Oxide Chemomechanics by Hybrid Atomistic Machine Learning Methods

Caccin, Marco

Awarding institution:
King's College London

Oxide Chemomechanics by Hybrid Atomistic Machine Learning Methods

Marco Caccin

Department of Physics
King's College London

This dissertation is submitted for the degree of
Doctor of Philosophy

May 2017

Acknowledgements

I would like to thank my supervisor Prof. Alessandro De Vita that offered me the opportunity of working at King's while also inspiring my work and giving me ample room for personal development. I could not conceive my technical development towards becoming a computational physicist without the priceless mentoring of Dr. James Kermode, who has always been my guide and deserves my gratitude and admiration. I am also grateful to my external supervisor Prof. O. Anatole von Lilienfeld for the discussions and the work we carried out together on machine learning for the prediction of molecular systems properties, and above all for giving me the chance of spending one of the most memorable months of my life at Argonne National Labs. Furthermore, I must thank Prof. Gabor Csányi and Dr. Albert Bartók–Pártay for the work carried out together on the GAP project and for the insightful discussions on the topic. An award of computer time was provided by the Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program. This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

The last four years in London have been extraordinary, and I owe it all to the people with which I shared my journey and the music and art I have been immersed in. Friends like Marcello, Marco, Giovanni (x2), Pooya, Alix, Dan, Josep, Natalia, Mateusz, Federico, Yasmine, Nanaxhi, Carla, Danilo, Filippo, Luigi, Artur, Emily, Sam, Katrine, Julian, Pedro, Mathilde, Francesca, Benedict, and too many more to fit in here, and places like the Silver Bullet jam and the Troy bar, have all left a permanent mark in my soul.

Grazie to my family and close friends in Italy that always believed in my potential, and to Roberta for brightening up every moment of the best part of my life with the gentle light of our love.

Abstract

Atomic scale phenomena concurring to atomic bond ruptures at a crack tip determine the chemomechanical properties of oxide materials, thus a better understanding of them is instrumental in addressing engineering issues related to the brittleness of oxides. In a fracturing material, the macroscopic stress field couples with the chemical reactions occurring at the crack tip in a bidirectional interplay requiring a concurrent multiscale (QM/MM) computational approach. Due to long range electrostatic interactions, the dynamics of chemically accurate description of the neighbourhood of a breaking bond requires ab initio calculations on several hundred atoms on a timescale of picoseconds. First, to make these prohibitively demanding calculations tractable I developed an ensemble parallel QM/MM computational strategy comprising a novel graph partitioning method for optimal load balancing that is able to efficiently parallellise the workload over hundreds of thousands of cores on supercomputing facilities. Secondly, I present a computational study of crack propagation in two–dimensional silica systems that have recently been experimentally synthesized, which provide ideal and physically observed structures that are key to the understanding of atomic scale phenomena in fracture events in oxides. The atomic structure, either crystalline or amorphous, and the emerging set of free energy barriers to crack advance are the basis to understand the fundamental difference in the crack dynamics observed at a larger scale. Finally, I explored different pathways to make efficient use of the information produced by ab initio calculations by studying machine learning methods capable of predicting local physical observables as a function of the local atomic environment. This includes a machine learning–augmented method to obtain free energy barriers of hybrid accuracy that only require DFT calculations on just a small fraction of the sampled atomic configurations.

Preface

This dissertation is a narrative of the work carried out between September 2012 and September 2016 at the Department of Physics of King's College London under the supervision of Prof. Alessandro De Vita, Prof. O. Anatole von Lilienfeld and Dr. James Kermode. During the course of my Ph.D. I have contributed to the publication

> Marco Caccin, Zhenwei Li, James R. Kermode, and Alessandro De Vita. 'A framework for machine learning augmented multiscale atomistic simulations on parallel supercomputers'. Published on International Journal of Quantum Chemistry

presented in Chapter 4, where material from this paper appears.

A large part of the software developed for this work has been released under GNU GPL3 open source license. The 'digestive ripening' partitioning algorithm for atomic systems of Chapter 4 is available at

`github.com/marcocaccin/DigestiveRipening`

and a large portion of the software developed for the work on two–dimensional silica of Chapters 5 and 6 is available at

`github.com/marcocaccin/MLTI`.

This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. It has not been submitted in whole or in part for any degree or diploma at this or any other university.

Marco Caccin
May 2017

Table of contents

List of figures

List of tables

Abbreviations

Roman Symbols

BFGS	Broyden–Fletcher–Goldfarb–Shanno (optimisation algorithm)
BG/Q	BlueGene/Q
CART	Classification and regression tree
CG	Conjugate gradients (optimisation algorithm)
CN	Compute node (BlueGene/Q)
DFT	Density functional theory
DR	Digestive Ripening (partitioning)
FEN	Front end node (BlueGene/Q)
FIRE	Fast inertial relaxation engine (optimisation algorithm)
GAP	Gaussian approximation potential
GBRT	Gradient boosting regression trees
GGA	Generalised gradient approximation
GP	Gaussian process
GPR	Gaussian process regression
IP	Interatomic potential
L–BFGS	limited memory BFGS (optimisation algorithm)
LDA	Local density approximation

LOTF	Learn on the fly
MD	Molecular dynamics
ML	Machine learning
MM	Molecular mechanics
PES	Potential energy surface
QM	Quantum mechanics
RKHS	Reproducing kernel Hilbert space
SGTI	Slow growth thermodynamic integration
SOAP	Smooth overlap over atomic positions
TI	Thermodynamic integration
TS	Tangney–Scandolo (interatomic potential)
TTS	Time to solution (of calculation)

Chapter 1

Introduction

Information is perhaps the single most defining feature of this era. Whether we consider producing information by means of human creation or computer–generated data, storing it in ever growing databases, or building knowledge from it, everything around information seems to share the same attribute of exponential growth. This inflationary evolution is arguably the effect of two concurrent phenomena. On one hand, the network of worldwide computing resources is an entity which grows in power because of technology innovations, in size because of the technology adoption of an ever growing human population, and more connected thanks to the internet. On the other hand, machine learning – which will be briefly overviewed in Chapter 2 – makes it possible to distil meaning from the otherwise overwhelming amount of information produced by the actions and interactions within such entity. It is then only natural for this revolution to trickle in every field it can be in contact with, including computational physics.

It is in this context that we ask ourselves questions such as 'how does glass break?'. Of the several interpretations this question can have, in this thesis we will concentrate on understanding the atomic scale phenomena that give rise to specific chemomechanical properties of oxide materials. Atomistic simulations is an essential tool to address this problem, and will be the topic of the background Chapter 3. Quantum mechanics is widely accepted to be the fundamental theory needed to describe reality at the atomic scale, but it must be approximated by numerical methods for any practical applications. Density functional theory (DFT) is a way of rephrasing Schrödinger equation in terms of the groundstate electron density explicitly. Approximate solutions to this approach are very popular and are widely considered to be sufficiently accurate to quantitatively describe atomic interactions arising in systems such as breaking oxide materials. Accuracy, though, comes at a computational cost, and simulations using this theory can only deal with up

to about a thousand atoms and a timescale of a few picoseconds. Interatomic potentials, instead, are simple interaction models designed to qualitatively capture the physics of a system within certain boundaries of applicability, and can be used to study millions of atoms and up to nanoseconds. Simulating processes such as the advance of a crack is a problem that at the same time requires the accuracy of DFT at the crack tip and systems large enough ($\geq 10^5$ atoms) to accommodate stress fields compatible with the linear elastic solution of a macroscopic system, and the two length scales are inextricably interconnected.

In this scenario a multiscale approach with hybrid accuracy is necessary, but the long–range nature of interactions in oxides inflates the computational cost of every accurate DFT calculation by a few orders of magnitude with respect to those performed in simpler systems such as silicon. To address this kind of problem it is thus necessary to leverage some of today's most powerful computing resources available, and Chapter 4 will describe the conception of a parallel computing framework to perform multiscale simulations on supercomputing facilities.

Despite the need for accuracy and computing power, there exist cases where a uniform precision interatomic potential can be a useful first step for understanding the qualitative behaviour of a material undergoing fracture. Benchmarking results show that one such example is two–dimensional silica, the subject of Chapter 5: here, a study on the dynamic crack propagation and on the crack advance reaction kinetics will be carried out on both crystalline and amorphous systems.

In the persistent endeavour to compute as little as possible, the attention will then turn to machine learning models capable of making better use of the accurate but expensive DFT calculations. Chapter 6 will present three such attempts, culminating in the last where all three main motifs of this work – oxide chemomechanics, machine learning and hybrid atomistic simulations – come together to create a method to evaluate free energy profiles of crack advance in two–dimensional silica at an accuracy approaching DFT.

Finally, Chapter 7 will link back to the Introduction, and expand it by summarising the insights gained along the journey and delineating future research directions.

Chapter 2

Machine Learning

A common task in experimental sciences is to generate a model that is able to discern the mapping between input and output of a measurement given a finite dataset. In a traditional interpolation framework, the design of a model is based upon prior beliefs – mathematical derivation, physical intuition or both – about the underlying process generating the data. The functional form of the model transforming an input data point to an output observable is thus once and for all fixed, and data is fed into the fitting procedure to find the set of parameters that best describes it, for example via ordinary least squares fitting. The fitted model can then be used to infer the outcome of a measurement on not yet observed data points, and its optimal parameters will have an interpretation in accordance with the derivation of the model. The procedure outlined falls within the definition of parametric regression and can be viewed as the simplest example among machine learning (ML) methods. More powerful machine learning algorithms differ from a conventional fitting procedure inasmuch as they provide a set of rules with which the model itself is generated from the data. By eliminating the separation between design of the model and parameter optimisation, this different approach extends the scope of fitting methods so that it becomes possible to treat problems in which the design of an explicit model is not possible.

The two main categories of machine learning problems are supervised learning and unsupervised learning [1]. Supervised learning is the task of learning the function describing a labelled dataset, where for each input data point an output value is provided in the form of a categorical label, a scalar value or an array. Supervised learning itself comprises two subcategories: regression and classification. Regression methods are used when the output data is one or more continuous variables, as for example in the case of traditional curve fitting. Classification methods are instead used when the output data is one or more discrete labels, where the task involves drawing decision boundaries among different classes.

Unsupervised learning, on the contrary, deals with finding structure in datasets without any preassigned label as in the case of clustering methods. In the following sections we will review some of the machine learning tools that have been used throughout the work of this thesis: the graph partitioning method of Chapter 4 will make use of various clustering (unsupervised learning) techniques, GAP potentials will be Gaussian process regression models trained on data points sparsified by k–means clustering, and the the ML–augmented thermodynamic integration method will use Gaussian process regression for Bayesian hyperparameter optimisation, k–means clustering for training data selection and boosted trees regression as a ML model.

2.1 Supervised Learning for Regression

Gaussian Process Regression

Gaussian process regression (GP regression or GPR) [2–4] is a nonparametric regression method for supervised learning. A GP is a stochastic process for which any finite linear combination of samples follows a multivariate Gaussian distribution. Formally, any dataset can be expressed in terms of a GP: given a set of N observations $\mathbf{y} = \{y_1, \ldots, y_N\} \in \mathcal{Y}^N$ measured at points $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \in \mathcal{X}^N$, one can think of it as a finite sample of the infinite–dimensional vector representing an unknown function y, where y is a Gaussian process

$$y \sim GP(m, k) \tag{2.1}$$

governed by its mean function m and its covariance function k. For a finite dataset, then, the sample will follow a multivariate Gaussian distribution

$$\mathbf{y} \sim \mathcal{N}\left(m(\mathbf{X}), k(\mathbf{X}, \mathbf{X})\right) . \tag{2.2}$$

Without loss of generality, the mean function can be set to be identically zero [4]. A GP then becomes entirely defined by its covariance function k, also known as kernel function or correlation function,

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R} . \tag{2.3}$$

The kernel function k determines the correlation between each pair of data points, which in a Bayesian perspective is equivalent to imposing a prior over the fitting function (*i.e.*, its smoothness and periodicity). Its choice is crucial to successfully model a dataset by means of a GP, and its only constraint is to produce a symmetric positive semidefinite kernel

matrix $k(\mathbf{X}, \mathbf{X})$ *i.e.*,

$$\mathbf{v}^T \, k(\mathbf{X}, \mathbf{X}) \, \mathbf{v} \geq 0 \;\; \forall \, \mathbf{v} \in \mathbb{R}^N \; \forall \, \mathbf{X} \in \mathcal{X}^N. \tag{2.4}$$

In a regression scheme, the aim is to predict the value of an observation $\hat{y}_t$ at a point in input space $\mathbf{x}_t$. If we denote $k(\mathbf{x}_t, \mathbf{X}) = [k(\mathbf{x}_t, \mathbf{x}_1), \ldots, k(\mathbf{x}_t, \mathbf{x}_N)]^T$, we see that the vector $\left[\mathbf{y}^T, \hat{y}_t\right]^T$ must too follow a joint Gaussian distribution

$$\begin{bmatrix} \mathbf{y} \\ \hat{y}_t \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(\mathbf{X}, \mathbf{X}) & k(\mathbf{x}_t, \mathbf{X})^T \\ k(\mathbf{x}_t, \mathbf{X}) & k(\mathbf{x}_t, \mathbf{x}_t) \end{bmatrix}\right), \tag{2.5}$$

and the posterior (conditional) probability distribution for $\hat{y}_t$

$$P\left(\hat{y}_t | (\mathbf{X}, \mathbf{y}), \mathbf{x}_t\right) \tag{2.6}$$

can be evaluated analytically and is itself a Gaussian distribution:

$$\hat{y}_t \sim \mathcal{N}\left(m(\mathbf{x}_t), s^2(\mathbf{x}_t)\right). \tag{2.7}$$

The variables m and s^2 in Eq. 2.7 are respectively the mean and variance of the posterior probability distribution at test point $\mathbf{x}_t$. The posterior mean

$$m(\mathbf{x}_t) = k\left(\mathbf{x}_t, \mathbf{X}\right)^T k\left(\mathbf{X}, \mathbf{X}\right)^{-1} \mathbf{y} \tag{2.8}$$

is the most likely value of the function at $\mathbf{x}_*$ and will be the prediction of our model, whereas the posterior variance

$$s^2(\mathbf{x}_t) = k\left(\mathbf{x}_t, \mathbf{x}_t\right) - k\left(\mathbf{x}_t, \mathbf{X}\right) k\left(\mathbf{X}, \mathbf{X}\right)^{-1} k\left(\mathbf{x}_t, \mathbf{X}\right)^T \tag{2.9}$$

is a measure of the uncertainty in our prediction (we must stress here that this derivation does not depend on the choice of kernel, as the Gaussian distribution properties apply to all GPs).

A very common choice is the Gaussian kernel (commonly referred as squared exponential kernel) used in the majority of the GPR applications in this thesis

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{2\sigma_c^2}\right), \tag{2.10}$$

which produces an infinitely differentiable fitting function and sets a decaying correlation between data points with a correlation length σ_c. If we now allow the data points to be corrupted with a zero mean Gaussian noise $\mathcal{N}(0, \sigma_n^2)$, our covariance function will become

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{2\sigma_c^2}\right) + \sigma_n^2 \delta_{ij} \,. \tag{2.11}$$

The parameters $\boldsymbol{\theta} = \{\sigma_c, \sigma_n\}$, also known as hyperparameters, have a fundamental role in determining the accuracy of GPR predictions. They can be determined *a priori* in cases when the noise level on the data and the typical length scale over which the function is expected to vary significantly. Alternatively, they can be inferred from the data by maximising the posterior probability of the hyperparameters $P(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$, also known as likelihood, by applying Bayes' theorem:

$$P(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) \propto P(\boldsymbol{\theta}|(\mathbf{X}, \mathbf{y}))\, P(\boldsymbol{\theta}) \,. \tag{2.12}$$

Assuming no prior knowledge over the hyperparameters, maximising the log–likelihood function

$$\log P(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^T k(\mathbf{X}, \mathbf{X})^{-1} \mathbf{y} - \frac{1}{2}\log\det k(\mathbf{X}, \mathbf{X}) + \text{constant} \tag{2.13}$$

will, in principle, give the best estimate for the hyperparameters that can be inferred from the provided information.

It is interesting to note that the same mathematical result for the posterior mean of a GPR can be obtained by *kernel ridge regression*, which requires the same problem to be viewed from a regularisation perspective [5, 6]. In this view, the fitting function f is assumed to belong to a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$ determined by the kernel function, which is just another way of saying that f will be a linear combination of kernel functions of different parameters. With this hypothesis, one wishes to minimise the objective function

$$\mathcal{L}(f) = \frac{1}{N}\sum_{i=1}^{N} (f(\mathbf{x}_i) - y_i)^2 + \sigma_n^2 |f|_k^2 \,, \tag{2.14}$$

where the first term is the mean squared error and the second is a Tikhonov regularisation term [7] whose purpose is to prevent overfitting. The space of functions $\mathcal{H}_k$ can be defined by its reproducing kernel k of Eq. 2.3, and the requirement for a symmetric and positive definite covariance matrix is translated into the ability to define a proper scalar product

$\langle \cdot, \cdot \rangle_k$, so that the function k

$$f(\mathbf{x}) = \langle f, k(\mathbf{x}, \cdot) \rangle_k \,, \;\; \forall \mathbf{x} \in \mathcal{X} \,, \; \forall f \in \mathcal{H}_k \,. \tag{2.15}$$

is unique. Calculating the kernel value between two data points $k(\mathbf{x}_i, \mathbf{x}_j)$ corresponds to evaluating the inner product $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ in feature space $\mathcal{H}_k$. Thanks to the *kernel trick*, this inner product in a high dimensional space can be performed without ever evaluating the mapping of the pair of points in feature space, and ϕ and $\mathcal{H}_k$ do not need to be known explicitly.

The fitting function $f(\mathbf{x})$ is written as a linear combination of kernel functions centred around each data point

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i k(\mathbf{x}, \mathbf{x}_i) \tag{2.16}$$

and the regularisation term in Eq. 2.14 will be

$$|f|_k^2 = \langle \sum_i c_i \phi(\mathbf{x}_i), \sum_j c_j \phi(\mathbf{x}_j) \rangle = \sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) \,. \tag{2.17}$$

The minimisation of Eq. 2.14 can then be performed analytically, with the regression coefficients vector $\mathbf{c} = (c_1, \ldots, c_N)^T$ being determined by

$$\mathbf{c} = \left(k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{1} \right)^{-1} \mathbf{y} \,, \tag{2.18}$$

reproducing exactly the solution obtained in Eq. 2.8. In other words, we can see that fitting a GPR model is equivalent to finding the best linear combination of kernel functions centred around each data point.

A GPR algorithm is a transparent ML method, where the tuneable hyperparameters can be interpreted directly from their effect on the covariance between data points, and even the priors on the model itself (*i.e.*, the functional forms of the kernel) can be tested one against another by means of Bayesian hypothesis testing. Another reason why it should be preferred over other ML methods is that the mathematical framework of a covariance matrix gives access to the analytical derivative of the fitting function, and that a model can be constructed from a dataset of function values and derivatives – for a detailed treatment of this topic see Ref. [8]. The main downside of this algorithm is its scalability, as it formally requires the inversion of a dense covariance matrix of size $N \times N$, which has a worst–case scenario scaling of $\mathcal{O}(N^3)$.

Regularised Boosted Regression Trees

Gradient boosting regression trees (GBRTs) [9] is a versatile, high performance and scalable technique that can be used for ML problems of classification, regression and ranking. Its currently most popular implementation is the XGBoost open source package [10], on which the overview of this section is based.

The GBRT technique belongs to the class of boosting methods, in which an ensemble of weak learners $\{f_k\}$, each of which would have a poor predictive power, is bundled together to create one strong learner that can achieve an arbitrary high accuracy [11–13]. For an ensemble model M, the prediction $\hat{y}_t$ at point $\mathbf{x}_t$ is simply the sum of the predictions of each learner:

$$\hat{y}_t = M(\mathbf{x}_t) = \sum_{k=1}^{K} f_k(\mathbf{x}_t)\,, \tag{2.19}$$

where in this case the weak learners $\{f_k\}$ are shallow regression trees, also commonly denoted as CART (Classification and Regression Tree) [14].
Starting from a given a dataset $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$, the target of the training procedure is to progressively minimise a predefined objective function by adding one new tree at each fitting iteration. Given a model M, the objective function $\mathcal{L}(M)$ is the sum of a loss term $L(M)$, which measures the accuracy of the model on the given dataset, and a regularisation term $\Omega(M)$, that prevents overfitting by restraining the model complexity. The loss function L is the sum of losses on each training set point $l(y_i, \hat{y}_i)$: a typical choice of loss function for a regression problem is the sum of squared errors

$$L(M) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) = \sum_{i=1}^{N} |y_i - \hat{y}_i|^2\ . \tag{2.20}$$

The regularisation term is the sum of complexities of each tree f_k. A tree is a function $f(\mathbf{x})$

$$f(\mathbf{x}) = w_{q(\mathbf{x})} \tag{2.21}$$

that assigns a score $w_{q(\mathbf{x})}$ to each of its T leaves by means of a set of bisection rules q for a given input point $\mathbf{x}$. In the XGBoost library, the complexity of a tree f is defined as

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2\,, \tag{2.22}$$

where γ and λ are coefficients regulating the cost of the number of leaves of a tree and their scores respectively; the complexity of a set of trees is trivially the sum of complexities of each tree. Starting from one tree, at each iteration t of the additive training procedure the aim is to minimise the objective function

$$\begin{aligned}\mathcal{L}^{(t)} &= L\left(M^{(t)}\right) + \Omega\left(M^{(t)}\right) \\ &= \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{j=1}^{t} \Omega(f_j) \\ &= \sum_{i=1}^{N} l\left(y_i, \sum_{j=1}^{t} f_j(\mathbf{x}_i)\right) + \sum_{j=1}^{t} \Omega(f_j)\,.\end{aligned} \tag{2.23}$$

The variation $\Delta\mathcal{L}^{(t)}$ of the objective function upon addition of a new tree at step t is then given by

$$\Delta\mathcal{L}^{(t)} = \sum_{i=1}^{N} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)\right) + \gamma T^{(t)} + \frac{1}{2}\lambda \sum_{j=1}^{T^{(t)}} \left(w_j^{(t)}\right)^2 . \tag{2.24}$$

Let us now expand the loss function in Taylor series around $\hat{y}_i^{(t-1)}$ and truncating at the second order in $f_t(\mathbf{x}_i)$. Denoting its gradient and Hessian as

$$\begin{aligned} g_i &= \frac{\partial l\left(y_i, \hat{y}_i\right)}{\partial \hat{y}_i}\Big|_{\hat{y}_i = \hat{y}_i^{(t-1)}} \\ h_i &= \frac{\partial^2 l\left(y_i, \hat{y}_i\right)}{\partial \hat{y}_i^2}\Big|_{\hat{y}_i = \hat{y}_i^{(t-1)}} \end{aligned} \tag{2.25}$$

we can write

$$l\left(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)\right) \simeq l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t(\mathbf{x}_i)^2\,. \tag{2.26}$$

Dropping the superscript $(t-1)$ and (t) for clarity, the objective function variation can be approximated with

$$\begin{aligned}\Delta\mathcal{L} &\simeq \sum_{i=1}^{N}\left(g_i f_t(\mathbf{x}_i)+\frac{1}{2}h_i f_t(\mathbf{x}_i)^2\right)+\gamma T+\frac{1}{2}\lambda\sum_{j=1}^{T}w_j^2 \\ &= \sum_{j=1}^{T}\left[w_j\left(\sum_{i\in I_j}g_i\right)+\frac{1}{2}w_j^2\left(\lambda+\sum_{i\in I_j}h_i\right)\right]+\gamma T \\ &= \sum_{j=1}^{T}\left[w_j G_j+\frac{1}{2}w_j^2\left(\lambda+H_j\right)\right]+\gamma T\,,\end{aligned} \tag{2.27}$$

where we denote as $I_j=\{i\,|\,q(\mathbf{x}_i)=j\}$ the *instance set* of node j – that is, the set of data points indices that get assigned to the leaf j of the tree – added at the current step; $\{G_j\}$ and $\{H_j\}$ are the summations of gradients and Hessians on each leaf. For a given new tree structure $q(\mathbf{x})$, minimising Eq. 2.27 yields the optimal leaves scores

$$w_j^*=-G_j/(\lambda+H_j) \tag{2.28}$$

and the corresponding gain in objective function

$$\mathcal{L}^*=-\frac{1}{2}\sum_{j=1}^{T}G_j^2/\left(\lambda+H_j\right)+\gamma T \tag{2.29}$$

The construction of a tree is performed in a greedy bottom–up approach one level at a time, where each leaf node belonging to the current level produces two new leaves so that the binary split produces the highest gain in objective function. Specifically, given one leaf node of instance set I splitting into left (L) and right (R) nodes of instance sets I_R and I_L ($I_R\cup I_L=I$), the objective function gain is

$$\Delta\mathcal{L}_{\text{split}}=\frac{1}{2}\left[\frac{\left(\sum_{i\in I_L}g_i\right)^2}{\lambda+\sum_{i\in I_L}h_i}+\frac{\left(\sum_{i\in I_R}g_i\right)^2}{\lambda+\sum_{i\in I_L}h_i}-\frac{\left(\sum_{i\in I}g_i\right)^2}{\lambda+\sum_{i\in I}h_i}\right]-\gamma\,. \tag{2.30}$$

Equation 2.30 gives a way to evaluate the training gain for some given split points of the new tree structure while it is built.

Engineering the algorithm that produces the candidate split points that best suits the dataset and computing architecture available is a field of open research; for a thorough overview of the available methods we refer to Ref. [15].

To further prevent overfitting, the XGBoost library implements two more crucial measures. The first is shrinkage, first proposed by Friedman [16], which consists of rescaling the weights of a newly added tree at each step by a factor η. The second is feature subsampling, which consists of restricting the search of the optimal leaf split during the tree building routine to a random subset of data features.

The resulting method is very stable against overfitting and is able to discern complex patterns in data with an effectiveness that in some applications is comparable to deep neural networks, while also scaling linearly with the number of training data points [17, 18]. A drawback of this kind of models is the large number of hyperparameters that need to be tuned to obtain an optimal accuracy. Another possible disadvantage of GBRTs in some applications is that the time complexity of the prediction operation scales linearly with the complexity of the model, since in ensemble learning paradigm the prediction on a test point involves gathering the predictions of each of the trees it is composed of.

2.2 Unsupervised Learning

The goal of unsupervised learning is, in broad terms, to build a representation for an unlabelled dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ by finding structure and patterns within it. Clustering methods belong to this class. In a clustering problem, one wishes to divide a dataset $\mathbf{X}$ into a (possibly not predefined) number k of subsets (also named clusters)

$$\{\mathbf{S}_1, \dots, \mathbf{S}_k\}\,, \quad \bigcup_i \mathbf{S}_i = \mathbf{X}\,, \quad \bigcap_i \mathbf{S}_i = \varnothing \tag{2.31}$$

so that, upon a definition of similarity measure, all elements belonging to a subset are most similar to each other than to any other element of another subset. This statement can be expressed in terms of a multi–objective optimisation problem, where one wants to minimise the inter–cluster distance and maximise the intra–cluster distance for all clusters.

K-Means Clustering

Of the variety of algorithms that have been devised to address the clustering problem, one of the simplest, fastest and thus most widely used is the so called k–means method [19]. The method requires the desired number of clusters K as input and utilises an iterative algorithm that minimises the *inertia* of the clusters defined as

$$\sum_{i=1}^{K} \sum_{\mathbf{x}_j \in \mathbf{S}_i} |\mathbf{x}_j - \boldsymbol{\mu}_i|^2\,, \tag{2.32}$$

where $\boldsymbol{\mu}_i$ is the centre of cluster $\mathbf{S}_i$:

$$\boldsymbol{\mu}_i = \frac{1}{|\mathbf{S}_i|} \sum_{\mathbf{x}_j \in \mathbf{S}_i} \mathbf{x}_j\,. \tag{2.33}$$

In the k–medoid variant of the method, the centre of the cluster is substituted with the data point closest to it. The most basic implementation of k–means is known as Lloyd's algorithm [20].

The starting point is an initial random guess for the cluster centres, and each data point is assigned to its closest centre. At each iteration, then, the centres are evaluated again and the data points labels are updated accordingly until convergence: the algorithm is guaranteed to converge to a local minimum in a finite number of steps, but is typically terminated

within a predefined maximum number of steps. A more advanced algorithm [21] allows a much faster convergence by carefully seeding the initial guesses for the cluster centres.

Regardless of the specific implementation, k–means is typically used when data points are proper vectors $\mathbf{x} \in \mathbb{R}^M$ that lie in an Euclidean space, and provides quick solutions for large data sets and a medium number (tens) of clusters. Since the algorithm minimises inertia, the resulting clusters are bound to be convex and of relatively similar size by construction.

Spectral Clustering

Another popular method is the spectral partitioning algorithm, which is instead based on graph theory concepts. Let $G(V, E)$ be an undirected simple graph constituted by a set of N nodes or vertices V interconnected by a set of pairwise links named edges E determined by the *weight matrix* $\mathbf{W}$:

$$\mathbf{W} = (w_{ij})_{i,j=1,\dots,N},\ w_{ij} = w_{ji} \geq 0. \tag{2.34}$$

The matrix $\mathbf{W}$ determines the connectivity of the graph as it has nonzero entries when an edge connect two nodes; in another perspective, it can be viewed as a measure of similarity between pairs of nodes. The simplest form of weight matrix is the adjacency matrix, whose entries are either 0 or 1 depending on whether two nodes are first neighbours and thus form an edge:

$$w_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are neighbours} \\ 0 & \text{otherwise} \end{cases} \tag{2.35}$$

When the nodes in a graph have Cartesian coordinates, a popular choice of weight matrix definition is to use a Gaussian kernel distance between points

$$w_{ij} = \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{2\sigma^2}\right)\ ; \tag{2.36}$$

in this case the weight matrix is equivalent to the covariance matrix encountered in Gaussian process regression. Let us now define the *degree matrix* as a diagonal matrix $\mathbf{D} = (d_{ij})_{i,j=1\dots,N}$ where

$$d_{ii} = \sum_{j=1}^{N} w_{ij}\ . \tag{2.37}$$

The unnormalised *Laplacian matrix*, the discrete analogue of the Laplacian operator in physics, is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W}\,, \tag{2.38}$$

is symmetric and positive semi–definite, and has N real non–negative eigenvalues. It can be shown that the lowest eigenvalue is always 0 and its multiplicity is equal to the number of connected components of the graph. In a connected graph the zero eigenvalue corresponds to the constant vector $(1,\dots,1)^T \in \mathbb{R}^N$; in a disconnected graph of k connected components $\{\mathbf{S}_1,\dots,\mathbf{S}_k\}$, the eigenvectors of the eigenvalue 0 correspond to the indicator vectors of each connected component $\{\mathbf{v}_{\mathbf{S}_1},\dots,\mathbf{v}_{\mathbf{S}_k}\}$ defined as

$$\mathbf{v}_{\mathbf{S}_{i,j}} = \begin{cases} 1 & \text{if } j \in \mathbf{S}_i \\ 0 & \text{otherwise}\,. \end{cases} \tag{2.39}$$

The main concepts of spectral clustering are based around two main papers, both dated 1973: Donath and Hoffman [22] proposed partitioning a graph based on the eigenvectors of its adjacency matrix; simultaneously, Fiedler [23, 24] proved that the eigenvector corresponding to the smallest nonzero eigenvalue of the Laplacian matrix (from then on named Fiedler's vector) can be used to bisect a connected graph. From these initial ideas a variety of algorithms have been devised, and with them several definitions of Laplacian matrix – for a detailed overview of the different implementations see Ref. [25].

Perhaps the simplest algorithm is the unnormalised spectral clustering algorithm, which works as following:

- Given a unnormalised Laplacian matrix L, compute its lowest k eigenvectors $\{\mathbf{v}_1,\dots,\mathbf{v}_k\}$
- Define $\mathbf{V} = (\mathbf{v}_1,\dots,\mathbf{v}_k) \in \mathbb{R}^{N\times k}$, and $\mathbf{y}_i \in \mathbb{R}^k$ as the i–th row of $\mathbf{V}$
- Apply a clustering algorithm such as k–means to the transformed data points $\{\mathbf{y}_1,\dots,\mathbf{y}_N\}$ and obtain the k subgraphs $\{\mathbf{S}_1,\dots,\mathbf{S}_k\}$.

The main peculiarity of spectral clustering is the freedom to pick any similarity measure for the data points to construct the Laplacian matrix. Because of this flexibility, this method can be advantageous when the geometry of the graph is non-compact and when data points do not lie in an Euclidean space. Since the method requires the evaluation of the whole weight matrix (which can be confused with a covariance matrix) and the evaluation of the lowest k eigenvectors, it is comparatively more expensive than the k–means algorithm and it can only be applied to relatively small datasets and small number of clusters.

Chapter 3

Atomistic Simulations

3.1 Electronic Structure Calculations

Quantum Mechanics (QM) is the most accurate theory to describe matter at the atomic level. It postulates that any physical system is entirely described by the wavefunction Ψ of its fundamental particles. The wavefunction must obey the many–body Schrödinger equation, which in its time–independent form states that

$$\hat{H}|\Psi\rangle = E|\Psi\rangle \tag{3.1}$$

where $\hat{H}$ is the Hamiltonian operator of the system and E is the groundstate eigenvalue of the total energy. If the multi–body equation describes a system of N interacting electrons with positions $\{\mathbf{r}_i\}_{i=1,\dots,N}$ and M nuclei at positions $\{\mathbf{R}_\alpha\}_{\alpha=1,\dots,M}$ of , the non–relativistic Hamiltonian operator is

$$\begin{aligned}\hat{H} &= \hat{T}_e + \hat{T}_n + \hat{V}_{ee} + \hat{V}_{nn} + \hat{V}_{en} \\ &= -\frac{1}{2}\sum_{i=1}^{N}\frac{\hbar^2}{m_e}\nabla_i^2 - \frac{1}{2}\sum_{\alpha=1}^{M}\frac{\hbar^2}{m_\alpha}\nabla_\alpha^2 + \\ &+ \sum_{i<j}^{N}\frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{\alpha<\beta}^{M} Z_\alpha Z_\beta \frac{e^2}{|\mathbf{R}_\alpha - \mathbf{R}_\beta|} - \sum_{i=1}^{N}\sum_{\alpha=1}^{M} Z_\alpha \frac{e^2}{|\mathbf{R}_\alpha - \mathbf{r}_i|}\end{aligned} \tag{3.2}$$

where m_e and e are the electron rest mass and charge, m_α and Z_α are the atomic mass and number of the α^{th} nucleus. The first two terms of Eq. 3.2 represent the kinetic energy of electrons and nuclei, the second two terms are the electron-electron and nucleus-nucleus electrostatic repulsion terms, and the last one is the electrostatic attraction between nuclei

and electrons. Solving Eq. 3.1 provides the wavefunction Ψ from which any observable can be computed, but for any practical case it does not have an analytic solution. Its solution can then be found numerically by considering that the groundstate energy must satisfy the variational principle

$$E = \min_{\Psi} \langle \Psi | \hat{H} | \Psi \rangle \,. \tag{3.3}$$

In order for the method to be computationally viable, several approximations and methods have been developed.

Born–Oppenheimer Approximation

The first simplification, ubiquitous in most quantum mechanical computer simulations, is known as the Born–Oppenheimer approximation or adiabatic approximation [26]. As the latter name suggest, one assumes that nuclear and electronic motions are decoupled and do not exchange energy. This is because the mass ratio between nuclei and electrons is such that their motion is on two different time scales, with the electrons moving as if the nuclei were fixed and thus always falling into their lowest energy eigenstate.

From the point of view of nuclei, for each nuclear configuration electrons generate a charge distribution given by the groundstate of the electron wavefunction. In other words, the Born-Oppenheimer approximation states that the total wavefunction can be factorised in two parts, a function Φ of the electron coordinates at fixed nuclei positions and a function Ψ of the nuclei positions

$$\Psi(\mathbf{R}, \mathbf{r}) = \Phi(\mathbf{r}|\mathbf{R}) \times \chi(\mathbf{R}) \,. \tag{3.4}$$

Leaving Ψ out, the electronic Schrödinger equation becomes

$$(\hat{T}_e + \hat{V}_{ee} + \hat{V}_{en})\, \Phi(\mathbf{r}|\mathbf{R}) = \varepsilon(\mathbf{R})\, \Phi(\mathbf{r}|\mathbf{R}) \,. \tag{3.5}$$

Here, $\hat{V}_{en}$ acts as a one-body external potential on the electrons moving around fixed nuclei. The decoupled motion of the nuclei, instead, is governed by the potential energy surface $\varepsilon(\mathbf{R})$. Very often, in atomistic simulations nuclei are considered as classical particles obeying Newton's equations of motion, so the problem becomes solving the electronic Schrödinger equation in Eq. 3.5.

Density functional Theory

Density Functional Theory (DFT) is a theory proposed by Hohenberg, Kohn and Sham [27, 28] which allows any ground state physical variable of a system of interacting electrons to be unambiguously expressed in terms of electron density. This greatly reduces the dimensionality of the many-body problem expressed in Eq. 3.5 to a problem of one effective particle, the electron density, defined as

$$n(\mathbf{r}) = \int |\Phi(\mathbf{r}, \mathbf{r}_2, \ldots, \mathbf{r}_N)|^2 \, d\mathbf{r}_2 d\mathbf{r}_3 \ldots d\mathbf{r}_N \,. \tag{3.6}$$

The first Hohenberg-Kohn theorem states that, given a system of N interacting electrons immersed in an external potential, the ground state electron density $n_0(\mathbf{r})$ uniquely identifies its Hamiltonian, therefore every observable of the system can be expressed as a functional of $n_0(\mathbf{r})$.
The second Hohenberg–Kohn theorem states that the global minimum of the total energy, here E_0, must be the ground state of the system, so the problem can be expressed as a minimisation of the energy functional with respect to the electron density

$$E_0 = \min_{\{n\}} E[n(\mathbf{r})] \tag{3.7}$$

subject to the physical constraints

$$\begin{aligned} n(\mathbf{r}) &\geq 0 \quad \forall \, \mathbf{r} \in \mathbb{R}^N \\ \int d\mathbf{r} \, n(\mathbf{r}) &= N \,. \end{aligned} \tag{3.8}$$

From the electronic Schrödinger equation in Eq. 3.5 we see that the total energy functional, called Kohn-Sham energy functional, is

$$E[n] = T_e[n] + E_{\text{ext}}[n] + E_{ee}[n] \,. \tag{3.9}$$

It is composed of a kinetic term T_e, an interaction with the external potential E_{ext}, and an electron–electron interaction E_{ee}. While the external potential contribution is trivially expressed as

$$E_{\text{ext}}[n] = \int d\mathbf{r} \, V_{\text{ext}}(\mathbf{r}) \, n(\mathbf{r}) \,, \tag{3.10}$$

the two other contributions are unknown. To make the method usable, Kohn and Sham proposed to consider a fictitious system of N non–interacting electrons sharing the same

groundstate density n_0 of the real system. If we expand the wavefunction of the non–interacting system on an orthonormal and antisymmetric basis set $\{|\phi_i\rangle\}_{i=1,\dots,N}$, the electron density is

$$n(\mathbf{r}) = \sum_{i=1}^{N} |\phi_i(\mathbf{r})|^2 \,, \tag{3.11}$$

and its kinetic energy, which is an approximation of the one of the real interacting system, can be expressed exactly:

$$T^{ni}[n] = -\frac{1}{2}\sum_{i=1}^{N} \langle\phi_i|\nabla_i^2|\phi_i\rangle \,. \tag{3.12}$$

The electron–electron interaction E_{ee}, on the other hand, can be further expanded into two parts. The first and often dominant contribution is the classical Coulomb energy, also known as Hartree energy

$$E_H[n(\mathbf{r})] = \frac{1}{2}\int d\mathbf{r}_1 d\mathbf{r}_2 \, \frac{n(\mathbf{r}_1)n(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} \,; \tag{3.13}$$

the second contribution is the exchange–correlation energy E_{XC}, which contains all multi–body interactions and the error arising from approximating the kinetic energy term. Rewriting the total energy functional we obtain

$$E = T^{ni} + E_{\text{ext}} + E_H + E_{XC} = T^{ni} + E_{KS} \,, \tag{3.14}$$

where E_{KS} is the so called Kohn–Sham energy functional.
By applying the variational principle in Eq. 3.7 to the Eq. 3.14 and considering a charge density arising from a wavefunction projected onto the basis set $\{|\phi_i\rangle\}$, the electronic groundstate can be found by solving the system of N Kohn–Sham equations

$$\left(-\frac{1}{2}\nabla_i^2 + V_{KS}\right)\phi_i = \varepsilon_i\phi_i \,, \tag{3.15}$$

where V_{KS} is the functional derivative of the energy functional:

$$V_{KS}(\mathbf{r}) = \frac{\delta E_{KS}}{\delta n(\mathbf{r})} \,. \tag{3.16}$$

Solving the Schrödinger equation for N interacting electrons has been simplified into the problem of minimising the Kohn-Sham energy functional of N non-interacting electrons with respect to each single particle wavefunction.

The solution of the system is performed in an iterative way: starting from an initial guess for the density, one calculates the corresponding Kohn–Sham potential arising from the charge distribution. The charge density is then updated and the algorithm loops until self–consistence is achieved, *i.e.*, until the charge density is consistent with the field.

