Title

DiSCaMB: a software library for aspherical atom model X-ray scattering factor calculations with CPUs and GPUs

Permalink

https://escholarship.org/uc/item/7qp958sb

Journal

Journal of Applied Crystallography, 51(1)

ISSN

0021-8898

Authors

Chodkiewicz, Michał L
Migacz, Szymon
Rudnicki, Witold
et al.

Publication Date

2018-02-01

DOI

10.1107/s1600576717015825

Peer reviewed

JOURNAL OF APPLIED CRYSTALLOGRAPHY

ISSN 1600-5767

Received 4 September 2017
Accepted 30 October 2017

Edited by Th. Proffen, Oak Ridge National Laboratory, USA

Keywords: structure factors; refinement; GPU; multipole model.

DiSCaMB: a software library for aspherical atom model X-ray scattering factor calculations with CPUs and GPUs

Michał L. Chodkiewicz,[a]* Szymon Migacz,[b] Witold Rudnicki,[b,c]* Anna Makal,[a] Jarosław A. Kalinowski,[d] Nigel W. Moriarty,[d] Ralf W. Grosse-Kunstleve,[e] Pavel V. Afonine,[d] Paul D. Adams[d,f] and Paulina Maria Dominiak[a]*

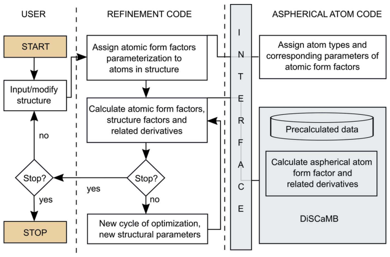
[a]Biological and Chemical Research Centre, Department of Chemistry, University of Warsaw, ulica Żwirki i Wigury 101, Warszawa, 02-089, Poland, [b]Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw, Warsaw, Poland, [c]Institute of Informatics, University of Białystok, Białystok, Poland, [d]Molecular Biophysics and Integrated Bioimaging Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA, [e]Physical Biosciences Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA, and [f]Department of Bioengineering, University of California at Berkeley, Berkeley, CA 94720, USA. *Correspondence e-mail: mchodkiewicz@chem.uw.edu.pl, w.rudnicki@icm.edu.pl, pdomin@chem.uw.edu.pl

It has been recently established that the accuracy of structural parameters from X-ray refinement of crystal structures can be improved by using a bank of aspherical pseudoatoms instead of the classical spherical model of atomic form factors. This comes, however, at the cost of increased complexity of the underlying calculations. In order to facilitate the adoption of this more advanced electron density model by the broader community of crystallographers, a new software implementation called *DiSCaMB*, 'densities in structural chemistry and molecular biology', has been developed. It addresses the challenge of providing for high performance on modern computing architectures. With parallelization options for both multi-core processors and graphics processing units (using CUDA), the library features calculation of X-ray scattering factors and their derivatives with respect to structural parameters, gives access to intermediate steps of the scattering factor calculations (thus allowing for experimentation with modifications of the underlying electron density model), and provides tools for basic structural crystallographic operations. Permissively (MIT) licensed, *DiSCaMB* is an open-source C++ library that can be embedded in both academic and commercial tools for X-ray structure refinement.

1. Introduction

The model of atomic electron density most widely applied in crystal structure refinement (the independent atom model, IAM) assumes spherical symmetry of the atomic density. It therefore cannot account for many effects resulting from the influence of the chemical environment on atomic density – *e.g.* chemical bonds, charge transfer, lone pairs, intermolecular interactions *etc.* – and the parameters obtained in the corresponding refinement are therefore subject to some systematic errors. Advances in measurements and computational techniques of X-ray diffraction have made most routine diffraction measurements on single crystals of small molecules (*i.e.* measurements of the resolution $d_{\min} \leq 5/6$ Å as recommended by the International Union of Crystallography; IUCr, 2012) sufficient to display the deficiencies of the IAM model.

More realistic models of the electron density have been proposed with refinable parameterization of the electron density (Hirshfeld, 1971; Stewart *et al.*, 1975; Hansen & Coppens, 1978). Owing to the large number of refined parameters (compared to IAM) such models usually require X-ray


USER
REFINEMENT CODE
ASPHERICAL ATOM CODE
START
Input/modify structure
Stop?
no
yes
STOP
Assign atomic form factors parameterization to atoms in structure
Calculate atomic form factors, structure factors and related derivatives
Stop?
yes
no
New cycle of optimization, new structural parameters
INTERFACE
Assign atom types and corresponding parameters of atomic form factors
Precalculated data
Calculate aspherical atom form factor and related derivatives
DiSCaMB

https://doi.org/10.1107/S1600576717015825

diffraction data of subatomic resolution ($d_{\min} \leq 0.5$ Å). However, even if no high-resolution data are available, the multipole model can still be applied using a so-called transferable aspherical atom model (TAAM). In this case parameters of atomic electron density are not refined but instead they are constrained to values typical for the corresponding atom type. Such parameters of atomic electron densities and algorithms for atom type assignment have been incorporated into databanks of pseudoatom parameters such as ELMAM (Pichon-Pesme *et al.*, 1995; Zarychta *et al.*, 2007; Domagała *et al.*, 2012), Invariom (Dittrich *et al.*, 2004, 2013; Dittrich, Hübschle *et al.*, 2006) and UBDB (Volkov *et al.*, 2004, 2007; Jarzembska & Dominiak, 2012). There exist a wealth of publications indicating that TAAM refinement gives superior results over IAM in terms of figures of merit, atomic positions and atomic displacement parameters (*e.g.* Jelsch *et al.*, 1998; Dittrich, Hübschle *et al.*, 2006; Dittrich, Munshi & Spackman, 2006; Volkov *et al.*, 2007; Dittrich *et al.*, 2008; Bąk *et al.*, 2011; Sanjuan-Szklarz *et al.*, 2016). In addition, TAAM provides access to quantitative estimation of the electron density and properties derived from it (dipole moments, electrostatic potentials *etc.*) for molecules and crystals.

