T-matrix computations for particles with high-order finite symmetries

T-matrix computations for particles with high-order finite symmetries

Michael Kahnert[a,b]

[a] *Swedish Meteorological and Hydrological Institute, Folkborgsvägen 1, SE-601 76 Norrköping, Sweden*
[b] *Chalmers University of Technology, Department of Earth and Space Science, SE-412 96 Gothenburg, Sweden*

Abstract

The use of group theoretical methods can substantially reduce numerical ill-conditioning problems in T-matrix computations. There are specific problems related to obtaining the irreducible characters of high-order symmetry groups, and to the construction of a transformation from the basis of vector spherical wave functions to the irreducible basis of high-order symmetry groups. These problems are addressed, and numerical solutions are discussed and tested. An important application of the method are non-convex particles perturbed with high-order polynomials. Such morphologies can serve as models for particles with small-scale surface roughness, such as mineral aerosols, atmospheric ice particles with rimed surfaces, and various types of cosmic dust particles. The method is tested for high-order 3D-Chebyshev particles, and the performance of the method is gauged by comparing the results to computations based on iteratively solving a Lippmann-Schwinger T-matrix equation. The latter method trades ill-conditioning problems for potential slow-convergence problems, and it is rather specific, as it is tailored to particles with small-scale surface roughness. The group theoretical method is general and not plagued by slow-convergence problems. The comparison of results shows that both methods achieve a comparable numerical stability. This suggests that for particles with high-order symmetries the group-theoretical approach is able to overcome the ill-conditioning problems. Remaining numerical limitations are likely to be associated with loss-of-precision problems in the numerical evaluation of the surface integrals.

Email address: michael.kahnert@smhi.se (Michael Kahnert)

Preprint submitted to JQSRT *July 31, 2012*

Keywords: Scattering, T-matrix, mineral dust, cosmic dust, ice clouds

1. Introduction

Numerical light scattering methods that are based on rigorous electromagnetic theory typically face two major challenges. The first challenge pertains to high CPU-time requirements, the second one to numerical ill-conditioning or loss-of-accuracy problems. Such problems become more pronounced for particles with large size parameters $x = 2\pi r/\lambda$ (where r is the volume-equivalent radius of the particle, and λ is the wavelength of light), and for particles composed of materials with large real or imaginary parts of the refractive index $m = n + i\kappa$. The use of group theory in light scattering computations addresses both of these problems [1].

Imposing symmetry assumptions is the only known method that can substantially decrease CPU-time requirements in numerically exact electromagnetic scattering computations. For instance, in T-matrix computations based on Waterman's null-field method [2] exploiting geometrical symmetries of the scattering particle can reduce computation time by several orders of magnitude, depending on the order of the symmetry group [3, 4]. (The order of a group is the number of elements in the group.) This is, in fact, one of the reasons why spheres and axisymmetric particles, such as spheroids and circular cylinders, have been very popular model particles in applications to aerosols, hydrometeors, hydrosols, and cosmic dust, even though they show little morphological similarity with typical irregular particles encountered in the atmosphere, ocean, or in the interplanetary or interstellar medium.

Using the irreducible representations of finite symmetry groups can significantly alleviate ill-conditioning problems in the null-field method [1], which can help to extend the range of size parameters for which convergent results can be obtained. Previously, this approach has only been tested for particles that belong to symmetry groups of relatively low order. More recently, particles with high-order symmetries have been considered as a model for particles with small-scale surface roughness [5, 6]. As an example, Fig. 1 shows a high-order 3D-Chebyshev particle with a spherical base geometry. The boundary surface of such particles is given by

$$r(\theta, \phi) = r_0(1 + \epsilon \cos n\theta \cos n\phi), \tag{1}$$

where r_0 is the size of the unperturbed sphere, θ and ϕ denote the polar and azimuth angles, respectively, ϵ is the deformation parameter, and n is

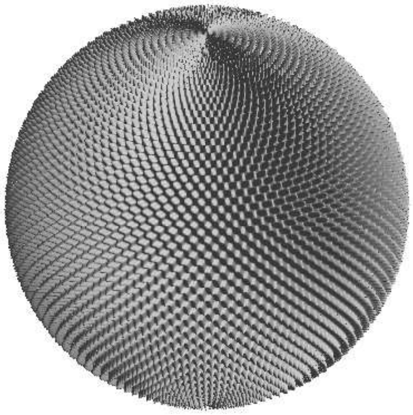

Figure 1: 3D-Chebyshev particle or order 80.

the order of the Chebyshev polynomial. Thus the surface of this particle is perturbed by Chebyshev polynomials in both the polar and the azimuth directions. Numerically exact T-matrix computations for such particles are computationally challenging. Group theory, in particular the use of irreducible representations, can be useful for improving the numerical stability of T-matrix computations for such particles.

However, there are practical challenges related to high-order symmetry groups in numerical applications. The main purpose of this paper is to discuss the numerical and practical problems that need to be addressed when applying the theory developed in [1] to particles with high-order symmetries. Practical problems in the application of the theory and ways to solve them are discussed in Sect. 2. Illustrative results are presented in Sect. 3, and concluding remarks are summarised in Sect. 4. A a short review of the T-matrix formalism and of finite point groups is given in the appendix.

2. Practical implementation of group theoretical methods in T-matrix computations

The T-matrix code used in this study is the *Tsym* code described in [1, 3]. This program is specifically tailored to non-axisymmetric particles with discrete symmetries. It is, to the best of my knowledge, the only code that fully exploits group theory not only by using commutation relations of the T-matrix, but also by making use of irreducible representations. However, the earlier versions of the code have been limited to low-order point groups. This is a rather severe shortcoming that limits the usefulness of the method to particles with relatively low symmetries. The main goal of the present study is to overcome these limitations. The paper discusses specific practical problems one encounters in applications of group theory to T-matrix computations for particles with high-order symmetries, and how these problems can be solved.

2.1. Statement of the problem

This section concisely introduces the main concepts of group theoretical methods in T-matrix computations. Readers unfamiliar with this subject are advised to first read the appendix. There, also a short introduction into the T-matrix formulation, Waterman's null-field method, and the iterative Lippmann-Schwinger T-matrix equation is given.

The main goal of using irreducible representations of finite groups in T-matrix computations is to reduce ill-conditioning problems by transforming into a new basis of the function space. This basis is known as the irreducible basis. The matrix $\mathbf{P}$ that transforms from the basis of vector spherical wave functions into the irreducible basis is constructed from projection matrices

$$\tilde{\mathbf{P}}^{(\mu)} = \sum_{g\in\mathcal{G}} \chi^{(\mu)*}(g)\mathbf{D}(g). \tag{2}$$

The summation extends over all elements g of the group $\mathcal{G}$. The superscript $\mu = 1,\ldots,r$ labels the μ-th invariant subspace, $\chi^{(\mu)}(g)$ are the irreducible characters, and $\mathbf{D}(g)$ denotes the reducible representations. The matrix $\tilde{\mathbf{P}}^{(\mu)}$ projects into the μ-th invariant subspace (see the appendix for more details). To compute $\tilde{\mathbf{P}}^{(\mu)}$ one needs to obtain the reducible representations $\mathbf{D}(g)$ and the irreducible characters $\chi^{(\mu)}$. Computation of $\mathbf{D}(g)$ has been discussed in [1, 3]. The focus in the following sections will be on the numerical computation of the $\chi^{(\mu)}$ for high-order finite point groups.

2.2. Tsym and GAP

The starting point for using irreducible representations of finite groups in the T-matrix formalism is the construction of the projection matrices in Eq. (2), which requires the characters $\chi^{(\mu)}$. For low-order point groups, one can often find tabulated values of the irreducible characters in the theoretical chemistry and solid state physics literature (e.g. [7]). The earlier versions of *Tsym* have relied on such tabulated values, thus limiting the program to low-order symmetry groups. For high-order point groups, the characters have to be computed. This computation can be done without prior knowledge of the irreducible representations or the invariant subspaces (e.g. [8, 9]). Such methods are implemented in the *Groups Algorithms and Programming* (*GAP*) system for computational discrete algebra [10]. *GAP* can be run either interactively or as a script language. Here, the use of *GAP* is described for computing the irreducible characters of finite point groups, and how these computations are coupled to the T-matrix program *Tsym*. Section 2.2.1 will mainly give a tutorial introduction on how to compute character tables of point groups with *GAP*, and on how to interpret the *GAP* output. The simple examples given there mainly serve illustrative purposes; however, they do not elucidate the difficulties one encounters in applications to high-order point groups, in automatising the computations, and in coupling the results to T-matrix calculations. These issues are addressed in Sect. 2.2.2. Although the

