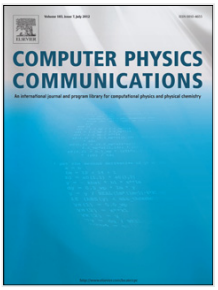
Accepted Manuscript



A highly-efficient technique for evaluating bond-orientational order parameters

Szymon Winczewski, Jacek Dziedzic, Jarosław Rybicki

PII: S0010-4655(15)00347-1
DOI: http://dx.doi.org/10.1016/j.cpc.2015.09.009
Reference: COMPHY 5759

To appear in: *Computer Physics Communications*

Received date: 11 March 2015
Revised date: 10 September 2015
Accepted date: 12 September 2015

Please cite this article as: S. Winczewski, J. Dziedzic, J. Rybicki, A highly-efficient technique for evaluating bond-orientational order parameters, *Computer Physics Communications* (2015), http://dx.doi.org/10.1016/j.cpc.2015.09.009

A highly-efficient technique for evaluating bond-orientational order parameters

Szymon Winczewski[a], Jacek Dziedzic[a,b], Jarosław Rybicki[a,c]

[a] *Faculty of Applied Physics and Mathematics, Gdansk University of Technology, Narutowicza 11/12, 80-952 Gdańsk, Poland.*
[b] *School of Chemistry, University of Southampton, Highfield, Southampton SO17 1BJ, UK*
[c] *TASK Computer Centre, Gdansk University of Technology, Narutowicza 11/12, 80-952 Gdańsk, Poland*

Abstract

We propose a novel, highly-efficient approach for the evaluation of bond-orientational order parameters (BOPs). Our approach exploits the properties of spherical harmonics and Wigner $3j$-symbols to reduce the number of terms in the expressions for BOPs, and employs simultaneous interpolation of normalised associated Legendre polynomials and trigonometric functions to dramatically reduce the total number of arithmetic operations. Using realistic test cases, we show how the above, combined with a CPU-cache-friendly data structure, leads to a 10- to 50-fold performance increase over approaches currently in use, depending on the size of the interpolation grids and the machine used. As the proposed approach is an approximation, we demonstrate that the errors it introduces are well-behaved, controllable and essentially negligible for practical grid sizes. We benchmark our approach against other structure identification methods (centro-symmetry parameter (CSP), common neighbour analysis (CNA), common neighbourhood parameter (CNP) and Voronoi analysis), generally regarded as much faster than BOPs, and demonstrate that our formulation is able to outperform them for all studied systems.

Keywords:
structure identification, molecular dynamics, bond-orientational order, spherical harmonics, linear interpolation
PACS: 61.50.Ah, 02.70.Ns, 02.60.Ed

1. Introduction

Bond-orientational order parameters (BOPs) were proposed by Steinhard *et al.* in 1981 [1, 2] as a generalisation of the two-dimensional hexatic order parameter [3]. Initially BOPs were applied to the study of the orientational order in liquids and glasses, later to become a standard tool in all of solid state physics. Nowadays BOPs are chiefly used to differentiate between crystalline phases, such as sc, bcc, fcc or hcp [4–11].

BOPs have been used to study nucleation and crystal growth [6, 10–17], helping to elucidate the structure of critical nuclei [5] and nucleation kinetics [18]. They also constitute a standard tool for the study of melting processes [19–21], where global BOPs are used as a direct indicator of a phase transtion, while local BOPs serve as measures for the determination of solid and liquid fractions. Studies of undercooling and glassification [10, 17, 22, 23] also employ BOPs, as do investigations of local icosahedral order in liquid metals [21, 24] and in other systems [1, 2, 25, 26].

Many model systems have been studied with the aid of BOPs: hard- [6, 16, 27–29] and soft-spheres [4], Lennard-Jones systems [5, 9, 13, 14, 18, 20, 30, 31] (including binary [32] and polydisperse [22]), quantum

Email address: wisnia@kdm.task.gda.pl (Szymon Winczewski)

Lennard-Jones solids [19, 33] and Gaussian-core systems [9, 17]. Systems described with more complex potentials, such as Morse [34], modified Buckingham [15] and many-body potentials (pair functionals) [7, 8, 35, 36] have also been studied using BOPs.

The BOP technique is commonly used in the investigations of nanoscale systems, such as atomic clusters [35–37], and gold nanowires [7, 8]. BOPs have also been used in studies of shear-induced phenomena (e.g. shear-induced ordering [38], shear-induced crystallisation [39], and shear-induced overaging [40]), anomalies in liquids [41, 42], the freezing of argon in porous carbon [43], quasicrystals [26], and even plasma [44].

The widespread use of the BOP technique spurred a number of extensions or generalisations over the last decade. A modification, where an additional averaging over nearest neighbours is performed during the calculation of local BOPs was proposed by Lechner *et al.* [9] in order to improve identification of a variety of crystalline structures. A new formulation, where BOPs are combined with Voronoi tesselation, was proposed by Mickel *et al.* [45]. This formulation removes the ambiguities introduced by the arbitrariness in the choice of a cutoff radius, allowing a better characterisation of the orientational order of disordered systems.

The fact that calculating BOPs involves repeated evaluation of spherical harmonics (SHs) [46] means that it is a computationally intensive approach. In a benchmark of methods for structural analysis Stukowski [47] assigns it a computational cost factor of 100, compared with 50 for Voronoi analysis [48, 49], 3 for common neighbour analysis (CNA) [50] and 1 for the centro-symmetry parameter technique (CSP) [51]. The high computational effort associated with BOPs narrows the spectrum of their potential applications, and we are not aware of any examples in the literature where they would be used for the analysis of large-scale simulations – these typically employ computationally cheaper methods, such as energy filtering (e.g. Ref. [52, 53]), or CSP (e.g. Ref. [54]). Being able to use a more involved method that BOP constitutes for large-scale systems is an enticing prospect. A discussion of accuracy and limitations inherent in a number of approaches to structural analysis is given in Ref. [47].

Moreover, the availability of the BOP approach to researchers is limited. To our knowledge, its implementation is not bundled with any of the well-known molecular dynamics codes or visualisation tools, while CSP, CNA or Voronoi analysis are offered by e.g. LAMMPS [55] or OVITO [56]. The authors are aware of only two implementations available to the scientific community: one due to Lechner *et al.* [9], and another one due to Wang *et al.* [36]. Both of these implementations are less computationally efficient compared to competing, simpler approaches.

With the above considerations in mind, we feel a highly-efficient approach for the evaluation of bond-orientational order parameters has the potential to widen their spectrum of application. In this paper we propose a novel, approximate method for efficient calculation of BOPs, which can reduce the computational effort by a factor of up to 50, allowing it to outperform even the four approaches generally regarded as faster, i.e. CSP, CNA, CNP (common neighbourhood parameter) [57] and Voronoi analysis.

The paper is organised as follows. In section 2 we describe the BOP approach, highlighting the steps in the calculation that can be optimised. Section 3 outlines the proposed technique for evaluating BOPs. Section 4 is devoted to benchmarking the efficiency and accuracy of the proposed approach. Section 5 contains conclusions.

2. The bond-orientational order parameter (BOP) method

Bond-orientational order parameters [1, 2] (BOPs) are used to characterise short-range order by classifying each atom as belonging to one of a number of close-packed structures. For every reference atom i, the classification is performed in four stages, outlined below.

Stage 1.
The set $B(i)$ of nearest neighbours j of atom i is determined. Nearest neighbours are defined as atoms that are no further away from atom i than a prescribed cutoff radius r_c. The bond vectors $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ joining atom i with the neighbours j are calculated as Cartesian components. The number of neighbours of atom i (cardinality of $B(i)$) will be denoted by $N_\mathrm{b}(i)$.

Structure	Parameter			
	Q_4	Q_6	$\hat{W}_4$	$\hat{W}_6$
sc	0.76376	0.35355	0.15932	0.01316
bcc	0.08202	0.50083	0.15932	0.01316
fcc	0.19094	0.57452	-0.15932	-0.01315
hcp	0.09722	0.48476	0.13410	-0.01244
icos	0	0.66332	0	-0.16975

Table 1: BOP values for typical ideal structures.

Stage 2.
Each bond vector $\mathbf{r}_{ij}$ is projected to the unit sphere, and its spherical coordinates $\theta(\mathbf{r}_{ij})$ and $\phi(\mathbf{r}_{ij})$ are calculated.

Stage 3.
A vector of complex spherical harmonics (SHs) [46] $Y_l^m(\theta(\mathbf{r}_{ij}), \phi(\mathbf{r}_{ij}))$ is evaluated for every bond vector $\mathbf{r}_{ij}$, for a chosen value of l and $m \in \{-l, \ldots, l\}$.

Stage 4.
A vector of complex quantities $Q_{l,m}(i)$, defined as

$$Q_{l,m}(i) = \frac{1}{N_\mathrm{b}(i)} \sum_{j \in B(i)} Y_l^m(\theta(\mathbf{r}_{ij}), \phi(\mathbf{r}_{ij})) \tag{1}$$

is constructed. Subsequently so-called second-order (Steinhardt) invariants are constructed, according to

$$Q_l(i) = \left(\frac{4\pi}{2l+1} \sum_{m=-l}^{l} |Q_{l,m}(i)|^2 \right)^{1/2}. \tag{2}$$

Third-order invariants [58] can also be constructed:

$$\hat{W}_l(i) = W_l(i) \times \left(\sum_{m=-l}^{l} |Q_{l,m}(i)|^2 \right)^{-3/2}, \tag{3}$$

where

$$W_l(i) = \sum_{\substack{m_1,m_2,m_3 \\ m_1+m_2+m_3=0}} \begin{pmatrix} l & l & l \\ m_1 & m_2 & m_3 \end{pmatrix} Q_{l,m_1}(i)\, Q_{l,m_2}(i)\, Q_{l,m_3}(i). \tag{4}$$

The quantities

$$\begin{pmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \end{pmatrix}$$

are Wigner $3j$-symbols [59].

Invariants for $l \leq 3$ vanish for lattices with cubic symmetry, and in practice bcc, fcc and hcp are differentiated using $l = 4$ and $l = 6$. Each atom i is classified as belonging to a particular structure by direct comparison of calculated invariants $Q_4(i)$, $Q_6(i)$, $\hat{W}_4(i)$ and $\hat{W}_6(i)$ with reference values. The values for a number of ideal structures are given in Table 1.

At non-zero temperatures thermal motions lead to a smearing of the BOP values, and in practice classification is performed on a two-dimensional plane of parameter values (e.g. $Q_4(i) - Q_6(i)$ or $Q_6(i) - \hat{W}_4(i)$), on which regions corresponding to particular packings are defined. For examples see e.g. Refs.[7, 15]. However, no single, consistent approach to the classification itself has been proposed to date.