In a similar way to the case of TAAM, the idea of transferability was applied also to constructing a databank of extremely localized molecular orbitals (ELMO; Meyer, Guillot, Ruiz-Lopez & Genoni, 2016; Meyer, Guillot, Ruiz-Lopez, Jelsch & Genoni, 2016). Application in crystallographic refinement is one of the intended uses of the databank, but no practical application for this purpose has been reported yet.

Recently, the Hirshfeld atom refinement procedure has been introduced. In this approach aspherical atom density is obtained by partitioning quantum mechanical electron density into atomic contributions (Jayatilaka & Dittrich, 2008). This method allows for very accurate determination of the hydrogen atom positions and atomic displacement parameters (Capelli *et al.*, 2014; Woińska *et al.*, 2016), yet it is computationally more demanding than the other methods.

The direct approach to structure factor calculation involves summation of contributions from each scattering center (an atom) at each reciprocal lattice point within a desired resolution limit. It has an approximately quadratic computational complexity with respect to the size of the system studied, resulting from the fact that both the number of atoms and the number of reciprocal lattice points within a fixed resolution limit scale approximately linearly with the unit-cell volume. With the major advantage of simplicity, direct summation is the method of choice in small-molecule crystallography, that is when computational cost is not a primary factor.

In the case of macromolecules, though, structure factor calculations using direct summation can be very time consuming. Fast Fourier transform based approaches with $N\log(N)$ scaling can remedy that problem. Unfortunately, these cannot be applied in a straightforward manner to the case of the Hansen–Coppens model as its implementation would require (not presently available) efficient implementation of a 'dynamical' atomic electron density calculation for the model. These are convolutions of the usual atomic density functions and atomic thermal displacement functions, which in the case of the Hansen–Coppens model involve, respectively, Slater and Gaussian radial functions. Alternative methodologies for dealing with aspherical atomic densities at scale have been introduced: polarizable Cartesian Gaussian multipoles (Schnieders *et al.*, 2009) and spherical interatomic scatterers (Afonine *et al.*, 2007). Nevertheless, it is still the Hansen–Coppens model that has been most extensively studied, in particular in macromolecular refinement, either with optimization of the multipolar parameters for ultra-high-resolution data (*e.g.* Jelsch *et al.*, 2000; Guillot *et al.*, 2008; Hirano *et al.*, 2016) or more commonly *via* TAAM refinement (*e.g.* Malinska & Dauter, 2016; Muzet *et al.*, 2003; Pröpper *et al.*, 2013; Schmidt *et al.*, 2003; Held & van Smaalen, 2014; Howard *et al.*, 2016). As TAAM seems to be a good candidate for replacing the IAM model in routine refinement for molecular crystals (Sanjuan-Szklarz *et al.*, 2016; Dittrich *et al.*, 2013), availability of software implementations becomes a prominent issue.

There are a few packages that provide multipolar/TAAM refinement, including *MoPro* (Jelsch *et al.*, 2005), *MOLLY* (Hansen & Coppens, 1978), *Jana2006* (Petrícek *et al.*, 2014) and *XD2016* (Volkov *et al.*, 2016). However, each of them is a highly specialized entity, requires very distinct data formats and, apart from *MoPro*, is hardly applicable to larger (macromolecular) cases. We therefore decided to build a new toolbox of software libraries, intended to facilitate integration of the aspherical atom model into a wide range of refinement programs commonly used in X-ray crystallography. To achieve this goal the software toolbox must provide easily embeddable components under a permissive license, and so we have used the MIT open-source license for the new *DiSCaMB* library described here. It is designed for use in structure refinement with X-ray diffraction data in both small-molecule and macromolecular crystallography. In the latter case, as $N\log(N)$ methods are currently not feasible for the Hansen–Coppens model, the implementation heavily exploits parallelization for both CPUs (central processing units – computer processors) and GPUs (graphics processing units – electronic circuits originally designed for graphics-related computations) to speed up calculations.

2. General considerations

One of the main goals of the development of the *DiSCaMB* library is to use it in software for X-ray structure refinement. Commonly, this kind of software needs to calculate both structure factors and related derivatives, and this is the task the library was intended for. To ensure general applicability of the library (reusability in multiple programs) it is crucial to define where and how it will fit into the structure common for X-ray refinement programs. The general structure of such a program and the way the library can be incorporated into it is illustrated in Fig. 1. The part specific to aspherical models of atomic electron density is shown as a separate component of the refinement program (ASPHERICAL ATOM CODE). Tasks related to this component involve assignment of the parameters of the aspherical atom model and calculation of structure factors and related derivatives. In the case of TAAM refinement the assignment of parameters corresponds to assignment of atom types and connecting the types with parameters stored in an aspherical atom bank. Atom type assignment is specific to a particular pseudoatom databank. In contrast, the structure factor calculation implemented in the library is universal and can be shared by all banks using the Hansen–Coppens model of electron density. The code responsible for communication between the aspherical atom code and refinement code is denoted in the diagram as INTERFACE. Ideally, it will be possible to replace code on one side of the interface without the need to change the code on the other side, *e.g.* easily exchange the 'aspherical atom code' with another code providing similar functionality. It is assumed that there are three main tasks in the refinement code related to multipolar model scattering factor calculations:

(1) Assignment of the multipolar model parameters to atoms.