In DFT, the exchange–correlation term in the equations remains the only unknown functional of n, and, over the years, several approximations have been implemented to estimate it and minimise the approximation errors. The simplest and still widely used approximation is the so called Local Density Approximation (LDA), in which the exchange and correlation energies are approximated with those calculated with highly accurate Quantum Monte Carlo methods for a uniform electron gas at different densities. Being parametrised on constant densities only, LDA is expected to be accurate enough when the electronic density varies slowly, *e.g.*, in solid state systems. Situations when density gradients are large (*e.g.*, in the case of molecular systems), instead require the use of a more complex family of functionals known as Generalized Gradient Approximation (GGA) functionals which are parametrised as a function of both the electronic density and its gradient. Among the latter category, the Perdew–Burke–Ernzerhof (PBE) exchange–correlation functionals [29] are often considered the most appropriate choice for the electronic structure calculation of solids.

Hellmann–Feynman Theorem

Let us consider a system with a Hamiltonian $H(\lambda)$ that depends on a parameter λ. The Hellmann–Feynman theorem [30] states that the derivative of the total energy with respect to the parameter is simply the expectation value of the derivative of the Hamiltonian with respect to that same parameter. Namely:

$$\frac{dE_\lambda}{d\lambda} = \langle \Psi_\lambda | \frac{d\hat{H}}{d\lambda} | \Psi_\lambda \rangle \,. \tag{3.17}$$

In atomistic simulations, this theorem allows us to directly compute the quantum mechanical forces acting on nuclei without resorting to finite differences approximate methods. Using the position $\mathbf{R}_j$ of a nucleus j as parameter and differentiating the total energy expressed in Eq. 3.14 calculated for the groundstate wavefunction solution of Eq. 3.15, the force $\mathbf{f}_j$

acting on j is:

$$\begin{aligned}\mathbf{f}_j &= -\frac{\partial E}{\partial \mathbf{R}_j} = -\frac{\partial E_{KS}}{\partial \mathbf{R}_j} \\ &= -\langle\Phi|\frac{\partial \hat{H}_{KS}}{\partial \mathbf{R}_j}|\Phi\rangle \\ &= -\sum_{i=1}^{N}\langle\phi_i|\frac{\partial \hat{H}_{KS}}{\partial \mathbf{R}_j}|\phi_i\rangle\,.\end{aligned} \tag{3.18}$$

3.2 Classical Interatomic Potentials

Despite the simplifications introduced by DFT and other *ab initio* methods, calculating observables at quantum mechanical accuracy is in many cases still too computationally expensive. If the aim of a simulation is observing the qualitative behaviour of a system or if no exotic phenomena are expected to occur (for instance, the evaluation of bulk properties of a material), being able to evaluate forces and energies at the highest achievable accuracy for any atomic configuration in the phase space of the system is probably not *necessary*. It is then conceivable to approximate the 'correct' Born—Oppenheimer potential energy surface given by electronic structure calculations by making assumptions regarding the interaction between atoms and restricting it to an analytical function. Such functions are known as Interatomic Potentials (IPs).

An interatomic potential consists of an analytic and differentiable function of the atomic coordinates, whose form is often designed according to physical considerations and intuition. Its parameters are fitted to reproduce some target observables (*e.g.*, lattice constants, elastic constants, etc.) as accurately as possible, the targets being experimental data or precise quantum mechanical simulations.
Simplification, though, comes at the cost of transferability: an IP should only be used to evaluate the material and the type of configurations for which it has been designed and fitted. This lack of transferability is the fundamental driving force for the proliferation of a wide range of IPs, all of which share some basic building blocks which we will now describe.

The total energy of a system E_{tot} can be expressed as the sum of a short–range component E_{SR} associated with covalent interaction and a long–range component E_{LR} associated with electrostatic and van der Waals interaction:

$$E_{\text{tot}} = E_{\text{SR}} + E_{\text{LR}}\,. \tag{3.19}$$

The short–range part is an expansion of a multi–body interaction truncated at the n^{th} term: $n = 2$ if only pair interactions are included, $n = 3$ when also comprising bonding angle terms and $n = 4$ when dihedral angles terms are added as well. Because this type of interaction can be mainly attributed to Pauli electron repulsion at small distances and Coulomb attraction between neighbouring screened nuclei, it quickly decays with the distance and hence the name.
The long–range part, instead, is the electrostatic energy of the multipole configuration in the system, and inherently decays much more slowly with the distance. Purely covalent crystals (such as silicon) and metals can be appropriately modelled by considering short–range contributions only, although for different reasons. Covalent crystals are characterised by very localised electrons and charge neutral atomic environments, thus no electrostatic effects are expected. Conversely, in metallic systems the electrostatic interaction between ions is completely screened by the conduction electrons. In oxide materials, instead, the concurrence of local charge unbalance and localised electrons results in interactions for which the electrostatic component is crucial.

Tangney–Scandolo Potential

The Tangney–Scandolo iteratomic potential (TS) has been first proposed in 2002 [31] for the simulation of silica, and it has been optimised and extended for its use on different oxides since then [32, 33]. In oxide systems, a perturbation of the positions induces a deformation of the electron density around oxygen ions. This is an essential phenomenological feature to understand the atomic interactions and can be effectively modelled by dipoles placed at each oxygen site. Upon this consideration, the TS potential is composed of a short–range pair potential and a long–range electrostatic interaction between all charges and oxygen dipoles:

$$E_{\text{tot}} = U^{\text{SR}} + U^{\text{el}} \,. \tag{3.20}$$

The short–range interaction is described by a Morse–Stretch two–body potential, which for each pair of atoms i and j has the form

$$U_{ij}^{\text{SR}} = D_{ij} \left(\exp\left(\gamma_{ij} \frac{r_{ij}^0 - r_{ij}}{r_{ij}^0} \right) - 2 \exp\left(\gamma_{ij} \frac{r_{ij}^0 - r_{ij}}{2 r_{ij}^0} \right) \right) \,. \tag{3.21}$$

Here, r_{ij}^0, D_{ij} and γ_{ij} are parameters of the model that depend on the species of atoms i and j, and control the equilibrium bondlength, the dissociation energy and the width of the potential well respectively. All pairs of atoms whose distance is larger than a threshold

$r_{\text{cutoff}}^{\text{MS}}$ (18 a.u. $\simeq$ 9.5 Å in parametrisations for SiO_2) are considered as non–interacting, thus yielding a sparse interaction matrix.
If we truncate the electric multipole expansion at dipole order, the energy contribution given by electrostatic interaction is, according to the superposition principle, the sum of all pairwise interactions (charge–charge, charge–dipole and dipole–dipole) plus an additional term representing the energy necessary to polarise a dipole:

$$\begin{aligned} U^{\text{el}} &= U_{\text{pair}}^{\text{el}} + U^{\text{pol}} \\ U^{\text{pol}} &= \sum_i \frac{|\mathbf{p}_i|^2}{2\alpha_i} \\ U_{\text{pair}}^{\text{el}} &= \frac{1}{2}\sum_i \sum_{j\neq i} \left[\frac{q_i q_j}{r_{ij}} + \frac{1}{r_{ij}^3}\left(q_i\,\mathbf{p}_i\cdot\mathbf{r}_{ij} - q_j\,\mathbf{p}_j\cdot\mathbf{r}_{ij}\right) + \right. \\ &\quad \left. + \frac{1}{r_{ij}^3}\left(\mathbf{p}_i\cdot\mathbf{p}_j - \frac{3}{r_{ij}^2}\left(\mathbf{p}_i\cdot\mathbf{r}_{ij}\right)\left(\mathbf{p}_j\cdot\mathbf{r}_{ij}\right)\right)\right]. \end{aligned} \tag{3.22}$$

The charges q_i and polarisabilities α_i of each atomic species i are the parameters of this model, where α_{O} is nonzero in all parametrisations.
According to Madden *et al.* [34], the dipole moments $\mathbf{p}_i$ appearing in the above equations are induced not only by the point values of electric field and its gradient due to the charges and dipoles of the other ions, but also by the confining effect on the electrons around the oxygen ions of the neighbouring ions that varies according to the local distortion of the crystal structure. This short–range correction is given by

$$\mathbf{p}_i^{\text{SR}} = \alpha_i \sum_{j\neq i} q_i \frac{\mathbf{r}_{ij}}{r_{ij}^3}\, g_{ij}(r_{ij})\,, \tag{3.23}$$

where

$$g_{ij}(r_{ij}) = c_{ij}\, e^{-b_{ij} r_{ij}} \sum_k^4 \frac{(b_{ij} r_{ij})^k}{k!}\,. \tag{3.24}$$

The coefficients c_{ij} and b_{ij} are the species–dependent parameters of the model, and for an adequate parametrisation it can be shown that the summation of Eq. 3.23 can be truncated after the second neighbours.

The dipole moments $\{\mathbf{p}_i\}$ of the oxygen ions are obtained by solving until self–consistency the system of equations

$$\mathbf{p}_i^{(n)} = \alpha_i \mathbf{E}(\mathbf{r}_i;\, \{\mathbf{p}_j^{(n-1)}\}_{j\neq i}, \{\mathbf{r}_j\}_{j\neq i}) + \mathbf{p}_i^{\text{SR}}\,, \tag{3.25}$$

where for each ion i the dipole moment at iteration n depends on the electric field $\mathbf{E}(\mathbf{r}_i)$, which in turn is determined by the charges and dipoles of all other ions at iteration $n-1$. In periodic systems, the typical method to obtain the full electrostatic energy U^{el} is to evaluate the sum of all pairwise contributions by means of the Ewald summation scheme, which is computationally expensive and poses scalability issues for large systems.
A recent advance [35] has instead proposed a method to make the electrostatic interaction short–ranged as well, thus achieving linear scaling. Namely, the electrostatic interaction between each couple of ions i and j summing up to $U^{\mathrm{el}}_{\mathrm{pair}}$ in Eq. 3.22 is multiplied by a Yukawa screening function

$$f_Y(r_{ij}) = e^{-\beta r_{ij}} f_c(r_{ij}) \,, \tag{3.26}$$

where β is the inverse of the typical length scale and f_c is a simple cutoff function whose requirements are to be smooth, to be zero beyond a certain cutoff length r_{c} and to have zero derivative at r_{c}. This modification effectively switches off interactions beyond a distance $r_{\mathrm{c}} \sim 20$ a.u. $\simeq 10.5$ Å, and is physically justified by noticing that, contrary to a completely arbitrary charge distribution, in solid state materials the electrostatic interaction between ions is largely screened by the electron density in between them, and very long–range contributions only play a negligible role.

3.3 Machine Learning Potentials

Over the last decade, a new class of atomic interaction models built upon machine learning methods has been introduced. The common idea shared by all potentials of this class is the following: instead of restricting the functional form of the atomic interaction *a priori* and fitting its parameters to best reproduce a dataset of accurate QM calculations, leading to the already mentioned transferability issue of classical IPs, we now wish to lift this constraint and let the physical data itself choose the generating model that best describes it. With a more open choice of fitting model, the aim is to extend its range of applicability and to reproduce quantum mechanical data more accurately.
As extensively argued in Bartók *et al.* [36], the dominant factor that determines the quality of a model is the representation of the chemical environment rather than the specific machine learning method deployed. In other words, transforming a raw input data point representing an atomic configuration (*e.g.*, Cartesian coordinates, charges, dipoles) into features that unequivocally and efficiently describe it is the key step to make a ML algorithm able to accurately fit it.

The range of proposed methods and applications is wide: notable examples are Gaussian process regression to fit a locally–defined atomic energy as a function of the bispectrum or similarly derived symmetry–invariant properties of the Cartesian coordinates [36, 37], neural networks to fit a locally–defined atomic energy as a function of some symmetry functions of the local atomic environment [38], direct Gaussian process regression fitting of the three dimensional atomic force acting on an atom from a set of internally defined vectors [39], kernel ridge regression fitting of atomisation energies and other molecular properties of small organic molecules as a function of the (sorted) eigenvalues of their Coulomb matrix [40]. We will now revise a method that is relevant for this thesis.

Gaussian Approximation Potentials

The aim of a Gaussian Approximation Potential (GAP) [37] is to automatically generate an interatomic potential by fitting a data set of *ab initio* configurations with a machine learning model. The functional form is not restricted as in the case of standard force fields, and only some properties such as smoothness and periodicity are defined by the kernel choice. Therefore, one could in principle achieve an arbitrary precision provided that the learning set contains enough information. By constraining ourselves to model the short–range component of the complete interaction, the basic assumption is to express the potential energy of a system as a sum of atomic energies

$$U^{\mathrm{SR}} = \sum_{i}^{\mathrm{atoms}} \varepsilon\left(\mathbf{q}_i\right) = \sum_{i}^{\mathrm{atoms}} \varepsilon_i \,, \tag{3.27}$$

where each $\mathbf{q}_i$ is some feature vector describing the local environment around atom i. The atomic energies ε_i are an abstraction rather than physical quantities, and are obtained by processing total energies and atomic forces from a dataset of *ab initio* calculations: a detailed derivation of how to use these data sources in a GPR scheme, here omitted because it is beyond the scope of this thesis, can be found in Ref. [41] and in the supplemental material of Ref. [37].

This data is then fitted by a Gaussian Process model, which for any given system configuration $\{\mathbf{q}_i^*\}$ will predict the expected value of each atomic energy $\varepsilon\left(\mathbf{q}_i^*\right)$ summing up to the predicted total energy $U^{\mathrm{SR}}\left(\{\mathbf{q}_i^*\}\right)$ using the standard formalism of kernel methods

$$\varepsilon(\mathbf{q}_i^*) = \sum_{j=1}^{M} \alpha_j k(\mathbf{q}_i^*, \mathbf{q}_j) \,, \tag{3.28}$$

where the index j runs over the number of training set configurations M, α_j and $\mathbf{q}_j$ are respectively the regression coefficient and the descriptor for the training set configuration j, and k is a Gaussian kernel function.

The fundamental components that determine the success or failure of any machine learning method is designing an appropriate descriptor and defining a suitable metric. All variants of the GAP algorithm share the same key idea: building a representation of an atomic neighbourhood that is meaningful to model ε_i involves creating a descriptor that is intrinsically invariant with respect to the same symmetry operations of an energy quantity: rigid translation, rotation and permutation of same–species atoms. An energy quantity is also invariant with respect to reflection operations, but will be omitted in this section as the method described does not imply it by design. We will now sketch the construction of a descriptor, or rather a kernel function directly, for the latest development of GAP descriptors named Smooth Overlap of Atomic Positions (SOAP). Let us first express a local environment around an atom i as a sum of Gaussian functions of some width σ_{blur} centred on atomic positions

$$\rho_i(\mathbf{r}) = \sum_{j \in \text{atoms}} \exp\left(\frac{|\mathbf{r} - \mathbf{r}_{ij}|^2}{2\sigma_{\text{blur}}^2}\right) f_{\text{cut}}(r_{ij})\,, \tag{3.29}$$

where the sum is truncated by the cutoff function f_{cut} at a given length scale r_{cut} specific of the system type. The exact form of this cutoff function is rather arbitrary, as its only constraint is not to introduce artificial discontinuities in the interaction. Desirable properties are then to smoothly decrease as the cutoff distance is approached and to be 0 and have 0 derivative at r_{cut}. In the latest version of GAP its definition is

$$f_{\text{cut}}(r) = \begin{cases} 1 & \text{if } r \leq r_{\text{cut}} - d \\ \frac{1}{2}\left(1 + \cos\left(\pi(r - r_{\text{cut}} + d)/d\right)\right) & \text{if } r_{\text{cut}} - d < r < r_{\text{cut}} \\ 0 & \text{if } r \geq r_{\text{cut}} \end{cases}, \tag{3.30}$$

where d is a transition width typically set to 1 Å. If we expand the angular part of the atomic density function ρ in Eq. 3.29 onto a basis set of spherical harmonics Y_{lm} truncated at some angular momentum index $l_{\max}$, we obtain

$$\rho(\mathbf{r}) = \sum_{j \in \text{atoms}} \sum_{l=0}^{l_{\max}} \sum_{m=-l}^{l} c_{lm}^{j}(r) Y_{lm}(\Omega) \tag{3.31}$$

where the factors $c^j_{lm}(r)$ are rotationally invariant and only depend on the radial coordinate and Ω is a solid angle. The radial part of the atomic density, encoded in the coefficients $c^j_{lm}(r)$, is itself projected onto an orthonormal basis set of $n_{\max}$ Gaussian functions $\{g_1(r), \ldots, g_{n_{\max}}(r)\}$

$$\sum_{j \in \text{atoms}} c^j_{lm}(r) = \sum_{j \in \text{atoms}} \sum_{n=1}^{n_{\max}} c^j_{nlm} g_n(r) = \sum_{n=1}^{n_{\max}} c_{nlm} g_n(r)\,, \tag{3.32}$$

so that the summation over the atom index in Eq. 3.31 remains manageable for neighbourhoods with a large number of atoms:

$$\rho(\mathbf{r}) = \sum_{n=1}^{n_{\max}} \sum_{l=0}^{l_{max}} \sum_{m=-l}^{l} c_{nlm} g_n(r) Y_{lm}(\Omega)\,. \tag{3.33}$$

In the SOAP method, the evaluation of feature vectors is bypassed completely by designing a similarity measure between pairs of local environments which directly acts as a kernel function over the raw density functions. Given two densities ρ_1 and ρ_2, we first define a similarity between them that satisfies permutation invariance:

$$S_{12} = S(\rho_1, \rho_2) = \int d\mathbf{r}\, \rho_1(\mathbf{r}) \rho_2(\mathbf{r})\,. \tag{3.34}$$

If we now integrate the above term over the space of rigid rotations, we can obtain a rotationally invariant similarity kernel:

$$\begin{aligned} k(\rho_1, \rho_2) &= \int d\hat{R}\, |S(\rho_1, \rho_2)|^n \\ &= \int d\hat{R} \left| \int d\mathbf{r}\, \rho_1(\mathbf{r}) \rho_2(\hat{R}\mathbf{r}) \right|^n , \end{aligned} \tag{3.35}$$

where n is a tuneable integer exponent. The integral in Eq. 3.34 can be expressed as a function of the rotation matrix as following:

$$S_{12}(\hat{R}) = \int d\mathbf{r}\, \rho_1^*(\mathbf{r}) \rho_2(\hat{R}\mathbf{r}) = \int d\mathbf{r}\, \rho_1^*(\mathbf{r}) \hat{R} \rho_2(\mathbf{r})\,. \tag{3.36}$$

A spherical harmonic $Y_{lm}(\Omega) = |lm\rangle$ transforms under rotation as

$$\begin{aligned}\hat{R}(\Omega)|lm\rangle &= \sum_{l'm'} |l'm'\rangle \langle l'm'|\hat{R}(\Omega)|lm\rangle \\ &= \sum_{m'} D^{l}_{m'm} |lm'\rangle ,\end{aligned} \tag{3.37}$$

where $D^{l}_{m'm}$ is a member of the Wigner D-matrices which form an irreducible representation of the solid body rigid rotations group SO(3). Remembering the orthogonality properties of the basis sets used, $S_{12}(\hat{R})$ can be evaluated analytically

$$\begin{aligned}S_{12}(\hat{R}) &= \\ &= \int dr\, d\Omega \left[\left(\sum_{nlm} c^{(1)}_{nlm} g_n(r) Y_{lm}(\Omega) \right)^{*} \left(\sum_{n'l'm'm''} c^{(2)}_{n'l'm'} g_{n'}(r) D^{l'}_{m''m'} Y_{l'm''}(\Omega) \right) \right] \\ &= \int dr\, d\Omega \sum_{nn'} \sum_{ll'mm'm''} c^{(1)*}_{nlm} c^{(2)}_{n'l'm'}\, g_n(r) g_{n'}(r)\, D^{l'}_{m''m'}(\hat{R}) Y^{*}_{lm}(\Omega) Y_{l'm''}(\Omega) \\ &= \sum_{nn'} \sum_{ll'mm'm''} D^{l'}_{m''m'}(\hat{R}) \int dr\, c^{(1)*}_{nlm} c^{(2)}_{n'l'm'}\, g_n(r) g_{n'}(r) \int d\Omega\, Y^{*}_{lm}(\Omega) Y_{l'm''}(\Omega) \\ &= \sum_{nn'} \sum_{ll'mm'm''} \delta_{ll'} \delta_{mm''} D^{l'}_{m''m'}(\hat{R})\, \delta_{nn'} c^{(1)*}_{nlm} c^{(2)}_{n'l'm'} \\ &= \sum_{lmm'} D^{l}_{mm'}(\hat{R}) \left(\sum_{n} c^{(1)*}_{nlm} c^{(2)}_{nl'm'} \right) \\ &= \sum_{lmm'} D^{l}_{mm'}(\hat{R}) I^{l}_{mm'}(\rho_1, \rho_2) ,\end{aligned} \tag{3.38}$$

having defined

$$I^{l}_{mm'}(\rho_1, \rho_2) = \sum_{n} c^{(1)*}_{nlm} c^{(2)}_{nl'm'} . \tag{3.39}$$

For a typical choice of SOAP exponent $n = 2$, thanks to the orthonormality of the Wigner D–matrices the expression in Eq. 3.35 has the analytic form

$$\begin{aligned} k(\rho_1, \rho_2) &= \int d\hat{R} \left(S_{12}(\hat{R}) \right)^* S_{12}(\hat{R}) \\ &= \int d\hat{R} \sum_{lmm'\lambda\mu\mu'} \left(I^l_{mm'}(\rho_1, \rho_2) D^l_{mm'}(\hat{R}) \right)^* I^\lambda_{\mu\mu'}(\rho_1, \rho_2) D^\lambda_{\mu\mu'}(\hat{R}) \\ &= \sum_{lmm'\lambda\mu\mu'} \left(I^l_{mm'}(\rho_1, \rho_2) \right)^* I^\lambda_{\mu\mu'}(\rho_1, \rho_2) \int d\hat{R}\, D^l_{mm'}(\hat{R}) D^\lambda_{\mu\mu'}(\hat{R}) \\ &= \sum_{lmm'\lambda\mu\mu'} \left(I^l_{mm'}(\rho_1, \rho_2) \right)^* I^\lambda_{\mu\mu'}(\rho_1, \rho_2) \delta_{l\lambda} \delta_{m\mu} \delta_{m'\mu'} \\ &= \sum_{lmm'} \left(I^l_{mm'}(\rho_1, \rho_2) \right)^* I^l_{mm'}(\rho_1, \rho_2) \,. \end{aligned} \tag{3.40}$$

and the 'SOAP kernel' is defined as

$$K(\rho_1, \rho_2) = \left(\frac{k(\rho_1, \rho_2)}{\sqrt{k(\rho_1, \rho_1)\, k(\rho_2, \rho_2)}} \right)^\zeta \quad , \zeta \in \mathbb{N}^+ \,, \tag{3.41}$$

where ζ is an additional degree of freedom of the model that, since $0 \leq K(\rho_1, \rho_2) \leq 1 \,\forall(\rho_1, \rho_2)$, allows the covariance matrix elements to be sharpened and thus increase the dissimilarity measure between configurations – it can be thought of having a similar effect of the (reciprocal of) σ_{blur} parameter on whole configurations rather than single atoms. The function K of Eq. 3.41 is the kernel choice in a Gaussian Process regression scheme, being a valid kernel function and satisfying all the properties that are desirable to measure differences in local atomic environments [36]:

- It is invariant over rigid rotations and permutations.
- The extension to multi–species systems is straightforward: in a system with more than one type of atoms, one ρ is defined for each species, with the precaution of then applying the rotational integral in Equation 3.35 to all at the same time.
- It is differentiable – necessary condition of a potential whose purpose is to evaluate forces and stresses.
- It is complete, which means that two distinct configurations always have a nonzero distance.
- It is smooth with respect to a variation of the Cartesian coordinates.

With the addition of a long–range interaction, GAP potentials using SOAP kernels are used in this thesis to evaluate bulk properties of the two main polymorphs of TiO_2.

3.4 Molecular Dynamics

Molecular dynamics (MD) is the general term describing a set of methods for evolving in time a system of atoms in a computer simulation. In these techniques nuclei are treated as classical particles obeying Newton's laws of motion, and move according to forces generated by a given interaction model such as the ones described in the previous section. The time integration of the second law of dynamics is performed through numerical methods by discretising time in intervals – 'timesteps' – of length Δt. The dynamics starts from an initial set of positions and velocities and is propagated in time by finite differences algorithms; the so called Velocity Verlet algorithm [42] is among the most commonly used, providing a precision of order $o(\Delta t^4)$ for both atomic positions and velocities:

$$\begin{aligned}
\mathbf{r}(t+\Delta t) &= \mathbf{r}(t) + \dot{\mathbf{r}}(t)\Delta t + \frac{1}{2}\ddot{\mathbf{r}}(t)\Delta t^2 \\
\dot{\mathbf{r}}(t+\Delta t/2) &= \dot{\mathbf{r}}(t) + \frac{1}{2}\ddot{\mathbf{r}}(t)\Delta t \\
\ddot{\mathbf{r}}(t+\Delta t) &= -\frac{\mathbf{f}\left[\mathbf{r}(t+\Delta t)\right]}{m} \\
\dot{\mathbf{r}}(t+\Delta t) &= \dot{\mathbf{r}}(t+\Delta t/2) + \frac{1}{2}\ddot{\mathbf{r}}(t+\Delta t)\Delta t\,.
\end{aligned} \tag{3.42}$$

An isolated system evolving according to a fixed Hamiltonian is a microcanonical ensemble, and as such is required to conserve its total energy: this gives a rationale for choosing the timestep, which has to be small enough to guarantee the conservation of energy. Often, MD simulations have to be performed in a canonical ensemble to study properties of a system at a fixed temperature T. To do so, the effect of an external thermal bath is simulated by the presence of a thermostat.

Nosé–Hoover Thermostat

The Nosé–Hoover thermostat [43, 44] is designed to simulate a canonical distribution, in which the heat bath is simulated by adding one fictitious degree of freedom of mass Q to

the system. The net effect is a viscous damping of the atomic motion

$$m_i \ddot{\mathbf{r}}_i = \mathbf{f}_i - \xi m_i \dot{\mathbf{r}}_i \,, \tag{3.43}$$

where the friction coefficient ξ evolves in time according to

$$\dot{\xi} = \frac{1}{Q}\left(\sum_{i=1}^{N} \frac{1}{2} m_i \dot{\mathbf{r}}_i^2 - \frac{3N+1}{2} k_B T\right) . \tag{3.44}$$

The choice of the parameter Q is critical as it controls the typical oscillation period of temperature, whose fluctuations can be difficult to suppress. A limitation of this thermostat is that it is not ergodic for small or stiff systems, for which the solution is to create a chain of thermostats [45].

Langevin Thermostat

The Langevin thermostat is a popular choice for canonical ensemble simulations and has been used in all of the constant–temperature simulations presented in this thesis. It simulates the heat exchange of the atomistic system with a thermal bath by means of additional atomic forces, one mimicking a viscous dissipation in a heat bath and the other a random force representing collisions with fictitious particles [46]. The total force experienced by a particle is then

$$\mathbf{f}_i = -\nabla_i U - \gamma \dot{\mathbf{r}}_i + \mathbf{f}_i^{(r)} \,, \tag{3.45}$$

where γ is a viscosity coefficient controlling the damping of atomic motion and $\mathbf{f}_i^{(r)}$ is a random force drawn from a Gaussian distribution

$$\mathbf{f}_i^{(r)} \sim N\left(\mathbf{0}, \frac{2 m_i \gamma k_B T}{\Delta t}\right) . \tag{3.46}$$

The parameter γ controls the typical timescale $\tau = 1/\gamma$ at which the thermostat operates. Too small values give vanishing thermostat forces that produce a deterministic thermostat that is non–ergodic; for too large values, instead, the thermostat forces are predominant and trajectories become Brownian–like that are not informative of the time evolution of the system.

3.5 Structural Optimisation

Given an atomistic system, one of the most common tasks to perform is to find its equilibrium structure, possibly subject to a set of constraints, corresponding to the configuration of minimum energy. Structural optimisation can be limited to finding the optimal Cartesian coordinates of atoms and is commonly known as geometry optimisation, or to finding the optimal unit cell parameters of a periodic system, in which case it is called cell optimisation, or a combination of the two. In the case of geometry optimisation (*e.g.*, isolated molecules and aperiodic structures, or periodic structures with predefined unit cell shape and dimensions), the target function to minimise is the potential energy as a function of the coordinates; in the case of cell optimisation (*e.g.*, for crystalline structures of given geometry), instead, the potential energy is minimised as a function of the lattice parameters by monitoring stresses acting on the unit cell. In all cases, the task is finding the local minimum of the potential function as a function of the considered variables. Of the several algorithms that have been proposed to approach this ubiquitous problem, we will now review the most relevant for this thesis.

Conjugate Gradients

A commonly used technique in all minimisation problems is the conjugate gradients (CG) method. It is an iterative method that makes use of the gradient of the target variable, here the potential energy, by at each step displacing the atomic positions $\mathbf{R}$ along a direction conjugate to the gradient vector with respect to the previous move. The update step of the coordinates is thus given by

$$\mathbf{R}_{n+1} = \mathbf{R}_n - \alpha_n \left(\nabla U(\mathbf{R}_n)\right)_\perp \ , \tag{3.47}$$

where the $\perp$ symbol represents the conjugation operation followed by normalisation, and α_n is the step size at step n. The convergence of the iterative algorithm is typically achieved when the maximum force component acting on an atom is smaller than a threshold. This method is suitable for finding the global minimum of PES that do not present local minima, and is stable and applicable to large systems as it only requires the evaluation of first derivatives of the potential energy.

Fast Inertial Relaxation Engine

The Fast Inertial Relaxation Engine (FIRE) algorithm [47] is an extension of a damped molecular dynamics method with an adaptive timestep. The system is modelled as a fictitious particle of mass m in $3N$ dimensions following the equation of motion

$$\dot{\mathbf{V}}(t) = \frac{\mathbf{F}(t)}{m} - \gamma(t)\,|\mathbf{V}(t)|\,\left(\hat{\mathbf{V}}(t) - \hat{\mathbf{F}}(t)\right)\,, \tag{3.48}$$

where the second term is a friction term that makes the particle accelerate in the downhill direction with an inertial component in the direction of the current velocity. Velocities are zeroed whenever an uphill motion is detected, and the timestep of the MD integration is automatically adapted within a range to speed up the convergence. The FIRE method has been proven to be very effective in efficiently optimising atomistic structures, and often requires far fewer steps than CG optimisation while still not requiring computationally demanding Hessian evaluations.

Broyden–Fletcher–Goldfarb–Shanno

The Broyden–Fletcher–Goldfarb–Shanno algorithm [48], commonly referred as BFGS, is a rather popular iterative minimisation method. Let us take an objective function f at the current iteration point x_k – here, the potential energy of an atomic system for a given arrangement of atoms – and take its quadratic approximation

$$f(\mathbf{x}_k + \Delta\mathbf{x}) \simeq m_k(\Delta\mathbf{x}) = f(\mathbf{x}_k) + \mathbf{g}_k^T\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T\mathbf{H}_k\Delta\mathbf{x}\,, \tag{3.49}$$

where $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$ and $\mathbf{H}_k = \nabla^2 f(\mathbf{x}_k)$ are the gradient and Hessian of f at point $\mathbf{x}_k$. The minimum of the quadratic function above is found for a displacement

$$\mathbf{p}_k = -\mathbf{H}_k^{-1}\mathbf{g}_k\,, \tag{3.50}$$

therefore we can use $\mathbf{p}_k$ as a direction along which to move to find the next iteration point $\mathbf{x}_{k+1}$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k\,. \tag{3.51}$$

Calculating and inverting the Hessian matrix, though, is too expensive in most practical applications, so it is approximated by a different $\mathbf{B}_k$ that behaves as an approximation of the correct $\mathbf{H}_k$: the peculiarity of the BFGS method is in how the inverse (approximate)

Hessian $\mathbf{B}_k^{-1}$ is computed and updated at each step. Let us define for convenience

$$\begin{aligned} \mathbf{s}_k &= \mathbf{x}_{k+1} - \mathbf{x}_k = \alpha_k \mathbf{p}_k \\ \mathbf{y}_k &= \mathbf{g}_{k+1} - \mathbf{g}_k \,. \end{aligned} \tag{3.52}$$

There are two properties that $\mathbf{B}_k$ must satisfy to be a valid approximation of the true Hessian:

- the *secant condition*, imposing that the gradient of the quadratic approximation of f matches the actual gradient $\mathbf{g}$ at least at the iteration points $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$

$$\mathbf{B}_{k+1}^{-1} \mathbf{s}_k = \mathbf{y}_k \tag{3.53}$$

- $\mathbf{B}_k$ must be positive semidefinite.

Under these conditions, it is required to make the smallest update possible on the approximate Hessian:

$$\mathbf{B}_{k+1}^{-1} = \arg\min_{\mathbf{A}} |\mathbf{A} - \mathbf{B}_k^{-1}| \,, \tag{3.54}$$

where $|\cdot|$ indicates the Frobenius norm. By denoting $\rho_k = \mathbf{y}_k^T \mathbf{s}_k$, the solution of Eq. 3.54 is

$$\mathbf{B}_{k+1}^{-1} = \left(\mathbf{I} - \rho_k \mathbf{s}_k \mathbf{y}_k^T\right) \mathbf{B}_k^{-1} \left(\mathbf{I} - \rho_k \mathbf{s}_k \mathbf{y}_k^T\right) + \rho_k \mathbf{s}_k \mathbf{s}_k^T \,. \tag{3.55}$$

A common modification of the above algorithm is the so–called limited memory BFGS, or L–BFGS: instead of constructing the inverse Hessian from all of the previous iterations, L–BFGS builds the $\mathbf{B}_{k+1}^{-1}$ matrix only from the entries of $\mathbf{s}_k$ and $\mathbf{y}_k$ of the latest few iterations. The initialisation of the inverse Hessian matrix $\mathbf{B}_0^{-1}$, usually referred as preconditioning, is key to determine the convergence speed of the algorithm. Recent advances have pointed out an optimal preconditioner for condensed phase materials, which greatly reduces the number of convergence steps and makes it possible to efficiently optimise the geometry of large atomic systems [49].

3.6 Transition State Theory

One of the main applications of atomistic simulations is the study of chemical reactions. In the simplest case, a structure transitions between two known states A and B along a reaction path that can be described by a single reaction coordinate; in more complex situations, a

system can transition to unknown different structures, in which case the aim is to find saddle points between local minima of the (free) energy landscape expressed in terms of a collection of reaction coordinates commonly known as collective variables. The aim of these simulations is to identify relative stabilities between structures A and B (*i.e.*, energy differences between local minima) and to evaluate reaction rate constants for the transition between states according to Arrhenius equation

$$k_{A\to B} = \nu \exp\left(-\frac{\Delta G_{A\to B}}{k_B T}\right) . \tag{3.56}$$

Upon knowledge of the attempt frequency ν, one can evaluate the reaction rate by calculating the free energy barrier $\Delta G_{A\to B}$ separating the two states.

Thermodynamic Integration

When the entropic effects are assumed to be negligible with respect to the enthalpic contribution, the free energy barrier in the Arrhenius equation 3.56 can be approximated with a potential energy barrier and methods such as the Nudged Elastic Band (NEB) method [50] can be used. When the evaluation of a free energy barrier is desired because entropic effects may play an important role (*e.g.*, for calculations of biological molecules properties where the conformational space is very large and the interactions are relatively weak), one can choose among a wide set of possible techniques that fall under the broad categories of molecular dynamics (MD) or Monte Carlo simulations [51], or somewhere in between the two [52]. An account for some of the most well–known methods can be found in the vividly titled review *'Transition path sampling: Throwing ropes over rough mountain passes, in the dark'* by Bolhuis *et al.* [53]: in summary, the core difference among methods is how the phase space is sampled. The thermodynamic integration method (TI) [54] is used to evaluate the free energy change of a system when a parameter is slowly changed from the initial to the final state. Given a parameter $\lambda \in [0, 1]$, commonly named the collective variable or reaction coordinate, that can be varied continuously to transform the system from an initial state A (at λ = 0) to a final state B (at λ = 1), the partition function Z of the system will be a function of λ and so will be the free energy F:

$$F(\lambda) = -\frac{1}{\beta}\log Z(\lambda) = -\frac{1}{\beta}\log \int d\mathbf{q}\, e^{-\beta H(\mathbf{q},\lambda)} . \tag{3.57}$$

By evaluating the expression for $\partial_\lambda F(\lambda)$

$$\begin{aligned}\frac{\partial F(\lambda)}{\partial \lambda} &= -\frac{1}{\beta Z(\lambda)}\frac{\partial Z(\lambda)}{\partial \lambda} \\ &= -\frac{1}{\beta Z(\lambda)}\int \mathrm{d}\mathbf{q}\,(-\beta)e^{-\beta H(\mathbf{q},\lambda)}\,\frac{\partial H(\mathbf{q},\lambda)}{\partial \lambda} \\ &= \frac{1}{Z(\lambda)}\int \mathrm{d}\mathbf{q}\,e^{-\beta H(\mathbf{q},\lambda)}\,\frac{\partial H(\mathbf{q},\lambda)}{\partial \lambda} \\ &= \left\langle \frac{\partial H(\lambda)}{\partial \lambda} \right\rangle_\lambda ,\end{aligned} \tag{3.58}$$

we note that it corresponds to the canonical ensemble average at a fixed value of λ of the derivative of the Hamiltonian with respect to the parameter λ. The free energy difference between A and B can then be found by integrating this quantity:

$$\begin{aligned}\Delta F_{A\to B} &= \int_0^1 \mathrm{d}\lambda\,\frac{\partial F(\lambda)}{\partial \lambda} \\ &= \int_0^1 \mathrm{d}\lambda\,\left\langle \frac{\partial H(\lambda)}{\partial \lambda} \right\rangle_\lambda .\end{aligned} \tag{3.59}$$

At a practical level, the evaluation of this integral can be performed numerically by evaluating the integrand at a given set of intermediate points λ_i, each of which can be obtained through a MD simulation at constant temperature. Another possibility is given by the *slow growth* method [55] (SGTI), where a single thermostatted MD simulation is performed by gradually varying λ by a small increment $\Delta\lambda$ at each timestep so that the system remains approximately in equilibrium at all times. In the slow growth algorithm the integrand in Eq. 3.59 is replaced by the instantaneous value of the derivative, and the integral simply becomes the summation of all of the observed values:

$$\Delta F_{A\to B} = \sum_i \frac{\partial H(\lambda)}{\partial \lambda}\Big|_{\lambda=\lambda_i}\Delta\lambda\,. \tag{3.60}$$

3.7 QM/MM Multiscale Simulations

Many natural processes are the result of atomic level phenomena occurring at different time and length scales. In some cases the different scales can be simulated independently, so that accurate properties obtained through smaller scale simulations (*e.g.*, diffusion coefficients, reaction barriers for chemical reactions, densities) can be used to parametrise a cheaper model apt to treat the larger scale system. In the cases of interest for this thesis, instead, a

property of a system is the result of the intertwined interaction of phenomena that interact bidirectionally, and thus cannot be studied separately. In a naive approach, treating the complete system with the model appropriate for the most critical subsystem would be enough to guarantee the veracity of a simulation; however, it is ubiquitously true that more accurate tools are more computationally expensive, thus using a uniform precision approach is often unfeasible in real life.

Multiscale atomistic methods address this problem by providing a framework to blend together two different models of atomic interaction, focussing the use of the more computationally demanding only in regions where its accuracy is required. One class of multiscale simulations are the Quantum Mechanics / Molecular Mechanics (QM/MM) methods, which have been first proposed by Warshel and Levitt in 1976 [56]. The archetypal example here is the study of crack advance in a fracturing material: the localised events of bond breaking at the crack tip, which requires chemical accuracy often achievable only by means of electronic structure calculations (*e.g.* DFT), are determined by atomic bond deformations due to the concentration of a macroscopic elastic stress field; meanwhile, the larger scale atomic configuration, for which an appropriate potential for the material under study is sufficient, is dynamically rearranged as a result of the breaking bonds. The technical goal of a QM/MM calculation is to treat QM labelled atoms as if they were immersed in a completely QM system at all times. More ambitiously, when forces calculated by the MM model on the MM region atoms are a good approximation of the 'correct' QM ones, the ideal goal would be to reproduce the same physical observables that would be obtained by a fully QM simulation.

There are two key issues that have to be tackled to reach these goals: *(i)* how to properly define a QM region and *(ii)* how to blend QM and MM at their interface. The first issue, called QM region identification, can be recast into asking how much of the complete system is necessary to include in a DFT calculation and how to carve it so that the forces in the QM region are converged with respect to a full *ab initio* treatment. The second issue, the QM/MM coupling, deals with the stratagems put in place so that two mismatching interaction models are harmoniously linked into one coherent composite model that does not show any boundary artefact.

QM region identification

Let us assume that, according to our knowledge about the system and the processes under study, a set of neighbouring atoms is deemed to be treated quantum mechanically because, for instance, a chemical reaction taking place there cannot be treated properly with an

interatomic potential. We will define this set as the core QM region $\mathcal{C}$; here we will only consider $\mathcal{C}$ as one connected set of atoms, but our treatment can be naturally extended to more than one core QM region without additional obstacles.