3. A highly-efficient approach for evaluating BOPs

The four stages of the calculation of BOPs outlined in Sec. 2 differ in computational effort. For the identification of neighbours (stage 1) efficient, linear-scaling algorithms, such as the linked-cell approach [60, 61] are typically used, and calculating bond vectors $\mathbf{r}_{ij}$ and their spherical coordinates (stage 2) are very simple operations. The computational effort of stages 1 and 2 is thus very small.

3.1. Evaluation of spherical harmonics by fast simultaneous interpolation

The most computationally demanding (about 60% of total effort for a standard implementation) is stage 3, where a large number of spherical harmonics (SHs) has to be evaluated, with the effort for the entire system scaling as $O(Nr_c^3)$. As an example, let us consider a system of modest size, $N = 4000$ atoms arranged in an fcc structure (thus with $N_b = 12$ neighbours within r_c). Assuming we are interested in invariants with $l = 4$ and $l = 6$, a total of $(2 \times 4 + 1 + 2 \times 6 + 1) \times 4000 \times 12 = 1.056 \times 10^6$ SHs need to be evaluated. In a direct approach these are calculated as

$$\begin{aligned} Y_l^m(\theta,\phi) &= \widetilde{P}_l^m(\cos\theta)e^{im\phi} \\ &= K_l^m P_l^m(\cos\theta)e^{im\phi} \\ &= \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}P_l^m(\cos\theta) \\ &\times \left[\cos(m\phi) + i\sin(m\phi)\right], \end{aligned} \tag{5}$$

where $P_l^m(x)$ are associated Legendre polynomials (ALPs) and

$$\widetilde{P}_l^m(x) = K_l^m P_l^m(x) \tag{6}$$

denote normalised associated Legendre polynomials (NALPs).

Below we propose a highly-efficient alternative to direct evaluation, based on simultaneous interpolation. Our approach proceeds by interpolating both the trigonometric functions and the NALPs in (6), exploiting the fact that each interpolation node only has to be calculated once for an entire set of interpolated functions. In the discussion that follows we assume a typical case where invariants with $l = 4$ and $l = 6$ are calculated.

The interpolated functions satisfy the necessary requirements for stable interpolation (bounded domain and codomain, continuity, low variability). We divide the domain of NALPs into p intervals, with a width of $h_x = 2/p$ each, locating the interpolation nodes at $x_j = -1 + jh_x$, where $j = 0, 1, \ldots, p-1$. Similarly, we divide the domain of the trigonometric functions into q intervals, with a width of $h_\phi = 2\pi/q$ each, locating the interpolation nodes at $\phi_j = jh_\phi$, where $j = 0, 1, \ldots, q-1$.

Interpolation tables, constructed at the outset of the calculation, are organised as follows:

1. For NALPs we store $\widetilde{P}_l^m(x_i)$ and $\Delta\widetilde{P}_l^m(x_i)$, where

$$\Delta\widetilde{P}_l^m(x_i) = [\widetilde{P}_l^m(x_{i+1}) - \widetilde{P}_l^m(x_i)]/h_x, \tag{7}$$

for all 12 pairs of indices $(l, m) = (4,0), (4,1), \ldots, (4,4), (6,0), (6,1), \ldots, (6,6)$. We exploit two well-known properties of SHs:

$$Y_l^m(\theta(-\mathbf{r}_{ij}), \phi(-\mathbf{r}_{ij})) = (-1)^l\, Y_l^m(\theta(\mathbf{r}_{ij}), \phi(\mathbf{r}_{ij})), \tag{8}$$

$$Y_l^{-m}(\theta,\phi) = (-1)^m\, \overline{Y}_l^m(\theta,\phi) \tag{9}$$

to elide interpolation and storage of values for $m < 0$. The data structure staggers $\widetilde{P}_l^m(x_i)$ and $\Delta\widetilde{P}_l^m(x_i)$ and uses l as the fast-changing index for optimal cache efficiency. The precise ordering of values for a single interpolation node is shown in Table 2. The size of the data structure is $24p$ double precision values.

2. For interpolating trigonometric functions we store $\sin(m\phi)$, $\cos(m\phi)$, $\Delta\sin(m\phi)$, $\Delta\cos(m\phi)$, where

$$\begin{aligned}\Delta\sin(m\phi_i) &= [\sin(m\phi_{i+1}) - \sin(m\phi_i)]/h_\phi,\\ \Delta\cos(m\phi_i) &= [\cos(m\phi_{i+1}) - \cos(m\phi_i)]/h_\phi,\end{aligned} \tag{10}$$

ordered as $\cos(m\phi)$, $\Delta\cos(m\phi)$, $\sin(m\phi)$, $\Delta\sin(m\phi)$, with $m = 1, 2, \ldots, 6$ being the slowly-changing index. The precise ordering of values for a single interpolation node is shown in Table 3. The size of the data structure is $24q$ double precision values. Note that in the calculation of interpolation slopes in (10) we always divide by h_ϕ, and not by mh_ϕ. This helps increase computational efficiency by allowing us to calculate each interpolation node only once for an entire set of functions.

Once the interpolation tables are constructed, the following steps are performed for every interpolation node j. The inputs to the interpolation are ϕ and $x = \cos\theta$.

1. Find interpolation node x_i, compute $\Delta x = x - x_i$.
2. Interpolate $\widetilde{P}_l^m(x)$ for the 12 pairs of (l, m) by linear interpolation:

$$\widetilde{P}_l^m(x) \approx \widetilde{P}_l^m(x_i) + \Delta\widetilde{P}_l^m(x_i) \times \Delta x,$$

3. Find interpolation node ϕ_i, compute $\Delta\phi = \phi - \phi_i$,
4. Interpolate $\sin(m\phi)$ and $\cos(m\phi)$, for $m \in [1, 6]$ by linear interpolation:

$$\begin{aligned}\sin(m\phi) &\approx \sin(m\phi_i) + \Delta\sin(m\phi_i) \times \Delta\phi\\ \cos(m\phi) &\approx \cos(m\phi_i) + \Delta\cos(m\phi_i) \times \Delta\phi,\end{aligned}$$

5. Evaluate spherical harmonics $\tilde{Y}_l^m$, for the 12 pairs of (l, m) according to

$$\tilde{Y}_l^m(\theta, \phi) = \widetilde{P}_l^m(\cos\theta)\,[\cos(m\phi) + i\sin(m\phi)]\,, \tag{11}$$

where the interpolated values are used for the NALPs and the trigonometric functions.

The efficiency of this approach, to which we will refer as fast simultaneous interpolation (FSI), stems from a combination of the following:

- Linear interpolation is inherently faster compared to direct evaluation of trigonometric functions or NALPs.
- For every bond, all 12 spherical harmonics are evaluated for the same arguments.
- The interpolation node is shared by an entire vector of 12 NALPs, and by six pairs of trigonometric functions, and thus needs to be evaluated only once.
- A data structure that ensures maximum locality, and thus efficient use of CPU cache is employed.

With the proposed approach, a calculation of all 12 SHs involves only 27 additions, 48 multiplications, 53 assignments and 2 rounding operations (to find the interpolation nodes). Fast simultaneous interpolation can be easily generalised to utilise higher-order interpolation – this would have the advantage of reducing the memory footprint (as coarser grids would provide equivalent accuracy) at the cost of increasing the number of arithmetic operations required. In Sec. 4.1 we show how the memory footprint of our approach is insignificant for grid sizes used in practice, which makes linear interpolation entirely satisfactory for our needs.

3.2. Reducing the number of operations needed to calculate BOPs from SHs

The final, fourth stage in the determination of BOPs consists in calculating the BOPs from the spherical harmonics evaluated in stage 3. The direct approach (i.e. through (1)-(4)) is suboptimal. If we again use the example of a system with $N_\mathrm{b} = 12$, the computational effort of evaluating Q_4, Q_6, $\hat{W}_4$, $\hat{W}_6$ (which is the typical scenario) is: 1464 additions, 1954 multiplications, 3 divisions and 2 square root operations *for a single atom*. The above assumes that evaluating a complex modulus involves 2 multiplications and one addition.

Below we outline how this effort can be reduced several-fold. First, we again exploit the properties of SHs (8), (9) to elide all calculations where $i > j$ and all calculations where $m < 0$, reducing the computational effort almost by a factor of 4. Subsequently we introduce:

$$q_{l,m}(i) = \sum_{j \in B(i)} Y_l^m(\theta(\mathbf{r}_{ij}), \phi(\mathbf{r}_{ij})), \tag{12}$$

$$q_l(i) = \left(\sum_{m=-l}^{l} |q_{l,m}(i)|^2 \right)^{1/2}, \tag{13}$$

and

$$w_l(i) = \sum_{\substack{m_1,m_2,m_3 \\ m_1+m_2+m_3=0}} \begin{pmatrix} l & l & l \\ m_1 & m_2 & m_3 \end{pmatrix} q_{l,m_1}(i)\ q_{l,m_2}(i)\ q_{l,m_3}(i), \tag{14}$$

which are analogous to the quantities in eqs. (1), (2), (4), except for normalisation factors. Once the above intermediate quantities are evaluated, BOPs can be calculated as

$$Q_l(i) = \left(\frac{4\pi}{2l+1} \right)^{1/2} \frac{q_l(i)}{N_\mathrm{b}(i)} \tag{15}$$

and

$$\hat{W}_l(i) = \frac{w_l(i)}{(q_l(i))^3}. \tag{16}$$

By again exploiting a property of SHs (9), we obtain

$$q_{l,-m}(i) = (-1)^m\ \overline{q}_{l,m}(i), \tag{17}$$

which in turn allows to rewrite (13) as

$$q_l(i) = \left(|q_{l,0}(i)|^2 + 2 \sum_{m=1}^{l} |q_{l,m}(i)|^2 \right)^{1/2}. \tag{18}$$

Let us now consider the sum (14). In a direct approach, 61 and 127 non-vanishing terms (those where $m_1 + m_2 + m_3 = 0$) need to be evaluated for $l = 4$ and $l = 6$, respectively, for a total of 188 terms. We now recall several well-known properties of the Wigner $3j$-symbol:

$$\begin{pmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \end{pmatrix} = \begin{pmatrix} l_2 & l_3 & l_1 \\ m_2 & m_3 & m_1 \end{pmatrix} = \begin{pmatrix} l_3 & l_1 & l_2 \\ m_3 & m_1 & m_2 \end{pmatrix}, \tag{19}$$

$$\begin{pmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \end{pmatrix} = (-1)^{l_1+l_2+l_3} \begin{pmatrix} l_2 & l_1 & l_3 \\ m_2 & m_1 & m_3 \end{pmatrix} = (-1)^{l_1+l_2+l_3} \begin{pmatrix} l_1 & l_3 & l_2 \\ m_1 & m_3 & m_2 \end{pmatrix}, \tag{20}$$

$$\begin{pmatrix} l_1 & l_2 & l_3 \\ -m_1 & -m_2 & -m_3 \end{pmatrix} = (-1)^{l_1+l_2+l_3} \begin{pmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \end{pmatrix}. \tag{21}$$