(2) Transferring the parameters to the *DiSCaMB* library code – at this point some data which could be subsequently used in multiple structure factor calculations are precalculated and stored.

(3) Transferring current structural parameters to the *DiSCaMB* library code for calculation of structure factors and their derivatives.

The *DiSCaMB* library alone does not have the functionality required to perform the first task, since it is expected that the code specific to the pseudoatom databank can perform this task. It is, however, able to read multipolar model parameters from files in the XD format generated by *LSDB* (Volkov *et al.*, 2004; Jarzembska & Dominiak, 2012). The *DiSCaMB* library can be involved in the next two tasks, being responsible for storing multipolar model parameters, calculation and storage of additional data, and using the data in (multiple) structure factor calculations. In the case of gradient calculations in the third task, two groups of algorithms are provided. One of them typically would be used by small-molecule refinement programs which usually apply nonlinear least-squares procedures. The construction of the matrices used in the procedure can be realized using various strategies and can optionally involve direct inclusion of constraints (Bourhis *et al.*, 2015). All of the strategies require access to derivatives of structure factors with respect to structural parameters at a given scattering vector (**h**) – this functionality is provided by the library. Such derivatives are then processed in different ways depending on the particular implementation. In the case of macromolecules, structural parameters are usually optimized using gradient methods. Here, derivatives of the target function with respect to the real and imaginary components of the structure factor are expected to be provided by the refinement code. These are subsequently used for calculation of the derivatives of the target function with respect to structural parameters using the chain rule. This macromolecular version of the structure factor and gradient calculation was parallelized since computational performance can be very important in this case.

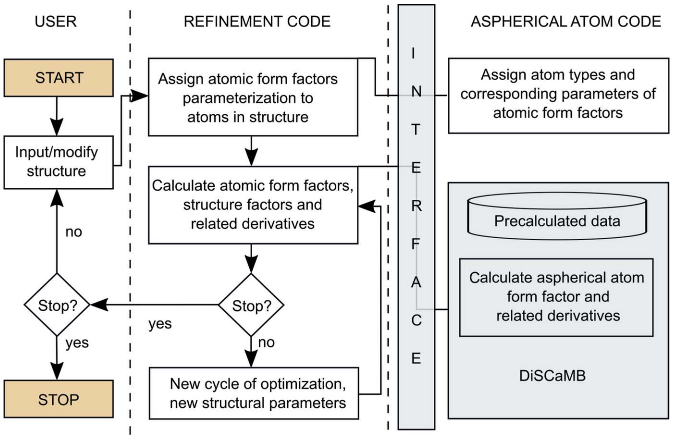

USER
REFINEMENT CODE
ASPHERICAL ATOM CODE
START
Input/modify structure
no
Stop?
yes
STOP
Assign atomic form factors parameterization to atoms in structure
Calculate atomic form factors, structure factors and related derivatives
Stop?
yes
no
New cycle of optimization, new structural parameters
INTERFACE
Assign atom types and corresponding parameters of atomic form factors
Precalculated data
Calculate aspherical atom form factor and related derivatives
DiSCaMB

Figure 1
Structure of an X-ray refinement program using a separate module for the aspherical atom model calculation with pseudoatom databank parameterization.

3. Multipolar model structure factors – theory and implementation

The expression for the aspherical atom model structure factor calculation is similar to that for the IAM model with the difference that the asphericity of atomic form factors has to be taken into account (see Appendix *B*). The general strategy of calculation is also analogous [see *e.g.* Bourhis *et al.* (2015) for details of the IAM model]. Here we briefly describe the atomic form factor calculations. Atomic electron density in the Hansen–Coppens multipolar model (Hansen & Coppens, 1978; Coppens, 1997) can be expressed as a linear combination of terms [$\rho_t(\mathbf{r})$] of the following type:

$$\rho_t(\mathbf{r}) = R(r)d_{lm}(\theta,\varphi) = R(r)d_{lm}(\hat{\mathbf{r}}). \qquad (1)$$

Here, $d_{lm}(\theta,\varphi)$ is a density normalized spherical harmonic function (see Appendix *B*), a vector with a circumflex ('hat') represents a versor (*i.e.* $\hat{\mathbf{r}} = \mathbf{r}/|\mathbf{r}|$) and $R(r)$ is a Slater-type radial function:

$$R(r;n,\xi) = r^n\exp(-\xi r). \qquad (2)$$

The corresponding scattering factor for the term $R(r;n,\xi)d_{lm}(\hat{\mathbf{r}})$ is given as

$$f_t(\mathbf{h};n,\xi,l,m) = 4\pi i^l g(h,n,\xi,l)\,d_{lm}(\hat{\mathbf{h}}). \qquad (3)$$

The function g is the three-dimensional Fourier–Bessel transform of the radial function:

$$g(h,n,\xi,l) = \int\limits_0^\infty r^n\exp(-\xi r)\,j_l(2\pi hr)\,r^2\,\mathrm{d}r, \qquad (4)$$

where j_l is the lth-order spherical Bessel function. Analytical formulas for the g functions for given l and n have been tabularized (Avery & Watson, 1977; Su & Coppens, 1990). They can also be obtained recursively (Avery & Watson, 1977; Deutsch, 1993) or evaluated from general analytical expressions (Restori, 1990). *DiSCaMB* provides functionality for calculation of both functions, g and spherical harmonics (for