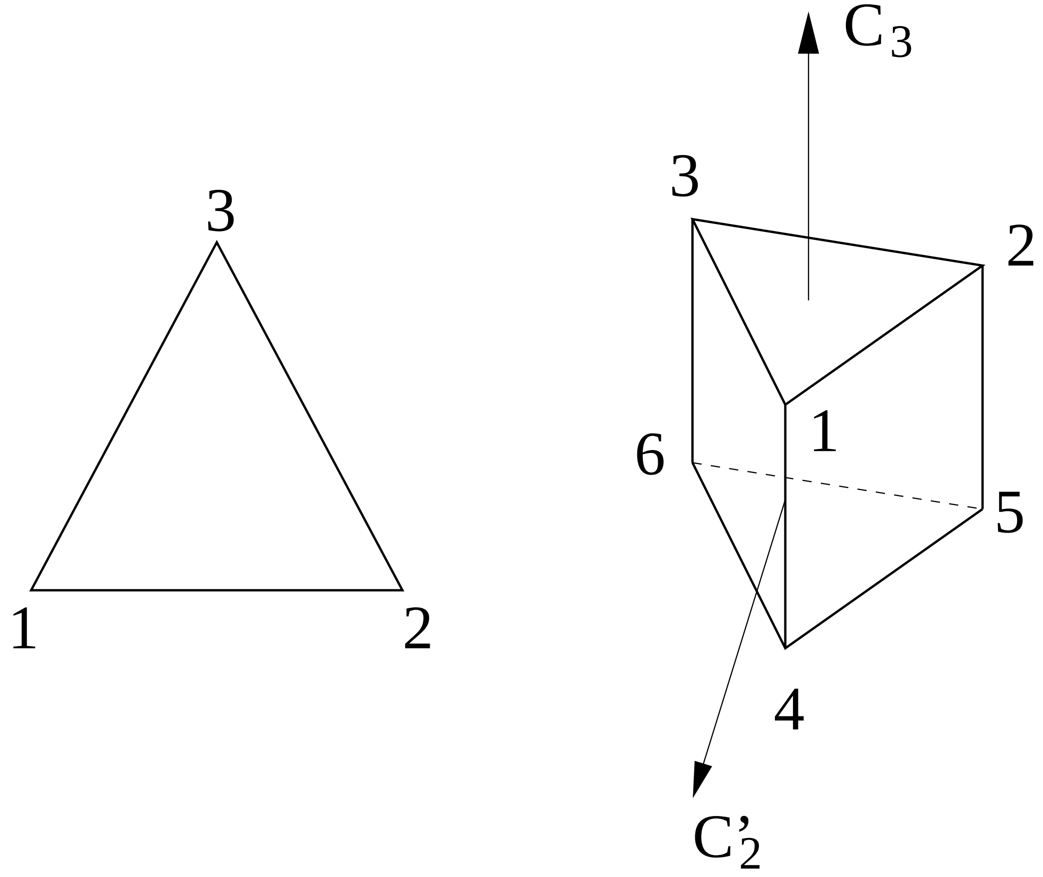

Figure 2: Illustration of the symmetry operations in the groups $\mathcal{C}_3$ and $\mathcal{D}_{3h}$.

discussion in Sect. 2.2.2 will be a bit more technical, it describes some of the main programming efforts required for extending T-matrix computations to high-order symmetry groups, and it may help other program developers to avoid some of the main pitfalls.

2.2.1. Application of GAP *to point groups*

Point groups can be defined in *GAP* by specifying the group generators. For instance, the group $\mathcal{C}_3$ can be characterised by the generator C_3. In *GAP*, the element C_3 is represented by the cyclic permutation $(1, 2, 3)$, which can be visualised as a rotation of a triangle with corners labelled by 1, 2, and 3 (see Fig. 2 left). Thus the group $\mathcal{C}_3$ can be specified in the *GAP* language by

$$\texttt{C3} := \texttt{Group}((\texttt{1},\texttt{2},\texttt{3}));; \tag{3}$$

The group $\mathcal{D}_{3h}$ has three generators, C_3, C_2', and σ_h. These operations can be visualised by their action on a triangular prism (see Fig. 2 right). The rotation C_3 cyclically permutes both the three upper corners and the three lower corners among themselves. Thus C_3 can be represented by $(1,2,3)(4,5,6)$. The dihedral rotation C_2' rotates the prism by an angle $\pi(= 2\pi/2)$ about an axis perpendicular to the C_3-axis. There are three dihedral symmetry axes, and each of them can be chosen as a generator of the group. The one indicated in the figure exchanges corners 1 with 4, 2 with 6, and 3 with 5, so $C_2' = (1,4)(2,6)(3,5)$. Finally, the horizontal reflection σ_h perpendicular to the C_3-axis exchanges the corners 1 with 4, 2 with 5, and 3 with 6, so $\sigma_h = (1,4)(2,5)(3,6)$. Thus the group $\mathcal{D}_{3h}$ can be specified in *GAP* by

$$\texttt{D3h} := \texttt{Group}((\texttt{1},\texttt{2},\texttt{3})(\texttt{4},\texttt{5},\texttt{6}),(\texttt{1},\texttt{4})(\texttt{2},\texttt{6})(\texttt{3},\texttt{5}),(\texttt{1},\texttt{4})(\texttt{2},\texttt{5})(\texttt{3},\texttt{6}));; \tag{4}$$

The irreducible characters of a group G can be computed and printed in *GAP* by the command `Irr(G)`. For instance, for the group `G` $=$ `C3` we can use the commands

$$\begin{array}{c} \texttt{tab} := \texttt{Irr}(\texttt{C3});; \\ \texttt{Display}(\texttt{tab}); \end{array} \tag{5}$$

which yields the output

$$\begin{array}{llrrrl} [& [& 1, & 1, & 1 &], \\ & [& 1, & E(3), & E(3)\hat{}2 &], \\ & [& 1, & E(3)\hat{}2, & E(3) &]] \end{array} \tag{6}$$

where, in general, $E(n) = \exp(2\pi\mathrm{i}/n)$. The three rows in this table belong to the three irreducible representations of the group $\mathcal{C}_3$, the columns belong to the three conjugacy classes of this group. We do not need to be concerned about the order of the rows. If we re-order the rows in the character table, then we will merely alter the order of the projection matrices in Eq. (2) accordingly. However, we do need to worry about the order of the columns. According to Eq. (2), the complex conjugate of the irreducible character $\chi^{(\mu)}(g)$ needs to be multiplied by the reducible representation $\mathbf{D}(g)$ for each group element g. Therefore we need to know which column in the character table belongs to which conjugacy class. (All group elements belonging to the same conjugacy class have the same character — see the appendix for details.) To find out about the order of the conjugacy classes of the character tables produced by *GAP*, one can use the command

$$\texttt{ConjugacyClasses(C3)}; \tag{7}$$

which yields

$$[()\hat{}G, (1,2,3)\hat{}G, (1,3,2)\hat{}G] \tag{8}$$

In this simple example, each of the three group elements is in a class by itself. The class $[E]$ containing the unit element is denoted by $()\hat{}G$, the one containing the rotation by $2\pi/3$, $[C_3]$, is denoted by $(1,2,3)\hat{}G$, and the class $[C_3^2]$ containing the rotation by $4\pi/3$ is denoted by $(1,3,2)\hat{}G$.

The group $\mathcal{D}_{3h}$ is slightly less trivial. *GAP* produces the following character table for this group:

$$\begin{array}{l} \texttt{tab} := \texttt{Irr(D3h)};; \\ \texttt{Display(tab)}; \end{array} \tag{9}$$

$$\begin{array}{lrrrrrr} [& [1, & 1, & 1, & 1, & 1, & 1], \\ & [1, & -1, & 1, & -1, & 1, & -1], \\ & [1, & -1, & 1, & 1, & -1, & 1], \\ & [1, & 1, & 1, & -1, & -1, & -1], \\ & [2, & 0, & -1, & -2, & 0, & 1], \\ & [2, & 0, & -1, & 2, & 0, & -1]] \end{array} \tag{10}$$

Again, in order to use these characters in Eq. (2) we need to know the correspondence between the table columns and the conjugacy classes of the group. The *GAP* command

$$\texttt{ConjugacyClasses(D3h)}; \tag{11}$$

yields

$$\begin{aligned} &[()\hat{}G,\ (2,3)(5,6)\hat{}G,\ (1,2,3)(4,5,6)\hat{}G,\ (1,4)(2,5)(3,6)\hat{}G, \\ &\quad (1,4)(2,6)(3,5)\hat{}G,\ (1,5,3,4,2,6)\hat{}G] \end{aligned} \tag{12}$$

The group has six conjugacy classes. The unit element is in a class by itself, $[E]$, which in the *GAP* output is denoted by $()\hat{}G$. The second class $[\sigma_v]$ contains three elements, namely, three vertical reflection planes $\sigma_v^{(0)}$, $\sigma_v^{(1)}$, and

$\sigma_v^{(2)}$. In the example shown in Fig. 2 (right), each of these reflection planes contains one of the vertical edges of the prism and intercepts the opposite face at a right angle. The reflection plane containing the edge that joins corners 1 and 4 exchanges corners 2 with 3 and 5 with 6. The conjugacy class represented by this group element is denoted by $[\sigma_v]$=$(2,3)(5,6)\hat{}G$. Note, however, that this choice is not unique. One could have chosen any of the three σ_v operations to represent this conjugacy class. Next, the class $[C_3]$=$(1,2,3)(4,5,6)\hat{}G$ contains the two elements C_3 and C_3^2. The following class $[\sigma_h]$=$(1,4)(2,5)(3,6)\hat{}G$ only contains the horizontal reflection operation σ_h. The next class $[C_2']$ contains three dihedral rotation operations $C_2'^{(0)}$, $C_2'^{(1)}$, and $C_2'^{(2)}$. The corresponding rotation axes each intercept one of the vertical edges and opposite faces at a right angle and pass through the centre of the prism. In the *GAP* output given in (12) the class is denoted by $(1,4)(2,6)(3,5)\hat{}G$. Thus, *GAP* represents this class by the element that exchanges corners 1 with 4, 2 with 6, and 3 with 5. This is the element belonging to the rotation axis that intercepts the edge 1-4 and the face 2-3-6-5. Finally, the group $\mathcal{D}_{3h}$ possesses one class $[S_3]$ that contains two rotation-reflection operations S_3 and $S_3^{(2)}$. The first one is given by $S_3 = \sigma_h \cdot C_3$, i.e., a C_3 rotation followed by a horizontal reflection. The second one is given by $S_3^{(2)} = \sigma_h \cdot C_3^2$. Inspection of Fig. 2 (right) shows that S_3 performs a cyclic permutation $(1,5,3,4,2,6)$. Thus the last conjugacy class in (12) is represented by the S_3 element.