By exploiting the above properties along with (15) we can reduce the computational effort to merely 9 terms for $l = 4$ and 16 terms for $l = 6$, recasting $w_4(i)$ as

$$\begin{aligned} w_4(i) &= \mathrm{Re}\left(q_{4,0}(i)\right)\left[\begin{pmatrix} 4 & 4 & 4 \\ 0 & 0 & 0 \end{pmatrix}|q_{4,0}(i)|^2 + 6\sum_{m=1}^{4}\begin{pmatrix} 4 & 4 & 4 \\ -m & 0 & m \end{pmatrix}|q_{4,m}(i)|^2\right] \\ &+ 12\begin{pmatrix} 4 & 4 & 4 \\ -4 & 1 & 3 \end{pmatrix}\mathrm{Re}\left(\overline{q}_{4,4}(i)\times q_{4,1}(i)\times q_{4,3}(i)\right) \\ &- 12\begin{pmatrix} 4 & 4 & 4 \\ -3 & 1 & 2 \end{pmatrix}\mathrm{Re}\left(\overline{q}_{4,3}(i)\times q_{4,1}(i)\times q_{4,2}(i)\right) \\ &+ 6\begin{pmatrix} 4 & 4 & 4 \\ -4 & 2 & 2 \end{pmatrix}\mathrm{Re}\left(\overline{q}_{4,4}(i)\times q_{4,2}(i)\times q_{4,2}(i)\right) \\ &+ 6\begin{pmatrix} 4 & 4 & 4 \\ -2 & 1 & 1 \end{pmatrix}\mathrm{Re}\left(\overline{q}_{4,2}(i)\times q_{4,1}(i)\times q_{4,1}(i)\right), \end{aligned} \tag{22}$$

and $w_6(i)$ as

$$\begin{aligned} w_6(i) &= \mathrm{Re}\left(q_{6,0}(i)\right)\left[\begin{pmatrix} 6 & 6 & 6 \\ 0 & 0 & 0 \end{pmatrix}|q_{6,0}(i)|^2 + 6\sum_{m=1}^{6}\begin{pmatrix} 6 & 6 & 6 \\ -m & 0 & m \end{pmatrix}|q_{6,m}(i)|^2\right] \\ &+ 12\begin{pmatrix} 6 & 6 & 6 \\ -6 & 1 & 5 \end{pmatrix}\mathrm{Re}\left(\overline{q}_{6,6}(i)\times q_{6,1}(i)\times q_{6,5}(i)\right) \\ &+ 12\begin{pmatrix} 6 & 6 & 6 \\ -6 & 2 & 4 \end{pmatrix}\mathrm{Re}\left(\overline{q}_{6,6}(i)\times q_{6,2}(i)\times q_{6,4}(i)\right) \\ &+ 12\begin{pmatrix} 6 & 6 & 6 \\ -4 & 1 & 3 \end{pmatrix}\mathrm{Re}\left(\overline{q}_{6,4}(i)\times q_{6,1}(i)\times q_{6,3}(i)\right) \\ &- 12\begin{pmatrix} 6 & 6 & 6 \\ -5 & 1 & 4 \end{pmatrix}\mathrm{Re}\left(\overline{q}_{6,5}(i)\times q_{6,1}(i)\times q_{6,4}(i)\right) \\ &- 12\begin{pmatrix} 6 & 6 & 6 \\ -5 & 2 & 3 \end{pmatrix}\mathrm{Re}\left(\overline{q}_{6,5}(i)\times q_{6,2}(i)\times q_{6,3}(i)\right) \\ &- 12\begin{pmatrix} 6 & 6 & 6 \\ -3 & 1 & 2 \end{pmatrix}\mathrm{Re}\left(\overline{q}_{6,3}(i)\times q_{6,1}(i)\times q_{6,2}(i)\right) \\ &+ 6\begin{pmatrix} 6 & 6 & 6 \\ -6 & 3 & 3 \end{pmatrix}\mathrm{Re}\left(\overline{q}_{6,6}(i)\times q_{6,3}(i)\times q_{6,3}(i)\right) \\ &+ 6\begin{pmatrix} 6 & 6 & 6 \\ -4 & 2 & 2 \end{pmatrix}\mathrm{Re}\left(\overline{q}_{6,4}(i)\times q_{6,2}(i)\times q_{6,2}(i)\right) \\ &+ 6\begin{pmatrix} 6 & 6 & 6 \\ -2 & 1 & 1 \end{pmatrix}\mathrm{Re}\left(\overline{q}_{6,2}(i)\times q_{6,1}(i)\times q_{6,1}(i)\right). \end{aligned} \tag{23}$$

The above forms, although more verbose, allow for markedly more efficient computation – not only is the number of terms reduced (from 188 to 25), but the terms themselves are simpler. The number of operations necessary to calculate BOPs for a single atom (for the same model case) is reduced to 337 additions (from 1464), 176 multiplications (from 1954), 2 divisions (from 3). The number of square-root operations remains at 2.

4. Efficiency and accuracy of the proposed approach

4.1. Fast simultaneous interpolation

Here we demonstrate the efficiency and accuracy of the simultaneous interpolation scheme we proposed in Sec. 3.1. As an efficient and accurate approach to evaluating large numbers of spherical harmonics can

also find use outside the context of the BOP method, we begin by benchmarking the sole calculation of the SHs.

Our benchmark consisted in calculating the set of 12 SHs for 1 million randomly [62] chosen pairs of (θ, ϕ). As a reference implementation we used the GNU Scientific Library [63] (GSL), which is widely used in the scientific community. Since GSL does not directly calculate SHs, but only NALPs, an additional calculation of trigonometric functions was needed, this was done through calls to native library functions.

The efficiency and accuracy of any interpolation scheme will depend on the size of the interpolation grid(s) used. Our approach uses two grids, with p and q intervals, respectively (cf. Sec. 3.1), resulting in $O(p+q)$ memory complexity. The time complexity of evaluating the SHs is $O(1)$ (independent of the size of either grid), while the computational effort of constructing the interpolation grids scales as $O(p+q)$. In practical applications the computational effort would be dominated by the interpolation itself, with the cost of constructing the grids becoming increasingly more insignificant with increasing numbers of SHs calculated.

We begin by reporting (Fig. 1) the mean error in the obtained SHs as a function of p and q, using the numerically exact GSL values as reference. Since the values of SHs can be arbitrarily close to complex zero, we elect to report errors defined as

$$E = \left| \tilde{Y}_l^m(\theta(\mathbf{r}_{ij}), \phi(\mathbf{r}_{ij})) - Y_l^m(\theta(\mathbf{r}_{ij}), \phi(\mathbf{r}_{ij})) \right|, \tag{24}$$

where $\tilde{Y}_l^m$ are the spherical harmonics evaluated with our approach according to (11), and Y_l^m are understood to be the numerically exact values obtained with GSL. By mean error, $\langle E \rangle$, we shall denote the average of E over the 12 SHs calculated for 1 million randomly generated pairs of (θ, ϕ). We acknowledge that for SHs extremely close (or exactly equal) to complex zero, the *relative* error of the approximation can become arbitrarily large. In applications where this would be deemed problematic (e.g. where accurate phases were vital for arbitrarily small moduli), numerically exact calculations could easily be carried out once the value of the modulus was found to lie below a preset threshold. Here we chose not employ such a fallback mechanism so as not to introduce a dependence of results on the value of the threshold.

Fig. 1 makes it apparent that highest accuracy, for a given memory footprint (proportional to $p+q$), is obtained by using similar grid sizes, i.e. $p \approx q$. In order to simplify further analysis, in the text that follows we assume $p = q$, and accordingly report efficiency and accuracy benchmarks for the case where the grid sizes are taken to be identical.

We now turn to benchmarking the efficiency of the interpolation scheme for the SHs. We carefully ensured that the control logic for traversing the pairs was identical between the reference and proposed approaches and that that only the walltimes of the actual SH evaluations were measured. In an effort to minimise the effect of any external factors (concurrent processes, paging, caching, etc.) we performed 10 runs for each approach, presenting averaged results. Since the relative performance of any two implementations is likely to vary depending on the machine of which they are executed, we performed benchmarks on three distinct machines. Machine A was a high-performance workstation, machine B was a typical computational node of a computer cluster, and machine C was a laptop. The three machines differed with respect to clock rates, CPU types, CPU cache and RAM speeds – the relevant hardware details are given in Table 4.

Fig. 2 shows the speedup of the proposed approach compared to the reference (GSL) implementation, demonstrating that simultaneous interpolation is several tens of times faster for machines typically used in scientific computations and 21-45 times faster on a laptop computer (machine C), with greatest relative speedups obtained for the fastest machine (machine A).

We will now demonstrate that the above speedups were obtained for grid sizes that guarantee negligible loss of accuracy. In Fig. 3 we show the mean error $\langle E \rangle$ of the calculated complex spherical harmonics as a function of the interpolation grid sizes $p = q$. Fast simultaneous interpolation is seen to yield very accurate SHs even for modest grid sizes – for the coarsest setting ($p = q = 600$ intervals) the errors were as small as $\langle E \rangle = 5.6 \times 10^{-5}$. The error and thus the accuracy is seen to be controllable – the mean error is well-described with a relation $\langle E \rangle = Cp^{-\alpha}$ (shown as line in the plot). The empirically obtained value of the exponent α was 1.68, meaning that a fourfold increase of the grid size p leads to a 10.3-fold decrease of $\langle E \rangle$.

Fig. 4 shows the distribution of the errors, demonstrating that even for the coarsest grids ($p = q = 600$)

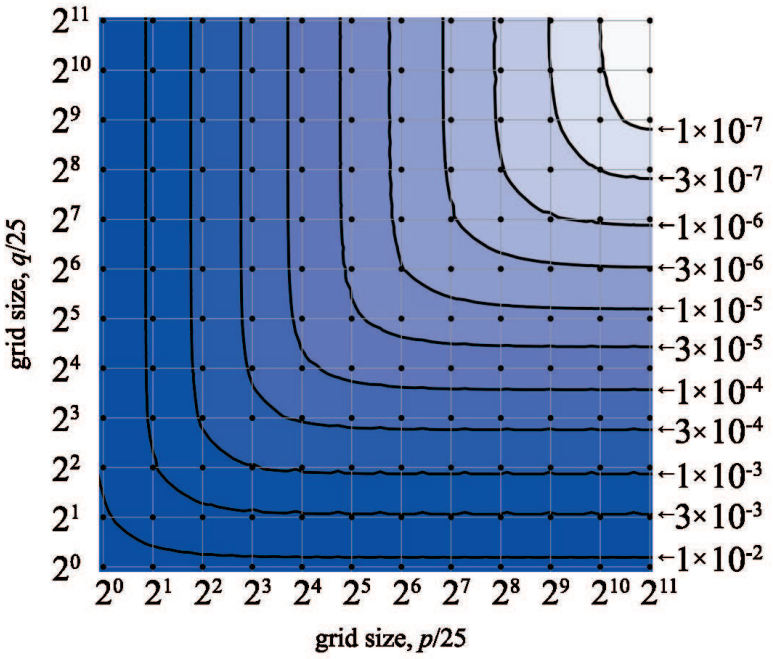


Figure 1: Errors introduced by the proposed approximation (shown as contour labels to the right of the plot). Black points represent tested grid size pairs. Grid sizes p and q between 25 and $25 \times 2^{11} = 51200$ were tested. Contour lines were obtained using biquadratic interpolation between the points.