$l \leq 4$), using tabularized formulas. This is sufficient for straightforward calculation of the terms [equation (3)] contributing to atomic form factors, but to calculate the form factor itself one must also find the values of the coefficients for these terms. For this purpose it is convenient to refer to the pseudoatom electron density in the Hansen–Coppens model in its usual representation and explicitly express its dependence on the model parameters:

$$\begin{aligned}&\rho_{\rm at}\big(\mathbf{r};\{\mathbf{p}_{\phi,i}\},\mathbf{o}_{\rm c},\mathbf{o}_{\rm v},\kappa,\kappa',P_{\rm val},\{n_l\},\xi,\{P_{lmp}\},\mathbf{M}\big)\\&\quad=P_{\rm core}\rho_{\rm core}\big(r;\{\mathbf{p}_{\phi,i}\},\mathbf{o}_{\rm c}\big)+\kappa^3P_{\rm val}\rho_{\rm val}\big(\kappa r;\{\mathbf{p}_{\phi,i}\},\mathbf{o}_{\rm v}\big)\\&\quad\quad+\sum_{l=0}^{l_{\max}}\kappa'^3N(n_l,\xi)R_l(\kappa' r;n_l,\xi)\sum_{m=-l}^{l}P_{lm}d_{lm}(\mathbf{M}\hat{\mathbf{r}}),\end{aligned}\qquad(5)$$

where $\rho_{\rm core}$ and $\rho_{\rm val}$ are the spherically averaged free-atom core and valence, respectively, theoretically derived electron densities, normalized to one electron. These two terms evaluate to a linear combination of Slater *s*-type functions:

$$\begin{aligned}\rho_{\rm core}\big(r;\{\mathbf{p}_{\phi,i}\},\mathbf{o}_{\rm c}\big)&=N\hat{S}\sum_i o_{{\rm c},i}\big|\phi_i(r)\big|^2\\&=\sum_k c_{{\rm c},k}r^{n_{{\rm c},k}}\exp(-\alpha_{{\rm c},k}rd_{00})\end{aligned}\qquad(6)$$

(analogously for $\rho_{\rm val}$), where $o_{{\rm c},i}$ are occupation factors for the core orbitals, $\{\mathbf{p}_{\phi,i}\}$ are parameters of orbitals $\{\phi_i\}$ of the isolated reference atom, N is a normalization factor and $\hat{S}$ symbolizes spherical symmetrization (see Appendix *B*). The occupancy factors can be either user defined or set to default values corresponding to filling the lowest-energy orbitals. The parameters $c_{{\rm c},k}$, $n_{{\rm c},k}$ and $\alpha_{{\rm c},k}$ depend on the parameters $\{\mathbf{p}_{\phi,i}\}$ of the orbitals $\{\phi_i\}$ and their occupancies $\mathbf{o}_{\rm c}$ (for the sake of clarity this dependence is not explicitly shown in the text). There are several sets of Slater-type atomic wavefunctions (Clementi & Roetti, 1974; Bunge *et al.*, 1993; Su & Coppens, 1998; Macchi & Coppens, 2001; Volkov & Macchi, unpublished work) in use for $\rho_{\rm core}$ and $\rho_{\rm val}$ parameterization. Any of them can, in principle, be used with the *DiSCaMB* library, and Clementi & Roetti wavefunctions data are provided (in the case of a hydrogen atom an analytical wavefunction is used).

The last term in equation (5) represents the deformation valence electron density $\{N(n_l,\xi)$ being normalization factors given by $[\int_0^\infty R_l(r;n_l,\xi_l)r^2\,{\rm d}r]^{-1}\}$. The values of the parameters ξ and n_l are predefined; their default values for a specific chemical element in a given electronic configuration may vary from program to program. The *DiSCaMB* code accepts either user-provided parameters or those used in UBDB (Jarzembska & Dominiak, 2012). Parameter ξ can, in principle, be l dependent, but the Hansen–Coppens model is mostly used in an l-independent fashion, as also coded in *DiSCaMB*. The matrix $\mathbf{M}$ transforms the position vector into the local coordinate system associated with the atom, and r is the distance from the nucleus. *DiSCaMB* provides tools for defining local coordinate systems in a similar manner to the one implemented in the *XD* package (Volkov *et al.*, 2016). The populations $P_{\rm val}$ and P_{lm}, and the dimensionless expansion–contraction parameters κ and κ', can be, in the Hansen–Coppens model, refined against experimental data. However, they are kept fixed during TAAM refinement, and for that reason *DiSCaMB* does not currently provide functionality for the calculation of derivatives with respect to these parameters. Therefore, it cannot currently be directly used for all-parameter multipolar refinement. A naïve TAAM refinement implementation is included in the *DiSCaMB* distribution as an example.

The atomic form factor corresponding to the pseudoatom electron density [equation (5)] is given as

$$\begin{aligned}&f_{\rm at}\big(\mathbf{h};\{\mathbf{p}_{\phi,i}\},\mathbf{o}_{\rm c},\mathbf{o}_{\rm v},\kappa,\kappa',P_{\rm val},\{n_l\},\xi,\{P_{lmp}\},\mathbf{M}\big)\\&\quad=P_{\rm core}f_{\rm core}\big(h;\{\mathbf{p}_{\phi,i}\},\mathbf{o}_{\rm c}\big)+P_{\rm val}f_{\rm val}\big(h/\kappa;\{\mathbf{p}_{\phi,i}\},\mathbf{o}_{\rm v}\big)\\&\quad\quad+f_{\rm dval}\big(\mathbf{Mh}/\kappa';\{P_{lm}\},\{n_l\},\xi\big),\end{aligned}\qquad(7)$$

where $f_{\rm core}$ and $f_{\rm val}$ are terms corresponding to the scattering from the core and the spherically averaged valence electron densities of the following type:

$$f_{\rm core}(h;\{\mathbf{p}_{\phi,i}\},\mathbf{o}_{\rm c})=\sum_k c_{{\rm c},k}\,g(h,n_{{\rm c},k},\alpha_{{\rm c},k},0).\qquad(8)$$

