



# Partial duality for ribbon graphs, III: a Gray code algorithm for enumeration

Jonathan L. Gross<sup>1</sup> · Toufik Mansour<sup>2</sup> · Thomas W. Tucker<sup>3</sup>

Received: 8 November 2020 / Accepted: 18 March 2021 / Published online: 2 April 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

Partially Poincaré-dualizing an embedded graph  $G$  on an arbitrary subset of edges was defined geometrically by Chmutov, using ribbon graphs. Part I of this series of papers introduced the *partial-duality polynomial*, which enumerates all the possible partial duals of the graph  $G$ , according to their Euler-genus, which can change according to the selection of the edge subset on which to dualize. Ellis-Monaghan and Moffatt have expanded the partial-duality concept to include the *Petrie dual*, the *Wilson dual*, and the two *triviality* operators. Abrams and Ellis-Monaghan have given the five operators the collective name *twualities*. Part II of this series of papers derived formulas for partial-twuality polynomials corresponding to several fundamental sequences of embedded graphs. Here in Part III, we present an algorithm to calculate the partial-twuality polynomial of a ribbon graph  $G$ , for all twualities, which involves organizing the edge subsets of  $G$  into a hypercube and traversing that hypercube via a Gray code.

**Keywords** Ribbon graph · Partial duality · Poincaré dual · Petrie dual · Rotation system · Euler-genus · Genus polynomial · Gray code

**Mathematics Subject Classification** Primary: 05C10

---

Jonathan Gross is supported by Simons Foundation Grant No. 315001.

Thomas Tucker is supported by Simons Foundation Grant No. 317689.

---

✉ Jonathan L. Gross  
gross@cs.columbia.edu

Toufik Mansour  
tmansour@univ.haifa.ac.il

Thomas W. Tucker  
ttucker@colgate.edu

<sup>1</sup> Department of Computer Science, Columbia University, New York, NY 10027, USA

<sup>2</sup> Department of Mathematics, University of Haifa, 3498838 Haifa, Israel

<sup>3</sup> Department of Mathematics, Colgate University, Hamilton, NY 13346, USA

This third paper on *ribbon graphs*, after [10] and [11], presents an algorithm to enumerate their *partial duals* according to Euler-genus. This algorithm applies to several partial dualities and *trialities*, not just to partial Poincaré duality. We assume some familiarity with the fundamentals of graph embeddings beyond planarity issues, as presented, for instance, by [13] or [22], and also with ribbon graphs, as described by [9]. Prior familiarity with *flags* and *monodromy* would be helpful, but is not assumed.

The algorithm has three principal procedures:

- The *initialization procedure* has as input parameters the number  $N$  of edges of a ribbon graph  $G$  and a *uni-rotation system*  $W$  for  $G$ . A uni-rotation is roughly describable as a cycle of *quarter-edges* (or equivalently, of flags) incident at a vertex. From this information, the procedure constructs what we call an *extended monodromy module* comprising the monodromy and some additional information about the embedding it represents.
- The *main procedure* examines all the partial duals of  $G$ , conceptualized as the nodes of a hypercube. At each node, it updates the current module, so that it specifies the next partial dual along a *Gray code*, and it passes some data to a procedure that calculates the Euler-genus into a running count of the occurrences of Euler-genus values.
- The *Euler-genus procedure* has as input parameters the number of edges of  $G$  and the *bi-rotations* for vertices and faces of the current partial dual. A bi-rotation is roughly described as a pair of cycles, each of which represents a direction of traversal of a set of edges.

In Sect. 1, we review the definitions of graphs, graph embeddings, *rotation systems* and *ribbon graphs*. Section 2 briefly reviews *flags* and the *monodromy* paradigm for specifying a graph embedding and resolves their relation to ribbon graphs. In Sect. 3, we review the duality and triality operators on a graph embedding. Section 4 describes the corresponding partial operators and the polynomials that enumerate the partial duals. Section 5 presents the algorithm for calculating the polynomials. We have used a Maple<sup>®</sup> implementation of our algorithm as an experimental research aid in developing the content of [10] and [11].

Italics on various technical terms indicates that their definitions or relevant descriptions appear later, usually along with some developmental detail.

## 1 Introduction

In general, we allow a *graph* to have loops and multi-edges. Each edge is construed to have two *half-edges* (that meet midway along the length of the edge). We label the edges of a graph by the integers  $1, 2, \dots, n$ . A *direction of traversal* of an edge may be given by specifying the half-edge that is traversed last (where a traversal in that direction terminates). If we are specifying graph embeddings by traditional rotation systems or by ribbon graphs, we may denote the half-edges of edge  $j$  by  $j^+$  and  $j^-$ . However, if specification involves *monodromy*, then a half-edge corresponds to the two *flags* incident on that half-edge. A graph is taken to be connected, unless it is evident from context that we mean otherwise.

All *embeddings are cellular*, so that the interior of every region is homeomorphic to an open disk. The *genus of an orientable surface*  $S$  is denoted by  $\gamma(S)$ . The *orientable surface of genus  $j$*  is denoted  $S_j$  and the *non-orientable surface of crosscap number  $k$*  is denoted  $N_k$ .

The word *map* is a synonym for “graph embedding”. Writing  $\mathbf{Map}(G)$  is simply an explicit way to emphasize the embedding, rather than the underlying graph.

*Rotation systems* and *monodromy* are the two forms of combinatorial specification of a graph embedding that are used within our algorithm for calculating partial-dual polynomials. In this section, we briefly review rotation systems and ribbon graphs. In the next section, we explain why our algorithm uses both rotation systems and monodromy.

**Notational Conventions.** For any two permutations  $\alpha$  and  $\beta$  of the same set of objects, the notation  $\alpha\beta$  indicates the *composition* in which  $\alpha$  is applied before  $\beta$ . Also, we use  $|\alpha|$  to denote the *number of cycles in a permutation*  $\alpha$ .

### 1.1 Rotation systems

For an oriented embedding of a graph  $G$ , the induced *rotation* at a vertex  $v$ , commonly denoted by  $\rho_v$ , is a cyclic permutation of the half-edges incident at  $v$ , in the order consistent with the orientation of the surface. In the context of rotations, we define the permutation  $\lambda$  on the half-edges of a graph  $G$  to be the involution that transposes a half-edge of an edge  $e$  with the other half-edge of  $e$ . The composition  $\rho = \prod_{v \in V(G)} \rho_v$  is called the *induced rotation system* for  $G$ . It is commonly denoted by  $\rho$ . Within the monodromy paradigm, the meanings of the notations  $\lambda$  and  $\rho$  are a little different.

