

Counting Cases in Substotope Algorithms

David C. Banks, Stephen A. Linton, and Paul K. Stockmeyer

Abstract—We describe how to count the cases that arise in a family of visualization techniques, including Marching Cubes, Sweeping Simplices, Contour Meshing, Interval Volumes, and Separating Surfaces. Counting the cases is the first step toward developing a generic visualization algorithm to produce substotopes (geometric substitutions of polytopes). We demonstrate the method using “GAP,” a software system for computational group theory. The case-counts are organized into a table that provides a taxonomy of members of the family; numbers in the table are derived from actual lists of cases, which are computed by our methods. The calculations confirm previously reported case-counts for four dimensions that are too large to check by hand and predict the number of cases that will arise in substotope algorithms that have not yet been invented. We show how Pólya theory produces a closed-form upper bound on the case counts.

Index Terms—Isosurface, level set, group action, orbit, geometric substitution, Marching Cubes, separating surface, Pólya counting, substotope.

1 A FAMILY OF VISUALIZATION ALGORITHMS

THE Marching Cubes (MC) algorithm was presented by Lorensen and Cline in 1987 as an exhaustive-search technique that generates a level set (isosurface) of a scalar function [1]. The algorithm iterates over each cube tessellating a compact subvolume of \mathbb{R}^3 on which the function f is defined. The sign of $f(v_i) - \text{const}$ is evaluated at the eight vertices v_i of a cube, where const is some user-defined constant (the isovalue). Neglecting the degenerate case where the sign is exactly zero, each of the eight vertices can be in one of two states: positive or negative (black or white). These produce $2^8 = 256$ patterns. Many of these patterns turn out to be equivalent under the symmetries of the cube (such as rotation or mirror-reflection). Other patterns are equivalent under reversal of colors (for example, all-black being equivalent to all-white). Through patient brute-force organization of the 256 patterns, one discovers there to be 14 or 15 equivalence classes of the colorings. Among the 15 cases is a chiral pair that are mirror images of each other, so these two are equivalent if orientation is ignored.

In the MC algorithm, the pattern of a given cube is matched to one of these 14 or 15 cases and a predetermined arrangement of polygons is fitted to meet the constraint $f(v_i) - \text{const} = 0$ for points p along edges of the cube. In other words, the cube is replaced by zero or more triangles approximating the level set. Fig. 2 (bottom right) shows one case of a geometric substitution that occurs in MC. The figure illustrates how substitution proceeds in variations of the basic MC algorithm: An n -simplex or n -cube is replaced

by a polytope. The n -cube, with 2^n vertices, generalizes the sequence *point, segment, square, cube*; the n -simplex, with $n + 1$ vertices, generalizes the sequence *point, segment, triangle, tetrahedron*; a polytope generalizes the sequence *point, segment, polygon, polyhedron* [2]. Polytopes that result from a geometric substitution are called *substotopes* [3]. Fig. 2 illustrates how substitution proceeds in variations on the basic MC algorithm: In general, an n -simplex or n -cube is replaced by a substotope. The following section surveys several substotope algorithms that have been developed in recent years.

1.1 Survey of Substotope Algorithms

MC has inspired numerous modifications and extensions. These variations reveal a family of algorithms whose members are distinguished by a few key parameters. A selection of these substotope algorithms is surveyed below.

Variation of the shape. If a 3-simplex (tetrahedron), rather than a 3-cube, tiles the domain, then the scalar function is evaluated at only four vertices [4], [5]. One advantage of using tetrahedra rather than cubes is that their analysis is simpler: Only three cases arise for the vertex colorings of a tetrahedron, rather than 14 for a cube. Fig. 2 (top right) shows one of these three cases.

Variation of the dimension. The two-dimensional version of MC is popularly called “Marching Squares,” which provides a simple motivation for the three-dimensional case. Although the algorithm is unpublished, it can be easily derived or found by searching the World Wide Web. Fig. 2 (bottom left) shows one of its cases.

When the MC algorithm is extended to dimension $n = 4$, two problems arise. First, the number of vertex patterns is large (65,536), so enumerating them all by hand is unrealistic. Second, it is difficult to perform the mental rotations necessary to decide when two color patterns of a 4-cube are equivalent. As Lorensen and Cline pointed out in the case of the 3-cube, “triangulating the 256 cases is possible but error prone. . . . [W]e reduced the problem to 14 patterns by inspection [p. 165].” Although this approach to counting cases works for Marching Squares and MC, it does not scale to dimensions higher than three. Recently, several researchers have tackled the case-counting problem

- D.C. Banks is with the Department of Computer Science, Florida State University, Tallahassee, FL 32306. E-mail: banks@csit.fsu.edu.
- S.A. Linton is with the School of Computer Science, University of St. Andrews, North Haugh, St. Andrews, Fife, KY16 9SS, UK. E-mail: sal@dcs.st-and.ac.uk.
- P.K. Stockmeyer is with the Department of Computer Science, College of William and Mary, PO Box 8795, Williamsburg, VA 23187-8795. E-mail: stockmeyer@cs.wm.edu.

Manuscript received 26 Sept. 2003; revised 19 Dec. 2003; accepted 29 Jan. 2003.

For information on obtaining reprints of this article, please send e-mail to: tcvg@computer.org, and reference IEEECS Log Number TVCGSI-0087-0903.

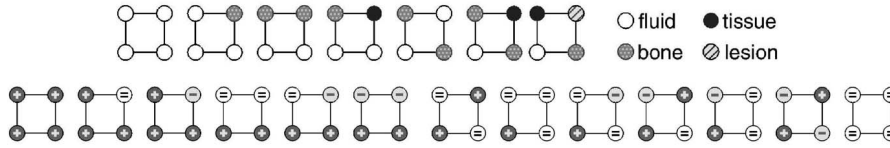


Fig. 1. Distinct cases of colorings of a square. Top row: Seven cases result from using four colors (fluid, bone, tissue, lesion) when applying Separating Surfaces to a square. Bottom row: Thirteen cases result from using three colors ($- = +$) with Marching Squares.

for the 4-cube in different ways, counting 496 cases (Kirchberg [6]), 272 cases (Roberts and Hill [7]), or 222 cases (Bhaniramka et al. [8]) that arise. Section 3 confirms these counts and explains why they are different.

Variation of the shape’s symmetry. The symmetries of a figure are due to transformations that preserve its shape and, perhaps, orientation. If one considers orientation (clockwise versus counterclockwise) to be irrelevant, one loses distinctions between certain colorings of squares. So, the number of cases depends in part on the choice one makes when considering symmetries of the shape. The original MC deals with two cases that form a chiral pair: One is the mirror image of the other. The problem reduces to 15 cases under orientation-preserving symmetries and to 14 cases under the larger symmetry group that includes mirror reflection.

An analogous situation exists in every dimension—the group of orientation-preserving symmetries is, in general, only half as large as the group of all symmetries. So, there may be figures that are equivalent under the larger group but not the smaller one. This is the reason that Roberts and Hill found a larger number of cases than Bhaniramka et al. (272 versus 222 cases)—they were employing a smaller symmetry group.

Variation of the number of colors. MC produces a level set by consulting a 2-colored function. But, substitope algorithms can also employ multiple colors, as was done in Interval Volumes [9], Contour Meshing [10], and Generalized MC [11], [12], [13], [14]. Fig. 3 (top row) shows an example of the geometric substitution in Interval Volumes for a 2-simplex and a 3-simplex with three colors.

Fig. 1 (bottom row) illustrates the 13 cases that result from extending Marching Squares from two colors to three, corresponding to $f < 0$, $f = 0$, $f > 0$. These cases can be easily deduced through brute-force inspection of the $3^4 = 81$ ways that three colors can be assigned to four vertices, then by organizing the colorings into equivalence classes. Until now there has been no published case count for MC with this third color included, perhaps because the $3^8 = 6,561$ colorings are

too numerous to catalog by hand. We calculate the solution and report it in Section 3.3 as one of several new examples of case counts.

Variation of the colors’ symmetry. Two colors may be considered equivalent according to various choices of a symmetry group acting on the set of colors [15]. The natural choices for the symmetries are 1) the identity (as in Kirchberg’s Marching Hypercubes), 2) reversal of colors (as in MC and Interval Volumes), or 3) all permutations of colors (as in Separating Surfaces [16], which considers all $4!$ permutations of colors at the four vertices of a tetrahedron). Fig. 3 (bottom row) shows an example of the geometric substitution in Separating Surfaces for a 2-simplex and a 3-simplex with three colors. Fig. 1 (top row) shows the cases that would arise if Separating Surfaces, which considers any permutation of colors to be equivalent, were extended to operate on squares using four colors with all $4!$ permutations of colors allowed.

1.2 Substotope Algorithm

The basic substitope algorithm consults a function f at each vertex of polytopes tiling a domain; the resulting substitopes are designed to create a visualization of the function rather than of the tiles. The *substitope algorithm* considers both the tiles and their colorings. This algorithm includes MC and all its variants. The steps of the algorithm are sketched below.

1. Traverse tiles in a region.
2. Assign a color to each vertex of a tile.
3. Match the coloring to a representative case.
4. Substitute polytopes for the colored tile.

The algorithm can be accelerated by skipping over the trivial substitutions [17], [18], [19], [20] rather than employing an exhaustive traversal of the domain in Step 1.

Some, but not all, of the variations on MC precompute a look-up table in Step 3, which serves as an acceleration technique when the geometric substitution is applied in Step 4. But, the geometric substitution can also be performed procedurally, for example, when the dimension

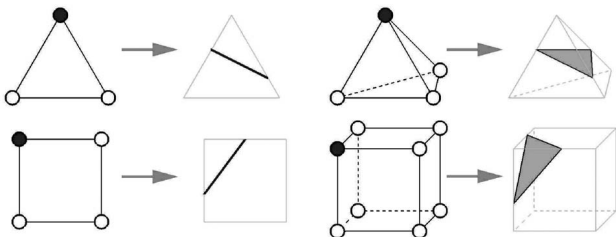


Fig. 2. Examples of geometric substitution rules in Marching Cubes, generalized to the n -simplex and n -cube, for $n \in \{2,3\}$. The colors correspond to sign of $f(v_i) - \text{const}$ at each vertex, such as black for positive and white for negative. Top row: 2-simplex replaced by line segment; 3-simplex replaced by triangle. Bottom row: 2-cube replaced by line segment; 3-cube replaced by triangle.