The deformation valence electron density contribution to scattering is given by

$$\begin{aligned}f_{\rm dval}\big(\mathbf{h};\kappa',\{P_{lm}\},\{n_l\},\xi\big)&=\sum_l 4\pi i^lN(n_l,\xi)\,g(h,l,n_l,\xi)\\&\quad\times\sum_m P_{lm}d_{lm}(\hat{\mathbf{h}}).\end{aligned}\qquad(9)$$

DiSCaMB exposes functionality allowing for performing the intermediate steps in the calculation of scattering factors, including those expressed in equations (3)–(9). This, in principle, may be useful for the construction of an alternative to the Hansen–Coppens model. *DiSCaMB* makes use of the notion of pseudoatom types. Atoms of the same type have the same values of $f_{\rm core}(h)$, $f_{\rm val}(h)$ and prefactors of the sum over m in $f_{\rm dval}$ – this feature is employed to speed up calculations. A common approach in the $f_{\rm core}(h)$ and $f_{\rm val}(h)$ calculation is to precalculate their values at a relatively small set of points and later use these data to interpolate values of $f_{\rm core}(h)$ and $f_{\rm val}(h)$. This approach was not coded in *DiSCaMB*, as we observed that the use of atom types provides for efficient calculation times in tests for large molecules.

4. Speeding up calculations with parallel computing on CPUs and GPUs

In recent years computers capable of performing parallel calculations have become very common because of the widespread use of multi-core processors in new computers and application of GPUs in general purpose programming (GPGPU). The *DiSCaMB* library provides an implementation of parallel code for calculation of the structure factors and related derivatives with both CPUs and GPUs. In terms of parallel computing the structure factor calculation may be classified as an embarrassingly parallel problem, which means that it is separable into a number of mostly independent tasks. Parallelization is performed by splitting the input set of $\mathbf{h}$ vectors into subsets and performing calculations for each subset as separate parallel tasks. The results from each task

Table 1
Performance test – the wall time of a single calculation of structure factors and their derivatives at 5/6 Å resolution.

Structure ID code†	QANQIG	3p4j	1ejg	4q4g	1f8b
Space group	$P2_12_12_1$	$P2_12_12_1$	$P2_1$	$P2_12_12_1$	$I432$
No. of atoms	169	586	942	3461	5916
No. of atom types‡	20	42	40	62	54
No. of reflections§	6490	23837	31597	153346	458685
Relative computational cost¶	4.4	55.9	59.5	2122.9	65125.9
Execution wall time					
Time units	ms	s	s	s	min:s
Intel Xeon E5-2697 v3					
1 core	515	8.7	10.1	310	105:21
4 cores	126	2.18	2.57	78	26:24
14 cores	37	0.63	0.74	22	7:43
NVIDIA Tesla K20	100	0.45	1.28	10.6	3:59
NVIDIA Tesla K80	60	0.3	2.97	7.3	2:53

† The CSD refcode in the case of QANQIG and the PDB ID in the other cases. QANQIG – 10 residue peptide (Aravinda *et al.*, 2004); 3p4j – Z-DNA hexamer duplex d(CGCGCG) (2) (Brzezinski *et al.*, 2011); 1ejg – crambin (Jelsch *et al.*, 2000); 4q4g – peptidoglycan endopeptidase RipA (Squeglia *et al.*, 2014); 1f8b – complex of native influenza virus neuraminidase (carbohydrate chains and solvent water molecules removed; Smith *et al.*, 2001). ‡ Structures were parametrized with the use of the UBDB bank (Jarzembska & Dominiak, 2012) and the *LSDB* program (Volkov *et al.*, 2004) supported by manual interventions in the case of disordered parts. § The lists of reflection indices were generated with the help of *cctbx* (Grosse-Kunstleve *et al.*, 2002). ¶ Equal to No. of symmetry operations used in calculations times No. of reflections times No. of atoms (in millions).

are then collected (in the case of the structure factors) and summed (in the case of the derivatives). The strategy for parallelization is more involved in the case of the code for GPUs, as described in Appendix *C*. The parallelization was performed for shared memory systems (*e.g.* single computer node) using OpenMP (http://www.openMP.org) for CPUs and CUDA (created by Nvidia, for Nvidia GPUs, http://nvidia.com/cuda) for GPUs.

The computational efficiency of the code was tested on structures ranging from a ten-residue peptide to a relatively large protein (~380 amino acids) (see Table 1). The time necessary for a single calculation of the structure factors and their derivatives at the 5/6 Å resolution limit with a single CPU core varied from 0.5 s to about 2 h. Fortunately, it can be efficiently reduced by running the calculations in parallel. The test shows a good scaling efficiency (E) of the code: $E = T_1/(NT_N)$, where T_k is execution time using k cores and N is the total number of cores. E exceeds 0.97 for all the cases in the table, which translates to a reduction of the execution time nearly by the number of processor cores used. This also suggests that the problem is well suited for further parallelization for distributed memory systems (*e.g.* multi-node machines like computational clusters). The use of a GPU allowed for a further reduction of the computational time compared to a 14 core processor by a factor of about 2–3 in the case of the largest systems studied.

5. Summary and outlook

We have presented the open-source C++ library *DiSCaMB* for the calculation of X-ray scattering factors from the Hansen–Coppens multipolar model of crystal electron density. The library is intended to be used in structure refinement with fixed multipolar parameters – *e.g.* refinement using a pseudoatom databank parameterization. It also provides access to intermediate steps in the calculations, allowing for modifications of the underlying electron density model. The library is designed for both small and large molecules. In the latter case the code is parallelized for both CPUs and GPUs. Performance tests show good scalability.