The total energy of the system in quantum mechanics is a quantity that can be expressed as a sum of single–atom contributions only by means of somewhat arbitrary partitioning methods [57], whereas forces are inherently local properties of each atom. Energy–based QM/MM methods exist and have successfully been used for the study of a range of different systems [58–60], but for our applications we will concentrate on force–based methods that allow for greater flexibility. In most applications, we can prove that our QM calculations are near–sighted [61], *i.e.*, the force acting on an atom is not affected by displacements of far away atoms. Under this assumption, we can perform QM calculations on a subset of atoms carved out of the whole system to obtain QM–accurate forces on $\mathcal{C}$. The set of extra atoms to be added in the calculation is commonly called buffer region and will be denoted as $\mathcal{B}$.

How much of the surrounding environment is necessary to include to have correct QM forces on $\mathcal{C}$ heavily depends on the system under study, and is usually set by a parameter that can be a buffer radius $r_{\mathcal{B}}$ or a number of first neighbours bond hops $n_{\mathcal{B}}$: $\mathcal{B}$ will be the union of all atoms not in $\mathcal{C}$ that are within $r_{\mathcal{B}}$ Ångstroms or $n_{\mathcal{B}}$ bond hops from any atom in $\mathcal{C}$. Since the computational cost of an *ab initio* calculation is several orders of magnitude higher than the one of a classical potential, the overall cluster $\mathcal{C} \cup \mathcal{B}$ composing the QM region should be the smallest possible that still guarantees adequate accuracy. The carving of a cluster out of the bulk system results in a number of non–physical unsaturated chemical bonds, which give rise to surface electronic states that hinder the convergence of forces in the QM core region as a function of buffer size.

Several stratagems have been proposed to reduce the size of the buffer region $|\mathcal{B}|$: in non–metallic systems it is common practice to terminate the surfaces with passivating atoms by placing H atoms or specifically tailored pseudo–atoms along the broken bonds. In conjunction with terminating atoms, when the QM and MM region are expected to present a strong electrostatic coupling it is also possible to perform calculations on the QM cluster embedded in a system of external charges that is supposed to mimic the electric field produced by the excluded system [60, 62, 63].

In some circumstances, as in a moving crack tip position [64], an adaptive QM region is necessary so that it can be updated during a simulation. To reduce instabilities over the identification of the core QM atoms, it is possible to define the QM region in a hysteretic fashion, so that the criterion of inclusion of an atom in $\mathcal{C}$ is less restrictive than the exclusion criterion [65]. It must be pointed out that an evolving QM region precludes the possibility

of strictly conserving the total energy, but this is not necessarily a negative feature when dealing with constant–temperature calculations, as the presence of a thermostat unavoidably precludes energy conservation.

QM/MM coupling

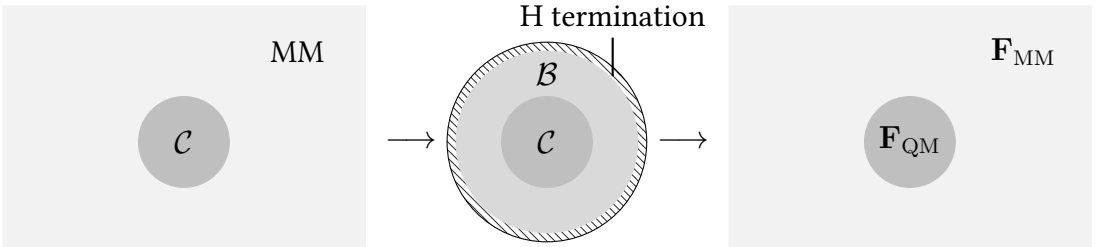
Once a cluster carving and buffering recipe allows us to calculate QM forces, this must be coupled with a method to use such information in a way that the mismatch at the interface between regions treated with different theories is minimised. A coupling method should first restore the lost action–reaction principle (resulting in conservation of total momentum): if two atoms are treated with different interaction models, it is no longer true that the forces induced by one on the other are equal and opposite. Conservation of total energy, although not strictly required in canonical ensemble simulations, is nevertheless a good indicator of the stability of the dynamical system; smaller energy drifts require gentler thermostats with a smaller footprint on the system trajectory.

Aside from conservation laws, it is also important that bulk quantities are matched between QM and MM regions. Matching lattice constants – or equilibrium bond lengths when dealing with non–crystalline materials – is necessary to avoid an artificially introduced hydrostatic pressure on the QM cluster: to do that, one can very simply rescale the QM cluster by a factor equal to the ratio between lattice constants (or equilibrium lengths) calculated by the QM and MM models. Matching elastic constants and Einstein oscillator frequencies (related to the energy change to a small displacement of a single atom) are also desirable to guarantee continuity.

Among the many coupling methods available, as thoroughly reviewed in Ref. [65], force mixing methods are simple and effective solutions. The simplest version is the *abrupt* force mixing method, where QM forces are used in $\mathcal{C}$ (and discarded in $\mathcal{B}$), and MM forces are used elsewhere as depicted in Fig. 3.1. Clearly, this does not guarantee any conservation of energy or momentum, while the matching of bulk properties is left to the QM cluster rescaling and the appropriateness of the interatomic potential. To guarantee momentum conservation, a very simple modification of the method is to correct each QM force with a small corrective force that guarantees that the total momentum is constant at all times during the simulation. More refined implementations are the hop–ramp and distance–ramp force mixing methods, where atoms in a shell within a prescribed bond hop or Euclidean distance are subject to a linear combination of the QM and MM forces

$$\mathbf{F}_{\mathrm{mix}} = \lambda \mathbf{F}_{\mathrm{QM}} + (1-\lambda)\mathbf{F}_{\mathrm{MM}}\,, \tag{3.61}$$

where the parameter λ is set to decrease from 1 to 0 when moving away from $\mathcal{C}$.



(a) Full system (b) QM region identification (c) QM/MM coupling

Fig. 3.1 Abrupt force mixing QM/MM scheme

Learn On The Fly

The 'Learn On The Fly' (LOTF) method [66, 67] is an extension of the abrupt force mixing QM/MM scheme. The computed forces, rather than being used directly in their pertinent regions to propagate the dynamics, are instead used to fit a global IP V_{LOTF} composed of the original MM potential V_{MM} and a simple adjustable potential V_{adj}

$$V_{\mathrm{LOTF}}(\boldsymbol{\theta}^*) = V_{\mathrm{MM}} + V_{\mathrm{adj}}(\boldsymbol{\theta}^*)\,, \tag{3.62}$$

where the optimal parameters $\boldsymbol{\theta}^*$ are obtained by least squares fit

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\mathrm{argmin}} \sum_{i \in \mathcal{C}} \left| \mathbf{F}_{i,\,\mathrm{QM}} + \frac{\partial V_{\mathrm{LOTF}}(\boldsymbol{\theta})}{\partial \mathbf{r}_i} \right|^2 . \tag{3.63}$$

The form of the adjustable potential is arbitrary; among the different definitions that have been proposed and benchmarked over time, a linear pairwise potential of the form

$$V_{\mathrm{adj}} = \sum_{ij} \alpha_{ij} r_{ij} \tag{3.64}$$

allows a fast linear algebra solution of the optimisation problem in Eq. 3.63 without any loss of accuracy with respect to more complex definitions [65, 68].

The LOTF method guarantees momentum conservation since the system is at all times subject to forces derived from a single model; as the model itself changes at every fitting procedure, instead, energy conservation is not recovered. A major advantage of the

algorithm is that the fitting parameters $\boldsymbol{\theta}^*$ follow a smooth time evolution and can be interpolated between QM calculations taking place every so often (for example, every 10 fs for Si fracturing at room temperature) instead of being calculated at every timestep, offering a speedup factor equal to the number of interpolation steps with respect to alternative QM/MM schemes. The full LOTF algorithm is implemented as a predictor–corrector method following these steps:

QM selection: Select $\mathcal{C}$ and optimise the LOTF parameters according to Eq. 3.63, obtain $\boldsymbol{\theta}^{\mathrm{pred}}$

Predictor: Propagate with $V_{\mathrm{LOTF}}(\boldsymbol{\theta}^{\mathrm{pred}})$ the dynamics for a given number of timesteps

QM Calculation: Perform QM calculations on the cluster as in Fig. 3.1

Fitting: Refit the LOTF parameters according to Eq. 3.63, obtain $\boldsymbol{\theta}^{\mathrm{corr}}$

Corrector: Revert the system back to prior the predictor step and rerun the dynamics linearly interpolating the potential parameters between $\boldsymbol{\theta}^{\mathrm{pred}}$ and $\boldsymbol{\theta}^{\mathrm{corr}}$.

Chapter 4

Ensemble QM/MM Simulations

QM/MM calculations reduce to the bare minimum the amount of QM–accurate information to be computed, but this bare minimum can still be startlingly expensive to compute in some physical systems and types of simulation. Let us take the case of fracture simulations of oxide materials, an example relevant for this thesis: one typical MD run may span a timescale of 10 ps and, because of the long–range electrostatic interactions, QM clusters tend to have very large buffers [69], with the result that each QM calculation has to be performed on several hundreds of atoms. To better grasp the issue, the same kind of QM/MM simulations on covalent crystals (*e.g.*, silicon) require QM cluster sizes of approximately one order of magnitude smaller: if we were to perform the same calculation on the same computer with a commonly available planewave DFT code (thus assuming a cubic scaling with respect to the number of valence electrons), the time to solution would be roughly one thousand times longer – 10 timesteps a day instead of a timestep every 10 seconds. Using linear scaling DFT codes (*e.g.*, ONETEP [70] or SIESTA [71]) is not a better choice for these systems either, as they are mostly tailored to be used on very large systems (up to millions of atoms) for one–off energy evaluations or geometry optimisations. The linear scaling behaviour is counterbalanced by a very large prefactor, so that the performance of these codes becomes advantageous at about, if not beyond, our typical QM size of less than 1000 atoms. For our intermediate problem size, an intermediate–scaling code such as the hybrid mixed planewave/Gaussian basis set CP2K is the best choice.

A naive thought would be to simply use the largest computer available to run the best available DFT code for that given computing architecture, physical system and simulation type. As we will discuss in this chapter, the proportions of the challenge and the subtleties arising with the use of vast computing resources require a more thoughtful approach, which

led to the development of a novel parallel computing architecture where a purpose–designed graph partitioning method optimally balances the load across simultaneous calculations.

4.1 Computational Framework

Computing architecture

The largest computer available during most of my studies is the IBM BlueGene/Q *Mira* supercomputer at the Argonne National Laboratories, thanks to the computing resources awarded to the INCITE project related to the topics of this dissertation. It is useful at this point to give a general overview of the machine, pinpointing some aspects that acted as both constraints and driving force for the development of novel algorithms and code scaling strategies. In a nutshell, the BlueGene/Q (BG/Q) architecture is the following:

- The basic component of a BG/Q machine is the BlueGene/Q compute chip, also referred as compute node. A compute node (CN) is a 16–cores 64 bit PowerPC A2 processor, and each core can run up to 4-way simultaneously multithreaded instructions at a clock speed of 1.6 GHz. Each compute node can access a DRAM memory of 16 GB.
- The networking follows a 5D torus connectivity: a midplane is a rack of 512 electrically connected compute nodes, and more midplanes are bundled together to form a single machine by means of optical interconnections.
- The compute nodes communicate by means of I/O nodes in a ratio 128:1;
- Both compute and I/O nodes run a very basic IBM CNK (compute node kernel) based on the Linux kernel, and do not allow user access. Notably, the kernel does not allow `fork` and `exec` commands to be executed: new processes cannot thus be spawned by an already running process.
- The user can access a BG/Q machine resources through the login nodes, also known as the front end nodes (FEN). These nodes run a standard Linux kernel, and are used to submit the job scripts.

Mira comprises 96 midplanes, or 49152 compute nodes, or 786432 compute cores. A job can be submitted to a partition of at least 512 nodes (1 midplane), and the available partition sizes are typically 2^N. Ensemble jobs are allowed: a single job can run multiple concurrent

tasks on smaller and independent subpartitions (also called 'blocks') within the allocated partition; 'subblock' jobs are allowed as well, meaning that an executable can run on a subpartition of size $2^N < 512$.

Ensemble QM/MM

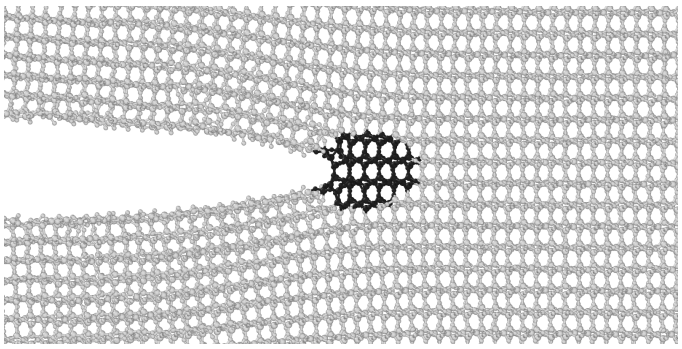


Fig. 4.1 Crack tip in α–quartz. Darker atoms represent the core QM region

Let us consider the problem of fracture in silica: here, the number of atoms around a crack front that require an accurate force calculation is typically well above 100, and the buffer size to guarantee the force evaluation to be converged is at least another 500 atoms. For a problem size of 600 to 1000 atoms, the most modern and performant DFT codes may still work in optimal scaling conditions on a minimum size partition (about 8000 cores); given that the good scaling properties of such codes is obtained by methods of the likes of domain decomposition (*e.g.*, sparsification of the Hamiltonian matrix thanks to localised electronic basis sets [72]), we cannot expect them to scale beyond a point where the number of computing cores is of the same order of magnitude of the number of basis functions. It is clear that we need a paradigm shift if we wish to utilise larger computer resources to cut the simulation time of a QM/MM simulation.

The fundamental idea presented here can be thought of as a reformulation of domain decomposition that can be used whenever the only quantities needed in a QM/MM calculation are the forces, such as in force mixing schemes like the LOTF method. Let us define the *time to solution* (TTS) of a DFT calculation as the wall clock time elapsed to reach a full electronic structure minimisation and compute the Hellman–Feynman forces on all atoms in a core QM region, and the *computational cost* of a DFT calculation as the total resources used, typically measured in CPU–hours, calculated as

$$N_{\mathrm{CPU}} \cdot TTS\,.$$

Given a core QM region $\mathcal{C}$, we wish to reduce the time to solution and, if possible, the computational cost of QM force evaluations by splitting it into k parts $\{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ that are

- non–overlapping: $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset \ \forall i \neq j$,
- complete: $\bigcup_{i=1}^{k} \mathcal{C}_i = \mathcal{C}$,
- connected: $\exists$ a bond–hopping path connecting all atoms within each part $\mathcal{C}_i$

Let the computational cost of a DFT calculation scale with the number of atoms as $\mathcal{O}(N^p)$, where p is between 1 (ideal linear scaling code) and 3 (standard planewave code). If we were to perform one single QM force evaluation on the whole QM cluster including buffer $\mathcal{C} \cap \mathcal{B}$, the TTS would be

$$TTS(\text{whole}) = \alpha \left(|\mathcal{C}| + |\mathcal{B}|\right)^p \ , \tag{4.1}$$

for some constant α, where we denoted the number of atoms with the cardinality symbol $|\cdot|$. If we instead ideally split the core QM region in k parts of equal size $|\mathcal{C}|/k$, the TTS is reduced to

$$TTS(\text{split}) = \alpha \left(|\mathcal{C}|/k + \max(|\mathcal{B}_i|)\right)^p \ . \tag{4.2}$$

Without stating specific assumptions on the relationship between the sizes of a core and its corresponding buffer region, it is clear that a larger core requires a larger buffer size, so the TTS will always be smaller in a QM splitting strategy – albeit we must point out that the overall computational cost may increase. Given the architectural constraint that no new processes can be spawned by the currently running ones once a calculation is started, we are bound to a rigid configuration where k QM calculators and one MM calculator will be running at the same time on separate subpartitions of the same job. Since DFT force evaluation is always the bottleneck of QM/MM simulations, we wish to assign the largest portion of computational power to the QM calculators. We decide for simplicity to divide the total partition size N_{tot} into N_{blocks} subpartitions of the same size $N_{\text{CN}} = N_{\text{tot}}/N_{\text{sub}}$: the classical dynamics operations can then be carried out on one subpartition, while all the remaining $k = 2^N - 1$ blocks can be assigned to the execution of independent DFT programs. With this strategy, the number of clusters k and the computing power per cluster N_{CN} are free parameters to be tuned so to always using the adopted QM calculator at its optimal computational power N_{CN} and typical problem size $|\mathcal{C}_i \cap \mathcal{B}_i|$, where 'optimal' is meant as the desired balance between computational cost and TTS.

As a further advantage of using the described distributed method in a BlueGene architecture, the job partition is fully tiled and no computing power is left idle. In fact, in a QM/MM

job where the QM calculator(s) is used as a plugin black box rather than being merged into one monolithic QM/MM executable, the MM operations require to be allocated on one of the 2^N subpartitions, leaving the remaining processing power of $2^N - 1$ subpartitions for the QM operations: these can only be completely used when more than one QM task is run simultaneously. This is exemplified if Fig. 4.2.

Additionally, if the QM region moves and varies its size during a simulation as in the case of crack propagation, techniques to allow an automatic identification and update of the QM region are necessary: in this context, MM codes able to interface to QM black boxes are the most versatile and used solution, but recent developments allow adaptive QM/MM simulations in a single executable [73]. The implementation of the ensemble QM/MM

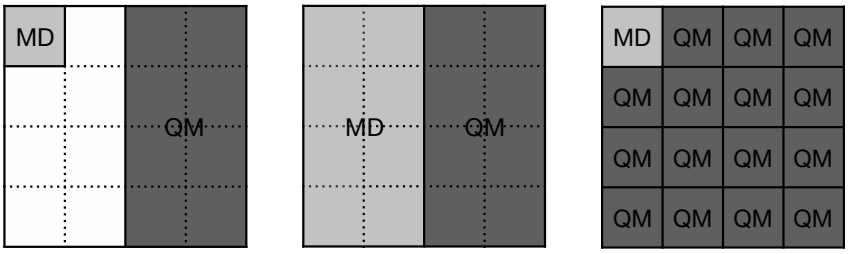


Fig. 4.2 Tiling options of a BlueGene/Q 16-block job partition for running the executables of a QM/MM calculation. One small square corresponds to one block, and an executable can only run on 2^N blocks. (a) The allocation is only partially occupied; (b) The allocation is fully occupied, but the MD partition is unnecessarily large; (c) Several simultaneous QM calculations allow for an optimal tiling of the available partition

architecture is based on the QUIP multi–purpose MD package, that is connected with the separate DFT runs by means of a socket–based communication protocol mediated via a lightweight Python script running on the front–end node (FEN) [74]. In brief, the structure is the following:

- The client MD code reads in the QM forces from the server, performs the LOTF dynamics between each predictor/corrector stage, and outputs to the server the atomic positions after each LOTF iteration;
- The client QM codes read in from the server the positions of the carved clusters and output the QM forces to the server;
- The server process running on the FEN handles the communication of the MD (positions) and QM (forces) executables, and performs the operations of QM region identification, partitioning, and cluster carving.

Fig. 4.3 shows a visual representation of the scheme.

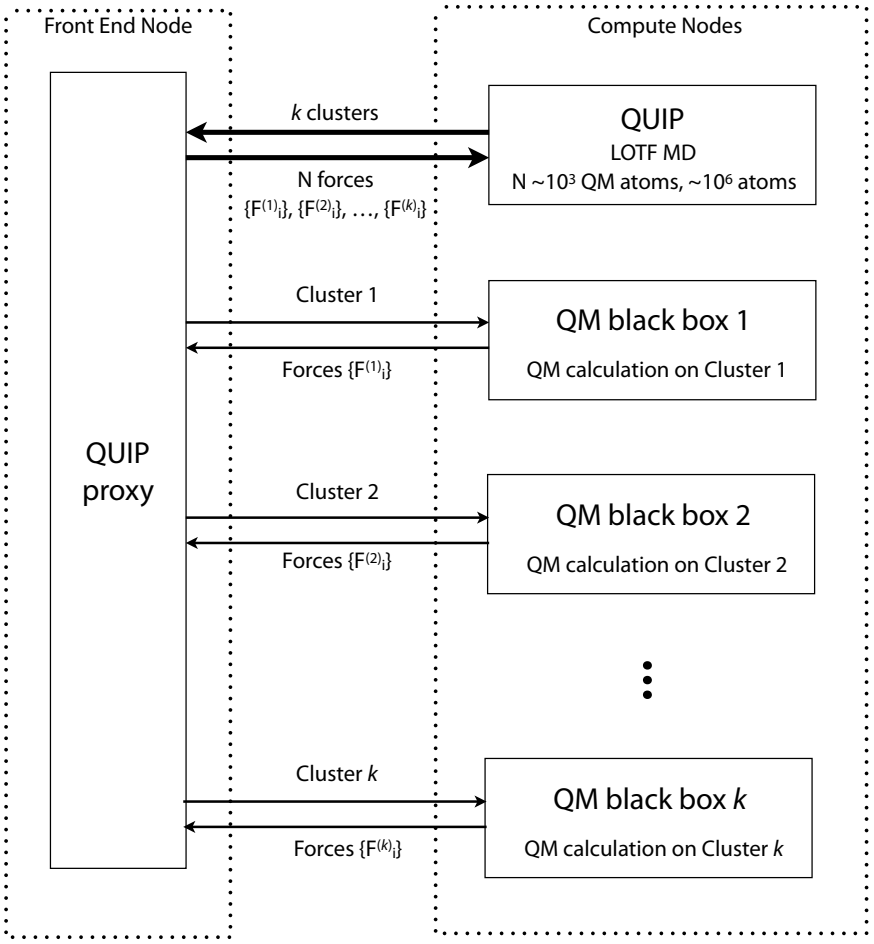


Fig. 4.3 Structure of the QM/MM ensemble parallel calculation architecture: the server process, on the left, is executed as part of the job submission script and remains alive to serve as communication beacon between the independent executables running on different partitions of the assigned compute nodes, along with performing the operations related to the creation and update of the QM clusters. Scheme adapted from Kermode Ref. [74]

'Digestive Ripening' Partitioning Algorithm

The performance of the ensemble parallel QM/MM scheme is largely determined by how well the computational load is balanced across the simultaneous DFT processes. Since QM forces need to be evaluated synchronously at every QM step of the LOTF algorithm (or at every timestep of a standard QM/MM scheme), the slowest calculation sets the pace for all others: therefore, ensuring that all DFT calculations reach convergence within the narrowest wallclock time window is of key importance. The current section will provide a thorough description of the partitioning method designed to reach this goal.

As a rough assumption, we can estimate that the time to solution for a DFT calculation in a system of uniform chemical composition is an increasing function of its number of atoms. The time to solution of our ensemble QM calculation is then determined by the size of the largest QM cluster, measured as the sum of the number of atoms in its core and buffer regions: the recipe for an optimal balance would then be to split our QM region so that all of the resulting clusters are of the same size. But while we have full control on the sizes and shapes of the core QM regions $\{\mathcal{C}_i\}$, the sizes of their corresponding buffer regions $\{\mathcal{B}_i\}$ are not easily predictable. The standard approach to growing a buffer around a core region $\mathcal{C}_i$ is to set its thickness in either Ångstroms or number of bond hops to tag the atoms that *need* to be in $\mathcal{B}_i$; after that, a set of empirical rules (specific for the material) will include further atoms from the system into $\mathcal{B}_i$ and complete the unsaturated bonds with artificial terminating hydrogens, all to make the resulting cluster a physically sound one – in silica, the main required feature is to preserve full SiO_4 tetrahedra.

We postulate that a QM region partitioning strategy should generate a set of core QM regions $\{\mathcal{C}_i\}$ that are:

- of *equal size*, so that that no cluster is bigger than any other already at the level we are in control of; also, intuitively we can expect a larger core to have a correspondingly larger 'skin' around it for a given skin thickness;
- as *convex* as possible to minimise the size of each buffer, which increases commensurately with the aspect ratio of the core for a given core size: the optimal cluster shape is the sphere (or the circle in 2D geometries).

To implement these ideas into an algorithm, let us first translate an atomic environment into an object for which the partitioning problem has long been studied: a graph. For simplicity, we will now consider the QM region as a single connected set of atoms – the extension of the algorithm to a set of disconnected QM regions is straightforward if each

connected component is treated independently. The mapping from a system of atoms $\mathcal{C}$ to its undirected graph $G(V, E)$ is rather natural when considering atoms as vertices V and chemical bonds between them as edges E: the splitting of $\mathcal{C}$ is now transformed into the problem of partitioning the graph into k subgraphs $\{S_1, S_2, \ldots, S_k\}$ of the same size and as convex as possible for a given definition of convexity. Let us define the convexity ξ of a graph $G(V, E)$ with a given number of vertices $|V|$ to be the sum of shortest path distances between all pair of vertices, *i.e.*:

$$\xi(G) = \sum_{i=1}^{|V|} \sum_{j>i}^{|V|} d(V_i, V_j)\,. \tag{4.3}$$

Such a definition mimics the concept of convexity of the original atomic system, so that a maximally convex graph corresponds to a round atomic cluster.

The list of different k–way graph partitioning algorithms available to date is endless [75], and each one of them has been developed to optimise one or a few performance indicators: scalability, speed, size balance, number of edge cuts. Our specific use case – small graphs of 10^2–10^3 vertices, very strict subgraph size distribution, maximal convexity – demand a specific solution, which I propose as a two–step algorithm.

The first step consists of making a first guess for the subgraphs: this can be performed by means of any available k–way graph partitioning algorithm. The current implementation allows for k–means partitioning [19], spectral clustering [22] and the multi–step METIS algorithm [76]: all of these are known to be among the methods that produce subgraphs of good size balance. The output of this step is a set of k subgraphs $\{S_1, S_2, \ldots, S_k\}$ and a set of cut edges E_{cut} whose union is the original graph:

$$\left(\bigcup_{i=1}^{k} S_i \right) \cup E_{\text{cut}} = G\,. \tag{4.4}$$

The second step consists of a refinement procedure by means of an algorithm I named *digestive ripening*, so called after a thermodynamic process, also known as inverse Ostwald ripening, in which the free energy of a set of nanoparticles is minimised by the exchange of atoms until a monodisperse set of round elements is obtained [77].
If we are to consider the just obtained subgraphs as nanoparticles evolving towards a monodisperse configuration, we must define a free energy–like quantity for a graph. Let us first calculate the shortest path edge–hopping distance matrix of the whole graph $D(G) \in \mathbb{N}^{|V|} \times \mathbb{N}^{|V|}$ by means of a breadth–first search algorithm; the distance matrix of

each subgraph S_i is obtained by slicing the matrix $D(G)$ by only including the columns and rows of the vertices of S_i. If we now take the sum of squares of all elements of the restricted distance matrix $D(S_i)$ as a cost function of the subgraph S_i

$$F(S_i) = \sum_{a,b} \left(D_{a,b}\left(S_i\right)\right)^2 \,, \tag{4.5}$$

we can observe that:

- It will increase commensurately with the aspect ratio of the corresponding atomic cluster for a given number of vertices, as the atoms are, on average, farther apart for elongated systems than for round ones
- It will increase together with the number of vertices for a given aspect ratio, as we sum over more elements.

This arbitrarily defined function F is then a sound definition of a quantity associable to a 'free energy' of a graph, and the system of subgraphs $\{S_i\}$ will find its groundstate – *i.e.*, the closest–to–monodispersed atomic cluster configuration – at the minimum of the global cost function

$$F(G) = \sum_{i=1}^{k} F(S_i)\,. \tag{4.6}$$

To obtain the set of optimal subgraphs $\{S_1^*, S_2^*, \ldots, S_k^*\}$, an iterative algorithm swaps nodes at the boundaries between different subgraphs at each iteration with the aim of lowering the value of the global cost function. To guarantee that the corresponding atomic clusters are formed of contiguous atoms and thus physically sound, we must require all subgraphs $\{S_1^*, S_2^*, \ldots, S_k^*\}$ to be connected: in practice, this is ensured by the stricter requirement that at all steps all subgraphs are connected.
Given an initial guess $\{S_1(V_1, E_1), \ldots, S_k(V_k, E_k)\}$ for the k–way partitioning of the connected graph $G(V, E)$, two such algorithms have been built and tested:

Greedy

1. Calculate the cost function of all subgraphs $\{F(S_i)\}$, then sort them in decreasing order of absolute difference from the average value. Store indices of sorting order I and pick the first entry $i^* \in I$
2. Find all of the edges $E_{i^*}^{\text{neigh}} \subseteq E$ connecting S_{i^*} with its neighbouring subgraphs

3. For every edge in $E_{i^*}^{\text{neigh}}$, try to assign both end vertices a, b to the subgraph with the lowest cost function and calculate the variation in cost function

$$\Delta = F(S_{i^*}^{\text{new}}) + F(S_j^{\text{new}}) - F(S_{i^*}^{\text{old}}) - F(S_j^{\text{old}})\,; \tag{4.7}$$

   store the move if $\Delta < 0$ and if the 'donor' subgraph remains connected

4. If it exists, apply the highest gain move (lowest Δ) and go to (1), otherwise choose next $i^* \in I$ and go to (2)

5. Exit if no further favourable move is found.

Random

1. Calculate the cost function of all subgraphs $\{F(S_i)\}$ and create E^*, a copy of the list of all *directed* edges of the whole graph $G(V, E)$

$$(i \to j) \in E^*, (j \to i) \in E^* \ \forall (i, j) \in E \tag{4.8}$$

2. Pick randomly one edge $v = (a \to b) \in E^*$
   - If both nodes a and b belong to different subgraphs, calculate the gain in cost function Δ for the operation of assigning a to the same subgraph of b: if $\Delta < 0$ accept move and go to (1)
   - if $\Delta \geq 0$ or if both nodes a and b belong to the same subgraph, remove v from E^* and go to (2)

3. If E^* becomes empty, we ran out of subgraph swaps possibilities and the algorithm has converged. Quit.

An obvious improvement of the random algorithm would be to allow escaping from local minima of the cost function: this has been tested by modifying step (2) with an acceptance criterion borrowed from the Metropolis–Hastings algorithm [78]. As long as the initial partitioning guess is obtained from a reliable partitioning method as the ones presented, this additional complexity does not appear to yield any benefits.

QM Region Update

Once the QM core region is partitioned into k parts $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$, the corresponding clusters are carved out of the original system by including buffer atoms and terminating hydrogens so that the Hellman–Feynman forces of the core atoms are 'correct', *i.e.*, converged with respect to the buffer size. Since the buffering routine is exactly the same in the case of a single QM region or k parts of it, we can conclude that the buffer parameters that guarantee force accuracy in the former case will be appropriate for the latter as well.

The set of k atomic clusters $\mathcal{C}_i \cup \mathcal{B}_i$ are then simultaneously sent to the compute nodes QM executables for calculation. As the MD simulation advances, the QM core region $\mathcal{C}$ is automatically tracked and selected according to a set of heuristic rules to follow the region(s) of the systems where higher accuracy is desired, such as the moving crack tip of a propagating crack. In these simulations, $\mathcal{C}$ does not change abruptly from one timestep to the next, and only every once in a while the difference between the QM regions of two subsequent timesteps ('old' and 'new') is large enough to justify a new partitioning and buffering operation. This should happen as infrequently as possible for two reasons: *(i)* if the atoms in a cluster have not changed from one timestep to the next, the QM executable does not need to be restarted and the 'old' wavefunctions and electron densities loaded in memory can be reused as a starting guess to speed up the DFT calculation; *(ii)* the operation itself can be time consuming and the CNs should be left idle as little as possible.

To decide whether or not to update the partitioning of $\mathcal{C}$, the first check is on the system as a whole: if the set of atoms in $\mathcal{C}$ and the list of atomic bonds in $\mathcal{C} \cup \mathcal{B}$ has remained unchanged, a new partitioning and buffering is avoided because its result would be the same as the already present clusters, so the old $\mathcal{C}_i$ may be reused. if that is true, the atomic positions of the old QM clusters are simply updated in place with the new values; at this point, each new cluster $\mathcal{C}_i \cup \mathcal{B}_i$ is compared with its old instance for compatibility:

1. the set of atoms in $\mathcal{C}_i \cup \mathcal{B}_i$ has not changed,
2. if the lattice vectors have not changed,
3. the RMS displacement of the atomic positions between the new and the old cluster is less than a given threshold (its value depends on the DFT executable in use).

If a given cluster meets all three of the above requirements, the setup already present in memory can be used – and is an appropriate guess – for a continuation of the previous QM calculation; otherwise, the DFT executable is restarted.

4.2 Results

QM Region Partitioning Benchmark

To first grasp the effectiveness of the digestive ripening (DR) partitioning method described above, we will compare it against some standard partitioning methods on a system that allows for very simple and informative visualisations. The artificially–generated atomic system is a two layers thick disc of Si atoms at equilibrium positions in the crystalline diamond structure, with a diameter of about 100 Å and containing 2097 atoms; the number of clusters is set to $K = 7$. The tested partitioning methods are the following:

- k–means partitioning, Euclidean distance metrics based on atomic positions (`scikit-learn` implementation [79])
- spectral clustering, Euclidean distance metrics based on atomic positions (`scikit-learn` implementation [79])
- `METIS` partitioning, distance metrics based on bond hop shortest path lengths [76]
- digestive ripening partitioning, greedy algorithm, k–means initial guess, distance metrics based on bond hop shortest path lengths
- digestive ripening partitioning, random algorithm, k–means initial guess, distance metrics based on bond hop shortest path lengths

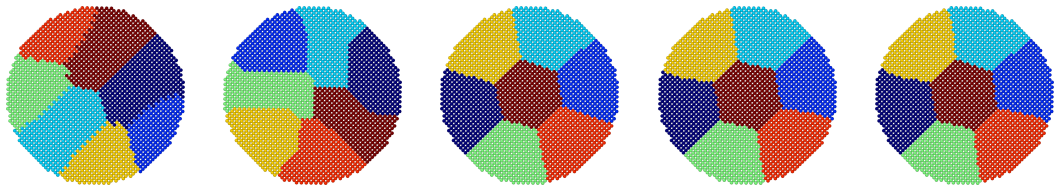
The results shown in Table 4.1 and Fig. 4.4 reveal that, among the standard partitioning algorithms, the spectral clustering method is completely inadequate for the task, the `METIS` method does guarantee a narrow cluster size distribution but not their roundness, whereas the k–means method is the best choice as it guarantees a good partitioning quality while at the same time being extremely fast. The DR partitioning produces maximally convex clusters comparable to the ones obtained with k–means but with a narrower size distribution than any other tested method.

Contrary to the other algorithms tested, the DR approach has only been implemented in a non–compiled and not heavily optimised Python library, so it is comparatively 2 orders of magnitude slower: this can limit the adoption of the algorithm at the current stage, but in our actual applications it is more than compensated by the reduction in size of the QM clusters. Interestingly, the random algorithm is at the same time faster to converge and produces slightly better clusters than the greedy algorithm, therefore it will be the choice adopted for the rest of the chapter.

Let us now briefly analyse the scaling of the DR–random algorithm as a function of the graph size N_{atoms} and of the number of clusters K (the graph size is selected by keeping the disc shape of the system constant and varying its diameter). From the results presented in Fig. 4.5, we can see that the partitioning time does not show a clear correlation with the number of partitions K for a given graph size N_{atoms}: it increases with K for small N_{atoms}, but the correlation is progressively lost as N_{atoms} grows. On the other hand, the time rapidly increases as N_{atoms} increases for a given number of partitions K, clearly pointing out that the algorithm can only be effectively applied to small graphs of at most $\sim 10^3$ nodes: fortunately, this is larger than the typical size of a core QM region, so the method can be deployed in the ensemble QM/MM scheme. In summary, the digestive ripening (random) algorithm can be effectively used to partition small graphs with extremely high quality in terms of size variance and roundness.

Table 4.1 Comparison between partitioning algorithms

Algorithm	Partitioning time[1] [s]	Cluster size st. dev.	Round clusters
spectral clustering	4.9	68.0	no
`METIS`	0.18	3.2	no
k–means	0.27	7.2	yes
DR greedy	89	3.0	yes
DR random[2]	25	2.6	yes



(a) (b) (c) (d) (e)

Fig. 4.4 Visual comparison between partitioning algorithms: (a) spectral clustering, (b) `METIS`, (c) k–means, (d) DR greedy, (e) DR random

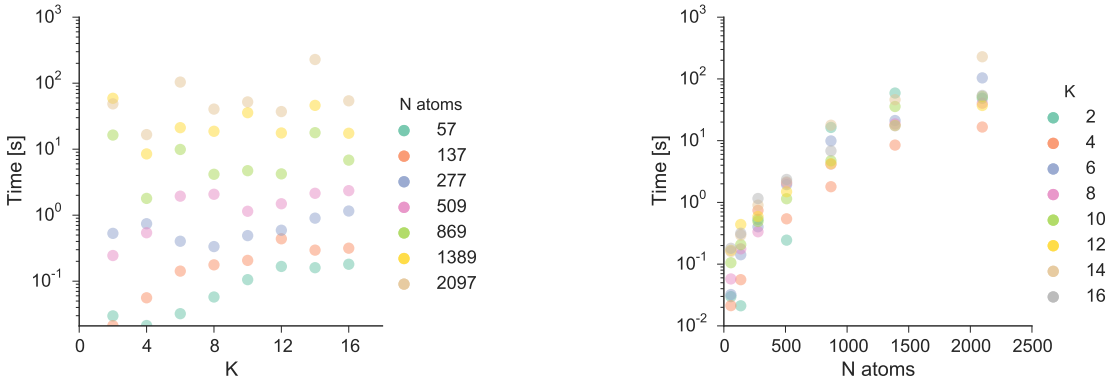


Fig. 4.5 Scaling of the random algorithm implementation of the digestive ripening partitioning[1,2]. Left: partitioning time vs. number of partitions K for a given graph size N_{atoms}. Right: partitioning time vs. graph size N_{atoms} for a given number of partitions K

[1]Results obtained using a single core of a 2.4 GHz Intel Core i5 CPU (I5-4258U)
[2]Since the method is random, the results have been averaged over 3 runs

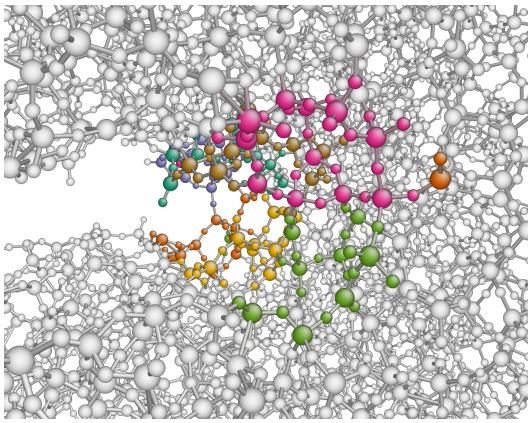


Fig. 4.6 Partitioned core QM region of a notched SiO_2 specimen. The colours identify the 7 different clusters obtained by a DR partitioning. The system is periodic along the crack front direction (direction orthogonal to the page). Only one unit cell is shown here

Let us now move on to the task of profiling the effectiveness of the DR method when compared with a standard k–means in a much more complex real case scenario: the partitioning of a core QM region of a notched crack tip in an amorphous SiO_2 specimen. The system, shown in Fig. 4.6, contains 218 core QM atoms and is periodic along the crack front direction (z): periodic boundary conditions (PBC) are applied along z with a periodicity of 30 Å, and to increase the system size we replicate the system along z; the ratio N_{atoms}/K is kept approximately constant for a direct comparison of cluster sizes across different system sizes. We benchmarked the method for N_{atoms} = 218, 436, 872 and a number of clusters K = 7, 15, 31 respectively. The task is particularly hard because of a set of concurrent factors. Firstly, the amorphous silica structure is very open and the density of bonds is very low: in a digestive ripening setting, this means that the number of allowed ripening moves at each step is rather small and the algorithm may fall into a local minimum more easily given an unfortunate initial guess. Secondly, the geometry is inhomogeneous: the presence of a crack tip means that the buffering procedure may or may not find vacuum in some growth directions, thus leading to some unavoidable cluster size imbalance.

In Fig. 4.7 we measure the time elapsed in the two steps of partitioning and buffering that would be performed on a front end node. The k–means algorithm is again very fast and its time to solution is negligible. The DR partitioning, instead, converges in a time that is smaller but comparable to the buffering operations. The DR partitioning method has been devised to optimise size and shape of the core QM regions, and in this context there is very little control over the corresponding sizes of the buffered clusters. Since our target is to equalise the time to solution for each QM force calculation, perhaps the most relevant simple metrics to quantify the computational cost of a cluster is its number of valence

electrons. To understand the usefulness of pursuing a QM load balance by optimising the partitioning method of just the core regions, we must prove that the two quantities are correlated. The data presented in Fig. 4.8 shows that, to a certain degree, this is true and so the approach is valid. We can also observe that the ripening procedure does compress the range of core QM sizes: this is often, but not always, accompanied by a reduction in the number of outliers in the vertical direction.