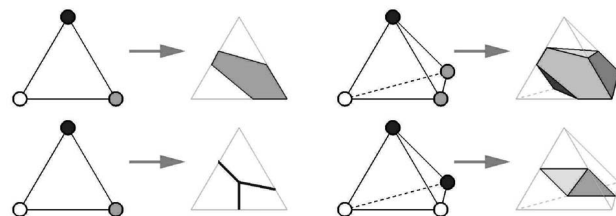


Fig. 3. Examples of substitopes. Top row (Interval Volumes): 2-simplex replaced by polygon; 3-simplex replaced by polyhedron. Colors denote intervals such as white for $(-\infty, a)$, gray for (a, b) , and black for (b, ∞) . Bottom row (Separating Surfaces): 2-simplex replaced by branching line segments; 3-simplex replaced by triangles. Colors denote membership in a set, e.g., $\{\text{fluid, bone, tissue, lesion}\}$.

is so large that the look-up table would exceed available memory [8]. Geometric substitution is by no means exclusive to substotope algorithms patterned after MC; it has found many uses in computer graphics to model shapes of plants [21], [22], complicated curves [23], and multi-resolution surfaces [24], [25]. But, substotope algorithms are distinct from these other geometric substitutions because of the central role played by vertex colors.

As the variations listed in Section 1.1 suggest, there are five key parameters that determine the number of cases for colored polytopes, listed below.

1. The shape symmetry.
2. The color symmetry.
3. The choice of tile $\in \{n\text{-simplex}, n\text{-cube}\}$.
4. The dimension n of the polytope.
5. The number k of colors.

The remainder of this paper explains how to count cases using group theory. In brief, orbits of groups acting on sets are enumerated using a computational algebra package. Section 2 describes the aspects of group theory that are required for solving the case-counting problem. Section 3 describes how a tool for computational group theory can be programmed to count cases and presents case counts organized into a taxonomy: a table with $2 \times 3 \times 2 \times 4 \times 4 = 192$ entries corresponding to different values of the five parameters listed above. Section 4 presents the history of the case-counting problem, especially for the 4-cube. Section 5 shows how Pólya theory can count the number of cases without explicitly enumerating them.

2 ACTION OF A GROUP ON A SET

For years, mathematicians have studied problems similar to counting cases of polytope colorings. In order for us to apply their results, we first convert the problem of counting cases (Step 3 of the substotope algorithm) into the appropriate mathematical language. This task requires the use of group theory, which owes its name to a paper published in 1854 by Cayley [26].

A group is a set G with a binary operation satisfying four criteria:

1. G is closed under the operation.
2. The operation obeys the associative law.
3. G has an identity element (denoted by the symbol 1).
4. Each element in G has an inverse.

Often the appearance of the binary operation is suppressed, so $a * b$ is written as ab and $a * a$ is written as a^2 . More details about groups can be found in textbooks on modern algebra, such as the popular one by Fraleigh [27]. Familiar examples of groups include integers with the addition operation and rational numbers (without zero) under multiplication.

In creating the table for MC, Lorensen and Cline produced a set of 256 cube colorings. Then, they considered the action of a symmetry group on the 256 cube colorings. A group *acts* on a set X by mapping X to itself as described below.

Definition 1. A group G is said to act on a set X if 1) the identity fixes every element of X : $1x = x$ and 2) the associative law holds: $(g_2g_1)x = g_2(g_1x)$, where $1, g_1, g_2 \in G$ and $x \in X$. (Note: some authors apply actions from the right rather than the left, thus writing xg_1g_2 .)

One well-known group action is the set of all permutations of n symbols. The group S_n , commonly called the *symmetric group*, contains all $n!$ of these permutations.

Definition 2. The symmetric group S_2 contains all permutations of the symbols 1 and 2.

$$S_2 = \{(1)(2), (1\ 2)\}.$$

We can relax the requirement that the symbols literally be “1” and “2” as long as we have symbols that come from a specified order. For example, we can interpret x and y as 1 and 2. Thus, we may write

$$S_2 = \{(x)(y), (x\ y)\}.$$

In this notation, S_2 induces an action on any point $(x, y) \in \mathbb{R}^2$. The first (identity) element leaves the x and y coordinates fixed; the second element is a cycle that sends x to y and y to x , producing a reflection about a diagonal line. If $\hat{g} = (x\ y) \in S_2$, for example, then

$$\begin{aligned}\hat{g}(x) &= (x\ y)(x) = y \\ \hat{g}(y) &= (x\ y)(y) = x.\end{aligned}$$

In the top line, the parenthetical expression $(x\ y)$ is the operator; the parenthetical expression (x) is its argument. Note that both actions of S_2 preserve the shape of an axis-aligned square centered at the origin.

The usual convention when writing a permutation is to list the cycles it induces on elements of the set. By convention, the identity mapping is generally denoted $()$ rather than $(x)(y)$; trivial cycles like (x) and (y) are often suppressed when the permutation is written. Thus, the symmetric group on two letters is the set containing the two permutations $()$ and $(x\ y)$.

The shape of the square is preserved by the action of mirror reflections (flips) exchanging x with $-x$ or y with $-y$. These two reflection groups are $\{(), (x - x)\}$ and $\{(), (y - y)\}$; each is equivalent (isomorphic) to the group S_2 . Their direct product contains all four combinations of flip operations.

$$S_2 \times S_2 \cong \{(), (x - x), (y - y), (y - y)(x - x)\}.$$

These four permutations correspond to the identity, a flip of the x -axis, a flip of the y -axis, and flips (note the order, from right to left) of the x , then y axes. In this particular example, the elements commute, so $(y - y)(x - x) = (x - x)(y - y)$, but, in general, the order matters.

The following sections explain how a group acts on a shape (the simplex or the cube) and on a set of colors, using Marching Squares as illustration. With only $2^4 = 16$ colorings, eight group elements, and four equivalence classes, Marching Squares is a natural choice for connecting the intuitive, but eventually overwhelming, brute-force approach together with the abstract, but more generalizable, group-theory approach to enumerating cases.

2.1 Groups Acting on the Set of Vertices

Let FC_n denote the full set of symmetries acting on an n -cube. FC_n is a group with $n!2^n$ elements; it includes all swaps and flips of the coordinate axes. This group, sometimes referred to in the literature as the hyperoctahedral group, is the *wreath product* of a flip with the permutations of axes. The wreath product is too complicated to describe here; for its definition,

comment	\hat{g}	$g(\hat{s})$	$g \in FC_2$
identity	$()$	(v_1, v_2, v_3, v_4)	$()$
swap	$(x\ y)$	(v_1, v_3, v_2, v_4)	$(2\ 3)$
flip	$(x\ -x)$	(v_2, v_1, v_4, v_3)	$(1\ 2)(3\ 4)$
flip	$(y\ -y)$	(v_3, v_4, v_1, v_2)	$(1\ 3)(2\ 4)$
swap, flip	$(x\ -x)(x\ y)$	(v_3, v_1, v_4, v_2)	$(1\ 2\ 4\ 3)$
swap, flip	$(y\ -y)(x\ y)$	(v_2, v_4, v_1, v_3)	$(1\ 3\ 4\ 2)$
swap, flip, flip	$(y\ -y)(x\ -x)(x\ y)$	(v_4, v_2, v_3, v_1)	$(1\ 4)$
flip	$(y\ -y)(x\ -x)$	(v_4, v_3, v_2, v_1)	$(1\ 4)(2\ 3)$

Fig. 4. Permutations acting on axes and vertices. Column \hat{g} lists permutations in terms of the x and y axes. The next column shows the permuted vertices of the square $\hat{s} = (v_1, v_2, v_3, v_4)$. Column g gives corresponding permutations in terms of vertices (not axes). Permutations are written as cycles, acting by composition from right to left.

see the algebra textbook by Cohn [28]. Combinations of swaps and flips are shown in Fig. 4 and Fig. 5.

The full set of symmetries FS_n on the n -simplex can be viewed as the group of all $(n+1)!$ possible permutations of the $n+1$ vertices. That is, $FS_n = S_{n+1}$. The *direct symmetry* group is the subset of the full symmetry group that preserves orientation. It contains the rotations but not orientation-reversing flips (mirror reflections). The direct symmetry group DC_n for the n -cube contains half of the elements of FC_n . The direct symmetry group DS_n for the n -simplex is called the alternating group A_{n+1} and contains half of S_{n+1} .

To apply FC_2 or DC_2 to the square, we first label the four vertices of the *reference square* \hat{s} whose vertices,

$$v_1 = (-1, -1) \quad v_2 = (1, -1) \quad v_3 = (-1, 1) \quad v_4 = (1, 1),$$

form the lower left, lower right, upper left, and upper right corners of a square centered at the origin. Squares, for the purpose of counting cases, are the flipped and swapped images of this reference square \hat{s} .

Definition 3. A *standard square* is the image of the reference square \hat{s} under any action of *shapeGroup*. That is, s is a standard square if and only if $s = g\hat{s}$ for some $g \in \text{shapeGroup}$.

Example 1. The element $(x - x)$ of $\text{shapeGroup} = FC_2$ acts on the square, flipping it in the x direction. So, its application to the reference square \hat{s} gives $(x - x)(\hat{s}) = (v_2, v_1, v_4, v_3)$.

Example 2. Under the action of the permutation $(x - x)$ on the x -axis, vertex v_1 in the reference square \hat{s} moves to position 2 and vertex v_2 moves to position 1. Likewise, vertices v_3 and v_4 swap positions in the tuple. The group element $(x - x)$ can be relabeled accordingly in terms of its effect on the vertices of \hat{s} , namely, $(1\ 2)(3\ 4)$. This relabeling is important in Section 3, which describes how the computational algebra package "GAP" can create *shapeGroup* automatically.

All eight actions of $\text{shapeGroup} = FC_2$ are listed in Fig. 4 and illustrated in Fig. 5. In the left-most column of Fig. 4, element $g \in FC_2$ is written in terms of the coordinates x and y it acts on. The middle column shows its action on the reference square $\hat{s} = (v_1, v_2, v_3, v_4)$. The right-hand column renames the group element as g , which acts on the vertices of the square. Note that four of the group elements preserve the orientation of the square; these elements comprise the

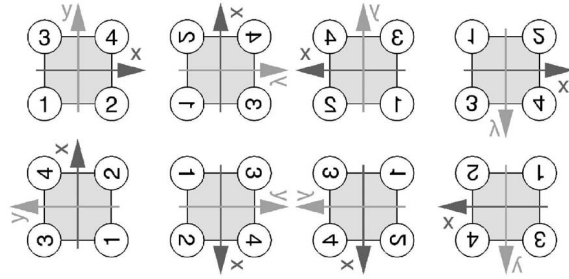


Fig. 5. Permutations from the full symmetry group FC_2 in Fig. 4 acting on the x and y axes and thereby on the standard square \hat{s} (shown in the upper left). Top row: First four permutations applied to \hat{s} . Bottom row: Next four permutations applied to \hat{s} . Note that half of the permutations preserve orientation: one, the identity $()$, on the top row and three, corresponding to rotations by 90° , 270° , and 180° on the bottom.

direct symmetry group DC_2 that acts on the square. The other four elements reverse the orientation of the square; these are the squares whose labels are restored when the page is held up to a mirror and rotated properly.

2.2 Groups Acting on the Set of Colors

In counting cases for MC, we see that one group acts on the vertices of a square by moving them around; another group acts on the set of colors, such as reversing black and white. We call the second group *colorGroup*.

