with chemical formula $A_2B_2O_7$, the phase stability is highly dependent on the ratio of ionic radii of cations A to B: $r(A^{3+}):r(B^{4+})$. It is found that for the systems with $r(A^{3+}):r(B^{4+}) > 1.42$, the cubic pyrochlore phase is stable, whereas the systems with smaller ratio of ionic radii are expected to have a defect fluorite phase at elevated temperature [67].

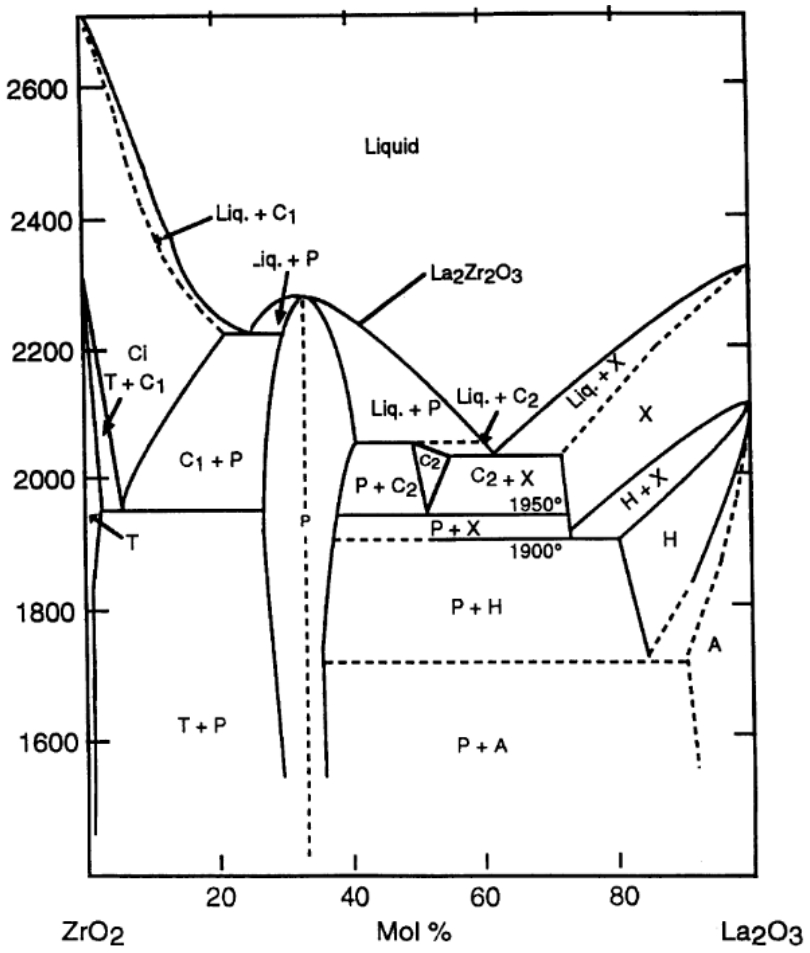

Figure 1: Binary phase diagram of La2O3-ZrO2, "P" and "C1" indicate the pyrochlore and fluorite phases respectively. Temperature unit in °C. [68]

As illustrated in Figure 1, a typical rare-earth zirconate system $La_2Zr_2O_7$ can retain a stable cubic pyrochlore structure from room temperature up to its melting point at ~2300°C, exhibiting excellent high-temperature stability. In lanthanide series from La to Lu, the stability of the pyrochlore phase reduces as the ionic radius decreases. According to the calculated phase diagram [42], systems with La, Pr, Nd, Sm, Eu and Gd have stable cubic pyrochlore phases at typical gas turbine operating temperature (~1300°C). System with RE elements from Tb to Lu, the high-temperature stable phase is either the fluorite phase or mixture of fluorite phase and δ-zircoate phase.

Among rare-earth zirconate pyrochlores, $La_2Zr_2O_7$ and $Gd_2Zr_2O_7$ are the most comprehensively studied. At 1000°C, $La_2Zr_2O_7$ has thermal conductivity of 1.56 W/m-K, more than 20% lower than the thermal conductivity of YSZ (2.12 W/m-K) [69]. The ionic conductivity of $La_2Zr_2O_7$ is significantly lower than that of YSZ, makes it less oxygen-transparent, and hence a good barrier against oxygen diffusion to reduce TGO growth rate [70]. Despite having lower melting temperature than YSZ, pyrochlores have low sintering activities, giving them stronger resistance against densification by sintering [71]. In addition, $Gd_2Zr_2O_7$ has been tested to have good resistance against CMAS infiltration, and the study of mechanism shows that this resistance is valid for the other rare-earth zirconate pyrochlores [72]. Upon contact with CMAS, the zirconate dissolves into the molten CMAS, reacts to form crystalline apatite reaction layer, blocking the possible flow channels and impede further infiltration. Finally, the structure of cubic pyrochlore can be described as "a network structure of corner linked BO_6 octahedra with the A atoms filling the interstices", thus both A-sites and B-sites can be substituted by impurity atoms, allowing further modification of thermal properties to the system [73]. The effect of different substitute atoms is an excellent research subject where computational materials design can be applied.

Currently the main drawback of rare-earth zirconate pyrochlores is the low CTE. The CTE of $La_2Zr_2O_7$ is $9.1\times10^{-6}\,K^{-1}$, which is much lower compared with $10.7\times10^{-6}\,K^{-1}$ for YSZ, making it subject to more significant CTE mismatch and hence shorter lifetime [69,74]. One approach to mitigate the CTE mismatch issue is to apply double-layer topcoat, with rare-earth zirconate pyrochlores on top of YSZ [75]. In a double-layer topcoat, the pyrochlores provide major thermal insulation and resistance against sintering and CMAS, while YSZ acts as a buffer to reduce CTE mismatch between pyrochlores and bond coat.

2.2.3. Perovskite Ceramic Oxides

Perovskite is a class of materials with chemical formula ABO_3 that has corner-sharing octahedral crystal structure. It is able to form solid solution with many different ions including ions with large atomic mass [74]. The properties of perovskite can be easily modified due to the ordering effect of cation in B-sites [76]. Some zirconate perovskites have extremely high melting point such as $SrZrO_3$ (3073K) and $BaZrO_3$ (2963K) [69]. However perovskites have significantly lower CTE than YSZ, and may undergo detrimental orthorhombic-to-pseudo-tetragonal phase transformation [69].

2.2.4. High Entropy Ceramic Oxides

In the very recent years, high entropy ceramics (HECs) has become an emerging class of materials that attracted many research interests. Although currently HECs do not have a standardized definition, they commonly refer to ceramic systems with five or more principle cations, where the ideal configuration entropy greater than $1.5R$ (R is the gas constant) per mol of cations [77]. HECs have remarkably low thermal conductivity due to the strong phonon scattering effect created by mass and bond disorder [78]. However, the compositional and configurational complexity of HECs demand intensive experiments and/or simulations to understand the mechanisms and to obtain predictive descriptors.

2.3. High-Throughput Computational Materials Screening

With increasing complexity of the emerging classes of materials, HECs for example, traditional experimental techniques are showing inadequate efficiency. The design and exploration of these complex materials via conventional experimentation would require numerous parallel experimental studies of thousands of materials simultaneously, which is a daunting task. Even with conventional computational methods which allow running tasks in parallel, the execution and management of numerous subprocesses can still be overwhelming. Hence, further automation of computational methods is desired. By the combination of first-principles computation and automated workflow frameworks, high-throughput (HT) materials screening is made possible, where simulations of a large pool of material systems are generated, managed, and analyzed in parallel. In some definitions, in a HT materials screening process, the throughput of data produced is beyond the capability of the researcher's direct intervention, and must be analyzed automatically [79]. Computational workflow frameworks, such as Atomate [80] and AFLOW [81], have been developed to conduct HT material screening by highly customizable workflows. In computational material design process, it is often required to specify different simulation parameters based on the material systems. These parameters can usually be determined automatically in a HT computational framework by performing convergence tests or with the help of other utilities. Furthermore, in HT material screening, the screening efficiency can be significantly improved with well-defined "descriptors" as criteria, which are empirical quantities connecting the calculated microscopic properties to the macroscopic properties of the materials [79,82].

2.4. Workflow in Computational Materials Science

2.4.1. Workflow Overview

One essential aspect of computational materials science is the efficiency of execution and management of computational subprocesses. In recent years, workflow emerges as a valuable tool to address such need. The term "workflow" refers to the sequence of operations where information is passed from one operation to another. In the field of computational materials science, a workflow consists of one or multiple computational subprocesses linked internally by pre-defined procedures to conduct a certain type of analysis of a material. Commonly, a workflow can be represented as a directed acyclic graph (DAG), in which the nodes represent the computational subprocesses involved and the edges represent the dependency relation between the subprocesses [83]. In some cases where iteration is required by the application, workflows may have cycles with properly defined break conditions in the graph representation, making it a directed cyclic graph (DCG) instead of DAG [84].

The most critical feature of a workflow is automation. A workflow that conducts and manages complex computational tasks with a high degree of automation can significantly reduce repetitive work and improve the efficiency of research. Ideally, user intervention is only required to specify input parameters at the beginning of the workflow, and to collect output results at the end of the workflow. Workflow with a high degree of automation also enables non-expert users to conduct high-throughput exploration of materials.

Other recommended features of a modern workflow include but not limited to reusability, reproducibility, and well-designed data management strategy. To avoid reinventing utilities with similar functionality, a workflow should be as reusable as possible. A workflow with adequate reusability can be used to build higher-level workflows as a whole or provide modular

subprocesses to build other workflows. For example, since many computational materials studies require optimization of lattice structure, a robust structure optimization subprocess or workflow is guaranteed to improve reusability.

In all fields of scientific research, reproducibility has great importance as it indicates the reliability of the study. By nature, computational studies can be more reproducible since digital data can be trivially cloned. To achieve this, a workflow needs to store information of the input parameters specified, the methodology used, the intermediate results obtained from the subprocesses, and even the environment in which the subprocesses were executed.

In order to achieve a high degree of automation for more complex scenarios, the workflow needs to be dynamic. A dynamic workflow is able to modify the parameters and dependencies of its existing subprocesses, add new subprocesses to the workflow, and remove existing subprocesses from the workflow, depending on the output of certain subprocesses. For example, a subprocess in a workflow instance created using FireWorks [85] returns an "FWAction" upon its completion, which can store data, pass data to the next subprocess, add new subprocesses etc. This feature is particularly important for thermal conductivity computational workflows since the output structure from the relaxation subprocess determines the number and the input parameters of new static subprocesses to be added.

2.4.2. Automation and Transparency

Most commonly, computations are executed on high-performance computer (HPC) clusters or other platforms with greater computing power instead of personal devices. Operations required in the life cycle of a computational workflow, including file preparation, job submission to queue managers, and output retrieval and parsing, are repetitive and tedious when executed manually on a remote platform. Many utilities have been developed with Application

Programming Interfaces (APIs) to automate and simplify these remote management tasks. Pymatgen (Python Materials Genomics) developed by Materials Project provides APIs for input files generation and output files parsing accustomed to various quantum calculation engines [86]. FireWorks provides adapters to various queueing managers including but not limited to SLURM, PBS, and Oracle Grid Engine to allocate resources and execute workflows [85]. Specifically for Vienna *ab initio* simulation package (VASP) [87], the first-principle simulation software used in this work, a variety of toolkits have been developed to assist in preparing input files and parsing output files, such as VASPKIT [88] and qvasp [89].

Another feature to improve the degree of automation of a workflow is a high-level interface that frees researchers from learning new codes and allows them to focus on the research objectives. When designing a workflow with a high-level interface, it is important to discuss the transparency of the interface [90]. By definition, in a software with a transparent interface, the details of the implementation are invisible (transparent) to users, and the number of input parameters required from the user is minimized. Most internal parameters either have default values or are determined automatically by the workflow. Oftentimes, such workflow with a highly transparent interface can be described as a generic turnkey solution. In the case of a workflow for computational materials science with a transparent interface, to analyze the properties of a material system, the user only needs to specify the initial lattice structure of the material system and the type of calculations to run. In some cases, users can set a general degree of precision without specifying the numeric values for the computational parameters. A workflow with a transparent interface can easily be used by non-expert users with minimal coding background, after going through simple tutorials or examples. Despite the simplicity and convenience, having a transparent interface may not always be the optimal solution. In a workflow, some input parameters of subprocesses cannot be

automatically determined, especially when they are dependent on how and where the workflow is executed [90]. For example, when the subprocesses are executed on a HPC platform, a workflow with a highly transparent interface is unable to easily determine domain-specific settings, which depend on the environment of the HPC platform, the type of queueing manager, and the type of quantum engine. Such high degree of automation binds a workflow to the specific environment on where it is developed. Another drawback of a highly transparent interface is that it does not allow expert users to modify or override the inputs. Even if all the inputs could be automatically determined, an expert user with knowledge beyond the "auto-input algorithm" may still want to override some of them. For example, instead of simply predicting a material property, an expert user may want to conduct convergence tests regarding one of the hidden parameters in order to improve the default input set.

In contrast, an interface with less transparency is described as being opaque. In a fully opaque interface, all the input parameters of subprocesses are exposed to the user. With such exposure, users are able to customize and override each and every input of any subprocesses in a workflow. It allows users to inspect and modify input parameters before and during the execution of a workflow, no matter if they can be automatically determined. However, a workflow with an opaque interface is usually not friendly to non-expert users, as non-expert users can be overwhelmed by a large number of input parameters.

A balanced solution is to design workflow with an optionally transparent interface. To achieve this goal, a workflow must: 1) expose all the input parameters of its subprocesses so that they can be overridden when needed, and 2) specify reasonable default values for most input parameters so that users can interact with a simple and straightforward interface that requires minimal amount of input parameters. While non-expert users enjoy the simplicity of a transparent

interface, expert users can still override the default values of input parameters as of an opaque interface. Hubert et al. have proposed a further improved solution that is to use an input generator [90], which is a function that generates the full set of input parameters required by a workflow based on a minimal set of essential parameters. This approach enables expert users to inspect and even modify the full set of input parameters before feeding them into the workflow.

2.4.3. Reusability and Reproducibility

As previously mentionedan important characteristic of a workflow that is often desired is its reusability, which represents the potential of using the workflow in other tasks of similar nature design, workflows can be built with existing lower-level workflows and subprocesses of existing workflows.

Designing workflows with great reusability can significantly reduce repetitive works, and hence improve the efficiency of materials exploration. One necessary design principle to achieve high reusability is modularity (in contrast to an "all-in-one" solution). If a subprocess in a workflow is frequently used by other types of workflows, then it should be coded separately from the main workflow as a module. With atomized subprocess modules as building blocks, developers can then easily construct workflows for different purposes by importing and connecting relevant modules. The modular design also significantly reduces the length of code required for workflows, improves readability, and helps achieve a high-level interface. Additionally, the modular design of open-source workflows enables developers to contribute independently to the repository. With clear and comprehensive documentation, users can easily share new modules, and incorporate modules developed by others in their own workflows. For example, based on existing components in Atomate library [80], higher-level workflows have been developed for X-ray absorption spectra calculation [91], ferroelectrics exploration [92], and vacancy formation energy calculation [93].

In order to reproduce computational results, it is necessary for a workflow to store the information of input parameters along with many other computational configurations, that is, to record data provenance. Data provenance means that a workflow should store not only the input parameters and output results of the workflow itself, but also the intermediate inputs and outputs of its subprocesses [90]. For example, when the lattice thermal conductivity of a material is calculated from an un-optimized unit cell, the optimized unit cell structure and displaced supercell structures should be accessible to the user. This makes the computational result both reproducible and intelligible. However, complete data provenance is not always necessary, especially for complex workflows that produce large provenance graphs. In practice, users should be able to investigate data provenance at different granularity levels, i.e., scoped provenance. On the occasion where ultimate reproducibility is required, one could store the reference virtual machines of the environment where workflows are executed [94].

2.4.4. Data Management

Manually running computational procedures involves uploading, downloading files to/from remote directories, and parsing output files, which are labor-intensive and prone to error. In order to ensure reproducibility and reduce the error due to manual operation, the coupling of workflow automation and data storage must be tight. For the high-throughput computational exploration of materials, it is exceptionally important to maintain the quality, accessibility, and reproducibility of a large amount of data.

To couple automated computational workflow and data storage, it is critical to incorporate database solutions into the workflow infrastructure. Effective database solution enables sharing of data among collaborators to avoid running repetitive computational workflows, improves the efficiency of data analysis, and accelerates the exploration of materials.

When selecting a database to couple with an automated workflow, one should carefully choose the type of database. There are mainly 2 types of databases: relational and non-relational.

Relational database, also known as Structured Query Language (SQL) database, stores data in tables where rows are the entries and columns are the attributes of entries. For decades, relational databases have been the most widely used type of database. A strict schema is required in advance to create a relational database. The schema of a relational database ensures data consistency. Each table in a relational database will have a column with unique values, which is called a primary key. When the primary key of one table is referenced by a column in another table, the column in the second table is called a foreign key. The creation of a foreign key establishes and enforces a link between two tables. This primary key and foreign key combination ensures data accuracy by preventing duplicate information. However, the rigidity of the database schema makes it very complex to modify the structure of data, and hence results in poor flexibility. Additionally, the querying operation in relational database either requires the user to have knowledge of the schema format and SQL syntax, or requires the developer to implement wrapper libraries for simplification.

Non-relational database, also known as NoSQL (Not only SQL) database, can be further divided into multiple sub-categories including but not limited to document database, key-value database, and graph database. Document database, which is the most commonly used non-relational database, stores data in documents in JavaScript Object Notation (JSON) format. JSON document is very similar to a dictionary object in Python, which contains key-value pairs of data. Non-relational database requires no strict schema, making it highly flexible and easier to design. Moreover, the querying syntax of non-relational database is easier than that of relational database for non-expert researchers, who often have more experience with Python than SQL.

Ultimately, when an automated computational workflow is coupled with a well-designed database solution and standardized data format, it is made possible to construct centralized public databases. Some examples are: Materials Project [95], ICSD [96], and TE Design Lab [82].