Conversely, given a graph  $G$ , we might specify at every vertex  $v$  an arbitrary cyclic permutation  $\bar{\rho}_v$  of the half-edges incident at  $v$ , and then call  $\bar{\rho} = \prod_{v \in V(G)} \bar{\rho}_v$  an *assigned rotation system*. It is well-known that an assigned rotation system induces an oriented embedding of  $G$ , in which the cycles of the permutation  $\bar{\rho}\lambda$  trace the face-boundary walks of an oriented embedding of  $G$ . This is constructed by fitting a polygon with the appropriate number of sides to each of those cycles. We have the following immediate implication:

**Proposition 1.1** *The total number of oriented embeddings of a graph  $G$  equals*

$$\prod_{v \in V(G)} (\delta(v) - 1)! \quad (1.1)$$

where  $V(G)$  is the vertex set of  $G$  and  $\delta(v)$  is its valence (or degree).  $\square$

### 1.2 Ribbon graphs

A *ribbon graph* can be specified as a set of disks called *vertex disks* and a set of rectangular strips called *ribbons* (or sometimes, “edges”), such that each of the two ends of each ribbon is attached to a vertex disk along an arc; no two of these arcs meet. The same mathematical object was previously called a *band decomposition* by [13].

We implicitly associate to a ribbon graph a set of polygonal disks called *face disks*, one for each boundary component, and we envision the boundary circuit of each face disk as pasted homeomorphically to the corresponding boundary component. The visually evocative terminology “ribbon graph” appears to have originated with [19].

Figure 1 shows a ribbon graph with two vertex disks and five ribbons. It has become customary to label the ribbons with consecutive positive integers, as illustrated in this figure.

Like some of our predecessors in the investigation of partial duals, we sometimes leave it to the reader to infer whether a particular occurrence of the phrase “ribbon graph” includes the face disks or not. For instance, the faces disks are absent (at least, momentarily) when we give a ribbon an extra half-twist. However, when we refer to the “Euler-genus of a ribbon graph”, we include the face disks. By tracing along the boundary of the ribbon graph in Fig. 1, we see that there is only one boundary component. Thus, saying that its Euler-genus is four means that the embedding surface has Euler-genus four.

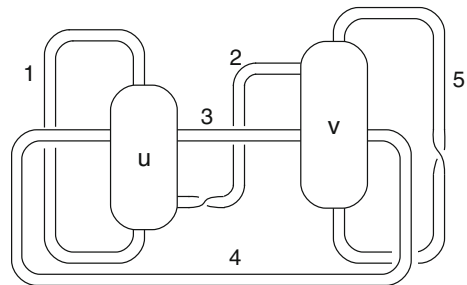
Our present objective is to facilitate calculation of the Euler-genus of the partial duals of a ribbon graph, continuing the approach of [10] and [11]. We present a Gray code algorithm to calculate the *partial-dual polynomials* that provide an inventory of the partial duals of a ribbon graph, according to Euler-genus, for each of the duality operators: *Poincaré duality* (\*), *Petrie duality* (×), and *Wilson duality* (\*\*). Our algorithm is *loopless*, with execution time  $\theta(2^n)$  for an  $n$ -edge graph.

**Example 1.2** Let  $G$  be the 5-edge ribbon graph of Fig. 1. The Poincaré, Petrie, and Wilson partial-dual polynomials are, respectively,

$$\begin{aligned}\partial\mathcal{E}_G^*(z) &= 2z^2 + 6z^3 + 16z^4 + 8z^5, \\ \partial\mathcal{E}_G^\times(z) &= 5z^2 + 15z^3 + 12z^4, \text{ and} \\ \partial\mathcal{E}_G^{**}(z) &= 2z^3 + 13z^4 + 17z^5.\end{aligned}$$

These partial-duality polynomials were calculated by computer. An effort to calculate any of them by hand indicates the benefit of automated calculation, even for such a small ribbon graph. In [10] and [11], we have used analytic methods involving symme-

**Fig. 1** A ribbon graph with two vertex disks and five ribbons



tries and recursions to derive partial-duality polynomials for ladder graphs, bouquets, dipoles, and cycle graphs.

## 2 Quarter-ribbons, flags, and monodromy

In this section, we describe a scheme for assigning flag labels. We see subsequently how this flag-labeling scheme simplifies calculations based on the monodromy.

### 2.1 Flags and quarter-ribbons, and monodromy

Figure 2 illustrates the decomposition of  $Map(G)$  into *flags* in the vicinity of an edge that has been designated as edge  $k$ . There are several steps to this decomposition.

Given an  $n$ -ribbon graph  $G$ , we construct the *barycentric subdivision* of  $Map(G)$ , as described in [11]. Each face of  $Bary(G)$  is called a *flag*. The four flags incident on ribbon  $k$  induce a decomposition into four *quarter-ribbons*, to which we assign the *canonical labels*

$$4k-3, 4k-2, 4k-1, \text{ and } 4k,$$

as illustrated in Fig. 2, for  $k = 1, 2, \dots, n$ . In turn, assigning to each flag the same label as its quarter-ribbon is called the *canonical labeling* of the flags.

The *monodromy* of a ribbon graph  $G$  consists of three full involutions  $r_0, r_1$ , and  $r_2$ , which are usually said to act on the flags of  $Map(G)$ . We will say, equivalently, that they act on the quarter-ribbons.

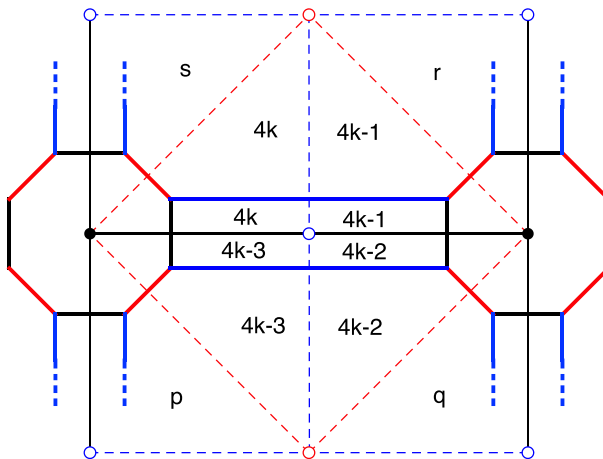


Fig. 2 The four quarter-ribbons of ribbon  $k$ , labeled as shown, along with four adjacent flags  $p, q, r, s$

$r_0$  transposes any two quarter-ribbons on the same side of an edge.