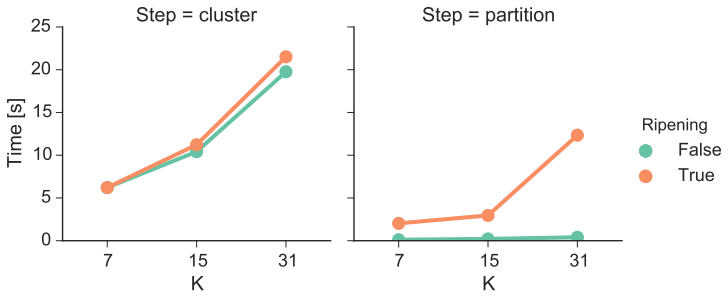


Fig. 4.7 Timing of partitioning and clustering operations. We compare the time necessary to partition the full core QM region $\mathcal{C}$ and the time to generate the corresponding set of QM clusters $\{\mathcal{C}_1 \cup \mathcal{B}_1, \mathcal{C}_2 \cup \mathcal{B}_2, \ldots, \mathcal{C}_K \cup \mathcal{B}_K\}$ without (green) or with (orange) DR

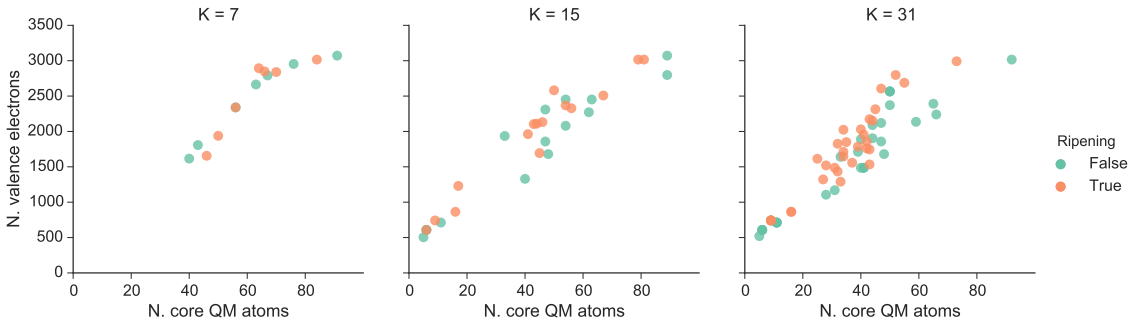


Fig. 4.8 Number of core QM atoms vs. total number of valence electrons in a QM cluster. Estimating the computational cost of a QM cluster with its total number of valence electrons, we correlate it against the sizes of the corresponding cores. Results are shown for 3 different system sizes before (green) and after (orange) the digestive ripening step.

Finally, Fig. 4.9 summarises the distributions of computational cost of the QM clusters before and after the ripening step. As hoped, the distributions are narrower when the DR algorithm is applied to improve the k–means guess, thus confirming that the developed method does indeed help balancing the computational load across independent QM calculations. The real figure of merit of the plot, though, is the maximum number of

valence electrons: the whole ensemble QM calculation is paced by its slowest calculation, which usually happens to be a cluster of abnormally large size. In all observed cases, the DR refinement does reduce the maximum values but only by a marginal amount: this is probably due to the limitations imposed by the atomic structure at hand: both the very open structure of SiO_2 and the presence of crack surfaces conjure up to not allowing for sufficient rearrangement of the subgraphs. Additionally, the buffer region sizes are bound to be extremely variable, since clusters with more surface atoms unavoidably contain far fewer buffer atoms than more 'bulk–like' clusters. Nevertheless, it is quite a positive indication that the DR algorithm never worsens the cluster size distributions; it is also worth noticing that even a small reduction in problem size can lead to a sizeable reduction in computing time, as the scaling of a DFT code is above–linear and up to $\mathcal{O}(N^3)$.

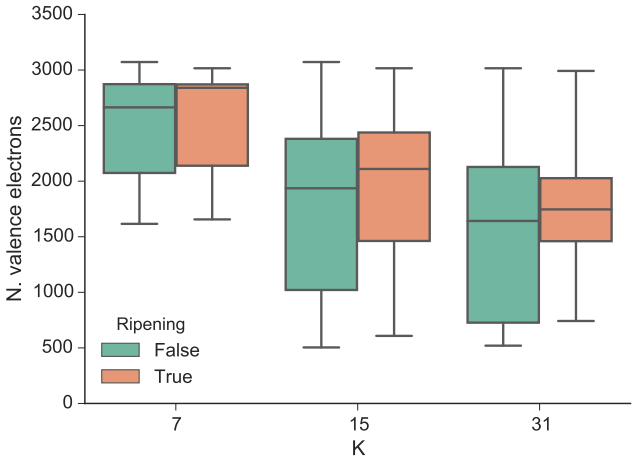


Fig. 4.9 Computational cost vs. partitioning algorithm. Here the computational cost of a QM cluster is estimated as its total number of valence electrons. Results are shown for 3 different system sizes before (green) and after (orange) the digestive ripening step.
In the boxplot, the box identifies the first and third quartile and the median, and the 'whiskers' represent the maximum and minimum values.

Ensemble QM/MM Scaling

The goal of this section is to analyse the effectiveness of the ensemble QM/MM framework. Since the QM calculation step is by far the most time consuming part of these simulations, we focussed on comparing the computational cost and time to solution of a complete DFT force evaluation of the ensemble QM approach against the standard single QM cluster approach. The calculations have been performed on the *Mira* IBM BG/Q machine located

at the Argonne Leadership Computing Facility. Most of the calculations have been performed with the planewave DFT package VASP 5.3 [80] using the PBE functional to treat the electronic exchange and correlation; the parallelisation of this code is limited to MPI, so on a BG/Q machine an executable can run on up to 16 processes per compute node (one per compute core, no hyper–threading). The largest single QM region calculations, explicitly flagged, have instead been carried out using the DFT package CP2K 2.4 [81] using a Goedecker–Teter–Hutter pseudopotential with PBE exchange–correlation functional with parameters producing a similar accuracy to the VASP calculations; CP2K is both MPI and OpenMP parallel, so it can make use of up to 64 processes per BG/Q compute node.
A free parameter of the ensemble scheme is the number of CNs assigned to each DFT executable N_{CN}: some preliminary benchmarks show that the VASP code can scale satisfactorily on up to 64 compute nodes (1024 MPI processes), so all of the ensemble QM calculations of the presented results have been performed at a fixed N_{CN} = 64.

Weak scaling The first benchmark is on *weak scaling*: the problem size per unit of processing power is kept constant, and the time to solution is monitored as problem size and allocated computing power are increased simultaneously. The problem size in our calculations is the number of forces to be evaluated on the core QM atoms of $\mathcal{C}$, which is the same as its number of atoms $|\mathcal{C}|$. The physical system is the crack tip of amorphous silica of Fig. 4.6 described earlier, where the problem size is a multiple of 218 atoms depending on how many times the periodic unit cell is replicated along the crack direction. From a physics perspective, replicating the system along the crack front direction can be necessary when studying local effects on the crack tip, such as kink formation and migration and chemical reactions [82]. In the figure, the coloured atoms make up the core QM region $\mathcal{C}$, which comprises the smallest set of atoms that needs to be treated at DFT accuracy that guarantees a trustworthy description of crack propagation. The buffer radius, controlling the accuracy of the calculated QM forces, is set to 7 Å: according to Ref. [69], we can estimate that this introduces a mean absolute error of at least 0.2 eV Å^{-1} on the computed forces. While being barely acceptable for a MD simulation, this value can nevertheless be accepted for our scaling benchmarking aims; a larger buffer size results in QM clusters too large to be evaluated by a VASP calculator, thus needing to extend the support from the current ensemble QM infrastructure to more modern DFT calculators such as CP2K.

Given the constraints of the BG/Q ensemble jobs tiling, the number of DFT executables – hence the number of parts K in which to divide $\mathcal{C}$ – has been set to 7, 15 and 31 (2^N - 1), for the total computational power $N_{\mathrm{DFT}} = K \cdot N_{\mathrm{CN}}$ = 448, 960 and 1984 CNs; the

corresponding problem sizes $|\mathcal{C}|$ are 218, 436 and 872 atoms. The timings have been taken from 4 subsequent QM steps of a LOTF MD, where all the calculations are continuations of the previous as they satisfied all of the QM region update requirements previously described. The TTS for each QM step is the maximum walltime among all cluster calculations.

A direct comparison with the equivalent calculations with a single VASP instance is not possible, as the smallest QM calculation for $|\mathcal{C}| = 218$ producing a cluster of $|\mathcal{C}| + |\mathcal{B}| \sim 1300$ atoms performed on 512 CNs is both too large a problem size and too large a computational power allocation for that DFT engine. Instead, these monolithic DFT calculations are possible when using the CP2K code, therefore we performed a small number of QM calculations (3 for $|\mathcal{C}| = 218$ on 512 CNs, 1 for $|\mathcal{C}| = 436$ on 1024 CNs) to compare our ensemble method with one of the most performant and scalable DFT codes available.

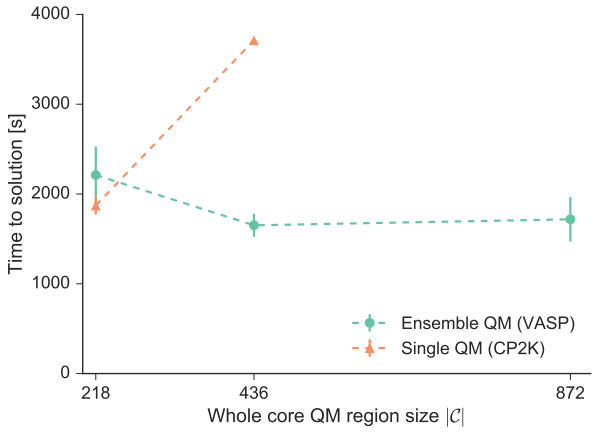


Fig. 4.10 Ensemble QM – weak scaling. The DFT forces of an increasingly large core QM region are evaluated as the computing power increases commensurately, and the time to obtain all $|\mathcal{C}|$ forces is measured. Perfect linear scaling corresponds to a horizontal line. The vertical bars represent 2 standard deviations of the timings among the set of QM steps

The results, shown in Fig. 4.10, highlight that the TTS remains approximately constant with our ensemble method, which can therefore be considered as perfectly linear scaling. The slight decrease in TTS at higher $|\mathcal{C}|$ should not be considered as a 'better than perfect' linear scaling, but rather as a variability due to the outlier cluster size of a given partitioning. On the contrary, a single QM calculation is far from being linear scaling: at $|\mathcal{C}| = 218$, the better efficiency of CP2K against VASP still guarantees a slightly lower TTS, but it increases dramatically for double the problem size, and the third data point for CP2K in Fig. 4.10 is missing because the calculations did not terminate within a walltime of 2 hours. We can

easily explain the good scaling of the ensemble QM calculation by observing that, since K increases with $|\mathcal{C}|$, the clusters are of approximately the same size for all problem sizes $|\mathcal{C}|$. Furthermore, our ensemble framework is tied to the VASP DFT code only in the current implementation; a better performing code would lower the TTS of each parallel calculation while at the same time maintaining the perfect linear scaling.

Finally, it is important to point out that the low number of data points is due to their abnormal cost that can hardly be justified for the sole purpose of benchmarking: from looking at the numbers, we can assume that each force evaluation took about 0.5 hours with computing resources of up to 32768 cores, for a single timestep cost of up to 16000 CPUh.

Strong scaling The second benchmark is on *strong scaling*, which measures the speedup obtained by allocating increasing computing resources to a problem of fixed size. The physical system can no longer be the amorphous SiO_2 crack front as before because, as we already pointed out, it is too large to be evaluated by a single VASP DFT executable. We chose instead a two unit cells wide periodic crack front in crystalline Si $(110)[1\bar{1}0]$ comprising $|\mathcal{C}| = 111$ atoms: the covalent nature of its chemical bonds makes the interactions predominantly short–ranged, thus allowing for smaller buffer regions that reduce the CPU–hours burden of such tests [65, 69].

The results of Fig. 4.11 show that in the single executable approach the calculations cannot be sped up by simply increasing the number of cores allocated – in fact, the value $N_{\mathrm{CN}} = 64$ of our ensemble calculations has been set exactly because scaling tests showed that that was the scaling limit of the VASP code on a BG/Q machine.
As the computational resources increase, in an ensemble QM approach $\mathcal{C}$ is divided into a progressively larger number K of parts, while the resources per cluster N_{CN} remains constant. Effectively, by varying N_{DFT} and consequently K, the ensemble method gives a tuneable degree of freedom for the calculation, so that one can select the typical cluster size that maximised the efficiency of the DFT code in use. In the results of Fig. 4.11, we see that by increasing K from 1 to 3 and then 7 the computational cost of the complete DFT force evaluation progressively decreases to find a minimum because the atomic clusters get smaller and smaller and more manageable by the VASP code; at $K = 15$, the trend is inverted because the cluster sizes become dominated by the buffer and most of the information computed is discarded.

On a side note, let us turn our attention to the bare numbers of the computational cost, measured in CPUh, of one QM force evaluation in the two systems. For a combination of increased buffer size (necessary because of long–range interactions) and larger crack front

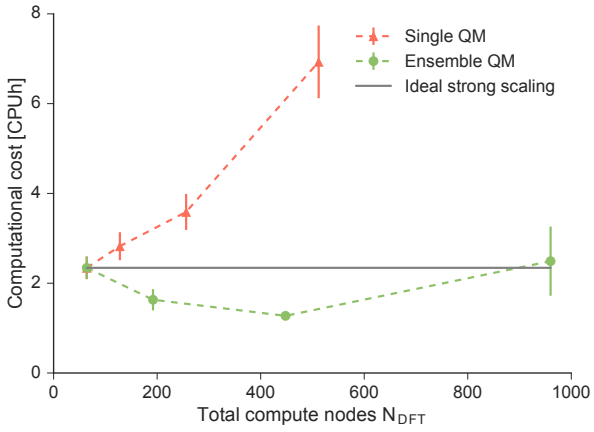


Fig. 4.11 Ensemble QM – strong scaling. We measure the total computational cost of a force evaluation on a QM core region of fixed size (of a Si crack tip specimen) against the total number of compute nodes assigned to DFT calculations N_{DFT}. Ensemble QM calculations are carried out for K = 1, 3 , 7, 15 parts corresponding to overall computational resources N_{DFT} = 64 $\cdot K$ CNs. Single DFT calculations are carried out with N_{DFT} = 64, 256, and 512 compute nodes. Ideal scaling would correspond to constant computational cost

length (necessary to mimic an amorphous material), we can estimate that a simulation in amorphous SiO_2 use 10^3 times more resources than the same simulation in crystalline Si. If we aim at performing QM/MM MD simulations in amorphous oxide systems as they are currently routinely being performed on covalent materials, it is clear that believing in Moore's law and waiting for its effects to happen is not enough. Instead, there must be incisive advances in the computational methods used: in such context, the frontier of machine learning methods for atomistic systems such as the ones presented in Chapter 6 must be explored.

4.3 Summary

In this chapter we introduced a whole QM/MM computational framework tailored at enabling QM/MM calculations of systems that require extreme computational effort, as in the case of fracture of amorphous silica. We built the method around the idea of splitting the QM region into several subparts and perform independent calculations on them by a swarm of concurrent QM engines that communicate back and forth to the MM engine that propagates the system dynamics by means of a master process. We also devised a new algorithm to generate the best possible partitioning of said QM region so to maximise the load balance across the different calculations. The new framework allows to optimally tile

the computing resources on BlueGene machines while still treating the QM calculators as ‘black boxes’; it shows perfect weak scaling, and also allows to overcome the strong scaling limitations of performant but not well scalable DFT codes.

Chapter 5

Fracture in Two–Dimensional Silica

5.1 Fracture Mechanics in Brittle Materials

Fracture is a physical phenomenon consisting of a cascade of ruptures of chemical bonds propagating along a crack front in a specimen subject to mechanical stress, resulting in the formation of two separated surfaces from a bulk specimen. Brittle materials, of which silicate glasses represent the archetypes, are characterised by the absence of plastic deformation as the crack advances. In the continuum mechanics approach first proposed by Griffith [83], a crack is favoured to advance by a distance dl and creating a new surface $dA = h\,dl$ if the process lowers the energy of the system. During a crack advance, the stored elastic energy is released through the formation of surfaces and dispersed as heat (plastic deformation is assumed not to exist in brittle fracture). If we define the *energy release rate* as the amount of elastic energy released per unit surface created

$$G = -\frac{\partial U_{\mathrm{el}}}{\partial A} = -\frac{1}{h}\frac{\partial U_{\mathrm{el}}}{\partial l}\,, \tag{5.1}$$

and the surface energy in terms of surface energy density γ, the condition for crack advance is

$$G > 2\gamma\,. \tag{5.2}$$

The value $G_c = 2\gamma$ is known as the *critical* energy release rate, and defines the elastic load beyond which fracture propagation is thermodynamically favoured. The *fracture toughness*, that is the resistance of a material to the advance of a crack, is measured by the *stress intensity factor*. The quantity of interest for structural design is the critical stress intensity factor K_{Ic}, the 'I' standing for mode I fracture – tensile loading. In two dimensional

problems, K_{Ic} is related to G_c by the equation

$$G_c = \frac{K_{Ic}^2}{E'} \,, \tag{5.3}$$

where E' is the apparent elastic modulus that is determined by the loading geometry, *e.g.*, plane stress or plane strain. As the mechanical load is greater than G_c, continuum elastic theory [84] predicts that the steady–state crack speed v is given by

$$v = c_R \left(1 - \frac{G_c}{G}\right) \;, G > G_c \,, \tag{5.4}$$

where the asymptotic limit c_R is the speed of acoustic surface waves, known as the Rayleigh speed.

Eq. 5.2, though, does not contemplate that matter is made of atoms whose chemical bonds only break after overcoming an energy barrier that is higher than simply the surface energy per atom. This discrepancy is known as *lattice trapping*, whose net effect is to increase the measured fracture toughness (term related to the quantity G_c) of a material: a crack will only advance when the mechanical load will correspond to an energy release rate

$$G > G^* > G_c \,, \tag{5.5}$$

where G^* is the apparent critical energy release rate. The atomic nature of matter also gives rise to the so called *velocity gap*: because of dynamical effects arising from a cascade of bond ruptures at the crack tip combined with the overabundant elastic energy consequence of lattice trapping, cracks do not propagate at a speed below a certain threshold, drastically revising the ideal solution of Eq. 5.4 [85].

From a reaction kinetic perspective, then, we can picture the phenomenon of fracture as the subsequent overcoming of one free energy barrier at a time, each barrier corresponding to one bond breaking event: the energy of the final state is tuned by the elastic stress applied to the specimen which in turn varies the barrier height (see Fig. 5.1). The kinetic barrier vanishes when the stress is large enough, resulting in catastrophic fracture propagation – that is, unobstructed crack advance; Assuming that the loading method ensures a constant energy release rate, the crack advance will proceed at constant speed. For a smaller stress, instead, fracture propagation is a collection of stochastic events of bond breaking.

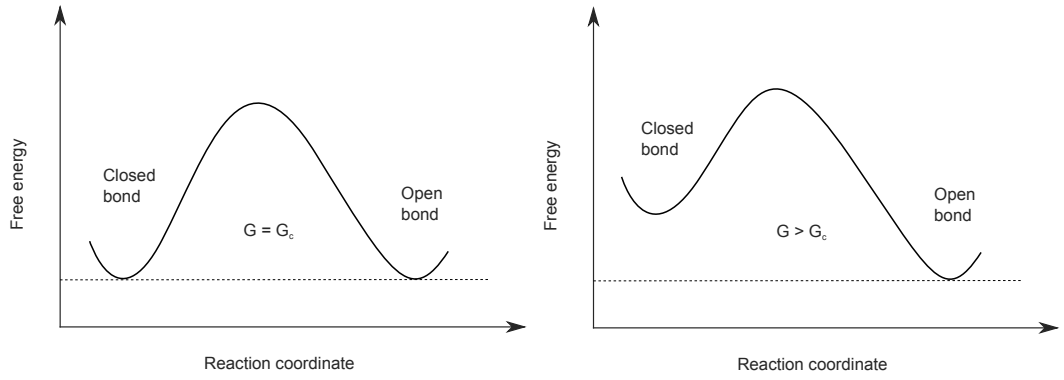


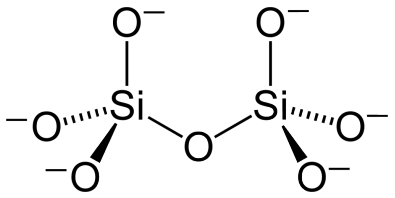
Fig. 5.1 Pictorial view of the reaction kinetics of bond breaking

5.2 Silica Structure

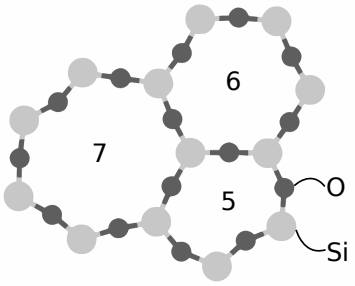
Silicates are the most abundant constituents of the Earth's crust and they are significant materials for technological fields ranging from mining to construction and semiconductor industries. Chemically pure silicon dioxide is a model oxide material, and exists in natural conditions in both crystalline and amorphous forms. In most of its forms the atoms are arranged to form a continuous network of tetrahedra, where each tetrahedron contains one Si atom at its centre and four O atoms at its corners, and each O is shared with a neighbouring tetrahedron as shown in Fig. 5.2a. Each pair of Si atoms belonging to adjacent tetrahedra is chemically bound to exactly one common O atom, with the O atom positioned as to form a Si–O–Si bridge: two Si atoms are said to be neighbours if they are bound to the same O atom or, equivalently, if they belong to neighbouring tetrahedra.

Crystalline structures of silica can be readily studied via standard diffraction techniques: SiO_2 most commonly exists as quartz, of which the α–quartz polymorph is stable at room temperature and pressure and is commonly referred as quartz. The characterisation of the atomic structure of amorphous silica is inherently a much more challenging task: over the years, experimental measurements [86, 87] and mathematical models [88–90] have helped understanding it in terms of quantities such as density, radial distribution function and O–Si–O angle distribution. Amorphous silica is considered as the prototypical example of a *network glass*, a mathematical abstraction that is subject to research in graph theory. The short–range structure is very ordered, with atoms typically positioned very close to the regular tetrahedron sites; on the contrary, the long–range structure is completely random. The middle–range structure offers instead a great amount of information about the mathematical and physical properties of the network, which are correlated to the so

called *ring statistics* [91]. A ring in a silica random network is a shortest path closed loop, and its length is defined as the number of Si atoms composing the loop: for example, in Fig. 5.2b we see one 7–membered ring, one 6–membered and one 5–membered.



(a) Short–range ordered tetrahedral structure



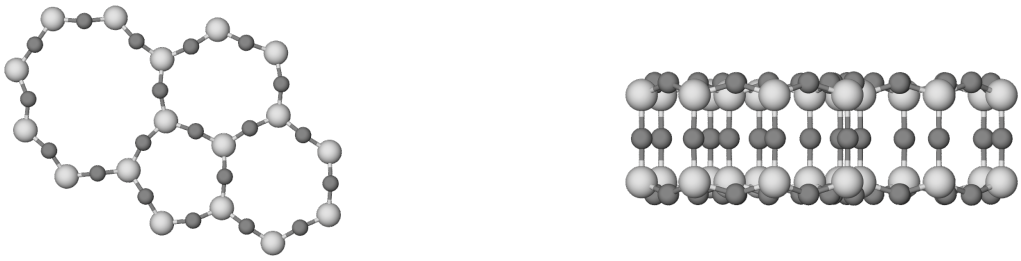
(b) Medium–range ring structure: numbers indicate ring size

Fig. 5.2 SiO_2 network glass short– and medium–range structure

From an atomistic modelling perspective, a typical procedure for constructing a unit cell of pseudo–amorphous silica is to *(i)* melt a large supercell of SiO_2 crystal (often cristobalite, as its density is the closest to glass) *(ii)* equilibrate its liquid state and then *(iii)* quench it to room temperature. If the experimental values of a set of observables are matched, the so obtained structure is considered to be a valid amorphous structure that can be used in atomistic simulations [92, 93]. Nonetheless, a picture of an atomic–resolved 3D atomic structure is yet practically unattainable, leaving a gap between experimentally observed properties of real structures and the proposed atomistic models.

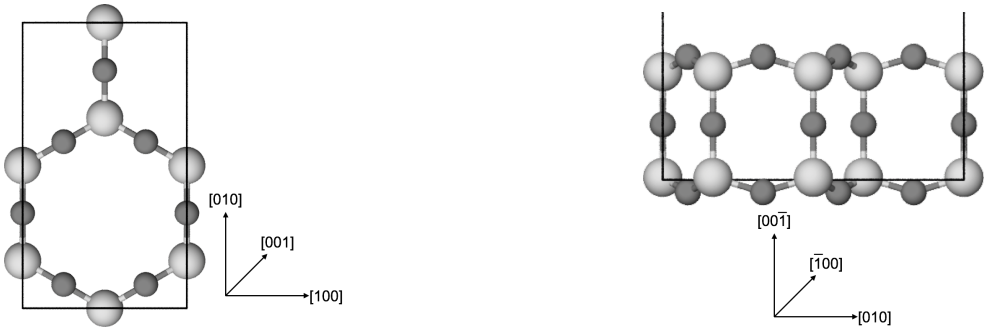
5.3 2D Silica Structure

A wealth of recent experiments have proved the possibility of growing 2–dimensional silica in both crystalline and amorphous forms, allowing to visualise the glass structure atom by atom using standard surface imaging techniques [94–97]. The atomic structure, seen in Fig. 5.5, is composed of a mirror–symmetric bilayer of Si atoms coordinated with 3 in–plane O atoms, with an intermediate layer of O atoms creating the remaining Si–O–Si bridge between the top and bottom layer. Rings orthogonal to the plane are always 4–membered; in–plane rings are 6–membered in the crystalline system resembling Tridymite SiO_2, and show a distribution peaked around 6 in the amorphous material.



(a) Top view (b) Front view

Fig. 5.3 Detail of a 2D SiO_2 amorphous structure



(a) Top view (b) Front view

Fig. 5.4 Unit cell of crystalline 2D SiO_2 structure. The cell dimension on the (001) plane is 5.30 Å $\times$ 9.18 Å according to the TS IP.

Due to its mirror symmetry, the bilayer structure does not tend to warp, contrary to monolayer structures such as graphene; interestingly, it appears to interact very weakly with the substrate on which it is grown, thus allowing the technical advantage of assuming a free standing surface surrounded in vacuum in simulations. Crystalline and amorphous structures can coexist in the same specimen and a transition can occur at their interface by atomic–scale rearrangements between ring pairs [98].

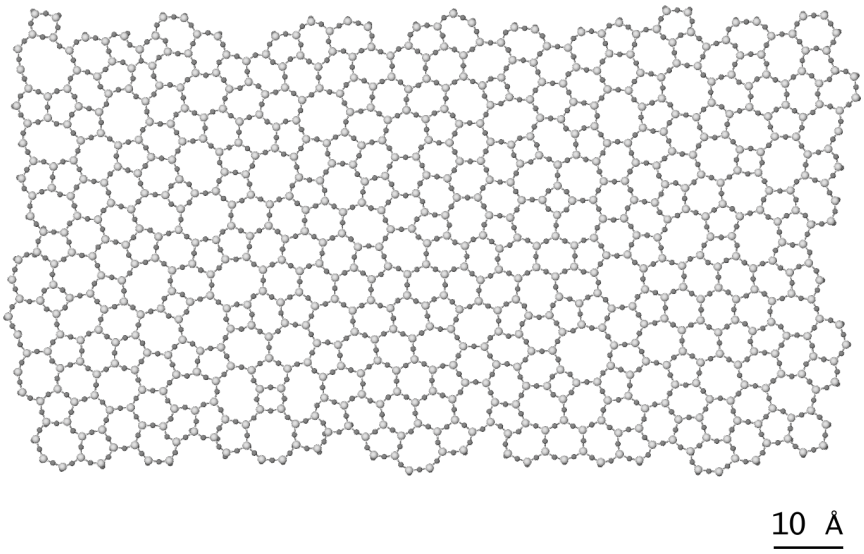


Fig. 5.5 Amorphous 2D SiO_2 structure. Atomistic model from STM image in Ref. [99]

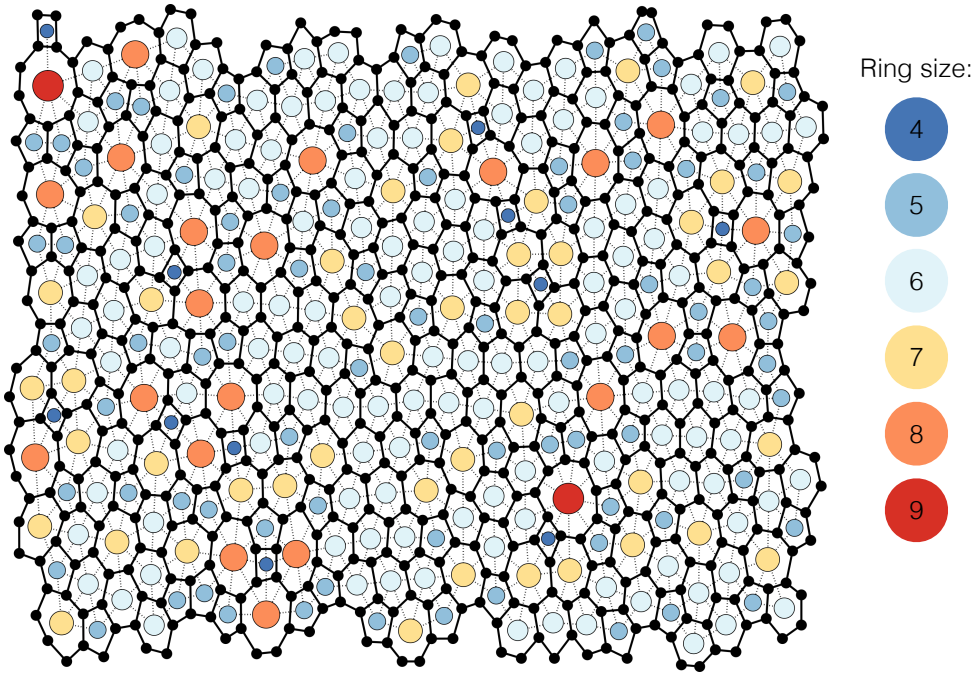
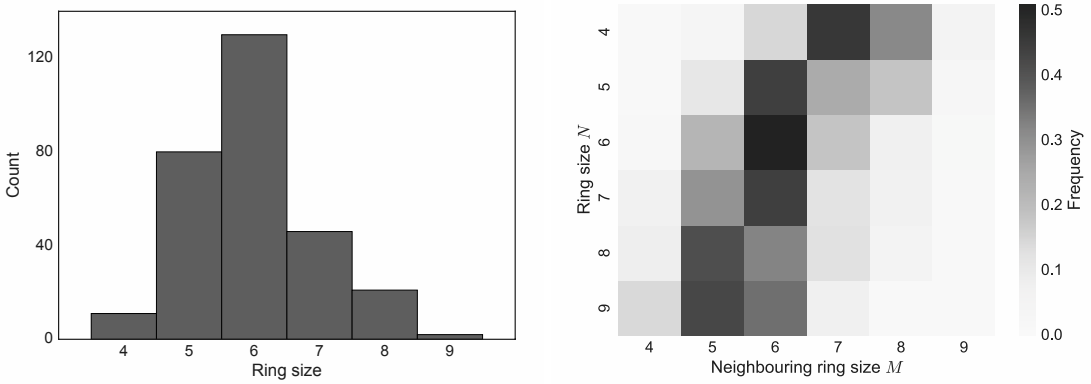


Fig. 5.6 Si random network and its dual ring structure in vitreous 2D SiO_2

In Fig. 5.5 we show the atomic structure of a silica bilayer reproduced from the STM image of a vitreous patch grown on Ru(0001) by Lichtenstein *et al.* [99]. From the experimentally observed positions of the surface Si atoms, the structure has been derived by placing a bridging O in between each pair of Si; a second layer has been replicated 4 Å below (roughly double the equilibrium Si–O bond length), and bridging O have been placed in the middle to bridge the two layers. The structure has then been optimised by means of FIRE relaxation using the short–ranged TS potential of Ref. [35] (despite the potential being fitted on bulk silica data, tests presented in this chapter will show it is an adequate potential for the bilayer structure as well). The final structure comprises 3833 atoms and its size is 123 Å $\times$ 67 Å, and has been used as the model amorphous structure for this chapter. The visualisation in Fig. 5.6 superimposes the random network of Si atoms (in black) and the dual random network of rings: each circle represents one ring, its colour encoding the size, and neighbouring rings are linked by a dotted line.

According to the model for atomic arrangement in glass of Zachariasen [88], the random structure can be obtained from the ordered honeycomb structure after a series of bond–switching mechanisms that transform two neighbouring rings of sizes (N, M) into $(N-1, M+1)$ [100]. As a result, the ring size distribution should be peaked around 6, and large rings should be more likely to be found next to small rings and vice versa. This is confirmed by the ring statistics analysis of the experimental structure: the skewed histogram in Fig. 5.7a, replicating the data analysis of Ref. [99], shows a predominance of 6–rings; the second prediction is supported by Fig. 5.7b representing the likelihood that a ring of size N is neighbour to a ring of size M, which suggests a trending line $N + M = 12$.



(a) Distribution of ring sizes

(b) Frequency of neighbouring ring size for each 'centre' ring size. Rows are normalised

Fig. 5.7 Ring statistics of amorphous 2D SiO_2

To further characterise the vitreous specimen, a Si–O–Si bond angle analysis has been performed. Thanks to the mirror–symmetry of the structure we can immediately make some considerations:

- The bond angles of Si–O–Si bridges between the two layers is constrained to be about 180°, and their distribution will be separate from the other angles and unrepresentative of the ring structure.
- The two bond angles corresponding to atoms in equivalent positions in the upper and lower layer should be the same.

Thus, the bond angle analysis has been limited to the upper layer of the structure, where each Si atom is linked to 3 oxygens 'bulging' out of the surface at an angle to be determined. The bond angle analysis has been performed on the non–boundary rings to exclude possible distortions caused by surface effects. Each Si–O–Si bridge is shared by two neighbouring rings, and the measured angles have been filtered by the ring sizes to which they correspond. In Fig. 5.8, which expands the analysis of Ref. [99], we can observe a skewed distribution peaked at around 147°, corresponding to the angle observed in the crystalline structure of Fig. 5.4 – and, notably, very close to the 144° of bulk α–quartz. The angle distribution appears to be independent of the ring size to which it belongs, and is quite different from the broader and symmetric angle distribution observed in 3D vitreous specimens. Table 5.1 analyses the data more in detail, and shows that the smaller are the rings sharing the Si–O–Si bridge, the wider on average is the angle. The explanation is straightforward: smaller rings must accommodate more atoms per unit surface, thus the smaller distance separating neighbouring Si atoms results in a more protruding O atom. Nevertheless, the angle ranges are highly overlapping, hinting that the short–range distortion of the tetrahedral structure is only slightly perturbed by the middle–range disorder.

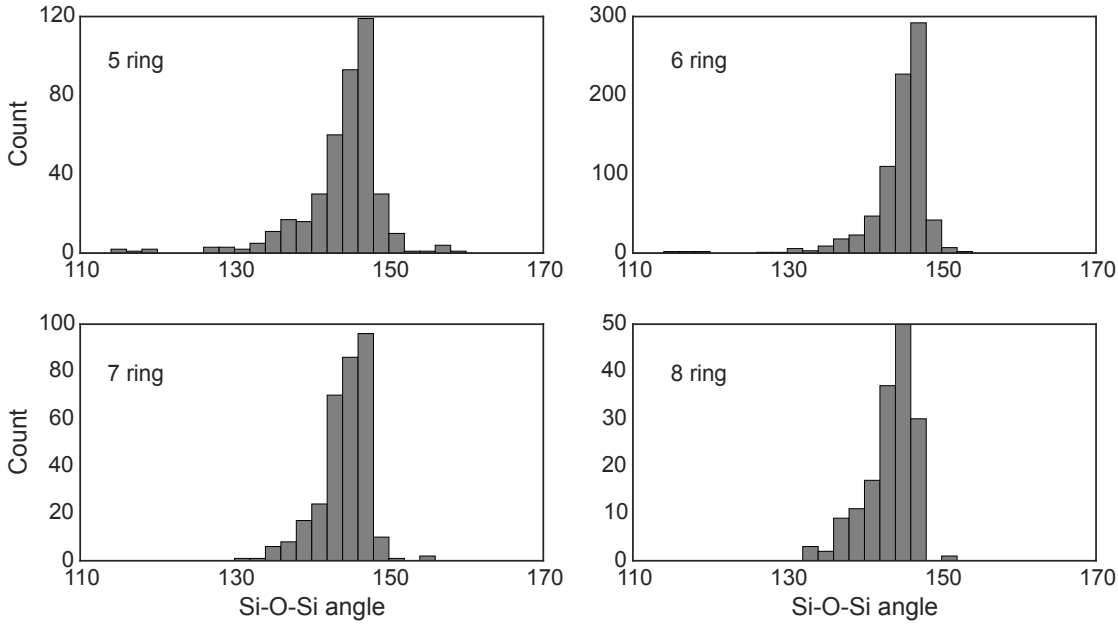


Fig. 5.8 Si–O–Si bond angle distribution on the surface of amorphous 2D SiO_2. Ring sizes 4 and 9 are omitted for scarcity of data. Each Si–O–Si bridge belongs to two rings

Table 5.1 Si–O–Si bond angle mean and standard deviation for different combinations of ring sizes

Ring size	5	6	7	8
5	146.8 ± 3.1	145.2 ± 3.7	144.2 ± 3.4	143.7 ± 2.8
6	145.2 ± 3.7	145.6 ± 2.1	145.3 ± 2.0	144.1 ± 2.5
7	144.2 ± 3.4	145.3 ± 2.0	144.9 ± 1.4	144.2 ± 1.9
8	143.7 ± 2.8	144.1 ± 2.5	144.2 ± 1.9	143.7 ± 0.9

5.4 Computational Methods

Force Evaluation

In this chapter, the candidate IP for describing 2D silica is the short–ranged TS potential, which has already been used to study quartz in a QM/MM scheme [69] and in the first simulations on silica bilayers [100]. The potential has been parametrised to best reproduce forces, energies and stresses of bulk quartz obtained through LDA DFT calculations [35].

As our 'gold standard' we have chosen DFT calculations carried out through the mixed Gaussian and planewave `QuickStep` code [101] contained in the `CP2K` package [81], using a Goedecker–Teter–Hutter (GTH) pseudopotential with PBE exchange–correlation functional. The formally correct choice of solver for the (electrostatic) Poisson equation for atomic clusters in vacuum is by means of spatially–confined wavelets: after careful evaluations showing no loss of accuracy, a periodic solver using the much faster Fast Fourier Transform libraries has been used. The procedure to calculate DFT forces on a set of core QM atoms $\mathcal{C}$ of a larger system is the following:

1. Grow a buffer region $\mathcal{B}$ containing the set of atoms within 12 bond hops from any of the core atoms and those that complete the silica tetrahedra of the set.
   The number of bond hops has been set after force convergence tests at increasing buffer size to a RMS error of 0.05 eV Å^{-1} with respect to a full bulk calculation.

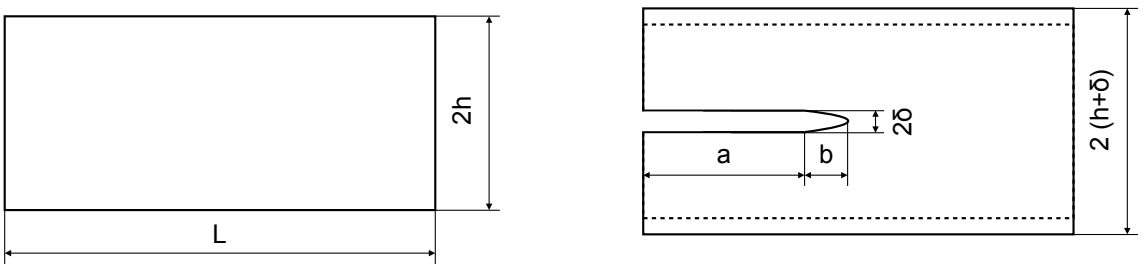
2. Carve a cluster and H–terminate the unsaturated O atoms, then add 20 Å of vacuum in each direction.
   The amount of vacuum has been verified to converge atomic forces on surface slab calculations within a RMS error of 0.02 eV Å^{-1}, where the reference value is the same calculation with 50 Å vacuum.

3. If the configuration comes from a TS simulation or if the TS and DFT forces must be compared, uniformly rescale the atomic positions by a factor 1.01244.
   The factor has been evaluated by comparing the unit cell dimensions of crystalline 2D silica at equilibrium using the two interaction models.

Loading geometry

We have used a customised thin–strip geometry to study fracture propagation. In a standard thin–strip loading depicted in Fig. 5.9, a rectangular slab of material is clamped at two opposite edges that have been displaced by a constant amount δ, and a notch is generated

at half height for crack initiation – see the `matscipy` code implementation [102]. Behind (left of) the crack tip the surfaces are displaced by an amount 2δ, whereas ahead (right) of the tip the system is loaded with an uniaxial vertical strain $\varepsilon = \delta/h$. The initial notch length is typically $a = 1/3L$, and the transition length b can be arbitrarily introduced for a quicker elastic relaxation of the system to a more blunt crack tip.