2.4.5. Workflow for TBC Exploration

Taking the advantage of automated computational workflow frameworks, toolkits have been developed to explore and study materials in various application fields. For example, in the exploration of topological insulators, an automated workflow developed based on AFLOW [97] framework was used to identify 28 potential topological insulators, some hardly discoverable without the high-through method, from the entire aflowlib.org repository [98]. In the exploration of 2D materials, an automated workflow developed based on AiiDA [94] framework was used to identify 56 easily exfoliable magnetic materials from 108,423 unique, experimentally known 3D compounds [99]. Furthermore, a preset workflow provided by Atomate [80] framework was used to identify 126 new ferroelectric materials from over 67,000 candidate materials [92].

The exploration of TBC materials involves the investigation of many material properties including thermal conductivity, thermal expansion coefficient, and relevant mechanical properties. To efficiently select promising TBC material candidates from a large pool of potential material systems, it is critical to develop automated workflows to predict each of the properties. In this work, we emphasize the development of an automated workflow for the prediction of lattice thermal conductivity, which is the most critical property of top coat materials.

In the community of computational materials science, open-sourced software packages have been developed for the prediction of various material properties. For first-principles lattice thermal conductivity calculations, some of the most widely used software packages are Phonopy [100], Phono3py [101], and ShengBTE [102]. In recent years, there are toolkits developed utilizing machine

learning (without first-principle calculations) to predict lattice thermal conductivities such as hiPhive [103,104]. Usually, these lattice thermal conductivity specialized utilities provide APIs for generations of displaced supercell configurations, and for extraction of required information from first-principle calculation output files. However, the degree of automation provided by these APIs is still insufficient for HT screening of top coat material candidates, thus advanced frameworks are required to connect them together.

In recent years, many frameworks and utilities have also been developed to facilitate automated computational workflows. Pymatgen [86] provides utilities to generate sets of VASP input files and to parse and plot phonon dispersion and density of states (DOS). FireWorks [85] provides a robust framework to generate, monitor, and control the computational workflow. Atomate [80], built on top of FireWorks, while implementing a higher-level interface to manage workflows, provides "standard" workflow presets (e.g. structural relaxation, static calculation, band structure calculation, elastic constant calculation, Gibbs free energy calculation) which can be used as building blocks for new workflows. Some other frameworks such as AiiDA [94], MPInterfaces [105], ASE [106], and AFLOWπ [107] also have the potential to be used to develop automated workflows.

Specifically for the computations of lattice thermal conductivity using the finite displacement method (which will be discussed in Chapter 3.2.4), the major portion of subprocesses should be generated based on the output of other subprocesses. Therefore, the framework adopted for such calculations must be able to create dynamic workflows. Additionally, the dynamical workflow also enables automated screening of materials using complex conditions, which could greatly reduce the computational cost in HT screening projects by removing disqualified candidate systems with from the pool.

Despite that efforts have been made to develop first-principles workflows for automated lattice thermal conductivity computations [108,109], there is no open-sourced software package that automates the full process of the exploration of TBC materials at the time of writing. Most studies in this field were still conducted with low degree of automation, thus raising the demand for further automation which this work aims to fulfill.

Chapter 3: Methodology

Traditionally, the development of TBC materials requires large number of trial-and-error experiments. These experimental methods usually involve of preparation of specimens and characterization of microstructures, thermal and mechanical properties. Such methodology is very time consuming and requires careful operation by qualified personnel, hence its limited scalability. In addition, experimental methods provide little knowledge on the underlying mechanisms which lead to TBC materials properties.

With the growing computational capability, it is made possible to explore new TBC materials using computer simulations. As have been demonstrated by many studies [110–113], first-principles density functional theory (DFT) calculations combining anharmonic lattice dynamics have been shown to be an effective way to evaluate thermal conductivities of various material systems with reasonable accuracy. This computational route, with its ability to resolve atomic details and reveal fundamental physical origins, also makes it possible to investigate the mechanisms and factors controlling lattice thermal conductivities, and thus to develop physics-based guidelines for future TBC materials design.

3.1. First-Principles Density Functional Theory Calculations

First-principles or *ab initio* method, since its birth in the 1990s, has become one of the most important numerical modelling techniques at atomistic scale. As the names imply, only the most fundamental properties (physical constants), instead of (or with least) empirical assumptions, are required for the calculations. This approach evaluates electronic-structure using quantum mechanical principles, and thus predicts the interaction of atoms and electrons in a system, which ultimately derives the macroscopic properties of interest. [114]

First-principles method is all about solving the many-body Schrödinger equation (SE), which in the time-independent, non-relativistic, Born-Oppenheimer approximation is expressed as:

$$\hat{H}\psi(r_1, r_2, \dots, r_N) = E\psi(r_1, r_2, \dots, r_N)\,, \tag{2}$$

where the Hamiltonian operator $\hat{H}$ consists of three terms: the kinetic energy $\hat{T}$, the electron-electron interaction $\hat{U}$, and the external potential $\hat{V}$. In computational materials study, the external potential is merely the interaction between electrons and nuclei. These three terms are expressed as:

$$\hat{T} = -\frac{1}{2}\sum_{i}^{N_e} \nabla_i^2\,, \tag{3}$$

$$\hat{U} = \sum_{i<j}^{N_e} \frac{1}{|r_i - r_j|}\,, \tag{4}$$

$$\hat{V} = -\sum_{i}^{N_e}\sum_{k}^{N_{at}} \frac{Z_k}{|r_i - R_k|}\,, \tag{5}$$

where N_e represents the number of electrons, N_{at} represents the number of atoms, r_i represents the spatial coordinate of electron i, and Z_k represents the charge on nucleus k at location R_k.

In order to obtain the ground state configuration, one need to solve the SE by iteratively altering the wave function ψ to find the configuration that minimize the total energy E. The exact solution to SE is usually not obtainable except for a few extremely simple systems such as hydrogen atom. Approximation methods including Hartree-Fock (HF) [115], perturbation theory [116], and configuration interaction [117], all using wave function to describe electrons, are too computationally expensive to be employed for large systems. Currently the most important and most widely used first-principles method is density functional theory (DFT). Using DFT, current

high-performance computing (HPC) centers are able to perform static calculations for systems with up to 1,000,000 atoms, and dynamic simulations for systems with up to 10,000 atoms [118].

The key concept of DFT is to use the electron density instead of the electron wave function as the descriptor of electronic structure of many-body systems. Electron density $\rho(r)$ is defined as the probability density of finding an electron in a small volume around certain spatial position. It is obtained by an integral over all but one spatial coordinate, described by the following equation:

$$\rho(r) = N \int_V |\psi(\mathrm{r}_1, \mathrm{s}_1, \ldots, \mathrm{r}_\mathrm{N}, \mathrm{s}_\mathrm{N})|^2 ds_1 dr_2 ds_2 \ldots dr_N ds_N \,, \tag{6}$$

where $|\psi(\mathrm{r}_1, \mathrm{s}_1, \ldots, \mathrm{r}_\mathrm{N}, \mathrm{s}_\mathrm{N})|^2$ is the probability density of finding the system with position coordinates between r_1 and $r_1 + dr_1$, …, r_N and $r_N + dr_N$, and spin coordinates equal to $s_1, \ldots, s_N$. Electron density has following properties: it is non-negative at any position; it is 0 at infinite distance; the integral over all space yields the total number of electrons. Unlike wave function, electron density is measurable by experiment. Remarkably, some measurable properties, especially those that have classical interpretation, can be easily expressed in terms of electron density.

In 1964, Hohenberg and Kohn proposed two theorems alongside of their proofs as the foundation of DFT [119]. The first theorem states that the external potential $v(r)$ is determined by the ground state electron density $\rho(r)$. Since the exact ground state WF and everything about the ground state system can be found knowing $v(r)$ and the number of electrons N, we can now describe the system completely by specifying the electron density.

The proof is as following: Suppose an electron density $\rho(r)$ leads to two distinct external potentials $v(r)$ and $v'(r)$, and thus two different Hamiltonians $\hat{H}$ and $\hat{H}'$, with two different exact wave functions $|\psi\rangle$ and $|\psi'\rangle$. According to the variational principle:

$$E = \langle\psi|\hat{H}|\psi\rangle \leq \langle\psi'|\hat{H}|\psi'\rangle = \langle\psi'|\hat{H}'|\psi'\rangle + \langle\psi'|\hat{H} - \hat{H}'|\psi'\rangle$$
$$= E' + \int [v(r) - v'(r)]\rho(r)dr\,. \tag{7}$$

And:

$$E' = \langle\psi'|\hat{H}|\psi'\rangle \leq \langle\psi|\hat{H}'|\psi\rangle = \langle\psi|\hat{H}|\psi\rangle - \langle\psi|\hat{H} - \hat{H}'|\psi\rangle$$
$$= E' - \int [v(r) - v'(r)]\rho(r)dr\,. \tag{8}$$

Adding eq. 7 and eq. 8 gives:

$$E + E' \leq E + E', \tag{9}$$

which is contradictory. Thus, it is impossible for there to exist two distinct external potentials given a single electron density.

The second theorem states that while N and $v(r)$ are fixed, for any trial density $\tilde{\rho}(r)$ satisfying $\tilde{\rho}(r) \geq 0$ and $\int \tilde{\rho}(r)dr = N$,

$$E \leq E_v[\tilde{\rho}]\,, \tag{10}$$

where E is the exact energy and $E_v[\tilde{\rho}]$ is the energy functional associated with the trial density. This theorem allows minimization of the total energy of a system by varying the ground state electron density under the aforementioned constraints. It is also known as the Hohenberg-Kohn density variational principle, similar to the wave function counterpart.

Since $\tilde{\rho}(r)$ determines the associated wave function $|\tilde{\psi}\rangle$, we can proof the second theorem by variational principle:

$$E_v(\rho) = E \leq \langle\tilde{\psi}|\hat{H}|\tilde{\psi}\rangle = E_v[\tilde{\rho}]\,. \tag{11}$$

Soon enough, in 1965, Kohn and Sham proposed an approach to express the energy functional by decomposing into known contributions from the independent electrons approximation and an unknown nonclassical term [120]:

$$E[\rho] = V_{ext}[\rho] + T_{KS}[\phi] + V_H[\rho] + E_{XC}[\rho]\,, \tag{12}$$

where V_{ext}, T_{KS}, V_H are external potential, kinetic energy, and Hartree energy, respectively, with the following expressions:

$$V_{ext}[\rho] = \int \rho(r) V_n(r) dr\,, \tag{13}$$

$$T_{KS}[\phi] = -\sum_i \int \phi_i^*(r) \frac{\nabla^2}{2} \phi_i(r) dr\,, \tag{14}$$

$$V_H[\rho] = \frac{1}{2} \int \int \frac{\rho(r)\rho(r')}{|r - r'|} dr dr'\ . \tag{15}$$

The unknown nonclassical term E_{XC} is referred to as the exchange-correlation term, which does not have an explicit expression. Many exchange-correlation functionals have been developed in attempt to provide decent approximation for DFT calculations, including local density approximation (LDA) [121], generalized gradient approximation (GGA) [122], meta-GGA [123], hybrid meta-GGA [124] etc.

To solve the ground state electron density ρ, an iterative procedure called self-consistent field (SCF) procedure is used. In SCF procedure, an initial guess of electron density is used to solve the Kohn-Sham equation, and a more accurate output electron density can be calculated. The output electron density is then used as the input of the next iteration. Once desired convergence is achieved, the iteration is concluded. The physical properties of the system can thus be calculated based on the converged ground state electron density.

3.2. Computational Background for Thermal Properties

3.2.1. Thermal Conductivity Overview

In crystalline solids, heat energy is transported mainly via 3 mechanisms: electrons, lattice waves (phonons), and electromagnetic waves (photons). The contribution to thermal conductivity

through any type of carrier is highly dependent on the mean free path of the carrier between collisions [125]. In metallic materials, electrons are delocalized and have relatively long mean free path, thus are the dominant thermal transportation mechanism. In dielectrics like most oxide ceramics, heat energy is transported almost entirely via lattice waves since electrons in dielectrics are highly localized and have negligible mean free path [125]. Since TBC materials are mostly ceramics, this work focuses only on the thermal conductivity contributed by lattice waves, which is frequently referred to as lattice thermal conductivity.

3.2.2. Harmonic and Anharmonic Lattice Dynamics

Consider a lattice structure which has atoms with small displacements near their equilibrium positions. The total potential energy of such lattice structure (E) can be expressed as the following Taylor series:

$$E = E_0 + \sum_{i;\alpha} \frac{\partial E}{\partial u_i^\alpha} u_i^\alpha + \frac{1}{2!} \sum_{i,j;\alpha,\beta} \frac{\partial^2 E}{\partial u_i^\alpha \partial u_j^\beta} u_i^\alpha u_j^\beta + \frac{1}{3!} \sum_{i,j,k;\alpha,\beta,\gamma} \frac{\partial^3 E}{\partial u_i^\alpha \partial u_j^\beta \partial u_k^\gamma} u_i^\alpha u_j^\beta u_k^\gamma + \cdots, \tag{16}$$

where E_0 is the total potential energy at equilibrium configuration, and u_i^α is the displacement of atom i from its equilibrium position in direction α. The first constant term (E_0) can be ignored by adjusting the reference level of potential energy. The second term (1st order term) on the right-hand side of eq. 16 is the first partial derivative of potential energy, which also represents the force exerted on an atom by all the others. Since the potential energy is minimized at the equilibrium configuration, its first partial derivative is 0, thus the second term can be ignored [126].

From the remaining 2nd order and third-order terms, the 2nd order "harmonic" and the 3rd order "anharmonic" interatomic force constants (IFCs) are defined as following:

$$\phi_{ij}^{\alpha\beta} = \frac{\partial^2 E}{\partial u_i^\alpha \partial u_j^\beta}, \tag{17}$$

$$\phi_{ijk}^{\alpha\beta\gamma} = \frac{\partial^3 E}{\partial u_i^\alpha \partial u_j^\beta \partial u_k^\gamma}. \tag{18}$$

Under harmonic approximation, only up to the 2nd order IFCs are considered to describe small displacements of atoms. It is a relatively simple model where interatomic forces are described with a spring analogy, and there is no interaction between phonons. With harmonic approximation, many physical properties including heat capacity, Gibbs energy, and phonon dispersion can be evaluated.

However, without interaction between different phonon modes, all the phonons have infinite mean free path which results in infinite thermal conductivity of the system. In addition, other phenomena such as thermal expansion and phase transition cannot be explained within harmonic approximation [125]. By considering the third-order anharmonic IFCs to describe anharmonic interactions, the aforementioned phenomena can be well explained.

3.2.3. Three-Phonon Scattering

The resistance mechanisms of phonon transportation include phonon-phonon interaction, phonon-defect scattering, phonon-boundary scattering, phonon-electron scattering. Among these mechanisms, phonon-phonon interaction has the most significant impact on thermal conductivity in a defect-free lattice system. Under anharmonic lattice dynamic approximation, interactions between different phonon modes result in limited phonon lifetime and mean free path. Three-phonon scattering, among all types of phonon-phonon interactions, is the most commonly occurred process, hence has the greatest impact on the lattice thermal conductivity of dielectric materials.

In a three-phonon scattering process, either one phonon splits into two other phonons, or two phonons combines into one phonon. In general, three-phonon scattering processes can be described by the following equations:

$$\hbar\omega_1 + \hbar\omega_2 = \hbar\omega_3\,, \tag{19}$$

$$\vec{q}_1 + \vec{q}_2 = \vec{q}_3 + \vec{G}\,, \tag{20}$$

where ω_i and $\vec{q}_i$ are frequency and wave vector of phonon i, respectively, and $\vec{G}$ is a reciprocal lattice vector. Eq. 19 represents the conservation of energy in the process. Eq. 20 however, represents a conditional conservation of crystal momentum. Based on the expressions above, three-phonon scattering processes can be further categorized into Normal process (N-process) where $\vec{G} = 0$, and Umklapp process (U-process) otherwise, as illustrated in Figure 2.

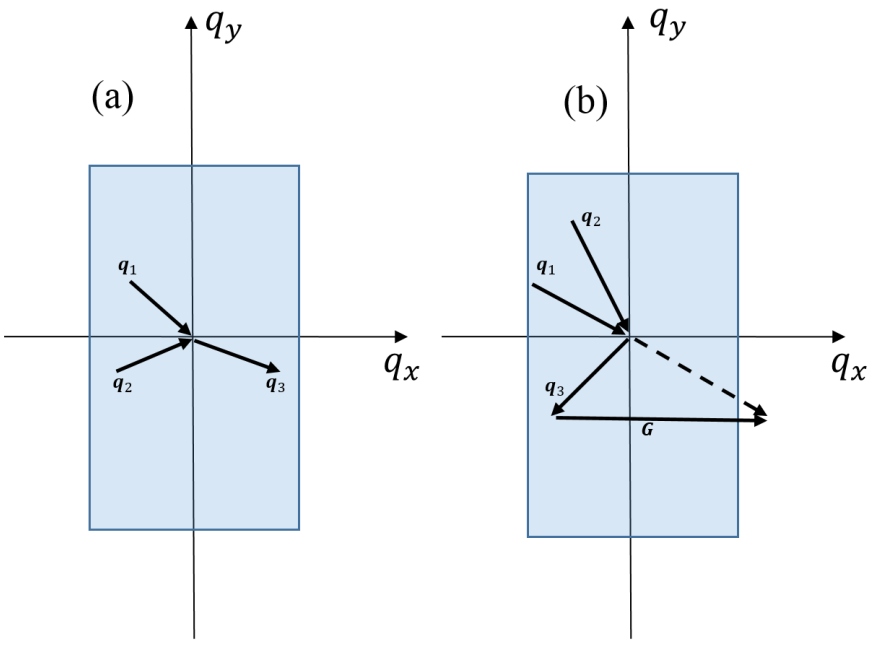

Figure 2: Schematics of (a) N-process and (b) U-process [110]

U-process occurs when the wave vectors of two incident phonons add up to a scattered wave vector that is outside of the first Brillouin zone, and the scattering process effectively

produces a phonon with reversed group velocity by subtracting the reciprocal lattice vector $\vec{G}$. U-process results in a reversed heat flux, and therefore has significant thermal resistive effect. N-process on the other hand, as a momentum-conserving process, has little effect on the heat flux. Remarkably, crystal momentum is called "quasimomentum" since it is only conserved to within an additive constant.