In Fig. 2,  $r_0$  includes the 2-cycles  $(4k-3, 4k-2)$  and  $(4k-1, 4k)$ . (2.1)

$r_1$  transposes any two quarter-ribbons in the same corner of  $Map(G)$ .

including  $(p, 4k-3)$ ,  $(q, 4k-2)$ ,  $(r, 4k-1)$ , and  $(s, 4k)$ . (2.2)

$r_2$  transposes any two quarter-ribbons that share a half-ribbon of  $G$ .

In Fig. 2,  $r_2$  includes the 2-cycles  $(4k-3, 4k)$  and  $(4k-2, 4k-1)$ . (2.3)

When  $r_0$  and  $r_2$  correspond to a canonical quarter-ribbon labeling, the triple  $\langle r_0, r_1, r_2 \rangle$  is called a *canonical monodromy*. We observe that the involutions  $r_0$  and  $r_2$  commute. A convenient accessory to the monodromy of a ribbon graph is the composite involution  $\lambda = r_0 r_2$ .

**Remark 2.1** In the context of rotation systems, the notation  $\lambda$  designates an involution that transposes the half-edges of each edge. In the context of monodromy, it designates an involution that transposes two pairs of quarter-ribbons for each edge.

Two of the involutions for the monodromy of any embedded graph  $G$  whose map has the canonical flag labeling are always of the same form. For the ribbon graph of Fig. 1 with five edges, or for any other graph with five edges, we have

$$r_0 = (1\ 2)(3\ 4)(5\ 6)(7\ 8)(9\ 10)(11\ 12)(13\ 14)(15\ 16)(17\ 18)(19\ 20). \quad (2.4)$$

$$r_2 = (1\ 4)(2\ 3)(5\ 8)(6\ 7)(9\ 12)(10\ 11)(13\ 16)(14\ 15)(17\ 20)(18\ 19). \quad (2.5)$$

Since  $\lambda = r_0 r_2$ , we also have

$$\lambda = (1\ 3)(2\ 4)(5\ 7)(6\ 8)(9\ 11)(10\ 12)(13\ 15)(14\ 16)(17\ 19)(18\ 20). \quad (2.6)$$

**Example 2.2** To write the involution  $r_1$  for Fig. 1, we draw it with quarter-edge labeling, as shown in Fig. 3. From the corners of either the ribbon graph (left) or of its *rotation projection* (right), as defined by [13, §3.2.5], we read the involution

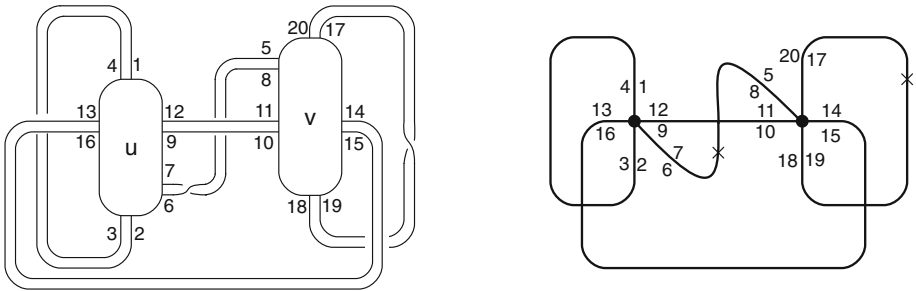
$$r_1 = (1\ 12)(2\ 6)(3\ 16)(4\ 13)(5\ 20)(7\ 9)(8\ 11)(10\ 18)(14\ 17)(15\ 19). \quad (2.7)$$

## 2.2 Face bi-rotations and vertex bi-rotations

Tracing along the boundary of the ribbon graph in Fig. 3, we can record the following cyclic sequence of quarter-ribbons encountered as the traversal (starting clockwise at quarter-edge 1) reaches the end of a ribbon:

$$1, 11, 7, 10, 17, 13, 3, 15, 20, 6. \quad (2.8)$$

Since the traversal (2.8) traces both sides of each of the five ribbons, this must be the boundary walk of the only face of the embedding. Since there are two vertices and five edges, we can now calculate the Euler characteristic of the embedding as



**Fig. 3** (left) Quarter-ribbon labeling on the ribbon graph of Fig. 1. (right) A rotation projection of that ribbon graph

$$\chi = 2 - 5 + 1 = -2.$$

Since the embedding is non-orientable, we recognize the surface as  $N_4$ , the surface with four crosscaps. An alternative way to calculate this face-boundary walk is via the composition  $r_1r_0$ , using (2.4) and (2.7).

$$r_1r_0 = (1\ 11\ 7\ 10\ 17\ 13\ 3\ 15\ 20\ 6)(2\ 5\ 19\ 16\ 4\ 14\ 18\ 9\ 8\ 12) \tag{2.9}$$

We see that the second 10-cycle of ribbons traversed corresponds to the first 10-cycle in reverse order, and we call these two cycles **reverse face-boundary walks**, or in context, **inverse cycles**. We call (2.9) a **face bi-rotation**, and we denote the composition  $r_1r_0$  by  $\phi$ .

In a dual sense, tracing clockwise around the boundaries of the vertex disks  $u$  and  $v$ , we record these cyclic sequences of quarter-ribbons that are encountered:

$$u. \ 1, 9, 6, 3, 13 \tag{2.10}$$

$$v. \ 5, 17, 15, 18, 11 \tag{2.11}$$

These cyclic sequences are a flag-based representation of what are traditionally called **rotations**. An alternative way to calculate these rotations employs the composition  $r_1r_2$ , using (2.7) and (2.5).

$$r_1r_2 = (1\ 9\ 6\ 3\ 13)(2\ 7\ 12\ 4\ 16)(5\ 17\ 15\ 18\ 11)(8\ 10\ 19\ 14\ 20) \tag{2.12}$$

We observe that this composition includes two additional 5-cycles, which correspond to quarter-ribbon cycles (2.10) and (2.11) in the opposite direction. We call the composition of the cycles (2.10) and (2.11) a **vertex uni-rotation**. We call the quarter-ribbon permutation (2.12) a **vertex bi-rotation**, and we denote it by  $\rho$ .