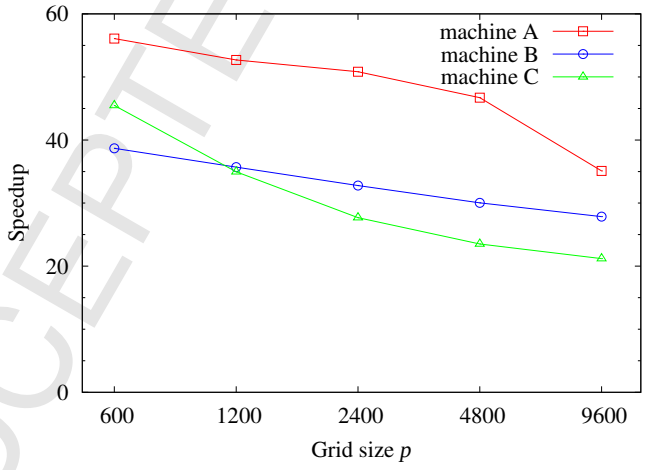


Figure 2: Speedup of the proposed simultaneous interpolation compared to reference (GSL) implementation, for calculating 1M spherical harmonics.

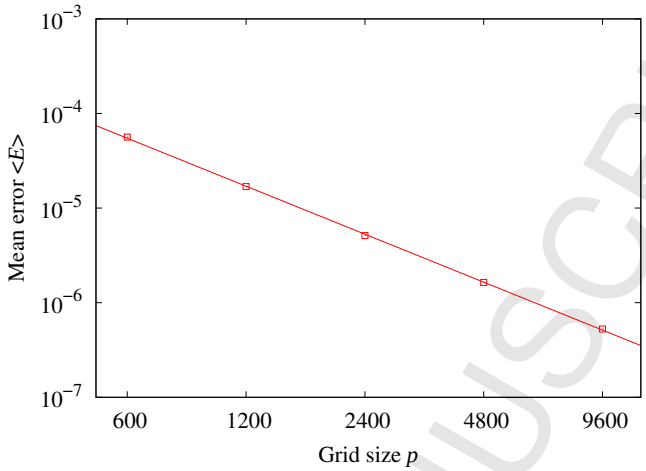


Figure 3: A log-log plot of the error introduced by the approximations of the fast simultaneous interpolation. The line is a result of fitting $Cp^{-\alpha}$ to the datapoints.

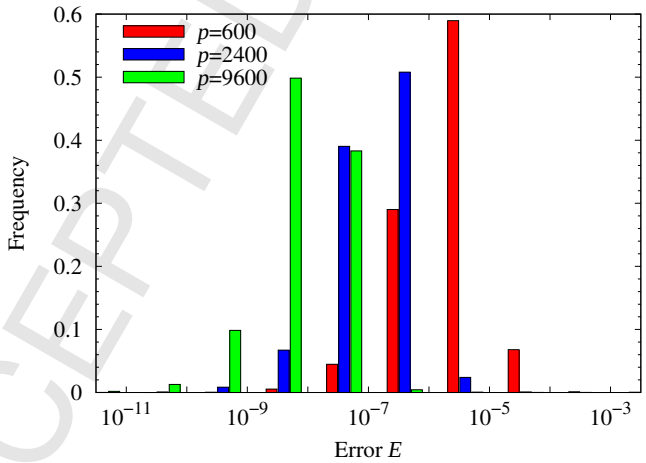


Figure 4: Distribution histograms for the error (calculated according to 24) in the calculated spherical harmonics for three representative grid sizes (p).

they are well-behaved and increasing grid sizes quickly moderates the error distribution. For the medium-quality grids ($p = q = 2400$) we find that in $> 99.98\%$ cases the error E did not exceed 10^{-4}.

We finish the discussion of fast simultaneous interpolation with a comment on memory use. The proposed approach has very modest memory requirements – even the finest grids used here ($p = q = 9600$), for which the approximation errors are extremely small ($\langle E \rangle = 5.3 \times 10^{-7}$), necessitate storing only $(p + q) \times 24 = (9600 + 9600) \times 24 = 460800$ double precision values, for a total of only 3.7 MB.

4.2. Efficiency and accuracy of the proposed approach to evaluating BOPs

We shall now assess the efficiency and accuracy of the approach to evaluating BOPs that combines the fast simultaneous interpolation (FSI) described in Sec. 3.1 and the sum simplifications discussed in Sec. 3.2. We compare the implementation of our approach against the two widely used implementations – one due to Lechner *et al.* [9] and one due to Wang *et al.* [36]. Both of these employ the exact, direct approach of (1)-(4). In order to disentangle the effects of FSI (which is an approximation) from the effects of the sum simplifications (which do not involve approximations), we also performed benchmarks for calculations that use GSL to evaluate SHs exactly.

In order to perform a realistic benchmark, we prepared thirteen test cases representing typical systems whose structure would be analysed using the BOP approach. The test cases sample a variety of system sizes and structures (cf. Table 5 for details). We measured the walltime corresponding to evaluating all four BOP values for all atoms in every test case, for 50 different configurations of positions. The configurations were obtained in advance from molecular dynamics simulations. We note that Lechner's implementation does not calculate $\hat{W}_4$ or $\hat{W}_6$, and these two invariants were not included in the datapoints for Lechner's implementation. We used the same three representative machines as in Sec. 4.1 and ensured the test conditions were identical for all implementations. Depending on the machine, the size of the test case, and the implementation we calculated walltimes as averages over 5-1000 independent runs. We were careful to time only the BOP calculation itself, excluding the time necessary to identify nearest neighbours (stage 1). The fact that the laptop machine (C) had very limited RAM (2 GB) prevented us from running the largest, 1M+-atom, systems (1e, 4e, 4f) on this machine. For the same reason, we were not able to benchmark the Lechner implementation for testcase 4f (8M+ atoms) on machine B.

Fig. 5 illustrates the speedups obtained for the thirteen test cases. Our approach is found to be between 11 and 40 times faster compared to the implementation of Lechner *et al.*, depending on machine and test case, and between 14 and 46 times faster compared to that of Wang *et al.*. We note (cf. the datapoints labeled "GSL") that most of the speedup (a factor of 7-24) can be attributed to the use of the interpolation scheme, while the the rearrangements described in Sec. 3.2 account for a further improvement in efficiency by a factor of 1.7-2.1.

As expected, the obtained speedups are seen to depend somewhat on the machine used for benchmarking. These differences result from a combination of hardware factors. We do not set out to identify the exact hardware details responsible for the degree of obtained speedup, rather, we only point out that the proposed computational approach consistently affords speedups of over an order of magnitude, regardless of architecture and hardware details.

We now turn to the assessment of errors that the proposed technique introduces in the values of the BOPs $Q_l(i)$ and $\hat{W}_l(i)$. We followed a practical approach, whereby we compared BOP values calculated using FSI with numerically exact values obtained using GSL for the thirteen test cases used earlier in the text (cf. Table 5 for details). Here we only report on errors measured for test case 3 (liquid AlCu), which performed the worst in terms of errors. The results that we present were averaged over 10000 snapshots taken from an MD calculation in order to ensure good statistics. We calculated mean absolute errors in all four parameters under consideration: Q_4, Q_6, $\hat{W}_4$, and $\hat{W}_6$.

Fig. 6 shows the measured absolute errors in BOPs as a function of the grid size p. For the coarsest grids ($p = q = 600$) the mean absolute errors were 5.3×10^{-5}, 1.1×10^{-4}, 1.4×10^{-4} and 5.0×10^{-5}, for Q_4, Q_6, $\hat{W}_4$, and $\hat{W}_6$, respectively. Using finer grids ($p = q = 9600$) reduced the errors by two orders of magnitude. The above means that the artifacts introduced by the interpolation scheme are well in the realm of negligibility. In typical applications of the BOP method being able to determinie the parameters to the

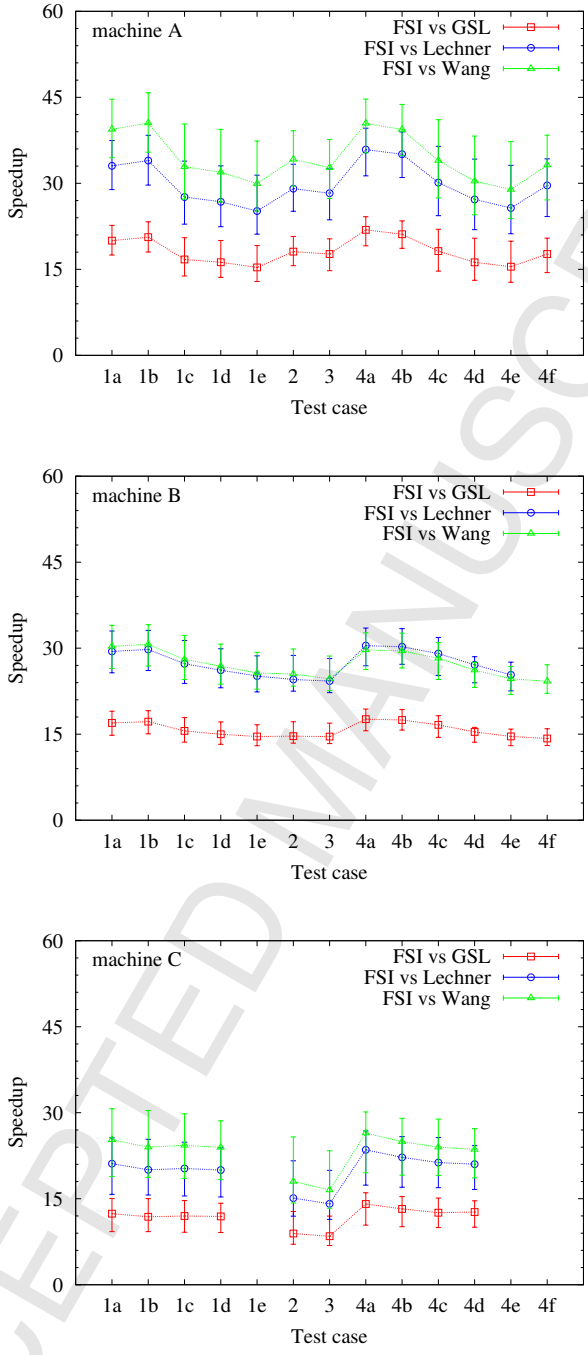


Figure 5: Measured speedup of the proposed approach over the implementations of Lechner *et al.*, Wang *et al.* and GSL. The three panels correspond to machines (A: top, B: middle, C: bottom). The datapoints correspond to grid sizes of $p = q = 2400$, while the error bars correspond to $p = q = 600$ (coarsest grids, highest speedup) and $s = 9600$ (finest grids, lowest speedup). Test cases (cf. Table 5) are shown on the X axis. The timings for the Lechner approach do not include the calculation of $\hat{W}_4$ or $\hat{W}_6$ (see text). Speedups for test cases 1e), 4ef) for machine C, and 4f) for machine B were not measured due to RAM limitations of the test environment (see text).