3.2.4. Finite Displacement Method

The most commonly used computational method to evaluate 2nd-order and 3rd order IFCs is supercell based, finite displacement method. From eq. 17, we can express the 2nd order IFCs approximated as:

$$\phi_{ij}^{\alpha\beta} = \frac{\partial^2 E}{\partial u_i^\alpha \partial u_j^\beta} = -\frac{\partial F_j^\beta}{\partial u_i^\alpha} \approx -\frac{F_j^\beta(u_i^\alpha = h) - F_j^\beta}{h}, \tag{21}$$

where F_j^β represents the force exerted on atom j in direction β, while atom i is displaced by a finite displacement h from equilibrium position in the direction α [101]. The force F_j^β is usually zero at equilibrium position.

Similarly, from eq. 18, for the 3rd order IFCs:

$$\begin{aligned}\phi_{ijk}^{\alpha\beta\gamma} = \frac{\partial^3 E}{\partial u_i^\alpha \partial u_j^\beta \partial u_k^\gamma} &\approx \frac{1}{2h}\left[\frac{\partial^2 E}{\partial u_j^\beta \partial u_k^\gamma}(u_i^\alpha = h) - \frac{\partial^2 E}{\partial u_j^\beta \partial u_k^\gamma}(u_i^\alpha = -h)\right] \\ &\approx \frac{1}{4h^2}\Big[-F_k^\gamma\left(u_i^\alpha = h, u_j^\beta = h\right) + F_k^\gamma\left(u_i^\alpha = h, u_j^\beta = -h\right) \\ &\quad + F_k^\gamma\left(u_i^\alpha = -h, u_j^\beta = h\right) - F_k^\gamma\left(u_i^\alpha = -h, u_j^\beta = -h\right)\Big],\end{aligned} \tag{22}$$

where in this case atom i and j are displaced by a finite displacement h from equilibrium position in the direction α and β respectively [102]. According to eq. 22, we can deduce that each $\phi_{ijk}^{\alpha\beta\gamma}$ element requires four DFT calculations with different supercell configurations. Generally, for a

unit cell with n atoms and a supercell with N unit cells, $4 \times 9n^2N$ DFT calculations are required, which is clearly impractical and not scalable [102]. Fortunately, the total number of required DFT calculations can be reduced considering system symmetries. From eq. 17 and eq. 22, changing the order of differentiation does not affect the result of these partial derivatives. Thus, we have the following permutation symmetries:

$$\phi_{ij}^{\alpha\beta} = \phi_{ji}^{\beta\alpha} \tag{23}$$

$$\phi_{ijk}^{\alpha\beta\gamma} = \phi_{ikj}^{\alpha\gamma\beta} = \phi_{kji}^{\gamma\beta\alpha} = \cdots. \tag{24}$$

Also, considering a general space-group symmetry operation:

$$\sum_{\alpha} T^{\alpha'\alpha} R_i^{\alpha} + b^{\alpha'} = R_{T_{b(i)}}^{\alpha'}, \tag{25}$$

where T is the point-group operator and b is the translation operator, and $T_{b(i)}$ indicates the atom to which the ith atom is mapped under the given operation. The 3rd order IFCs therefore satisfy the following relation:

$$\phi_{T_{b(i)}T_{b(j)}T_{b(k)}}^{\alpha'\beta'\gamma'} = \sum_{\alpha\beta\gamma} T^{\alpha'\alpha} T^{\beta'\beta} T^{\gamma'\gamma} \phi_{ijk}^{\alpha\beta\gamma}. \tag{26}$$

Additionally, by introducing a cut-off radius, we can further reduce the number of atomic triplets considered. Usually, the value of cut-off radius needs to be determined through convergence test based on either the force constants or the result thermal conductivity.

3.2.5. Calculation of Lattice Thermal Conductivity

In this work, lattice thermal conductivity is computed by solving linearized Boltzmann transport equation (LBTE) under the single mode relaxation time (SMRT) approximation. The lattice thermal conductivity tensor can be expressed as:

$$\kappa_{\alpha\beta} = \frac{1}{NV_0} \sum_{\lambda} C_{\lambda} v_{\lambda}^{\alpha} v_{\lambda}^{\beta} \tau_{\lambda}, \tag{27}$$

where α, β are the Cartesian indices, λ is the abbreviation of phonon mode with band index s and wave vector q, N is the number of unit cells in the system, V_0 is the volume of a unit cell, C_λ is the mode dependent heat capacity, v_λ^α and τ_λ are the group velocity and SMRT of the phonon mode λ respectively.

The mode dependent heat capacity is expressed as:

$$C_\lambda = \sum_\lambda k_B \left(\frac{\hbar\omega_\lambda}{k_B T}\right)^2 \frac{\exp\left(\frac{\hbar\omega_\lambda}{k_B T}\right)}{\left(\exp\left(\frac{\hbar\omega_\lambda}{k_B T}\right)-1\right)^2}. \tag{28}$$

From the 2nd order IFCs, we can compute the harmonic frequency ω_{qs} and the polarization vector $W(k, qs)$ of phonon mode λ by solving eigenvalue problem of a dynamical matrix $D(q)$:

$$\sum_{k'\beta} D_{\alpha\beta}\,(kk', q) W_\beta(k, qs) = \omega_{qs}^2 W_\alpha(k, qs)\,, \tag{29}$$

with

$$D_{\alpha\beta}(kk', q) = \frac{1}{\sqrt{m_k m_{k'}}} \sum_{j'} \Phi_{0k,j'k'}^{\alpha\beta}\; e^{iq[r(j'k')-r(0k)]}\,. \tag{30}$$

Here a slightly different notation is used to label atoms where $r(jk)$ represents the position of the k-th atom in the j-th unit cell, and m_k is the atomic mass of atom type k.

Based on the eigenvalue equation, the group velocity v_λ^α of phonon mode λ in direction α can then be calculated by:

$$\begin{aligned} v_\lambda^\alpha &= \frac{\partial\omega_\lambda}{\partial q_\alpha} \\ &= \frac{1}{2\omega_\lambda} \sum_{kk'\beta\gamma} W_\beta(k,\lambda) \frac{\partial D_{\beta\gamma}(kk', q)}{\partial q_\alpha} W_\gamma(k',\lambda)\,. \end{aligned} \tag{31}$$

From the 3rd order IFCs, we can compute the imaginary part of the phonon self-energy using:

$$\begin{aligned}\Gamma_{\lambda}(\omega) = \frac{18\pi}{\hbar}\sum_{\lambda'\lambda''}|\Phi_{-\lambda\lambda'\lambda''}|^2 \\ \cdot\{(n_{\lambda'}+n_{\lambda''}+1)\delta(\omega-\omega_{\lambda'}-\omega_{\lambda''}) \\ +(n_{\lambda'}-n_{\lambda''})[\delta(\omega+\omega_{\lambda'}-\omega_{\lambda''})-\delta(\omega-\omega_{\lambda'}+\omega_{\lambda''})]\},\end{aligned} \tag{32}$$

where $\Phi_{-\lambda\lambda'\lambda''}$ is the strength of 3-phonon interaction among phonons with mode λ, λ', and λ'' given by:

$$\begin{aligned}\Phi_{-\lambda\lambda'\lambda''} = \frac{1}{\sqrt{N}}\frac{1}{3!}\sum_{kk'k''}\sum_{\alpha\beta\gamma}W_{\alpha}(k,\lambda)W_{\beta}(k',\lambda')\,W_{\gamma}(k'',\lambda'')\sqrt{\frac{\hbar}{2m_k\omega_{\lambda}}}\sqrt{\frac{\hbar}{2m_{k'}\omega_{\lambda'}}}\sqrt{\frac{\hbar}{2m_{k''}\omega_{\lambda''}}} \\ \times\sum_{jj'}\Phi^{\alpha\beta\gamma}_{0k,j'k',j''k''}\,e^{iq'[r(j'k')-r(0k)]}\,e^{iq''[r(j''k'')-r(0k)]}\,e^{i(q+q'+q'')r(0k)}\,\Delta(q+q'+q''),\end{aligned} \tag{33}$$

where $\Delta(q+q'+q'')$ equals to 1 if $(q+q'+q'')$ is a reciprocal lattice vector, otherwise equals to 0.

Finally, we can obtain the phonon relaxation time, which is the last piece of puzzle for lattice thermal conductivity calculation, given by:

$$\tau_{\lambda} = \frac{1}{2\Gamma_{\lambda}(\omega_{\lambda})}. \tag{34}$$

In summary, with phonon group velocity and heat capacity from 2nd order IFCs, and phonon relaxation time from 3rd order IFCs, we may compute the lattice thermal conductivity of any material system at given temperature.

3.3. Workflow for Lattice Thermal Conductivity Computation

As indicated in Section 3.2.4, calculating lattice thermal conductivities using the finite displacement method requires a large number of static DFT calculations of displaced supercell configurations. The calculation process also involves multiple types of computations including structural relaxation DFT calculation, static DFT calculation, and phonon-related calculations. Employing an automated computational workflow framework allows the linking of different types of computations, and hence greatly simplifies the execution and management of such exhaustive computational subprocesses. Additionally, since both the number of the displaced configurations and the supercell structure of each configuration depends on the relaxed unit cell structure, the static DFT calculation subprocesses should be added to the workflow upon the completion of the relaxation subprocess. To achieve this, the workflow framework must have the ability to change the workflow structure during runtime, i.e., to create dynamic workflows.

In a computational workflow for lattice thermal conductivity, the unit cell structure is first relaxed. The output of the relaxation subprocess (the relaxed unit cell structure) shall then be used to create the corresponding supercell configurations and static calculation subprocesses. The output of static calculation subprocesses shall be parsed to obtain an organized dataset of interatomic forces, which will finally be used to calculate the lattice thermal conductivity.

Chapter 4: Ph3pyWF Tool Package

4.1. Overview

The computational route to lattice thermal conductivity involves a large quantity of subtasks. Running and managing such quantity of subtasks is very complicated, especially when handling errors. Traditionally, to run a complete thermal properties analysis using VASP [87], researchers have to perform many operations manually and monitor the simulation progress. First, input files for the structural relaxation VASP job need to be prepared manually before job submission to SLURM. After unpredictable computation time, the output files are collected and used to generate input files for subsequent static VASP jobs via Phonopy/Phono3py. The second step generates hundreds or even thousands of static jobs and should be placed in designated directories for post-analysis. Each of these jobs is submitted individually to SLURM. In the case of job failure, it is even more complicated to locate the job and identify the cause of failure. Therefore, automated execution and management of the computational workflow are required.

An open-source workflow python package with name “Ph3pyWF” has been developed in this work to fulfill such demand of automation. It is designed to conduct high-throughput thermal conductivity computations with a user-friendly interface and progress monitor. The software also allows simple and robust error identification and recovery.

4.2. Workflow Architecture

The high-level architecture of Ph3pyWF package is developed based on Atomate [80] and FireWorks [85] computational workflow framework. From a bottom-up aspect of view, Ph3pyWF contains three levels of components: Firetask, Firework, and Workflow.

Firetasks are the most fundamental building blocks of a Workflow. A Firetask in essence is a script that performs specific operations, such as writing/reading files, running VASP simulation, interacting with database, etc. A Firework contains a series of Firetask(s) that will be executed in sequence. Specification of a Firework passes input parameters to the Firetask(s) contained in the Firework. A Workflow is a set of Firework(s) connected with certain dependencies. For example, if Firework B requires output from Firework A as input parameters, Firework A will be the "parent" of Firework B. Firework B is marked with the state "WAITING" and "READY" before and after completion of Firework A respectively.

In addition, Firetasks can return "FWAction" which performs modification of the Workflow specifications, addition of new Fireworks, cancellation of remaining Fireworks, etc., depending on the output. This feature allows users to dynamically control the workflow.

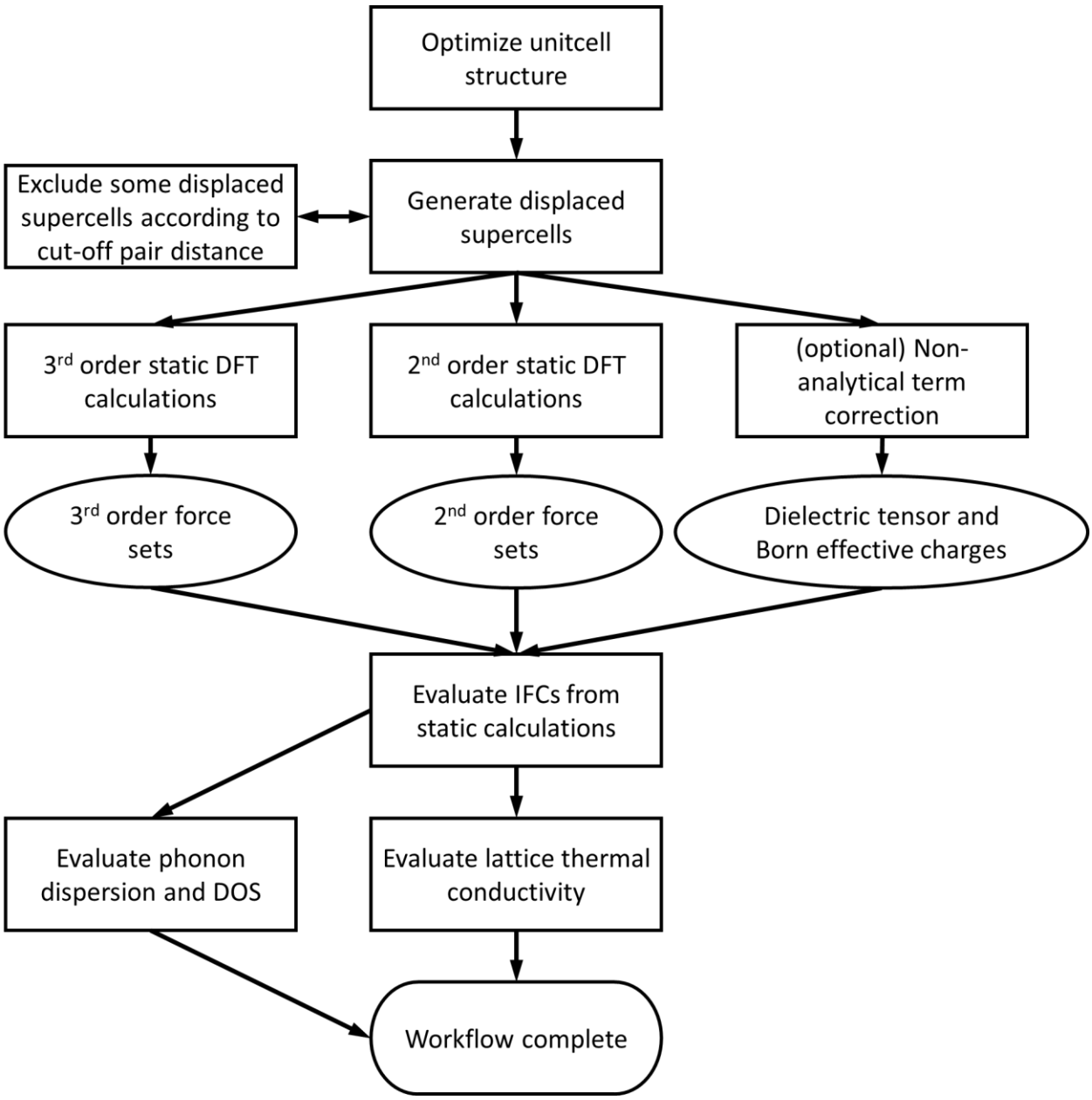

Figure 3: Outline of Ph3pyWF architecture

Figure 3 illustrates the Fireworks and their dependencies in a Workflow object of Ph3pyWF. The first task is to obtain the optimized unit cell structure of the material system of interest. Preset Firework for structural optimization is available in Atomate library, and is adopted for this task with slight modification to the default parameters. This Firework prepares VASP input files according to the given structure and parameters, runs VASP simulation, and parses output files to extract useful results.

With the optimized unit cell structure, displaced supercell structures can then be generated using the finite displacement method. A new Firetask (`Ph3pyAdderTask`) has been developed for this purpose. When creating a new workflow, a Firework is created containing only this Firetask. In this Firework, the optimized unit cell structure is fetched from the database by querying for the document containing the result of the optimization Firework. Phono3py is then used to generate a list of displaced supercell structures. Users can specify different supercell sizes for the 2nd and 3rd order IFCs calculation. By default, the 2nd order IFCs are calculated from a subset of displaced supercell configurations for the 3rd order IFCs. During this Firework, Phono3py writes a file (`disp_fc3.yaml`) containing the information of displaced supercell structures, which is later required by the post-analysis Firework. This file is parsed as a dictionary object to be conveniently stored in the database.

For each displaced supercell structure in the list, a corresponding static VASP job is appended to a list of Fireworks. After the completion of the second Firework (generate displaced supercells), static Fireworks are inserted into the Workflow by an FWAction. They are inserted using mode “detour” to ensure that all of the inserted Fireworks are parents of the post-analysis Firework. These Fireworks are slightly modified preset static calculation Fireworks. Most simulation parameters, except those related to the static job type, are identical to that of the optimization Firework.

In the case where different supercell sizes for the 2nd and 3rd order IFCs calculation are specified, additional Fireworks for the 2nd order IFCs are also inserted into the Workflow by the aforementioned FWAction. Fireworks for the 2nd IFCs have names formatted differently than that for the 3rd order IFCs to provide not only better readability for users, but also more robust data retrieval for post-analysis Firework.

After all the static Fireworks are completed, thermal conductivity can be calculated. The post-analysis Firework fetches the following data from the database: optimized unit cell structure, list and information of displaced supercells generated, interatomic forces, and other required information depending on the specifications. With these data fetched, the Firework then uses Phono3py to calculate IFCs and thermal conductivity at a given range of temperatures. Phonopy is also used in this Firework to calculate phonon DOS and dispersion band structure. Phono3py post-analysis generates several large output files. The key results (temperatures, lattice thermal conductivities, phonon DOS and phonon band structure) are stored in the database as dictionary objects. Optionally, large output files, which contain information on phonon group velocities, and phonon frequency, can be compressed and stored in the database in binary format.