**Remark 2.3** An  $n$ -vertex ribbon graph  $G$  has  $2^n$  possible choices of a vertex uni-rotation, each of which corresponds to the choice of a spanning tree. We could draw a (rotation-preserving) planar projection of  $G$  so that each ribbon of the spanning tree is drawn flat, that is, with no twist. However, what simplifies the use of our algorithm

is that, no matter which of these  $2^n$  possible uni-rotations we supply as input data, the initialization procedure (Algorithm 5.2) will construct the same bi-rotation.

**Proposition 2.4** *Let  $\psi$  be a vertex uni-rotation consistent with the canonical monodromy  $(r_0, r_1, r_2)$  for a ribbon graph  $G$ . Then*

- (a)  $\rho = \psi(r_2\psi r_2)^{-1}$  is the corresponding vertex bi-rotation.
- (b)  $\phi = \rho\lambda$  is the corresponding face bi-rotation.
- (c)  $r_1 = \rho r_2$ .

**Proof** This is straightforward.

- (a)  $(r_2\psi r_2)^{-1}$  is evidently the composition of the cycles that are inverses to the cycles of  $\psi$ .
- (b) Since  $\lambda = r_2 r_0$  and  $\rho = r_1 r_2$ , we can employ this sequence of substitutions:

$$\phi = r_1 r_0 = r_1 (r_2 \lambda) = (r_1 r_2) \lambda = \rho \lambda.$$

- (c) Since  $\rho = r_1 r_2$ , we have  $\rho r_2 = r_1$ .

□

It is sometime convenient to use the phrase **extended monodromy** to refer to a collection of permutations that, in addition to  $r_0, r_1$ , and  $r_2$ , also includes some additional permutations, such as  $\lambda, \rho$ , and  $\phi$ .

### 2.3 Why we use two forms of embedding specification

In Sects. 4 and 5, we show how to change the extended monodromy of any partial dual of an embedded graph  $G$  to the extended monodromy of the next partial dual in the Gray code, by applying permutation compositions in Table 1. This is easier than trying to make the change with rotation systems alone. On the other hand, it is half as much effort to specify the initial embedding with a uni-rotation, which involves mentioning only half the quarter-edges, than to specify an involution of the monodromy, which requires writing all the quarter-edges (twice the effort, even though, using a canonical monodromy, specifying only  $r_1$  is sufficient.)

## 3 Ribbon graphs and their dualities

Given an embedding (called the **primal embedding**) of a graph in a surface, we form a **Poincaré dual of  $G$  and its embedding** as follows:

- (1) place a **dual vertex** in the interior of each region of the primal embedding;
- (2) through each primal edge  $e$  (i.e., of the graph  $G$ ), draw a **dual edge  $e^*$**  that joins the dual vertex on one side of  $e$  to the dual vertex on its other side. If a single face of the primal embedding is incident on both sides of edge  $e$ , then the dual edge  $e^*$  is a loop.



**Table 1** Effect of partial-duality operators on the monodromy

$G^{\bullet A}$	$(r_0, r_2, \lambda)^{\bullet A}$	$r_0^{\bullet A}$ $= r_0 s _A$	$r_2^{\bullet A}$ $= r_2 t _A$	$\rho^{\bullet A}$ $= r_1 r_2^{\bullet A}$	$\lambda^{\bullet A}$ $= r_0^{\bullet A} r_2^{\bullet A}$	$\phi^{\bullet A}$ $= r_1 r_0^{\bullet A}$
$G$	$(r_0, r_2, \lambda)$	$r_0 = r_0$	$r_2 = r_2$	$\rho$	$\lambda$	$\phi$
$G^{* A}$	$(r_2, r_0, \lambda)$	$r_0^{* A} = r_0 \lambda _A$	$r_2^{* A} = r_2 \lambda _A$	$\rho \lambda _A$	$\lambda$	$\phi \lambda _A$
$G^{\times A}$	$(\lambda, r_2, r_0)$	$r_0^{\times A} = r_0 r_2 _A$	$r_2^{\times A} = r_2$	$\rho$	$\lambda r_2 _A$	$\phi r_2 _A$
$G^{*\times A}$	$(\lambda, r_0, r_2)$	$r_0^{*\times A} = r_0 r_2 _A$	$r_2^{*\times A} = r_2 \lambda _A$	$\rho \lambda _A$	$\lambda r_0 _A$	$\phi r_2 _A$
$G^{\times* A}$	$(r_2, \lambda, r_0)$	$r_0^{\times* A} = r_0 \lambda _A$	$r_2^{\times* A} = r_2 r_0 _A$	$\rho r_0 _A$	$\lambda r_2 _A$	$\phi \lambda _A$
$G^{*\times* A}$	$(r_0, \lambda, r_2)$	$r_0^{*\times* A} = r_0$	$r_2^{*\times* A} = r_2 r_0 _A$	$\rho r_0 _A$	$\lambda r_0 _A$	$\phi$

According to [2], knowledge of the duality between the cube and the octahedron, and between the dodecahedron and the icosahedron goes back to the “fifteenth book of Euclid”, which [2] estimates as having originated in the sixth century C.E. Poincaré developed algebraic consequences of a generalization of this topological form of duality to higher dimensions.

**Corollary 3.1** *The rotation system for the Poincaré dual of an embedding with rotation system  $\rho$  is given by the composition  $\rho \lambda$ .* □

We recall that the full **Poincaré dual**  $G^*$  of an embedded graph  $G$  (referred to as the “primal” graph) is constructed as follows:

- (1) Into the interior of each primal face, a dual vertex is inserted.
- (2) Through each primal edge, a dual edge is drawn that joins the dual vertex on one side of that primal edge to the dual vertex on the other side. If both sides of the primal edge lie on the same primal face, then the dual vertex in that face is joined to itself.

A completely topological definition of a **Poincaré partial dual** on an edge subset  $A$ , here denoted by  $G^{*|A}$ , was given by Chmutov [5] (see also [9]), and is not reproduced here. Here, as in [11], we represent the Poincaré dual and partial dual by combinatorial operations on the monodromy.

The **Petrie dual**  $G^\times$  of a ribbon graph  $G$  is constructed by detaching one end of each ribbon at the arc where it meets a vertex disk, giving that ribbon an extra half-twist, and then reattaching the end at the same vertex disk arc. For any set  $A$  of ribbons of  $G$ , there is a topological construction of the **partial Petrie dual** on  $A$ , denoted  $G^{\times|A}$  by applying that half-twist operation only to the ribbons in  $A$ . Here we represent the Petrie dual and partial dual by combinatorial operations on the monodromy.