2.2.2. Automatised computation of irreducible characters in Tsym *by use of* GAP

The examples given above were relatively simple, because only low-order point groups have been considered. For high-order point groups the character tables and, even more so, the representations of the conjugacy classes can become revoltingly complex, and it is no longer practical to generate the character tables interactively and analyse the conjugacy classes by inspection. It is therefore necessary to automatise the computations. To this end, *Tsym* uses a FORTRAN program to generate an input script to *GAP*, and another FORTRAN program to analyse the conjugacy classes and to post-process the character table. The post-processing arranges the columns in the table according to the *Tsym* standard ordering of conjugacy classes. A master script runs the input generation, calls *GAP*, and post-processes the results. The user merely has to specify the group; possible choices are the non-cubic groups $\mathcal{C}_n$, $\mathcal{C}_{nv}$, $\mathcal{C}_{nh}$, $\mathcal{D}_n$, $\mathcal{D}_{nh}$, $\mathcal{D}_{nd}$, $\mathcal{S}_n$, $\mathcal{C}_s$, $\mathcal{C}_i$, as well as the cubic groups $\mathcal{T}$, $\mathcal{T}_d$, $\mathcal{O}$, and $\mathcal{O}_h$. The first seven denote families of groups with a variable index n of the main rotational symmetry axis, which the user also needs to specify. The output is a character table in which the conjugacy classes are arranged in a standardised order. The table is then read in by *Tsym* and used in the computation of the projection matrices according to Eq. (2).

There are a few technical issues encountered when applying *GAP* to high-order point groups, which will be briefly address here.

- The representations of group generators and conjugacy classes will, in general, become more complicated for high rotational symmetries n. For instance, the group $\mathcal{C}_n$ contains $n-1$ conjugacy classes $[C_n], [C_n^2], \ldots, [C_n^{n-1}]$, where the *GAP* representation of the rotations take the form

$$\begin{aligned}
C_n &= (1, 2, 3, \ldots, n) \\
C_n^2 &= \left(1, 3, \ldots, \begin{array}{lcl} n-1 & : & n \text{ even} \\ n & : & n \text{ odd} \end{array}\right)\left(2, 4, \ldots, \begin{array}{lcl} n & : & n \text{ even} \\ n-1 & : & n \text{ odd} \end{array}\right) \\
&\vdots \\
C_n^j &= (1, j+1, 2j+1 \ldots) \cdots (g_{nj}, j+g_{nj}, 2j+g_{nj}, \ldots) \\
&\vdots
\end{aligned}$$

  where g_{nj} is the greatest common divisor of n and j. So, rotations (as well as rotation-reflections) in higher-order groups can consist of several permutations. Again, one can use regular polygons to visualise this. For instance, for the simple example of a rectangle with corners 1, 2, 3, and 4, the rotation C_4^2 will consist of two permutations $(1,3)(2,4)$.

- When testing *GAP* for groups such as $\mathcal{D}_{nh}$ with $n \leq 130$, it appears that the ordering of the conjugacy classes follows a fixed pattern. This is no longer the case for $n > 130$. In fact, the ordering of the conjugacy classes in *GAP* is quite random for high-order groups. This is why it is absolutely essential to analyse, for each group, the ordering of the conjugacy classes in the *GAP* character tables.

- Further, for $n \leq 130$ *GAP* consistently chooses the same elements for representing the conjugacy classes. For larger n the choice of representatives does no longer seem to follow a predictable pattern. For example, the class $[C_{130}^{(6)}]$ of the group $\mathcal{D}_{130h}$ is not, as one may expect, represented as

$$\left[C_{130}^{6}\right] = (1, 7, 13, 19, \ldots)\hat{}G \tag{13}$$

but rather as

$$\left[C_{130}^{124}\right] = (1, 125, 119, 113, 107\ldots)\hat{}G \tag{14}$$

where $C_{130}^{124} = C_{130}^{130-6}$. The automatised analysis of the conjugacy classes therefore has to anticipate different choices of representing the classes.

- For high-order groups the expressions for the conjugacy classes become very long. In such cases, *GAP* may truncate the expression. For instance, the class $[C_{130}^{10}]$ of the group $\mathcal{D}_{130h}$ is represented in *GAP* as

$$\begin{array}{l}
(1, 11, 21, 31, 41, 51, 61, 71, 81, 91, 101, 111, 121)\\
(2, 12, 22, 32, 42, 52, 62, 72, 82, 92, 102, 112, 122)\\
(3, 13, 23, 33, 43, 53, 63, 73, 83, 93, 103, 113, 123)\\
(4, 14, 24, 34, 44, 54, 64, 74, 84, 94, 104, 114, 124)\\
(5, 15, 25, 35, 45, 55, 65, 75, 85, 95, 105, 115, 125)\\
(6, 16, 26, 36, 46, 56, 66, 76, 86, 96, 106, 116, 126)\\
(7, 17, 27, 37, 47, 57, 67, 77, 87, 97, 107, 117, 127)\\
(8, 18, 28, 38, 48, 58, 68, 78, 88, 98, 108, 118, 128)\\
(9, 19, 29, 39, 49, 59, 69, 79, 89, 99, 109, 119, 129)\\
(10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130)\\
(131, 141, 151, 161, 171, 181, 191, 201, 211, 221, 231, 241, 251)\\
(132, 142, 152, 162, 172, 182, 192, 202, 212, 222, 232, 242, 252)\\
(133, 143, 153, 163, 173, 183, 193, 203, 213, 223, 233, 243, 253)\\
(134, 144, 154, 164, 174, 184, 194, 204, 214, 224, 234, 244, 254)\\
(135, 145, 155, 165, 175, 185, 195, 205, 215, 225, 235, 245, 255)\\
(136, 146, 156, 166, 176, 186, 196, 206, 216, 226, 236, 246, 256)([\ldots])\hat{}G
\end{array}$$

Thus the last four lines are omitted, which is indicated by "[...]". Again, the automated analysis of the conjugacy classes has to anticipate such truncations.

- The entries of the character tables are strings rather than numbers, which can be seen in (6). The *Tsym* program converts these strings to complex numbers. For high-order point groups the relevant subroutines need to be prepared to handle fairly long string variables. For instance, for the group $\mathcal{D}_{242h}$ a single character entry can be as long as in the following example:

$$\begin{aligned}
&- \; E(121)\hat{}7 - E(121)\hat{}15 - E(121)\hat{}18 - E(121)\hat{}26 - E(121)\hat{}29 \\
&- \; E(121)\hat{}37 - E(121)\hat{}40 - E(121)\hat{}48 - E(121)\hat{}51 - E(121)\hat{}59 \\
&- \; E(121)\hat{}62 - E(121)\hat{}70 - E(121)\hat{}73 - E(121)\hat{}81 - E(121)\hat{}84 \\
&- \; E(121)\hat{}92 - E(121)\hat{}95 - E(121)\hat{}103 - E(121)\hat{}106 \\
&- \; E(121)\hat{}114
\end{aligned} \tag{15}$$

Automatising the generation of *GAP* input scripts and automatising the analysis of the conjugacy classes via FORTRAN programs constitute the main programming efforts in linking *GAP* to a T-matrix program such as *Tsym*.

2.3. Computation of the P-matrix

Equation (2) allows us to project any vector $\mathbf{x}$ into any of the invariant subspaces T_μ, viz. $\tilde{\mathbf{P}}^{(\mu)} \cdot \mathbf{x} \in T_\mu$. What we actually want is a transformation matrix $\mathbf{P}$ into the irreducible basis. Transformation of the Q-matrix into the irreducible basis, i.e.

$$\mathbf{Q}_{\text{irr}} = \mathbf{P} \cdot \mathbf{Q} \cdot \mathbf{P}^{-1} \tag{16}$$

will bring the Q-matrix into block-diagonal form.

Three different numerical approaches have been tested in *Tsym* to construct the transformation $\mathbf{P}$ from the projectors $\tilde{\mathbf{P}}^{(\mu)}$.

- The "pedestrian method" is to loop through the row vectors of $\tilde{\mathbf{P}}^{(\mu)}$ to identify the linearly independent row vectors of the projection matrix. The number of linearly independent row vectors should be equal to the dimension of the range of the projection matrix, which, should be equal to the dimension d_μ of the μth invariant subspace. The latter is equal to $\alpha_\mu \chi^{(\mu)}(E)$, where α_μ denotes how many times the μth irreducible representation occurs in the reducible one, and where $\chi^{(\mu)}(E)$ is equal to the dimension of the μth irreducible representation. Let $\tilde{\mathbf{p}}_1^{(\mu)}, \ldots, \tilde{\mathbf{p}}_{d_\mu}^{(\mu)}$

denote these row vectors. Then, by collecting the linearly independent row vectors of all projectors, we can define the matrix

$$\mathbf{P} := \begin{pmatrix} \tilde{\mathbf{p}}_1^{(1)} \\ \vdots \\ \tilde{\mathbf{p}}_{d_1}^{(1)} \\ \tilde{\mathbf{p}}_1^{(2)} \\ \vdots \\ \tilde{\mathbf{p}}_{d_2}^{(2)} \\ \vdots \\ \tilde{\mathbf{p}}_1^{(r)} \\ \vdots \\ \tilde{\mathbf{p}}_{d_r}^{(r)} \end{pmatrix} . \tag{17}$$

The sum $\sum_{\mu=1}^{r} d_\mu$ is equal to the dimension of the vector space, so $\mathbf{P}$ is a square matrix.