4.3. Typical Usage

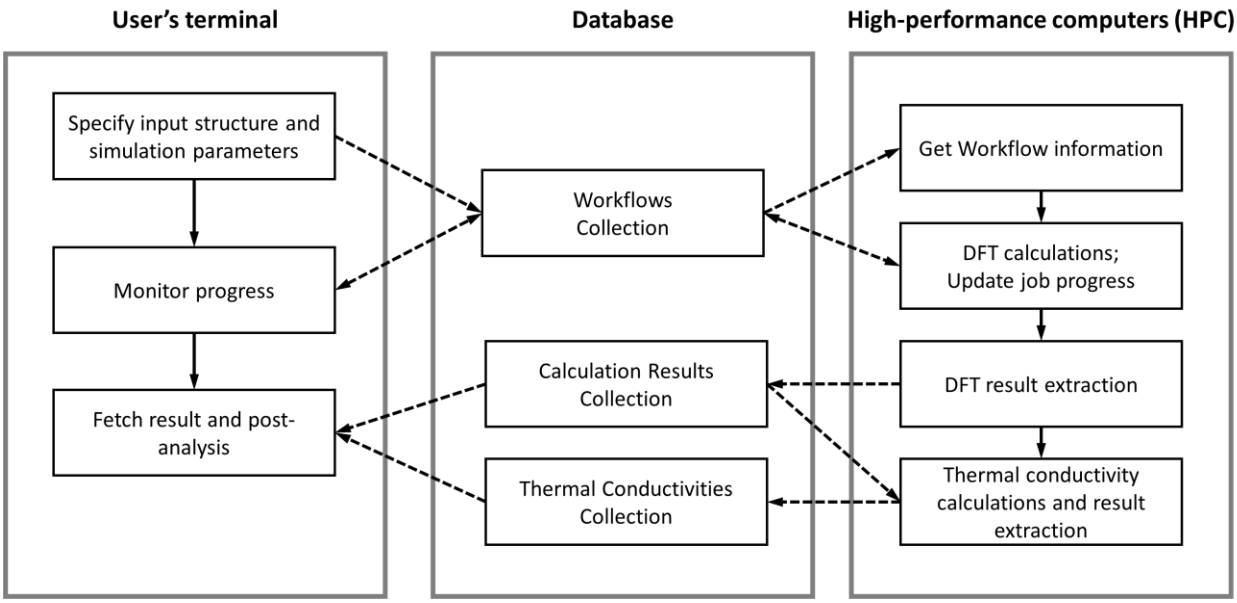

Figure 4: Schematic of operation procedure using Ph3pyWF. Solid arrows denote order of execution. Dashed arrows denote sending and receiving of data.

A typical operation procedure using Ph3pyWF is shown in Figure 4. There are three places where events take place, the user's terminal, database, and high-performance computers (HPC). The user's terminal can be any machine that runs Python and has internet access. It is the main

interface for the user to create and manage workflows. For DFT computations, high-performance computers (HPC) are required to provide sufficient computational resources. In this work, the HPC resource used was Cedar cluster provided by Compute Canada. HPC conducts all the computational subprocesses and stores temporary input and output files. Due to HPC policy, these files are stored in a "scratch" storage and are purged periodically unless manually transferred to a permanent storage location. Most commonly, the user needs to connect to the HPC to submit computation jobs and allocate computing resources. With other software packages, it is also possible to access HPC via the user's terminal. The database is used to store all the workflow information, intermediate computation results, and final thermal conductivity results. It provides permanent storage of organized data for more convenient queries and analyses. MongoDB is used in this work as the database.

Before creating the very first workflow instance, domain-specific environment parameters need to be configured on the user's terminal and HPC. For simplicity, this work will not discuss the details of the configuration files other than their purpose since they are available in the documentation of respective libraries.

For Atomate [80] and Fireworks [85] environment, five configuration files are required: `db.json`, `my_fworker.yaml`, `my_launchpad.yaml`, `my_qadaptor.yaml`, and `FW_config.yaml`. The file `db.json` provides information including server address and credential to connect to the MongoDB coupled with the workflow for storing and parsing of output (other collections in the MongoDB are also accessible with redirecting). The file `my_fworker.yaml` provides the parameters that control which subprocesses are executed on this machine, which is useful when multiple machines (or clusters) are available, as well as the environmental parameters including the path to VASP executable. The file `my_launchpad.yaml` provides database information similar to that in `db.json`, but for querying

and updating the status of Fireworks and Workflows. The file `my_qadaptor.yaml` stores the type of queueing manager used in the environment and the configuration options for computational resource allocation, such as number of CPU cores, number of nodes, and total wall time. Finally, the file `FW_config.yaml` indicates the path to the aforementioned 4 configuration files. Commonly, multiple folders, each contains aforementioned 4 configuration files with different parameters, can be created. The user can then apply different sets of configurations by only changing the path parameter in `FW_config.yaml`. Since no intensive computation is conducted on user's terminal, only `db.json` and `FW_config.yaml` should be set, while all 5 files should be set on the HPC.

In addition, a file with filename `.pmgrc.yaml` in home directory should be set for Pymatgen [86] configurations. On HPC, pseudopotentials should be stored in specific directory structure, and the location should be specified in this file. When running VASP computation jobs, POTCAR files are created based on pseudopotential files found at location provided by this file. Optionally, Materials Project API key can also be specified in this file for more convenient interaction with the open database provided by Materials Project API.

New Ph3pyWF workflow instances are created on the user's terminal. JupyterLab is used in this work for a more interactive coding environment. The user needs to provide the input unit cell structure in Pymatgen format. This can be done by manually writing the POSCAR file and convert to Pymatgen format or extract from online crystal structure databases such as Materials Project. Most parameters can be determined automatically in Ph3pyWF, but the user can also specify certain parameters if needed. A more detailed description of all the parameters can be found in the following section. The workflow is then uploaded to the database.

On HPC, instead of submitting individual computation jobs, generic jobs are submitted for all the workflows. When the resource is allocated, the software fetches information on workflows

from the database, and starts running Fireworks with “READY” state. Each submitted job can run one Firework at one time but can run multiple Fireworks throughout the time allocated.

The progress of the workflow is continuously updated in the database as the jobs run. In the case of job failure, the failed Fireworks are marked as “FIZZLED” and can be easily identified in the database. The error message and launch directory are also included in the database for debugging. Usually, the error is not caused by critical mistakes and the user only needs to rerun the Firework by resetting the Firework state to “READY”. In Ph3pyWF, failed static VASP jobs do not affect other static jobs in the same workflow, nor any other Fireworks in different workflows. Auxiliary utilities have been developed to monitor workflow progress and rerun failed Fireworks. In the case where a Firework failed multiple times, warning messages will be raised to indicate that further investigation is required.

Upon completion of VASP calculations, output files are automatically parsed, and the key results will be uploaded to specific sectors (“ph3py_task” collection) in the database.

4.4. Computational Parameters

The modular design of this workflow software allows expert users to customize their input parameter at different levels of granularity.

Table 1: Input parameters of Ph3pyWF at workflow level

Parameter	Type	Default value
`structure`	pymatgen.Structure	<N/A>
`skip_relax`	Boolean	`False`
`name`	String	“phono3py wf”
`c`	Dictionary	`None`

Table 1 lists the parameters at the top-most level, when instantiating a new workflow object. The only required input parameter without default value is the input unit cell structure. The input

unit cell structure should be a Pymatgen Structure object, which can be generated by parsing structure files including but not limited to POSCAR, CONTCAR, Crystallographic Information File (CIF), or fetched from online database. If `skip_relax` is set to True, the workflow will skip the structural relaxation and use the input unit cell structure directly to generate displaced supercells. User can change the suffix of workflow name by specifying the parameter `name`. The workflow name will be the concatenation of the chemical formula of the input structure and the name specified by user. The last parameter `c` is a dictionary object containing more detailed calculation settings which will be explained in the following paragraphs. By default, `c` does not contain any data, and any setting specified by the user will override the default calculation settings.

Table 2: Supported parameters in dictionary object c to modify the calculation settings of subprocesses

Parameter	Type	Default value
`tag`	String	<Automatically generated>
`supercell_size_fc3`	Tuple	`(2,2,2)`
`supercell_size_fc2`	Tuple	<Same as supercell_size_fc3>
`cutoff_pair_distance`	Float	`None`
`atom_disp`	Float	`0.03`
`vasp_input_set_relax`	VaspInputSet	`Ph3pyRelaxSet`
`vasp_input_set_static`	VaspInputSet	`Ph3pyStaticSet`
`db_file`	String	`">>db_file<<"`
`metadata`	Dictionary	`{}`
`USER_*_SETTINGS`	Dictionary	`{}`
`t_min`	Float	`200`
`t_max`	Float	`1401`
`t_step`	Float	`50`
`primitive_matrix`	ndarray	`None`
`mesh`	List	`[11,11,11]`
`is_nac`	Boolean	`True`
`is_symmetry`	Boolean	`True`
`symprec`	Float	`1e-5`

Table 2 lists the supported parameters in the aforementioned dictionary object `c`. These are the parameters for Firework level subprocesses.

The parameter `tag` is a unique string for identification of a workflow and any connected workflows. Without user specification, the value of `tag` is automatically generated depending on the time when the workflow is instantiated (e.g., `"2021-09-21-14-32-07-656382"`). Such default generated value is prone to collision, where the same `tag` is assigned to multiple irrelevant workflows, but when new workflows are instantiated at low frequency it is more readable for users. For high-throughput computation, it is recommended to use universally unique identifier (UUID) as `tag` to avoid collision. The Fireworks in a workflow have names containing this `tag` string. This `tag` is commonly used to query and update Fireworks in a workflow and can be used as a parameter to instantiate auxiliary workflows that are connected to the main workflow.

The parameter `supercell_size_fc3` is a 3-element tuple that specifies the supercell dimension for 3rd order IFCs. Supercell is generated by elongating along lattice axes of unit cell. The default value is (2,2,2), in which case a $2 \times 2 \times 2$ supercell is created. The parameter `supercell_size_fc2` specifies the supercell dimension for 2nd order IFCs and is optional. By default, 2nd order IFCs calculation uses the same supercell dimension as the 3rd order IFCs supercell dimension. User may specify larger and different supercell dimension for 2nd order IFCs since two-atom interaction have longer range in real space than three-atom interaction. If `supercell_size_fc2` is specified, additional displaced supercell structures are generated, and the corresponding static Fireworks are added to the workflow. It is recommended to set the supercell dimension according to the unit cell structure. A “rule of thumb” is to ensure that the supercell have lengths more than 9 Angstrom along each lattice orientation.

The parameter `cutoff_pair_distance` specifies the cutoff pair distance in Angstrom in supercells to reduce the number of static Fireworks. With finite displacement method, two of the three atoms in an interaction triplet are displaced. If the distance between the two displaced atoms in a displaced supercell is larger than the specified cutoff pair distance, such supercell configuration is excluded from the IFCs calculations, and no corresponding static Firework is added.

The parameter `atom_disp` specifies the magnitude of atomic displacement in Angstrom for finite displacement method. For VASP calculations, the suggested default value is 0.03 Angstrom.

The parameters `vasp_input_set_relax` and `vasp_input_set_static` specify the VASP calculation settings for structural relaxation and static force calculations respectively. The default input sets are included in Ph3pyWF library, and their settings are discussed in the later section. One may use different VASP input sets imported from external sources or customized by local files.

The parameter `db_file` specifies the path to the `db.json` file which contains the database credentials. By default, the value is a non-path formatted string "`>>db_file<<`", with which Ph3pyWF will check local configuration files for the path upon execution.

The parameter `metadata` is used to store user defined metadata such as descriptions of the workflow. This parameter has no effect on the calculation.

`USER_*_SETTINGS` represents a set of VASP input parameters, including `USER_INCAR_SETTINGS`, `USER_INCAR_SETTINGS_STATIC`, `USER_POTCAR_SETTINGS`, `USER_POTCAR_FUNCTIONAL`, `USER_KPOINTS_SETTINGS`, and `USER_KPOINTS_SETTINGS_STATIC`, to override the default settings specified by `vasp_input_set_relax` and `vasp_input_set_static`. The

detailed description of each setting can be found in Pymatgen documentation. POTCAR settings are universal and are applied to all the Fireworks in a workflow.

The parameters `t_min`, `t_max`, and `t_step` specify the temperatures in Kelvin at which lattice thermal conductivities are evaluated. The default values of `t_min`, `t_max`, and `t_step` generate list temperature values: 200K, 250K, …, 1350K, 1400K. Note that `t_min` is included in the list of temperature values while `t_max` is excluded.

The parameter `primitive_matrix` is a 3×3 array that specifies the transformation matrix from non-primitive cell to primitive cell. When primitive unit cell is used as input structure, this parameter is unnecessary to be specified.

The parameter `mesh` is a 3-element list that specifies the q-point mesh sampling grid used for thermal conductivity calculation. The default q-point mesh is 11 points along each axis. For supercell with large number of atoms, $11 \times 11 \times 11$ q-point mesh may result in long computing time exceeding the allocated wall time of a single job, thus it is recommended to specify a less dense q-point mesh initially and run a q-point convergence analysis (discussed in later sections) afterward.

The parameter `is_nac` specifies whether or not to apply non-analytical term correction for harmonic phonons. If `is_nac` is set to `True`, an additional Firework is added to the workflow. The added Firework run a modified static VASP calculation to obtain static dielectric tensor and the Born effective charges, which will be used by the final phonon properties calculation.

The parameters `is_symmetry` and `symprec` specify if crystal symmetry is used and the tolerance used to find crystal symmetry respectively. When the value of `symprec` is too small, exceeding large number of displaced structures may be generated.

4.5. First-Principles Calculation Settings

Table 3: Default VASP parameters for structural relaxation calculations

Parameter	Default value
ALGO	Normal
EDIFF_PER_ATOM	1e-9
EDIFFG	-1e-5
ENCUT	520
IBRION	2
ISIF	3
ISMEAR	0
ISPIN	2
LASPH	True
LDAU	True
LREAL	AUTO
NELM	100
NSW	300
PREC	Accurate
SIGMA	0.01
KPOINTS	reciprocal_density = 64
POTCAR_FUNCTIONAL	PBE_54

All the first-principles calculations in this work were performed using the Vienna *ab initio* simulation package (VASP) [87] with projected augmented wave (PAW) potential. Table 3 lists the default VASP parameters for structural relaxation calculations. The default exchange correlation potential is implemented using Perdew–Burke–Ernzerhof (PBE) generalized gradient approximation (GGA) [122]. User may also apply local-density approximation (LDA) [121] by modifying `USER_INCAR_SETTINGS`. The kinetic energy cutoff is set to 520 eV. The Brillouin zone is sampled using Monkhorst-Pack (MP) k-point meshes with reciprocal volume density of 64 Å^{-3}. The convergence criteria are set to $10^{-9}\ eV$ and $10^{-5}\ eV\ \text{Å}^{-1}$ for the total energy and the force respectively.

Table 4: Default VASP parameters for static calculations

Parameter	Default value
ALGO	Normal
EDIFF_PER_ATOM	1e-9
ENCUT	520
IBRION	-1
ISIF	3
ISMEAR	0
ISPIN	2
LASPH	True
LDAU	True
LREAL	AUTO
NELM	100
NSW	300
PREC	Accurate
SIGMA	0.01
KPOINTS	reciprocal_density = 32
POTCAR_FUNCTIONAL	PBE_54

Table 4 lists the default VASP parameters for static calculations. By default, static calculations use Monkhorst-Pack (MP) k-point meshes with reciprocal volume density of 32 $Å^{-3}$. Most other parameters are identical to those for structural relaxation calculations. The VASP parameters for static dielectric tensor and the Born effective charges calculations are modified based on the VASP parameters for static calculations.

4.6. Error Identification and Correction

When the computational procedure is executed and managed manually, handling subprocess failure is arguably the most frustrating work. Using the finite displacement method for thermal conductivity calculation, hundreds to thousands of subprocesses are generated. Without the help of automated workflow, these subprocesses are usually submitted to the HPC through

temporary scripts, with each subprocess occupying a single queued job. Oftentimes the user is unable to estimate the time required for each subprocess, thus the allocated wall time may be insufficient and lead to incomplete jobs. When manually running lattice thermal conductivity computations, the user may not be able to realize the presence of failed or incomplete jobs until the final step (IFCs and thermal conductivity calculation). Also, most thermal conductivity calculation software packages are unable to provide information about which particular jobs failed. In such scenario, the user has to go through the output files of all subprocesses to identify the failed jobs and the reason for failure, and then rerun those jobs.

In this work, the workflow software developed is capable of handling most errors associated with VASP input parameters, and also provides a simple interface for error identification and correction. Ph3pyWF is developed based on Atomate infrastructure which uses Custodian [127] framework to run the VASP executable. It is able to detect many common error messages during VASP calculations, then make corrections to the input parameters and rerun the calculation automatically.

In the case where a Firework fails due to an error that Custodian is unable to handle automatically, the state of failed Firework is marked as “FIZZLED” in the workflows collection in the database, and the error message is stored in the launches collection. The user can easily query the database to identify failed jobs and reasons for failure to handle correspondingly. If the stored error message provides insufficient information for correction methods, the user may check launch information in the database for the launch directory of failed Firework, and navigate to the launch directory to investigate input and output files, as well as the log file.

Ph3pyWF allows running multiple Fireworks in serial in a single queued job. The last Firework running when the allocated time limit is reached will inevitably become a “lost run”.

Such Firework will remain in “RUNNING” state forever and no error message will be stored. FireWorks library provides interfaces to detect lost runs by checking if the elapsed time from the last update of a Firework exceeds user-specified expiration time.

4.7. Progress Monitor

Ph3pyWF also implements auxiliary utilities to monitor the progress of workflows and to automatically handle failed and lost Fireworks. A script with the name `guard_ph3pywf`, which can be used by the command line, is included in the Ph3pyWF package. By specifying the `tag` of workflow(s), or the id of any Firework (`fw_id`) in workflow(s), the script will periodically check the progress of specified workflow(s), and report information including: the number of all Fireworks in a workflow, the number of running, completed, and failed Fireworks. The script will also detect lost Fireworks, and rerun failed and lost Fireworks. If any Firework fails more than three times, a warning message will be prompted to notify the user to check for unrecoverable errors.