Wilson [23] showed that under composition, the duality operators  $*$  and  $\times$  form a group of order six, now called the **Wilson group**, with the presentation

$$\mathcal{G} = \langle *, \times \mid *^2, \times^2, (*\times)^3 \rangle$$

and is abstractly isomorphic to the symmetric group  $\Sigma_3$ . Since its order is three, the composite operator  $*\times$  is called a **trality**. It is easily proved that the composite operator

$\times*$  is also a triality and that  $(*\times)^2 = \times*$ . The operator  $*\times*$  is of order two and is often called the *Wilson dual* or the *Wilsonian*.

Abrams and Ellis-Monaghan [1] have introduced the word *twualities* to refer to the three dualities and the two trialities collectively.

### 4 Partial-twuality polynomials

The *Euler-genus* of a ribbon graph  $G$  is given by the formula

$$\epsilon(G) = 2 - (|V(G)| - |E(G)| + |F(G)|), \tag{4.1}$$

where  $V(G)$  is the set of vertex disks,  $E(G)$  is the set of ribbons (edges), and  $F(G)$  is the set of face disks. Every partial dual of a ribbon graph  $G$  has the same number of ribbons as  $G$  itself. Thus, to calculate the Euler-genus of some partial dual of  $G$ , it is sufficient, in view of Proposition 2.4, to know the monodromy of that partial dual, since

$$|V(G)| = \frac{|r_1 r_2|}{2} \text{ and } |F(G)| = \frac{|r_1 r_0|}{2}. \tag{4.2}$$

Table 1 (reproduced from [11, Table 3.1]) is our guideline to calculating the partial dual of any ribbon graph for any twuality. The symbols  $s$  and  $t$  in the labels for columns  $r_0^{\bullet|A}$  and  $r_2^{\bullet|A}$  are generic names for the various right multipliers of  $r_0$  and  $r_2$  that appear to the right of the equals sign in the entries within those respective columns.

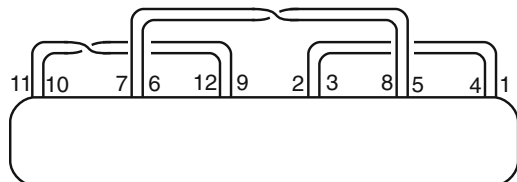
The *partial-duality polynomial* of the ribbon graph  $G$  for each of the twualities is the polynomial for which the coefficient of  $z^r$  is the number of partial duals of  $G$  whose Euler-genus is  $r$ . It follows that if  $G$  has  $n$  ribbons, then, since the number of subsets of ribbons is  $2^n$ , it follows that the sum of those coefficients is  $2^n$ . The five partial-duality polynomials of  $G$  are denoted

$$\partial \mathcal{E}_G^*(z), \partial \mathcal{E}_G^\times(z), \partial \mathcal{E}_G^{**}(z), \partial \mathcal{E}_G^{\times*}(z), \text{ and } \partial \mathcal{E}_G^{***}(z).$$

**Example 4.1** As an example of a monodromy-based calculation, we calculate the Euler-genus of the Petrie partial dual on ribbons 1 and 3 of the quarter-ribbon labeled ribbon graph  $G$  in Fig. 4.

We can see immediately that there is one vertex and that there are three ribbons. For this very small example, we could easily draw the designated Petrie partial dual by twisting ribbon 1 and untwisting ribbon 3 and then doing face-tracing by hand to

**Fig. 4** A ribbon graph with quarter-ribbon labeling



determine that there is only one face. Thus, we now know that  $\epsilon(G^{\times|A}) = 3$ , that is, the Euler-genus of the partial dual is three.

The ribbon graph  $G$  has as a vertex uni-rotation the cyclic permutation

$$(1\ 10\ 6\ 9\ 3\ 5),$$

to which an automated procedure could apply Proposition 2.4, thereby to obtain the vertex bi-rotation and the face bi-rotation

$$\begin{aligned} \rho &= (1\ 10\ 6\ 9\ 3\ 5)(2\ 12\ 7\ 11\ 4\ 8) \\ \phi &= (1\ 12\ 5\ 3\ 7\ 9)(2\ 10\ 8\ 4\ 6\ 11), \end{aligned}$$

and the (extended) monodromy

$$\begin{aligned} r_0 &= (1\ 2)(3\ 4)(5\ 6)(7\ 8)(9\ 10)(11\ 12) \\ r_2 &= (1\ 4)(2\ 3)(5\ 8)(6\ 7)(9\ 12)(10\ 11) \\ r_1 &= (1\ 11)(2\ 9)(3\ 8)(4\ 5)(6\ 12)(7\ 10) \\ \lambda &= (1\ 3)(2\ 4)(5\ 7)(6\ 8)(9\ 11)(10\ 12). \end{aligned}$$

To obtain the face bi-rotation for  $G^{\times\{1,3\}}$ , for instance, we evaluate Table 1 entry in Row  $G^{\times|A}$ , Column  $\phi r_2^{\bullet|A}$  for  $A = \{1, 3\}$ .

$$\phi r_2|_{\{1,3\}} = (1\ 9\ 4\ 6\ 10\ 8)(2\ 11\ 3\ 7\ 12\ 5)$$

Since the face bi-rotation has two cycles, the number of faces is one.

### 5 Algorithm for partial-twuality polynomials

For a given partial-twuality operator  $\bullet$ , in order to use (4.2) to calculate the partial-duality polynomial  $\partial \mathcal{E}_G^\bullet$ , we need to calculate the vertex bi-rotation  $\rho^{\bullet|A}$  and face bi-rotation  $\phi^{\bullet|A}$  of the partial dual  $G^{\bullet|A}$ , for every ribbon subset  $A \subseteq E(G)$ . To accomplish this, we conceptually identify the ribbon subsets with the nodes of the standard Boolean lattice of subsets of  $\{1, 2, \dots, n\}$ , where  $|E(G)| = n$ . Each ribbon  $e$  of  $G$  is identified with one of the  $n$  ordered bits. This identifies each node of the  $n$ -cube with some integer from 0 to  $2^n - 1$ , as well as with a ribbon subset. With that in mind, calculating the inventory of partial duals is a straightforward matter of traversing the  $n$ -cube, while doing the following at each node:

- (1) update the monodromy for the partial dual represented by the node just encountered;
- (2) calculate the numbers of vertex disks and face disks for the ribbon graph specified by that monodromy;
- (3) calculate the Euler-genus of the corresponding embedding, and increment the corresponding counter.