A vertex v_i in a square might be assigned either of the symbols $+$ and $-$ (or, equivalently, either of two colors $color_1$ and $color_2$) to indicate the sign of $f(v_i) - const$, where $const$ is the isovalue. The color of vertex v_i is determined by a coloring function $\chi(v_i)$ which maps vertices to colors.

If $\hat{\sigma}$ is a permutation on the colors, then $color_i$ is mapped to the color $\hat{\sigma}(color_i)$. The notation is simplified if we use the permutation σ that maps one color *index* to another color *index* as shown below.

$$\hat{\sigma}(color_i) = color_{\sigma(i)}.$$

Example 3. The permutation $\sigma = (1\ 2)$ acts as follows on the color indices 1 and 2:

$$\begin{aligned} \sigma(1) &= (1\ 2)(1) = 2 \\ \sigma(2) &= (1\ 2)(2) = 1. \end{aligned}$$

There are three different color groups that play a crucial (albeit unmentioned) role in the subtopology algorithms surveyed in Section 1.1. These are described below.

2.2.1 Identity Group

The identity group Id_k , acting on k colors, contains the single element $()$, which leaves the colors fixed. Nielson and Sung considered the ordering of the colors to be significant in their Interval Volumes algorithm, meaning the identity group Id_k was acting on k colors.

2.2.2 Reversal Group

The reversal group Rev_k acting on the numbers $1..k$ swaps the first with the last element, the second with the next-to-last element, and so forth. MC, Marching Squares, Marching Hypercubes, Sweeping Simplices, and Contour Meshing all use the reversal group to reorder the colors corresponding to $+$ and $-$ in generating a level set. The group Rev_k contains

only two permutations: the identity permutation $()$ and the permutation ρ defined below.

$$\rho = \begin{cases} (1\ k)(2\ k-1)\dots(\frac{k}{2}\ \frac{k+2}{2}) & \text{if } k \text{ is even} \\ (1\ k)(2\ k-1)\dots(\frac{k-1}{2}\ \frac{k+3}{2}) & \text{if } k \text{ is odd.} \end{cases}$$

2.2.3 Symmetric Group

The symmetric group S_k arbitrarily permutes k colors. In the Separating Surfaces algorithm, Nielson and Franke considered orderings of colors to be equivalent under any permutations, thereby implicitly allowing the symmetric group S_k to act on the k colors.

So, *shapeGroup* (namely, the direct symmetry group or the full symmetry group) acts on the vertices of a square and *colorGroup* (namely, the identity group, the reversal group, or the symmetric group) acts on the colors. Together, they act on the combinatorial set of all k^{n+1} colorings of the n -simplex or k^{2^n} colorings of the n -cube. The next section describes the group acting on colorings.

2.3 Groups Acting on the Set of Colorings

The shape group and the color group work together to act on colorings of a polytope. It is convenient to write the coloring function $\chi(v_i)$ as χ_i , suppressing the v , so that a coloring of the square can be written in the compact form given below.

Definition 4. A coloring of the n -tuple of vertices is the corresponding n -tuple of colors (χ_1, \dots, χ_n) and its permutations by *shapeGroup* \times *colorGroup*.

Example 4. A coloring of the triangle is a 3-tuple of colors (χ_1, χ_2, χ_3) ; a coloring of the square is a 4-tuple $(\chi_1, \chi_2, \chi_3, \chi_4)$.

Example 5. Suppose *color*₁ is purple and *color*₂ is orange. The coloring (1,1,1,1) is a square with all purple vertices. The coloring (1, 1, 2, 2) has purple for the bottom two vertices and orange for the top two. The coloring (1, 2, 1, 2) has purple on the left side and orange on the right. If *shapeGroup* = FC_2 and *colorGroup* = Rev_2 , then the colorings (1, 1, 1, 1) and (2, 2, 2, 2) are equivalent via reversal of colors and $(1, 1, 2, 2) \cong (1, 2, 1, 2)$ via rotation of the square.

The direct product *shapeGroup* \times *colorGroup* acts on a coloring in the obvious way: An element of *shapeGroup* shuffles the *order* of the four colors and an element of *colorGroup* permutes the *value* of the colors. We call this product *coloringGroup*. An element h of *coloringGroup* has the form $(g, \hat{\sigma})$, where g acts on vertices and $\hat{\sigma}$ acts on colors.

Example 6. The action of $((2\ 3), \hat{\sigma})$ on a coloring $(\chi_1, \chi_3, \chi_2, \chi_4)$ of the square is

$$((2\ 3), \hat{\sigma})(\chi_1, \chi_2, \chi_3, \chi_4) = (\hat{\sigma}(\chi_1), \hat{\sigma}(\chi_3), \hat{\sigma}(\chi_2), \hat{\sigma}(\chi_4)),$$

where the two middle terms get switched—the result of permutation (2 3) acting on the tuple—and the colors get permuted.

Representing *color* _{i} by its subscript i allows the action on the coloring (1, 1, 2, 1) to be written as follows, using σ (to permute indices) rather than $\hat{\sigma}$:

$$((2\ 3), \sigma)(1, 1, 2, 1) = (\sigma(1), \sigma(2), \sigma(1), \sigma(1)).$$

Again, the second and third elements in the tuple get swapped by (2 3), while σ is applied to the color indices. We next show an example with a specific permutation from *shapeGroup* and a specific permutation from *colorGroup* acting on a specific coloring.

Example 7. The group element $((2\ 3), (1\ 2))$ acts on the coloring (1, 1, 2, 1) as follows:

$$\begin{aligned} & ((2\ 3), (1\ 2))(1, 1, 2, 1) \\ = & ((1\ 2)(1), (1\ 2)(2), (1\ 2)(1), (1\ 2)(1)) \\ = & (2, 1, 2, 2). \end{aligned}$$

The second and third elements of the tuple get swapped and all the colors get reversed. In this illustration, the action (1 2) of the coloring group happened to be applied second, after the action of the shape group. The two actions commute, so the order of application does not matter.

Two colorings x_1 and x_2 are said to be equivalent if a group action maps one into the other (by permuting the vertices and colors). With *coloringGroup* = $FC_2 \times Rev_2$, for example, all eight of the squares are equivalent whose vertices are three black and one white or one black and three white. Each of these squares can be mapped to any other via the action of some element of *coloringGroup*. Each equivalence class of colorings forms an orbit, which is defined as follows:

Definition 5. The orbit of the coloring x under the coloring group G is the set $\{g(x) : g \in G\}$.

Any two colorings x_i and x_j in the same orbit are equivalent under the equivalence relation defined by the action of G . Thus, the orbits form a partition of the set of all colorings. Fig. 6 shows the six orbits (represented by colorings 1111, 1112, 1122, 1221, 2221, and 2222) that partition the set of 16 colorings of the 2-cube under the action of $FC_2 \times Id_2$. Each row contains an orbit, closed under the action of rotations and flips. When color reversal is allowed, the 0-black coloring 1111 joins the equivalence class with the 4-black coloring 2222 and the 1-black coloring 1112 joins the equivalence class with the 3-black coloring 2221; in other words, color reversal reduces the number of equivalence classes, thereby reducing their count from six to four. Fig. 7 shows these four orbits for $FC_2 \times Rev_2$. Each row is closed under rotations, flips, and color reversal.

By casting the problem of counting cases in terms of groups, we can exploit powerful computational tools to enumerate orbits in situations where the large dimension or large combination of colorings makes hand-enumeration overwhelming.

3 TABLE OF SUBSTITOPE CASES

Computational group theory is concerned with the algorithmic solution of problems in group theory (e.g., the solutions to Rubik’s cube). An article by Seress gives an overview of computational group theory [29]. We investigated how a computational group theory system can solve the problem of counting cases for coloring groups [3].

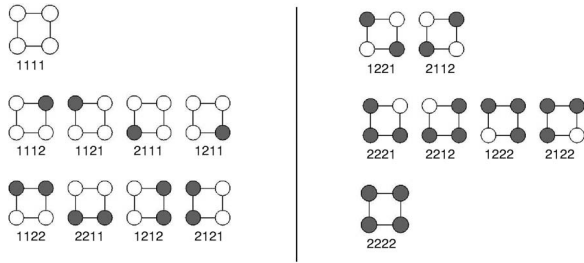


Fig. 6. The six orbits of $FC_2 \times Id_2$ acting on colorings of the 2-cube. In each row, any coloring can be mapped to any other via rotations and flips shown in Fig. 5. Colors 1 and 2 denote white and black; each coloring is given in the order lower-left, lower-right, upper-left, upper-right. Left side, top to bottom: orbits with zero black, one black, and two adjacent black vertices. Right side, top to bottom: orbits with two opposite black, three black, and four black vertices.

3.1 Computational Group Theory with GAP

GAP (Groups, Algorithms, and Programming) [30] is free software that supports computational group theory. We describe how to use GAP to solve the particular problem of counting cases for a two-colored square. (Note: GAP uses the convention of applying group actions from the right, as mentioned in the definition of “action” in Section 2.)

Below is a transcript of an interactive session using GAP for computing the number of cases (orbits) in Marching Squares. The session has been lightly edited to improve formatting. The user’s input is shown in sans-serif font and GAP’s reply is shown in the fixed-width typewriter font. We begin by creating *shapeGroup* as the wreath product $S_2 \wr S_n$, *colorGroup* as the reversal group, and *coloringGroup* as their product for dimension n with k colors (note: the double semicolons suppress feedback from GAP).

```
n := 2;;
k := 2;;
shapeGroup := WreathProductProductAction (
  SymmetricGroup(2), SymmetricGroup(n));;
colorGroup := Group (PermList (Reversed ([1..k])););
coloringGroup := DirectProduct (shapeGroup, colorGroup);;
Next, we construct projection operators to extract each
group ( $FC_2$  and  $Rev_2$ ) back from the direct product.
shapeProjection := Projection (coloringGroup, 1);;
colorProjection := Projection (coloringGroup, 2);;
Next, we generate the list of colors and colorings, allowing
GAP to answer back with its results. Note that all  $2^4 = 16$ 
colorings are produced, beginning with all white and
ending with all black.
numVerts := 2^n;;
coloredVerts := ListWithIdenticalEntries (numVerts, [1..k]); }
[ [1..2], [1..2], [1..2], [1..2] ]
colorings := Cartesian (coloredVerts);
[[ [1,1,1,1], [1,1,1,2], [1,1,2,1], [1,1,2,2],
  [1,2,1,1], [1,2,1,2], [1,2,2,1], [1,2,2,2],
  [2,1,1,1], [2,1,1,2], [2,1,2,1], [2,1,2,2],
  [2,2,1,1], [2,2,1,2], [2,2,2,1], [2,2,2,2] ]
Then, we define a function to perform the action of a group
element on a coloring. The projections of the element
```

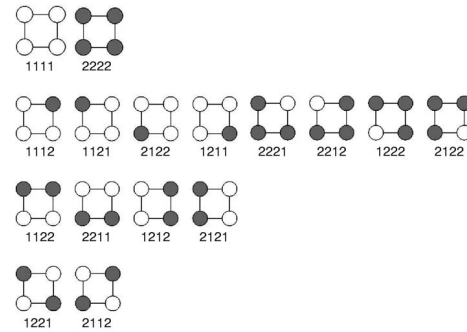


Fig. 7. The four orbits of $FC_2 \times Rev_2$ acting on colorings of the 2-cube. In each row, any coloring can be mapped to any other via rotations, flips, and color reversal. Top to bottom: orbits with all black or all white, with one black or one white, with two adjacent black and two adjacent white, and with two opposite black and two opposite white vertices.