Future development plans include completing the aspherical atom code part in Fig. 1 by providing functionality that performs assignment of atom types and the corresponding atomic form factor parameters. Other planned developments include writing code for structure factor calculation with arbitrary externally provided aspherical atomic form factors (this would facilitate, for example, the inclusion of Hirshfeld atom partition based form factors in the refinement code). We also plan to speed up the calculations by further parallelization (targeting distributed memory systems and the utilization of the vector processing capabilities of modern CPUs) and development of algorithms for optimized calculations of the structure factors.

APPENDIX *A*
Summary of features

Main features. Calculation of multipolar structure factors and their derivatives with respect to structural parameters (position, occupancy and atomic displacement parameters) – various versions suitable for large and small molecules. Parallelized for use with multi-core processors and graphics processing units. Easy access to computational steps required to calculate the structure factors. Algorithms aware of pseudoatom types.

Structure factor calculations. Access to atomic wavefunction data. Conversion of the atomic wavefunction into corresponding spherically averaged density (linear combination of Slaters) for a given electronic configuration. Calculation of the multipolar model form factor with possibility for separate calculation of its individual components. Calculation of IAM form factors and corresponding structure factors [with Waasmaier and Kirfel parameterization (Waasmaier & Kirfel, 1995) and H-atom parameters taken from *cctbx* (Grosse-Kunstleve *et al.*, 2002)]. Local coordinate system calculation.

Structural data handling. Symmetry operations – multiplication, application to vectors, string and matrix notation. Conversions between fractional and Cartesian coordinate systems in direct and reciprocal space, conversions of atomic displacement parameters and positional parameter derivatives.

Utilities. Mathematical utilities – basic three-dimensional algebra and calculations involving spherical harmonics. Utilities for string operations, error handling, performance measurement. Reading Hansen–Coppens model parameters from *XD* input files generated by *LSDB* (Volkov *et al.*, 2004; Jarzembska & Dominiak, 2012).

Project organization. Build process managed with *CMake* (Martin & Hoffman, 2015) for enabling compilation environment of user choice. Code documentation includes *Doxygen*-generated (van Heesch, 2016) documentation. Extensive test set [testing against data generated with *XD* (Volkov *et al.*, 2016) and *cctbx* (Grosse-Kunstleve *et al.*, 2002; Gildea *et al.*, 2011)]. Examples illustrating usage including very basic implementation of refinement with BFGS method optimizer as implemented in *ALGLIB* (Bochkanov, 2017).

Availability. *DiSCaMB* is released under the MIT. The source code is freely available at the project web site, http://crystal.chem.uw.edu.pl.

APPENDIX *B*
Additional information on multipolar structure factor calculation

The expression for the aspherical atom model structure factor can be written in the following way:

$$F(\mathbf{h}) = \sum_{a\in \text{asu}} \text{occ}_a\, m_a \sum_{\{\mathbf{R}_k|\mathbf{t}_k\}} f_\text{a}(\mathbf{R}_k^\text{T}\mathbf{h})\, T(\mathbf{R}_k^\text{T}\mathbf{h}) \exp[2\pi i\mathbf{h}^\text{T}(\mathbf{R}_k\mathbf{r}_a + \mathbf{t}_k)]. \tag{10}$$

The first summation runs over atoms in the asymmetric unit, occ_a is the atomic occupancy and m_a is the multiplicity. The second summation runs over the symmetry operations ($\{\mathbf{R}_k|\mathbf{t}_k\}$) which would generate all symmetry-equivalent atoms in the unit cell for an atom in a general position. $T(\mathbf{h})$ stands for the temperature factor and $f_\text{a}(\mathbf{h})$ for the aspherical form factor.

It is a commonly applied convention in the Hansen–Coppens model to use density-normalized real spherical harmonics indexed with three indices $[\int|d_{lmp}(\theta,\varphi)|\,\text{d}S = (2-\delta_{0l})]$, with the p index being either + or − for $m > 0$ and no p index for $m = 0$. Then the sum over m in the deformation valence part is usually written as $\sum_{m=0}^{l} P_{lm\pm} d_{lm\pm}(\theta,\varphi)$. We have used the equivalent two-index notation $[d_{lm}(\theta,\varphi)]$ with the sign index (p) being directly incorporated into the m index, which now takes also negative values. Formulas for $d_{lm}(\theta,\varphi)$ have been calculated and tabularized (Hansen & Coppens, 1978; Paturle & Coppens, 1988; Coppens, 1997; Michael & Volkov, 2015).

Free-atom core and valence electron densities are combinations of orbital-related electron densities. Such orbitals in a basis of Slater-type functions take the following form:

$$\varphi_{lm}(\mathbf{r}) = Y_{lm}(\theta,\varphi)\sum_k b_k N_k r^{p_k}\exp(-\beta_k|\mathbf{r}|) = Y_{lm}(\theta,\varphi)R(r), \tag{11}$$

where $Y_{lm}(\theta,\varphi)$ are spherical harmonics. Spherical symmeterization of the corresponding electron density consists of its projection onto the space of the fully symmetric function defined on the surface of a sphere, *i.e.* the space spanned by spherical harmonic $Y_{00} = (2\pi^{1/2})^{-1}$:

$$\begin{aligned}\hat{S}[\phi_{lm}(\mathbf{r})\phi^*_{lm}(\mathbf{r})] &= |Y_{00}\rangle\langle Y_{00}|\phi_{lm}(\mathbf{r})\phi^*_{lm}\rangle \\ &= (4\pi)^{-1}\langle Y_{lm}|Y_{lm}\rangle R^2(r) = R^2(r)/4\pi. \end{aligned}\tag{12}$$

APPENDIX *C*
Details of the implementation for GPU

The GPU implementation is parallelized over reflections (**h** vectors). Each CUDA thread gets one reflection and computes the inner loops over atoms and symmetry operations. At the end of computation each thread stores its own structure factor F into global memory.