(a) Unloaded specimen

(b) Loaded and notched specimen

Fig. 5.9 Thin strip loading geometry

In a 2D specimen the vertical surfaces are free, so the stress component σ_z in the direction orthogonal to these is zero: the system is then in a so called *plane stress* configuration, and the strain energy release rate is

$$G = \frac{\partial U_{\mathrm{el}}}{\partial A} = \int_0^{2h} \frac{1}{2} E\varepsilon^2 \, dh = E\varepsilon^2 h = E\frac{\delta^2}{h} \, . \tag{5.6}$$

We must note that this is clearly an approximation, as it assumes that all the elastic energy stored in a column of base dA of bulk material under tensile stress is converted into surface energy of two surfaces of size dA part of infinitely long surfaces.

The combination of material properties of high brittleness (low G_c), stiff elastic constants (high E) and appreciable long–range interactions found in 2D silica and other oxides (large 2δ), though, brings to the surface a limitation of the loading method. The separation 2δ becomes large enough ($\sim$10 Å) to consider the two fractured halves to be non–interacting in only two undesirable cases: *(i)* when the applied G is far beyond the critical value G_c, excluding the possibility of near–Griffith velocity gap studies; *(ii)* when the height of the system h is very large, making the simulations unnecessarily wasteful of resources.

To circumvent the issue, the thin strip loading scheme has been modified by replacing the fixed atoms constraint at the boundaries with Hookean springs constraining the top

and bottom layer atoms close to the vertical position of the now removed clamps. The modified loading method is the following:

1. Given a geometry–optimised rectangular slab of height $2h$, create the notched system by applying a strain sufficient to obtain the desired surface separation 2δ

2. Add a Hookean spring constraint of spring constant k to each Si atom at the top and bottom edges as illustrated in Fig. 5.10: given an edge Si atom i of y_i^0 vertical coordinate after step 1, the restoring vertical force of the constraint is given by

$$\mathbf{f}_i = k\left(y_i^0 - y_i\right)\hat{\mathbf{y}} \tag{5.7}$$

3. To mimic an infinitely long sample and prevent the bowing of the vertical edges, fix the horizontal position of left and right edge atoms

4. Relax the system under the applied constraints by means of either the FIRE (more versatile, not guaranteed to converge) or preconditioned L–BFGS (only allows some constraints to be used, but is guaranteed to converge) with a convergence threshold on the atomic forces of $|\mathbf{f}_{\max}| < 0.01\,\text{eV Å}^{-1}$.

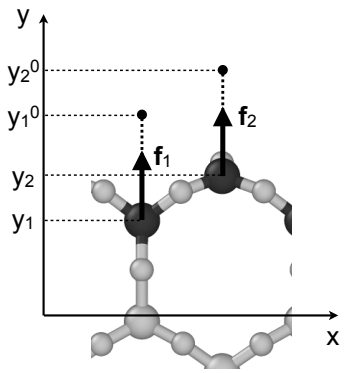


Fig. 5.10 Hookean edge constraint depiction. The Si atoms clamped to fixed points by springs are coloured in black. Here we show a detail of the top edge of a crystalline slab

The modified loading scheme allows to disjoin the crack surfaces separation and the elastic energy stored in the system: at the left of the crack tip, the system is elastically unloaded as in the standard thin strip setup but with a surface separation tuned by step 1; ahead of the tip, the overall system behaves as three springs in parallel, where the elastic energy is stored in both the material and the constraints and is tuneable by adjusting the spring constant k.

The edges of the specimen should be far away enough from the crack tip in order not to create elastic field artefacts in its neighbourhood [103]. Therefore, if we wish to compare how a crack advances in the amorphous material for different atomic arrangements of the crack tip, we must maintain the dimensions of the specimen constant – a, L, h of Fig. 5.9. To do so, the periodic strip system of Fig. 5.11 has been generated from the amorphous patch of Fig. 5.5 by carefully adding and removing atoms so to obtain matching vertical edges while retaining the correct stoichiometry and structural integrity. From this unit cell, different crack tip configurations can be obtained by slicing a horizontally–replicated supercell with a window of length L shifted by the desired amount.

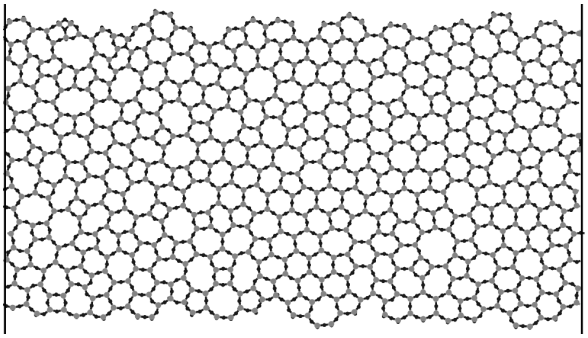


Fig. 5.11 Amorphous 2D SiO_2 structure periodic along the horizontal axis. The horizontal bars represent the unit cell boundaries

In the modified thin strip geometry, G cannot be evaluated with the straightforward Eq. 5.6 obtained for the unaltered geometry, but rather has to be estimated using the original definition of Eq. 5.1, *i.e.*, the elastic energy stored per unit surface normal to the crack surface. For a fixed elastic constant value k and an applied strain ε, G is measured using the following method:

1. Take an unstrained periodic strip of the relevant structure and perform constraint–free geometry optimisation (the periodic strip is a $1 \times N_y \times 1$ unit cell for the crystalline material and the structure of Fig. 5.11 for a vitreous specimen). Measure the potential energy U_0

2. Strain the system by an amount ε, apply the Hookean spring constraints previously described and relax it. Measure the potential energy U_ε, adding up both the atomic interaction and the spring constraints contributions.

3. Estimate the energy release rate as

$$G = \frac{\partial U_{\mathrm{el}}}{\partial A} \simeq \frac{U_\varepsilon - U_0}{A}\,, \tag{5.8}$$

where A is the nominal area of the periodic strip in the plane orthogonal to the direction of deformation.

Where in crystalline specimens this definition is the exact equivalent to Eq. 5.6 under the modified constraints, in vitreous specimens it can be thought of as the mean energy release rate averaged over the possible crack tip positions. Also, we can observe that the two limits for $k = 0$ and $k = +\infty$ correspond to the unstrained geometry and to the standard thin strip geometry respectively.

Thermalisation approach

In all of the following MD simulations, the dynamics of the system is propagated at a target temperature T_{sim} by means of Langevin dynamics approximating a NVT ensemble as implemented in the molecular dynamics package `ase` [104]. We used a constant friction coefficient γ = 0.002 atomic units. To guarantee the system to reach thermal equilibrium and to avoid sloshing of momenta, the setup involves the following steps:

1. Start from the statically optimised specimen obtained as described in the loading geometry section, and apply random momenta to the atoms according to a Maxwell–Boltzmann distribution at a lower temperature than T_{sim} – a typical starting temperature is 100 K.

2. Perform a dynamics where momenta are rescaled every few timesteps, so to linearly increase the target temperature to T_{sim} – the simulation time for this step is of the order of 10 ps.

3. Let the system equilibrate in a standard NVT dynamics – the equilibration time is about 10 ps, beyond which a visual inspection of the trajectory does not reveal any residue of momenta sloshing.

This thermalisation strategy in step 2 is aimed at speeding up the thermalisation process by curbing the sloshing of kinetic energy from one area to another of our relatively small systems, and is obtained by letting the Langevin thermostat gradually add random momenta to the atoms. We must also note that, in order to prevent a premature crack propagation, some bond lengths for atoms at the crack tip may need to be constrained.

Dynamic fracture propagation setup

The study of dynamic propagation of the crack is used to *(i)* observe how atomic bonds snap at the crack tip, *(ii)* measure the crack velocity and possibly *(iii)* analyse bond breaking event statistics and time correlations. For these kinds of simulations, the constraints are the springs at the top and bottom layers of the slab and the constraints forcing the atoms at the vertical edges to move in the vertical direction only. The loading method can either be static if the applied mechanical load is known to be sufficient to enable crack propagation during the simulation time, or increasing when, for instance, one is searching for the minimum loading that causes the crack to move (*i.e.*, the apparent critical energy release rate G^*). In the latter case, the strain on the slab ε is increased at each timestep at a rate $\dot{\varepsilon} \sim 10^{-5}\,\mathrm{fs}^{-1}$ (a 1% strain increase every ps of MD) until the first bond at the crack tip is broken, after which the strain is kept constant at the last value. To measure such event, we build an adjacency matrix for the atoms in the proximity of the crack tip, where a Si and O atom are considered to be bonded if their distance is lower than $r_{\mathrm{cut}} = 2\,\text{Å}$.

It is worthwhile to point out in advance that MD simulations can only probe crack speeds compatible with the simulated time: this will be in the order of 10^1 ps, so if we wish to observe an average of at least 2 crack advance events every 10 ps and one such event moves the crack tip forward by about 5 Å, the minimum observable crack speed is roughly $100\,\mathrm{m\,s}^{-1}$.

By solving Arrhenius equation Eq. 3.56 at a temperature of 300 K for an estimated speed of sound $c = 5 \times 10^3\,\mathrm{m\,s}^{-1}$ (such is the order of magnitude of c in quartz) and a maximum wavenumber $k = 0.3\,\text{Å}^{-1}$ (using a maximum lattice vector of 5 Å) corresponding to a maximum acoustic phonon frequency, *i.e.*, attempt rate, $\omega = ck = 10^{13}\,\mathrm{s}^{-1}$, we can also find out that the height of the free energy barrier to the crack advance must be no higher than about 0.12 eV to observe any crack propagation at all in a non–accelerated dynamics setup.

Thermodynamic integration setup

We wish to study the free energy barrier for advancing a crack by one atomic bond. To do so, we will perform MD simulations constrained to the slow growth thermodynamic integration method (SGTI). Common to the other phase space sampling methods, the TI method is highly sensitive to the choice of reaction coordinate, which determines the constraint force to be monitored. The most robust choice tested so far is the following: take the distance vector $\mathbf{d}^0_{AB}$ between two Si atoms A and B belonging to layer 1 and 2

respectively and situated at the opposite sides of the Si–O–Si bridges split by the crack advance, and set to zero its z component to obtain the new vector

$$\mathbf{d}_{AB} = \mathbf{d}^0_{AB} - \mathbf{d}^0_{AB} \cdot \hat{\mathbf{z}} \tag{5.9}$$

lying on the (x, y) plane as depicted in Fig. 5.12. The collective variable is then the in–plane distance $d_{AB} = |\mathbf{d}_{AB}|$: constraining it to a fixed value effectively results in constraining the average length between the two Si–O–Si bridges at the crack tip, which ensures that the phase space compatible to the double bond opening event is properly sampled, *i.e.*, no one bond can snap rapidly and unconstrained while the other is constrained.

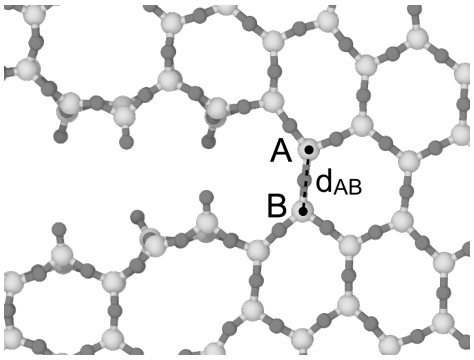


Fig. 5.12 Thermodynamic integration constraint. Atoms A and B lie on two different layers, and the distance is measured on the plane of the page.

The holonomic constraint on the distance d_{AB} is imposed during a MD by applying a constraint force on the atoms of equal magnitude and opposite direction. Given the forces $\mathbf{f}_A$ and $\mathbf{f}_B$ and the in–plane distance vector $\mathbf{d}_{AB}$, the constraint force is

$$\mathbf{f}_{\text{cnst}} = \frac{1}{2} \left(\mathbf{f}_B - \mathbf{f}_A \right) \cdot \hat{\mathbf{d}}_{AB} \, \hat{\mathbf{d}}_{AB} \, , \tag{5.10}$$

which alters the atomic forces as following:

$$\begin{aligned} \mathbf{f}_A &\to \mathbf{f}_A + \mathbf{f}_{\text{cnst}} \\ \mathbf{f}_B &\to \mathbf{f}_B - \mathbf{f}_{\text{cnst}} \, . \end{aligned} \tag{5.11}$$

In the slow growth method, the reaction coordinate must typically increase by a constant tiny amount Δd at each MD timestep: this has been implemented by adjusting the atomic

positions of atoms A and B as following:

$$\begin{aligned} \mathbf{r}_A &\to \mathbf{r}_A - 0.5\,\Delta d\,\hat{\mathbf{d}}_{AB} \\ \mathbf{r}_B &\to \mathbf{r}_B + 0.5\,\Delta d\,\hat{\mathbf{d}}_{AB}\,. \end{aligned} \tag{5.12}$$

Cumulatively integrating the irreversible work done by the constraint forces in this succession of these constrained displacements one obtains the full free energy profile for the bond opening process.

We must note that the inverse process of 'closing back' the bonds is not taken into account in the free energy calculations. Since each bond rupture event is an irreversible process and so is the inverse process of bonding, both are hysteretic: if the double bond opens at d_o it will then only close at $d_c = d_o - \delta < d_o$, and the phase space region where both bonds are broken and the distance is in the range $[d_c, d_o]$ is not spanned by the crack advance processes that will be the focus of our calculations.

5.5 Tangney–Scandolo Interatomic Potential Benchmark

Fracture simulations require accuracy in three key quantities:

- Elastic constants, defining the elastic behaviour of the material;
- Surface energy, defining the energetic cost of creation of new surfaces as a result of crack advance;
- Atomic forces at the crack tip, determining the reaction path of bond rupture events.

The aim of this section is to assess the validity of the TS IP in dealing with fracture simulations on two–dimensional silica: each one of the three quantities are evaluated in an independent test, and their results are assessed together in a final commentary.

Elastic modulus

In regards to the elastic constants, the benchmarking has been limited to measuring the effective Young's modulus E, which is the one elastic quantity determining the fracture toughness in a thin strip geometry. As a further limitation, E has only been evaluated in the crystalline system, as a full DFT treatment of the entire vitreous specimen is out of question and a measurement on a small subpart of it would inevitably be biased by surface effects. The elastic modulus has been calculated by progressively elongating the unit cell

along the [010] 'armchair' direction (using the graphene terminology given the structural analogy), and relaxing the atomic positions at each step. According to

$$U = \int_V \frac{1}{2} E \varepsilon^2 dV \,, \tag{5.13}$$

the curvature of the parabola fitting the strain–energy curve is equal to $0.5EV$. V is the unit cell volume, where the nominal thickness along the [001] direction has been set to 4 Å, roughly equal to the separation between layers.

The result is shown in Fig. 5.13: the elastic modulus measured by DFT is 463 GPa, and the result obtained using the TS IP is 523 GPa.

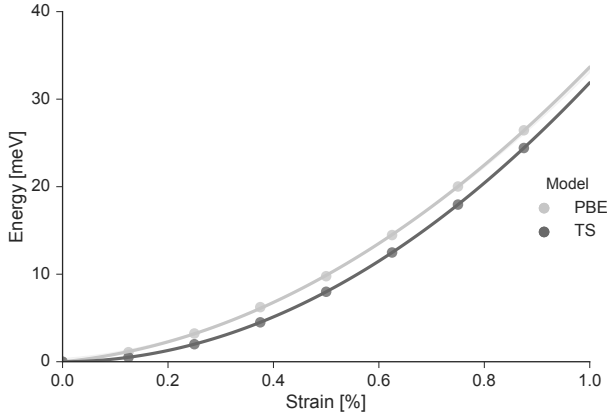


Fig. 5.13 Energy–strain curves for elongation along [010] of 2D SiO_2 – PBE vs. TS

Surface energy

The surface energy evaluation has been performed on the crystalline system first. From the vitreous patch of Fig. 5.11, we can see that the average orientation of the silica rings in a mid–height horizontal crack would favour the creation of zigzag rather than armchair surfaces. Therefore, we will concentrate on zigzag crystalline surfaces.

The standard computational approach for measuring the (010) surface energy (let (010) be the $\hat{\mathbf{y}}$ direction) consists of *(i)* generating a $N_x \times N_y \times 1$ supercell from the fully optimised unit cell, *(ii)* increase the cell dimension along [010] to add vacuum and obtain a slab with two surfaces and *(iii)* relax the geometry of the slab. According to Fig. 5.4a, the creation of two (010) 'zigzag' surfaces involves breaking two Si–O bonds per unit cell, resulting in two dangling O atoms each of which can end up on either surface. Each unsaturated Si–O

protrusion can be thought of as an electrostatic dipole, thus the surface energy will strongly depend on the specific configuration of dipoles lying on the top and bottom surfaces. A set of calculations has been performed using the TS IP for all possible combinations of surface termination for systems up to $N_x = 4$ unit cells long; after convergence tests, the vertical supercell size has been set to $N_y = 5$ and the vacuum size to 20 Å.

As expected for a system of dipoles of roughly the same direction, the minimum energy system corresponds to the alternate two–unit cell configuration depicted in Fig. 5.14 that maximises their distance.

Given the energy E_{bulk} of a bulk unit cell containing N_{bulk} atoms and the energy of the surface slab E_{surface} containing N_{surf} atoms, the surface energy γ is calculated by the formula

$$\gamma = \frac{E_{\mathrm{surf}} - E_{\mathrm{bulk}}\, N_{\mathrm{surf}}/N_{\mathrm{bulk}}}{2A}\,, \tag{5.14}$$

where A is the nominal surface area of the slab, here obtained assuming a nominal bilayer thickness of 4 Å.

According to the TS IP, the surface energy per unit cell per surface ranges between 4.8 eV (minimum energy configuration of Fig. 5.14) and 6.8 eV (all oxygens lying on one of the two surfaces). The minimum surface energy density of the (010) surface is then 3.6 J m^{-2}. We hypothesise that the hierarchy of energy as a function of surface oxygen arrangement is determined by dipole–dipole interactions: since electrostatic interaction is accurately described by the TS IP, we can expect the hierarchy not to change in DFT calculations. Thus, we evaluated the minimum energy configuration of Fig. 5.14 by means of DFT, obtaining a value of 5.2 eV per unit cell per surface, corresponding to a surface energy of 3.9 J m^{-2}.

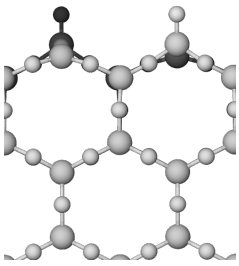


Fig. 5.14 Minimum energy (010) surface of crystalline 2D $\mathrm{SiO_2}$. Grey shades represent coordinate along the [001] axis.

To obtain an estimate for the surface energy density of vitreous $\mathrm{SiO_2}$, the semi–periodic system of Fig. 5.11 has been cut in half horizontally, preserving whole rings and placing the surface oxygens according to the staggered minimum energy ordering as shown in Fig. 5.15. The nominal surface energy density of the vitreous specimen calculated with the TS IP is

3.8 J m^{-2} (or 0.95 eV Å^{-1} if expressed per unit length of the surface), just slightly higher than the value obtained in the crystalline system. Due to the large system size, in this case a direct comparison with DFT results is not possible. A summary of these surface energy calculations results is presented in Table 5.2

Table 5.2 Summary of surface energy density calculations. The convergence criterion is $|\mathbf{f}_{\max}| < 0.01\,\text{eV Å}^{-1}$

System	Model	γ [J m^{-2}]
Crystal (010)	IP TS	3.6
Crystal (010)	DFT	3.9
Glass	IP TS	3.8

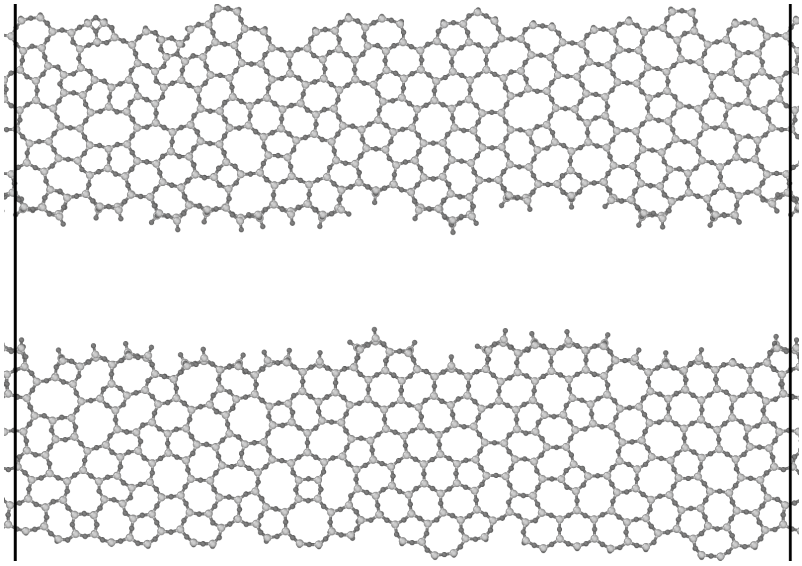


Fig. 5.15 Surface of vitreous 2D SiO_2, after geometry relaxation. The atoms of top and bottom edge rings are kept fixed to avoid unwanted surface reconstructions on the preexistent surfaces during structural optimisation.

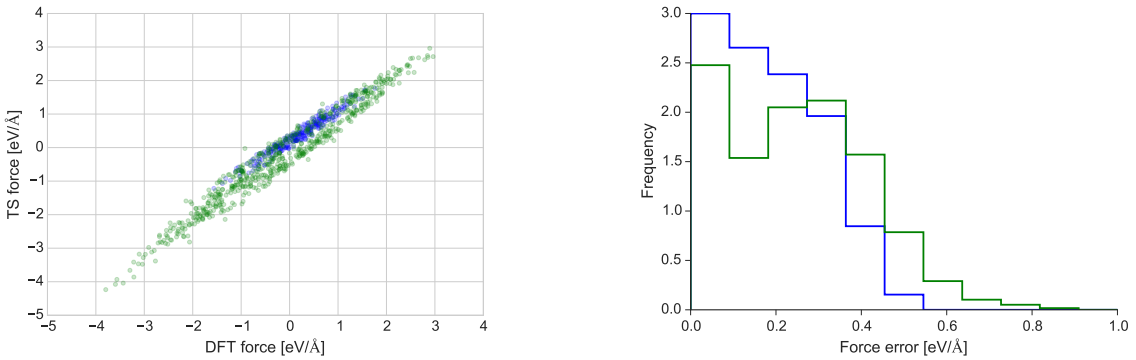
Forces at the crack tip

In order to evaluate the effectiveness of the TS IP to estimate atomic forces during crack advance, we concentrated our benchmark on the constraint force f_{cnst} of Eq. 5.10. The dataset consists of two SGTI trajectories of crack advance in which a single 6–membered ring opens in vitreous 2D glass at 300 K at high G = 2.36 G_c and near–Griffith G = 0.96 G_c loads. The trajectories have been obtained by evolving the system according to the TS model, and the MD has been stopped after the integrated TS free energy barrier was overcome – *i.e.*, when the Si–O–Si bridges of both layers are completely broken. At this point, we calculated the DFT constraint forces on an evenly time–spaced subset of the MD configurations.

Remembering that in a cluster calculation the DFT constraint force itself is converged to an RMS error of 0.05 eV Å^{-1}, we can observe from Fig. 5.16 that the TS potential is generally able to follow the force trend but with significant error bars. The relevant figure of merit for Fig. 5.16 is presented in Table 5.3, where we measure the accuracy of the TS constraint force $f^{\mathrm{TS}}_{\mathrm{cnst}}$, expressed in terms of RMS deviation with respect to the reference $f^{\mathrm{DFT}}_{\mathrm{cnst}}$ values.

Table 5.3 Benchmark of SGTI constraint force

k [eV Å^{-2}]	G/G_c	N. DFT forces	$f^{\mathrm{DFT}}_{\mathrm{cnst}}$ RMS mean [eV Å^{-1}]	$f^{\mathrm{TS}}_{\mathrm{cnst}}$ RMS error [eV Å^{-1}]
$+\infty$	2.36	286	0.63	0.23
0.7	0.96	644	1.38	0.30



(a) Scatter plot of f_{cnst}

(b) Histogram plot of absolute errors in f_{cnst}

Fig. 5.16 Comparison between TS and DFT constraint forces in SGTI. The colour highlights the dataset: green $G = 0.96\ G_c$, blue $G = 2.36\ G_c$

To better understand what is the effect of such errors in the actual measurement of a free energy barrier ΔF, I compared the integrated TS barrier with a hybrid accuracy TS–DFT barrier (phase space sampling according to the TS potential, integrated force from the DFT model). Such hybrid method is by no means intended to replace a fully accurate DFT free energy barrier calculation, but rather as the best estimate of ΔF profile that does not require QM/MM MDs that are unmanageable for the typical QM cluster sizes of about 650 atoms and a simulation time of around 10 ps.

The QM constraint force evaluation has been limited to the part of the SGTI trajectory around the peak of the free energy profile, since it is the only one relevant to the calculation of the barrier. These free energy profiles are shown in Fig. 5.17.

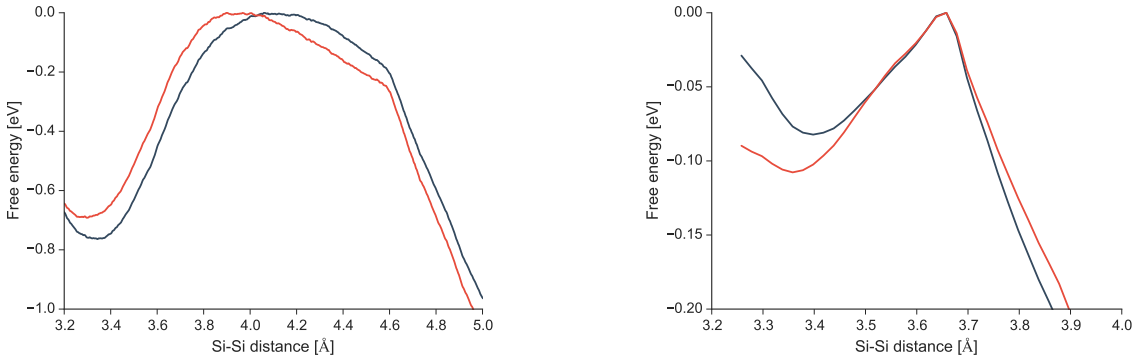
Despite the small discrepancies of ΔF evaluated by the TS or the hybrid TS–DFT models, it is worth noticing that the free energy barrier height enters in the calculation of Arrhenius rate constant as $\exp(-\beta\Delta F)$, thus amplifying small differences. For this reason, both the free energy barrier heights and their relative rate constants at 300 K are reported in Table 5.4.

Table 5.4 Benchmark of SGTI free energy barrier height

G/G_c	Model	ΔF barrier [eV]	Rate constant
0.96	TS	0.69	2.4×10^{-12}
0.96	TS–DFT hybrid	0.76	1.5×10^{-13}
2.36	TS	0.11	1.5×10^{-2}
2.36	TS–DFT hybrid	0.08	4.2×10^{-2}

Summary

The results from this section show that the TS interatomic potential, while clearly not being accurate to the highest degree, is nevertheless reproducing target DFT properties within an acceptable level of confidence. We must highlight that our classical potential has been fitted on neither surface configurations nor two–dimensional systems, therefore its positive performance is as welcome as it is unexpected. In regards to the specific task of evaluating free energy barriers for crack opening, our benchmark hints that the discrepancy between the two interaction models is moderate, and that the results are qualitatively comparable. We can then conclude that a meaningful computational investigation on the mechanisms of crack advance can be performed in a purely classical framework.



(a) Free energy barrier for $G = 0.96\ G_c$

(b) Free energy barrier for $G = 2.36\ G_c$

Fig. 5.17 Comparison between TS and DFT free energy barriers in SGTI. The colour highlights the model: red TS, grey DFT

5.6 Results: Dynamic Crack Propagation

The aim of this study is to visualise how cracks propagate dynamically and relate the differences in observed behaviour to their atomic structures. For the task we prepared two specimens, one crystalline and one vitreous, of a sufficient length along the crack direction to evaluate a steady state crack velocity. In both cases, the strain ramp loading method has been used to be able to observe fracture propagation at relatively low elastic loads; the target temperature of all calculations is 300 K.

Crystalline specimen

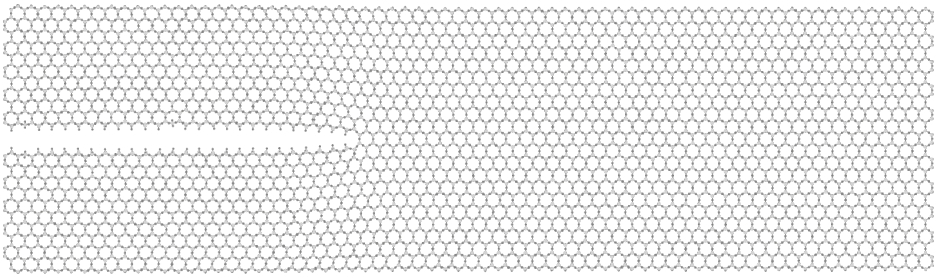


Fig. 5.18 Crystalline specimen for crack MD study, 369 Å × 106 Å

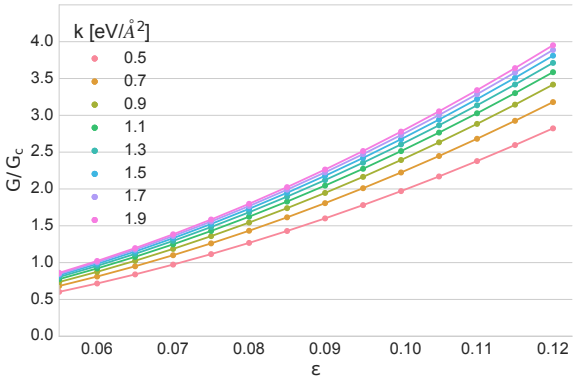


Fig. 5.19 Energy release rate vs. strain curves of crystalline 2D–SiO_2, 70 × 11 supercell slab, strained along [010], clamped to Hookean constraints of spring constant k. $G_c = 2\gamma = 7.28\,\mathrm{J\,m^{-2}}$

The crystalline specimen, shown in Fig. 5.18 is a 70 × 11 supercell containing 18286 atoms and of in–plane size of 369 Å × 106 Å; the fracture system is [100](010) to obtain the minimum energy zigzag surface. A set of calculations have been carried out at different initial strains so to obtain different starting configurations for the crack propagation. All

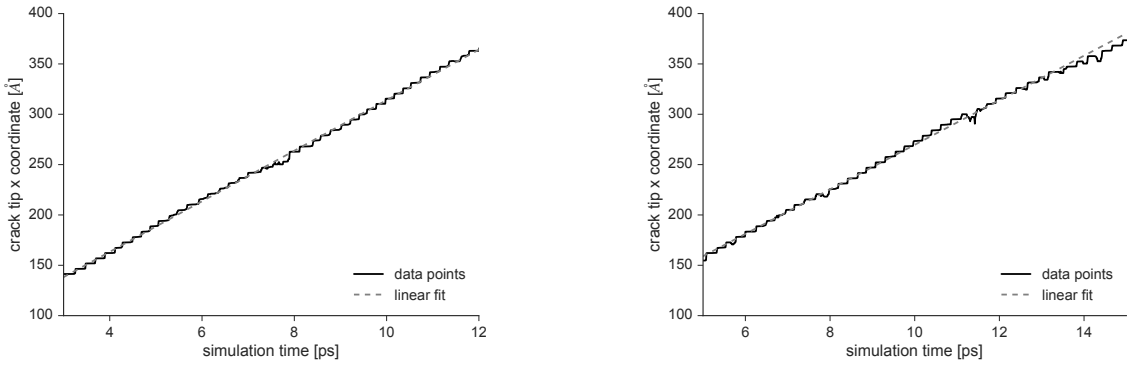
runs have been carried out at room temperature, and using Hookean constraints of spring constant $k = 1.9\,\text{eV}\,\text{Å}^{-2}$ at top and bottom edges (this choice is arbitrary); atoms at the left and right edges are constrained to move along the vertical direction only to prevent edge warping.

Prior to MD calculations, we evaluated the energy release rate G as a function of the vertical strain ε and of the constraints spring constant k consistently with our definition of Eq. 5.8 by measuring the potential energy per unit length of the strained system. In Fig. 5.19, we can observe that $G_c = 2\gamma$ for $k = 1.9\,\text{eV}\,\text{Å}^{-2}$ is reached for a strain $\varepsilon \simeq 0.06$.

The measurement of the crack tip position is automated in a very basic implementation: it is defined as the x–coordinate of the Si atom of the highest momentum in the y direction. The rationale for this choice is that the rupture of a Si–O–Si bridge at the crack tip causes a rapid separation of its two Si atoms, which will travel in opposite directions in roughly the y direction. Furthermore, the observed crack speeds are so high that as the crack tip atoms of a rupturing bond reach their relative surfaces distance, the Si–O–Si bridge ahead is already in the process of breaking. To reduce the noise in the position measurement, the raw data points are postprocessed by a median filter with a window size of 180 fs (*i.e.*, approximately the time necessary for a Si–O–Si bridge to change its state from closed to open).

In a simulation starting with an initial strain of 0.055 below the critical value, the crack did not advance until a strain $\varepsilon = 0.103$ was reached, corresponding to $G \simeq 2.8\,G_c$: once the first bond is broken, the crack quickly reaches a steady state of constant speed of $2400\,\text{m}\,\text{s}^{-1}$ as can be seen in Fig. 5.20a. In a different simulation with a higher initial strain of 0.074, the crack propagated at a strain $\varepsilon = 0.096$ ($G \simeq 2.5\,G_c$) and at an average speed of $2200\,\text{m}\,\text{s}^{-1}$. The coordinate of the crack tip for this simulation can be observed in Fig. 5.20b.

The frequency of bond breaking events appears to be paced by the time the system takes to transfer the elastic load from the last broken Si–O–Si bridge to the next intact bridge ahead: the rupture of a double Si–O–Si bridge at the tip causes a sudden acceleration in opposite directions of rings just above and below, so that the Si–O–Si double bridge at the new crack tip position experiences a temporary and localised strain increase. Granted that the energy release rate is sufficient for crack propagation – as it will always be for $G > G^*$ sufficient to start crack propagation – this 'surf effect' guarantees that the fracture will propagate at a high speed in a barrier–free catastrophic mode. To help visualising the process of crack propagation in the crystalline structure, Fig. 5.21 presents snapshots of a bond breaking event.

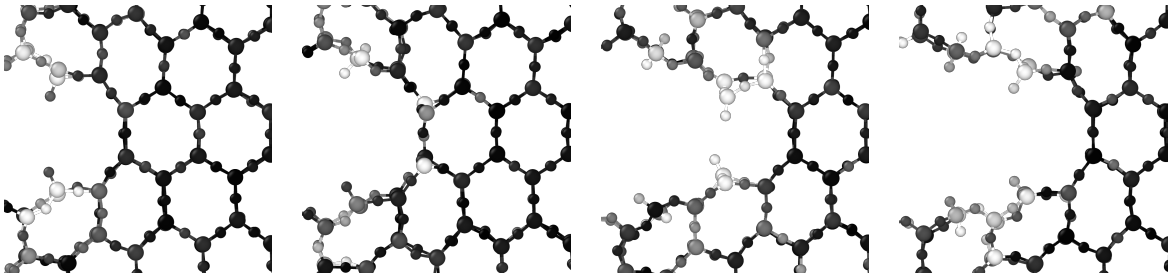


(a) G = 2.8 G_c (b) G = 2.5 G_c

Fig. 5.20 Crack tip position in crystalline 2D SiO_2 fracture MD

Concurrently, another dynamical phenomenon takes place to instead limit the speed of crack propagation. After an initial acceleration of the crack tip in which the bonds break to form ordered surfaces, the conspicuous heat generation near the crack tip gives enough kinetic energy to the tip atoms so that less ordered and more energetic structures are formed: the creation of a disordered crack tip can momentarily slow down the crack propagation, which results in an alternate acceleration/halting pattern of the crack tip motion. In line with this physical intuition, it has been observed that simulations at higher G are more prone to an intermittent crack motion. This phenomenon is not specific to the atomic structure under investigation but rather to the excess elastic load, as qualitatively similar behaviours have been observed in different materials such as silicon [105].

Overall, we can then expect this material to have a narrow range of possible crack propagation speeds and a very clear velocity gap. To verify the presence of a velocity gap we modified the original setup by reducing the spring constant k of the edge constraints for all springs whose x–coordinate is 100 Å or more ahead of the initial crack tip position. By doing so, in the first 100 Å the fracture will propagate subjected to a load $G_1 > G^*$ to then slow down when entering a region of lower energy release rate $G_2 < G_1$: if the crack stops, we then deduce that it cannot propagate at the load G_2; if instead it propagates at a lower speed, we will deduce that cracks can propagate at G_2 but the event of breaking the first bond is not likely enough to be observed in a standard simulation timescale. It must be reminded that, because of the time limits inherent to our MD simulations, a crack advancing at a speed lower than about 100 m s^{-1} is indistinguishable from a still crack.



(a) Δt = 0 fs (b) Δt = 80 fs (c) Δt = 160 fs (d) Δt = 240 fs

Fig. 5.21 Crack advance snapshots in crystalline 2D silica fracture MD. Atoms are coloured by their kinetic energy, from low/black to high/white. Here we show one bond opening: (a) mechanical load is transferred to the Si–O–Si bridge at the crack tip, (b) the chemical bonds break and the bridge atoms start separating, (c) the tip atoms gain large amounts of kinetic energy during the bond opening, (d) the crack has advanced by one step

Table 5.5 Observed crack speeds in crystalline 2D SiO_2. The load G is estimated by linearly interpolating the data points from Fig. 5.19. The crack speed is estimated by the slope of the linear interpolation of the crack tip position vs. time graph

Setup	ε	k [eV Å^{-2}]	G [J m^{-2}]	G/G_c	speed [m s^{-1}]
uniform k	0.096	1.9	18.5	2.5	2200
uniform k	0.103	1.9	21.5	2.8	2500
k drop	0.105	0.95	19.5	2.6	2200
k drop	0.105	0.5	15.8	2.2	0
k drop	0.103	0.7	17.2	2.4	0

The results of Table 5.5 confirm our hypothesis: if there exists a range of elastic loads allowing a crack speed between 100 m s^{-1} and 2200 m s^{-1}, it must be within a narrow window between 17.2 J m^{-2} and 18.5 J m^{-2}. With the appropriate unit conversions, we can translate this information to the energy gain $G-2\gamma$ for a crack advance of one unit cell (5.3 Å) by double Si–O–Si bridge breaking: a crack cannot propagate at a measurable speed if the energy gain is lower than 13 eV per unit cell, and it will instead propagate unconstrained when the energy gain is at least 15 eV per unit cell.

Such results can only be explained by the presence of a very high free energy barrier to bond opening at G_c, which remains larger than the estimated $\sim$0.12 eV even when the end state of the crack advance reaction is 13 eV below the initial state; when the elastic load is high enough to finally lower the barrier enough, the energy gain per bond breaking reaction is so high that the crack immediately propagates at the highest possible speed

for the material while dissipating large amounts of energy. The prediction on the barrier height as a function of the load will be tested by the thermodynamic integration results of the next section.

Vitreous specimen

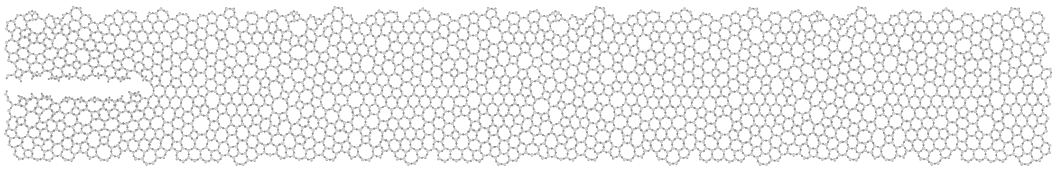


Fig. 5.22 Amorphous specimen for crack MD study, 490 Å × 74 Å

The vitreous specimen, shown in Fig. 5.22, has been built by replicating 4 times the vitreous periodic strip; it contains 15608 atoms and its in–plane size is 490 Å × 74 Å. We performed a series of simulations at a temperature of 300 K, and at different lengths of the crack seed to study whether the ring structure at the crack tip affects how a crack begins to propagate.

The MD trajectories show that fracture preferentially propagates by rupturing larger rings, deviating its trajectory from a straight line when the tip encounters small rings; we show an example of crack advance in Fig. 5.23. We can observe that:

- breaking of size 4 rings has never been observed,
- breaking of size 5 rings only occurred once in all simulations in a highly disordered and almost 'explosive' crack propagation at high load,
- breaking of size 6 rings is common but only occurs when the ring is aligned exactly in front of the crack tip,
- breaking of size 7 rings is preferred, even at the cost of deviating the direction of crack propagation,
- breaking of size 8 and 9 rings can occur in the bulk of the material for the typical strains necessary to observe crack advance, resulting in the formation of voids ahead of the crack tip;

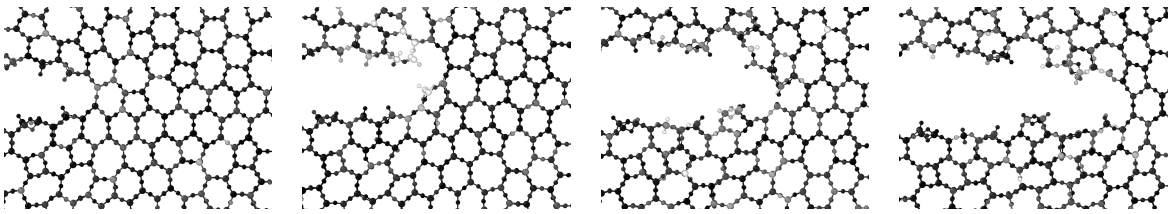
In summary, larger rings are weaker.