$(g, \sigma) \in coloringGroup$ yield the components g and σ that shuffle the order of the tuple and that permute the colors.

```
action := function (coloring, groupElement)
  local shapePerm, colorPerm, shuffled;
  shapePerm := Image (shapeProjection, groupElement);
  colorPerm := Image (colorProjection, groupElement);
  shuffled := Permuted (coloring, shapePerm);
  return (OnTuples (shuffled, colorPerm));
end;;
```

We let GAP produce the orbits and count how many there are. Notice that these orbits agree exactly with the tuples we computed by hand in Fig. 7.

```
orbits := OrbitsDomain (coloringGroup, colorings, action);
[[ [1,1,1,1], [2,2,2,2] ],
 [ [1,1,1,2], [1,1,2,1], [1,2,1,1], [2,2,2,1],
  [2,1,1,1], [2,2,1,2], [2,1,2,2], [1,2,2,2] ],
 [ [1,1,2,2], [2,2,1,1], [1,2,1,2], [2,1,2,1] ],
 [ [1,2,2,1], [2,1,1,2] ] ]
Length (orbits);
4
```

This demonstration shows how GAP can enumerate the orbits of a group action in Marching Squares and, thus, determine the number of cases for polytope colorings that arise in the two-dimensional version of Marching Cubes. The variable names suggest how to extend this example to handle other cases; for example, one can simply change the value of n from 2 to 3 to enumerate the orbits and count them for MC. One can also change the definition of *shapeGroup* or *colorGroup* at the beginning of the GAP code to generate the orbits for still other colorings. The next section describes how this approach can be extended to handle additional geometries and symmetries, constructing a complete taxonomy of case-counts for subtopes.

3.2 Parameter Study

To build a complete taxonomy of case-counts, we wrote a program to let GAP loop over all parameter values. The shape groups are the direct symmetries and full symmetries of the n -simplex and n -cube, yielding four possible choices. The color groups include three possibilities: the identity, reversal, and full permutation. The shape, the shape group, and the color group are the major parameters of the

<i>n-simplex</i>				<i>direct symmetry</i>					
n^k	1	2	3	4	n^k	1	2	3	4
1	1	4	9	16	Id_k	1	4	9	16
2	1	4	11	24		2	6	24	70
3	1	5	15	36		3	23	333	2916
4	1	6	21	56		4	^{2,7} 496	230076	22456756
	1	2	5	8	Rev_k	1	2	5	8
	1	2	6	12		1	4	14	38
	1	3	9	20		1	¹ 15	⁷ 183	1508
	1	3	12	28		1	² 272	⁷ 115606	>11000000
	1	2	2	2	S_k	1	2	2	2
	1	2	3	3		1	4	6	7
	1	3	4	5		1	15	72	166
	1	3	5	6		1	272	38914	>930000
<i>full symmetry</i>									
	1	3	6	10	Id_k	1	3	6	10
	1	4	10	20		1	6	21	55
	1	5	³ 15	35		1	⁷ 22	267	1996
	1	6	21	56		1	⁷ 402	132102	11756666
	1	2	4	6	Rev_k	1	2	4	6
	1	2	6	10		1	4	⁷ 13	31
	1	⁴ 3	⁷ 9	19		1	¹ 14	⁷ 147	⁷ 1036
	1	^{5,7} 3	^{5,7} 12	28		1	² 222	⁷ 66524	>5600000
	1	2	2	2	S_k	1	2	2	2
	1	2	3	3		1	4	6	⁷ 7
	1	3	4	⁶ 5		1	14	58	⁷ 124
	1	3	5	6		1	222	22490	>460000

Fig. 8. Table of case counts for substotopes. Each of the 12 subtables contains the case-counts for a given shape group and color group in dimension n with k colors. ¹MC. ²Marching Hypercubes. ³Interval volumes. ⁴Sweeping Simplices. ⁵Contour Meshing. ⁶Separating Surfaces. ⁷ Counting Cases.

taxonomy of substotpe cases, yielding 12 subtables within the taxonomy. The dimension n and the number k of colors are the minor parameters, in the range [1..4], that vary within each subtable of Fig. 8. Each entry in the table gives the number of cases (orbits) for the corresponding colored polytopes. A table entry noted in boldface indicates a combination of parameters that was at work in any of six algorithms surveyed in Section 1.1. The GAP source code orbitTable.gap that generated the table is freely available for download at the GAP Web site. Computational details about generating the table are given in our previous paper [3]. Where calculation of the case-counts (for four colors of the 4-cube) exceeded the memory capacity of our machines, we merely estimated a lower bound for these entries based on the fact that no orbit can be bigger than the order of the group or else used results from Section 5.

3.3 Discussion

The following examples illustrate the table’s predictive utility. Entries in the table are specified by the five parameters (*shapeGroup*, *colorGroup*, *shape*, n , k).

Example 8. What happens when the algorithm for Separating Surfaces is extended to 3-cubes with four colors? Consulting the table (*full*, S_k , *n-cube*, 3, 4) we see that 124 cases arise. For the square ($n = 2$) with four colors only seven cases arise; they are illustrated in Fig. 1 (top

row), with a possible interpretation of colors that might be derived from medical data.

Example 9. What happens when Marching Squares is extended to handle the degenerate situation $f(v_i) - const = 0$ at vertices v_i ? This case would almost never occur if the scalar function were truly real-valued. But, in practice, integer-valued isosurfaces of integer-valued data sets are routinely displayed, so a level set will, with nonzero probability, pass through many grid point degeneracies. When the set of colors is augmented so it includes the degenerate zero value as a third color for the square (*full*, Rev_k , *n-cube*, 2, 3), 13 cases arise; they are illustrated in Fig. 1 (bottom row).

Example 10. What happens when Marching Hypercubes is extended to handle the same degenerate situation where $f(v_i) - const = 0$? The table predicts that, for $n = 4$, the number of cases explodes from 222 with two colors to 66,524 with three colors under full symmetry and from 272 with two colors to 115,606 with three colors under direct symmetry.

Example 11. Weigle and Banks mentioned the degenerate situation where $f(v_i) - const = 0$ for the n -simplex in Contour Meshing, but did not enumerate all the cases. How many cases would be found for the 4-simplex with three colors? The table predicts that, for (*full*, Rev_k , *n-simplex*, 4, 3), there are 12 cases.

Example 12. What happens when Interval Volumes is applied to cubes instead of tetrahedra? The table predicts that, for (*full*, Rev_k , *n-cube*, 3, 3), there are 147 cases.

Example 13. What happens when another interval is added to Interval Volumes, creating the four “colors” $(-\infty, a)$, (a, b) , (b, c) , and (c, ∞) ? With $k = 4$, the table predicts that, for the tetrahedron (*full*, Rev_k , *n-simplex*, 3, 4), there are 19 cases and, for the cube (*full*, Rev_k , *n-cube*, 3, 4), there are 1,036 cases in a hypothetical 4-color version of Interval Volumes.

Example 14. When *shapeGroup* = DC_4 for the 4-cube, 496 cases arise when the colors are not permuted, whereas there are 272 cases under color reversal. Under full symmetry of the 4-cube, 402 cases arise when the colors are not permuted, whereas there are 222 cases under color reversal. Thus, 50 of the 272 cases for *direct* have orientation-reversed partners among the 222 cases for *full* in four-dimensional MC.

These examples illustrate how the table in Fig. 8 can be used to determine the number of cases required to implement substotpe algorithms and how GAP can enumerate the cases to permit deeper understanding of how the cases occur.

4 HISTORY OF COUNTING CASES

The problem of enumerating the colorings of an n -cube with two colors has a long history, predating Marching Cubes by more than a century. Mathematicians in the 1870s were considering the kinds of logical propositions that can exist. In Boolean functions of n logical variables, each variable can assume a value of either True or False, leading to 2^n combinations in the domain of the function. If we

associate a spatial dimension in \mathbb{R}^n with each variable (True = 1 and False = 0), then each element of the domain can be represented as an n -vector with coordinates of 0s and 1s. Attempts to enumerate these equivalence classes of functions were carried out in the 1800s, apparently without their realizing that 2-colorings of the n -cube were also being counted. In 1871, Jevons presented a paper that described the cases that arise in 2-colorings of the 2-cube and the 3-cube [31], but in the guise of Boolean functions. Jevons only considered 192 of the 256 propositions corresponding to 2-colorings of the 3-cube; he found 16 of the 22 equivalence classes. Regarding the 4-cube, Jevons remarked that “some years of continuous labour would be required to ascertain the types of laws [equivalence classes] which may govern the [65,536] combinations of only four things” (p. 143).

In 1877, Clifford claimed a solution to Jevons’s puzzle of counting cases for 2-colorings of the 4-cube. In geometric terms, Clifford’s approach was to consider each coloring as a vector with four components (one for each Boolean variable) with values of 0 or 1. He reported that the $2^{16} = 65,536$ distinct functions fell into 396 equivalence classes. This impressive (but incorrect) result was publicized in that same year in a work by Jevons on formal logic [32]. Our calculation using GAP found 402 equivalence classes for 2-colorings of the 4-cube with $FC_4 \times Id_2$, not 396 [3]. (In an unpublished manuscript fragment “Enumeration of the Types of Compound Statements” [33], Clifford begins an attempt to generalize to the n -cube for arbitrary n and also attempts to determine the size of each equivalence class. For $n = 4$ and $c = 1, 2, 3$, he correctly finds the distribution of orbits to be 1, 4, and 6.)

Clifford’s results went unchallenged until 1940, when Pólya demonstrated how his famous enumeration methods could be applied to this problem in an automated manner. He enumerated the 402 classes of 2-colorings of the 4-cube, and showed where Clifford’s manual tabulations had overcounted in some cases and undercounted in others: “Clifford trouve, par une méthode laborieuse, les valeurs $N_4^{(6)} = 47$, $N_4^{(7)} = 55$, $N_4^{(8)} = 78$, tandis que je trouve les nombres 50, 56, 74. Des vérifications variées de mes calculs et l’uniformité de ma méthode me font croire que mes nombres sont justes” [34].¹ So, the question of how to determine the number of cases for $FC_4 \times Id_2$ dates back to 1871 and its solution dates back to 1940.

5 PÓLYA COUNTING

The subtables in Fig. 8 for which $colorGroup = Id_k$ can be determined through Burnside-Pólya counting theory. The group-theorist Burnside derived a formula for counting the orbits of a permutation group [35, Section 145, Theorem VII] and Pólya developed a theory that exploited Burnside’s result to solve a vast number of counting problems in his classic paper [36]. This work was translated into English as *Combinatorial Enumeration of Groups, Graphs, and Chemical Compounds*, with commentary and a wealth of references to further applications of Pólya’s method, by Read [37].