We modularize our algorithm, so that initialization and the calculation of Euler-genus are given as separate procedures.

**Pseudocode Conventions.** We specify algorithms in an *informal pseudocode*, which somewhat resembles Maple<sup>®</sup>.

- By default, program variables (not including input and output parameters) are local. Their data types (e.g., integer, character string, permutation, array) are usually inferred from context.
- The index of a subscripted variable is understood as a coordinate in an array. The default base coordinate of an array is zero.
- For emphasis, we sometimes write instructions that assign a variable its default value.
- Boldface is used for clarity; no other meaning is intended.
- In the pseudocode, for any two permutation-valued variables  $\alpha$  and  $\beta$  of the same set of objects, the notation  $\alpha \cdot \beta$  indicates the composition in which  $\alpha$  is applied before  $\beta$ . This is opposite to the meaning of juxtaposition in prose.

### 5.1 Gray code traversal of the $n$ -cube

A *Gray code* for the  $n$ -cube is a permutation of the non-negative integers  $1, 2, \dots, 2^n - 1$ , such that the binary representations of the Gray codes  $gray(k)$  and  $gray(k+1)$  of two successive integers differ in only one bit. For a ribbon graph  $G$  with ribbons labeled  $1, 2, \dots, n$ , each number

$$k \in \{0, 1, \dots, 2^n - 1\}$$

stands for a subset of ribbons. The  $j^{\text{th}}$  bit in the binary representation of  $gray(k)$  corresponds to ribbon  $j$ , and the number  $k$  corresponds to the subset of  $E(G)$  in which ribbon  $j$  is a member if and only if the  $j^{\text{th}}$  bit of  $gray(k)$  is 1.

The  $0^{\text{th}}$  number in our sequence of integers is  $gray(0) = 0$ . To find the next binary numeral after the binary numeral  $gray(k)$ , we calculate  $lowbit(k)$  to be the lowest order 1-bit of the binary representation of  $k + 1$ , and we change that bit in the binary numeral  $gray(k)$ .

**Algorithm 5.1** (*lowest bit of binary representation of number  $n$* )

- (1)  $lowbit := \text{procedure}(n)$
- (2)  $j = 0 :$
- (3) **if**  $n = 0$  **then return**  $(0) : \text{end if} :$
- (4) **while**  $n \bmod 2^j = 0$  **do**
- (5)      $j := j + 1 : \text{end do} :$
- (6) **return**  $(j) : \text{end proc} :$

**Example 5.1** The following table presents the values of  $lowbit(k + 1)$  and  $gray(k)$  for  $k = 0, 1, \dots, 9$ .

$k$	0	1	2	3	4	5	6	7	8	9	...
$lowbit(k + 1)$	1	2	1	4	1	2	1	8	1	2	...
$gray(k)$	0	1	3	2	6	7	5	4	12	13	...

## 5.2 Initialization procedure

The input parameters of our initialization procedure are the number of ribbons of the ribbon graph  $G$  and the vertex uni-rotation relative to a given spanning tree. The output parameter is an extended monodromy for  $G$ . The pound sign symbol “#” indicates a comment. The presumption of the last few lines is that everything is returned as a single module representing an extended monodromy. The composition of permutations  $\alpha$  and  $\beta$  is denoted by  $\alpha \cdot \beta$ .

**Algorithm 5.2** (*transform a uni-rotation into an extended monodromy*)

```

(1) ExtMono:=procedure( $N, W$ )
(2) ## parameter  $N$  is number of ribbons of graph  $G$ , i.e.,  $N = |E(G)|$ 
(3) ## parameter  $W$  is uni-rotation for  $G$ 
(4) #
(5) ## canonical involution  $r_0[1..N] = (1, 2)(3, 4) \dots (4N-3, 4N-2)(4N-1, 4N)$ 
(6)  $r_{0_0} := ()$  : # initialization
(7) for  $j$  from 1 to  $N$  do # use canonical flag labels
(8)  $r_{0_j} := (4j-3, 4j-2)(4j-1, 4j)$  : end do:
(9) #
(10) ## canonical involution  $r_2[1..N] = (1, 4)(2, 3) \dots (4N-3, 4N)(4N-2, 4N-1)$ 
(11)  $r_{2_0} := ()$  : # initialization
(12) for  $j$  from 1 to  $N$  do # use canonical flag labels
(13)  $r_{2_j} := (4j-3, 4j)(4j-2, 4j-1)$  : end do:
(14) #
(15) ## canonical involution  $\lambda[1..N] = (1, 3)(2, 4) \dots (4N-3, 4N-1)(4N-2, 4N)$ 
(16)  $LAM := ()$  : # initialization
(17) for  $j$  from 1 to  $N$  do
(18)  $\lambda_j := r_{0_j} \cdot r_{2_j}$  :
(19)  $LAM := LAM \cdot \lambda_j$  : end do:
(20) #
(21) ## construct initial vertex bi-rotsys  $\rho$  and initial face bi-rotsys  $\phi$  via Prop 2.4
(22)  $UR := W$  :
(23) for  $j$  from 1 to  $N$  do ## loop installs co-rotsys labels in  $UR$ 
(24)  $UR := r_{2_j} \cdot UR \cdot r_{2_j}$  : end do:
(25)  $\rho := W \cdot UR^{-1}$  : #  $\rho$  is now vertex bi-rotsys
(26)  $\phi := \rho \cdot LAM$  : #  $\phi$  is now face bi-rotsys
(27) #
(28) ## construct extended monodromy module  $M$  and return
(29)  $M_1 := r_0$  :  $M_2 := r_2$  :  $M_3 := \lambda$  :  $M_4 := \rho$  :  $M_5 := \phi$  :
(30) return( $M$ ) : end proc:

```

## 5.3 Euler-genus of a ribbon graph

The input parameters of our procedure for calculating the Euler-genus of a ribbon graph  $G$  are the number of ribbons, the vertex bi-rotation, and the face bi-rotation of  $G$ . The output parameter is the Euler-genus.