4.8. Post-analysis using Ph3pyWF

After a Ph3pyWF workflow is completed, calculated lattice thermal conductivity, phonon dispersion band structure, and phonon density of state are stored in the database along with descriptive metadata in a relative readable format. To further automate the analysis procedure, Ph3pyWF also includes API for simple post-analysis and graph plotting. By only specifying the tag of a workflow, the post-analysis utility automatically fetches results of the specified workflow from the database. With class methods, the user may easily plot and save figures of lattice thermal conductivity, phonon dispersion band structure, and phonon density of state.

Additionally, a component of standard Ph3pyWF, `Phono3pyAnalysisToDb`, can be used alone as a workflow and launched independently to redo the final thermal conductivity calculations

for a completed workflow. The user may specify different q-point mesh and different temperature ranges in this workflow. This workflow fetches existing VASP computation results computed in the associated workflow, therefore avoiding repetitively rerunning the computationally expensive first-principles calculations. By default, this workflow overwrites the result document, and updated results can be fetched without changing the query string. Optionally the user can create copy documents with different tags, but the query string needs to be changed accordingly for each copy.

Another workflow, `wf_ph3py_get_kappa_convergence`, can also be launched independently to redo the final thermal conductivity calculations. This workflow will calculate thermal conductivities using different q-point meshes, and then evaluate the convergence (fitted) value of thermal conductivity. The detail of this workflow is discussed in section 5.1. This workflow adds new information on top of the result document instead of overwriting the initial thermal conductivity values calculated using a single q-point mesh. Similarly, this workflow overwrites the fitted result in a result document by default and is able to create copies when specified.

Chapter 5: Results and Discussion

5.1. Convergence Test

5.1.1. Supercell Size

The supercell size is one of the inputs that are recommended to be specified manually. A convergence test was conducted to investigate the effect of supercell size on the calculated lattice thermal conductivity in cubic MgO system. MgO is selected due to its simple structure, low number of atoms in the primitive cell, and high lattice symmetry, with which the computational cost is affordable even when a large supercell size is employed.

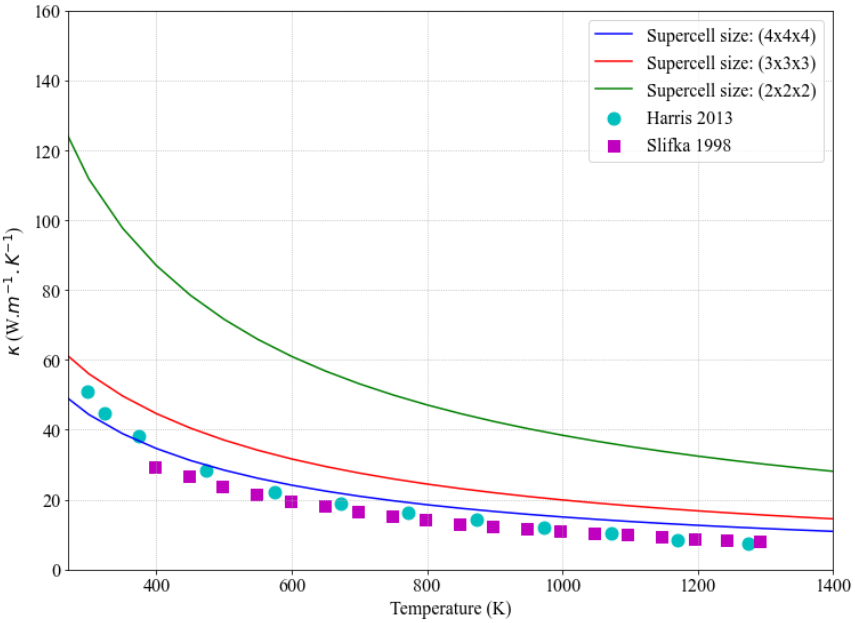

Figure 5: Effect of supercell size on the calculated lattice thermal conductivity of cubic MgO. The solid curves represent the calculated values of κ using Ph3pyWF with supercell sizes from $2 \times 2 \times 2$ to $4 \times 4 \times 4$. The filled symbols represent the experimental measured values of κ. [128,129]

Figure 5 shows the calculated lattice thermal conductivities of MgO calculated using Ph3pyWF with different supercell sizes. In all 3 workflows, the reciprocal density of k-point meshes for all first-principles calculations were all set to 128 $Å^{-3}$, and no cutoff pair distance were specified. Ph3pyWF automatically generated $4 \times 4 \times 4$, $3 \times 3 \times 3$, and $2 \times 2 \times 2$ Monkhorst-

Pack k-point meshes for the static calculations in the workflows with supercell sizes of $2 \times 2 \times 2$, $3 \times 3 \times 3$, and $4 \times 4 \times 4$ respectively. The lattice parameters of supercells in 3 cases range from 6.02 Å to 12.04 Å. The results indicate a convergence pattern with increasing supercell size. From this convergence test we adopted a guideline that the acceptable convergence can be reached with lattice parameter along each axis of the supercell greater than 10 Å. In the case where the lattice parameter along an axis of a primitive unit cell is already greater than 9 Å, no elongation will be done along that axis to keep computational cost at a reasonable level.

Table 5: Computational cost of each workflow in supercell size convergence test

Supercell size	Total CPU hours	Average computing time per job (s)
$2 \times 2 \times 2$	~11	~600
$3 \times 3 \times 3$	~30	~520
$4 \times 4 \times 4$	~1000	~8000

The computational cost of each workflow in this convergence test is listed in Table 5. It is apparent that the total computing time grows exponentially with supercell size. The longer average computing time per job of the workflow with $2 \times 2 \times 2$ supercell than that of the workflow with $3 \times 3 \times 3$ is likely caused by the k-points generated by reciprocal density, which employed meshes with more grid points in the workflow with smaller supercells.

5.1.2. Q-points Density

Studies [102,130] have suggested that the convergence of calculated lattice thermal conductivity with respect to q-point mesh dimension shows a pattern that can be fitted to exponential curve with form:

$$\kappa = \kappa_\infty \left(1 - e^{-\frac{q_i}{\epsilon}}\right), \tag{35}$$

where q_i is the number of q-points in direction i, κ_∞ and ϵ are unknown parameters to fit. The value of κ_∞ is the converged value of lattice thermal conductivity to be extrapolated, and the value of ϵ reflects the rate of convergence.

In this work, an auxiliary workflow component has been developed to perform such convergence test: `wf_ph3py_get_kappa_convergence`. This workflow shall be launched after the completion of a base workflow of Ph3pyWF since itself does not involve first-principles calculations.

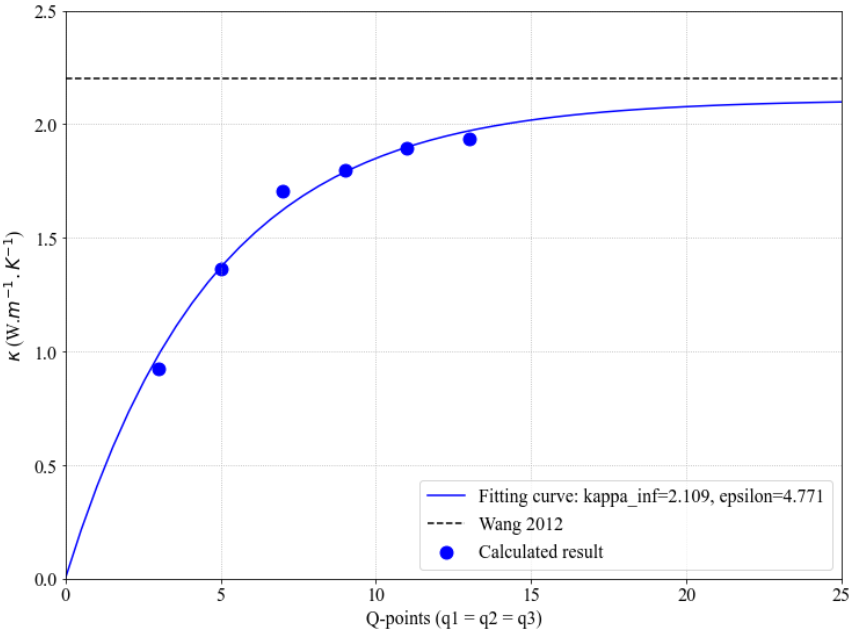

Figure 6: Effect of q-point mesh on the calculated lattice thermal conductivity of $La_2Zr_2O_7$ at 473 K. The filled symbols represent the values of κ calculated using different dimensions of q-point meshes. The solid curve represents the fitting curve of the calculated points to a fitting function $\kappa = \kappa_\infty\left(1 - e^{-q_i/\epsilon}\right)$. The dashed line represents the experimentally measured value from Wang 2012 [131].

Figure 6 shows the result obtained using `wf_ph3py_get_kappa_convergence` based on a completed Ph3pyWF for $La_2Zr_2O_7$, which has been discussed in the "Validation Results" section. The q-point meshes used have numbers of q-points range from 3 to 13 in each direction. Only meshes with odd number of q-points in each direction were used to ensure gamma-centered mesh. The extrapolated value of lattice thermal conductivity (κ_∞) is around 5% lower than the experimental value.

The calculated points also show good alignment with the fitting curve, indicating the possibility to perform calculations using low density q-point meshes and obtain final result by extrapolation. Since the computational cost grows exponentially with the density of q-point mesh, this fitting method is exceptionally cost-effective.

5.2. Validation Results

In this work, Ph3pyWF used to perform calculation for several oxide systems: ZrO_2, MgO, Al_2O_3, ZnO, BeO, $La_2Zr_2O_7$, and $Gd_2Zr_2O_7$. The capabilities of Ph3pyWF are validated by comparing the calculated lattice thermal conductivity with the experimental values.

5.2.1. ZrO_2

The lattice thermal conductivity and phonon dispersion band structure of monoclinic ZrO_2 (space group $P2_1/c$) was calculated using Ph3pyWF. The IFCs were calculated using $2 \times 2 \times 2$ supercells (96 atoms) constructed from the primitive cell (12 atoms). The phonon properties were calculated using $11 \times 11 \times 11$ q-point mesh. The cutoff pair distance was set as 4.0 Å to reduce the number of static jobs from more than 10,000 to around 2,000.

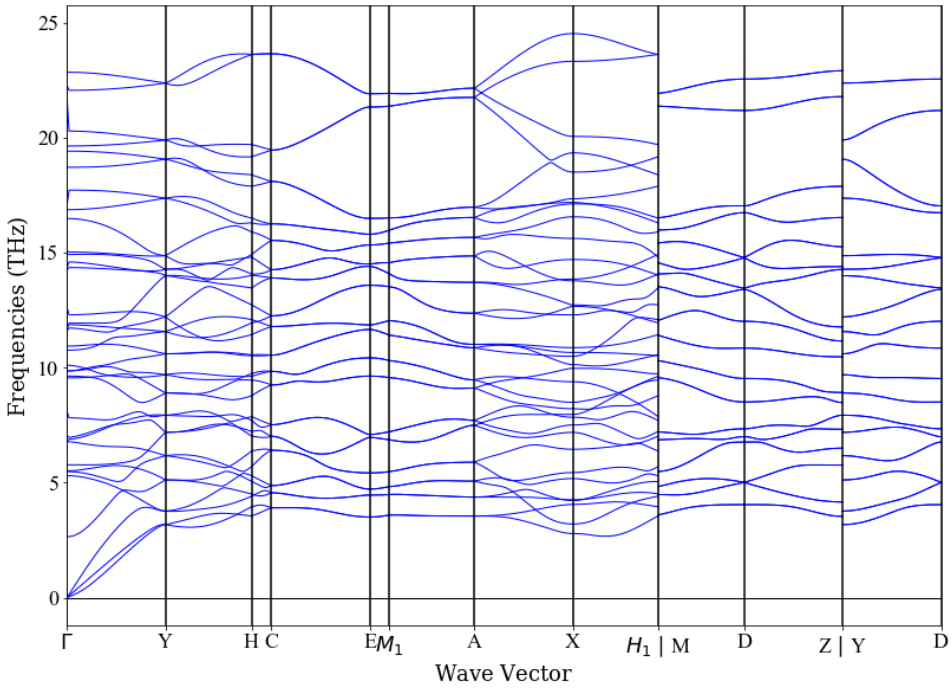

Figure 7: Phonon dispersion band structure for ZrO_2 calculated using Ph3pyWF.

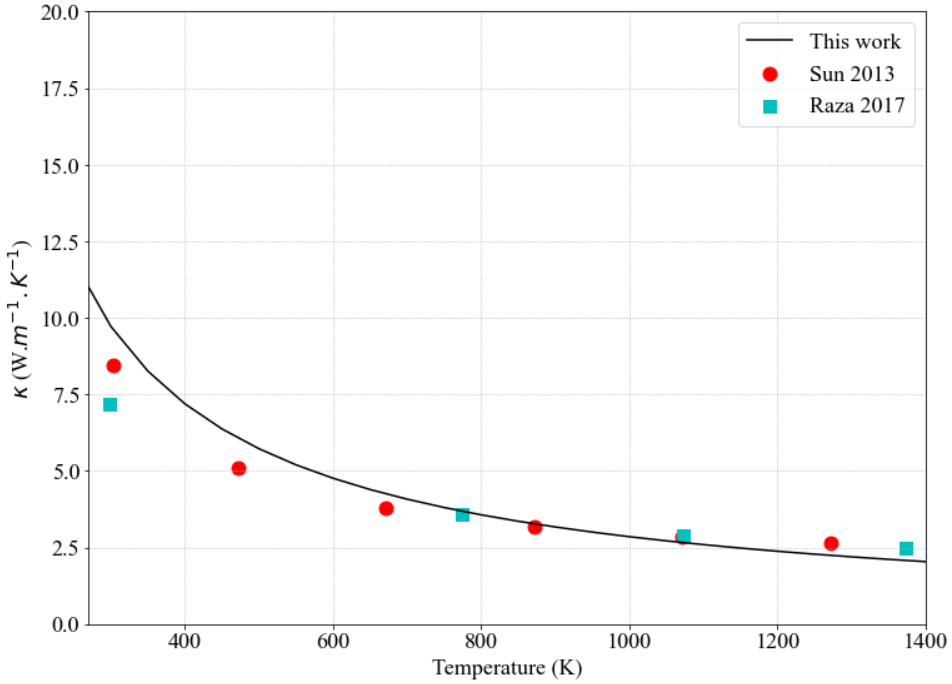

Figure 8: Lattice thermal conductivity (κ) of monoclinic ZrO_2. The solid curve represents the calculated values of κ using Ph3pyWF. The filled symbols represent the experimental measured values of κ. [132,133]

The calculated phonon dispersion band structure shown in Figure 7 shows no presence of imaginary frequency, therefore indicates the stability of this structure. Figure 8 shows the lattice

thermal conductivity of monoclinic ZrO_2 calculated using Ph3pyWF and the experimentally measured values. The calculated lattice thermal conductivity values show good agreement with the experimental values, with slight overestimation at temperatures below 800K. This workflow has 2541 subprocesses and costed approximately 400 CPU hours in total, with ~560 seconds per subprocess on average.

5.2.2. MgO

The lattice thermal conductivity and phonon dispersion band structure of cubic MgO (space group $Fm\bar{3}m$) was calculated using Ph3pyWF. The IFCs were calculated using $4 \times 4 \times 4$ supercells (128 atoms) constructed from the primitive cell (2 atoms). No cutoff pair distance was specified in this workflow. The reciprocal density of k-point meshes for structural relaxation and static calculations were all set to $32\ \text{Å}^{-3}$.

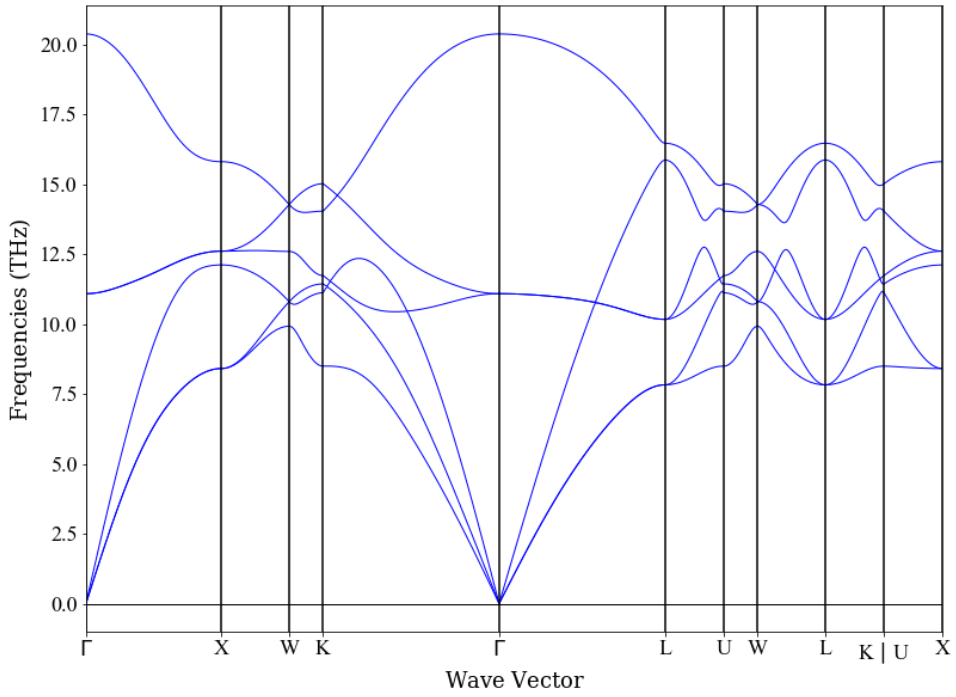

Figure 9: Phonon dispersion band structure for MgO calculated using Ph3pyWF.