The data from our MD simulations shows that, at some point during its motion, the crack tip will encounter a configuration of ring sizes that does not allow any further advance:

at this point, the rings near the crack tip have enough time to rotate so to reconfigure the atomic structure into a blunt tip that reduces the elastic stress. If at this point the elastic load is increased to restart the crack propagation, or if throughout the simulation the elastic load is high enough to avoid crack halting, the excess potential energy is released by formation and growth of voids within the bulk of the material along fault lines that connect large (hence weaker) rings far away from the original crack tip, as shown in Fig. 5.24. From this observation, it is possible to hypothesise that fracture in vitreous two–dimensional silica occurs by coalescence of voids.

Interestingly, the voids show a quasi–periodicity that follows that of the atomic system, suggesting that the percolation of voids has a strong deterministic component, *i.e.* for a given ring configuration, the path of neighbouring ring edges along which fracture occurs is unique. Assuming that the crack advances following a minimum energy path, we can picture the fracture of the vitreous system as the stepwise motion of a fictitious particle (the crack tip) being pulled along the crack propagation direction, encountering a set of possible moves choices at each step (which ring to break next); the probability of a move is related to the free energy barrier height to breaking the ring, which is in turn related to the ring size. For the path to be deterministic, the barrier heights must be substantially different so that only the event of lowest energy barrier occurs at each step of crack propagation. From these considerations, we can assume that the fracture path is predetermined by the topology of the ring structure and by the direction and intensity of the applied stress.

The aim of the SGTI calculations of the following section will be to give a more quantitative tool to help explaining the crack advance mechanisms in these MD simulations.



(a) Δt = 0 ps (b) Δt = 0.45 ps (c) Δt = 0.93 ps (d) Δt = 1.43 ps

Fig. 5.23 Crack advance snapshots in vitreous 2D silica fracture MD. Atoms are coloured by their kinetic energy, from low/black to high/white. (a) the 6–ring at the crack tip is about to break, (b) the crack does an upwards step to break a 7–ring rather than a straight 5–ring, (c) the crack does one downwards step to break another 7–ring instead of a straight 6–ring, (d) the crack advances straight by breaking more 6–rings

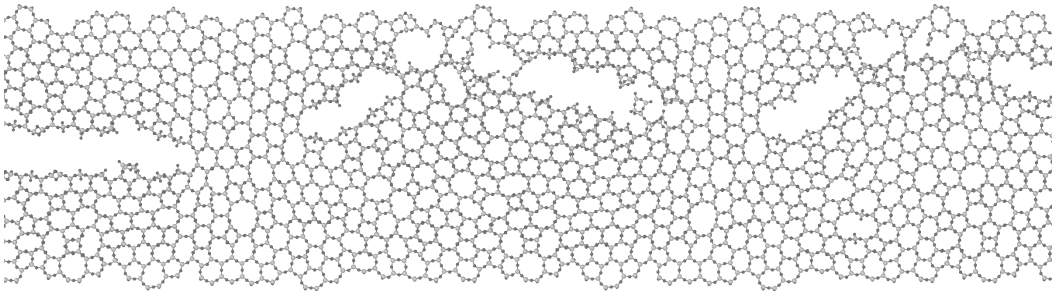


Fig. 5.24 Void formations in MD fracture simulation of amorphous 2D SiO_2. After the crack front on the left stopped for a strain $\varepsilon = 0.096$, the load was smoothly increased. Rather than restarting, at a strain of $\varepsilon = 0.109$ a void forms and grows at a Si–O–Si edge shared between a 8–ring and a 7–ring forming a new crack front; further increasing the strain, away from both crack fronts more 7/8 and 6/8–ring edges break to form new voids that grow and coalesce

5.7 Results: Free Energy Barrier for Crack Advance

In the previous section a connection between atomic structure around the crack tip and mode of propagation of a crack was hypothesised: fracture in the crystalline system propagates on a straight line at a steady pace due to the structural order; the disorder of the vitreous system, instead, results in irregular crack motion and uneven fracture surfaces. In this section we will perform a set of SGTI calculations for a quantitative analysis of the free energy barriers to the crack advance that are responsible for the emergence of the dynamical behaviour of a crack.

As the final free energy of the system after bond breaking will certainly be lower than the value before breaking for the process to be thermodynamically favourable, the key quantity of interest will be the barrier height for each mechanism at each load. The simulations have been performed on specimens of similar dimensions, and the strain $\varepsilon = 0.10$ has been kept constant across all simulations so to have a separation between the cracked surfaces of about 8 Å, thus minimising the measurement errors due to different intensities of surface–surface interactions. The energy release rate is tuned by the spring constant k of the edge constraints, and the ring opening mechanism is selected by defining the collective variable of the SGTI simulation.

For our slow growth thermodynamical integrations to be slow enough to properly sample the phase space corresponding to a given range of values of the collective variable, we fixed its speed to 0.5 Å ps^{-1}, which corresponds to at least 8 ps of simulation time for the observation of one event that draws the the opposite tip Si atoms from an initial distance (projected along the crack front direction) of about 3.2 Å to a final value of 7 Å to 8 Å.

Crystalline specimen

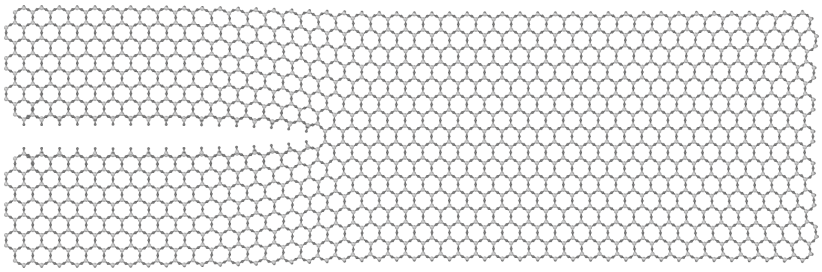
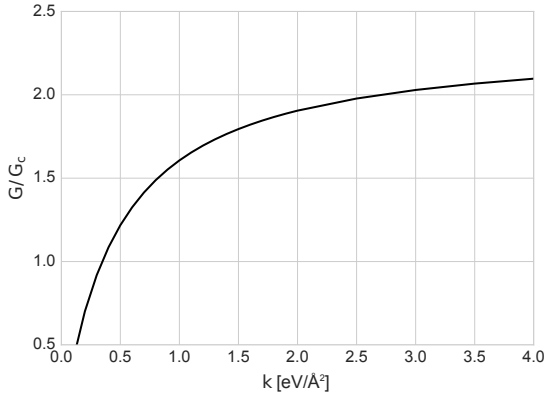


Fig. 5.25 Crystalline specimen for crack advance SGTI study, 245 Å × 78 Å

The foremost aim of this investigation is to gain detailed understanding about how the shape and height of the barriers as a function of the energy release rate G are responsible

for such large lattice trapping and velocity gap. The specimen used in this set of simulations is shown in Fig. 5.25: according to the data in Fig. 5.26, the critical release rate G_c is reached for a spring constant of the edge constraints of 0.35 eV Å^{-2}. The crack advance mechanism has been simulated at increasing loads beyond G_c.



k [eV Å^{-2}]	G [J m^{-2}]	G/Gc
0.35	7.3	1.01
0.45	8.4	1.16
0.60	9.6	1.33
1.00	11.7	1.61
2.00	13.9	1.90
4.00	15.3	2.10

Fig. 5.26 Energy release rate vs. spring constant curve of crystalline 2D SiO_2, measured on a 64 × 8 supercell strained ε = 0.10 along [010], clamped to Hookean constraints of spring constant k. $G_c = 2\gamma = 7.28\,\mathrm{J\,m^{-2}}$

The results are condensed in Fig. 5.27. The studied mechanism involves the asynchronous rupture of two bonds holding together the Si–O–Si bridges at the crack tip, which is reflected by the presence of two 'cusps' in the free energy profile (we here use the term 'cusp' loosely to indicate a sudden change in first derivative over a short interval of values of the collective variable). Dividing the bond breaking process in three parts delimited by the cusps positions, from left to right we can see the elastic response of both Si–O–Si bridges when still intact, the the elastic response of the one remaining Si–O–Si bridge and the elastic relaxation after the crack has advanced by one step. The equilibrium length of a Si–O–Si bridge, corresponding to the first minimum is about 3.2 Å and slightly increases with the elastic load; the second minimum, not shown in the plots, corresponds to the equilibrium distance of the two fractured surfaces at the crack tip, and will be smaller than the 8 Å separation between surfaces far away from the tip. All three parts of the profile can be approximated by a quadratic curve, in agreement with a linear elastic behaviour. The smoothness of the change of the first derivative around the cusps signals that the failing chemical bond is hopping between the bound and unbound states over a transition interval. Interestingly, the intermediate 'half open' state appears to be metastable in only a narrow range around G/G_c = 1.16: only there there exists a local minimum between the two bond breaking events – yet a shallow one of no more than 0.15 eV – that would allow the crack

to remain half open for a finite amount of time, thus decorrelating the two bond breaking events. The curve for G/G_c = 2.10 appears as an outlier because of an instability of the tip ring after the rupture of both bonds causing an artificially induced reconstruction beyond a reaction coordinate of 4.25 Å.

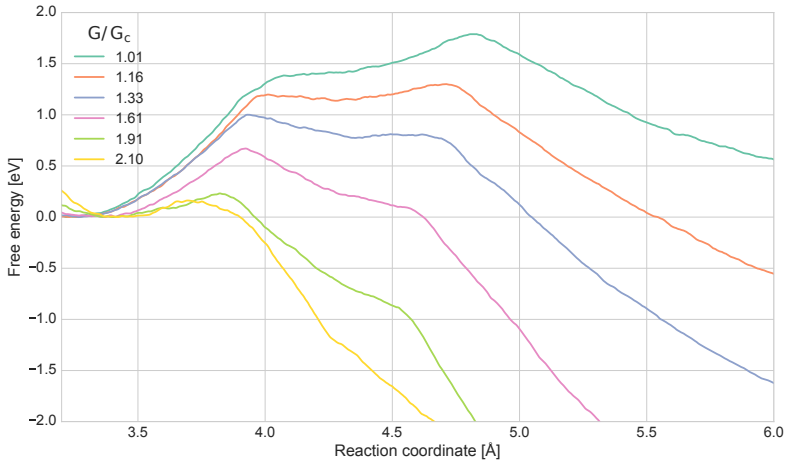


Fig. 5.27 Free energy barriers to crack advance in crystalline 2D SiO_2 for a set of energy release rates G. The minimum of the closed configuration sets the zero for each curve. The plot is cropped to highlight the free energy profiles around the peaks

By further distilling the SGTIs data, we obtain the plots of Fig. 5.28 showing the value of collective variable and free energy of each peak. The residuals of the linear fits for the heights of both peaks are almost negligible, following the standard behaviour of activated processes, so the peak height ΔF can be written as a function of the energy release rate G as

$$\Delta F(G) = \Delta F(G_c) - \alpha\, G \ , \ \alpha > 0\,; \tag{5.15}$$

the reaction coordinate position of the peaks, instead, appear to decrease and get closer. Increasing the energy release rate effectively 'tilts' the free energy curves by lowering the free energy of the final state, along with a shift to the left and a compression of the half open region interval in collective variable coordinate. Since there is no metastable intermediate state at G = 1.01 G_c, the height of the free energy barrier is the one of the right peak at 1.78 eV: such a large barrier slowly decreases as the elastic load increases, and its value is still 0.17 eV for more than twice the critical energy release rate. Overall, these calculations explain the origin of the abnormally large lattice trapping and velocity gap observed in the dynamical crack propagation simulations, confirming that they originate from large and persistent free energy barriers to crack advance. According to the linear fit of Fig. 5.28

for the first peak height, the limit for lattice trapping should be at a vanishing barrier height for approximately $G^* = 2.2\ G_c$, whereas the MD simulations implied a larger lattice trapping $G^* \geq 2.4\ G_c$. It is possible that such discrepancy is due to the particular loading method used to decelerate the crack in the dynamical simulations, and that a uniform and time–independent energy release rate larger than $2.2\ G_c$ would allow crack propagation.

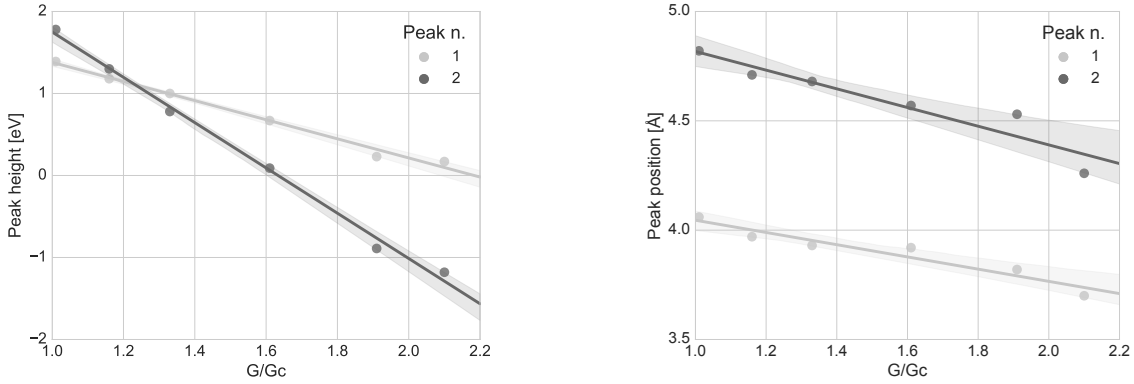


Fig. 5.28 Location and height of the two free energy peaks for crack advance in crystalline 2D SiO_2. The measured points are represented with dots, the solid lines are their linear interpolation, and the solid shade represents the 95% confidence interval of the interpolation

Vitreous specimen

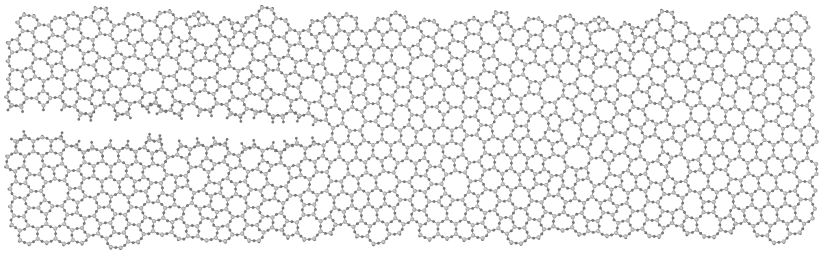
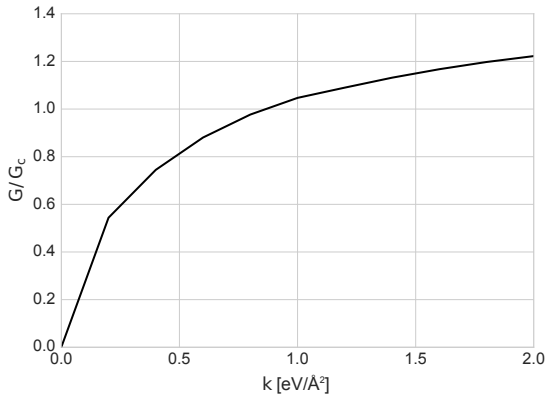


Fig. 5.29 Example of vitreous specimen for crack advance SGTI study, 249×74 Å^2

Let us picture the path taken by the crack advance in the amorphous system as a chain of events where each of which consists of a multiple choice among ring ruptures of different free energy barriers determined by the local ring structure: the aim here is to perform a set of SGTI calculations on different crack tip configurations to better understand why the observed fracture mechanism of the amorphous specimen is so different from its crystalline counterpart. The atomic structure of the specimen used for this set of calculations is shown in Fig. 5.29. In the loading geometry used, the energy release rate is controlled by the spring

constant of the edge constraint, and is parametrised according to the calibration results of Fig. 5.30. The calculations here presented are constrained to one energy release rate equal to 0.98 G_c = 7.6 J m^{-2}, so to obtain a set of free energy barriers for the selected subset of crack tip configurations as a function of the path taken for the crack advance.



k [eV Å^{-2}]	G [J m^{-2}]	G/Gc
0.2	4.2	0.54
0.4	5.8	0.74
0.6	6.9	0.88
0.8	7.6	0.98
1.0	8.2	1.05
1.2	8.5	1.09
1.6	9.1	1.17
2.0	9.5	1.22

Fig. 5.30 Energy release rate vs. strain curve of vitreous 2D SiO_2, strained ε = 0.10 and clamped to Hookean constraints of spring constant k. $G_c = 2\gamma = 7.8\,\mathrm{J\,m^{-2}}$

From considerations about the ring structure, if the last broken ring is of size N, we can estimate the number of possible paths to be taken by a crack in the next step as $N/2$. Given that N is most likely to be 6 or 7, to identify the ring structure of a crack tip we will use the notation N_1–N_2–N_3 where N_i are the three ring sizes in clockwise order that are most likely to be broken in an actual fracture propagation. The results are presented by putting the atomic structure of the crack tip and the corresponding free energy barriers side by side; the strain is always in the vertical direction, and the letters in the left side structures label the double Si–O–Si that is being broken in a certain fracture path; the letter labels and the colours of the free energy profiles are in order of probability of the event to take place, from highest/a/green to lower/b/orange to lowest/c/blue.

Firstly, we can observe in Fig. 5.31 and 5.32 that the free energy profiles for the straight (label a) crack advance of the crystal–like 6–6–6 tip ring structures are identical in the two configurations that only differ in surface reconstructions and in ring sizes of atoms farther away from the tip; in addition, their shape is the same as the ones obtained in the crystalline calculations; the free energy barrier height, defined by the first bond breaking event, is 0.70 eV to 0.75 eV. Crack advances in directions other than orthogonally to the applied stress are severely hindered, and asymmetries in the corresponding free energy profiles (labels b and c in Fig. 5.31) are likely to be due to the stress conditions caused by the different neighbouring ring structures.

In the second group of structures, comprising Fig. 5.33 and 5.34, the 'straight' direction of crack propagation involves the rupture of a 5–membered ring, an unlikely event according to the MD simulations earlier presented. This observation is confirmed in the SGTI calculations: the barrier height for breaking a straight 5–ring is about 1 eV (right peak), against the 0.7 eV to overcome to break an off–plane 7–ring. Since glassy structures of 2D–SiO_2 are generated by a set of bond switching mechanisms from a honeycomb crystal, it is likely for a 5–ring to be next to a 7–ring, so crack tip configurations like the two structures presented are rather common. Notably, we also gain more evidence that the ring size is not the only feature determining the barrier to crack advance: both rings corresponding to mechanisms a and c in Fig. 5.34 are 7–membered but c is the least likely to happen because their shape is very different.

Fig. 5.35 presents us with a different problem: the previous bond breaking caused the crack to deflect from a straight line, so that the new 'straight' crack advance is not orthogonal to the applied stress direction (vertical) and none of the Si–O–Si bridges at the new crack tip are vertically aligned. Perhaps counterintuitively, the rupture of the 7–membered ring b is unfavourable with respect to the equally misaligned 6–membered ring b – 1.05 eV against 0.65 eV. To explain that, we must note that in this particular atomic arrangement the opening of ring b is hindered by the 5–membered ring protruding out of the surface as seen on the snapshot superimposed to the plot: its presence effectively reduces the mobility of the atoms constituting the opening ring, thus reducing its breadth of elastic relaxation.

Another insightful crack tip configuration is the one presented in Fig. 5.36. Any bond opening is very unfavourable in this atomic arrangement: every free energy barrier is much higher than any other encountered so far at the same load, and only one (label a) presents a local minimum for the open state. This is the most disordered crack tip structure among the investigated ones, as it presents 4 to 8–membered rings and an uneven crack surface; in our MD simulation, similar configurations cause a sudden arrest of a moving crack, unless a much higher energy release rate is applied to lower the 3.2 eV barrier – a mechanical load for which larger rings within the bulk break apart and create voids.

The crack tip configuration with the most similar peak heights is the one in Fig. 5.34, where the barrier height of mechanism a is 0.27 eV lower than the one of mechanism b. From a reaction kinetic perspective, we can calculate the ratio between the rate constants of two mechanisms of crack advance by ruptures of different rings from Eq. 3.56 at 300 K, finding that ring a is more likely to break by a factor of about 3×10^5. Such large value is

the smallest probability ratio observed, and supports the hypothesis that the path for the crack advance is nearly deterministic in this amorphous material.

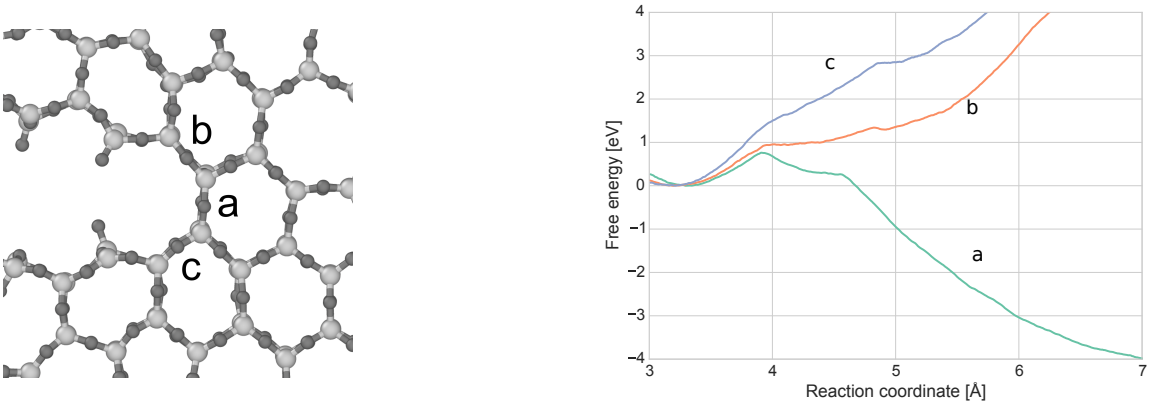


Fig. 5.31 6–6–6 amorphous crack tip and corresponding free energy barriers to crack advance. Barrier a height: 0.76 eV

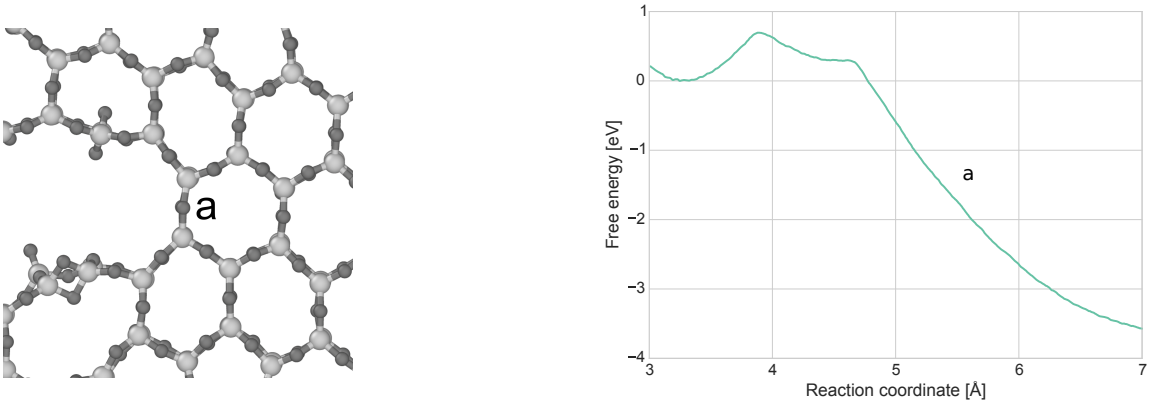


Fig. 5.32 6–6–6 amorphous crack tip and corresponding free energy barrier to a straight crack advance. Barriers for off–plane advancements are omitted given the structural similarity to Fig. 5.31. Barrier a height: 0.70 eV

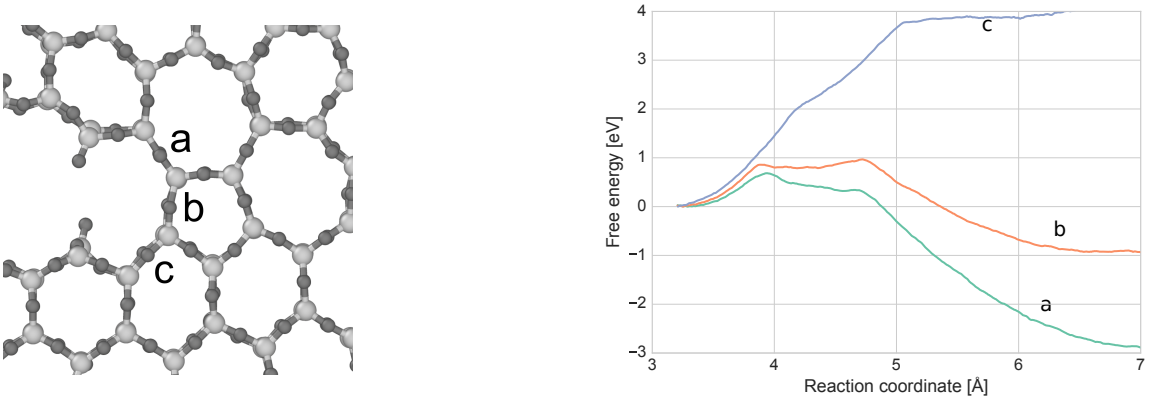


Fig. 5.33 7–5–6 amorphous crack tip and corresponding free energy barriers to crack advance. Barrier heights: a 0.69 eV, b 0.97 eV (right peak)

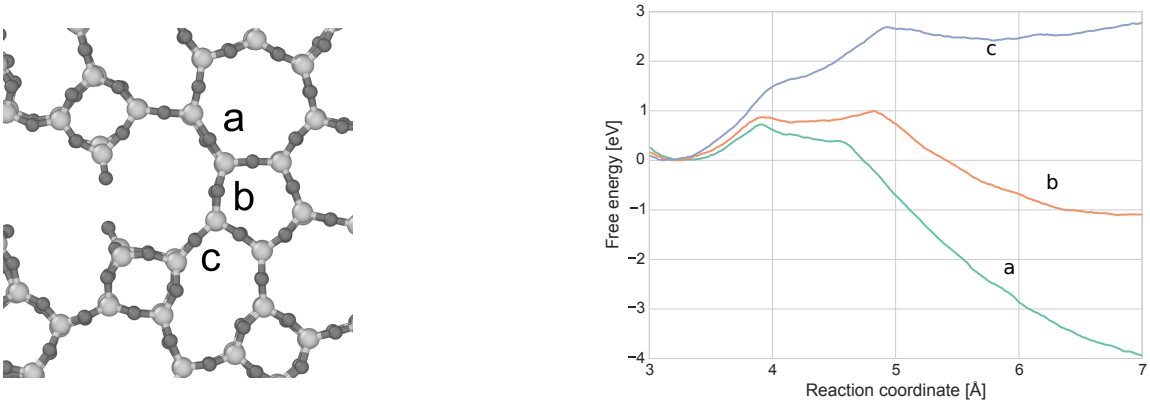


Fig. 5.34 7–5–7 amorphous crack tip and corresponding free energy barriers to crack advance. Barrier heights: a 0.73 eV, b 1.00 eV (right peak)

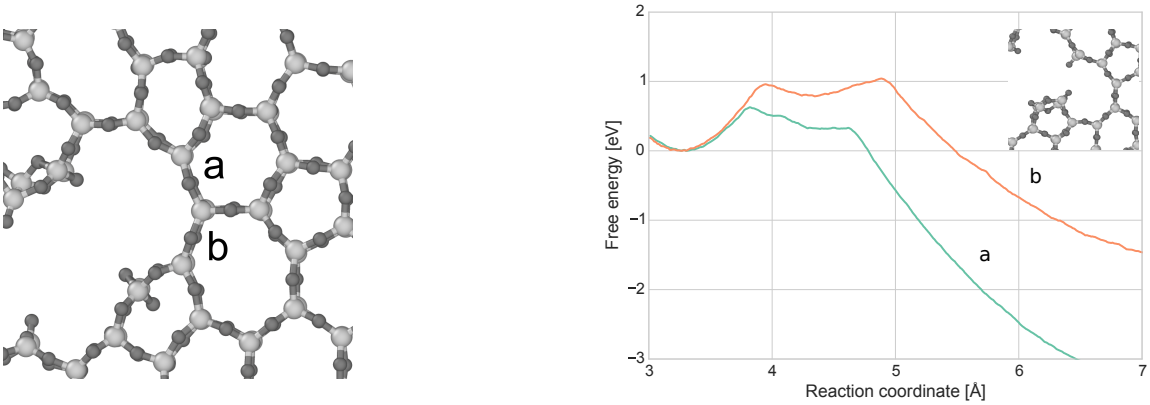


Fig. 5.35 5–6–7 amorphous crack tip and corresponding free energy barriers to crack advance. The rupture of the 5–membered ring is omitted, as it would be very unfavourable. Barrier heights: a 0.63 eV, b 1.04 eV (right peak)

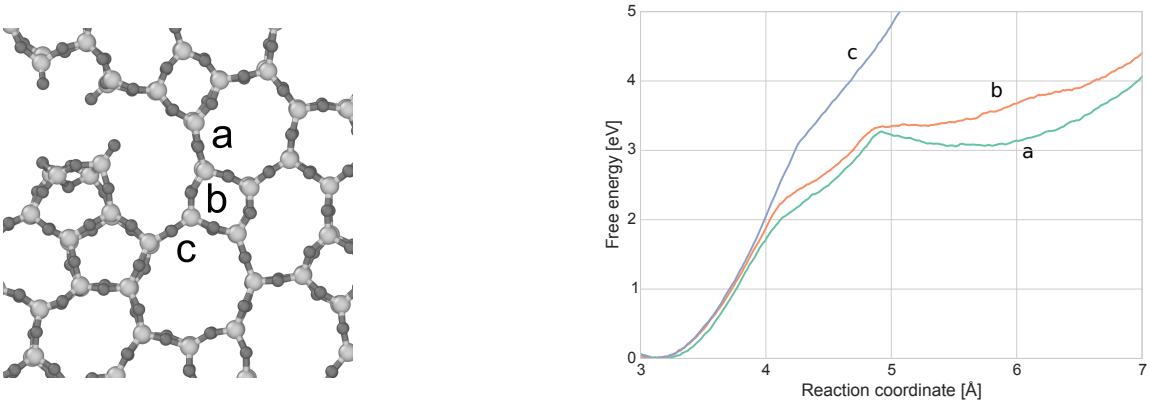


Fig. 5.36 7–4–8 amorphous crack tip and corresponding free energy barriers to crack advance. Barrier a height: 3.26 eV

5.8 Summary

In this chapter we presented the problem of fracture in oxide materials and learnt about two–dimensional silica, a newly synthesised material that in its simplicity can be the starting point to the understanding of the mechanisms involved in fracture propagation in this class of materials. 2D silica exists in both crystalline and amorphous forms and can be visualised atom by atom by experiments, solving the problem of inferring the atomic arrangement in glassy systems. We first overviewed some topological properties of an amorphous 2D silica network and confirmed that the amorphous material may be obtained by a cascade of bond switching mechanisms from the perfect honeycomb–like crystal. We then defined the set of computational experiments to perform on this material: the observation of dynamical crack propagations and the detailed calculation of free energy barriers to single crack advance events. To do so, we built a set of tools to, among other things, apply the desired stress energy release rate for a system of a fixed size and the definition of a thermodynamic integration setup suitable to the study of crack propagation. Thanks to the results of tests benchmarking the accuracy of classical potentials against 'ground truth' *ab initio* data, we discovered that the Tangney–Scandolo potential is sufficiently accurate to describe the properties of interest in fracture (elastic constants, surface energies, atomic forces at the crack tip), therefore we chose to treat our system in a purely classical framework for the rest of the chapter.

Dynamical crack propagation simulations showed a stark contrast between the behaviour of the crystalline and the amorphous material. Crystalline specimens appear to be excellent examples of a brittle material, showing a very prominent lattice trapping and velocity gap, where a crack either does not move or moves on a straight direction at extremely high speed and a steady pace while releasing large amounts of energy during each bond breaking event. On the contrary, the disordered nature of the amorphous material makes the crack direction change according to the local atomic structure, typically preferring to break large than small rings; perhaps unavoidably, a highly disordered crack tip configuration will halt the crack advance, so that some rings within the bulk of the material will be weaker than the crack tip, and any further mechanical load increase would result in a formation and coalescence of new crack fronts in different points inside the material.

To give a quantitative explanation of such results, we studied in detail the thermodynamic process of the double bond opening necessary to propagate a crack by one step. In the crystal, a very large free energy barrier to crack advance at the critical energy release rate –

and its persistence at rates more than twice such value – gives a rationale for explaining the extent of the observed lattice trapping. Because of its very structure, studying the amorphous material free energy barriers can only be partial. The computed barriers give a quantitative reading key to what is visualised during a fracture propagation: rings of size 4 or 5 are never the most energetically favourable to break, whereas a crack deviation from the main propagation direction can be the most favourable path if that involves breaking larger rings. From the limited results of the analysed crack tip configurations, we observe that the separation in free energy barrier height between different crack propagation mechanisms is large enough to suppose that the path to crack advance is determined by the ring structure.

Chapter 6

Machine Learning Applications in Atomistic Simulations

6.1 Titania Elastic Constants with SOAP–kernel GAP

Introduction

From a purely theoretical perspective, the recently proposed SOAP kernel [36] is a promising similarity measure for atomic configurations in kernel–based ML potentials thanks to its set of desirable properties. As with other ML–based interatomic potentials, though, a potential making use of it would inherently be short–ranged because it would rely on a representation of the local neighbourhood around an atom, which is truncated after the first few nearest neighbours. Simplistic solutions revolving around the idea of increasing the radial cutoff of the descriptor should be discarded: as the physical size of the represented system increases, so does the configurational space a descriptor must span, effectively making the definition of a similarity measure impossible. Long–range effects such as electrostatic interactions, especially important in oxide materials, cannot thus be directly included in these schemes but rather need to be treated with an additional model. To extend GAP potentials to oxides, the approach proposed consists of learning the force acting on an atom at the net of the one calculated by a classical interatomic potential that is able to appropriately treat the long–range interactions. This is equivalent to learning the conformation–dependent short–range correction to an IP that best fits a reference DFT dataset.

Parametrisations of the Tangney–Scandolo class of potentials outperform other IPs in silica structures where silicon atoms are 4–coordinated [106]; its standard parametrisation for titania, though, is not completely satisfactory as it was only fitted to DFT data for

the rutile polymorph [33]. If the source of errors lies in the short–range interaction, the modelling of titania can then be a case where a GAP potential may be key to creating a reliable interaction model that can overcome the limitations imposed by a fixed functional form IP.

Database

The database used for training consists of ∼1500 snapshots of first–principle MDs at 300 K, 500 K and 1000 K of both rutile and anatase phases – each snapshot contributing to a number of training set entries equal to its number of atoms. The breakdown of the content of the database can be inspected in Table 6.1. The database has been generated by Silvia Cereda at King's College London using the DFT package CASTEP [107] with ultrasoft pseudopotentials and the the LDA approximation to the exchange correlation functional, a planewave basis set with a cutoff energy of 1500 eV and a Monkhorst–Pack mesh with a maximum k–point spacing of 0.4 $Å^{-1}$. The choice of such a high cutoff energy was motivated by the sensitivity of the stress tensor to this parameter.

The teaching procedure is constrained by the available computational resources, partly in terms of raw computational power (to calculate and invert the full covariance matrix of a GP model) but mostly because of the vast memory requirements for the construction of the SOAP descriptors and sparsification procedure. Because of these reasons, in the version of the code used in this thesis the fitting procedure of a GAP model typically makes use of around 1000–2000 of the 10^4 available configurations chosen by a k–means based sparsification algorithm [8, 108].

Methods

The first question to be asked is whether such an approach can provide sufficient accuracy in some simple benchmarks. The first observables that we wish to reproduce are bulk properties such as the elastic constants, which define the linear elastic response of a material and are crucial in any MD simulation where the mechanical properties of the material are of key importance. In continuum mechanics, stress and strain field tensors can be expressed by the matrices $(\sigma_{ij})_{ij}, (\varepsilon_{ij})_{ij} \in \mathbb{R}^{3\times3}$ when projected on a Cartesian coordinate system [109]. When the elastic response is in a linear regime, the two can be related through a *stiffness tensor* $(C_{ijkl})_{ijkl}$ by the relation

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl}\,, \tag{6.1}$$

Table 6.1 Composition of the DFT database used for GAP training

Configuration type	N. Atoms	N. Configurations	N. Atomic Environments
Rutile			
Unit cell	6	719	4314
$2 \times 2 \times 2$ supercell	48	30	1440
$2 \times 2 \times 3$ supercell	72	17	1224
(110) surface	48	11	528
Anatase			
Unit cell	6	15	2400
$2 \times 2 \times 2$ supercell	96	14	1400
$2 \times 3 \times 1$ supercell	72	24	1728
$2 \times 2 \times 1$ supercell	48	21	1008
$3 \times 3 \times 1$ supercell	108	30	3240
(101) and (001) surfaces	60	11	660

here expressed in the Einstein notation of summing over repeated indices used throughout the current section. In a reversible process the strain energy density is equal to the work done per unit volume

$$\mathrm{d}u = \sigma_{ij}\mathrm{d}\varepsilon_{ij} = C_{ijkl}\varepsilon_{kl}\mathrm{d}\varepsilon_{ij} \,. \tag{6.2}$$

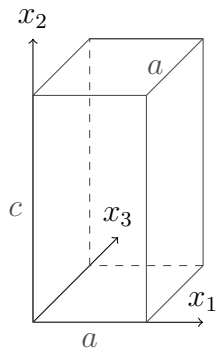
Manipulating Eq. 6.2, we obtain that the elastic constants are proportional to the second derivatives of the elastic energy density, known as *virial stress components*

$$C_{ijkl} = \frac{\partial^2 u}{\partial\varepsilon_{ij}\partial\varepsilon_{kl}} \,. \tag{6.3}$$

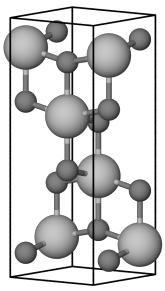
Symmetries of the stress and strain tensors imply that only 6 out of their 9 components are independent. The stress tensor $(\sigma_{ij})_{ij}$ can then be written as a vector $\boldsymbol{\sigma} \in \mathbb{R}^6$ composed of 3 tensile stresses along the orthogonal Cartesian directions and 3 shear stresses acting on the orthogonal planes defined by the same directions; likewise, the strain vector $\boldsymbol{\varepsilon} \in \mathbb{R}^6$ will be composed of 3 elongation strains and 3 shear strains. The relationship between these two vectors will then be defined by a stiffness matrix $\mathbf{C} = (C_{ij})_{ij} \in \mathbb{R}^{6\times6}$

$$\sigma_i = C_{ij}\varepsilon_j \,, \tag{6.4}$$

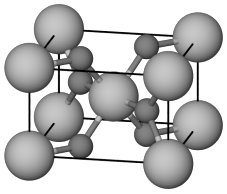
whose number of linearly independent entries depend on the lattice symmetries of the crystalline material under study. Both rutile and anatase have a tetragonal lattice as shown



(a) Tetragonal lattice



(b) Anatase



(c) Rutile

Fig. 6.1 (a) Tetragonal lattice defined by lattice constants a and c; (b,c) Unit cells of anatase and rutile phases of TiO_2 – large Ti atoms, small O atoms

in Fig. 6.1, which has 6 independent elastic constants:

$$\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{13} & 0 & 0 & 0 \\ C_{13} & C_{13} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{pmatrix} \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{pmatrix} . \tag{6.5}$$

In both structures the atoms are arranged as octahedra with one Ti atom at the centre and 6 O atoms at the corners, sharing 4 edges in anatase and 2 in rutile [110].

In order to calculate the elastic constants, one possible method is to apply a set of strain patterns to the equilibrium unit cell and measuring the corresponding energies. Recalling the formula for the elastic energy density (*i.e., elastic energy per unit volume*), the variation of energy density upon a deformation $\boldsymbol{\varepsilon}$ with respect to the equilibrium configuration is

$$\Delta u = \frac{1}{2} C_{ij} \varepsilon_i \varepsilon_j \, . \tag{6.6}$$

If, for instance, we now apply a strain pattern $\boldsymbol{\varepsilon} = (0, 0, \varepsilon, 0, 0, 0)$ for different values of ε, the curvature of the energy–strain parabola as shown in Fig. 6.2 (*i.e.*, the virial stress component) is proportional to an elastic constant:

$$C_{33} = \frac{2}{V} \frac{\partial^2 U}{\partial \varepsilon^2} \, . \tag{6.7}$$

When the strain pattern is composite, instead, the virial will be a linear combination of elastic constants: for example, a strain pattern $\boldsymbol{\varepsilon} = (\varepsilon, \varepsilon, 0, 0, 0, 0)$ results in a virial stress component

$$\frac{1}{V}\frac{\partial^2 u}{\partial \varepsilon^2} = \frac{1}{2}\left(C_{11} + C_{22}\right) + C_{12} \tag{6.8}$$

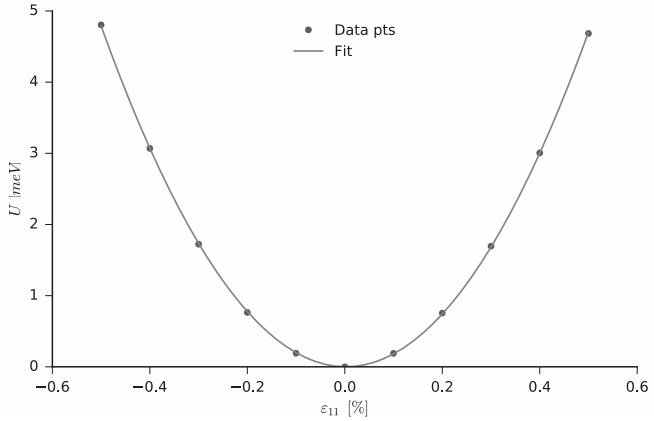


Fig. 6.2 Energy–strain curve for unidirectional elongation of anatase structure calculated with TS IP.