This method has been used since the earliest versions of *Tsym*. It is simple, easily implemented, and relatively fast. It works well for low-order symmetry groups. However, it does have practical drawbacks that can become important for higher-order symmetry groups. Linear independence of two vectors $\mathbf{x}$ and $\mathbf{y}$ means that both vectors are identical up to a multiplicative constant, i.e. there exists some constant C such that $x_i - C\,y_i = 0$ for all vector components i. Checking this criterion numerically requires some pre-defined precision. If the precision is set too small or too large, then the algorithm may fail to identify the correct number of linearly independent row vectors.

- A more systematic way to construct the matrix $\mathbf{P}$ is to perform a singular value decomposition (SVD) of the projectors $\tilde{\mathbf{P}}^{(\mu)}$, i.e.

$$\tilde{\mathbf{P}}^{(\mu)} = \mathbf{U}^{(\mu)} \cdot \mathbf{\Sigma}^{(\mu)} \cdot \mathbf{V}^{(\mu)\dagger}. \tag{18}$$

The diagonal matrix $\mathbf{\Sigma}^{(\mu)}$ should contain exactly $d_\mu = \alpha_\mu \chi^{(\mu)}(E)$ non-zero singular values. One can exploit this fact to check the numerical accuracy of the SVD algorithm. The singular value $\Sigma_{d_\mu+1}$ either has to be exactly zero, or, at least, the ratio $\Sigma_{d_\mu+1}/\Sigma_{d_\mu}$ should be very small.

The columns of $\mathbf{U}^{(\mu)}$ and the rows of $\mathbf{V}^{(\mu)\dagger}$ are the left singular and right singular vectors, respectively. The d_μ column vectors $\mathbf{u}_1^{(\mu)}, \ldots, \mathbf{u}_{d_\mu}^{(\mu)}$ belonging to the non-zero singular values are a basis of the range of $\tilde{\mathbf{P}}^{(\mu)}$. We can construct the sought transformation into the irreducible basis by collecting these vectors from each projector, i.e.

$$\mathbf{P} := \left(\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_{d_1}^{(1)}, \mathbf{u}_1^{(2)}, \ldots, \mathbf{u}_{d_2}^{(2)}, \ldots, \mathbf{u}_1^{(r)}, \ldots, \mathbf{u}_{d_r}^{(r)} \right). \tag{19}$$

Alternatively, we can use the right singular row vectors $\mathbf{v}_1^{(\mu)\dagger}, \ldots, \mathbf{v}_{d_\mu}^{(\mu)\dagger}$ belonging to the non-zero singular values to define

$$\mathbf{P} := \begin{pmatrix} \mathbf{v}_1^{(1)\dagger} \\ \vdots \\ \mathbf{v}_{d_1}^{(1)\dagger} \\ \mathbf{v}_1^{(2)\dagger} \\ \vdots \\ \mathbf{v}_{d_2}^{(2)\dagger} \\ \vdots \\ \mathbf{v}_1^{(r)\dagger} \\ \vdots \\ \mathbf{v}_{d_r}^{(r)\dagger} \end{pmatrix}. \tag{20}$$

Computation of the inverse $\mathbf{P}^{-1}$ becomes trivial. Since $\mathbf{U}^{(\mu)}$ and $\mathbf{V}^{(\mu)}$ are unitary, so is $\mathbf{P}$, i.e. $\mathbf{P}^{-1} = \mathbf{P}^\dagger$.

Tsym uses the LAPACK routine ZGESVD for performing the SVD of the projectors. This method is very robust and produces highly accurate results even for high-order symmetry groups and vector spaces of high dimension. The main disadvantage of this approach is that it can become very slow if the dimension of the vector space (which is determined by the truncation index of the T-matrix) becomes large. A lesser concern is the order of the symmetry group. The computation time scales only linearly with the number of irreducible representations r.

- The projection matrices are highly sparse. This fact is not taken into account in the general SVD routine ZGESVD. The newest version of

Tsym also offers the option of using the PROPACK routine ZLANSVD, which is an SVD routine specifically devised for large and highly sparse matrices. It uses a Lanczos bidiagonalisation with partial reorthogonalisation [11]. This method is significantly faster than the general SVD algorithm. However, in some of the test cases involving highly dimensional vector spaces and high-order symmetry groups, the numerical accuracy was not always as high as that achieved with ZGESVD.

3. Applications

As an example, let us consider light scattering by high-order 3D-Chebyshev particles, such as that shown in Fig. 1. It was shown in [12] that light scattering computations for particles with small-scale surface roughness can be numerically challenging and, as a consequence, are often limited to small size parameters. This is particularly the case when differential scattering quantities have to be computed with high accuracy.

As mentioned previously, there are two very different approaches for improving the numerical stability of T-matrix computations for such particles. One is based on an iterative solution of the Lippmann-Schwinger equation (32), the other makes use of irreducible representations of high-order symmetry groups, as discussed in the previous sections. As an illustration, a comparison of both methods is presented for 3D-Chebyshev particles of order $n = 160$, with a size parameter $x = 40$, and a refractive index $m = 3 + 0.1\mathrm{i}$. The refractive index is typical for hematite (neglecting birefringence). It is chosen here because materials with such high real and imaginary parts of the refractive index usually pose a greater numerical challenge in T-matrix computations. The Chebyshev particle belongs to the group $\mathcal{D}_{160h}$, which has order $M_o = 640$, and which contains $r = 166$ conjugacy classes.

Figure 3 shows the elements F_{11} (upper left) and $-F_{12}/F_{11}$ (upper right) of the orientation-averaged Mueller matrix of 3D-Chebyshev particles with deformation parameter $\epsilon = 0.0175$ (see Eq. (1)), where the computations exploited irreducible representations in the inversion of the Q-matrix. The lower two panels show differences in the Mueller matrix elements computed with the group theoretical method and with the iterative method, where the maximum iteration order p (see Eq. (32)) varies between $p = 1$ (thin solid line), $p = 3$ (dashed line), and $p = 6$ (thick solid line). It is evident that the iterative solution quickly converges against the solution obtained with the group theoretical method as the maximum iteration order is increased.

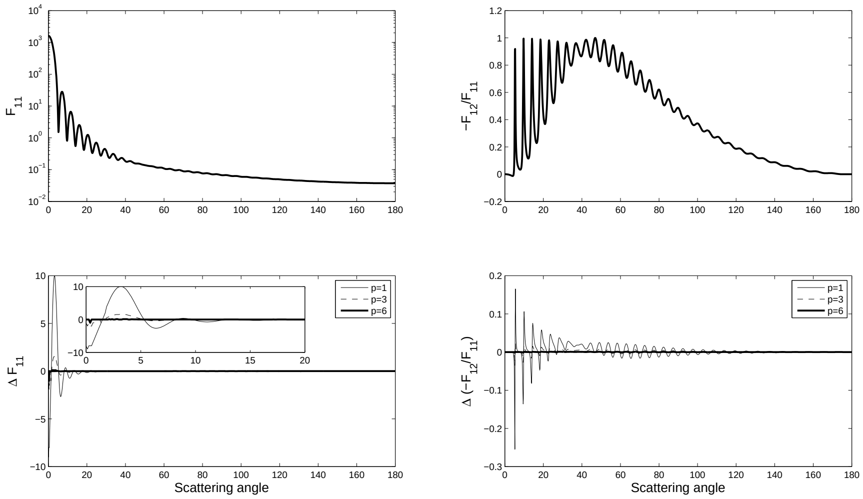

Figure 3: Mueller matrix elements F_{11} (upper left) and $-F_{12}/F_{11}$ (upper right) of 3D-Chebyshev particles with $\epsilon = 0.0175$, computed with the group theoretical method. The lower panels show differences between these elements and corresponding results obtained with the iterative approach.

The two methods considered here differ in the numerical approach for inverting the Q-matrix, but they both use the same routines for numerically evaluating the surface integrals in Waterman's null-field method. Therefore, the good agreement of the results indicates that both approaches perform equally well for the inversion of the Q-matrix, but is does not guarantee that the Q-matrix is computed with sufficiently high accuracy. To check this, one always needs to test in T-matrix calculations that the computational results have converged with respect to the truncation of the series expansions in (22)–(24), and with respect to the number of quadrature points in the numerical computation of the surface integrals. In addition, one can use the reciprocity condition for the polarised differential scattering cross section [13, 14]. This condition states that

$$k^2 \left(\frac{\mathrm{d}\sigma}{\mathrm{d}\Omega}\right)_{\alpha,\beta}(\hat{k}_{\mathrm{inc}}, \hat{k}_{\mathrm{sca}}) = k^2 \left(\frac{\mathrm{d}\sigma}{\mathrm{d}\Omega}\right)_{\beta,\alpha}(-\hat{k}_{\mathrm{sca}}, -\hat{k}_{\mathrm{inc}}). \tag{21}$$

where k denotes the wavenumber, and $(\mathrm{d}\sigma/\mathrm{d}\Omega)_{\alpha,\beta}(\hat{k}_{\mathrm{inc}}, \hat{k}_{\mathrm{sca}})$ is the polarised differential scattering cross section for an incident wave with polarisation $\alpha = h$ or $\alpha = v$ travelling in the direction $\hat{k}_{\mathrm{inc}}$, and a scattered wave with polarisation $\beta = h$ or $\beta = v$ travelling in the direction $\hat{k}_{\mathrm{sca}}$. h- and v-polarisations refer to waves polarised in the scattering plane and perpendicular to it, respectively. Thus the reciprocity condition expresses the invariance of the scattering process under reversal of the optical path, i.e. under exchange of source and receptor point. It is a necessary condition for the correctness of numerical electromagnetic scattering computations.