**Algorithm 5.3** (calculate the Euler-genus)

- (1) **EulerGenus** := procedure( $N, \rho, \phi$ )
- (2) **##** input parameter  $N$  is number of ribbons of graph  $G$ , i.e.,  $N = |E(G)|$
- (3) **##** input parameters  $\rho$  and  $\phi$  are vertex bi-rotation, face bi-rotation
- (4) **#**
- (5)  $v := |\rho|/2$  : # number of cycles of  $\rho =$  twice number of vertices
- (6)  $f := |\phi|/2$  : # number of cycles of  $\phi =$  twice number of faces
- (7)  $\chi := v - N + f$  : # Euler characteristic
- (8)  $\epsilon := 2 - \chi$  : # Euler-genus for CONNECTED ribbon graph
- (9) **return** ( $\epsilon$ ) : **end proc** :

**5.4 A loopless algorithm for the partial-duality polynomial**

A *loopless algorithm* is an imperative algorithm that generates an initial object in constant time, and then successive combinatorial objects, in linear time in the size of the input [7,15]. It is clear that Algorithm 5.4 is loopless. The main loop traverses a Gray code in the hypercube of ribbon subsets of the ribbon graph  $G$  received as input.

The input parameters are the number  $N$  of ribbons of  $G$ , a vertex uni-rotation for  $G$ , and the duality operation  $\bullet$  of interest. Within the loop at the node  $j$  representing the ribbon subset  $A$ , it executes the following steps:

- (1) calculate the monodromy of the partial dual  $G^{\bullet|A}$  represented by node  $j$  from the monodromy of the previous node, based on the permutation compositions indicated in Table 1;
- (2) calculate the Euler-genus  $\epsilon(G^{\bullet|A})$  of the ribbon graph corresponding to node  $j$ ;
- (3) increment the counter for the coefficient of  $z^{\epsilon(G^{\bullet|A})}$ .

**Algorithm 5.4** (partial-twuality polynomial for ribbon graph  $G$ )

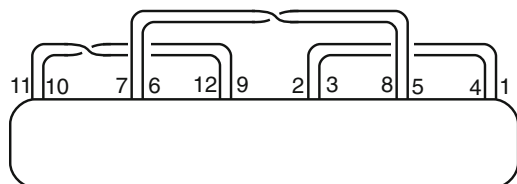
- (1) **PTwuPoly** := procedure( $N, W, TC$ )
- (2) **##** input parameter  $N$  is number of edges of  $G$ , i.e.,  $N = |E(G)|$
- (3) **##** input parameter  $W$  is uni-rotation for  $G$
- (4) **##** input parameter  $TC$  is code for operation  $*$ ,  $\times$ ,  $**$ ,  $*\times$ , or  $\times*$
- (5) **#**
- (6) **### INITIALIZE**
- (7)  $M := \text{ExtMono}(N, W)$  : # transform uni-rot  $W$  into extended monodromy  $M$
- (8)  $r0 := M_1$  :  $r2 := M_2$  :  $\lambda := M_3$  :  $\rho := M_4$  :  $\phi := M_5$  :
- (9)  $\text{coeff} := \text{vector}(0..N, 0)$  : # initialize coeffs of partial dual poly at zero
- (10)  $\text{Gray} := \text{vector}(1..N, 0)$  : # kludge used only for trialities  $**$  and  $\times*$
- (11) **#**
- (12) **### MAIN LOOP: ACCUMULATE EULER-GENUS COUNTS**  
while TRAVERSING GRAY CODE
- (13) **for**  $j := 0$  **to**  $2^N - 1$  **do**
- (14) **if**  $TC = *$  **then**  $\rho := \rho \cdot \lambda_{\text{lowbit}(j)}$  :  $\phi := \phi \cdot \lambda_{\text{lowbit}(j)}$   
 $r0_{\text{save}} := r0_{\text{lowbit}(j)}$  :  $r0_{\text{lowbit}(j)} := r2_{\text{lowbit}(j)}$  :  $r2_{\text{lowbit}(j)} := r0_{\text{save}}$  :
- (15) **elif**  $TC = \times$  **then**  $\phi := \phi \cdot r2_{\text{lowbit}(j)}$  :  
 $r0_{\text{save}} := r0_{\text{lowbit}(j)}$  :  $r0_{\text{lowbit}(j)} := \lambda_{\text{lowbit}(j)}$  :  $\lambda_{\text{lowbit}(j)} := r0_{\text{save}}$  :

```

(16) elif TC = *×* then
    ρ:=ρ·r0lowbit(j):
    r2save:=r2lowbit(j) : r2lowbit(j):=λlowbit(j) : λlowbit(j):=r2save :
(17) elif TC = *× then
(18) if j > 0and Graylowbit(j) = 0then
    ρ:=ρ·λlowbit(j) : φ:=φ·r2lowbit(j) : r0save:=r0lowbit(j) :
    r0lowbit(j):=λlowbit(j) : λlowbit(j):=r2lowbit(j) : r2lowbit(j):=r0save :
(19) else
    ρ:=ρ·r0lowbit(j) : φ:=φ·λlowbit(j) : r0save:=r0lowbit(j) :
    r0lowbit(j):=r2lowbit(j) : r2lowbit(j):=λlowbit(j) : λlowbit(j):=r0save :
end if
(20) elif TC = × * then
(21) if j > 0and Graylowbit(j) = 0then
    ρ:=ρ·r0lowbit(j) : φ:=φ·λlowbit(j) : r0save:=r0lowbit(j) :
    r0lowbit(j):=r2lowbit(j) : r2lowbit(j):=λlowbit(j) : λlowbit(j):=r0save :
(22) else
    ρ:=ρ·λlowbit(j) : φ:=φ·r2lowbit(j) : r0save:=r0lowbit(j) :
    r0lowbit(j):=λlowbit(j) : λlowbit(j):=r2lowbit(j) : r2lowbit(j):=r0save :
end if
(23) else print("INVALID DUALITY CODE"); return; # execution halts
end if
(24) ## the vector Gray is a kludge used only for trialities *× and ×*
(25) if j > 0then Graylowbit(j) := (Graylowbit(j) + 1) mod 2 : end if:
(26) ## CALCULATE and RECORD EULER-GENUS
(27) ε := EulerGenus(N, ρ, φ):
(28) coeffε := coeffε + 1:
(29) end do ## END of MAIN LOOP
(30) #
(31) ### BUILD and PRINT PARTIAL-TWUALITY POLYNOMIAL
(32) poly := 0: # initialize polynomial
(33) for j := 0 to N do
(34) poly := poly + coF[j] * zj: od:
(35) print(poly) :
(36) endproc:
    
```