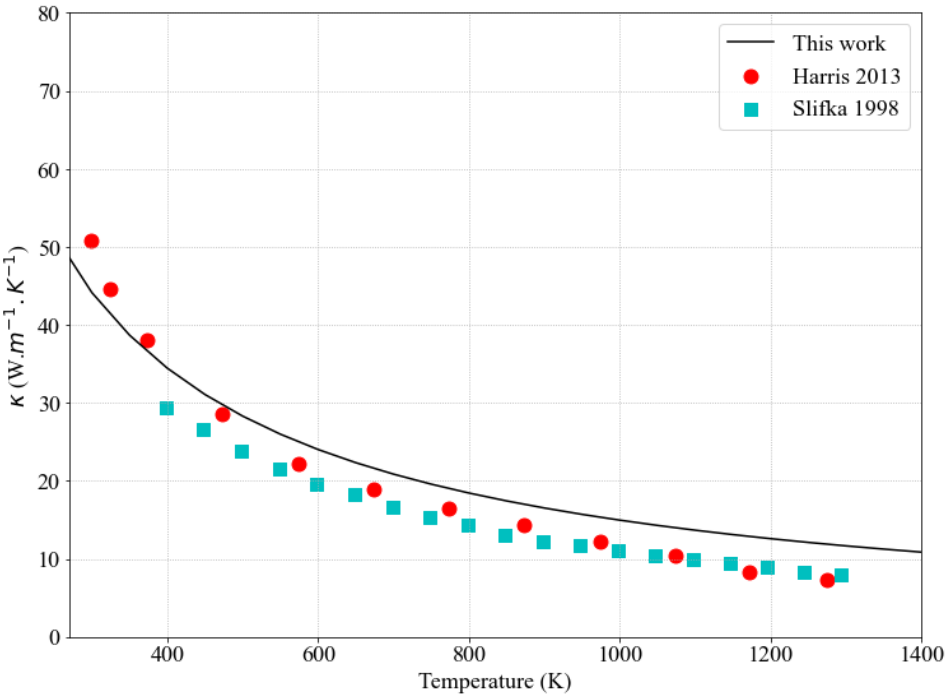

Figure 10: Lattice thermal conductivity (κ) of monoclinic MgO. The solid curve represents the calculated values of κ using Ph3pyWF. The filled symbols represent the experimental measured values of κ. [128,129]

The calculated phonon dispersion band structure shown in Figure 9 shows no presence of imaginary frequency, therefore indicates the stability of this structure. Figure 10 shows the lattice thermal conductivity of cubic MgO calculated using Ph3pyWF and the experimentally measured values. The calculated lattice thermal conductivity values show good agreement with the experimental values in general, with overestimation at temperature above 500K. This workflow has 445 subprocesses and costed approximately 100 CPU hours in total, with ~810 seconds per subprocess on average. The lower reciprocal density of k-points significantly reduced the computational cost in this workflow, comparing with the similar workflow discussed in the convergence test, without sacrificing too much computational accuracy.

5.2.3. Al_2O_3

The lattice thermal conductivity of corundum Al_2O_3 (space group $R\bar{3}c$) was calculated using Ph3pyWF. The IFCs were calculated using $3 \times 3 \times 3$ supercells (270 atoms) constructed

from the primitive cell (10 atoms). The cutoff pair distance was set as 6.0 Å to reduce the number of static jobs from more than 2400 to around 960.

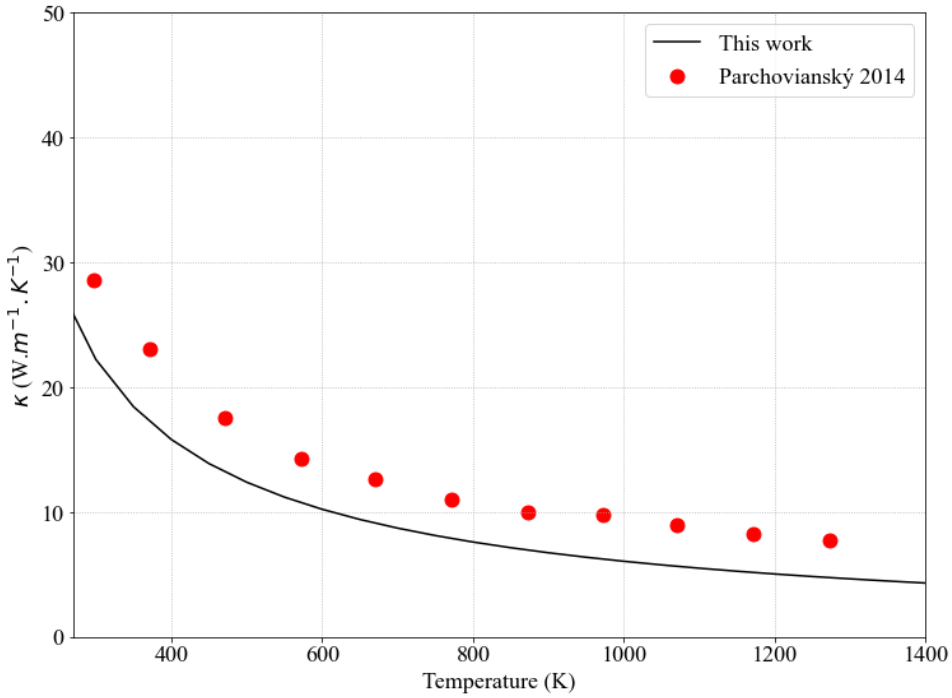

Figure 11: Lattice thermal conductivity (κ) of corundum Al_2O_3. The solid curve represents the calculated values of κ using Ph3pyWF. The filled symbols represent the experimental measured values of κ. [134]

Figure 11 shows the lattice thermal conductivity of corundum Al_2O_3 calculated using Ph3pyWF and the experimentally measured values. Despite having underestimation over all range of temperatures, the calculated values show acceptable agreement with the experimental values in terms of the general curve shape. This workflow has 962 subprocesses and costed approximately 120 CPU hours in total, with ~400 seconds per subprocess on average.

5.2.4. ZnO and BeO

Ph3pyWF has been tested on 2 wurtzite structure (space group $P6_3mc$) oxide systems: ZnO and BeO. The IFCs of ZnO and BeO were calculated using $3 \times 3 \times 2$ supercells (72 atoms) and $3 \times 3 \times 3$ supercells (108 atoms) respectively constructed from the primitive cell (4 atoms). In both workflows, the use of symmetry in VASP were switched off due to reoccurring errors.

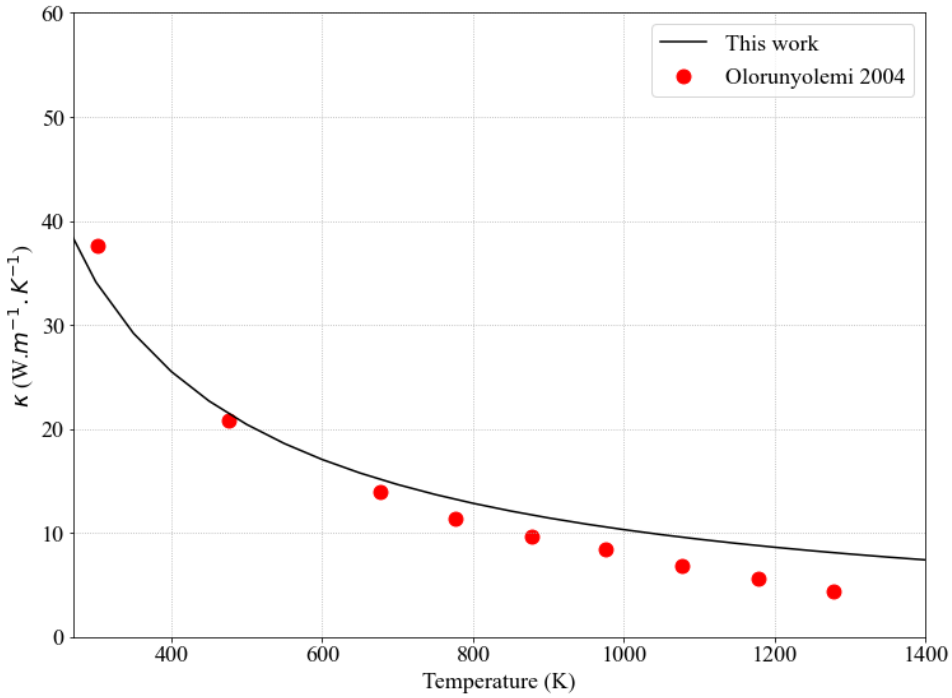

Figure 12: Lattice thermal conductivity (κ) of wurtzite ZnO. The solid curve represents the calculated values of κ using Ph3pyWF. The filled symbols represent the experimental measured values of κ. [135]

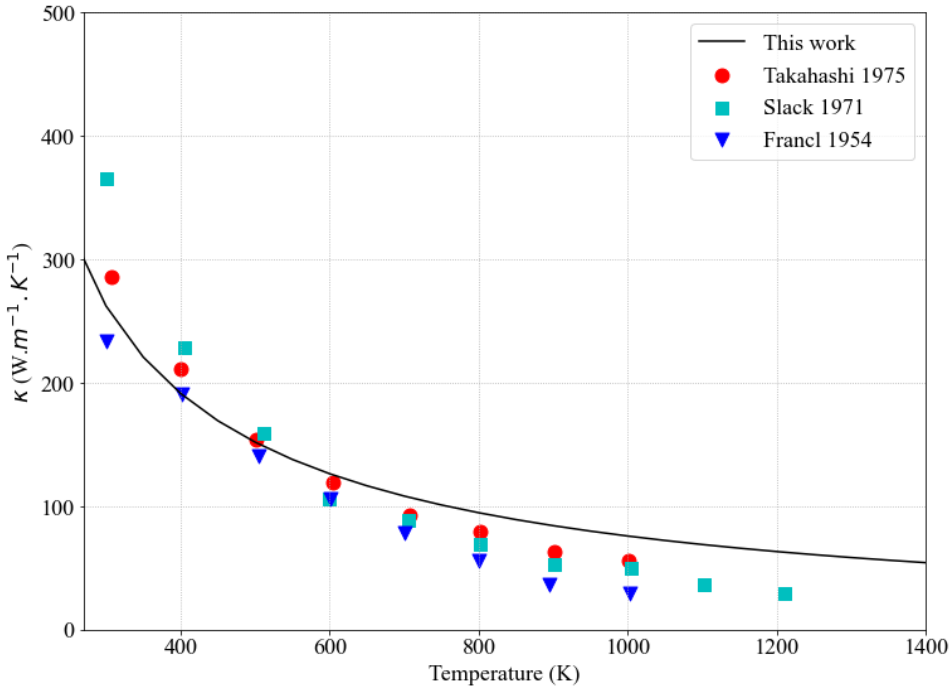

Figure 13: Lattice thermal conductivity (κ) of wurtzite BeO. The solid curve represents the calculated values of κ using Ph3pyWF. The filled symbols represent the experimental measured values of κ. [136–138]

The lattice thermal conductivities of wurtzite ZnO and BeO calculated using Ph3pyWF and their experimentally measured values are shown in Figure 12 and Figure 13 respectively. The

calculated values of both systems show good agreement with the experimental values at relatively low temperature range, and overestimation at temperature above 600 K. Nevertheless, the calculated results still verified the exceptionally high thermal conductivity of BeO. The workflow of ZnO has 1257 subprocesses and costed approximately 400 CPU hours in total, with ~1150 seconds per subprocess on average. The workflow of BeO has 2251 subprocesses and costed approximately 500 CPU hours in total, with ~820 seconds per subprocess on average.

5.2.5. $La_2Zr_2O_7$ and $Gd_2Zr_2O_7$

Once the workflow software has been validated with the aforementioned binary oxide systems, we move on to further validate Ph3pyWF on more complicated oxide systems such as pyrochlores. In this work, the lattice thermal conductivities of $La_2Zr_2O_7$ and $Gd_2Zr_2O_7$, both are promising candidate of TBC topcoat materials with pyrochlore structure (space group $Fd\bar{3}m$) and Zr occupying sites at 16*c* (0, 0, 0), were calculated using Ph3pyWF.

The IFCs of $La_2Zr_2O_7$ and $Gd_2Zr_2O_7$ were both calculated using $2 \times 2 \times 2$ supercells (176 atoms) constructed from the primitive cell (22 atoms). A cutoff pair distance of 5 Å was set in both workflows, reducing the number of static jobs from more than 3800 to less than 900. Local density approximation (LDA) was used instead of generalized gradient approximation (GGA) as exchange-correlation functional.

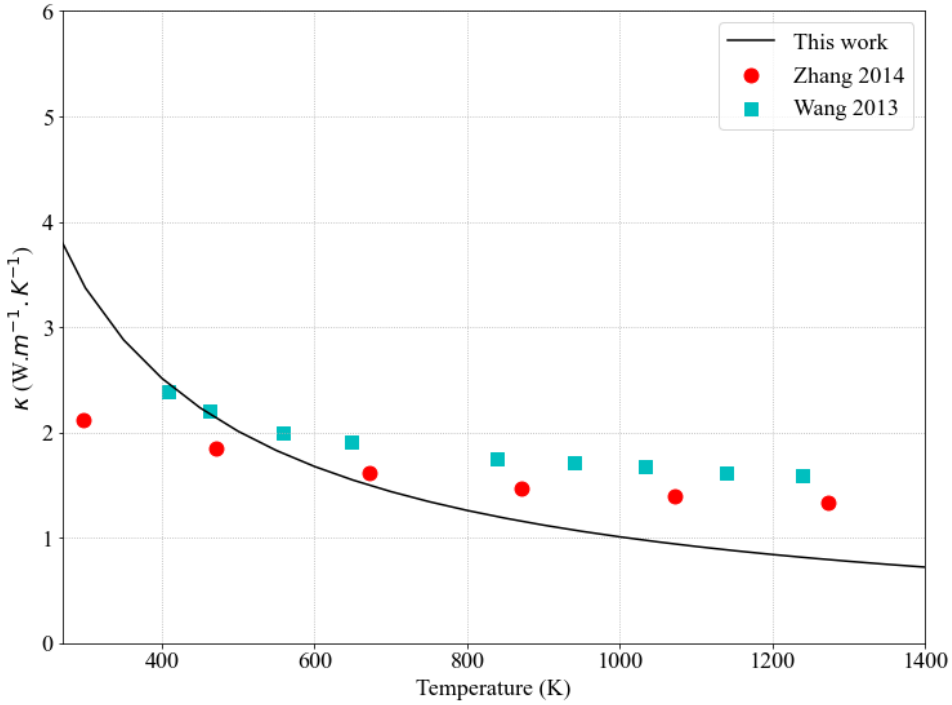

Figure 14: Lattice thermal conductivity (κ) of pyrochlore $La_2Zr_2O_7$. The solid curve represents the calculated values of κ using Ph3pyWF. The filled symbols represent the experimental measured values of κ. [139,140]

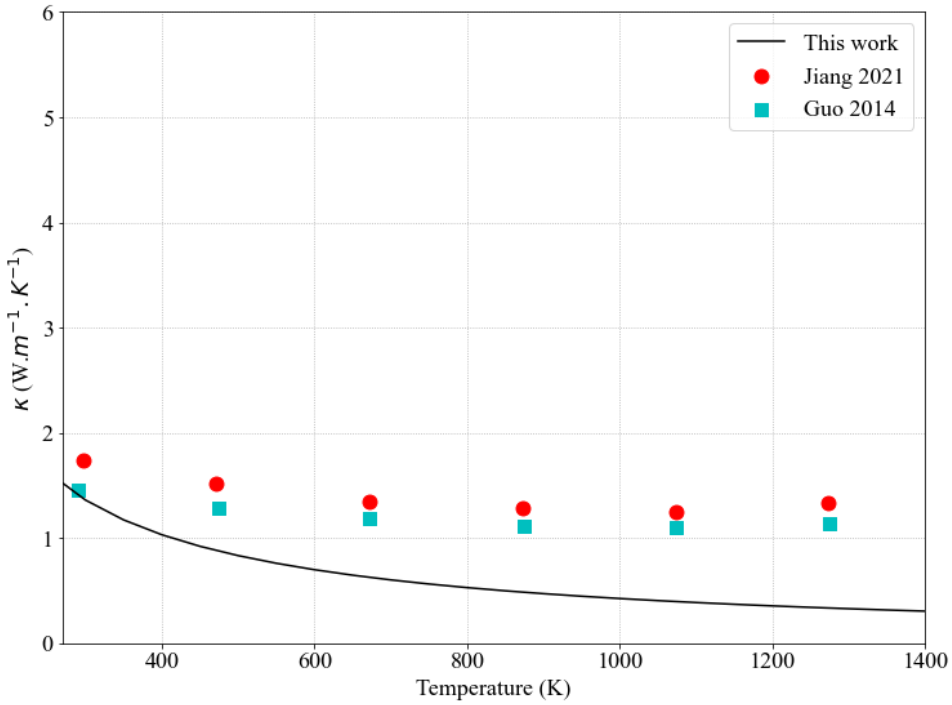

Figure 15: Lattice thermal conductivity (κ) of pyrochlore $Gd_2Zr_2O_7$. The solid curve represents the calculated values of κ using Ph3pyWF. The filled symbols represent the experimental measured values of κ. [141,142]

The lattice thermal conductivities of $La_2Zr_2O_7$ and $Gd_2Zr_2O_7$ calculated using Ph3pyWF and their experimentally measured values are shown in Figure 14 and Figure 15 respectively. The calculated results verified the characteristic low lattice thermal conductivities in both systems. For $La_2Zr_2O_7$, the calculated values show good agreement with the experimental values at relatively low temperature. Underestimation is observed in both cases, and the flattening pattern at high temperature is not reflected in the calculated lattice thermal conductivities. This can be explained by the significant thermal conductivity contribution from radiational thermal transportation, which is not involved in the calculations of this workflow.