Results

We present in Table 6.3 some results achieved so far by different SOAP–kernel GAP models. The hyperparameters used in the two GAP potentials are presented in Table 6.2. The only hyperparameter that can be set based on physical considerations is the cutoff radius r_{cut}: we are interested in modelling the short–range interaction of the two $\mathrm{TiO_2}$ phases with a single potential, and since both phases are formed of octahedra whose arrangement determines the crystalline structure, its value must be set so to exactly include the first neighbouring octhahedron of an atom. With this choice, the two phases will be indistinguishable for the ML model, which will then be able to learn data from both phases interchangeably. The database labels are R = 1250 bulk rutile snapshots, A = 1000 anatase bulk, S = 200 surface rutile and anatase snapshots.

We must stress that experimental data or computational data from literature, which show quite a wide range of values, does not play any role at this stage of benchmarking. Each interaction model has to be compared solely with results from the model it has been

Table 6.2 GAP parameters of the tested models. The symbols are defined in Section 3.3

Model	$l_{\max}$	$n_{\max}$	σ_{blur} [Å]	ζ	r_{cut} [Å]	d [Å]	database
GAP^1	10	8	0.5	4	4.5	1.0	R
GAP^2	4	8	0.5	4	4.5	1.0	R+A+S
GAP^3	12	10	0.5	4	4.5	1.0	R+A

Table 6.3 Elastic constants comparison between different models. Units are GPa

	Method	B	C_{11}	C_{33}	C_{44}	C_{66}	C_{12}	C_{13}
Rutile	DFT	251	294	527	125	245	223	174
	TS	246	319	429	104	214	193	186
	GAP^1	258	250	560	117	220	235	179
	GAP^2	278	278	570	139	239	247	194
	GAP^3	257	253	488	107	237	282	202
Anatase	DFT	216	391	202	47	61	162	160
	TS	171	398	186	51	59	136	139
	GAP^1	221	369	233	56	51	168	139
	GAP^2	245	365	271	34	42	171	192
	GAP^3	239	377	225	64	67	169	135

fitted upon, here *ab initio* calculations with the same parameters as the DFT data used in the training set. Firstly, we can see that the plain TS potential [33], whose parameters have been fitted to reproduce bulk properties of both phases, already performs quite well for most elastic constants, especially in the case of rutile.

GAP^1, that has only been trained on rutile configurations, is able to enhance or at least not worsen the predictions of TS; perhaps surprisingly, errors in anatase are within acceptable limits despite the training set not containing, at least in principle, any specific information about such environments. Since both phases are made of the same octahedral building blocks, though, it is expected that GAP can model the short–range energy variation resulting from structural distortions of the octahedra in one phase given a database from the other phase. The very small torsional elastic constants C_{44} and C_{66} of anatase, instead, are not well reproduced.

In the case of GAP^2, instead, the descriptor is poorer but the model contains information about bulk and surface configurations of both phases. As one can expect, the accuracy in rutile does decrease slightly with respect to GAP^1; meanwhile, the additional training set does not result in any improvement of the elastic constants of anatase. Quite counterintuitively, then, using more data to fit the ML potential does not result in a better model.

This can be partly attributed to using fewer features to describe atomic environments, and partly to the fact that data from bulk and surface configurations may be harder to fit than in the previous case.

To test this hypothesis, we trained a GAP^3 model with bulk rutile and anatase data and an even higher basis set resolution of the descriptor but only including bulk configurations. We found that the potential does not offer tangible advantages with respect to either of the previous, often having an acceptable behaviour in between GAP^1 and GAP^2 when not overshooting the constants estimates in either direction. Having used a high dimensional descriptor and considering the interwoven nature of descriptor and ML hyperparameters of the SOAP method, the results may hint to a case of overfitting.

The models used to produce the presented results are representative of a much wider set, and the emerging trend appears to be that $l_{\max}$ and $n_{\max}$, the hyperparameters determining the basis set size of the descriptor, should be large enough to allow for an accurate description of an atom neighbourhood but not too large as to introduce irrelevant features that are damaging the ML model in use; the other hyperparameters, such as ζ and σ_{blur}, are disconnected from physical considerations and can either be kept fixed at values chosen by experience or optimised ad–hoc given the choice of the remaining hyperparameters.

Summary

GAP models can be used to fine tune the short–range part of a given interaction model. In oxide materials, where electrostatic interactions play a fundamental role, a GAP model may be able to improve the accuracy of an interatomic potential by adding a richer description of the short–range interaction while leaving the long–range part untouched. We tested GAP models built on top of TS IP for bulk $\mathrm{TiO_2}$ systems. The results show some accuracy improvements with respect to the already well–behaved TS potential, but as of now they appear to reach an accuracy threshold that prevents the achievement of a completely satisfactory matching of DFT results.

6.2 Direct Force Learning with SOAP

Introduction

When the only quantities of interest in an atomistic simulation are accurate forces, as in the case of several types of MD simulations, the construction of a full IP from which to derive them may be altogether unnecessary: an alternative approach is to directly learn forces. A force learning approach requires learning the mapping between a representation of an atomic neighbourhood around a 'central' atom and its corresponding force. Contrary to energy learning schemes where the target variable is a simple scalar, a force is a 3–dimensional vector that is covariant with respect to the input space vectors (*e.g.*, positions, forces, dipoles) under the operations of rotation and point and plane inversion. To visualise this statement, we can picture a force on an atom and the atomic positions around it to rotate rigidly when using the first atom as a pivot point. Since the input and output spaces are tied together by a set of symmetry operations, there are multiple viable representation strategies. These include:

- Instead of learning the target forces directly, build a ML model for their (scalar–valued) projections onto a set of k vectorial features of the input space, and reconstruct the real force in Cartesian space from its $k > 3$ projections in a least square approach. This method can be found in Ref. [39], whose development has been part of my studies, and a very similar method was simultaneously developed by an external research group in the work of Ref. [111].

- Represent both input and output space as 3–dimensional vectorial features and build a covariant ML model. According to Refs. [112–114], in a GPR approach this translates into building a covariance matrix whose entries are 3×3 matrices instead of scalar values. Given two atomic configurations ρ_i and ρ_j, the corresponding kernel is

$$\mathbf{K}(\rho_i, \rho_j) = \sum_{l=1}^{L} D_l(\rho_i, \rho_j) |\mathbf{v}_l(\rho_i)\rangle\langle\mathbf{v}_l(\rho_j)| \ : \tag{6.9}$$

  $\{|\mathbf{v}_l\rangle\}_{l=1,\cdots,L}$ is a set of any vectorial features calculated from a local atomic configuration that are invariant under permutation and covariant under rotation, whereas $\{D_l\}_{l=1,\cdots,L}$, instead, are L (possibly, but not necessarily) different real–valued and rotationally invariant covariance functions

- Use an input space representation that is invariant with respect to said symmetries and then align the corresponding force to some frame of reference, which may itself be inferred from the input space

The aim of this section is to explore the viability of the third option.

Method

The task is to find a way to rationally orientate the forces learnt from our dataset of configurations, and then use this rotation in the prediction step. Namely, the standard GPR prediction of Eq. 2.8 must be adapted to predict the force $\mathbf{y}_t$

$$\mathbf{y}_t = k(\mathbf{X}, \mathbf{x}_t)^T \, k(\mathbf{X}, \mathbf{X})^{-1} \, \mathbf{Y} \,, \tag{6.10}$$

where

$$\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N)^T \tag{6.11}$$

is a vector of teaching set forces, where each force $\mathbf{y}_i$ has been rotated according to the rotation that results in the maximal alignment $\tilde{\mathbf{R}}_i$ between the i^{th} configuration and the test configuration t:

$$\mathbf{y}_i = \tilde{\mathbf{R}}_i \mathbf{y}'_i \tag{6.12}$$

If we take two atomistic environments centred around a central atom, we can think of placing the two central atoms at the origin and rotate one of the two configurations using its central atom as the pivoting point. The 'maximum alignment' rotation is the rotation matrix that, when applied to the configuration free to rotate, maximises a similarity measure between the two environments: in the general framework of GAP–related descriptors, this is for example the overlap integral of the atom–centred Gaussians. Since knowledge about orientation is lost as soon as we define a rotationally invariant kernel, we must develop a method to consistently map the information of a force acting in an atomic environment to the new environment t.

Without a fixed reference frame, one can think of computing a weighted average of the force over the whole space of rigid rotations, where the weight of each rotation is measured by the corresponding SOAP overlap $S_{it}(\hat{R})$ defined in Eq. 3.36. This approach may work, for the resulting force aligned with respect to t mostly contains contributions from high overlap configurations; on the other hand, it may strongly suffer from symmetry related problems – *e.g.*, two different rotations producing a roughly equal but high overlap may cancel out their contribution to the total force.

The idea we are going to test is to evaluate the 'best alignment' in a SOAP–like fashion as

$$\mathbf{y}_i = \frac{\int d\hat{R} S_{it}(\hat{R}) \mathbf{R}(\hat{R})}{\int d\hat{R} S_{it}(\hat{R})} \mathbf{y}'_i \,, \tag{6.13}$$

where the denominator is a normalisation term. Following the SOAP derivation and notation, let us denote the test configuration as i'. We find that the rotation–dependent overlap can be expressed analytically

$$S_{ii'}(\hat{R}) = \sum_{lmm'} \left(D^l_{mm'} \sum_n c^{(i)*}_{nlm} c^{(i')}_{nlm'} \right) . \tag{6.14}$$

From Equation 6.13, the task is now to express the integral

$$\mathbf{J} = \int d\hat{R}\, S_{ii'}(\hat{R}) \mathbf{R}(\hat{R}) \,. \tag{6.15}$$

The Wigner D-matrices $\{D^1_{mm'}\}$ are a complete basis set for 3-D rotation matrices $\mathbf{R}(\hat{R})$, so no matter how finely we represent ρ in terms of l expansion, only the $l = 1$ terms will have a non-zero contribution in this integral. Making use of the orthogonality properties of the D-matrices

$$\int d\Omega\, D^1_{mm'}(\Omega) D^1_{\mu\mu'}(\Omega) = \frac{8}{3}\pi\, \delta_{m\mu}\delta_{m'\mu'} \,, \tag{6.16}$$

for each component of $\mathbf{J}$ we must evaluate a formula of the kind

$$\begin{aligned} J_{ab}(ii') &= \int d\hat{R}\, S_{ii'}(\hat{R})\, \mathbf{R}_{ab}(\{D^1_{mm'}(\hat{R})\}_{mm'}) \\ &= \int d\hat{R} \left(\sum_{lmm'} I^l_{mm'}(ii') D^l_{mm'}(\hat{R}) \right) \left(c_{ab} \sum_{\{\mu\mu'\}_{ab}} D^1_{\mu\mu'}(\hat{R}) \right) \\ &= c_{ab} \sum_{lmm'\mu\mu'} I^l_{mm'}(ii') \int d\hat{R}\, D^l_{mm'}(\hat{R}) D^1_{\mu\mu'}(\hat{R}) \\ &= \frac{8}{3}\pi^2 c_{ab} \sum_{\{\mu\mu'\}_{ab}} I^1_{\mu\mu'}(ii') \,, \end{aligned} \tag{6.17}$$

where the summation $\sum_{\{\mu\mu'\}_{ab}}$ can contain negative signs and it is performed over the correct indices for the component R_{ab}. By explicitly expressing Eq. 6.17 in terms of its

radial basis functions expansion, we get the following final expression:

$$J_{ab}(ii') = \frac{8}{3}\pi^2 c_{ab} \sum_n \sum_{\{\mu\mu'\}_{ab}} c^{(i)}_{n1\mu} c^{(i')}_{n1\mu'} \tag{6.18}$$

For example, R_{31} is written as:

$$\begin{aligned} J_{31}(ii') &= -\frac{8\pi^2 i}{3\sqrt{2}} \left(I^1_{-10} + I^1_{10}\right) \\ &= -\frac{8\pi^2 i}{3\sqrt{2}} \sum_n \left(c^{(i)\,*}_{n1-1} c^{(i')}_{n10} + c^{(i)\,*}_{n11} c^{(i')}_{n10}\right) \end{aligned} \tag{6.19}$$

Results

Firstly, from a purely theoretical point of view we must notice that a matrix defined by Eq. 6.15 does not necessarily represent a simple 'alignment', because the set of rotation matrices is not a group under the operation of summation. Arguably, though, in the limit of vanishing Gaussian blur of the atomic positions only one rotation would have a non–zero contribution to the integral, and the resulting matrix would actually represent a rotation. We thus tested this definition by using very small blurring (up to 3 orders of magnitude smaller than the ones used in the SOAP kernel itself), so to add as little 'spurious' contributions as possible into the integral **J**.

To test the proposed scheme, we calculated the matrix **J** for different configurations of bulk silicon atoms sampled from a MD at a constant temperature of 330 K, and we checked to what degree **J** satisfies the properties of a rotation matrix, namely:

- The norm of every column and row must be 1. Since the normalisation constant in Eq. 6.13 has not been explicitly derived, we can loosen this requirement and demand the same norm for every row and column.
- $J_{ab} = -J_{ba} \;\; \forall\, a \neq b$

These tests showed that the resulting matrices never resemble a rotation matrix. For a visually understandable example, Fig. 6.3 shows the effect of applying a normalised **J** matrix to the atomic positions of a silicon tetrahedron (quite similar configuration to the actual ones used to obtain it): the structure becomes squeezed in two dimensions, and distances are not preserved. Reducing the atomic blurring parameter any further is not an option either, as it rapidly results in reaching machine precision and overflow errors.

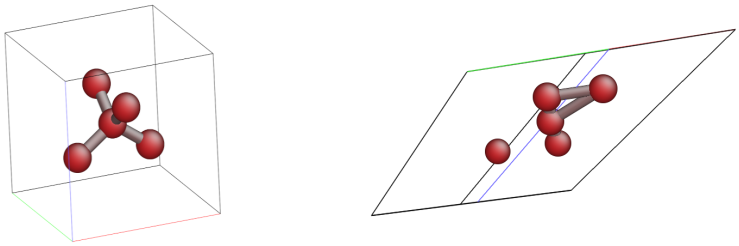


Fig. 6.3 Alignment matrix **J** applied on a tetrahedral arrangement of atoms. Left: before, right: after. This matrix has been obtained by computing the overlap between two equivalent configurations with the same orientation. Ideally, the resulting alignment would be the identity operator

Summary

In the present form, the proposed scheme is not a viable way to build a ML method to learn forces directly. A different way of reducing the contribution of low overlap configurations in the alignment would be to raise the power of S_{ij} in Eq. 6.15 in order to integrate a more 'peaked' function, but the analytic expressions of **J** for $S^2_{ii'}$ and $S^3_{ii'}$ degenerate into meaningless formulae; the evaluation of higher order expressions has not been attempted. The direct force learning problem should therefore be addressed with different solutions, such as the two other general principles laid out in the introduction.

6.3 Machine Learning–augmented Thermodynamic Integration

Introduction

Performing atomistic simulations to evaluate free energy barriers is a task that inherently requires long and repetitive calculations because any algorithm used to reconstruct a free energy surface must be able to sample the volume of phase space corresponding to a range of values of one or more collective variables describing the state of the system during some thermodynamic process. It only then seems natural that free energy calculations are a very promising ground for the application of machine learning techniques, with the aim of making the simulation faster or more accurate by using the computed information more efficiently.

The first steps in this direction have been taken by Stecher *et al.* [115], where a GPR model has been used for the on–the–fly reconstruction of a 2D free energy surface from data points obtained by umbrella sampling calculations. The GPR model has been used here to choose where to calculate the next data point to minimise the number of function evaluations and, concurrently, to estimate the error bars of the reconstructed free energy landscape, resulting in roughly a twofold computational efficiency gain.

The method proposed in this section, the machine learning–augmented thermodynamic integration (or MLTI), is aimed at enhancing the accuracy of free energy barriers calculated via thermodynamic integration to an intermediate level between a fully classical treatment and a prohibitive fully QM treatment by means of an ML model that makes the best possible use of the limited amount of QM–accurate information computed. We will apply the method to a physical problem encountered in the previous chapter, the SGTI calculation of free energy barriers to crack advance in two–dimensional silica.

Method

Let us have a system that undergoes a thermodynamic process whose state can be summarised by a single reaction coordinate (also known as collective variable), so that it can be studied by means of TI. Let us assume that there exists a computationally inexpensive MM model, such as a classical IP, that is sufficiently accurate to produce trustworthy trajectories in TI simulations, *i.e.*, where the explored phase space is similar to the one that would be spanned by a 'ground truth' fully QM simulation – by some criterion to be defined. Let us also assume that a fully QM TI simulation, or even simply a full QM/MM TI simulation

where the QM region comprises the neighbourhood of the atoms constrained by the TI algorithm, cannot be performed because it is too computationally demanding.

The key idea is then to use a classical SGTI trajectory and compute the QM–accurate constraint force on only a subset of its configurations: given an appropriate choice of input features that are able to describe an atomic environment and correlate to its constraint force, we can train a ML model to predict the constraint forces on all configurations in the trajectory. In principle, this postprocessing approach would make optimal use of the available computational resources to only perform QM calculations where it is most necessary, and instead use a 'ML interpolation' to fill out the missing information. When integrating the ML constraint forces on the trajectory configurations sampled from the phase space of the system treated by the classical model, the calculated free energy profile will have a hybrid accuracy whose upper bound is the QM one.

Limitations The method as stated above has one evident limitation: the more accurate the MM model is, the better it will span the phase space of the atomic system and the better the MLTI scheme will perform, but at the same time the MLTI method will gradually lose its own reason for existence because of the shrinking accuracy gain. Conversely, the opposite extreme of an inadequate choice of MM model will nullify the validity of the method, as the phase space will be incorrectly sampled. These considerations point to the specific use case of the MLTI method, namely when the MM method is accurate enough to produce reliable TI trajectories but not quite accurate enough to guarantee chemical accuracy (where a common definition of chemical accuracy is a RMS error of a database of MM forces of 0.05 eV $Å^{-1}$ with respect to the corresponding QM ones).

Furthermore, it is necessary to point out that a quantitative metrics to verify how well the MM model is able to provide a correct sampling of the 'correct' QM phase space of the system does not exist: if it did, it would require a database of configurations sampling the QM phase space, which at the same time would make it possible to compute a fully QM free energy profile by means of TI – exactly what has been assumed to be impossible and motivated the development of the method. Unless we remain in a purely benchmark setting, where an *ab initio* trajectory may be used as reference, we must therefore rely on some qualitative metrics of the MM model instead. Firstly, by a simple visual inspection of a MD trajectory there cannot be unphysical reconstructions or artefacts. Secondly, at a semi–quantitative level, there must be a satisfactory match of the physical quantities of interest for the thermodynamic process being studied, and a sufficiently small RMS

error on atomic forces so that the integrated system trajectories are not expected to differ significantly from the ideal QM ones.

Descriptor and Machine Learning model The main aspect determining the success or failure of a ML scheme is the creation of a set of features of the input space (also known as descriptor) that can be processed by the ML model to learn its correlation with respect to the corresponding output feature(s). If our rationale is to build an atomic neighbourhood descriptor that best correlates with the constraint force of a SGTI simulation, arguably the quantity that best correlates to a target force is another force that approximates it. Our choice of input features is therefore a set of constraint forces evaluated by an equal number of different TS IP, where we obtain different results from one class of potentials by tweaking the Morse–Stretch short–range parameters around their default values (see definition in Eq. 3.21); to curb the computational effort in feature evaluation, the force evaluation is performed on the same atomic clusters centred around the constrained atoms as the DFT calculations instead of the full slab of $\sim 10^4$ atoms. As an additional feature, the standard TS IP constraint force of the original MD simulation has been included, for it may provide some additional information and comes at no extra cost as it has already been evaluated. Such definition has several desirable features:

- As already mentioned, all input features have a high correlation with respect to the output feature
- Contrary to most other definitions of descriptor in solid state physics literature, it is not necessary to set a predefined and arbitrary cutoff radius beyond which atoms are considered to be irrelevant (a cutoff is still implicitly set by the IP, and for the short–ranged TS is about 10 Å [35])
- The number of input features does not depend on the number of atoms of the system
- The input features are as smooth and well–behaved as the unadulterated TS IP, since the variable parameter only influences the short–range interactions and not the electrostatics calculations.

The foreseeable downside of such feature selection is that they are expected to have a fairly high correlation not only with the output variable but also among themselves, which potentially restricts the range of choices of machine learning models and the amount of information carried.

The problem at hand is the regression of a single output variable (the scalar–valued QM–accurate constraint force of Eq. 5.10) given a set number of continuous input features (classical constraint forces), which is one of the most straightforward problems in machine learning. The target variable is related to the physical process of bond breaking: since for the atomic structure considered there is no smooth transition between the unbroken and broken states, we can expect the constraint force to present at least one discontinuity. It is therefore advisable to use a model that produces discontinuous outputs, such as a GP with a discontinuous kernel function, such as the absolute exponential kernel, or an inherently discontinuous model such as a regression tree.

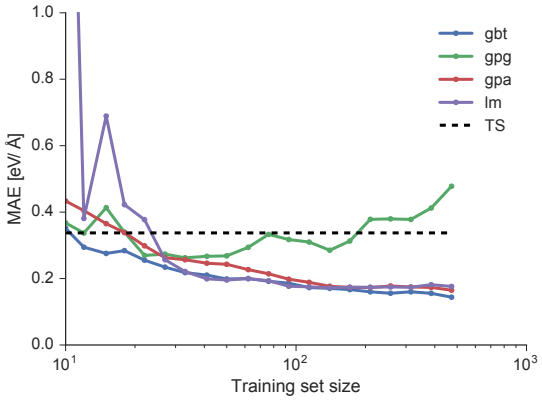


Fig. 6.4 Mean absolute error (MAE) on the constraint force vs. database size for different ML models, tested on the amorphous database. To each line corresponds a type of ML model: XGBoost: 'gbt', Gaussian process with Gaussian kernel: 'gpg', Gaussian process with absolute exponential kernel: 'gpa', least square linear regression: 'lm', TS potential (no model): 'TS'. Each data point is averaged over 10 trainings on randomly chosen data points.

These conjectures have been tested in a preliminary set of tests. A subset of these is shown in Fig. 6.4, where we can observe a few interesting facts:

- A GP with squared exponential kernel shows a rather clear indication that the kernel choice is inadequate for the task, as it can be worse than not having any ML model at all and the error metric does not decrease with the training set size
- A GP with absolute exponential kernel shows instead a correct learning behaviour, but the error does not reach the asymptote as fast as other models
- A simple ordinary least squares linear regression is surprisingly good at modelling the constraint force for training set sizes larger than about 20–30 samples, but is completely inappropriate below this threshold.

- A boosted regression trees model shows instead the best behaviour of all, by always beating the 'no model' baseline and reaching the error asymptote faster than all other ML approaches.

The model that best combines accuracy, speed and scalability is then a boosted regression trees model, of which I used the `XGBoost` implementation.

Dataset

The proof of concept tests have been carried out using a database containing QM–accurate forces computed for two system trajectories. Both have been obtained via classical SGTI MD simulations at 300 K to observe a single crack advance step. One trajectory is of the crystalline system as in Fig. 5.25 loaded at G = 1.01 G_c = 7.3 J m^{-2} (its free energy profile is the green curve of the plot in Fig. 5.27), containing 840 configurations sampled every 10 fs of the dynamics; the second trajectory corresponds to the crack advance in an amorphous system for an arrangement of the atoms at the crack tip as of Fig. 5.32 and loaded at G = 0.98 G_c = 7.6 J m^{-2} containing 592 configurations sampled every 10 fs. Fig. 6.5 shows some of the main features of these two datasets: for each trajectory we can identify two peaks corresponding to the two bond breaking events necessary to separate the Si–O–Si bridge at the crack tip, and we can see that the magnitude of the constraint force spans a range of more than 5 eV Å^{-1}.

The reasons for choosing a dataset from two trajectories of different systems are:

- firstly, we can rate the *learnability* of a property as a function of the system complexity. Since the amorphous structure is geometrically more complex than its crystalline counterpart, it is important to know if this also translates into a harder learning problem

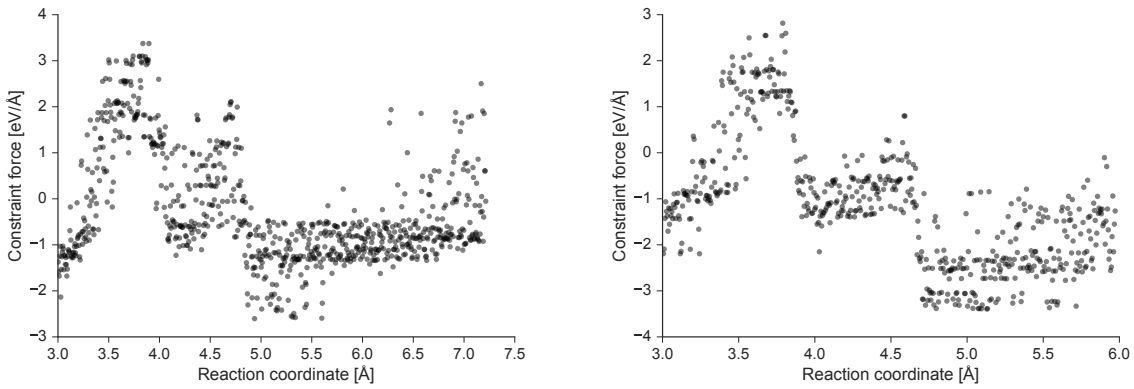
- secondly, we can obtain some insights about the *transferability* of the ML model from one system to another, *i.e.*, whether, and to what extent, the model fitted to the data of one structure can be used to predict the property of a different structure

- thirdly, and strictly linked to the previous point, to measure the *compatibility* of the data sources, so to understand if merging datasets from different structures and simulations is a viable strategy.

On all frames of these trajectories, the QM–accurate constraint forces have been calculated by means of DFT calculations carried out with the `CP2K` package; since the core QM

region is composed of the two Si atoms that are subject to the TI constraint, the force evaluations can be performed on clusters built around these two atoms using the same approach described earlier in Section 5.4 – Force evaluation, with a typical cluster size of about 680 atoms.

Since the QM forces are computed as a postprocessing step of the classical MD trajectory, all calculations are independent and can run at the highest resource efficiency in an embarrassingly parallel calculation: we used the 'farming' mode of CP2K, in which the main executable internally splits the computing resources into several partitions where concurrent instances of the DFT code are executed for different input files. Despite the achieved efficiency, me must point out that the force evaluation on the two trajectories made use of approximately 3 million CPUh: with an average cost per data point of roughly 2000 CPUh, the conceivable size of a dataset is in the 10^3 to 10^4 magnitude range, thus keeping the applications of machine learning methods to QM/MM calculations in oxide systems a few orders of magnitude short of the 'big data' realm.



(a) Crystal dataset

(b) Glass dataset

Fig. 6.5 DFT Constraint force vs. reaction coordinate for both trajectories of the dataset

Model parametrisation

The parameters that have to be set in our approach can be divided into two categories: those defining the features and those defining the ML model. To explore the accuracy across the vast parameter space, we measured the mean RMS error on the force in 3–fold cross validation trainings using the full dataset.

As already mentioned, the input features are constraint forces calculated by means of different TS potentials with a modified short–range interaction. With a fixed choice of the

ML model parameters to some standard values (learning rate = 0.01, maximum tree depth = 6, subsample = 0.6, number of learners = 300), we tried to build features by modifying D_{ij}, γ_{ij} and r_{ij}^0 and found that the best performing descriptors are obtained when rescaling the γ_{ij} parameters alone. We then obtained 16 features by setting an equal number of rescaling constants $c_i \in [0.5, 2]$ that act at the same time on all three parameters γ_{OO}, γ_{SiO} and γ_{SiSi} determining the short–range interaction strength, plus the original TS constraint force as the additional 17th feature. The choice of $\{c_i\}$ has not been thoroughly checked to be the one producing the highest cross-validation accuracy because the amount of information enclosed in such features is already sufficient to produce valuable results in the upcoming sections.
In most data science applications it is customary to preprocess the input features, so we additionally tested the effectiveness of (*i*) applying a standard scaler, so that each feature is rescaled to have zero mean and unit variance, (*ii*) applying a standard scaler followed by decomposition into principal components retaining all of the initial 17 features, or (*iii*) creation of quadratic features. None of these approaches, either on their own or combined, provided any improvement and have thus been left out of the pipeline for the feature evaluation.

Having settled on a definition of input features, we then proceeded to finding the best set of hyperparameters for the `XGBoost` model. In order to efficiently explore the parameter space, we used the Bayesian constrained optimisation method implemented in the `BayesianOptimization` package [116]. In this algorithm, the performance of a ML model is seen as a sample of a Gaussian process: using this overlay GP to model some accuracy metric as a function of the `XGBoost` model hyperparameters, we can use the GP to steer the parameter exploration towards the areas where the expected variance is the highest or to focus the sampling around the areas where the expected mean is the lowest, or more interestingly a combination of the two. As accuracy metric we used the average over 5 mean absolute errors on the test set of 5 trainings of the same model, where each training set comprised 10% of the whole dataset and the corresponding test set the remaining 90% – a rather unusual ratio that is in line with the aspirations of our resource–constrained application. The result of the exploration and optimisation process is the set of hyperparameters in Table 6.4.

When deploying our ML model either in a benchmark or in a real application, we must devise a method to select which subset of the N_{traj} available configurations obtained by the classical SGTI MD should be evaluated by the DFT calculator for a finite amount of computational resources. If we can only afford N_{eval} DFT calculations, the simplest

approach would be to simply evaluate one configuration every $N_{\text{interp}} = N_{\text{traj}}/N_{\text{eval}}$, but doing so does not necessarily maximise the amount of useful information computed. This problem can be recast in terms of finding structure within the N_{traj} data points so to identify N_{eval} maximally separated clusters, and to perform a DFT calculation on the most representative configuration of each cluster. To do so, we performed a k–means clustering on the normalised input features of the data set – the scaling being necessary to have a meaningful Euclidean distance in input space coordinates – and the configurations to be sent for QM calculations are defined as the centroids of each of the N_{eval} identified clusters. We must point out that this approach may be confused with *active learning*, but is currently something simpler because all of the training examples are chosen at the same time during the postprocessing of the classical trajectory: querying a QM calculator for an additional training data point at a time until some confidence measure is met would in principle be possible, but such a serial approach would be considerably slower than a batch QM evaluation.

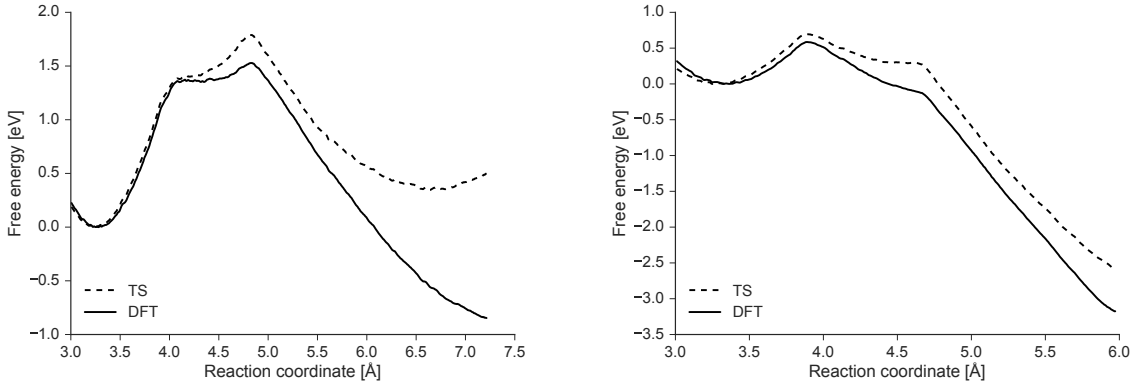
Table 6.4 XGBoost hyperparameters selected

Keyword	Value	Description
objective	reg:linear	Linear regression learning task in each leaf
eta	0.05	Learning rate (step size shrinkage of the boosting step)
n_estimators	500	Number of boosting iterations
max_depth	5	Maximum depth of a tree
subsample	0.6	Subsample ratio of training set
colsample_bytree	1	Subsample ratio of columns
min_child_weight	1	Minimum sum of instance weight needed to split a leaf
gamma	0.01	Minimum loss reduction needed to split a leaf
lambda	1	L2 regularisation term in the tree weights

Results

The computed datasets produce the two free energy profiles in Fig. 6.6, where the dashed lines represent the fully classical calculations and the solid lines are obtained by integrating the DFT constraint forces on the classical trajectory. In both cases, the TS IP is able to correctly guess the shape of the barrier but it overestimates its height.

The most intuitive metrics to evaluate the efficacy of the presented method is the mean absolute error (MAE) on the predicted constraint force when compared against the target DFT one: given a ML model M and a test set $\left((\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T, (y_1, y_2, \ldots, y_N)^T \right)$



(a) Crystal dataset

(b) Glass dataset

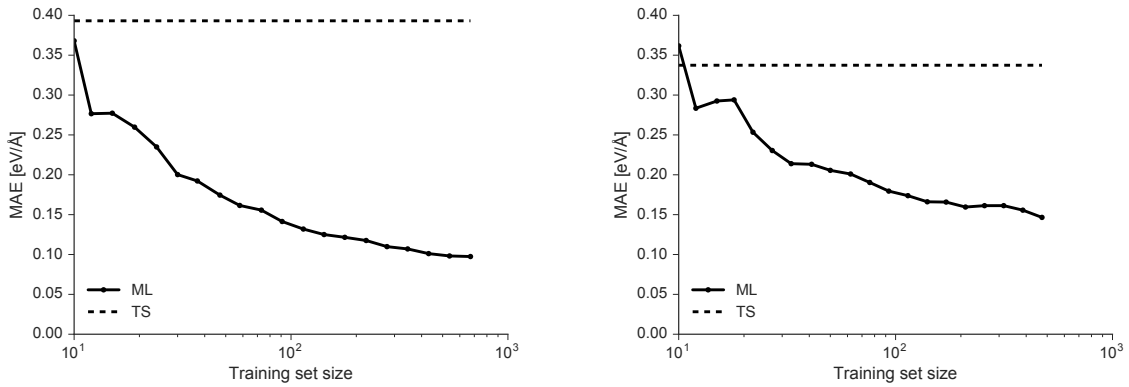
Fig. 6.6 Free energy profiles comparison between full TS IP modelling and hybrid QM constraint force / MM trajectory approach for the two datasets

containing ground truth DFT forces $\{y_i\}$, this is equal to

$$\mathrm{MAE}_y(M) = \frac{1}{N}\sum_{i=1}^{N} |M(\mathbf{x}_i) - y_i| \ . \tag{6.20}$$

To get a first grasp on the capabilities of the presented ML model, let us first compare the ML predictions and the TS force evaluations to the corresponding DFT values in both of the available datasets separately. According to the learning curves of Fig. 6.7, our ML model is able to beat the baseline TS potential already with very modest training sets containing a few tens of configurations, and the error monotonically decreases as more data points are added to the training set. The curves appear to have an asymptotic error of roughly 0.10 eV Å^{-1} in the crystal dataset and 0.15 eV Å^{-1} in the glass dataset, in line with the intuition that learning properties gets harder as the atomic structure becomes less ordered. The emergence of these hard limits is a manifestation of the incompleteness of a system representation by the defined input features.

By instead considering how the predicted constraint forces are used, the most relevant way in which a ML model should be evaluated is by comparing how closely the profile of the free energy barriers gets to the target DFT one, both in terms of shape and overall height. The free energy of a system at a value of reaction coordinate s_i is, up to an additive constant, the sum (approximation of the cumulative integral) of the constraint forces along



(a) Crystal dataset

(b) Glass dataset

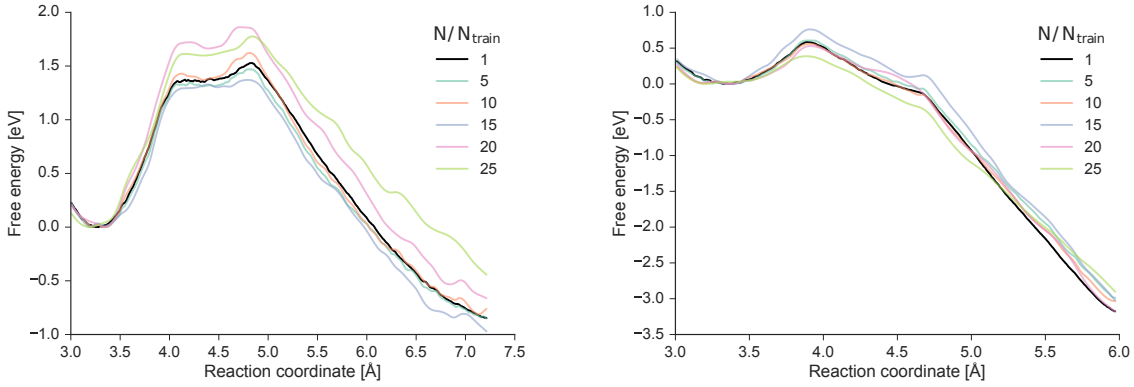
Fig. 6.7 Mean absolute error on the predicted constraint force vs. size of training set. Here a model trained with a training set of crystal configurations is only used to predict forces on crystal configurations, and vice versa for the amorphous configurations. Given a dataset of size N and a training set size $N_{\text{train}} \in [10, 0.8\,N]$, the force predictions have been performed on all other $N - N_{\text{train}}$ configurations. The baseline MAE of the TS potential is shown with a dashed line

the SGTI trajectory up to the i^{th} element

$$F_i = \sum_{j=1}^{i} y_j \Delta s_j \,, \tag{6.21}$$

and for a meaningful comparison we chose to align every curve by setting the free energy at its minimum to zero before the first bond opening.

To understand whether a ML approach is really of any use at all, we must think of what would be the best way to use a given number of QM calculations N_{train} without using any method requiring an advanced treatment of information: arguably, this would be to perform the DFT calculations on equally time–spaced configurations, interpolate the so obtained constraint forces within each interval and integrate these over the reaction coordinate – clearly, in this setting it would be inconceivable to reuse any QM constraint force computed for different trajectories. Let us choose a linear interpolation, as there is no rationale to justifying a more complex interpolation (*e.g.*, cubic). The effectiveness of such interpolation method in turn depends on how sensitive the thermodynamic integration is to the sampling rate of a SGTI trajectory, *i.e.*, how quickly the free energy profile calculated using Eq. 6.21 gets corrupted as the integration is performed over progressively longer reaction coordinate intervals $\{\Delta s_j\}$. As we can see from the profiles of Fig. 6.8, evaluating



(a) Crystal dataset

(b) Glass dataset

Fig. 6.8 DFT free energy profiles vs. linear interpolation interval. The DFT constraint force is calculated every N/N_{train} interpolation steps and is linearly interpolated in between

one DFT configuration every 5 does not result in any detectable loss in accuracy, whereas an interpolation step of 25 will certainly produce unreliable profiles; for interpolation steps in between the two extremes, the quality of the free energy profile is not very predictable as it heavily depends on which are the QM–calculated configurations. Let us now explore the performance of the proposed ML method.

The first tests involve the learnability of the property, and is restricted to learning data from one SGTI trajectory to predict the full free energy profile of the same trajectory: in this framework, one can think of the ML model as a 'smart' interpolator between training points. The results presented in Fig. 6.9 and Fig. 6.10 show the outcomes of the ML approach as a function of the decreasing computational resources available, starting from $N/N_{\text{train}} = 10$ (59 training points for the amorphous, 84 for the crystal) to an extreme of $N/N_{\text{train}} = 50$ (11 training points for the amorphous, 16 for the crystal). For both structures, the ML–reconstructed free energy profile is, for any practical application, indistinguishable from the target DFT one up to at least $N/N_{\text{train}} = 20$, and begins showing a small but noticeable discrepancy above $N/N_{\text{train}} = 30$. For $N/N_{\text{train}} = 50$, corresponding to performing just a handful of DFT calculations, the ML model is no longer accurate enough for the crystal but, surprisingly, remains usable in the amorphous system, where the barrier height and shape are still very accurately reproduced. We can then confirm that the ML model, together with the specific definition of descriptor, are able to learn the constraint force in a SGTI simulation, and can maintain the desired accuracy for a computational cost N_{train} that is at least 3 times smaller than an optimal non–ML approach.