1. “Clifford found, through his laborious approach, the values 47, 55, and 78 for cases 6, 7, and 8, whereas I found the numbers 50, 56, and 74. Based on several examples that validate my calculations and based on the generality of my method, I believe my numbers to be correct.”

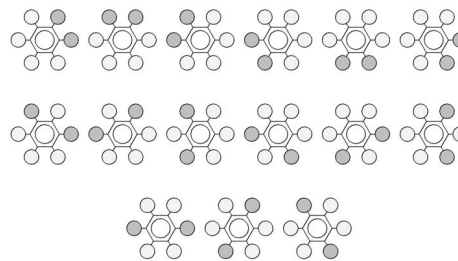


Fig. 9. The three stereoisomers of dichlorobenzene, whose structure is a hexagonal benzene ring with six atoms attached. Four of these atoms are hydrogen (light color) and two of them are chlorine (dark color). Each row contains colorings that are equivalent under the action of the shape group for a hexagon. Top row: ortho-dichlorobenzene, used as a paint remover. Middle: meta-dichlorobenzene, used as a toilet deodorizer. Bottom: para-dichlorobenzene, used as an insecticide.

Why are chemical compounds included in Pólya’s exposition of counting cases? The reason is simple: A molecule is a graph whose vertices are atoms. The different kinds of atoms (e.g., carbon, hydrogen, chlorine) are the “colors.” This is illustrated by the following example: The organic molecule dichlorobenzene has the chemical formula $C_6H_4Cl_2$; the six carbon atoms are connected in a hexagon, and each carbon is attached to an atom having color “hydrogen” or color “chlorine.” The two chlorines can be separated by 120° , 240° , or 360° ; each configuration (or stereoisomer) is an equivalence class of colorings. These equivalence classes are illustrated in Fig. 9. Pólya realized that Burnside’s formula could be applied to count the number of stereoisomers of chemical compounds.

In this section, we present a brief introduction to Pólya’s method and show how it can be used to derive some of the numbers in Fig. 8 without brute-force enumeration. A central concept in this method is the *cycle index* of a permutation group G , where G is the *shapeGroup* in our discussion. This multivariable polynomial records the cycle structure of each permutation in G . Its definition is given below.

Definition 6. The cycle index Z of a group G is the polynomial

$$Z(G; z_1, z_2, \dots, z_d) = \left(\sum_{g \in G} z_1^{j_1(g)} z_2^{j_2(g)} \dots z_d^{j_d(g)} \right) / |G|,$$

where $j_i(g)$ is the number of cycles of length i in the permutation $g \in G$, $|G|$ is the order of (i.e., size of) G , and d is the degree of G (the size of the set permuted by G). Note that $j_i(g)$ might happen to be zero; for example, no rigid rotation of the square contains a 3-cycle. It is a known result in group theory that $\sum_{i=1}^d i \cdot j_i(g) = d$ for each $g \in G$, which can serve as a check when computing cycle indices by hand.

5.1 Pólya Applied to a 2-Cube with Full Symmetry

We illustrate the cycle index using the full symmetry group FC_2 for the square (presented in Fig. 5), also called the *dihedral group* D_4 . This group has order $|D_4| = 8$, degree $d = 4$, and contains the following elements.

$$\begin{array}{cccc} (1)(2)(3)(4) & (1\ 2)(3\ 4) & (1\ 3)(2\ 4) & (1\ 4)(2\ 3) \\ (1)(2\ 3)(4) & (1\ 2\ 4\ 3) & (1\ 3\ 4\ 2) & (1\ 4)(2)(3). \end{array}$$

$g \in G$	j_1	j_2	j_3	j_4	$z_1^{j_1}$	$z_2^{j_2}$	$z_3^{j_3}$	$z_4^{j_4}$	prod
(1)(2)(3)(4)	4	0	0	0	z_1^4	z_2^0	z_3^0	z_4^0	z_1^4
(1 2)(3 4)	0	2	0	0	z_1^0	z_2^2	z_3^0	z_4^0	z_2^2
(1 3)(2 4)	0	2	0	0	z_1^0	z_2^2	z_3^0	z_4^0	z_2^2
(1 4)(2 3)	0	2	0	0	z_1^0	z_2^2	z_3^0	z_4^0	z_2^2
(1)(2 3)(4)	2	1	0	0	z_1^2	z_2^1	z_3^0	z_4^0	$z_1^2 z_2^1$
(1 2 4 3)	0	0	0	1	z_1^0	z_2^0	z_3^0	z_4^1	z_4^1
(1 3 4 2)	0	0	0	1	z_1^0	z_2^0	z_3^0	z_4^1	z_4^1
(1 4)(2)(3)	2	1	0	0	z_1^2	z_2^1	z_3^0	z_4^0	$z_1^2 z_2^1$

Fig. 10. Constituents of the cycle index for the shape group $G = D_4$ of full symmetries of the square. The left column contains the group elements. The next column contains the cycle counts. In the top row, the identity element has four cycles of length 1, so $j_1 = 4$. In the second row, the group element has two cycles of length 1, so $j_1 = 2$. The next column contains the z -terms of the cycle index. The value of j_i becomes the exponent of z_i . The right column contains the product of the z terms. The cycle index is the sum of these products.

Fig. 10 provides explicit details for computing the cycle index for the dihedral group, which corresponds to the discussion below. The second column shows the number of cycles, within element g , of length 1, 2, 3, and 4. The third column shows each z term, whose subscripts are the cycle lengths and whose superscripts are the number of cycles of that length. The right column gives the product of these terms. When the superscript of a term is zero, that term z_i^0 evaluates to unity and does not contribute to the product.

One z_1^4 term. The identity permutation in D_4 , denoted by () or (1)(2)(3)(4), contains four disjoint cycles, each of length 1. Thus, according to the formula, this permutation contributes a term of z_1^4 to the cycle index.

Three z_2^2 terms. There are three permutations that contain a pair of cycles. These permutations (1 2)(3 4), (1 3)(2 4), and (1 4)(2 3) each consist of two cycles of length 2, so each contributes a term of z_2^2 to the cycle index.

Two $z_1^2 z_2^1$ terms. The permutation (1)(2 3)(4) consists of two cycles of length 1, each of which contributes a factor of z_1^2 to this permutation's term, and one cycle of length 2, which contributes a factor of z_2^1 . Thus, this permutation is represented by a term $z_1^2 z_2^1$ in the cycle index. The permutation (1 4)(2)(3) has the same cycle structure and contributes the same term $z_1^2 z_2^1$ to the cycle index.

Two z_4^1 terms. The two permutations (1 2 4 3) and (1 3 4 2) each consist of a single cycle of length 4. Each contributes a term of z_4^1 to the cycle index.

Adding up these terms and dividing by the order of the group, we have

$$Z(FC_2; z_1, z_2, z_3, z_4) = (1z_1^4 + 2z_1^2 z_2^1 + 3z_2^2 + 2z_4^1)/8$$

as the cycle index of the shape group $FC_2 \cong D_4$. Remarkably, the cycle index polynomial serves as a generating system to produce all the values in the (full symmetry, Id_k , n -cube) subtable of Fig. 8, with $n = 2$. The details are described below.

An elementary version of Pólya's theorem states that, under the identity action of the color group, the number of

inequivalent colorings of the square, with k colors, is obtained by replacing each variable z_i in the cycle index of the *shapeGroup* with the integer k . Thus, for $k = 2$ colors, the number of inequivalent colorings of the square is

$$\begin{aligned} Z(FC_2; 2, 2, 2, 2) &= (2^4 + 2 \cdot 2^2 \cdot 2^1 + 3 \cdot 2^2 + 2 \cdot 2^1)/8 \\ &= (16 + 16 + 12 + 4)/8 \\ &= 6, \end{aligned}$$

which confirms the 2, 2 entry in the (full symmetry, Id_k , n -cube) subtable. These are the six equivalence classes of colorings where black and white are not exchanged; for example, the all-white coloring is considered distinct from the all-black coloring.

More generally, we see that the number of inequivalent k -colorings of the square is

$$Z(FC_2; k, k, k, k) = (k^4 + 2k^3 + 3k^2 + 2k)/8.$$

This formula can be used to confirm and extend the entire second row of the (full, Id_k , n -cube) subtable in Fig. 8. So, for example, with $k = 3$ colors, we have

$$\begin{aligned} Z(FC_2; 3, 3, 3, 3) &= (3^4 + 2 \cdot 3^3 + 3 \cdot 3^2 + 2 \cdot 3)/8 \\ &= (81 + 54 + 27 + 6)/8 \\ &= 21. \end{aligned}$$

For $k = 4$ colors, we have

$$\begin{aligned} Z(FC_2; 4, 4, 4, 4) &= (4^4 + 2 \cdot 4^3 + 3 \cdot 4^2 + 2 \cdot 4)/8 \\ &= (256 + 128 + 48 + 8)/8 \\ &= 55. \end{aligned}$$

We call this polynomial in k the *reduced* cycle index of the group action.

5.2 Pólya Applied to a 3-Cube with Direct Symmetry

For a more complicated example, we consider the direct symmetry group DC_3 acting on the vertices of a 3-cube. A 3-cube has eight vertices, so this shape group has degree $d = 8$. We number the vertices of the cube from 1 to 8, as shown in Fig. 11 (left). We can convince ourselves that the group has order 24 as follows: From a standard position, the cube can be manipulated so that any of the six faces is on the bottom. Then, the cube can be rotated about a vertical axis to bring any of the four sides to the front. This yields a total of $6 \times 4 = 24$ permutations, thus DC_3 has order 24. The cycle structures of each permutation in DC_3 are explained below, collected according to rotation axes.

One identity. The identity permutation (1)(2)(3)(4)(5)(6)(7)(8) consists of eight cycles each of length 1. Thus, its contribution to the cycle index is the term z_1^8 .

Nine rotations through opposite faces. We next consider rotations about the vertical axis passing up through the center of the 1, 2, 3, 4 face on the bottom and the 5, 6, 8, 7 face on the top. A 90° counterclockwise rotation of the cube about this axis produces the permutation (1 2 4 3)(5 6 8 7). A 90° clockwise rotation produces the inverse permutation (1 3 4 2)(5 7 8 6). Each of these permutations consists of two cycles of length 4, so each contributes a term of z_4^2 to the cycle index. Moreover, the 180° rotation about this axis produces the

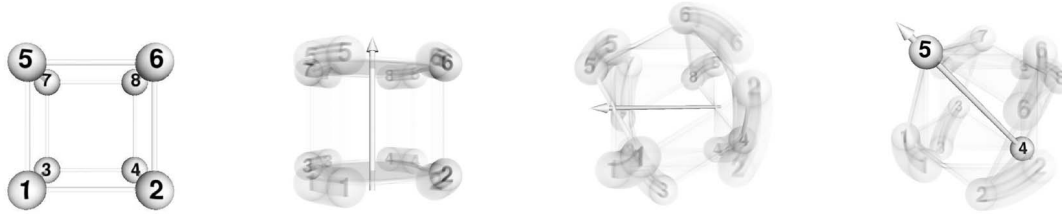


Fig. 11. The motion-blurred images of continuous rotations of the 3-cube suggest the discrete rotations in the full symmetry group FC_3 . From left to right: identity, rotations through opposite faces, rotations through opposite edges, rotations through opposite vertices.

permutation $(1\ 4)(2\ 3)(5\ 8)(6\ 7)$, consisting of four cycles of length 2 and producing a contribution of z_2^4 to the cycle index.