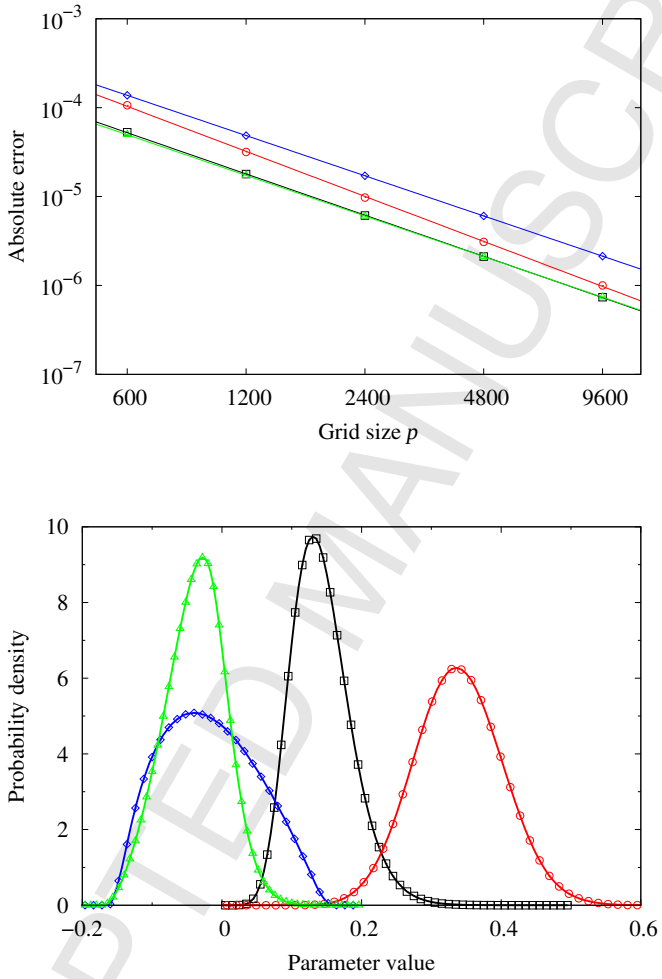


Figure 6: Errors introduced by the proposed approach for evaluating BOPs, measured for and averaged over 10000 snapshots of liquid AlCu (test case 3). The log-log plot in the top panel shows the mean absolute error (squares), with lines denoting results of a fit to the form $Cp^{-\alpha}$. To facilitate comparison with typical values of the parameters, the bottom panel shows the distributions of the BOPs themselves. Both panels use the same coding: black squares – Q_4, red circles – Q_6, blue diamonds – $\hat{W}_4$, green triangles – $\hat{W}_6$.

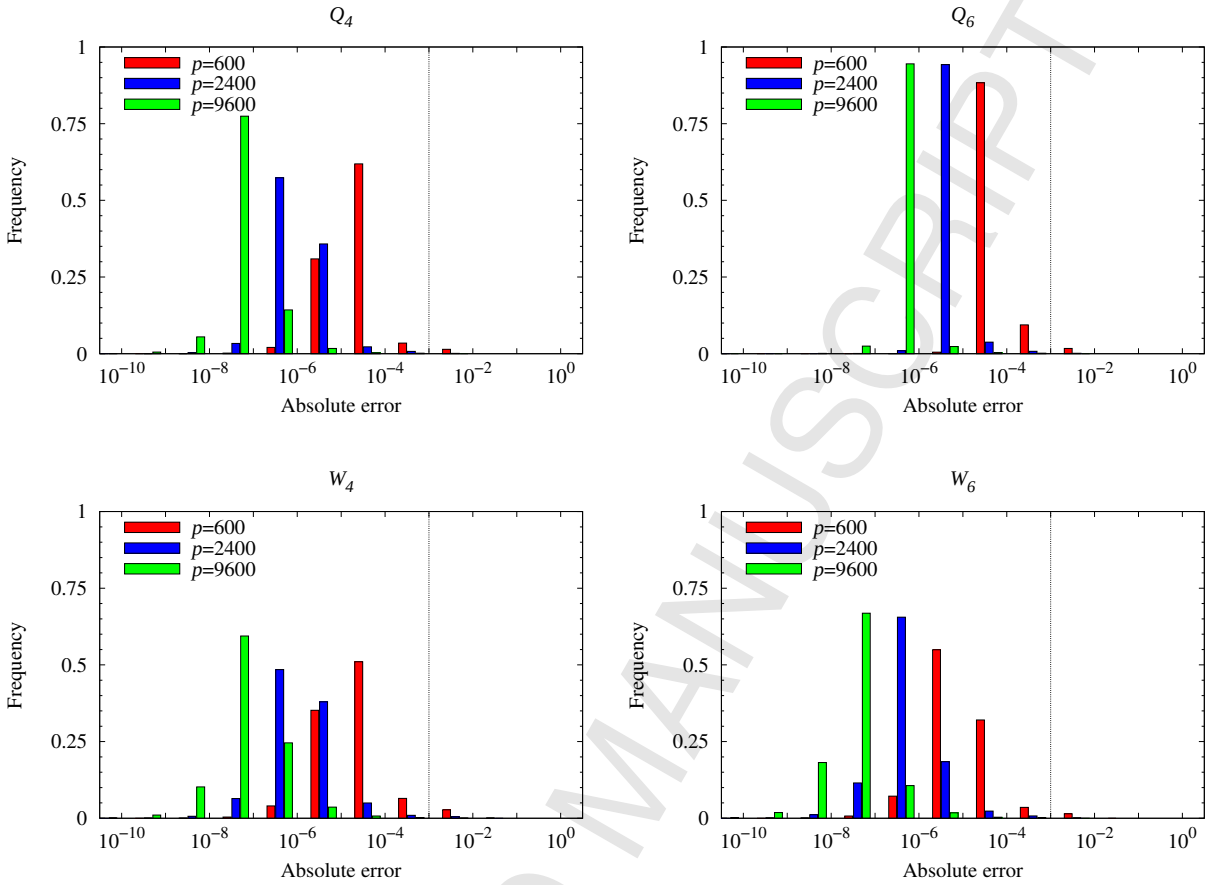


Figure 7: Distribution of absolute errors in BOPs for three grid sizes (coarsest: $p = 600$, medium: $p = 2400$, finest: $p = 9600$). The vertical line denotes an arbitrarily defined acceptable error level of 10^{-3}.

absolute accuracy of 10^{-3} is entirely satisfactory. Our results show that, in the mean sense, such accuracy is obtained even for the coarsest grids we tested ($p = q = 600$).

We appreciate that knowing the *mean* absolute accuracy may not be sufficient for some applications of the BOP method. In Fig. 7 we show how the errors are distributed for three representative grid sizes, for all BOPs of interest. The same test case was used. We find that for grids with $p = q = 9600$, the prevalence rate of absolute errors larger than the acceptance criterion assumed above (10^{-3}) is as small as 0.0004% (for Q_4), 0.001% (for Q_6), 0.06% (for $\hat{W}_4$), 0.002% (for $\hat{W}_6$). None of the 4×10^7 sets of four BOPs constituting our test case exhibited an absolute error larger than 10^{-2}, meaning such errors are expected to occur at most in 2.5×10^{-6}% cases.

4.3. Comparison of computational effort with other methods of structure characterisation

We demonstrated that the proposed approach reduces the computational effort of evaluating bond-orientational order parameters by more than an order of magnitude. The associated increase in computational efficiency is significant enough to warrant a comparison with other approaches to structure characterisation that are generally deemed to be more efficient.

A number of structure identification methods is reviewed in Ref. [47], including four widely used approaches – the centro-symmetry parameter (CSP), common neighbour analysis (CNA), the common neighbourhood parameter (CNP) and Voronoi analysis. Stukowski assigns to each of the reviewed approaches a

unitless computational cost factor, which essentially measures the relative slowness of an approach compared to calculating the centro-symmetry parameter (CSP) [51], excluding the effort of neighbour identification. The computational cost factors given in Ref. [47] for CNA, Voronoi and BOP approaches are, respectively, 3, 50 and 100 (Stukowski did not estimate the computational cost factor of the CNP method).

Below we re-evaluate the computational cost factor using the approach for evaluating BOPs proposed in this work. For the purposes of this comparison, we re-implemented the CSP, CNA, CNP and Voronoi approaches from scratch. The obtained implementations of CSP, CNA and Voronoi analysis mirror those of Stukowski. In both cases Rycroft's `voro++` library [64] has been used to efficiently perform Voronoi tesselation. Below we outline the operations performed within the framework of each approach.

In the CSP approach a single, real-valued parameter given by

$$\mathrm{CSP}(i) = \sum_{j \in B(i)/2} |\mathbf{r}_{ij} + \mathbf{r}_{ik}|^2 \tag{25}$$

needs to be determined for every characterised central atom i. In the above, $\mathbf{r}_{ij}$ and $\mathbf{r}_{ik}$ denote the vectors pointing from atom i to its two opposite neighbours $j, k \in B(i)$, and the number of elements in the sum is equal to half the number of nearest neighbours. The calculation of $\mathrm{CSP}(i)$ necessitates identifying $N_\mathrm{b}(i)/2$ pairs of opposite neighbours, which corresponds to calculating, for every possible pair (j, k) out of the total of $N_\mathrm{b}(i)(N_\mathrm{b}(i)-1)/2$, the value $|\mathbf{r}_{ij} + \mathbf{r}_{ik}|^2$ and using the $N_\mathrm{b}(i)/2$ smallest values in (25). This formulation of CSP has been proposed by Kelchner *et al.* [51] and is implemented, e.g. in the LAMMPS package [55].