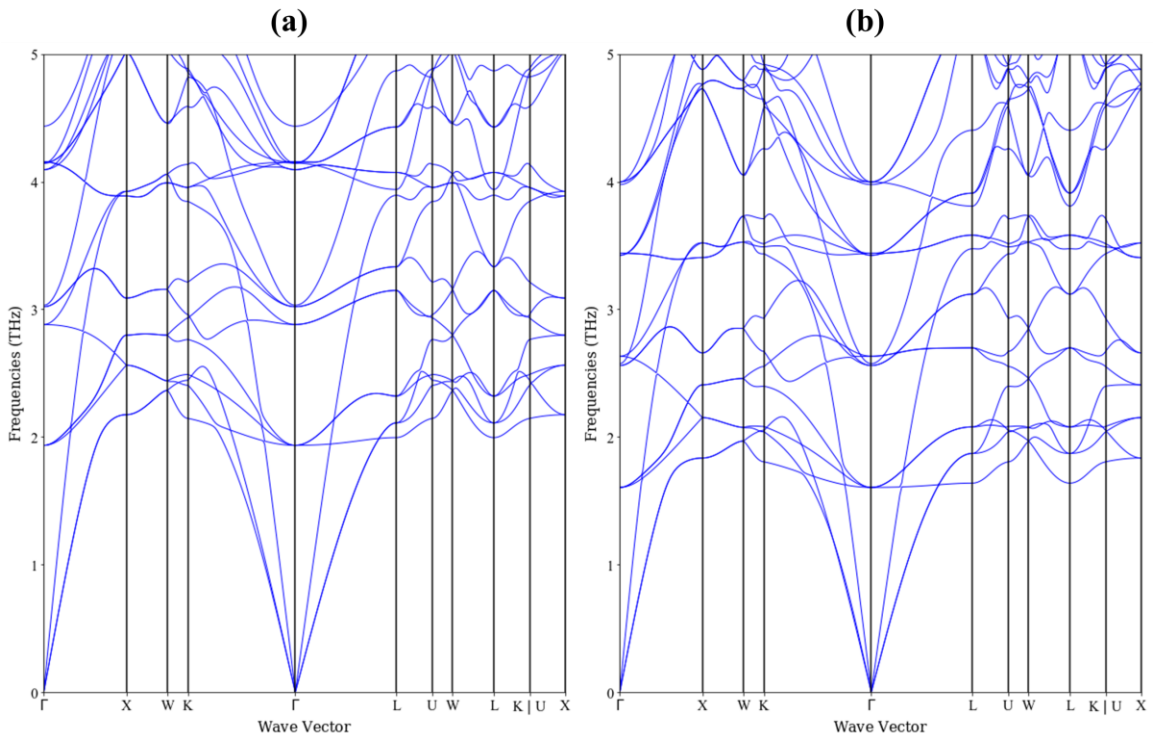

Figure 16: Phonon dispersion band structures for (a) $La_2Zr_2O_7$ and (b) $Gd_2Zr_2O_7$ calculated using Ph3pyWF.

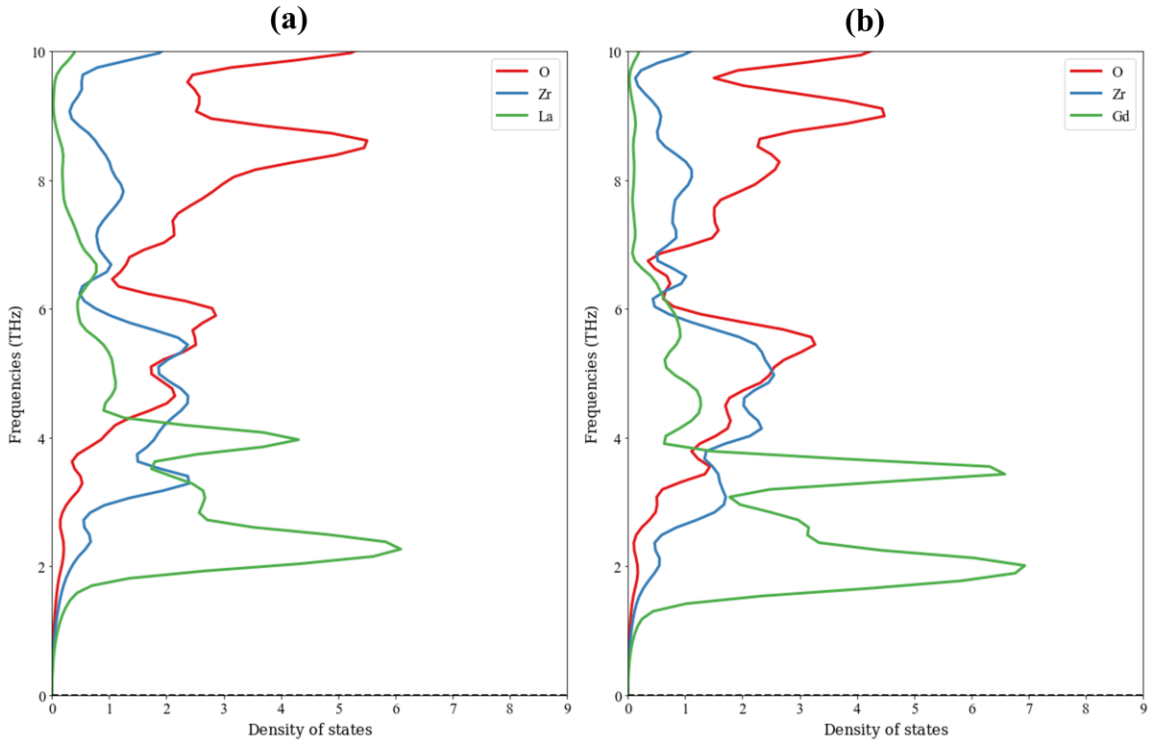

Figure 17: Phonon partial density of state (PDOS) of (a) $La_2Zr_2O_7$ and (b) $Gd_2Zr_2O_7$ calculated using Ph3pyWF.

Lan et al. [111] suggested that for pyrochlore systems with rare-earth elements occupying A sites (16d), the lower the frequency of optical phonon branches the lower the thermal conductivity due to the scattering of the transverse acoustic branches caused by the low-lying optical branches. This theory provides reasonable explanation to the lower calculated lattice thermal conductivity of $Gd_2Zr_2O_7$. The calculated results indicate the presence of low-lying optical phonon branches at lower frequency in $Gd_2Zr_2O_7$ (~1.5 THz) than in $La_2Zr_2O_7$ (~2 THz), which can be observed in the phonon dispersion band structures shown in Figure 16. The phonon PDOS shown in Figure 17 also indicate lower branches attributed by Gd in $Gd_2Zr_2O_7$ (peak at ~2.1THz) than that attributed by La in $La_2Zr_2O_7$ (peak at ~2.4THz).

Due to the larger and more complicated unit cell and supercell, these 2 workflows costed more computational resources than the other validation workflows. Both workflows have around 900 subprocesses, and each costed approximately 1000 CPU hours, with more than 4000 seconds per subprocess on average.

Chapter 6: Conclusions

In this work, an automated workflow software package, Ph3pyWF, was developed for calculating lattice thermal conductivities of TBC materials using the finite displacement method and SMRT approximation. This workflow software employs a state-of-the-art scientific workflow framework to combine multiple computational material science software packages and ultimately accelerate the exploration of new TBC topcoat materials. After continuous development and testing, the following conclusions are established.

- Ph3pyWF has shown a high degree of automation and is capable to provide a near-turn-key solution for lattice thermal conductivity and phonon properties calculations. It automates the full process of calculation, covering procedures including input file preparation, job submission, and output parsing. Compared with manual procedures, this workflow is much more efficient and avoids repetitive work, especially during error handling.
- Ph3pyWF satisfies the requirements of optional transparency and scoped provenance. It is highly transparent when running with default settings, where few input parameters have to be specified, but can be opaque for expert users to adjust specific parameters. The critical calculation results of all subprocesses in a workflow are stored in a database alongside the dependency information among subprocesses.
- The lattice thermal conductivities of several material systems were calculated using Ph3pyWF. The results validated the capabilities of Ph3pyWF to predict the properties of oxides with a wide range of thermal conductivities at relatively high accuracy.
- This work, as a good showcase, also demonstrate this framework's capability of defining dynamic workflows, which is necessary for more complicated studies.

Chapter 7: Future Outlook

One of the key objectives of this work is the degree of automation provided by Ph3pyWF, which unfortunately cannot be directly reflected in the description in this thesis. Once the source code of Ph3pyWF is made public, we aim to invite more users and contributors in various fields of study to provide feedback and improve this workflow. Some points for improvement and possibilities for future works are listed below.

- More robust version control and unit testing shall be employed as Ph3pyWF enters the public domain to reduce inconsistency and instability.
- More input parameters can be determined automatically by the workflow, including supercell size, k-point mesh, and cutoff pair distance.
- Adding compatibility with other quantum calculation engines will greatly encourage more users to accelerate their research with Ph3pyWF.
- Once the software reaches a stable version, implementation of a graphical user interface would significantly improve the user experience especially when monitoring and controlling more complicated workflows.
- Ph3pyWF can be used to conduct high-throughput screening of materials. For this purpose, it needs to be more dynamic. For example, skipping 3rd order IFCs calculations if phonon dispersion indicates a significant presence of imaginary frequency.
- Use Ph3pyWF to generate training datasets for machine learning applications in computational materials science.
- Combine Ph3pyWF and other existing workflow components to calculate other properties, such as CTE, and thus comprehensively evaluate TBC candidates.

References

1. Boyce, M. P. An Overview of Gas Turbines. *Gas Turbine Eng. Handb.* 3–56 (2006) doi:10.1016/b978-075067846-9/50004-3.
2. Frank J. Brooks & Brooks, F. J. GE gas turbine performance characteristics. *GE Power Syst. Schenectady, NY* 20 (2000).
3. Parker, D. W. Thermal barrier coatings for gas turbines, automotive engines and diesel equipment. *Mater. Des.* **13**, 345–351 (1992).
4. Hejwowski, T. & Weroński, A. The effect of thermal barrier coatings on diesel engine performance. *Vacuum* **65**, 427–432 (2002).
5. Fu, C. *et al.* DFT calculations: A powerful tool for better understanding of electrocatalytic oxygen reduction reactions on Pt-based metallic catalysts. *Comput. Mater. Sci.* **170**, 109202 (2019).
6. Nørskov, J. K., Bligaard, T., Rossmeisl, J. & Christensen, C. H. Towards the computational design of solid catalysts. *Nat. Chem.* **1**, 37–46 (2009).
7. Maldonado, A. G. & Rothenberg, G. Predictive modeling in homogeneous catalysis: A tutorial. *Chem. Soc. Rev.* **39**, 1891–1902 (2010).
8. Madsen, G. K. H. Automated search for new thermoelectric materials: The case of LiZnSb. *J. Am. Chem. Soc.* **128**, 12140–12146 (2006).
9. Gutiérrez Moreno, J. J., Cao, J., Fronzi, M. & Assadi, M. H. N. A review of recent progress in thermoelectric materials through computational methods. *Mater. Renew. Sustain. Energy* **9**, 16 (2020).
10. Miyata, M. *et al.* High-Throughput Screening of Sulfide Thermoelectric Materials Using Electron Transport Calculations with OpenMX and BoltzTraP. *J. Electron. Mater.* **47**, 3254–3259 (2018).
11. Li, R. *et al.* High-Throughput Screening for Advanced Thermoelectric Materials: Diamond-Like ABX 2 Compounds. *ACS Appl. Mater. Interfaces* **11**, 24859–24866 (2019).
12. Hautier, G. *et al.* Novel mixed polyanions lithium-ion battery cathode materials predicted by high-throughput ab initio computations. *J. Mater. Chem.* **21**, 17147 (2011).
13. Hautier, G. *et al.* Phosphates as Lithium-Ion Battery Cathodes: An Evaluation Based on High-Throughput ab Initio Calculations. *Chem. Mater.* **23**, 3495–3508 (2011).

14. Kirklin, S., Meredig, B. & Wolverton, C. High-Throughput Computational Screening of New Li-Ion Battery Anode Materials. *Adv. Energy Mater.* **3**, 252–262 (2013).
15. II, E. W. C. A Model for Technological Change Applied to the Turbojet Revolution. *Technol. Cult.* **14**, 553 (1973).
16. Miller, R. A. Thermal barrier coatings for aircraft engines: History and directions. *J. Therm. Spray Technol.* **6**, 35–42 (1997).
17. Aeronautics, F. O. R. Review of an Investigation of Ceramic Coatings for. (1947).
18. Bose, S. & DeMasi-Marcin, J. Thermal barrier coating experience in gas turbine engines at Pratt & Whitney. *J. Therm. Spray Technol.* **6**, 99–104 (1997).
19. Smialek, J. L. & Miller, R. A. Revisiting the birth of 7YSZ thermal barrier coatings: Stephan stecura. *Coatings* **8**, 7–12 (2018).
20. Aabid, A. & Khan, S. A. Optimization of Heat Transfer on Thermal Barrier Coated Gas Turbine Blade. *IOP Conf. Ser. Mater. Sci. Eng.* **370**, (2018).
21. Lugscheider, E. & Kvernes, I. Oxidation/degradation of MCrAlY Coatings. *Materwiss. Werksttech.* **19**, 241–247 (1988).
22. Akhtari-Zavareh, M. & Sarhan, A. A. D. M. Introducing New Coating Material Alloy with Potential Elements for High Corrosion Resistance for Oil and Gas Application. in *Advances in Tribology* vol. 7 16763–16776 (InTech, 2016).
23. Tawancy, H. M., Sridhar, N., Abbas, N. M. & Rickerby, D. S. Comparative performance of selected bond coats in advanced thermal barrier coating systems. *J. Mater. Sci.* **35**, 3615–3629 (2000).
24. Wortman, D. J., Duderstadt, E. C. & Nelson, W. A. Bond coat development for thermal barrier coatings. *Proc. ASME Turbo Expo* **5**, (1989).
25. Avci, A., Eker, A. A. & Eker, B. *Microstructure and Oxidation Behavior of Atmospheric Plasma-Sprayed Thermal Barrier Coatings*. *Exergetic, Energetic and Environmental Dimensions* (Elsevier, 2018). doi:10.1016/B978-0-12-813734-5.00045-7.
26. Padture, N. P., Gell, M. & Jordan, E. H. Thermal barrier coatings for gas-turbine engine applications. *Science (80-.).* **296**, 280–284 (2002).
27. Zhao, H., Yu, F., Bennett, T. D. & Wadley, H. N. G. Morphology and thermal conductivity of yttria-stabilized zirconia coatings. *Acta Mater.* **54**, 5195–5207 (2006).
28. Nicholls, J. R., Lawson, K. J., Johnstone, A. & Rickerby, D. S. Methods to reduce the

thermal conductivity of EB-PVD TBCs. *Surf. Coatings Technol.* **151**–**152**, 383–391 (2002).

29. Nicholls, J. ., Deakin, M. . & Rickerby, D. . A comparison between the erosion behaviour of thermal spray and electron beam physical vapour deposition thermal barrier coatings. *Wear* **233**–**235**, 352–361 (1999).
30. Mumm, D. R. & Evans, G. A. Mechanisms controlling the performance and durability of thermal barrier coatings. *Key Eng. Mater.* **197**, 199–230 (2001).
31. Cahit, A., Ogawa, K., Turk, A. & Ozdemir, I. Thermal Shock and Cycling Behavior of Thermal Barrier Coatings (TBCs) Used in Gas Turbines. in *Progress in Gas Turbine Performance* 13 (InTech, 2013). doi:10.5772/54412.
32. Rhys-Jones, T. N. The use of thermally sprayed coatings for compressor and turbine applications in aero engines. *Surf. Coatings Technol.* **42**, 1–11 (1990).
33. Harsha, K. S. S. *Principles of vapor deposition of thin films*. (Elsevier, 2005).
34. Hass, D. D., Parrish, P. A. & Wadley, H. N. G. Electron beam directed vapor deposition of thermal barrier coatings. *J. Vac. Sci. Technol. A Vacuum, Surfaces, Film.* **16**, 3396–3401 (1998).
35. Schiller, S., Heisig, U. & Panzer, S. *Electron beam technology*. (John Wiley & Sons, 1982).
36. Groves, J. F. & Wadley, H. N. G. Functionally graded materials synthesis via low vacuum directed vapor deposition. *Compos. Part B Eng.* **28**, 57–69 (1997).
37. Hass, D. D., Slifka, A. J. & Wadley, H. N. G. Low thermal conductivity vapor deposited zirconia microstructures. *Acta Mater.* **49**, 973–983 (2001).
38. Lu, T. J., Levi, C. G., Wadley, H. N. G. & Evans, A. G. Distributed Porosity as a Control Parameter for Oxide Thermal Barriers Made by Physical Vapor Deposition. *J. Am. Ceram. Soc.* **84**, 2937–2946 (2001).
39. Verdian, M. M., Raeissi, K. & Salehi, M. Corrosion performance of HVOF and APS thermally sprayed NiTi intermetallic coatings in 3.5% NaCl solution. *Corros. Sci.* **52**, 1052–1059 (2010).
40. El-Eskandarany, M. S. Utilization of ball-milled powders for surface protective coating. *Mech. Alloy.* 309–334 (2020) doi:10.1016/b978-0-12-818180-5.00012-1.
41. Yilbas, B. S. *HVOF coating of Nickel based alloys: Surface and mechanical characteristics*. *Comprehensive Materials Finishing* vols 3–3 (Elsevier Ltd., 2017).
42. Levi, C. G. Emerging materials and processes for thermal barrier systems. *Curr. Opin. Solid*

State Mater. Sci. **8**, 77–91 (2004).

43. Toberer, E. S., Zevalkink, A. & Snyder, G. J. Phonon engineering through crystal chemistry. *J. Mater. Chem.* **21**, 15843 (2011).
44. Nolas, G. S., Cohn, J. L., Slack, G. A. & Schujman, S. B. Semiconducting Ge clathrates: Promising candidates for thermoelectric applications. *Appl. Phys. Lett.* **73**, 178–180 (1998).
45. Keppens, V. *et al.* Localized vibrational modes in metallic solids. *Nature* **395**, 876–878 (1998).
46. Hiroi, Z., Yamaura, J. & Hattori, K. Rattling Good Superconductor: β-Pyrochlore Oxides AOs $_2$ O $_6$. *J. Phys. Soc. Japan* **81**, 011012 (2012).
47. Limarga, A. M. & Clarke, D. R. The grain size and temperature dependence of the thermal conductivity of polycrystalline, tetragonal yttria-stabilized zirconia. *Appl. Phys. Lett.* **98**, 211906 (2011).
48. Vaßen, R., Kerkhoff, G. & Stöver, D. Development of a micromechanical life prediction model for plasma sprayed thermal barrier coatings. *Mater. Sci. Eng. A* **303**, 100–109 (2001).
49. Vaßen, R. *et al.* Recent Activities in the Field of Thermal Barrier Coatings Including Burner Rig Testing in the European Union. *Adv. Eng. Mater.* **10**, 907–921 (2008).
50. Levi, C. G., Hutchinson, J. W., Vidal-Sétif, M. H. & Johnson, C. A. Environmental degradation of thermal-barrier coatings by molten deposits. *MRS Bull.* **37**, 932–941 (2012).
51. Krämer, S., Yang, J., Levi, C. G. & Johnson, C. A. Thermochemical Interaction of Thermal Barrier Coatings with Molten CaO?MgO?Al $_2$ O $_3$?SiO $_2$ (CMAS) Deposits. *J. Am. Ceram. Soc.* **89**, 3167–3175 (2006).
52. Niranatlumpong, P., Ponton, C. B. & Evans, H. E. Failure of protective oxides on plasma-sprayed NiCrAlY overlay coatings. *Oxid. Met.* **53**, 241–258 (2000).
53. Marple, B. R., Voyer, J., Moreau, C. & Nagy, D. R. Corrosion of thermal barrier coatings by vanadium and sulfur compounds. *Mater. High Temp.* **17**, 397–412 (2000).
54. Nielsen, R. Zirconium and Zirconium Compounds. in *Ullmann's Encyclopedia of Industrial Chemistry* (Wiley-VCH Verlag GmbH & Co. KGaA, 2000). doi:10.1002/14356007.a28_543.
55. Hong, J. S., De La Torre, S. D., Gao, L., Miyamoto, K. & Miyamoto, H. Synthesis and Mechanical Properties of ZrO2/Al2O3 Composites Prepared by Spark Plasma Sintering. *J. Mater. Sci. Lett.* **17**, 1313–1315 (1998).