There are two other axes of rotation that pass through matching pairs of faces: one passing side-to-side through the centers of the 1, 3, 7, 5 face on the left and the 2, 4, 8, 6 face on the right; the other passing front-to-back through the centers of the 1, 2, 6, 5 face in front and the 3, 4, 8, 7 face in back. There are three nontrivial rotations about each of these axes, with the same cycle structure as the three listed above for the top-bottom faces. The total contribution to the cycle index of the permutations resulting from rotations about the left-right axis and top-bottom axis is thus $6z_4^2 + 3z_2^4$.

Six rotations through opposite edges. Next, we consider an axis through the middle of the 1, 5 edge and its opposite 4, 8 edge. A rotation of 180° about this axis produces the permutation $(1\ 5)(2\ 7)(3\ 6)(4\ 8)$, for a contribution of z_2^4 . There are a total of six such axes through opposite edges, so the corresponding permutations contribute a total of $6z_2^4$ to the cycle index.

Eight rotations through opposite vertices. Finally, we consider an axis passing diagonally through the cube, through vertices 4 and 5. A 120° rotation about this axis fixes points 4 and 5 and spins the trio of points 1, 6, and 7 (lying in a plane perpendicular to the axis) and the trio of points 2, 8, and 3. Thus, it produces the permutation $(1\ 6\ 7)(2\ 8\ 3)(4)(5)$. A rotation in the opposite direction produces the inverse permutation $(1\ 7\ 6)(2\ 3\ 8)(4)(5)$. Each of these permutations contributes $z_1^2 z_3^2$ to the cycle index. Moreover, there are three other axes of this form: one through opposite vertices 1 and 8, one through vertices 2 and 7, and one through vertices 3 and 6. Collectively, the eight permutations of this form contribute $8z_1^2 z_3^2$ to the cycle index.

That makes $1 + 9 + 6 + 8 = 24$ permutations, so the list is complete. Adding up all the contributions and dividing by the order of the group, we have

$$Z(DC_3; z_1, z_2, z_3, z_4) = (z_1^8 + 6z_4^2 + 9z_2^4 + 8z_1^2 z_3^2) / 24$$

as the cycle index of the direct 3-cube group DC_3 . Note that the longest cycle of any permutation in this group is only 4, although the degree of the group is 8 (i.e., there are eight vertices being acted on). Rather than include all eight of them as parameters of the function Z , we have suppressed the four “dummy variables” z_i for $i > 4$ to save space below.

As with the 2-cube, we obtain the number of ways to k -color the 3-cube by replacing each variable z_i by the

integer k . Thus, for $k = 2$ colors, the number of different colorings of the cube, up to rotation, is

$$Z(DC_3; 2, 2, 2, 2) = (2^8 + 6 \cdot 2^2 + 9 \cdot 2^4 + 8 \cdot 2^2 \cdot 2^2) / 24 = 23,$$

which confirms the 3, 2 entry in the upper right subtable of Fig. 8.

Generalizing once again, we can combine like terms and see that

$$Z(DC_3; k, k, k, k) = (k^8 + 17k^4 + 6k^2) / 24$$

gives the number of different k -colorings of the cube, up to rotation. This formula can be used to confirm and extend the entire third row of the upper right subtable of Fig. 8. So, for example, with $k = 3$ colors, we have

$$Z(DC_3; 3, 3, 3, 3) = (3^8 + 17 \cdot 3^4 + 6 \cdot 3^2) / 24 = 333.$$

When $k = 4$, we have

$$Z(DC_3; 4, 4, 4, 4) = (4^8 + 17 \cdot 4^4 + 6 \cdot 4^2) / 24 = 2,916.$$

Clearly the most difficult part of using Pólya’s theory to count cube colorings is the derivation of the appropriate cycle indices. Beyond three dimensions, the process quickly becomes tedious, nonvisual, and error-prone. Fortunately, the four-dimensional formulas are known and methods for systematically deriving the higher dimensional formulas exist. The formula for the cycle index of the full group FC_4 of symmetries of the 4-cube was derived by Pólya [34]. He used this result to count the 2-colorings of the 4-cube in the context of counting equivalence classes of Boolean functions of four variables. The formula is

$$Z(FC_4; z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8) = 1/384 (z_1^{16} + 51z_2^8 + 12z_1^8 z_2^4 + 84z_4^4 + 32z_1^4 z_3^4 + 96z_2^2 z_6^2 + 12z_1^4 z_2^6 + 48z_1^2 z_2^1 z_4^3 + 48z_8^2).$$

Pólya counted the different 2-colorings of the 4-cube, up to rotation and reflection, by replacing each z_i by 2. More generally, we can obtain the number of k -colorings of the 4-cube by replacing each z_i with the integer k , yielding

$$Z(FC_4; k, k, k, k, k, k, k, k) = 1/384 (k^{16} + 12k^{12} + 12k^{10} + 83k^8 + 48k^6 + 180k^4 + 48k^2)$$

as the reduced cycle index.

We can confirm Pólya’s (and GAP’s) count of 402 by replacing k with 2 in this formula. Replacing k with 4 fills one of the missing entries in the table that brute-force enumeration with GAP failed to compute due to insufficient memory [3]. It is easy to see why four colors caused trouble: There are $4^{2^4} = 4, 294, 967, 296$ 4-colorings of the 4-cube to consider via brute-force traversal. The reduced cycle index makes the calculation easy: There are 11,756,666 cases. We used GAP to derive

$$Z(DC_4; z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8) = (k^{16} + 12k^{11} + 63k^8 + 63k^4 + 48k^2)/192$$

as the reduced cycle index for DC_4 . Letting $k = 4$, we fill another entry (22,456,756) that was missing in the previously published table [3].

5.3 Pólya Applied to a 2-Simplex with Full Symmetry

Now, we turn our attention to symmetries of the n -simplex. The full symmetry group acting on a triangle (2-simplex with three vertices) is the symmetric group S_3 ; its elements are all the permutations of three symbols. In the case of the triangle, the symbols are the three vertices. The $3! = 6$ elements of the group are the identity (1)(2)(3), the two rotations (1 3 2) and (1 2 3), and the three flips (1)(2 3), (1 3)(2), (1 2)(3). Thus, there is a term z_1^3 arising from three cycles of length 1, there are two terms z_3 arising from one cycle of length 3, and three terms $z_1 z_2$ arising from one cycle of length 1 and one cycle of length 2.

Adding up all the contributions and dividing by the order of the group, we have

$$Z(S_3; z_1, z_2, z_3) = (z_1^3 + 2z_3 + 3z_1 z_2)/6$$

as the cycle index of the full 2-simplex group S_3 .

Letting $z_i = k$ and combining like terms, we see that the number of different k -colorings of the triangle is

$$Z(DC_3; k, k, k) = (k^3 + 3k^2 + 2k)/6.$$

This formula can be used to confirm and extend the entire second row of the left-hand, full symmetry, Id_k subtable of Fig. 8. With $k = 2, 3, 4$ colors, we have

$$\begin{aligned} Z(S_3; 2, 2, 2) &= (2^3 + 3 \cdot 2^2 + 2 \cdot 2)/6 = 4 \\ Z(S_3; 3, 3, 3) &= (3^3 + 3 \cdot 3^2 + 2 \cdot 3)/6 = 10 \\ Z(S_3; 4, 4, 4) &= (4^3 + 3 \cdot 4^2 + 2 \cdot 4)/6 = 20 \end{aligned}$$

which match GAP’s brute-force calculations.

5.4 Pólya Applied to a 3-Simplex with Direct Symmetry

In dimension $n = 3$, the tetrahedron (3-simplex with four vertices) can be acted on by the full symmetry group or by the direct symmetry group that preserves orientation. The latter group is the alternating group A_4 . Like S_4 , A_4 has elements that permute four symbols, but each element of A_4 can be expressed by an even number of even cycles, as shown below.

$$\begin{array}{cccc} (1)(2)(3)(4) & (1\ 2)(3\ 4) & (1\ 3)(2\ 4) & (1\ 4)(2\ 3) \\ (1)(2\ 3\ 4) & (3)(1\ 2\ 4) & (4)(1\ 3\ 2) & (2)(1\ 4\ 3) \\ (1)(243) & (4)(1\ 2\ 3) & (2)(1\ 3\ 4) & (3)(1\ 4\ 2) \end{array}$$

Thus, the cycle index includes a term z_1^4 arising from four cycles of length 1. It contains three terms z_2^2 arising from two cycles of length 2 and eight terms $z_1 z_3$ arising from one cycle of length 1 and one cycle of length 3. Adding up all the contributions and dividing by the order of the group, we have

$$Z(A_4; z_1, z_2, z_3, z_4) = (z_1^4 + 3z_2^2 + 8z_1 z_3)/12$$

as the cycle index of the direct 3-simplex group A_4 . Letting $z_i = k$ and combining like terms, we see that the number of different k -colorings of the tetrahedron, up to rotation, is

$$Z(A_4; k, k, k, k) = (k^4 + 11k^2)/12.$$

This formula can be used to confirm and extend the entire third row of the left-hand, direct symmetry, Id_k subtable of Fig. 8. With $k = 2, 3, 4$ colors, we have

$$\begin{aligned} Z(A_4; 2, 2, 2, 2) &= (2^4 + 11 \cdot 2^2)/12 = 5 \\ Z(A_4; 3, 3, 3, 3) &= (3^4 + 11 \cdot 3^2)/12 = 15 \\ Z(A_4; 4, 4, 4, 4) &= (4^4 + 11 \cdot 4^2)/12 = 36 \end{aligned}$$

which match GAP’s brute-force calculations.