In the case of CNA, the analysis consists in the determination of three integer numbers $n_\mathrm{cn}(ij)$, $n_\mathrm{b}(ij)$, and $n_\mathrm{lcb}(ij)$ for every bond $\mathbf{r}_{ij}$ joining the central atom i with its neighbours $j \in B(i)$. These are, respectively, the number of neighbour atoms the central atom i and its neighbour $j \in B(i)$ have in common, $n_\mathrm{cn}(ij)$; the total number of bonds between these common neighbours, $n_\mathrm{b}(ij)$; and the number of bonds in the longest chain of bonds connecting the common neighbours, $n_\mathrm{lcb}(ij)$. Apart from the need to determine $N_\mathrm{b}(i)$ such triples, CNA involves the determination of the number of bonds of different types, which differ in the values of n_cn, n_b, and n_lcb.

The common neighbourhood parameter (CNP), proposed by Tsuzuki *et al.* [57], combines the advantages of CSP and CNA. Here, for every central atom i, a single, real-valued parameter is calculated according to:

$$\mathrm{CNP}(i) = \frac{1}{N_\mathrm{b}(i)} \sum_{j \in B(i)} \left| \sum_{k \in B(i) \cap B(j)} (\mathbf{r}_{ik} + \mathbf{r}_{jk}) \right|^2 . \tag{26}$$

The values of $\mathrm{CNP}(i)$ are calculated directly, by iterating over the neighbours of atom i in the first sum, and the common neighbours of atoms i and j in the second sum.

In the case of Voronoi analysis, it becomes necessary to determine, for every atom i in the system, its corresponding Voronoi polyhedron. The polyhedra are subsequently described through the use of signatures $(f_3, f_4, f_5, f_6, \ldots)$, where the symbol f_e denotes the number of faces with e edges.

We measured the time needed for performing structural identification for each of the methods: t_BOP, t_CSP, t_CNA, t_CNP, t_Voronoi. The time required for neighbour identification and the calculation of bond vectors was excluded. Measurements were performed for all thirteen test cases, on all three machines described earlier. Results were averaged over 10 independent runs. We appreciate that the obtained relative efficiencies of the methods are going to depend on implementation details. The results we show are valid for our implementations, which we diligently optimised.

The missing points in Fig. 8 correspond to scenarios where RAM limitations prevented us from performing the benchmark for the largest (1M+ atoms) systems. This is mostly seen on machine C (a laptop PC with merely 2 GB of RAM), or for Voronoi analysis (which has much higher memory requirements compared to CSP, CNA, CNP or BOP). The above reflects the simplified nature of our test environment, where all calculations have been done in post-processing, in a single-CPU environment. In practice structural analysis of 1M+-atom systems would be done in a distributed memory parallel environment.

The measured relative timings ($t_\mathrm{CSP}/t_\mathrm{BOP}$, $t_\mathrm{CNA}/t_\mathrm{BOP}$, $t_\mathrm{CNP}/t_\mathrm{BOP}$ and $t_\mathrm{Voronoi}/t_\mathrm{BOP}$) shown in Fig. 8 demonstrate that with the approach we propose, BOP analysis can be carried out in less time than Voronoi

analysis (by at least an order of magnitude), and, crucially, in less time than CNA analysis. In particular, depending on the test case and the machine used, we achieve a speed-up between 8.1 and 50.1 compared to (our implementation of) Voronoi analysis and between 1.1 and 10.6 compared to (our implementation of) CNA analysis. Crucially, our timings also indicate that the computational effort of the proposed approach for evaluating BOP is comparable to that of the CSP method (with ratios $t_{\mathrm{CSP}}/t_{\mathrm{BOP}}$ between 0.5 and 2.1), and the CNP method (with ratios $t_{\mathrm{CNP}}/t_{\mathrm{BOP}}$ between 1.0 and 4.4).

We can thus conclude that the computational effort of our approach to structure identification by BOP is significantly lower than that of Voronoi analysis and even lower than (or at least comparable to) that of CSP, CNP and CNA methods. We posit that BOP analysis should no longer be regarded as a more computationally involved method of structure characterisation.

5. Conclusions

We devised an efficient technique for evaluating bond-orientational order parameters (BOPs). Our approach combines simultaneous interpolation in the evaluation of spherical harmonics with rearrangements in the expressions for obtaining BOPs from spherical harmonics. Cache-friendly data structures are employed in the interpolation.

The memory complexity of our approach is linear in the grid size. The computational complexity of initialisation (constructing the interpolation grids) is linear, while the time of actual evaluation is independent of the grid size. For sufficiently large numbers of evaluations and grid sizes used in practical calculations the latter time dominates.

As one stage of our approach relies on interpolation, we carefully measured the errors in the approximation both for the calculated spherical harmonics and the resultant BOPs. The errors inherent to our approach are seen to be well-behaved and controllable, even with linear interpolation, vanishing according to $p^{-1.68}$ with the grid size p. The errors are found to be negligible already at $p = 9600$, which corresponds to a memory footprint of only 3.7 MB and an initialisation time below 200 ms.

Benchmarks for a number of realistic test cases performed on three hardware configurations demonstrate that our approach is between 11 and 46 times faster compared to other widely used approaches to evaluating BOPs, and can even outperform other methods of structure identification that are generally regarded as computationally cheaper.

An implementation of the approach has been made available under the GNU GPLv3 license (see Supplementary Materials).

Acknowledgements

We acknowledge the support of the Polish Ministry of Science and Higher Education (grant IP2012 043972) and of the TASK Academic Computer Centre (Gdańsk, Poland). This research was also supported in part by the PL-Grid Infrastructure (grant POIG.02.03.00-00-096/10).

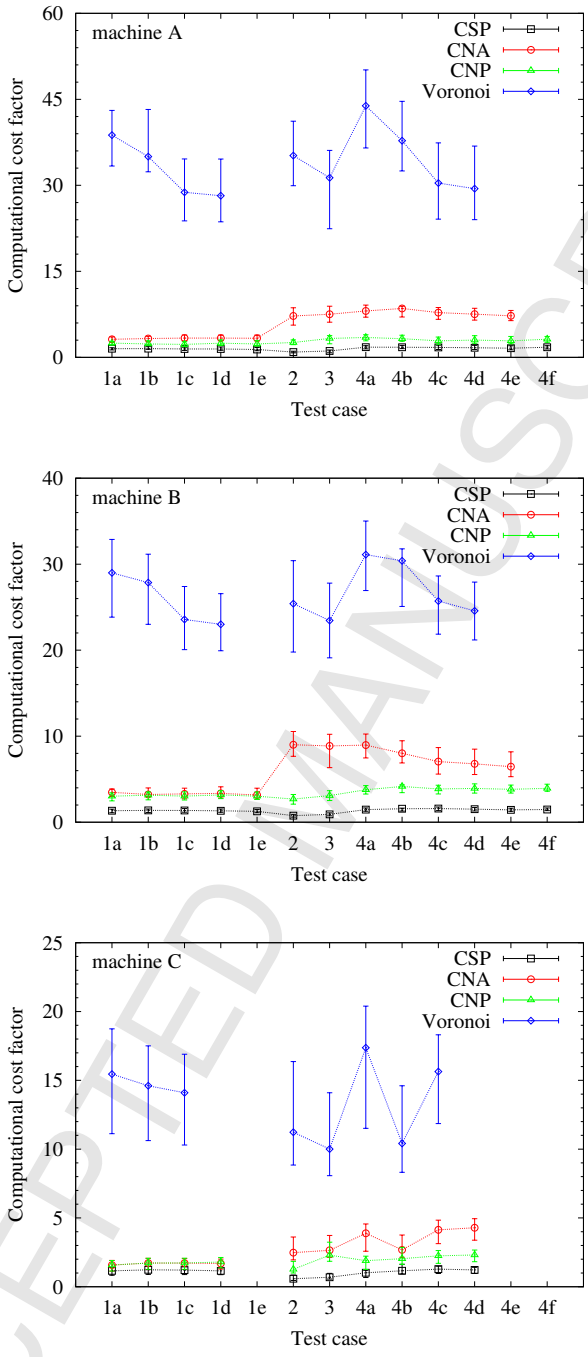


Figure 8: Computational cost factor of CSP, CNA, CNP and Voronoi methods *relative* to the proposed approach to evaluating BOPs, i.e. the ratio of the time needed to perform CSP, CNA, CNP or Voronoi analysis to the time needed to perform BOP analysis. The three panels correspond to machines (A: top, B: middle, C: bottom). The datapoints correspond to grid sizes of $p = q = 2400$, while the error bars correspond to $p = q = 600$ (coarsest grids, highest speedup) and $s = 9600$ (finest grids, lowest speedup). Test cases (cf. Table 5) are shown on the X axis. Cost factors for the largest systems (1de, 4def) were not calculated on every machine due to RAM limitations of the test environment (see text).

[1] P. J. Steinhardt, D. R. Nelson, M. Ronchetti, Icosahedral bond orientational order in supercooled liquids, Phys. Rev. Lett. 47 (1981) 1297–1300. doi:10.1103/PhysRevLett.47.1297.
URL http://link.aps.org/doi/10.1103/PhysRevLett.47.1297

[2] P. J. Steinhardt, D. R. Nelson, M. Ronchetti, Bond-orientational order in liquids and glasses, Phys. Rev. B 28 (1983) 784–805. doi:10.1103/PhysRevB.28.784.
URL http://link.aps.org/doi/10.1103/PhysRevB.28.784

[3] D. R. Nelson, B. I. Halperin, Dislocation-mediated melting in two dimensions, Phys. Rev. B 19 (1979) 2457–2484. doi:10.1103/PhysRevB.19.2457.
URL http://link.aps.org/doi/10.1103/PhysRevB.19.2457

[4] J. S. van Duijneveldt, D. Frenkel, Computer simulation study of free energy barriers in crystal nucleation, The Journal of Chemical Physics 96 (6) (1992) 4655–4668. doi:http://dx.doi.org/10.1063/1.462802.
URL http://scitation.aip.org/content/aip/journal/jcp/96/6/10.1063/1.462802

[5] P. R. ten Wolde, M. J. Ruiz-Montero, D. Frenkel, Numerical evidence for bcc ordering at the surface of a critical fcc nucleus, Phys. Rev. Lett. 75 (1995) 2714–2717. doi:10.1103/PhysRevLett.75.2714.
URL http://link.aps.org/doi/10.1103/PhysRevLett.75.2714

[6] I. Volkov, M. Cieplak, J. Koplik, J. R. Banavar, Molecular dynamics simulations of crystallization of hard spheres, Phys. Rev. E 66 (2002) 061401. doi:10.1103/PhysRevE.66.061401.
URL http://link.aps.org/doi/10.1103/PhysRevE.66.061401