To be more specific, let us choose a scattering angle $\Theta = \cos^{-1}(\hat{k}_{\mathrm{inc}}\cdot\hat{k}_{\mathrm{sca}}) = 90^\circ$. Table 1 shows the percent-error obtained by comparing the differential scattering cross sections for hh- and vv-polarisation at $\Theta = 90^\circ$ with those computed in the reciprocal case. Reciprocity errors less than 5 % are usually considered to be good, errors less than 3 % are considered excellent (e.g. [15]). Clearly, the group theoretical approach yields highly accurate results. The

Table 1: Reciprocity errors in percent for ϵ =0.0175.

error	Irred Rep	p=1	p=3	p=6
δ_{hh}	$8.7\cdot 10^{-2}$	0.12	$9.1\cdot 10^{-2}$	$8.7\cdot 10^{-2}$
δ_{vv}	$1.1\cdot 10^{-3}$	$4.7\cdot 10^{-2}$	$7.3\cdot 10^{-3}$	$1.3\cdot 10^{-3}$

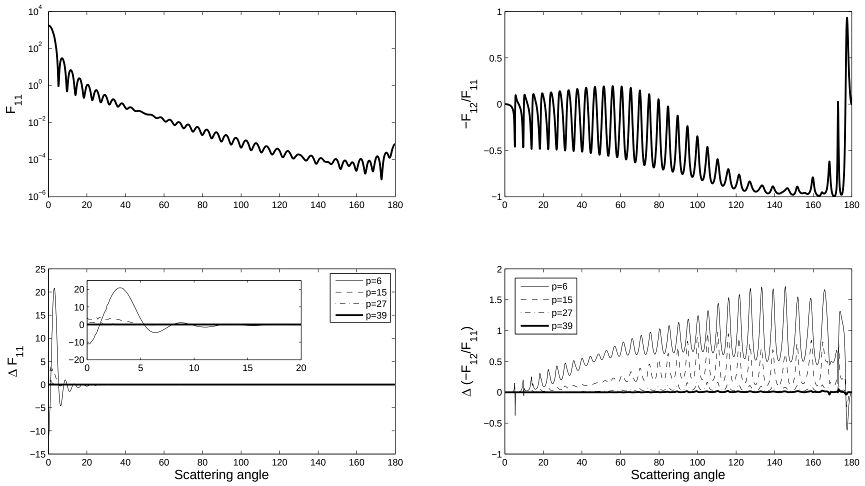

Figure 4: As Fig. 3, but for $\epsilon = 0.03$.

iterative method is already quite accurate at $p = 1$, and it quickly reaches the same accuracy as the group theoretical method as the iteration order is increased to $p = 6$.

Figure 4 shows results analogous to those in Fig. 3, but for a deformation parameter $\epsilon = 0.03$. Corresponding reciprocity errors are shown in Table 2. The reciprocity errors for the group theoretical method are well below 3 %. For the iterative approach Fig. 3 and Table 2 present results for p =6, 15, 27, and 39. Iteration orders in the range p =27–39 are required for the Mueller matrix and reciprocity errors to converge to the results obtained with the group theoretical method. This shows that rapid convergence of the iterative method is only achieved if the perturbed geometry with Q-matrix $\mathbf{Q}$ does not deviate too much from the unperturbed shape with Q-matrix $\mathbf{Q}_0$. As the deformation parameter ϵ is increased, the iterative method may

Table 2: As Table 1, but for ϵ =0.03.

error	Irred Rep	p=6	p=15	p=27	p=39
δ_{hh}	1.9	5.2	4.6	2.3	1.3
δ_{vv}	1.5	0.83	0.76	1.3	1.4

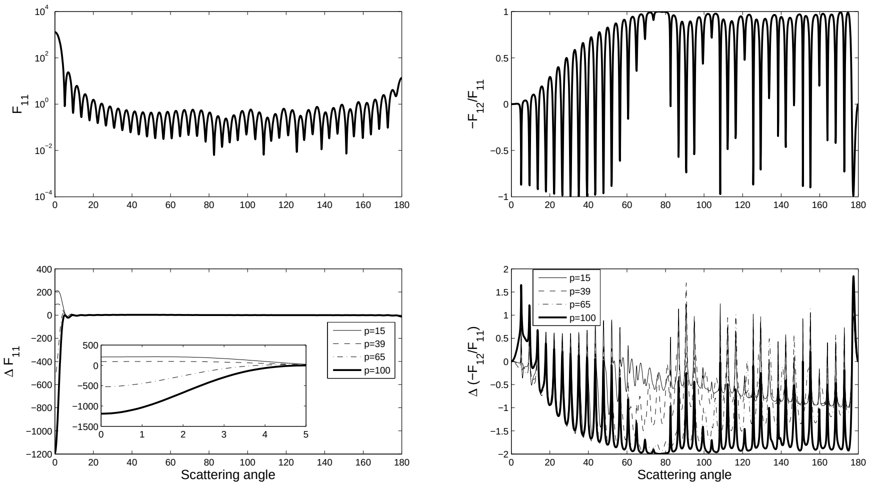

Figure 5: As Fig. 3, but for $\epsilon = 0.04$.

be plagued by slow-convergence problems.

In both test cases considered here, the reciprocity errors in the group theoretical method are comparable with those in the iterative method, provided that one chooses a sufficiently high iteration order. This indicates that the numerical stability of the Q-matrix inversion of the group theoretical method can compete with the numerical robustness of the iterative method, which is quite remarkable. (Recall that for a spherical base geometry, the ill-conditioning problem is completely eliminated in the iterative method!) By comparing the reciprocity errors for $\epsilon = 0.0175$ and $\epsilon = 0.03$, one can see that the numerical accuracy decreases as the deformation parameter is increased (even though the error is still well below the 3 % mark). The cause for this is *not* a deterioration of the conditioning of the Q-matrix inversion. (If this were the cause, then one would only observe higher reciprocity errors in the group theoretical method, not in the iterative method.) This indicates that there are numerical error sources other than the Q-matrix inversion.

These observations and their interpretation become even clearer as on increases the deformation parameter ϵ beyond the current numerical capabilities of the code. Figure 5 and Table 3 show results analogous to those in the preceding figures and tables, but for $\epsilon = 0.04$. Results for the iterative method are shown for maximum iteration orders of p =15, 39, 65, and 100.

Table 3: As Table 1, but for ϵ =0.04.

error	Irred Rep	p=15	p=39	p=65	p=100
δ_{hh}	11	0.40	2.0	1.7	2.8
δ_{vv}	2.5	0.66	1.5	23	95

The Mueller matrix elements computed with the two methods diverge as the iteration order p is increased. At the same time, the reciprocity errors in the iterative method increase with p. Naively, one may have expected that the convergence of the iterative method simply becomes slower as one increases ϵ. Instead, the divergence of the iteration results provides strong evidence for the presence of numerical error sources unrelated to the Q-matrix inversion.

Indeed, it has been pointed out recently [16, 17] that the numerical evaluation of the surface integrals themselves can be plagued by severe loss of precision problems. This will yield an inaccurate Q-matrix, thus an inaccurate T-matrix, irrespective of what method is used for inverting the Q-matrix. Thus it is likely that the high reciprocity error of the group theoretical method in Table 3 is caused by such error sources, rather than by ill-conditioning problems in the Q-matrix inversion.

4. Concluding remarks

The use of irreducible representations of finite groups is a very general and powerful method for reducing ill-conditioning problems in T-matrix computations. The main practical problem in using the method is to obtain the irreducible characters. Tabulated characters can be found in the literature only for low-order point groups, which has severely limited the practical usefulness of the method to particles with low-order symmetries. The main goal of the present study was to substantially extend the applicability of group theoretical methods in T-matrix computations by overcoming these limitations.

The characters of high-order point groups need to be obtained by methods of computational group theory. This paper discussed the use of the *GAP* programming system for calculating the irreducible characters of high-order point groups. The *GAP* character tables have been coupled to the T-matrix code *Tsym* by analysing the conjugacy classes of the tables. The use of *GAP* within *Tsym* has been automatised. Further, different methods, including singular value decomposition for general and sparse matrices, have been discussed and tested for constructing the transformation into the irreducible basis.

Applications of *Tsym* in conjunction with *GAP* character tables to high-order 3D-Chebyshev particles with small-scale surface roughness showed that the numerical stability of the method can compete with that of the iterative Lippmann-Schwinger approach. The latter completely removes the ill-conditioning problems for Chebyshev particles with spherical base geometries. This clearly demonstrates the increase in numerical stability that can be achieved by the group theoretical method. While the iterative approach may suffer from slow-convergence problems and is therefore limited to particles with small-scale surface roughness, the group theoretical method is not restricted by any assumptions about the particle morphology other than very general symmetry assumptions.