For general n , the cycle index $Z(A_n; z_1, \dots, z_n)$ for the alternating group can be computed from the cycle index of the symmetric group by the formula

$$\begin{aligned} Z(A_n; z_1, \dots, z_n) &= Z(S_n; z_1, z_2, z_3, z_4, \dots) \\ &\quad + Z(S_n; z_1, -z_2, z_3, -z_4, \dots). \end{aligned}$$

For example, we can compute the cycle index of the group A_4 as follows:

$$\begin{aligned} Z(A_4; z_1, z_2, z_3, z_4) &= Z(S_4; z_1, z_2, z_3, z_4) \\ &\quad + Z(S_4; z_2, -z_2, z_3, -z_4) \\ &= \frac{1}{24} (z_1^4 + 6z_1^2 z_2 + 3z_2^2 + 8z_1 z_3 + 6z_4) \\ &\quad + \frac{1}{24} (z_1^4 - 6z_1^2 z_2 + 3z_2^2 + 8z_1 z_3 - 6z_4) \\ &= \frac{1}{12} (z_1^4 + 3z_2^2 + 8z_1 z_3). \end{aligned}$$

We have illustrated the counting of colored n -simplexes using cycle indices in order to emphasize the universal applicability of Pólya’s theorem in the counting of equivalence classes of colored objects of various sorts. However, there is a shortcut in the case of the n -simplex. It can be shown that

$$Z(S_n; k, k, \dots, k) = \binom{n+k-1}{n} = \frac{(n+k-1)!}{n!(k-1)!}.$$

For example, we have

$$\begin{aligned} Z(S_4; k, k, k, k) &= \binom{k+3}{4} \\ &= \frac{k(k+1)(k+2)(k+3)}{4!} \\ &= \frac{1}{24} (k^4 + 6k^3 + 11k^2 + 6k). \end{aligned}$$

In addition, it can be shown that

$$\begin{aligned} Z(A_n; k, k, \dots, k) &= \binom{n+k-1}{n} + \binom{k}{n} \\ &= \frac{(n+k-1)!}{n!(k-1)!} + \frac{k!}{n!(k-n)!}. \end{aligned}$$

For example, we have

$$\begin{aligned} Z(A_4; k, k, \dots, k) &= \binom{k+3}{4} + \binom{k}{4} \\ &= \frac{k(k+1)(k+2)(k+3)}{4!} \\ &\quad + \frac{k(k-1)(k-2)(k-3)}{4!} \\ &= \frac{1}{24}(k^4 + 6k^3 + 11k^2 + 6k) \\ &\quad + \frac{1}{24}(k^4 - 6k^3 + 11k^2 - 6k) \\ &= \frac{1}{12}(k^4 + 11k^2). \end{aligned}$$

Unfortunately, no such simple formulas exist in the case of n -cubes.

We used GAP to automate the task of determining the cycle structure of the various shape groups. From the cycle structure, we determined the cycle index and the reduced cycle index (a polynomial in the k colors). The results are summarized in Figs. 12 and 13. By choosing values of k from 1 to 4, one can immediately derive the four subtables for the n -simplex and n -cube with colorGroup Id_k . We are exploring the use of GAP to compute the cycle indices of arbitrary groups.

5.5 Asymptotics

Compared to brute-force enumeration, Pólya's counting technique considerably reduces the complexity of determining sizes of orbits as n and k increase. Rather than inspect each of the k^{2^n} colorings of an n -cube, one instead counts the number of cycles in each of the $2^n n!$ elements of the full symmetry group FC_n (each of the $2^{n-1} n!$ elements of the direct symmetry group DC_n). An element permutes 2^n vertices, so counting cycles in a permutation requires inspection of 2^n symbols, a total of $2^n n! 2^n$ operations. Stirling's approximation

$$n! \approx n^n e^{-n} \sqrt{2\pi n}$$

for $n!$ can be used to compare the expense of brute-force counting versus Pólya counting. Taking the log of $2^n n! 2^n$ for Pólya, we see that $\log(2^n n! 2^n)$ is dominated by the term $n \log(n)$. In a brute-force approach, all k^{2^n} colorings are produced, then each is compared to previously generated colorings (or equivalence classes thereof) to enumerate the orbits. Let $c(n)$ represent the average cost of the comparing colorings for an n -cube. We find

$$\log(k^{2^n} c(n)) = 2^n \log(k) + \log(c(n)),$$

which asymptotically dominates $n \log(n)$. Thus, Pólya's counting technique provides a distinct advantage in determining the number of colorings of a cube when n grows large. Of course, when we make use of known

formulas for the cycle indices of the permutation groups, it is not necessary to examine the cycle structure of every group element. This makes the advantage enjoyed by the Pólya method even greater.

For the n -simplex, brute-force counting requires examination of each of the k^{n+1} colorings, whereas Pólya counting requires examination of $n+1$ symbols in at most $n!$ group elements for a total of at most $(n+1)n!$ operations. So, Pólya is asymptotically faster than brute-force for counting cases of n -simplex colorings as well as for n -cube colorings.

Although Pólya-counting is asymptotically faster than brute-force enumeration of cases, the examples in the previous sections show that finding the cycle index via counting cycles of group elements is still laborious and, thus, is error-prone. The process can be expedited using GAP.

The function `cyclesInPerm` defined below generates a list of cycles in a permutation `perm` acting on `degree` vertices.

```
cyclesInPerm := function (perm, degree)
  local cLen, x, cNumbers;
  cLen := Collected(CycleLengths(perm, [1..degree]));
  cNumbers := ListWithIdenticalEntries(degree, 0);
  for x in cLen
    do cNumbers[x[1]] := x[2]; od;
  return cNumbers;
end;
```

The function exploits GAP's capability of computing the cycle lengths in a permutation. For example, the permutation $(2\ 3) = (1)(2\ 3)(4)$, acting on four vertices, has two cycles of length 1, one cycle of length 2, zero cycles of length 3, and zero cycles of length 4. This is shown in the fifth row of Fig. 10. When GAP applies the function `cyclesInPerm((2,3), 4)`, it produces the correct list $[2, 1, 0, 0]$ of cycle counts.

Using the function `cyclesInPerm` to count cycles in a single permutation, we create a function `cyclesInGroup` to count cycles in all the elements of a group. This is shown below.

```
cyclesInGroup := function (group, degree)
  local i, gList, gNumbers;
  gList := List(group);
  gNumbers := ListWithIdenticalEntries(Size(gList), [ ]);
  for i in [1..Size(gList)]
    do gNumbers[i] := cyclesInPerm(gList[i], degree); od;
  return gNumbers;
end;
```

With FC_4 denoting the full symmetry group of the 2-cube with four vertices, GAP evaluates `cyclesInGroup(FC4, 4)` to produce the list

```
[ [4, 0, 0, 0], [0, 2, 0, 0], [0, 2, 0, 0], [0, 2, 0, 0],
  [2, 1, 0, 0], [0, 0, 0, 1], [0, 0, 0, 1], [2, 1, 0, 0] ]
```

giving the cycle structure of each of the eight elements of FC_4 . This list exactly matches the list shown in the second column of Fig. 8.

GAP's `Collected` command merges duplicates together while preserving a count of the number of identical items. This count is precisely the coefficient of the monomial term in the cycle index.

n	direct symmetry of n -simplex $Z(A_{n+1}; z_1 \dots z_d)$
1	$1z_1^2$
2	$\frac{1}{3} (1z_1^3 + 2z_3)$
3	$\frac{1}{12} (1z_1^4 + 8z_1z_3 + 3z_2^2)$
4	$\frac{1}{60} (1z_1^5 + 20z_1^2z_3 + 15z_1z_2^2 + 24z_5)$
n	direct symmetry of n -cube $Z(DC_n; z_1 \dots z_d)$
1	$1z_1^2$
2	$\frac{1}{4} (1z_1^4 + 1z_2^2 + 2z_4)$
3	$\frac{1}{24} (1z_1^8 + 8z_1^2z_3^2 + 9z_2^4 + 6z_4^2)$
4	$\frac{1}{192} (1z_1^{16} + 12z_1^4z_2^6 + 32z_1^4z_3^4 + 31z_2^8 + 32z_2^2z_6^2 + 36z_4^4 + 48z_8^2)$
n	full symmetry of n -simplex $Z(S_{n+1}; z_1 \dots z_d)$
1	$\frac{1}{2} (1z_1^2 + 1z_2)$
2	$\frac{1}{6} (1z_1^3 + 3z_1z_2 + 2z_3)$
3	$\frac{1}{24} (1z_1^4 + 6z_1^2z_2 + 8z_1z_3 + 3z_2^2 + 6z_4)$
4	$\frac{1}{120} (1z_1^5 + 10z_1^3z_2 + 20z_1^2z_3 + 15z_1z_2^2 + 30z_1z_4 + 20z_2z_3 + 24z_5)$
n	full symmetry of n -cube $Z(FC_n; z_1 \dots z_d)$
1	$\frac{1}{2} (z_1^2 + z_2)$
2	$\frac{1}{8} (1z_1^4 + 2z_1^2z_2 + 3z_2^2 + 2z_4)$
3	$\frac{1}{48} (1z_1^8 + 6z_1^4z_2^2 + 8z_1^2z_3^2 + 13z_2^4 + 8z_2z_6 + 12z_4^2)$
4	$\frac{1}{384} (1z_1^{16} + 12z_1^8z_2^4 + 12z_1^4z_2^6 + 32z_1^4z_3^4 + 48z_1^2z_2z_4^3 + 51z_2^8 + 96z_2^2z_6^2 + 84z_4^4 + 48z_8^2)$

n	direct symmetry of n -simplex $Z(A_{n+1}; k..k)$
1	k^2
2	$\frac{1}{3} (k^3 + 2k)$
3	$\frac{1}{12} (k^4 + 11k^2)$
4	$\frac{1}{60} (k^5 + 35k^3 + 24k)$
n	direct symmetry of n -cube $Z(DC_n; k..k)$
1	k^2
2	$\frac{1}{4} (k^4 + k^2 + 2k)$
3	$\frac{1}{24} (k^8 + 17k^4 + 6k^2)$
4	$\frac{1}{192} (k^{16} + 12k^{10} + 63k^8 + 68k^4 + 48k^2)$
n	full symmetry of n -simplex $Z(S_{n+1}; k..k)$
1	$\frac{1}{2} (k^2 + k)$
2	$\frac{1}{6} (k^3 + 3k^2 + 2k)$
3	$\frac{1}{24} (k^4 + 6k^3 + 11k^2 + 6k)$
4	$\frac{1}{120} (k^5 + 10k^4 + 35k^3 + 50k^2 + 24k)$
n	full symmetry of n -cube $Z(FC_n; k..k)$
1	$\frac{1}{2} (k^2 + k)$
2	$\frac{1}{8} (k^4 + 2k^3 + 3k^2 + 2k)$
3	$\frac{1}{48} (k^8 + 6k^6 + 21k^4 + 20k^2)$
4	$\frac{1}{384} (k^{16} + 12k^{12} + 12k^{10} + 83k^8 + 48k^6 + 180k^4 + 48k^2)$

Fig. 12. Cycle index polynomials. Replacing each z_i with the number k of colors yields the reduced cycle index summarized in Fig. 13.

Fig. 13. Reduced cycle index polynomials. These polynomials generate the subtables of Fig. 8 corresponding to the action of Id_k on the colors.

List(Collected(gNumbers), Reversed);
 [[2, [0, 0, 0, 1]], [3, [0, 2, 0, 0]],
 [2, [2, 1, 0, 0]], [1 [4, 0, 0, 0]]]

The list after each coefficient indicates the exponents of the z_i terms. For example, the element [2, [0, 0, 0, 1]] corresponds to the monomial term $2z_1^0z_2^0z_3^0z_4^1 = 2z_4$. Taken together, the four collected terms produce the polynomial

$$p(z_1, z_2, z_3, z_4) = 2z_4^1 + 3z_2^2 + 2z_1^2z_2^1 + 1z_1^4$$

from which the cycle index Z is found using $Z = p/|G|$. Using GAP, one can therefore determine the cycle index for the full symmetry group and the direct symmetry group acting on the n -simplex or on the n -cube. The results are given in Fig. 12. Substituting k for each z_i yields the reduced cycle indices in Fig. 13.