[7] Y. Wang, S. Teitel, C. Dellago, Surface-driven bulk reorganization of gold nanorods, Nano Letters 5 (11) (2005) 2174–2178. arXiv:http://dx.doi.org/10.1021/nl051149h, doi:10.1021/nl051149h.
URL http://dx.doi.org/10.1021/nl051149h

[8] Y. Wang, S. Teitel, C. Dellago, Effect of surface structure on shape transformations of gold nanorods, Journal of Computational and Theoretical Nanoscience 4 (2) (2007-04-01T00:00:00) 282–290. doi:doi:10.1166/jctn.2007.013.
URL http://www.ingentaconnect.com/content/asp/jctn/2007/00000004/00000002/art00013

[9] W. Lechner, C. Dellago, Accurate determination of crystal structures based on averaged local bond order parameters, The Journal of Chemical Physics 129 (11) (2008) –. doi:http://dx.doi.org/10.1063/1.2977970.
URL http://scitation.aip.org/content/aip/journal/jcp/129/11/10.1063/1.2977970

[10] T. Kawasaki, H. Tanaka, Structural origin of dynamic heterogeneity in three-dimensional colloidal glass formers and its link to crystal nucleation, Journal of Physics: Condensed Matter 22 (23) (2010) 232102.
URL http://stacks.iop.org/0953-8984/22/i=23/a=232102

[11] R. Ni, M. Dijkstra, Crystal nucleation of colloidal hard dumbbells, The Journal of Chemical Physics 134 (3) (2011) 034501. doi:http://dx.doi.org/10.1063/1.3528222.
URL http://scitation.aip.org/content/aip/journal/jcp/134/3/10.1063/1.3528222

[12] P. Rein ten Wolde, D. Frenkel, Homogeneous nucleation and the ostwald step rule, Phys. Chem. Chem. Phys. 1 (1999) 2191–2196. doi:10.1039/A809346F.
URL http://dx.doi.org/10.1039/A809346F

[13] D. Moroni, P. R. ten Wolde, P. G. Bolhuis, Interplay between structure and size in a critical crystal nucleus, Phys. Rev. Lett. 94 (2005) 235703. doi:10.1103/PhysRevLett.94.235703.
URL http://link.aps.org/doi/10.1103/PhysRevLett.94.235703

[14] H. Wang, H. Gould, W. Klein, Homogeneous and heterogeneous nucleation of lennard-jones liquids, Phys. Rev. E 76 (2007) 031604. doi:10.1103/PhysRevE.76.031604.
URL http://link.aps.org/doi/10.1103/PhysRevE.76.031604

[15] C. Desgranges, J. Delhommelle, Crystallization mechanisms for supercooled liquid xe at high pressure and temperature: Hybrid monte carlo molecular simulations, Phys. Rev. B 77 (2008) 054201. doi:10.1103/PhysRevB.77.054201.
URL http://link.aps.org/doi/10.1103/PhysRevB.77.054201

[16] T. Schilling, H. J. Schöpe, M. Oettel, G. Opletal, I. Snook, Precursor-mediated crystallization process in suspensions of hard spheres, Phys. Rev. Lett. 105 (2010) 025701. doi:10.1103/PhysRevLett.105.025701.
URL http://link.aps.org/doi/10.1103/PhysRevLett.105.025701

[17] A. Ikeda, K. Miyazaki, Glass transition of the monodisperse gaussian core model, Phys. Rev. Lett. 106 (2011) 015701. doi:10.1103/PhysRevLett.106.015701.
URL http://link.aps.org/doi/10.1103/PhysRevLett.106.015701

[18] P. Rein ten Wolde, M. J. Ruiz-Montero, D. Frenkel, Numerical calculation of the rate of crystal nucleation in a lennard-jones system at moderate undercooling, The Journal of Chemical Physics 104 (24) (1996) 9932–9947. doi:http://dx.doi.org/10.1063/1.471721.
URL http://scitation.aip.org/content/aip/journal/jcp/104/24/10.1063/1.471721

[19] C. Chakravarty, R. M. Lynden-Bell, Landau free energy curves for melting of quantum solids, The Journal of Chemical Physics 113 (20) (2000) 9239–9247. doi:http://dx.doi.org/10.1063/1.1316105.
URL http://scitation.aip.org/content/aip/journal/jcp/113/20/10.1063/1.1316105

[20] C. Chakravarty, P. G. Debenedetti, F. H. Stillinger, Lindemann measures for the solid-liquid phase transition, The Journal of Chemical Physics 126 (20) (2007) 204508. doi:http://dx.doi.org/10.1063/1.2737054.
URL http://scitation.aip.org/content/aip/journal/jcp/126/20/10.1063/1.2737054

[21] R. Lynden-Bell, J. Van Duijneveldt, D. Frenkel, Free energy changes on freezing and melting ductile metals, Molecular Physics 80 (4) (1993) 801–814. arXiv:http://dx.doi.org/10.1080/00268979300102661, doi:10.1080/00268979300102661.
URL http://dx.doi.org/10.1080/00268979300102661

[22] S. E. Abraham, B. Bagchi, Suppression of the rate of growth of dynamic heterogeneities and its relation to the local

structure in a supercooled polydisperse liquid, Phys. Rev. E 78 (2008) 051501. doi:10.1103/PhysRevE.78.051501.
URL http://link.aps.org/doi/10.1103/PhysRevE.78.051501

[23] H. Tanaka, T. Kawasaki, H. Shintani, K. Watanabe, Critical-like behaviour of glass-forming liquids, Nature Materials 9 (4) (2010) 324–331. doi:10.1038/nmat2634.
URL http://dx.doi.org/10.1038/nmat2634

[24] A. Kozub, Study of icosahedral clusters in close-packed simple liquids, TASK Quarterly 16 (1) (2012) 75.

[25] C. R. Iacovella, A. S. Keys, M. A. Horsch, S. C. Glotzer, Icosahedral packing of polymer-tethered nanospheres and stabilization of the gyroid phase, Phys. Rev. E 75 (2007) 040801. doi:10.1103/PhysRevE.75.040801.
URL http://link.aps.org/doi/10.1103/PhysRevE.75.040801

[26] A. S. Keys, S. C. Glotzer, How do quasicrystals grow?, Phys. Rev. Lett. 99 (2007) 235503. doi:10.1103/PhysRevLett.99.235503.
URL http://link.aps.org/doi/10.1103/PhysRevLett.99.235503

[27] K. Lochmann, A. Anikeenko, A. Elsner, N. Medvedev, D. Stoyan, Statistical verification of crystallization in hard sphere packings under densification, The European Physical Journal B - Condensed Matter and Complex Systems 53 (1) (2006) 67–76. doi:10.1140/epjb/e2006-00348-9.
URL http://dx.doi.org/10.1140/epjb/e2006-00348-9

[28] A. Wouterse, A. P. Philipse, Geometrical cluster ensemble analysis of random sphere packings, The Journal of Chemical Physics 125 (19) (2006) 194709. doi:http://dx.doi.org/10.1063/1.2390700.
URL http://scitation.aip.org/content/aip/journal/jcp/125/19/10.1063/1.2390700

[29] W.-S. Xu, Z.-Y. Sun, L.-J. An, Dense packing in the monodisperse hard-sphere system: A numerical study, The European Physical Journal E 31 (4) (2010) 377–382. doi:10.1140/epje/i2010-10583-5.
URL http://dx.doi.org/10.1140/epje/i2010-10583-5

[30] P. Shah, C. Chakravarty, Comparison of inherent, instantaneous, and saddle configurations of the bulk lennard-jones system, The Journal of Chemical Physics 115 (19) (2001) 8784–8794. doi:http://dx.doi.org/10.1063/1.1413739.
URL http://scitation.aip.org/content/aip/journal/jcp/115/19/10.1063/1.1413739

[31] L.-C. Valdes, F. Affouard, M. Descamps, J. Habasaki, Mixing effects in glass-forming lennard-jones mixtures, The Journal of Chemical Physics 130 (15) (2009) 154505. doi:http://dx.doi.org/10.1063/1.3106759.
URL http://scitation.aip.org/content/aip/journal/jcp/130/15/10.1063/1.3106759

[32] F. Calvo, D. J. Wales, Stepwise melting of a model glass former under confinement, The Journal of Chemical Physics 131 (13) (2009) 134504. doi:http://dx.doi.org/10.1063/1.3239468.
URL http://scitation.aip.org/content/aip/journal/jcp/131/13/10.1063/1.3239468

[33] C. Chakravarty, Path integral simulations of quantum lennard-jones solids, The Journal of Chemical Physics 116 (20) (2002) 8938–8947. doi:http://dx.doi.org/10.1063/1.1471243.
URL http://scitation.aip.org/content/aip/journal/jcp/116/20/10.1063/1.1471243

[34] P. Shah, P. Chakrabarti, C. Chakravarty, Structure and melting of morse solids, Molecular Physics 99 (7) (2001) 573–583. arXiv:http://dx.doi.org/10.1080/00268970010018972, doi:10.1080/00268970010018972.
URL http://dx.doi.org/10.1080/00268970010018972

[35] Y. Wang, S. Teitel, C. Dellago, Melting and equilibrium shape of icosahedral gold nanoparticles, Chemical Physics Letters 394 (4-6) (2004) 257 – 261. doi:http://dx.doi.org/10.1016/j.cplett.2004.06.139.
URL http://www.sciencedirect.com/science/article/pii/S0009261404010267

[36] Y. Wang, S. Teitel, C. Dellago, Melting of icosahedral gold nanoclusters from molecular dynamics simulations, The Journal of Chemical Physics 122 (21) (2005) 214722. doi:http://dx.doi.org/10.1063/1.1917756.
URL http://scitation.aip.org/content/aip/journal/jcp/122/21/10.1063/1.1917756

[37] R. M. Lynden-Bell, D. J. Wales, Free energy barriers to melting in atomic clusters, The Journal of Chemical Physics 101 (2) (1994) 1460–1476. doi:http://dx.doi.org/10.1063/1.467771.
URL http://scitation.aip.org/content/aip/journal/jcp/101/2/10.1063/1.467771

[38] A. V. Mokshin, J.-L. Barrat, Shear induced structural ordering of a model metallic glass, The Journal of Chemical Physics 130 (3) (2009) 034502. doi:http://dx.doi.org/10.1063/1.3058433.
URL http://scitation.aip.org/content/aip/journal/jcp/130/3/10.1063/1.3058433

[39] N. Duff, D. J. Lacks, Shear-induced crystallization in jammed systems, Phys. Rev. E 75 (2007) 031501. doi:10.1103/PhysRevE.75.031501.
URL http://link.aps.org/doi/10.1103/PhysRevE.75.031501