A close analysis of the results obtained for high-order Chebyshev particles indicates that numerical problems encountered when the deformation parameter ϵ becomes too large are likely to be caused by error sources other than ill-conditioning of the Q-matrix inversion. One possible source of error may be loss of precision in the evaluation of the surface integrals in Waterman's T-matrix method, as suggested in [16, 17].

In summary, the group theoretical method successfully stabilises the numerical Q-matrix inversion and achieves a robustness comparable to that of the iterative method. In the latter the ill-conditioning problem has been traded for a potential slow-convergence problem. No such trade-off has to be made in the group theoretical approach. Thus the method is very general and fast. Further numerical progress in T-matrix computations may strongly depend on successfully addressing loss-of-precision problems associated with the calculation of the surface integrals.

Appendix

T-matrix formulation of electromagnetic scattering

To describe scattering of electromagnetic radiation by a homogeneous particle one needs to solve the vector Helmholtz equation by use of the boundary conditions. A general solution to the vector Helmholtz equation in spherical coordinates can be expanded in vector spherical wave functions $\vec{\Psi}_{\nu}^{(j)}$. Assuming (and suppressing) a harmonic time-dependency, the incident, scattered, and internal fields have the general form

$$\begin{aligned} \mathbf{E}^{\rm inc}(\mathbf{r}) &= \sum_\nu a_\nu \vec{\Psi}_\nu^{(1)}(k,\mathbf{r}) & (22)\\ \mathbf{E}^{\rm sca}(\mathbf{r}) &= \sum_\nu p_\nu \vec{\Psi}_\nu^{(3)}(k,\mathbf{r}) & (23)\\ \mathbf{E}^{\rm int}(\mathbf{r}) &= \sum_\nu c_\nu \vec{\Psi}_\nu^{(1)}(k_s,\mathbf{r}), & (24) \end{aligned}$$

where k and k_s are the wavenumbers in the surrounding medium and inside the particle, respectively. The vector spherical wave functions of the first kind, $\vec{\Psi}_\nu^{(1)}$, are regular at the origin, while those of the third kind, $\vec{\Psi}_\nu^{(3)}$, satisfy the radiation condition. The superindex ν represents a triple index n, m, τ, where n is the degree, m the order, and τ the mode of the vector spherical wave functions. In the absence of free charges, the fields have to be divergence-free, thus transverse. The summation over the mode index τ therefore only extends over the two transverse modes. The boundary conditions for dielectric particles state that the tangential component of the fields need to be continuous across the particle boundary, i.e.

$$\hat{n} \times (\mathbf{E}^{\rm inc} + \mathbf{E}^{\rm sca} - \mathbf{E}^{\rm int})\big|_{\rm boundary} = \mathbf{0}, \qquad (25)$$

where $\hat{n}$ is the outward pointing normal vector on the particle surface. We have a similar expansion and analogous boundary conditions for the magnetic fields $\mathbf{H}^{\rm inc}$, $\mathbf{H}^{\rm sca}$, and $\mathbf{H}^{\rm int}$.

Since the boundary conditions are linear, one obtains linear relations among the expansion coefficients of the fields,

$$\begin{aligned} \mathbf{a} &= \mathbf{Q}\cdot\mathbf{c} & (26)\\ \mathbf{p} &= -Rg\mathbf{Q}\cdot\mathbf{c} & (27)\\ \mathbf{p} &= \mathbf{T}\cdot\mathbf{a}. & (28) \end{aligned}$$

The quantity of interest is the T-matrix, which allows us to compute for any given incident field (characterised by its expansion coefficients a_ν) the scattered field expansion coefficients p_ν. The T-matrix contains the complete information about a particle's absorption and scattering properties. All other quantities of interest, such as differential scattering cross sections and phase functions, the Mueller matrix and its moment expansions, as well as total scattering and absorption cross sections can be computed from the T-matrix (e.g. [18]).

In Waterman's null-field method [2] one derives surface integral expressions for the elements of the matrices $\mathbf{Q}$ and $Rg\mathbf{Q}$, from which one can compute the desired T-matrix using

$$\mathbf{T} = -Rg\mathbf{Q} \cdot \mathbf{Q}^{-1}, \tag{29}$$

which follows from Eqs. (26)–(28). The matrix inversion in Eq. (29) can introduce numerical ill-conditioning problems in the computation of the T-matrix. Much efforts have been invested into reducing such ill-conditioning problems. For instance, the use of extended precision variables [19] and suitable numerical matrix inversion routines [20] have been discussed. For particles with discrete geometrical symmetries one can use irreducible representations of finite groups to alleviate the ill-conditioning problems [1]. One main focus of the present paper is the application of this group theoretical method to particles with high-order symmetries.

For particles with small-scale surface roughness, a perturbation approach has recently been proposed [12]. The idea is to consider the Q-matrices of the perturbed and unperturbed geometries, $\mathbf{Q}$ and $\mathbf{Q}_0$, respectively, and to formally introduce the difference $\Delta\mathbf{Q} = \mathbf{Q} - \mathbf{Q}_0$. Substitution into Eq. (29) yields, after rearranging some terms, a Lippmann-Schwinger equation of the form

$$\mathbf{T} = -(Rg\mathbf{Q} + \mathbf{T} \cdot \Delta\mathbf{Q}) \cdot \mathbf{Q}_0^{-1}. \tag{30}$$

A Lippmann-Schwinger equation can be solved iteratively according to the scheme

$$\begin{aligned} \mathbf{T}^{(0)} &= -Rg\mathbf{Q} \cdot \mathbf{Q}_0^{-1} & (31) \\ \mathbf{T}^{(p)} &= -(Rg\mathbf{Q} + \mathbf{T}^{(p-1)} \cdot \Delta\mathbf{Q}) \cdot \mathbf{Q}_0^{-1}, \quad p \geq 1. & (32) \end{aligned}$$

The advantage of this scheme is that one only needs to invert the Q-matrix $\mathbf{Q}_0$ of the unperturbed geometry, which is usually much more well-conditioned than inversion of $\mathbf{Q}$. In fact, for a rough particle with spherical base geometry, such as the one in Fig. 1, $\mathbf{Q}_0$ is a diagonal matrix, so the inversion becomes trivial. A further advantage of this method is that it is very easy to implement into an existing T-matrix code, unlike the method based on irreducible representations. However, little is known about the convergence of the iteration scheme. The method seems to be limited to particles with small-scale surface roughness. A comparative study of the group theoretical method and the iterative solution of the Lippmann-Schwinger equation is presented in Sect. 3.

A short review of groups and representations

Group theory can be used in three different ways in T-matrix computations. (i) One can reduce the number of non-zero, independent T-matrix elements that actually need to be computed, thus expediting the computations [3, 21]. (ii) In Waterman's null-field method, one can reduce the surface area over which the integration needs to be carried out, which, again, reduces CPU time [3]. (iii) One can exploit the irreducible representations of groups to reduce ill-conditioning problems in the inversion of the Q-matrix [1]. This is achieved by bringing the Q-matrix into block-diagonal form. In this paper, the focus is on item (iii). This section introduces the essential terminology used throughout this paper. It is not assumed that the reader has any prior knowledge of group theory. I chose brevity over completeness, and I tried to choose didactic clarity over mathematical rigour.

Groups: The symmetries of particles are characterised by symmetry elements such as rotation axes or reflection planes. So the symmetry *elements* are geometric objects, such as lines and planes. The corresponding coordinate transformations (rotations, reflections, or rotation-reflections) are called symmetry *operations.* The set of all symmetry operations of a particle forms a group.

More formally, a group $(\mathcal{G}, \bullet)$ consists of a set $\mathcal{G}$ and a binary operation $\bullet : \mathcal{G} \times \mathcal{G} \to \mathcal{G}$ (the group multiplication) with the following properties. (i) There exists a neutral element E with respect to the binary operation, i.e. $g \bullet E = E \bullet g = g$ for all group elements $g \in \mathcal{G}$. (ii) Each group element g possesses an inverse element g^{-1}, i.e. $g \bullet g^{-1} = g^{-1} \bullet g = E$. (iii) The binary operation is associative, i.e. $g_1 \bullet (g_2 \bullet g_3)$=$(g_1 \bullet g_2) \bullet g_3$ for all $g_1, g_2, g_3 \in \mathcal{G}$.

Point groups: In light scattering by particles with discrete symmetries, the groups we are interested in are known as point groups. The term derives from the fact that all symmetry operations leave one point in space (the centre of the particle) unchanged. If this were not the case, the group would also contain translations, which cannot be symmetry elements of finite particles. All point groups are subgroups of the orthogonal group $\mathcal{O}(3)$.

There exist infinite point groups (such as the $\mathcal{K}$ group of spherical symmetry, and the groups $\mathcal{D}_{\infty h}$ and $\mathcal{C}_{\infty v}$ of axisymmetric particles) as well as finite point groups. The focus in this paper is on the latter. Finite point groups contain discrete rotational or no rotational symmetry operations, and they may also contain reflection and discrete rotation-reflection operations. The number of symmetry operations in a group is called the order of a group M_o. Finite point groups have finite orders.