**Example 5.2** We reproduce the ribbon graph of Fig. 4. To calculate the Poincaré-dual polynomial to that ribbon graph, we supply to Algorithm 5.4, the partial-twuality polynomial algorithm, the input parameters:

Fig. 5 Reproduction of Fig. 4



$$\begin{aligned}
 N &= 3 \\
 W &= (1\ 10\ 6\ 9\ 3\ 5) \\
 TC &= *
 \end{aligned}$$

The procedure Algorithm 5.2 returns the extended monodromy:

$$\begin{aligned}
 r_0[1, 2, 3] &= (1\ 2)(3\ 4), (5\ 6)(7\ 8), (9\ 10)(11\ 12) \\
 r_2[1, 2, 3] &= (1\ 4)(2\ 3), (5\ 8)(6\ 7), (9\ 12)(10\ 11) \\
 \lambda[1, 2, 3] &= (1\ 3)(2\ 4), (5\ 7)(6\ 8), (9\ 11)(10\ 12) \\
 \rho &= (1\ 10\ 6\ 9\ 3\ 5)(2\ 12\ 7\ 11\ 4\ 8) \\
 \phi &= (1\ 9\ 4\ 6\ 10\ 8)(2\ 11\ 3\ 7\ 12\ 5)
 \end{aligned}$$

We trace the values of the procedure variables  $j$  (node number),  $Gray(j)$  (its Gray code),  $A_j$  (the corresponding ribbon subset),  $\epsilon(G^{A_j})$  (the Euler-genus for the corresponding partial dual), and the vector  $coeff$  (whose coordinates record the number of times that each Euler-genus occurs).

$j$	$Gray(j)$	$A_j$	$\epsilon(G^{A_j})$	$coeff[0, 1, 2, 3]$
0	0	$\emptyset$	3	[0, 0, 0, 1]
1	1	{1}	1	[0, 1, 0, 1]
2	3	{1, 2}	3	[0, 1, 0, 2]
3	2	{2}	2	[0, 1, 1, 2]
4	6	{2, 3}	1	[0, 2, 1, 2]
5	7	{1, 2, 3}	3	[0, 2, 1, 3]
6	5	{1, 3}	2	[0, 2, 2, 3]
7	4	{3}	3	[0, 2, 2, 4]

Accordingly, the graph of Fig. 4 has the Poincaré partial-duality polynomial

$$2z + 2z^2 + 4z^3.$$

Thus, like the polynomial (4.3) of [11], it is unimodal but not log-concave.

## References

1. Abrams, L., Ellis-Monaghan, J.: New dualities from old: generating geometric, Petrie, and Wilson dualities and trialities of ribbon graphs. [arXiv:1901.03739v2](https://arxiv.org/abs/1901.03739v2) [math.CO], 9 Aug (2019)
2. Biggs, N.L., Lloyd, E.K., Wilson, R.J.: Graph Theory 1736–1936. Oxford University Press, Oxford (1976)
3. Bollobás, B., Riordan, O.: A polynomial invariant of graphs on orientable surfaces. Proc. London Math. Soc. **83**, 513–531 (2001)
4. Bollobás, B., Riordan, O.: A polynomial of graphs on surfaces. Math. Ann. **323**, 81–96 (2002)
5. Chmutov, S.: Generalized duality for graphs on surfaces and the signed Bollobás-Riordan polynomial. J. Combin. Theory Ser. B **99**, 617–638 (2009)



6. Edmonds, J.R.: A combinatorial representation for polyhedral surfaces (abstract). *Notices Amer. Math. Soc.* **7**, 646 (1960)
7. Ehrlich, G.: Loopless algorithms for generating permutations, combinations, and other combinatorial configurations. *J. ACM* **20**, 500–513 (1973)
8. Ellis-Monaghan, J., Moffatt, I.: Twisted duality for embedded graphs. *Trans. Amer. Math. Soc.* **364**, 1529–1569 (2012)
9. Ellis-Monaghan, J., Moffatt, I.: *Graphs on Surfaces: Dualities, Polynomials, and Knots*. Springer, Heidelberg (2013)
10. Gross, J.L., Mansour, T., Tucker, T.W.: Partial-duality for ribbon graphs, I: Distributions, *Europ. J. Combin.* **86** (2020), article 103084
11. Gross, J.L., Mansour, T., Tucker, T.W.: Partial-duality for ribbon graphs, II: Permutations and Monodromy, *Europ. J. Combin.*, to appear (2021)
12. Gross, J.L., Robbins, D.P., Tucker, T.W.: Genus distributions for bouquets of circles. *J. Combin. Theory Ser. B* **47**, 292–306 (1989)
13. Gross, J.L., Tucker, T.W.: *Topological Graph Theory*. Wiley, 1987 (reprinted by Dover, 2001)
14. Heffter, L.: Über das problem der nachbargebiete. *Math. Ann.* **38**, 477–508 (1891)
15. Knuth, D.E.: *The Art of Computer Programming*, Vol. 4: Generating all tuples and permutations addison-Wesley, (2005)
16. Jones, G.A., Singerman, D.: Theory of maps on orientable surfaces. *Proc. London Math. Soc.* **III**(37), 273–307 (1978)
17. Moffatt, I.: Partial duality and Bollobás and Riordan’s ribbon graph polynomial. *Discrete Math.* **310**, 174–183 (2010)
18. Moffatt, I.: Separability and the genus of a partial dual. *Europ. J. Combin.* **34**, 355–378 (2013)
19. Reshetikhin, N., Turaev, V.: Ribbon graphs and their invariants derived from quantum groups. *Comm. Math. Phys.* **127**, 1–26 (1990)
20. Tutte, W.T.: What is a map?, in *New Directions in the Theory of Graphs*, ed. Academic Press, F. Harary (1973)
21. Tutte, W.T.: *Graph Theory*, volume 21 of *Encyclopedia of Mathematics and its Applications*, Addison-Wesley, (1984)
22. White, A.T.: *Graphs of Groups on Surfaces*. Elsevier, North-Holland (2001)
23. Wilson, S.: Operators over regular maps. *Pacific J. Math.* **81**, 559–568 (1979)
24. Youngs, J.W.T.: Minimal imbeddings and the genus of a graph. *J. Math. and Mech.* **12**, 303–315 (1963)

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.