Computation of derivatives (occupancy derivative, three positional derivatives and six atomic displacement parameter derivatives) requires a reduction over all GPU threads to sum the contributions from all reflections (threads). This reduction could be handled in many different ways; for example, we could store all partial results from each thread into global memory and perform the reduction in a separate kernel. This solution requires a lot of global memory and a lot of global memory bandwidth.

A second straightforward approach is to divide this reduction into two steps: first perform partial reduction in each GPU block using shared memory and after that produce the final result for each atom in a separate kernel, adding up all the partial contributions from the blocks. This solution forces us to use synchronization [`__syncthreads()` function] inside the loop that performs the main computation, which significantly slows the execution. In our implementation we have chosen a different solution which is in between the two approaches described above. We perform a partial reduction for each warp (currently one warp is 32 consecutive threads). This reduction is done in registers using the warp shuffle instructions available in the NVIDIA Kepler architecture. After the reduction only one thread from each warp stores the partial result in global memory. This significantly reduces (32 times) the quantity of data that need to be transferred to global memory and we are not required to synchronize threads within blocks. The final reduction (adding up the contributions from each warp) is performed in a separate kernel and takes approximately 0.5% of the total execution time.

Coefficients P_{lm} for each atom type are stored in the constant memory of a GPU to improve the performance and reduce the required global memory bandwidth. The limited capacity of constant memory (64 kB) allows us to store the required coefficients for 250 atom types.

Acknowledgements

Professor Philip Coppens (University at Buffalo) is acknowledged for his invaluable contribution to the project in its early stages and Dr Dominik Gront (University of Warsaw) for his support in the GPU part of the project. This work has utilized computing resources at the PL-Grid Infrastructure and at the University of Białystok in Poland.

Funding information

This research was supported by the Foundation of Polish Science through grant No. POMOST/2010-2/3 co-financed from the Regional Development Fund of the European Union; in part by the NIH (project 1P01 GM063210 to PDA), a Laboratory Directed Research and Development Seed Grant from LBNL (to PDA), by the US Department of Energy under contract No. DE-AC02-05CH11231, and by the Polish National Center for Science (DEC-2012/04/A/ST5/00609).

References

Afonine, P. V., Grosse-Kunstleve, R. W., Adams, P. D., Lunin, V. Y. & Urzhumtsev, A. (2007). *Acta Cryst.* D**63**, 1194–1197.

Aravinda, S., Datta, S., Shamala, N. & Balaram, P. (2004). *Angew. Chem. Int. Ed.* **43**, 6728–6731.

Avery, J. & Watson, K. J. (1977). *Acta Cryst.* A**33**, 679–680.

Bąk, J. M., Domagała, S., Hübschle, C., Jelsch, C., Dittrich, B. & Dominiak, P. M. (2011). *Acta Cryst.* A**67**, 141–153.

Bochkanov, S. (2017). *ALGLIB*, http://www.alglib.net.

Bourhis, L. J., Dolomanov, O. V., Gildea, R. J., Howard, J. A. K. & Puschmann, H. (2015). *Acta Cryst.* A**71**, 59–75.

Brzezinski, K. B. A., Brzuszkiewicz, A., Dauter, M., Kubicki, M., Jaskolski, M. & Dauter, Z. (2011). *Nucleic Acids Res.* **39**, 6238–6248.

Bunge, C. F., Barrientos, J. A. & Bunge, A. V. (1993). *At. Data Nucl. Data Tables*, **53**, 113–162.

Capelli, S. C., Bürgi, H.-B., Dittrich, B., Grabowsky, S. & Jayatilaka, D. (2014). *IUCrJ*, **1**, 361–379.

Clementi, E. & Roetti, C. (1974). *At. Data Nucl. Data Tables*, **14**, 177–478.

Coppens, P. (1997). *X-ray Charge Densities and Chemical Bonding.* Oxford: International Union of Crystallography/Oxford University Press.

Deutsch, M. (1993). *J. Appl. Cryst.* **26**, 683–686.

Dittrich, B., Hübschle, C. B., Luger, P. & Spackman, M. A. (2006). *Acta Cryst.* D**62**, 1325–1335.

Dittrich, B., Hübschle, C. B., Pröpper, K., Dietrich, F., Stolper, T. & Holstein, J. J. (2013). *Acta Cryst.* B**69**, 91–104.

Dittrich, B., Koritsánszky, T. & Luger, P. (2004). *Angew. Chem. Int. Ed.* **43**, 2718–2721.

Dittrich, B., McKinnon, J. J. & Warren, J. E. (2008). *Acta Cryst.* B**64**, 750–759.

Dittrich, B., Munshi, P. & Spackman, M. A. (2006). *Acta Cryst.* C**62**, o633–o635.

Domagała, S., Fournier, B., Liebschner, D., Guillot, B. & Jelsch, C. (2012). *Acta Cryst.* A**68**, 337–351.

Gildea, R. J., Bourhis, L. J., Dolomanov, O. V., Grosse-Kunstleve, R. W., Puschmann, H., Adams, P. D. & Howard, J. A. K. (2011). *J. Appl. Cryst.* **44**, 1259–1263.

Grosse-Kunstleve, R. W., Sauter, N. K., Moriarty, N. W. & Adams, P. D. (2002). *J. Appl. Cryst.* **35**, 126–136.