56. Bose, S. *High temperature coatings*. (Butterworth-Heinemann, 2017).
57. Stubican, V. S. Phase Equilibria and Metastabilities in the Systems ZrO 2-MgO, ZrO 2-CaO and ZrO 2-Y 2 O 3. *Adv. Ceram.* **24**, 71–82 (1986).
58. Belmonte, M. Advanced ceramic materials for high temperature applications. *Adv. Eng. Mater.* **8**, 693–703 (2006).
59. Stecura, S. Effects of compositional changes on the performance of a thermal barrier coating system. in *Ann. Conf. on Composite and Advanced Mater.* (1978).
60. Schmitt, M. P., Rai, A. K., Bhattacharya, R., Zhu, D. & Wolfe, D. E. Multilayer thermal barrier coating (TBC) architectures utilizing rare earth doped YSZ and rare earth pyrochlores. *Surf. Coatings Technol.* **251**, 56–63 (2014).
61. Zhao, X. & Xiao, P. Thermal Barrier Coatings on Nickel Superalloy Substrates. *Mater. Sci. Forum* **606**, 1–26 (2008).
62. Mercer, C., Williams, J. ., Clarke, D. . & Evans, A. . On a ferroelastic mechanism governing the toughness of metastable tetragonal-prime (*t ′*) yttria-stabilized zirconia. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **463**, 1393–1408 (2007).
63. Dwivedi, G., Viswanathan, V., Sampath, S., Shyam, A. & Lara-Curzio, E. Fracture Toughness of Plasma-Sprayed Thermal Barrier Ceramics: Influence of Processing, Microstructure, and Thermal Aging. *J. Am. Ceram. Soc.* **97**, 2736–2744 (2014).
64. Taylor, R., Brandon, J. R. & Morrell, P. Microstructure, composition and property relationships of plasma-sprayed thermal barrier coatings. *Surf. Coatings Technol.* **50**, 141–149 (1992).
65. Han, M., Tang, X., Yin, H. & Peng, S. Fabrication, microstructure and properties of a YSZ electrolyte for SOFCs. *J. Power Sources* **165**, 757–763 (2007).
66. Krämer, S., Yang, J., Levi, C. G. & Johnson, C. A. Thermochemical interaction of thermal barrier coatings with molten CaO-MgO-Al 2O 3-SiO 2 (CMAS) deposits. *J. Am. Ceram. Soc.* **89**, 3167–3175 (2006).
67. Minervini, L., Grimes, R. W. & Sickafus, K. E. Disorder in pyrochlore oxides. *J. Am. Ceram. Soc.* **83**, 1873–1878 (2000).
68. Maloney, M. J. *et al.* Thermal barrier coating systems and materials. (2000).
69. Cao, X. Q., Vassen, R. & Stoever, D. Ceramic materials for thermal barrier coatings. *J. Eur. Ceram. Soc.* **24**, 1–10 (2004).

70. Cao, X. Q. *et al.* Thermal Stability of Lanthanum Zirconate Plasma-Sprayed Coating. *J. Am. Ceram. Soc.* **84**, 2086–2090 (2001).
71. Vassen, R., Cao, X., Tietz, F., Basu, D. & Stöver, D. Zirconates as new materials for thermal barrier coatings. *J. Am. Ceram. Soc.* **83**, 2023–2028 (2000).
72. Krämer, S., Yang, J. & Levi, C. G. Infiltration-Inhibiting Reaction of Gadolinium Zirconate Thermal Barrier Coatings with CMAS Melts. *J. Am. Ceram. Soc.* **91**, 576–583 (2008).
73. Subramanian, M. A., Aravamudan, G. & Subba Rao, G. V. Oxide pyrochlores - A review. *Prog. Solid State Chem.* **15**, 55–143 (1983).
74. Vaßen, R., Jarligo, M. O., Steinke, T., Mack, D. E. & Stöver, D. Overview on advanced thermal barrier coatings. *Surf. Coatings Technol.* **205**, 938–942 (2010).
75. Development, C. P. New Thermal Barrier Coatings Based on Pyrochlore/YSZ Double-Layer Systems. **61**, 351–361 (2004).
76. Guo, R., Bhalla, A. S. & Cross, L. E. Ba(Mg $_{1/3}$ Ta $_{2/3}$)O $_{3}$ single crystal fiber grown by the laser heated pedestal growth technique. *J. Appl. Phys.* **75**, 4704–4708 (1994).
77. Wright, A. J. *et al.* From high-entropy ceramics to compositionally-complex ceramics: A case study of fluorite oxides. *J. Eur. Ceram. Soc.* **40**, 2120–2129 (2020).
78. Gild, J. *et al.* High-entropy fluorite oxides. *J. Eur. Ceram. Soc.* **38**, 3578–3584 (2018).
79. Curtarolo, S. *et al.* The high-throughput highway to computational materials design. *Nat. Mater.* **12**, 191–201 (2013).
80. Mathew, K. *et al.* Atomate: A high-level interface to generate, execute, and analyze computational materials science workflows. *Comput. Mater. Sci.* **139**, 140–152 (2017).
81. Calderon, C. E. *et al.* The AFLOW standard for high-throughput materials science calculations. *Comput. Mater. Sci.* **108**, 233–238 (2015).
82. Gorai, P. *et al.* TE Design Lab: A virtual laboratory for thermoelectric material design. *Comput. Mater. Sci.* **112**, 368–376 (2016).
83. Taylor, I. J., Deelman, E., Gannon, D. B. & Shields, M. *Workflows for e-Science: scientific workflows for grids*. vol. 1 (Springer, 2007).
84. Krämer, M., Würz, H. M. & Altenhofen, C. Executing cyclic scientific workflows in the cloud. *J. Cloud Comput.* **10**, (2021).
85. Jain, A. *et al.* Fireworks: A dynamic workflow system designed for highthroughput applications. *Concurr. Comput. Pract. Exp.* **27**, 5037–5059 (2015).

86. Ong, S. P. *et al.* Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. *Comput. Mater. Sci.* **68**, 314–319 (2013).
87. Hafner, J. Ab-initio simulations of materials using VASP: Density-functional theory and beyond. *Journal of Computational Chemistry* vol. 29 (2008).
88. Wang, V., Xu, N., Liu, J.-C., Tang, G. & Geng, W.-T. VASPKIT: A user-friendly interface facilitating high-throughput computing and analysis using VASP code. *Comput. Phys. Commun.* **267**, 108033 (2021).
89. Yi, W., Tang, G., Chen, X., Yang, B. & Liu, X. qvasp: A flexible toolkit for VASP users in materials simulations. *Comput. Phys. Commun.* **257**, 107535 (2020).
90. Huber, S. P. *et al.* Common workflows for computing material properties using different quantum engines. *npj Comput. Mater.* **7**, (2021).
91. Zheng, C. *et al.* Automated generation and ensemble-learned matching of X-ray absorption spectra. *npj Comput. Mater.* **4**, 1–9 (2018).
92. Smidt, T. E., Mack, S. A., Reyes-Lillo, S. E., Jain, A. & Neaton, J. B. An automatically curated first-principles database of ferroelectrics. *Sci. Data* **7**, 1–22 (2020).
93. Cheng, Y. X. *et al.* Vacancy formation energy and its connection with bonding environment in solid: A high-throughput calculation and machine learning study. *Comput. Mater. Sci.* **183**, 109803 (2020).
94. Pizzi, G., Cepellotti, A., Sabatini, R., Marzari, N. & Kozinsky, B. AiiDA: automated interactive infrastructure and database for computational science. *Comput. Mater. Sci.* **111**, 218–230 (2016).
95. Jain, A. *et al.* Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL Mater.* **1**, 011002 (2013).
96. ICSD - the world's largest database for completely identified inorganic crystal structures. https://icsd.products.fiz-karlsruhe.de/en.
97. Curtarolo, S. *et al.* AFLOW: An automatic framework for high-throughput materials discovery. *Comput. Mater. Sci.* **58**, 218–226 (2012).
98. Yang, K., Setyawan, W., Wang, S., Buongiorno Nardelli, M. & Curtarolo, S. A search model for topological insulators with high-throughput robustness descriptors. *Nat. Mater.* **11**, 614–619 (2012).
99. Mounet, N. *et al.* Two-dimensional materials from high-throughput computational

exfoliation of experimentally known compounds. *Nat. Nanotechnol.* **13**, 246–252 (2018).

100. Togo, A. & Tanaka, I. First principles phonon calculations in materials science. *Scr. Mater.* **108**, 1–5 (2015).
101. Togo, A., Chaput, L. & Tanaka, I. Distributions of phonon lifetimes in Brillouin zones. *Phys. Rev. B - Condens. Matter Mater. Phys.* **91**, (2015).
102. Li, W., Carrete, J., Katcho, N. A. & Mingo, N. ShengBTE: A solver of the Boltzmann transport equation for phonons. *Comput. Phys. Commun.* **185**, 1747–1758 (2014).
103. Brorsson, J. *et al.* Efficient Calculation of the Lattice Thermal Conductivity by Atomistic Simulations with Ab Initio Accuracy. *Adv. Theory Simulations* **5**, 1–13 (2022).
104. Eriksson, F., Fransson, E. & Erhart, P. The Hiphive Package for the Extraction of High-Order Force Constants by Machine Learning. *Adv. Theory Simulations* **2**, 1800184 (2019).
105. Mathew, K. *et al.* MPInterfaces: A Materials Project based Python tool for high-throughput computational screening of interfacial systems. *Comput. Mater. Sci.* **122**, 183–190 (2016).
106. Hjorth Larsen, A. *et al.* The atomic simulation environment - A Python library for working with atoms. *J. Phys. Condens. Matter* **29**, (2017).
107. Supka, A. R. *et al.* AFLOWπ: A minimalist approach to high-throughput ab initio calculations including the generation of tight-binding hamiltonians. *Comput. Mater. Sci.* **136**, 76–84 (2017).
108. Liu, Y. *et al.* Discovery of ABO 3 perovskites as thermal barrier coatings through high-throughput first principles calculations. *Mater. Res. Lett.* **7**, 145–151 (2019).
109. Fan, T. & Oganov, A. R. AICON: A program for calculating thermal conductivity quickly and accurately. *Comput. Phys. Commun.* **251**, (2020).
110. Taheri, A. Predicting Phonon Transport in Two-Dimensional materials Using First-Principles Calculations and the Boltzmann by. (2020).
111. Lan, G., Ouyang, B. & Song, J. The role of low-lying optical phonons in lattice thermal conductance of rare-earth pyrochlores: A first-principle study. *Acta Mater.* **91**, 304–317 (2015).
112. Seko, A. *et al.* Prediction of Low-Thermal-Conductivity Compounds with First-Principles Anharmonic Lattice-Dynamics Calculations and Bayesian Optimization. *Phys. Rev. Lett.* **115**, 1–5 (2015).
113. Tadano, T., Gohda, Y. & Tsuneyuki, S. Anharmonic force constants extracted from first-

principles molecular dynamics: Applications to heat transfer simulations. *J. Phys. Condens. Matter* **26**, (2014).

114. Kapral, R. & Ciccotti, G. Molecular dynamics. *Theory Appl. Comput. Chem.* 425–441 (2005) doi:10.1016/b978-044451719-7/50059-7.
115. Hartree, D. R. The Wave Mechanics of an Atom with a Non-Coulomb Central Field Part I Theory and Methods. *Math. Proc. Cambridge Philos. Soc.* **24**, 89–110 (1928).
116. The exact calculation of long-range forces between atoms by perturbation theory. *Proc. R. Soc. London. Ser. A. Math. Phys. Sci.* **233**, 70–74 (1955).
117. David Sherrill, C. & Schaefer, H. F. The Configuration Interaction Method: Advances in Highly Correlated Approaches. in 143–269 (1999). doi:10.1016/S0065-3276(08)60532-8.
118. Schütt, O. & VandeVondele, J. Machine Learning Adaptive Basis Sets for Efficient Large Scale Density Functional Theory Simulation. *J. Chem. Theory Comput.* **14**, 4168–4175 (2018).
119. Hohenberg, P. & Kohn, W. Inhomogeneous Electron Gas. *Phys. Rev.* **136**, B864–B871 (1964).
120. Kohn, W. & Sham, L. J. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* **140**, A1133–A1138 (1965).
121. Ceperley, D. M. & Alder, B. J. Ground state of the electron gas by a stochastic method. *Phys. Rev. Lett.* **45**, 566–569 (1980).
122. Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **77**, 3865–3868 (1996).
123. Tao, J., Perdew, J. P., Staroverov, V. N. & Scuseria, G. E. Climbing the density functional ladder: Nonempirical meta–generalized gradient approximation designed for molecules and solids. *Phys. Rev. Lett.* **91**, 3–6 (2003).
124. Staroverov, V. N., Scuseria, G. E., Tao, J. & Perdew, J. P. Comparative assessment of a new nonempirical density functional: Molecules and hydrogen-bonded complexes. *J. Chem. Phys.* **119**, 12129–12137 (2003).
125. Kittel, C., McEuen, P. & McEuen, P. *Introduction to solid state physics*. vol. 8 (Wiley New York, 1996).
126. Kshirsagar, S., Mandadapu, K. K. & Papadopoulos, P. Classical molecular dynamics simulations of crystal lattices with truncated Taylor series-based interatomic potentials.

Comput. Mater. Sci. **120**, 127–134 (2016).

127. Custodian. https://github.com/materialsproject/custodian.
128. Harris, D. C. *et al.* Properties of an infrared-transparent MgO: Y2O3 nanocomposite. *J. Am. Ceram. Soc.* **96**, 3828–3835 (2013).
129. Slifka, A. J., Filla, B. J. & Phelps, J. M. Thermal Conductivity of Magnesium Oxide from Absolute, Steady-State Measurements. *J. Res. Natl. Inst. Stand. Technol.* **103**, 357–363 (1998).
130. Lan, G., Ou, P., Chen, C. & Song, J. A complete computational route to predict reduction of thermal conductivities of complex oxide ceramics by doping: A case study of La2Zr2O7. *J. Alloys Compd.* **826**, 154224 (2020).
131. Wang, Y., Yang, F. & Xiao, P. Glass-like thermal conductivities in (La 1-x1 Y x1) 2 (Zr 1-x2 Y x2) 2 O 7-x2 (x = x 1 + x 2, 0≤ x ≤ 1.0) solid solutions. *Acta Mater.* **60**, 7024–7033 (2012).
132. Sun, L., Guo, H., Peng, H., Gong, S. & Xu, H. Phase stability and thermal conductivity of ytterbia and yttria co-doped zirconia. *Prog. Nat. Sci. Mater. Int.* **23**, 440–445 (2013).
133. Raza, M. Chimie des Interactions Plasma-Surface (ChIPS), Faculty of Sciences Ph . D . Thesis Oxygen vacancy stabilized zirconia ; synthesis and properties Mohsin Raza Chimie des Interactions Plasma-Surface (ChIPS) Chimie des Interactions Plasma-Surface (ChIPS. (2017) doi:10.13140/RG.2.2.30274.58566.
134. Parchovianský, M., Galusek, D., Švančárek, P., Sedláček, J. & Šajgalík, P. Thermal behavior, electrical conductivity and microstructure of hot pressed Al2O3/SiC nanocomposites. *Ceram. Int.* **40**, 14421–14429 (2014).
135. Olorunyolemi, T. *et al.* Thermal conductivity of zinc oxide: From green to sintered state. *J. Am. Ceram. Soc.* **85**, 1249–1253 (2002).
136. Takahashi, Y. & Murabayashi, M. Measurement of Thermal Properties of Nuclear Materials By Laser Flash Method. *J. Nucl. Sci. Technol.* **12**, 133–144 (1975).
137. Slack, G. A. & Austerman, S. B. Thermal conductivity of BeO single crystals. *J. Appl. Phys.* **42**, 4713–4717 (1971).
138. FRANCL, J. & KINGERY, W. D. Thermal Conductivity: IV, Apparatus for Determining Thermal Conductivity by a Comparative Method. *J. Am. Ceram. Soc.* **37**, 80–84 (1954).
139. Zhang, Y. *et al.* Low thermal conductivity in La2Zr2O7 pyrochlore with A-site partially

substituted with equimolar Yb2O 3 and Er2O3. *Ceram. Int.* **40**, 9151–9157 (2014).

140. Wang, Y., Yang, F. & Xiao, P. Rattlers or oxygen vacancies: Determinant of high temperature plateau thermal conductivity in doped pyrochlores. *Appl. Phys. Lett.* **102**, (2013).

141. Jiang, T., Xie, M., Guan, L., Wang, X. & Song, X. Effect of Nb5+ and Cu2+ codoping on thermal properties of Gd2Zr2O7 ceramic. *J. Rare Earths* **39**, 180–185 (2021).

142. Guo, L., Guo, H., Peng, H. & Gong, S. Thermophysical properties of Yb2O3 doped Gd2Zr2O7 and thermal cycling durability of (Gd0.9Yb0.1)2Zr2O7/YSZ thermal barrier coatings. *J. Eur. Ceram. Soc.* **34**, 1255–1263 (2014).