The asymptotic growth of each reduced cycle index polynomial is determined by its leading term (in which k has the largest exponent). This leading term corresponds to the cycle structure of the identity element acting on the set of d vertices. The identity permutation consists of d cycles, each of length 1, which contributes z_1^d to the cycle index. There can be no term with an exponent larger than d because d is the degree of the group action on the set. The number of vertices is $n + 1$ for an n -simplex and is 2^n for an n -cube. Hence, the reduced cycle index is dominated by the term k^{n+1} for a simplex and by k^{2^n} for a cube. Group actions on the set of colors, such as reversal or full permutation, make some colorings become equivalent and thus reduce the number of orbits compared to the action of Id_k . Hence, the largest number of equivalence classes for a simplex or

cube occurs when the color group is the identity. For any fixed n , as k increases, we have:

$$Z(S_{n+1}; k, k, \dots, k) \sim \frac{k^{n+1}}{(n+1)!}$$

$$Z(A_{n+1}; k, k, \dots, k) \sim \frac{2k^{n+1}}{(n+1)!}$$

$$Z(FC_n; k, k, \dots, k) \sim \frac{k^{2^n}}{2^n n!}$$

$$Z(DC_n; k, k, \dots, k) \sim \frac{k^{2^n}}{2^{n-1} n!}.$$

These approximations indicate the relative complexity of enumerating cases for simplexes and cubes with k colors. Note that the growth in the number of simplex cases is dwarfed by the number of cube cases. We have not yet computed the equations that result from applying non-trivial color groups; we plan to present those results in a future paper.

6 SUMMARY

Marching Cubes (MC), and algorithms like it, share the essential feature of applying geometric substitution to colored polytopes, producing substitopes. These algorithms vary in the choice of symmetry groups (direct symmetry or full symmetry) acting on vertices and permutation groups (identity or reversal or symmetric) acting on colors. They vary in the choice of polytope (n -cube or n -simplex) and in the choice of dimension n and number of colors k . The color is merely an abstraction of any function evaluated at a vertex.

We presented a technique for enumerating the cases of polytope colorings, where two colorings belong to the same

equivalence class (or orbit) if a group action sends one coloring to the other. A software tool for computational group theory, called GAP, can enumerate all the orbits. Not only can GAP count the number of orbits automatically, but it can also enumerate the members of each orbit. This capability eliminates a very difficult part of developing any substitope algorithm. This tool works well for small values of n and k . One benefit of using GAP is that it automatically and independently confirms the results previously announced for counting the cases in four-dimensional MC [8], [7], [6], results that cannot reasonably be checked by hand and that have been of interest to mathematicians since the 1871 work by Jevons. Moreover, it permits us to predict the case-counts for other substitope algorithms that have yet to be invented, simply by treating the shape symmetry and the color symmetry as parameters of the case-counting function. The parameters impose a clear taxonomy on this collection of algorithms where no such organization has heretofore been suggested. This indicates that a very deep, very generic underlying algorithm for visualization exists which can be incarnated in many different ways. The taxonomy also shows the intimate connection between group theory, geometry, and visualization using substitope algorithms.

ACKNOWLEDGMENTS

The authors gratefully acknowledge support by US National Science Foundation grants CISE-#0083898 and FRG-#0101429 and by the Scottish Higher Education Funding Council (SHEFC) and the Particle Physics and Astronomy Research Council (PPARC) for use of the Beowulf cluster to run ParGAP on the larger problem sizes. They thank Eric Klassen for fruitful discussions about group actions and orbits.

REFERENCES

- [1] W.E. Lorenson and H.E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Proc. SIGGRAPH 1987*, pp. 163-169, 1987.
- [2] H.S.M. Coxeter, *Regular Polytopes*. Dover Publications, 1973.
- [3] D.C. Banks and S. Linton, "Counting Cases in Marching Cubes: Toward a Generic Algorithm for Producing Substitopes," *Proc. Visualization 2003*, 2003.
- [4] J. Bloomenthal, "Polygonization of Implicit Surfaces," *Computer-Aided Geometric Design*, vol. 5, pp. 341-355, 1988.
- [5] H.-W. Shen and C.R. Johnson, "Sweeping Simplices: A Fast Iso-Surface Extraction Algorithm for Unstructured Grids," *Proc. IEEE Visualization 1995*, pp. 143-151, 1995.
- [6] S.F. Kirchberg, "Marching Hypercubes—ein Verfahren zur Konstruktion von Hyperflächen aus 4D-Rasterdaten," Universität Dortmund, diplomarbeit am Lehrstuhl 7, 1993.
- [7] J.C. Roberts and S. Hill, "Piecewise Linear Hypersurfaces Using the Marching Cubes Algorithm," *Visual Data Exploration and Analysis VI*, *Proc. SPIE*, R. Erbacher and A. Pang, eds., pp. 170-181, Jan. 1999.
- [8] P. Bhaniramka, R. Wenger, and R. Crawfis, "Isosurfacing in Higher Dimensions," *Proc. IEEE Visualization 2000*, pp. 267-273, 2000.
- [9] G.M. Nielson and J. Sung, "Interval Volume Tetrahedrization," *Proc. IEEE Visualization 1997*, pp. 221-228, 1997.
- [10] C. Weigle and D.C. Banks, "Complex-Valued Contour Meshing," *Proc. IEEE Visualization 1996*, pp. 173-180, 1996.
- [11] H.-C. Hege, M. Seebass, D. Stalling, and M. Zöckler, "A Generalized Marching Cubes Algorithm Based on Non-Binary Classifications," Technical Report SC 97-05, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Dec. 1997.
- [12] D. Stalling, M. Zöckler, O. Sander, and H.-C. Hege, "Weighted Labels for 3D Image Segmentation," Technical Report SC 98-39, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Dec. 1998.
- [13] Module: GMC, Template Graphics Software (User manual entry for amira visualization software), 2000.
- [14] K. Montgomery, C. Bruyns, and S. Wildermuth, "A Virtual Environment for Simulated Rat Dissection: A Case Study of Visualization for Astronaut Training," *Proc. IEEE Visualization 2001*, pp. 509-512, 2001.
- [15] H.-C. Hege, "Enumeration of Symmetry Classes in Mesh Generation," *Report on Dagstuhl Seminar 9821: Hierarchical Methods in Computer Graphics*, M. Gross, H. Müller, P. Schröder, and H.-P. Seidel, eds., pp. 9-10, May 1998.
- [16] G.M. Nielson and R. Franke, "Computing the Separating Surface for Segmented Data," *Proc. IEEE Visualization 1997*, pp. 229-233, 1997.
- [17] H.-W. Shen, C.D. Hansen, Y. Livnat, and C.R. Johnson, "Isosurfacing in Span Space with Utmost Efficiency (ISSUE)," *Proc. IEEE Visualization 1996*, pp. 287-294, 1996.
- [18] J. Wilhelms and A.V. Gelder, "Octrees for Faster Isosurface Generation," *ACM Trans. Graphics*, vol. 11, no. 3, pp. 201-227, July 1992.
- [19] P. Cignoni, P. Marino, C. Montani, E. Puppo, and R. Scopigno, "Speeding Up Isosurface Extraction Using Interval Trees," *J. Symbolic Logic*, vol. 3, no. 2, pp. 158-170, 1997.
- [20] R.S. Gallagher, "Span Filter: An Optimization Scheme for Volume Visualization of Large Finite Element Models," *Proc. IEEE Visualization*, 1991.
- [21] A. Lindenmayer, "Developmental Systems without Cellular Interaction, Their Languages and Grammars," *J. Theoretical Biology*, vol. 30, pp. 455-484, 1971.
- [22] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*, 1990.
- [23] A. Glassner, "A Tutorial on Geometric Replacements," *IEEE Computer Graphics and Applications*, vol. 12, no. 1, pp. 22-36, Jan. 1992.
- [24] M. Lounsbery, T.D. DeRose, and J. Warren, "Multiresolution Analysis for Surfaces of Arbitrary Topological Type," *ACM Trans. Graphics*, vol. 16, no. 1, pp. 34-73, 1997.
- [25] L. Kobbelt, S. Campagna, and H.-P. Seidel, "A General Framework for Mesh Decimation," *Proc. Graphics Interface*, pp. 43-50, 1998.
- [26] A. Cayley, "On the Theory of Groups, as Depending on the Symbolic Equation $\theta^n = 1$," *Philosophical Magazine*, vol. 7, pp. 40-47, 1854.
- [27] J.G. Fraleigh, *A First Course in Abstract Algebra*, sixth ed. Addison-Wesley, 1998.
- [28] P.M. Cohn, *Algebra, volume 1*, second ed. Wiley, 1984.
- [29] Á. Seress, "An Introduction to Computational Group Theory," *Notices of the AMS*, vol. 44, no. 6, pp. 671-679, June/July 1997.
- [30] M. Schönert, *GAP—Groups, Algorithms and Programming*, Lehrstuhl D für Mathematik, RTWH, Aachen, 1994.
- [31] W.S. Jevons, "Solution of the Inverse or Inductive Problem, Involving Two Classes," *Proc. Manchester Literary and Philosophical Soc.*, vol. xi, pp. 65-68, Dec. 1871.
- [32] W.S. Jevons, *The Principles of Science: A Treatise on Logic and Scientific Method*. 1874, second edition, reprinted with corrections, 1879, 1883, 1887, 1892, 1900, 1905, 1907, 1913.
- [33] W.K. Clifford, *Mathematical Papers*, R. Tucker, ed. London: MacMillan and Co., 1882, introduction by H.J. Stephen Smith; reprinted by Chelsea, N.Y., 1968.
- [34] G. Pólya, "Sur les Types des Propositions Composées," *J. Symbolic Logic*, vol. 5, no. 3, pp. 98-103, Sept. 1940.
- [35] W. Burnside, *Theory of Groups of Finite Order*, second ed. London: Cambridge Univ. Press, 1911.
- [36] G. Pólya, "Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und Chemische Verbindungen," *Acta Math.*, vol. 68, pp. 145-254, 1937.
- [37] G. Pólya and R.C. Read, *Combinatorial Enumeration of Groups, Graphs, and Chemical Compounds*. New York: Springer-Verlag, 1987.
- [38] "GAP—Groups, Algorithms, and Programming, Version 4.3," The GAP Group, 2002, <http://www.gap-system.org>.
- [39] W.K. Clifford, "On the Types of Compound Statement Involving Four Classes," *Proc. Manchester Literary and Philosophical Soc.*, vol. xvi, pp. 88-101, Jan. 1877.

Biographies and photos for the authors were unavailable at the publication time.