Guillot, B., Jelsch, C., Podjarny, A. & Lecomte, C. (2008). *Acta Cryst.* D**64**, 567–588.

Hansen, N. K. & Coppens, P. (1978). *Acta Cryst.* A**34**, 909–921.

Heesch, D. van (2016). *Doxygen*, http://www.doxygen.org.

Held, J. & van Smaalen, S. (2014). *Acta Cryst.* D**70**, 1136–1146.

Hirano, Y., Takeda, K. & Miki, K. (2016). *Nature*, **534**, 281–284.

Hirshfeld, F. L. (1971). *Acta Cryst.* B**27**, 769–781.

Howard, E. I., Guillot, B., Blakeley, M. P., Haertlein, M., Moulin, M., Mitschler, A., Cousido-Siah, A., Fadel, F., Valsecchi, W. M., Tomizaki, T., Petrova, T., Claudot, J. & Podjarny, A. (2016). *IUCrJ*, **3**, 115–126.

International Union of Crystallography (2012). *Acta Cryst.* C**68**, e3–e11.

Jarzembska, K. N. & Dominiak, P. M. (2012). *Acta Cryst.* A**68**, 139–147.

Jayatilaka, D. & Dittrich, B. (2008). *Acta Cryst.* A**64**, 383–393.

Jelsch, C., Guillot, B., Lagoutte, A. & Lecomte, C. (2005). *J. Appl. Cryst.* **38**, 38–54.

Jelsch, C., Pichon-Pesme, V., Lecomte, C. & Aubry, A. (1998). *Acta Cryst.* D**54**, 1306–1318.

Jelsch, C., Teeter, M. M., Lamzin, V., Pichon-Pesme, V., Blessing, R. H. & Lecomte, C. (2000). *Proc. Natl Acad. Sci. USA*, **97**, 3171–3176.

Macchi, P. & Coppens, P. (2001). *Acta Cryst.* A**57**, 656–662.

Malinska, M. & Dauter, Z. (2016). *Acta Cryst.* D**72**, 770–779.

Martin, K. & Hoffman, B. (2015). *Mastering CMake: A Cross-Platform Build System.* New York: Kitware.

Meyer, B., Guillot, B., Ruiz-Lopez, M. F. & Genoni, A. (2016). *J. Chem. Theory Comput.* **12**, 1052–1067.

Meyer, B., Guillot, B., Ruiz-Lopez, M. F., Jelsch, C. & Genoni, A. (2016). *J. Chem. Theory Comput.* **12**, 1068–1081.

Michael, J. R. & Volkov, A. (2015). *Acta Cryst.* A**71**, 245–249.

Muzet, N., Guillot, B., Jelsch, C., Howard, E. & Lecomte, C. (2003). *Proc. Natl Acad. Sci. USA*, **100**, 8742–8747.

Paturle, A. & Coppens, P. (1988). *Acta Cryst.* A**44**, 6–7.

Petrícek, V., Dusek, M. & Palatinus, L. (2014). *Z. Kristallogr.* **229**, 345–352.

Pichon-Pesme, V., Lecomte, C. & Lachekar, H. (1995). *J. Phys. Chem.* **99**, 6242–6250.

Pröpper, K., Holstein, J. J., Hübschle, C. B., Bond, C. S. & Dittrich, B. (2013). *Acta Cryst.* D**69**, 1530–1539.

Restori, R. (1990). *Acta Cryst.* A**46**, 150–151.

Sanjuan-Szklarz, W. F., Hoser, A. A., Gutmann, M., Madsen, A. Ø. & Woźniak, K. (2016). *IUCrJ*, **3**, 2052–2525.

Schmidt, A., Jelsch, C., Østergaard, P., Rypniewski, W. & Lamzin, V. S. (2003). *J. Biol. Chem.* **278**, 43357–43362.

Schnieders, M. J., Fenn, T. D., Pande, V. S. & Brunger, A. T. (2009). *Acta Cryst.* D**65**, 952–965.

Smith, B. J., Colman, P. M., Von Itzstein, M., Danylec, B. & Varghese, J. N. (2001). *Protein Sci.* **10**, 689–696.

Squeglia, F., Ruggiero, A., Romano, M., Vitagliano, L. & Berisio, R. (2014). *Acta Cryst.* D**70**, 2295–2300.

Stewart, R. F., Bentley, J. & Goodman, B. (1975). *J. Chem. Phys.* **63**, 3786–3793.

Su, Z. & Coppens, P. (1990). *J. Appl. Cryst.* **23**, 71–73.

Su, Z. & Coppens, P. (1998). *Acta Cryst.* A**54**, 646–652.

Volkov, A., Li, X., Koritsanszky, T. & Coppens, P. (2004). *J. Phys. Chem. A*, **108**, 4283–4300.

Volkov, A., Macchi, P., Farrugia, L. J., Gatti, C., Mallinson, P., Richter, T. & Koritsanszky, T. (2016). *XD2016 – A Computer Program Package for Multipole Refinement, Topological Analysis of Charge Densities and Evaluation of Intermolecular Energies from Experimental and Theoretical Structure Factors.* http://www.chem.gla.ac.uk/~louis/xd-home/.

Volkov, A., Messerschmidt, M. & Coppens, P. (2007). *Acta Cryst.* D**63**, 160–170.

Waasmaier, D. & Kirfel, A. (1995). *Acta Cryst.* A**51**, 416–431.

Woińska, M., Grabowsky, S., Dominiak, P. M., Woźniak, K. & Jayatilaka, D. (2016). *Sci. Adv.* **2**, e1600192.

Zarychta, B., Pichon-Pesme, V., Guillot, B., Lecomte, C. & Jelsch, C. (2007). *Acta Cryst.* A**63**, 108–125.