Examples: As an example, consider the regular triangle shown in Fig. 2 (left). A rotation by $2\pi/3$ about an axis through the centre of and perpendicular to the triangle brings the object into a new orientation in which it is indistinguishable from its original position. We denote this rotation by C_3. The same is true for a rotation C_3^2 by an angle $2 \times 2\pi/3$. The operations E, C_3, and C_3^2 form a point group denoted by $\mathcal{C}_3$. Now consider the regular triangular prism in Fig. 2 (right). Apart from the operations E, C_3, and C_3^2, we can identify more symmetries. There are three so-called dihedral rotational symmetry axes passing through the centre of the prism and perpendicular to the main rotational axis. Flipping the particle by an angle π about any of these axes is a symmetry operation, denoted by $C_2^{\prime(j)}$, $j = 0, 1, 2$. Further, the particle possesses four reflection planes. There is one "horizontal" reflection plane parallel to the triangular top and bottom faces and passing through the centre of the prism. The corresponding symmetry operation is denoted by σ_h. There are also three "vertical" reflection planes, each containing one of the vertical edges of the prism and intercepting the opposite face at a right angle. The corresponding operations are denoted by $\sigma_v^{(j)}$, $j = 0, 1, 2$. Finally, we can combine the rotations C_3^j $(j = 1, 2)$ with σ_h to obtain the rotation-reflection operations $S_3 = \sigma_h \bullet C_3$ and $S_3^{(2)} = \sigma_h \bullet C_3^2$. In total, there are 12 symmetry operations. The corresponding group is denoted by $\mathcal{D}_{3h}$.

Generators: A quick inspection reveals that we can generate all operations in the group $\mathcal{C}_3$ from the operation C_3, i.e. $C_3^2 = C_3 \cdot C_3$, and $E = C_3 \cdot C_3 \cdot C_3$. The operation C_3 is therefore called a generator of the group. The group $\mathcal{C}_3$ has only one generator. However, the choice of the generator is not unique; we could also choose C_3^2 as the generator of the group (since $C_3 = C_3^2 \cdot C_3^2$, and $E = C_3^2 \cdot C_3^2 \cdot C_3^2$). The group $\mathcal{D}_{3h}$ has three generators. One possible choice for the generators is C_3, $C_2^{\prime(0)}$, and σ_h. (In fact, this choice is suggested by the notation we use for denoting the point groups — $\mathcal{D}$=dihedral, 3=C_3, h=σ_h.)

Representations: At this point we only have some abstract ideas about symmetry operations. But in order to work with symmetries we need to know how the symmetry operations actually act on the elements of a specific vector space. For instance, in three-dimensional Euclidean space $I\!R^3$ a C_3

rotation about the z-axis can be represented by the regular matrix

$$\left(\begin{array}{lll} \cos 2\pi/3 & \sin 2\pi/3 & 0 \\ -\sin 2\pi/3 & \cos 2\pi/3 & 0 \\ 0 & 0 & 1 \end{array} \right). \tag{33}$$

In the vector space of the coefficient vectors in Eqs. (26)–(29), which is the space on which the Q- and T-matrices operate, it turns out that C_3 can be represented by a unitary, diagonal matrix with elements $\exp(2\pi \mathrm{i} m/3)$ (where the integer m comes from the order of the vector spherical wave function expansion of the fields). More generally, a representation $\mathbf{D}$ can be thought of as a map from the group $\mathcal{G}$ into a group of linear bijective operations ("automorphisms") that operate on a vector space. Automorphisms can be represented by regular (i.e. invertible) matrices. A representation of a group has the property

$$\mathbf{D}(g_1 \bullet g_2) = \mathbf{D}(g_1) \cdot \mathbf{D}(g_2) \tag{34}$$

for all $g_1, g_2 \in \mathcal{G}$, where "$\cdot$" denotes matrix multiplication. More detailed explanations on how to derive the representations of point groups in the space of vector spherical wave functions can be found in [21]. A complete list of representations of non-cubic point groups is given in [1].

Invariant subspaces and irreducible representations: Consider the example given in (33). When operating on a vector (x, y, z) this matrix "mixes" the x and y components, but it does not mix x and z or y and z components. So the subspace spanned by the x and y unit vectors is invariant under the C_3 operation. Also, the z axis is invariant under the C_3 rotation. The matrix in (33) is composed of two block matrices, one (2×2) matrix, and one (1×1) matrix

$$\left(\begin{array}{cc} \left(\begin{array}{cc} \cos 2\pi/3 & \sin 2\pi/3 \\ -\sin 2\pi/3 & \cos 2\pi/3 \end{array} \right) & \mathbf{0} \\ \mathbf{0} & (1) \end{array} \right), \tag{35}$$

which we write symbolically as

$$\left(\begin{array}{cc} \mathbf{D}^{(1)}(C_3) & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^{(2)}(C_3) \end{array} \right) = \mathbf{D}^{(1)}(C_3) \oplus \mathbf{D}^{(2)}(C_3). \tag{36}$$

The matrix

$$\mathbf{D}^{(1)}(C_3) = \left(\begin{array}{cc} \cos 2\pi/3 & \sin 2\pi/3 \\ -\sin 2\pi/3 & \cos 2\pi/3 \end{array} \right) \tag{37}$$

is a representation of C_3 in two-dimensional Euclidean space $I\!R^2$. Thus we can reduce the representation $\mathbf{D}$ in three-dimensional space into one representation $\mathbf{D}^{(1)}$ in two-dimensional space, and another representation $\mathbf{D}^{(2)}$ in one-dimensional space.

More generally, consider a vector space V, a representation of a group $\mathbf{D}(g)$, and a subspace $T \subset V$. If $\mathbf{D}(g) \cdot \mathbf{x} \in T$ for all group elements $g \in \mathcal{G}$ and for all vectors $\mathbf{x} \in T$, i.e., if all of the represented group elements transform the elements of T only among themselves, then T is called an invariant subspace of V. Any vector space V possesses two trivial invariant subspaces, $T = V$ and $T = \{\vec{0}\}$, where $\vec{0}$ denotes the null vector. If V possesses any proper (i.e. non-trivial) invariant subspaces, then $\mathbf{D}$ is called a reducible representation of $\mathcal{G}$. Otherwise, $\mathbf{D}$ is called an irreducible representation of $\mathcal{G}$. For instance, for the group $\mathcal{C}_3$, $I\!R^2$ is an invariant subspace of $I\!R^3$. Thus $\mathbf{D}(C_3)$ in (33) is a reducible representation, while $\mathbf{D}^{(1)}(C_3)$ in (37) is an irreducible representation.

One can reduce a reducible representation by finding the invariant, irreducible subspaces T_μ, $\mu = 1, \ldots, r$, where r denotes the number of invariant subspaces. The invariant subspaces are disjoint, and the union of these spaces is the entire vector space V, i.e.

$$T_\mu \cap T_\nu \;=\; \emptyset, \,\forall \mu \neq \nu \tag{38}$$

$$\bigcup_{\mu=1}^{r} T_\mu = V. \tag{39}$$

If we specify a basis in each subspace T_μ, then the union of these bases forms a basis of the entire vector space V. In this basis, the representations of the group have block-diagonal form, such as in Eq. (36). $\mathbf{D}(g)$ has the same block-diagonal form for all group element $g \in \mathcal{G}$. As a generalisation of (36), we write the reduction of a representation $\mathbf{D}$ symbolically as a direct sum

$$\mathbf{D} = \bigoplus_{\mu=1}^{r} \alpha_\mu \mathbf{D}^{(\mu)} \tag{40}$$

where $\mathbf{D}^{(\mu)}$ denotes the μth irreducible representation. An irreducible representation can, in fact, occur more than one time in a reducible one. This is indicated by the cardinal numbers α_μ.

To summarise, the problem of finding the irreducible representations can be solved by finding the invariant subspaces. That problem, in turn, essentially amounts to finding a transformation of the basis of the vector space into the irreducible basis, i.e., into a set of basis vectors that is composed of the bases of each invariant irreducible subspace.

Block-diagonalisation of the Q-matrix: It turns out that the vector spherical wave functions are, in general, a reducible basis of the function space (except for the spherical group $\mathcal{K}$). What is the benefit of abandoning this basis and searching for the irreducible basis? It can be shown [1] that in that basis not only the representation $\mathbf{D}$, but also the Q- and T-matrices become block-diagonal. Thus, rather than inverting one giant Q-matrix, one can reduce the problem to inverting only the smaller block matrices of which the Q-matrix in the irreducible basis is composed. This has been shown to significantly reduce ill-conditioning problems in the matrix inversion [1]. The number of block matrices is equal to the number of irreducible representations, which is equal to the number of invariant subspaces. For instance, a 3D Chebyshev particle of even order n belongs to the group $\mathcal{D}_{nh}$. The order of this group is $M_o = 4n$, and the number of irreducible representations is $r = n + 6$. Thus for high-order Chebyshev particles the Q-matrix is reduced to a large number of substantially smaller block matrices, which will significantly reduce ill-conditioning problems in the inversion of the Q-matrix.

Characters: The key to finding the irreducible basis is to obtain the irreducible characters $\chi^{(\mu)}$ of the group. They are defined as the traces of the irreducible representations.

$$\chi^{(\mu)}(g) = Tr\mathbf{D}^{(\mu)}(g) = \sum_i D^{(\mu)}_{i,i}(g), \tag{41}$$

where the sum extends over all dimensions of the μth invariant subspace. Remarkably enough, it is possible to compute the irreducible characters without prior knowledge of the irreducible representations or the invariant subspaces [8, 9]. Applications of such methods of computational group theory to T-matrix computations are discussed in Sect. 2.2.