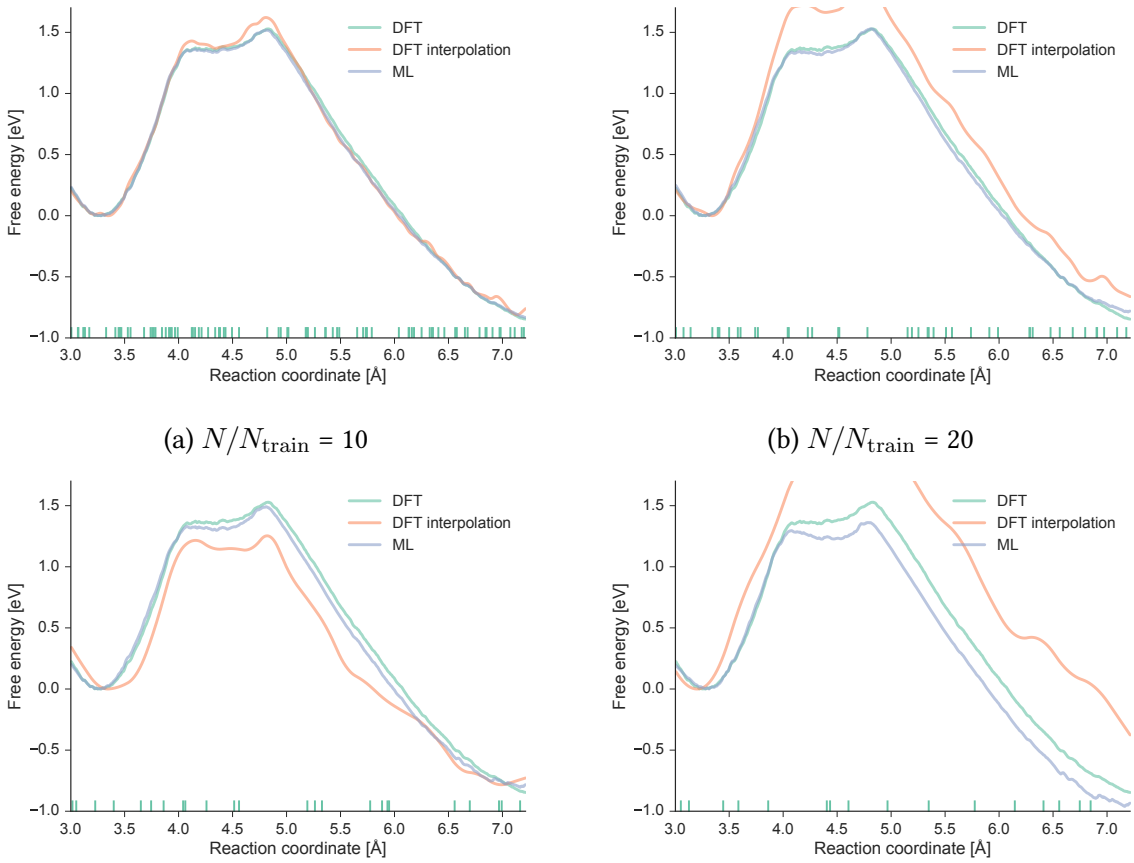
The second test consists of verifying the compatibility between data sources: by merging the databases of the two trajectories into one inhomogeneous database, we now wish to perform a single training of the ML model and predict both free energy profiles. The results in Fig. 6.11 are presented as a function of N/N_{train}: for a given value of N/N_{train}, the difference between the profiles in Fig. 6.9 and 6.10 and the ones in Fig. 6.11 is that on top of the same number of training points referred to the relevant trajectory there is a similar number of training points of the other trajectory. We can imagine three possible outcomes: *(i)* the ML model becomes 'confused' and the predictions become less accurate because the input features are not able to effectively distinguish the two structures, *(ii)* the predictions remain the same because the input features manage to separate the knowledge of the two structures, or *(iii)* the predictions improve because the ML model can make use of the additional information coming from a separate data source.

Perhaps understandably, the results fall in the first category but the profiles are still reproduced quite accurately. In particular, the predictions on crystalline configurations seem to suffer more from non–relevant training set data than their counterparts for the amorphous structure, whose free energy profile remains close to the target one even for a small number of data points. In summary, at best the predictions do not get worse: at least in the limit of small datasets of at most a hundred configurations of our tests, with the current choice of descriptor and of hyperparameters of the ML model it is not advisable to predict a free energy profile by also reusing QM data previously computed on different simulations.

The most stringent and final test is the transferability test: a ML model is trained using data points of one trajectory and used to predict the free energy profile corresponding to the other trajectory. This would be the ultimate goal of an all–encompassing ML model that has managed to learn the relationship between input features and output constraint force in all of the phase space that can be sampled by a thermodynamic integration of the given process and material. With the very small and certainly not complete database available, and considering the compatibility test results, this test is bound to fail, but it can be nevertheless very informative to observe how our ML models behave in such situation. The two free energy profiles of Fig. 6.12 help us better understand what the ML model does and does not learn. The shapes of the curves are guessed correctly, but their scaling is wrong and they tend to shift towards the wrong target curve: this behaviour indicates that the ML model is indeed able to learn the sign and intensity of the output variable, but only in relation to its value in some neighbouring configurations of the training set. We

can thus argue that the observed discrepancy is due to the different distributions of target values in the two datasets as of Fig. 6.5.

In conclusion, the proposed method is very effective at evaluating DFT free energy barriers with minimal computational resources, although by always bearing in mind that the classical system trajectory must represent a valid sampling of the phase space. The current tests do not show that QM data of one or more simulations can be used in a new TI simulation to improve, if not altogether define, the force predictions used to integrate its free energy profile: this, though, may well be an apparent limitation arising from the small dataset used and from the simplistic (yet justified) choice of descriptor.



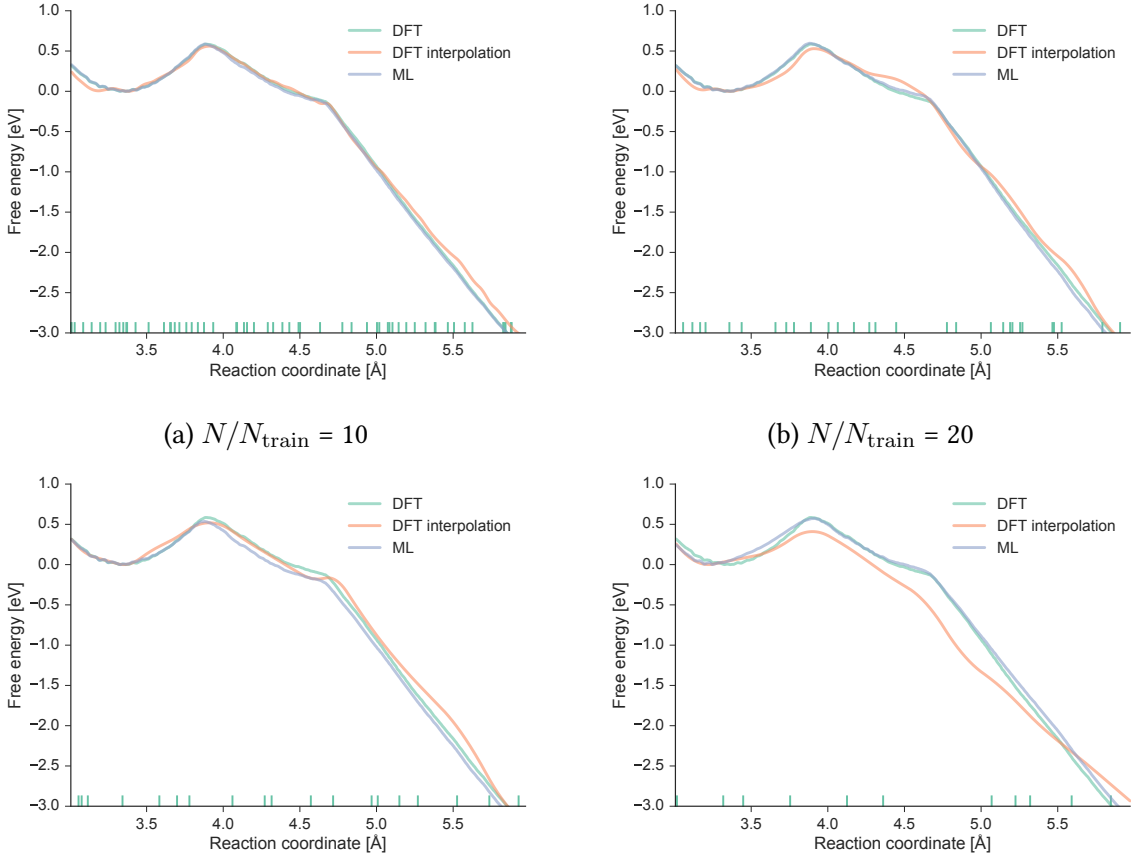
(a) $N/N_{\text{train}} = 10$

(b) $N/N_{\text{train}} = 20$

(c) $N/N_{\text{train}} = 30$

(d) $N/N_{\text{train}} = 50$

Fig. 6.9 Free energy profiles vs. training set size for the crystalline system. The DFT constraint force is learnt every N/N_{train} SGTI steps, where the total dataset contains $N = 840$ configurations. The training set points are indicated by ticks on the horizontal axis. The target solution represented by the DFT curve (teal), the linear interpolation of DFT constraint forces (best non–ML guess for a given number N/N_{train}) (orange), and the proposed ML model prediction (blue) are shown together



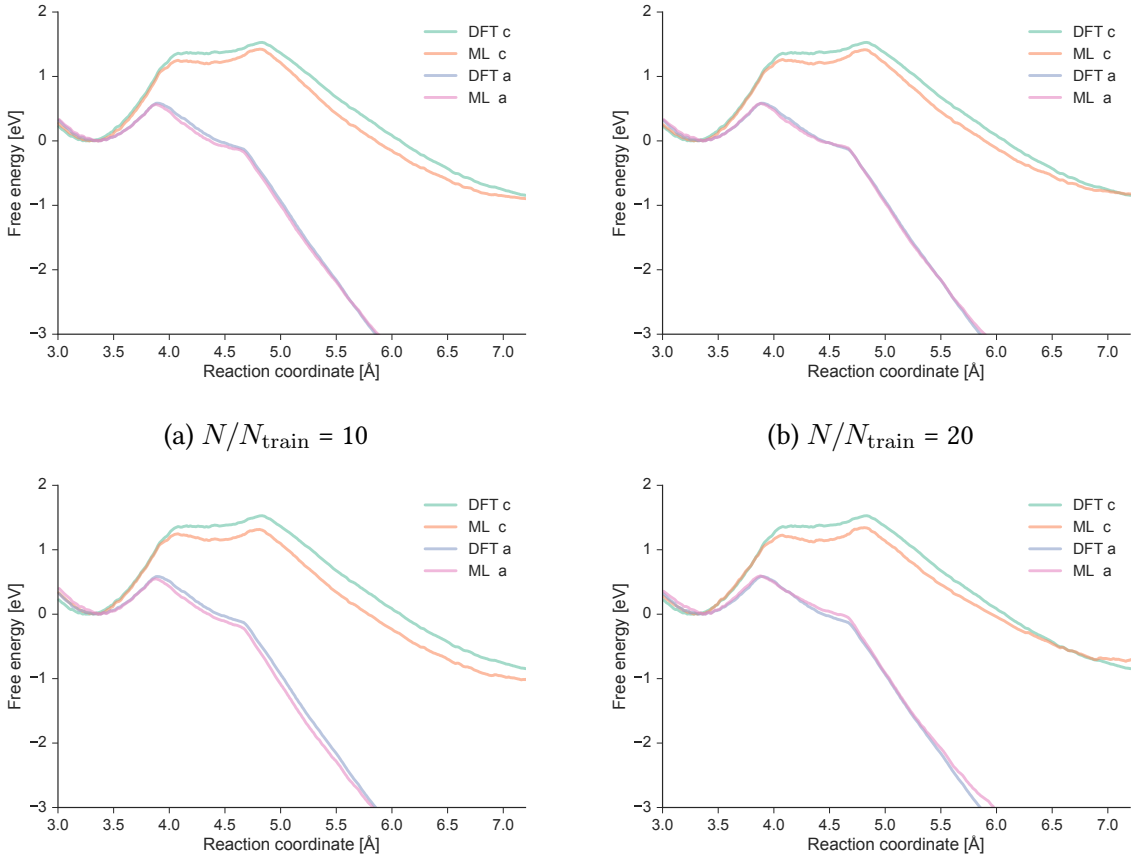
(a) $N/N_{\text{train}} = 10$

(b) $N/N_{\text{train}} = 20$

(c) $N/N_{\text{train}} = 30$

(d) $N/N_{\text{train}} = 50$

Fig. 6.10 Free energy profiles vs. training set size for the amorphous system. The DFT constraint force is learnt every N/N_{train} SGTI steps, where the total dataset contains $N = 592$ configurations. The training set points are indicated by ticks on the horizontal axis. The target solution represented by the DFT curve (teal), the linear interpolation of DFT constraint forces (best non–ML guess for a given number N/N_{train}) (orange), and the proposed ML model prediction (blue) are shown together



(a) $N/N_{\text{train}} = 10$

(b) $N/N_{\text{train}} = 20$

(c) $N/N_{\text{train}} = 30$

(d) $N/N_{\text{train}} = 50$

Fig. 6.11 Free energy profiles vs. training set size using a mixed training set. The DFT constraint force is learnt every N/N_{train} SGTI steps from the merged dataset of size 1432 containing both crystalline and amorphous configurations. The target solution is labelled as DFT and the proposed ML model as ML; 'a' and 'c' stand for amorphous and crystalline respectively

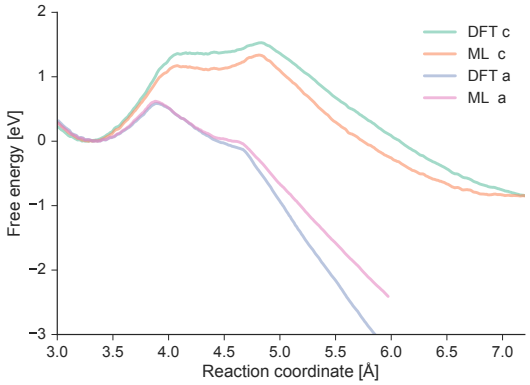


Fig. 6.12 ML Free energy profiles of crystalline and amorphous systems trajectories using a ML model trained on the data of the other trajectory. Here all data points have been used for training ($N = N_{\text{train}}$) to best approximate the ideal use case condition – *i.e.*, a large database of different trajectories

Summary

When a chemical reaction is studied in the (slow growth) thermodynamic integration approach, one constructs its free energy profile by integrating the scalar valued constraint force that is required to progressively drag a single reaction coordinate from a value corresponding to the initial state to that corresponding to a final state. By using an inexpensive IP that is able to generate the system trajectory to a certain degree of trustworthiness and so appropriately sampling the phase space, one could at a later stage compute the QM–accurate constraint forces on that trajectory to improve the accuracy of the free energy profile. Since TI simulations involve long MD runs producing repetitive system trajectories, we observe that the QM constraint force is a quantity that can be learnt by a ML model, so that the computational cost of improving free energy profiles with *ab initio* data would ideally be reduced to a bare minimum. At the basis of the approach is the definition of a descriptor of an atomic environment, which we defined as a set of constraint forces evaluated by an equal number of interatomic potentials. The proposed method has been applied to the study of free energy profiles calculations for the crack advance in 2D SiO_2. The results shown prove that it is possible to obtain the same free energy curves whether the *ab initio* forces are computed on every frame of the trajectory or if they are computed in just a few tens of configurations and interpolated elsewhere by a ML model.

6.4 Summary

The pivot idea of this chapter is to introduce a new level of accuracy in between classical potentials and *ab initio* calculations in the form of machine learning models. These can, in theory, make the best use of computationally expensive QM–accurate physical data by radically changing how such data is used: rather than imposing a rigid functional form to the atomic interaction model, the power of a ML model is to let the data itself determine it. This research field is only a few years old, and out of the endless possible paths to explore we went in depth into analysing three attempts of this kind: *(i)* a GAP model to improve the accuracy of an interatomic potential by correcting the short–range interaction in TiO_2, *(ii)* a definition of descriptor for a direct force learning approach, and *(iii)* a method to bring free energy barriers calculations towards a quantum accuracy. Path *(i)* does indeed show some positive results, but it is questionable whether the additional effort is worth the marginal improvement in accuracy; instead, we came to the valuable conclusion that path *(ii)* is a dead end and should not be pursued any further. Path *(iii)*, the machine learning–augmented thermodynamic integration framework (MLTI), is instead one application in which machine learning can become a valuable tool to not only speed up, but to altogether make possible, the evaluation of free energy barriers at an accuracy that, in the best conditions, may approach the DFT one.

Chapter 7

Conclusions

In this thesis I explored different routes at the vanguard of multiscale simulations, with the common final aim of making the quantum mechanically–accurate study of fracture – and more widely chemomechanics – in oxide materials possible. Multiscale fracture propagation simulations and stress corrosion simulations only in the recent years became routine for relatively simple systems such as crystalline silicon: these achievements are due to a combination of advances in the computational tools and in reaching the availability of a sufficient computing power. Repeating the same kind of studies in oxide materials is more computationally demanding, and the study of amorphous structures adds one further layer of complexity to the problem. The answers to questions such as 'how does silica glass break' thus lie beyond the development of new computational tools.

The first issue is computational cost: electrostatics plays a fundamental role to determine the atomic interactions in oxides, therefore *ab initio* calculations necessary to evaluate QM–accurate forces on the atoms at the crack tip require the inclusion of several hundreds of neighbouring atoms. Using the most recent supercomputing architectures is thus necessary but not sufficient: using them effectively requires profound changes in the software deployed. With this in mind, a novel ensemble parallel framework has been proposed in Chapter 4, in which the QM calculation workload is split across different executables. The efficiency of the method depends on the load balance across QM calculators, which in turn is related to the size balance of the atomic clusters to be evaluated. Following this consideration, I created a new partitioning algorithm and showed that it outperforms the load balance otherwise obtained by standard methods considered to be optimal for the task. The ensemble parallel scheme making use of this partitioning method showed weak linear scaling and can effectively be used on 10^5 cores.

When simulating an amorphous material, another issue is to define its atomic structure. Recent experiments produced two dimensional silica structures, in which the crystalline and amorphous phases coexist and can be imaged at atomic resolution. Since these structures are the simplest existing examples of polymorphic silica, they represent an ideal candidate for an initial study of fracture propagation in silicate materials. Having shown that a reasonable accuracy can be achieved using the TS potential only, I created the computational setup and performed a set of calculations in both crystalline and amorphous systems to understand *(i)* the dynamics of a propagating crack, using MD simulations, and *(ii)* the reaction kinetic details of bond breaking events when a crack advances, by means of the thermodynamic integration method.

The results of part *(i)* show that the crystalline material has an exceptionally large lattice trapping and velocity gap, as a crack does not advance until the elastic load is beyond more than twice the critical energy release rate, and when it propagates it does so at a speed exceeding 2000 m s^{-1}. The MD results for the amorphous material, instead, draw a completely different picture: a moving crack follows a well defined path determined by the random ring structure of the glass, and it appears to halt at some unfavourable crack tip arrangements; these simulations hint that fracture propagation at a macroscopic scale is the result of the creation of voids in weak regions of the material followed by their coalescence.

Step *(ii)* helps interpreting these results by observing the free energy profile of each crack advance event. In the crystalline system, the lattice trapping and velocity gap observed can be explained by the presence of very energetically unfavored intermediate states that produce a free energy barrier of almost 1.8 eV that only vanishes at around 2.2 G_c. In amorphous systems, instead, these simulations validate the hypothesis that a crack is favoured to advance in preferential directions determined by the silica ring structure near the crack tip, favouring the rupture of large rings and finding insurmountable hurdles for some very disordered crack tip configurations.

The investigation carried out in Chapter 5 is inevitably incomplete, and many avenues of further research can branch out from the current results: more simulations are needed to explore the stress and temperature dependence of the free energy barriers; a new set of simulations is needed to explore the existence of a velocity gap in crystalline 2D–SiO_2 in the narrow energy release rate range where slow crack advance may be observed (17.2 J m^{-2} to 18.5 J m^{-2} at 300 K); an advisable step would be to validate the results with higher accuracy QM/MM simulations, which would also make possible the study of stress corrosion, *e.g.*, the effects of dissociation of water molecules at the crack tip in a setup similar to Ref. [117]; as the research on the simplified 2D–SiO_2 systems progresses, the long term goal is to use

the so obtained tools and insights as the starting point for understanding the physics of more technologically relevant 3D materials as soon as the computational methods and the available computing power will allow that.

With an ideal goal of building an interaction model as accurate as DFT and as computationally inexpensive as an IP, in Chapter 6 we explored three entirely different routes in using ML for atomistic simulations of oxides.

We laid out the first steps in the development of a GAP potential for the polymorphs of TiO_2 to overcome the limitations of the TS potential parametrisation for this material. The results, involving the calculation of elastic constants in anatase and rutile bulk crystals, are encouraging but the improvements from the TS IP baseline are not consistent at the current stage. The natural evolution of the research on this topic is to investigate the source(s) of errors of a GAP model, which begins with understanding whether the errors in the original TS potential arise from short– or long–range interactions: only in the former case a short–range GAP model can be justified. Current results from force locality tests in bulk anatase similar to those in Ref. [69] show that an ideal GAP model – *i.e.*, one that would perfectly model the atomic interactions for a range up to its cutoff radius – would be able to reach chemical accuracy (0.05 eV $Å^{-1}$) for a cutoff of at least 6 Å: more tests are needed to understand how this value changes in surface calculations and in the rutile structure.

In Section 6.2 we hypothesised a strategy for the direct learning of an atomic force given a local environment, and the investigation of its mathematical implications produced a negative result. The approach of learning 3D force vectors would be an elegant and general solution to generate force fields without the explicit definition of a potential, and this result can help to better focus the efforts of further developments.

All of the topics covered in this work come together in the final Section 6.3, where I propose a general method for the ML reconstruction of one–dimensional free energy profiles of chemical reactions. The key concept is that a classical TI trajectory is postprocessed by a ML method that learns QM–accurate data on a subset of its configurations. The method has successfully been applied to the study of crack advance of a single atomic step in 2D-SiO_2. Here, the results show that the devised ML model allows to evaluate by DFT – and learn – only a very small number of trajectory frames (15 to 40) while still reproducing the same free energy profiles that would be obtained by performing a DFT calculation on every frame of the trajectory. The next natural steps in this direction are:

- perform a thorough search to find the descriptor most suitable for the problem (*e.g.*, by using LASSO techniques as in Ref. [118])

- extend the database of DFT configurations to include TI trajectories of crack advance for different crack tip arrangements and loads, to test the feasibility of building a single ML model for the 2D SiO_2 material as a whole that can predict QM–accurate TI constraint forces in all crack tip arrangements.

Additionally, the proposed method is not tied to one particular material or chemical reaction and can be applied to the study of any system whose state can be described by a single collective variable. The proposed approach is instrumental in bringing free energy calculations of large atomic systems from the accuracy currently achievable by classical potentials towards the accuracy of higher *ab initio* methods.

References

[1] J. H. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[2] M. N. Gibbs. *Bayesian Gaussian processes for regression and classification*. PhD thesis, University of Cambridge, 1997.

[3] D. JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[4] C. E. Rasmussen and C. KI Williams. *Gaussian processes for machine learning*. The MIT press, 2006.

[5] G. S. Kimeldorf and G. Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41(2):495–502, 1970.

[6] B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[7] Andreĭ Nikolaevich Tikhonov and Vasiliĭ Arsenin. *Solutions of ill-posed problems*. Winston, V. H., 1977.

[8] A. Bartók-Pártay. *The Gaussian Approximation Potential: an interatomic potential derived from first principles quantum mechanics*. PhD thesis, University of Cambridge, 2010.

[9] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[10] Xgboost. https://github.com/dmlc/xgboost. Accessed: Wednesday 3rd May, 2017.

[11] R. E. Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

[12] Y. Freund. Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285, 1995.

[13] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[14] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. Classification and regression trees. *Wadsworth International Group*, 93(99):101, 1984.

[15] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. *arXiv preprint arXiv:1603.02754*, 2016.

[16] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

[17] C Adam-Bourdarios, G Cowan, C Germain-Renaud, I Guyon, B Kégl, and D Rousseau. The higgs machine learning challenge. In *Journal of Physics: Conference Series*, number 7. IOP Publishing, 2015.

[18] N Mirabal, E Charles, EC Ferrara, PL Gonthier, AK Harding, Miguel A Sanchez-Conde, and DJ Thompson. 3fgl demographics outside the galactic plane using supervised machine learning: pulsar and dark matter subhalo interpretations. *The Astrophysical Journal*, 825(1):69, 2016.

[19] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob.*, volume 1, pages 281–297. Univ. of Calif. Press, 1967.

[20] S. P. Lloyd. Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[21] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[22] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973.

[23] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[24] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(4):619–633, 1975.

[25] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[26] M. Born and R. Oppenheimer. Zur Quantentheorie der Molekeln. *Annalen der Physik*, 389(20):457–484, 1927.

[27] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, Nov 1964.

[28] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133–A1138, Nov 1965.

[29] J. P. Perdew, K. Burke, and M. Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, 77:3865–3868, Oct 1996.

[30] R. P. Feynman. Forces in molecules. *Phys. Rev.*, 56:340–343, Aug 1939.

[31] P. Tangney and S. Scandolo. An ab initio parametrized interatomic force field for silica. *J. Chem. Phys.*, 117(19), 2002.

[32] P. Tangney and S. Scandolo. A many-body interatomic potential for ionic systems: Application to mgo. *J. Chem. Phys.*, 119(18):9673–9685, 2003.

[33] X. J. Han, L. Bergqvist, P. H. Dederichs, H. Müller-Krumbhaar, J. K. Christie, S. Scandolo, and P. Tangney. Polarizable interatomic force field for TiO_2 parametrized using density functional theory. *Phys. Rev. B*, 81(13):134108, 2010.

[34] A. J. Rowley, P. Jemmer, M. Wilson, and P. A. Madden. Evaluation of the many-body contributions to the interionic interactions in MgO. *J. Chem. Phys.*, 108(24):10209–10219, 1998.

[35] J. R. Kermode, S. Cereda, P. Tangney, and A. De Vita. A first principles based polarizable O(N) interatomic force field for bulk silica. *J. Chem. Phys.*, 133(9):094102, 2010.

[36] A. Bartók-Pártay, R. Kondor, and G. Csányi. On representing chemical environments. *Phys. Rev. B*, 87:184115, May 2013.

[37] A. Bartók-Pártay, M. C. Payne, R. Kondor, and G. Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.*, 104:136403, Apr 2010.

[38] J. Behler and M. Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.*, 98:146401, Apr 2007.

[39] Z. Li, J. R. Kermode, and A. De Vita. Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces. *Phys. Rev. Lett.*, 114:096405, Mar 2015.

[40] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.*, 108:058301, Jan 2012.

[41] A. Bartók-Pártay and G. Csányi. Gaussian approximation potentials: A brief tutorial introduction. *Int. J. Quantum Chem.*, 115(16):1051–1057, 2015.

[42] L. Verlet. Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Phys. Rev.*, 159:98–103, Jul 1967.

[43] S. Nosé. A unified formulation of the constant temperature molecular dynamics methods. *J. Chem. Phys.*, 81(1):511–519, 1984.

[44] W. G. Hoover. Canonical dynamics: equilibrium phase-space distributions. *Phys. Rev. A*, 31(3):1695, 1985.

[45] G. J Martyna, M. L Klein, and M. Tuckerman. Nosé–Hoover chains: the canonical ensemble via continuous dynamics. *J. Chem. Phys.*, 97(4):2635–2643, 1992.

[46] T. Schneider and E. Stoll. Molecular-dynamics study of a three-dimensional one-component model for distortive phase transitions. *Phys. Rev. B*, 17:1302–1322, Feb 1978.

[47] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch. Structural relaxation made simple. *Phys. Rev. Lett.*, 97:170201, Oct 2006.

[48] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[49] D. Packwood, J. R. Kermode, L. Mones, N. Bernstein, J. Woolley, N. Gould, C. Ortner, and G. Csányi. A universal preconditioner for simulating condensed phase materials. *J. Chem. Phys.*, 144(16):164109, apr 2016.

[50] G. Mills and H. Jónsson. Quantum and thermal effects in H_2 dissociative adsorption: Evaluation of free energy barriers in multidimensional quantum systems. *Phys. Rev. Lett.*, 72:1124–1127, Feb 1994.

[51] K. Binder and D. Heermann. *Monte Carlo simulation in statistical physics: an introduction*. Springer Science & Business Media, 2010.

[52] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Phys.Lett. B*, 195(2):216–222, 1987.

[53] P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler. Transition path sampling: Throwing ropes over rough mountain passes, in the dark. *Annual review of physical chemistry*, 53(1):291–318, 2002.

[54] J. Schlitter, W. Swegat, and T. Mülders. Distance-type reaction coordinates for modelling activated processes. *Molecular modeling annual*, 7(6):171–177, 2001.

[55] W. F. van Gunsteren and H. JC Berendsen. Computer simulation of molecular dynamics: methodology, applications, and perspectives in chemistry. *Angew. Chem. Int. Edit.*, 29(9):992–1023, 1990.

[56] A. Warshel and M. Levitt. Theoretical studies of enzymic reactions: dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme. *Journal of molecular biology*, 103(2):227–249, 1976.

[57] Y. Kikuchi, Y. Imamura, and H. Nakai. One-body energy decomposition schemes revisited: Assessment of mulliken-, grid-, and conventional energy density analyses. *Int. J. Quantum Chem.*, 109(11):2464–2473, 2009.

[58] R. Khare, S. L. Mielke, J. T. Paci, S. Zhang, R. Ballarini, G. C. Schatz, and T. Belytschko. Coupled quantum mechanical/molecular mechanical modeling of the fracture of defective carbon nanotubes and graphene sheets. *Phys. Rev. B*, 75(7):075412, 2007.

[59] S. Ogata, E. Lidorikis, F. Shimojo, A. Nakano, P. Vashishta, and R. K. Kalia. Hybrid finite-element/molecular-dynamics/electronic-density-functional approach to materials simulations on parallel computers. *Com. Phys. Comm.*, 138(2):143–154, 2001.

[60] P. Sherwood, A. H. de Vries, M. F. Guest, G. Schreckenbach, C. R. A. Catlow, S. A. French, A. A. Sokol, S. T. Bromley, W. Thiel, A. J. Turner, et al. QUASI: a general purpose implementation of the QM/MM approach and its application to problems in catalysis. *J. Mol. Struc.-THEOCHEM*, 632(1):1–28, 2003.

[61] W. Kohn. Density functional and density matrix method scaling linearly with the number of atoms. *Phys. Rev. Lett.*, 76(17):3168, 1996.

[62] N. Reuter, A. Dejaegere, B. Maigret, and M. Karplus. Frontier bonds in QM/MM methods: a comparison of different approaches. *J. Phys. Chem. A*, 104(8):1720–1735, 2000.

[63] A. Laio, J. VandeVondele, and U. Rothlisberger. A Hamiltonian electrostatic coupling scheme for hybrid Car–Parrinello molecular dynamics simulations. *J. Chem. Phys.*, 116(16):6941–6947, 2002.

[64] J. R. Kermode, T. Albaret, D. Sherman, N. Bernstein, P. Gumbsch, M. C. Payne, G. Csányi, and A. De Vita. Low-speed fracture instabilities in a brittle crystal. *Nature*, 455(7217):1224–1227, 2008.

[65] N. Bernstein, J. R. Kermode, and G. Csányi. Hybrid atomistic simulation methods for materials systems. *Rep. Prog. Phys.*, 72(2):026501, 2009.

[66] A. De Vita and R. Car. A novel scheme for accurate MD simulations of large systems. In *MRS Proceedings*, volume 491, page 473. Cambridge Univ Press, 1997.

[67] G. Csányi, T. Albaret, M. C. Payne, and A. De Vita. "Learn on the fly": a hybrid classical and quantum-mechanical molecular dynamics simulation. *Phys. Rev. Lett.*, 93(17):175503, 2004.

[68] G. Csányi, G. Moras, J. R. Kermode, M. C. Payne, A. Mainwood, and A. De Vita. Multiscale modeling of defects in semiconductors: a novel molecular dynamics scheme. In *Theory of Defects in Semiconductors*, pages 193–212. Springer, 2007.

[69] A. Peguiron, L. Colombi Ciacchi, A. De Vita, J. R. Kermode, and G. Moras. Accuracy of buffered-force QM/MM simulations of silica. *J. Chem. Phys.*, 142(6):064116, feb 2015.

[70] C.-K. Skylaris, P. D. Haynes, A. A. Mostofi, and M. C. Payne. Introducing ONETEP: Linear-scaling density functional simulations on parallel computers. *J. Chem. Phys.*, 122(8):084119, 2005.

[71] J. M. Soler, E. Artacho, J. D. Gale, A. García, J. Junquera, P. Ordejón, and D. Sánchez-Portal. The SIESTA method for ab initio order–N materials simulation. *J. Phys.-Condens. Mat.*, 14(11):2745, 2002.

[72] N. DM Hine, P. D. Haynes, A. A. Mostofi, C.-K. Skylaris, and M. C. Payne. Linear-scaling density-functional theory with tens of thousands of atoms: Expanding the scope and scale of calculations with onetep. *Com. Phys. Comm.*, 180(7):1041–1053, 2009.

[73] L. Mones, A. Jones, A. W. Götz, T. Laino, R. C. Walker, B. Leimkuhler, G. Csányi, and N. Bernstein. The adaptive buffered force QM/MM method in the CP2K and AMBER software packages. *J. Comput. Chem.*, 36(9):633–648, 2015.

[74] M. Caccin, Z. Li, J. R. Kermode, and A. De Vita. A framework for machine-learning-augmented multiscale atomistic simulations on parallel supercomputers. *Int. J. Quantum Chem.*, 115(16):1129–1139, aug 2015.

[75] Charles-Edmond Bichot and Patrick Siarry. *General Introduction to Graph Partitioning*, pages 1–25. John Wiley & Sons, Inc., 2013.

[76] George Karypis and Vipin Kumar. Metis–unstructured graph partitioning and sparse matrix ordering system, version 2.0. 1995.

[77] D. Lee, S. Park, J. Lee, and N. Hwang. A theoretical model for digestive ripening. *Acta Mater.*, 55(15):5281–5288, sep 2007.

[78] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[79] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[80] G. Kresse and J. Hafner. Ab initio molecular dynamics for liquid metals. *Phys. Rev. B*, 47(1):558, 1993.

[81] J. Hutter, M. Iannuzzi, F. Schiffmann, and J. VandeVondele. cp2k: atomistic simulations of condensed matter systems. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 4(1):15–25, 2014.

[82] J. R. Kermode, A. Gleizer, G. Kovel, L. Pastewka, G. Csányi, D. Sherman, and A. De Vita. Low speed crack propagation via kink formation and advance on the silicon (110) cleavage plane. *Phys. Rev. Lett.*, 115(13):135501, sep 2015.

[83] A. A. Griffith. The phenomena of rupture and flow in solids. *Philosophical transactions of the royal society of london. Series A, containing papers of a mathematical or physical character*, 221:163–198, 1921.

[84] L. B. Freund. *Dynamic fracture mechanics*. Cambridge university press, 1998.

[85] M. Marder. Molecular dynamics of cracks. *Computing in science & engineering*, 1(5):48–55, 1999.

[86] B. E. Warren and J. Biscce. The structure of silica glass by X-ray diffraction studies. *Journal of the American Ceramic Society*, 21(2):49–54, 1938.

[87] R. L. Mozzi and B. E. Warren. The structure of vitreous silica. *Journal of Applied Crystallography*, 2(4):164–172, 1969.

[88] W. H. Zachariasen. The atomic arrangement in glass. *J. Am. Chem. Soc.*, 54(10):3841–3851, 1932.

[89] R. J. Bell and P. Dean. Properties of vitreous silica: Analysis of random network models. *Nature*, 212(5068):1354–1356, 12 1966.

[90] D. L. Evans and S. V. King. Random network model of vitreous silica. *Nature*, 212(5068):1353–1354, 12 1966.

[91] D. S. Franzblau. Computation of ring statistics for network models of solids. *Phys. Rev. B*, 44(10):4925, 1991.

[92] J. Sarnthein, A. Pasquarello, and R. Car. Model of vitreous SiO_2 generated by an ab initio molecular-dynamics quench from the melt. *Phys. Rev. B*, 52(17):12690, 1995.

[93] R. M. Van Ginhoven, H. Jónsson, and L. R. Corrales. Silica glass structure generation for ab initio calculations using small samples of amorphous silica. *Phys. Rev. B*, 71(2):024208, 2005.

[94] J. Weissenrieder, S. Kaya, J.-L. Lu, H.-J. Gao, S. Shaikhutdinov, H.-J. Freund, M. Sierka, T. K. Todorova, and J. Sauer. Atomic structure of a thin silica film on a Mo(112) substrate: A two-dimensional network of SiO_4 tetrahedra. *Phys. Rev. Lett.*, 95(7):076103, 2005.

[95] L. Lichtenstein, C. Büchner, B. Yang, S. Shaikhutdinov, M. Heyde, M. Sierka, R. Włodarczyk, J. Sauer, and H.-J. Freund. The atomic structure of a metal-supported vitreous thin silica film. *Angew. Chem. Int. Edit.*, 51(2):404–407, 2012.

[96] P. Y. Huang, S. Kurasch, A. Srivastava, V. Skakalova, J. Kotakoski, A. V. Krasheninnikov, R. Hovden, Q. Mao, J. C. Meyer, J. Smet, et al. Direct imaging of a two-dimensional silica glass on graphene. *Nano Lett.*, 12(2):1081–1086, 2012.

[97] D. Löffler, J. J. Uhlrich, M. Baron, B. Yang, X. Yu, L. Lichtenstein, L. Heinke, C. Büchner, M. Heyde, S. Shaikhutdinov, H.-J. Freund, R. Włodarczyk, M. Sierka, and J. Sauer. Growth and structure of crystalline silica sheet on Ru(0001). *Phys. Rev. Lett.*, 105:146104, Sep 2010.

[98] P. Y. Huang, S. Kurasch, J. S. Alden, A. Shekhawat, A. A. Alemi, P. L. McEuen, J. P. Sethna, U. Kaiser, and D. A. Muller. Imaging atomic rearrangements in two-dimensional silica glass: Watching silica's dance. *Science*, 342(6155):224–227, 2013.

[99] L. Lichtenstein, M. Heyde, and H.-J. Freund. Crystalline-vitreous interface in two dimensional silica. *Phys. Rev. Lett.*, 109(10):106101, 2012.

[100] M. Wilson, A. Kumar, D. Sherrington, and M. F. Thorpe. Modeling vitreous silica bilayers. *Phys. Rev. B*, 87(21):214108, 2013.

[101] J. VandeVondele, M. Krack, F. Mohamed, M. Parrinello, T. Chassaing, and J. Hutter. Quickstep: Fast and accurate density functional calculations using a mixed gaussian and plane waves approach. *Com. Phys.Comm.*, 167(2):103–128, 2005.

[102] L. Pastewka and J. R. Kermode. Matscipy: Repository for generic python materials science tools.

[103] J. R. Kermode. *Multiscale Hybrid Simulation of Brittle Fracture*. PhD thesis, University of Cambridge, 2007.

[104] S. R. Bahn and K. W. Jacobsen. An object-oriented scripting interface to a legacy electronic structure code. *Comput. Sci. Eng.*, 4(3):56–66, May-Jun 2002.

[105] J. G. Swadener, M. I. Baskes, and M. Nastasi. Molecular dynamics simulation of brittle fracture in silicon. *Phys. Rev. Lett.*, 89:085503, Aug 2002.

[106] D. Herzbach, K. Binder, and M. H. Müser. Comparison of model potentials for molecular dynamics simulations of silica. *J. Chem. Phys.*, 123(12):124711, 2005.

[107] M. D. Segall, P. Lindan, M. J. Probert, C. J. Pickard, P. J. Hasnip, S. J. Clark, and M. C. Payne. First-principles simulation: ideas, illustrations and the CASTEP code. *J. Phys.-Condens. Mat.*, 14(11):2717, 2002.

[108] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2005.

[109] L. D. Landau and E. M. Lifshitz. Theory of elasticity, 3rd, 1986.

[110] J. Muscat, V. Swamy, and N. M. Harrison. First-principles calculations of the phase stability of TiO_2. *Phys. Rev. B*, 65(22):224112, 2002.

[111] Matthias Rupp, Raghunathan Ramakrishnan, and O Anatole von Lilienfeld. Machine learning for quantum mechanical properties of atoms in molecules. *arXiv preprint arXiv:1505.00350*, 2015.

[112] C. A. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17(1):177–204, 2005.

[113] C. Carmeli, E. De Vito, and A. Toigo. Vector valued reproducing kernel Hilbert spaces of integrable functions and Mercer theorem. *Analysis and Applications*, 4(04):377–408, 2006.

[114] M. A. Alvarez, L.o Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: a review. *arXiv preprint arXiv:1106.6251*, pages 1–37, June 2011.

[115] T. Stecher, N. Bernstein, and G. Csányi. Free energy surface reconstruction from umbrella samples using Gaussian process regression. *J. Chem. Theory Comput.*, 10(9):4079–4097, 2014. PMID: 26588550.

[116] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[117] A. Gleizer, G. Peralta, J. R. Kermode, A. De Vita, and D. Sherman. Dissociative chemisorption of O_2 inducing stress corrosion cracking in silicon crystals. *Phys. Rev. Lett.*, 112(11):115501, 2014.

[118] L. M. Ghiringhelli, J. Vybiral, S. V. Levchenko, C. Draxl, and M. Scheffler. Big data of materials science: Critical role of the descriptor. *Phys. Rev. Lett.*, 114(10):105503, 2015.