[40] M. L. Wallace, B. Joós, Shear-induced overaging in a polymer glass, Phys. Rev. Lett. 96 (2006) 025501. doi:10.1103/PhysRevLett.96.025501.
URL http://link.aps.org/doi/10.1103/PhysRevLett.96.025501

[41] A. B. de Oliveira, P. A. Netz, T. Colla, M. C. Barbosa, Structural anomalies for a three dimensional isotropic core-softened potential, The Journal of Chemical Physics 125 (12) (2006) 124503. doi:http://dx.doi.org/10.1063/1.2357119.
URL http://scitation.aip.org/content/aip/journal/jcp/125/12/10.1063/1.2357119

[42] Z. Yan, S. V. Buldyrev, P. Kumar, N. Giovambattista, P. G. Debenedetti, H. E. Stanley, Structure of the first- and second-neighbor shells of simulated water: Quantitative relation to translational and orientational order, Phys. Rev. E 76 (2007) 051201. doi:10.1103/PhysRevE.76.051201.
URL http://link.aps.org/doi/10.1103/PhysRevE.76.051201

[43] B. Coasne, S. K. Jain, L. Naamar, K. E. Gubbins, Freezing of argon in ordered and disordered porous carbon, Phys. Rev. B 76 (2007) 085416. doi:10.1103/PhysRevB.76.085416.
URL http://link.aps.org/doi/10.1103/PhysRevB.76.085416

[44] S. Ogata, Monte carlo simulation study of crystallization in rapidly supercooled one-component plasmas, Phys. Rev. A 45 (1992) 1122–1134. doi:10.1103/PhysRevA.45.1122.
URL http://link.aps.org/doi/10.1103/PhysRevA.45.1122

[45] W. Mickel, S. C. Kapfer, G. E. Schroeder-Turk, K. Mecke, Shortcomings of the bond orientational order parameters for the analysis of disordered particulate matter, The Journal of Chemical Physics 138 (4) (2013) 044501. doi:http://dx.doi.org/10.1063/1.4774084.
URL http://scitation.aip.org/content/aip/journal/jcp/138/4/10.1063/1.4774084

[46] L. Landau, E. Lifschitz, Quantum Mechanics, Pergamon Press, 1965.

[47] A. Stukowski, Structure identification methods for atomistic simulations of crystalline materials, Modelling and Simulation in Materials Science and Engineering 20 (4) (2012) 045021.
URL http://stacks.iop.org/0965-0393/20/i=4/a=045021

[48] G. Voronoi, Nouvelles applications des parametres continus a la theorie des formes quadratiques, J. Reine Angew. Math. 134 (1908) 199.

[49] J. L. Finney, Random packings and the structure of simple liquids. i. the geometry of random close packing, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 319 (1539) (1970) 479–493. doi:10.1098/rspa.1970.0189.

[50] J. D. Honeycutt, H. C. Andersen, Molecular dynamics study of melting and freezing of small lennard-jones clusters, The Journal of Physical Chemistry 91 (19) (1987) 4950–4963. arXiv:http://dx.doi.org/10.1021/j100303a014, doi:10.1021/j100303a014.
URL http://dx.doi.org/10.1021/j100303a014

[51] C. L. Kelchner, S. J. Plimpton, J. C. Hamilton, Dislocation nucleation and defect structure during surface indentation, Phys. Rev. B 58 (1998) 11085–11088. doi:10.1103/PhysRevB.58.11085.
URL http://link.aps.org/doi/10.1103/PhysRevB.58.11085

[52] S. J. Zhou, D. L. Preston, P. S. Lomdahl, D. M. Beazley, Large-scale molecular dynamics simulations of dislocation intersection in copper, Science 279 (5356) (1998) 1525–1527. arXiv:http://www.sciencemag.org/content/279/5356/1525.full.pdf, doi:10.1126/science.279.5356.1525.
URL http://www.sciencemag.org/content/279/5356/1525.abstract

[53] M. J. Buehler, A. Hartmaier, H. Gao, M. A. Duchaineau, F. F. Abraham, The dynamical complexity of work-hardening: a large-scale molecular dynamics simulation, Acta Mechanica Sinica 21 (2) (2005) 103–111. doi:10.1007/s10409-005-0019-9.
URL http://dx.doi.org/10.1007/s10409-005-0019-9

[54] D. Tanguy, M. Mareschal, T. Germann, B. Holian, P. Lomdahl, R. Ravelo, Plasticity induced by a shock wave: large scale molecular dynamics simulations, Materials Science and Engineering: A 387-389 (0) (2004) 262 – 265, 13th International Conference on the Strength of Materials. doi:http://dx.doi.org/10.1016/j.msea.2004.02.088.
URL http://www.sciencedirect.com/science/article/pii/S0921509304006392

[55] S. Plimpton, Fast parallel algorithms for short-range molecular dynamics, Journal of Computational Physics 117 (1) (1995) 1 – 19. doi:http://dx.doi.org/10.1006/jcph.1995.1039.
URL http://www.sciencedirect.com/science/article/pii/S002199918571039X

[56] A. Stukowski, Visualization and analysis of atomistic simulation data with ovito-the open visualization tool, Modelling and Simulation in Materials Science and Engineering 18 (1) (2010) 015012.
URL http://stacks.iop.org/0965-0393/18/i=1/a=015012

[57] H. Tsuzuki, P. S. Branicio, J. P. Rino, Structural characterization of deformed crystals by analysis of common atomic neighborhood, Computer Physics Communications 177 (6) (2007) 518 – 523. doi:http://dx.doi.org/10.1016/j.cpc.2007.05.018.
URL http://www.sciencedirect.com/science/article/pii/S0010465507002536

[58] E. Wigner, Group theory and its application to the quantum mechanics of atomic spectra, Academic Press, 1959.

[59] A. Messiah, Quantum Mechanics, Dover Publications, 1961.

[60] R. Hockney, J. Eastwood, Computer simulations using particles, McGraw-Hill, 1981.

[61] M. Allen, D. Tildesley, Computer simulation of liquids, Oxford University Press, 1989.

[62] K. V. Tretiakov, K. W. Wojciechowski, Efficient monte carlo simulations using a shuffled nested weyl sequence random number generator, Phys. Rev. E 60 (1999) 7626–7628. doi:10.1103/PhysRevE.60.7626.
URL http://link.aps.org/doi/10.1103/PhysRevE.60.7626

[63] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, F. Rossi, GNU Scientific Library Reference Manual (2013).
URL http://www.gnu.org/software/gsl/

[64] C. H. Rycroft, Voro++: A three-dimensional voronoi cell library in c++, Chaos: An Interdisciplinary Journal of Nonlinear Science 19 (4). doi:http://dx.doi.org/10.1063/1.3215722.
URL http://scitation.aip.org/content/aip/journal/chaos/19/4/10.1063/1.3215722

Offset	0	1	2	3	4	5	6	7
Contents	$\widetilde{P}_4^0(x_i)$	$\Delta\widetilde{P}_4^0(x_i)$	$\widetilde{P}_6^0(x_i)$	$\Delta\widetilde{P}_6^0(x_i)$	$\widetilde{P}_4^1(x_i)$	$\Delta\widetilde{P}_4^1(x_i)$	$\widetilde{P}_6^1(x_i)$	$\Delta\widetilde{P}_6^1(x_i)$
Offset	8	9	10	11	12	13	14	15
Contents	$\widetilde{P}_4^2(x_i)$	$\Delta\widetilde{P}_4^2(x_i)$	$\widetilde{P}_6^2(x_i)$	$\Delta\widetilde{P}_6^2(x_i)$	$\widetilde{P}_4^3(x_i)$	$\Delta\widetilde{P}_4^3(x_i)$	$\widetilde{P}_6^3(x_i)$	$\Delta\widetilde{P}_6^3(x_i)$
Offset	16	17	18	19	20	21	22	23
Contents	$\widetilde{P}_4^4(x_i)$	$\Delta\widetilde{P}_4^4(x_i)$	$\widetilde{P}_6^4(x_i)$	$\Delta\widetilde{P}_6^4(x_i)$	$\widetilde{P}_6^5(x_i)$	$\Delta\widetilde{P}_6^5(x_i)$	$\widetilde{P}_6^6(x_i)$	$\Delta\widetilde{P}_6^6(x_i)$

Table 2: A cache-friendly ordering of values for a single NALP interpolation node.

Offset	0	1	2	3	4	5	6	7
Contents	$\cos(\phi_i)$	$\Delta\cos(\phi_i)$	$\sin(\phi_i)$	$\Delta\sin(\phi_i)$	$\cos(2\phi_i)$	$\Delta\cos(2\phi_i)$	$\sin(2\phi_i)$	$\Delta\sin(2\phi_i)$
Offset	8	9	10	11	12	13	14	15
Contents	$\cos(3\phi_i)$	$\Delta\cos(3\phi_i)$	$\sin(3\phi_i)$	$\Delta\sin(3\phi_i)$	$\cos(4\phi_i)$	$\Delta\cos(4\phi_i)$	$\sin(4\phi_i)$	$\Delta\sin(4\phi_i)$
Offset	16	17	18	19	20	21	22	23
Contents	$\cos(5\phi_i)$	$\Delta\cos(5\phi_i)$	$\sin(5\phi_i)$	$\Delta\sin(5\phi_i)$	$\cos(6\phi_i)$	$\Delta\cos(6\phi_i)$	$\sin(6\phi_i)$	$\Delta\sin(6\phi_i)$

Table 3: A cache-friendly ordering of values for a single trigonometric function interpolation node.

Machine	A	B	C
Processor type	Intel i7-3820	Intel Xeon L5640	Intel Atom 330
Clock rate	3.60 GHz	2.27 GHz	1.60 GHz
Level 1 cache			
instruction	4 × 32 KB	6 × 32 KB	2 × 32 KB
data	4 × 32 KB	6 × 32 KB	2 × 24 KB
Level 2 cache	2 × 256 KB	6 × 256 KB	2 × 512 KB
Level 3 cache	10 MB	12 MB	—
Memory	DDR3, 16 GB @1333MHz	DDR3, 16 GB @1333MHz	DDR2, 2 GB @667MHz

Table 4: Hardware details of the machines used in benchmarks.

Test case	System characteristics			
	Description	Size (atoms)	Coordination number	Cutoff radius r_c (Å)
1a	fcc Cu single-crystal at 300 K	500	12	3.07
1b		4000		
1c		32000		
1d		256000		
1e		2048000		
2	liquid Cu at 1900 K	500	12.16	3.54
3	liquid AlCu alloy at 1300 K	4000	12.73	3.77
4a	bcc Mo single-crystal at 300 K	250	14	3.5
4b		2000		
4c		16000		
4d		128000		
4e		1024000		
4f		8192000		

Table 5: Test cases used in error analysis and benchmarks.