Projection operators: Let us define the following operators:

$$\tilde{\mathbf{P}}^{(\mu)} = \sum_{g\in\mathcal{G}} \chi^{(\mu)*}(g)\mathbf{D}(g). \tag{42}$$

The summation extends over all elements of the group. It can be shown [1] that these matrices are projection operators that project any arbitrary vector $\mathbf{x}$ into the μth invariant subspace, i.e. $\tilde{\mathbf{P}}^{(\mu)} \cdot \mathbf{x} \in T_\mu$. Once we have the projectors, we can use them to construct the desired transformation that transforms the basis of the vector space into the irreducible basis. The details of how this is done are discussed in Sect. 2.3. To compute the projectors, we need the reducible representations $\mathbf{D}(g)$ and the irreducible characters $\chi^{(\mu)}(g)$. The computation of the reducible representations has been presented in [1, 21]. The computation of the characters is discussed in Sect. 2.2.

Conjugacy classes: The characters $\chi^{\mu}(g)$ are usually listed in a so-called character table. How many rows and columns does a character table have? For each g there are r characters $\chi^{(1)}(g), \ldots, \chi^{(r)}(g)$, each belonging to one of the irreducible representations/invariant subspaces. These are listed in the rows of the character table, so there are r rows in total. One may expect that there are M_o columns, since for each μ there are M_o characters $\chi^{\mu}(g)$ (where M_o is the order of the group). However, it turns out that only r of these characters are different. Thus there are only r columns in the table, each corresponding to one of the so-called conjugacy classes of the group.

Two group elements $g_1, g_2 \in \mathcal{G}$ are called conjugate to each other (symbolically $g_1 \sim g_2$), if there exists a third group element $h \in \mathcal{G}$ such that $g_1 = h^{-1} \bullet g_2 \bullet h$. Mathematically speaking, conjugacy is an equivalence relation. This means that it is (i) reflexive, i.e. $g \sim g$ for all g, (ii) symmetric, i.e. $g_1 \sim g_2 \Rightarrow g_2 \sim g_1$ for all g_1, g_2, and (iii) transitive, i.e. $g_1 \sim g_2$ and $g_2 \sim g_3 \Rightarrow g_1 \sim g_3$ for all g_1, g_2, g_3. An equivalence relation can be used for defining equivalence classes, viz.

$$[g]_{\sim} := \left\{ g' \in \mathcal{G} \mid \exists h \in \mathcal{G};\ g' = h^{-1} \bullet g \bullet h \right\}. \tag{43}$$

The equivalence class $[g]_{\sim}$ is called a conjugacy class of the group. It contains all group elements that are conjugate to g. The group element g is called a representative of the class $[g]_{\sim}$. Any other element that is conjugate to g could also be used to represent the class, i.e. if $g \sim g_1$, then $[g]_{\sim} = [g_1]_{\sim}$. One can show that the number of conjugacy classes in a group is equal to the number r of invariant subspaces, thus to the number of irreducible representations.

All group elements belonging to the same conjugacy class have the same characters. This can be shown as follows. Let $g_1 \sim g_2$, then

$$\begin{aligned} \chi^{(\mu)}(g_1) &= Tr(\mathbf{D}^{(\mu)}(g_1)) = Tr(\mathbf{D}^{(\mu)}(h^{-1} \bullet g_2 \bullet h)) = \\ &= Tr(\mathbf{D}^{(\mu)}(h^{-1}) \cdot \mathbf{D}^{(\mu)}(g_2) \cdot \mathbf{D}^{(\mu)}(h)) = Tr(\mathbf{D}^{(\mu)}(g_2)) \end{aligned} \tag{44}$$

where we used the property (34). The last step uses the fact that the factors in the argument of the trace can be permuted, and that $\mathbf{D}^{(\mu)}(h^{-1}) =$

$\mathbf{D}^{(\mu)-1}(h)$. The last term in Eq. (44) is equal to $\chi^{(\mu)}(g_2)$, thus proving that all elements in a conjugacy class have the same characters. To summarise, the irreducible characters of a group are listed in a table with r columns and r rows. The rows correspond to the r irreducible representations, while the columns correspond to the r conjugacy classes of the group.

Number of times an irreducible representation occurs in a reducible one: For completeness, a theorem is stated here without proof that is used in Sect. 2.3. The proof can be found, e.g., in [7]. The cardinal numbers α_μ in Eq. (40) can be computed as follows.

$$\alpha_\mu = M_o^{-1} \sum_{i=1}^{r} M_i \chi^{\mathrm{red}}(g_i) \chi^{(\mu)*}(g_i). \tag{45}$$

Here M_i denotes the number of group elements in the ith conjugacy class, χ^{red} are the characters of the *reducible* representations, and g_i is a representative of the ith conjugacy class. In [1] a complete list of all non-cubic finite point groups is given. For each group the order M_o and the number of conjugacy classes r is given, and the conjugacy classes and the number of elements M_i in each class are listed. Also, explicit expressions of the reducible representations are given in the form of unitary matrices, from which χ^{red} can be computed. With this information, computation of α_μ is rather straightforward, provided that we have the irreducible characters $\chi^{(\mu)}$.

Knowledge of the α_μ is helpful for the computation of the transformation into the irreducible basis. If we multiply α_μ by the dimension of the μth irreducible representation, then we get the dimension of the μth invariant subspace. (The dimension of the μth irreducible representation is simply the trace of the unit element E in that representation, i.e. $\chi^{(\mu)}(E)$.) The dimension of the μth invariant subspace is equal to the dimension of the range of the projector $\tilde{\mathbf{P}}^{(\mu)}$. This information is needed in Sect. 2.3 to construct the transformation to the irreducible basis from the projection matrices.

[1] M. Kahnert, Irreducible representations of finite groups in the T matrix formulation of the electromagnetic scattering problem, J. Opt. Soc. Am. A 22 (2005) 1187–1199.

[2] P. C. Waterman, Matrix formulation of electromagnetic scattering, Proc. IEEE 53 (1965) 805–812.

[3] F. M. Kahnert, J. J. Stamnes, K. Stamnes, Application of the extended boundary condition method to homogeneous particles with point group symmetries, Appl. Opt. 40 (2001) 3110–3123.

[4] M. Kahnert, Boundary symmetries in linear differential and integral equation problems applied to the self-consistent Green's function formalism of acoustic and electromagnetic scattering, Opt. Commun. 265 (2006) 383–393.

[5] M. Kahnert, T. Nousiainen, P. Mauno, On the impact of non-sphericity and small-scale surface roughness on the optical properties of hematite aerosols, J. Quant. Spectrosc. Radiat. Transfer 112 (2011) 1815–1824.

[6] M. Kahnert, T. Nousiainen, M. A. Thomas, J. Tyynelä, Light scattering by particles with small-scale surface roughness: comparison of four classes of model geometries, J. Quant. Spectrosc. Radiat. Transfer (2012) http://dx.doi.org/10.1016/j.jqsrt.2012.03.017.

[7] D. M. Bishop, Group Theory and Chemistry, Dover Publications, Mineola, 1993.

[8] J. D. Dixon, High speed computation of group characters, Numerische Mathematik 10 (1965) 446–450.

[9] J. J. Cannon, Computers in group theory: a survey, Commun. ACM 12 (1969) 3–11.

[10] The GAP Group, GAP – Groups, Algorithms, and Programming, Version 4.5.5 (2012).
URL http://www.gap-system.org

[11] R. M. Larsen, Lanczos bidiagonalization with partial reorthogonalization, Tech. Rep. DAIMI PB-357, Department of Computer Science, Aarhus University (1998).

[12] M. Kahnert, T. Rother, Modeling optical properties of particles with small-scale surface roughness: combination of group theory with a perturbation approach, Opt. Express 19 (2011) 11138–11151.

[13] T. Rother, Electromagnetic wave scattering on nonspherical particles, Springer, Berlin, 2009.

[14] K. Schmidt, M. Yurkin, M. Kahnert, A case study on the reciprocity in light scattering computations, Opt. Express(submitted).

[15] M. Kahnert, T. Nousiainen, H. Lindqvist, M. Ebert, Optical properties of light absorbing carbon aggregates mixed with sulfate: assessment of different model geometries for climate forcing calculations, Opt. Express 20 (2012) 10042–10058.

[16] P. C. Waterman, The T-matrix revisited, J. Opt. Soc. Am. A 24 (2007) 2257–2267.

[17] W. R. C. Somerville, B. Auguié, E. C. L. Ru, Severe loss of precision in calculations of the T-matrix integrals, J. Quant. Spectrosc. Radiat. Transfer 113 (2012) 524–535.

[18] M. I. Mishchenko, , L. D. Travis, A. A. Lacis, Scattering, absorption, and emission of light by small particles, Cambridge University Press, Cambridge, 2002.

[19] M. I. Mishchenko, L. D. Travis, T-matrix computations of light scattering by large spheroidal particles, Opt. Commun. 109 (1994) 16–21.

[20] D. J. Wielaard, M. I. Mishchenko, A. Macke, B. E. Carlson, Improved T-matrix computations for large, nonabsorbing and weakly absorbing nonspherical particles and comparison with geometrical-optics approximation, Appl. Opt. 36 (1997) 4305–4313.

[21] F. M. Schulz, K. Stamnes, J. J. Stamnes, Point group symmetries in electromagnetic scattering, J. Opt. Soc. Am. A 16 (1999) 853–865.

Acknowledgements

Funding from the Swedish Research Council (projects 621-2008-4387 and 621-2011-3346) is gratefully acknowledged. The author thanks the developers of the Groups Algorithms and Programming (GAP) system for making their computational group theory program publicly available.