



HAL
open science

The distance-dependent two-point function of triangulations: a new derivation from old results

Emmanuel Guitter

► **To cite this version:**

Emmanuel Guitter. The distance-dependent two-point function of triangulations: a new derivation from old results. *Annales de l'Institut Henri Poincaré (D) Combinatorics, Physics and their Interactions*, 2017, 4, pp.177-211. 10.4171/AIHPD/38 . hal-01236866

HAL Id: hal-01236866

<https://hal.science/hal-01236866>

Submitted on 25 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THE DISTANCE-DEPENDENT TWO-POINT FUNCTION OF TRIANGULATIONS: A NEW DERIVATION FROM OLD RESULTS

EMMANUEL GUITTER

ABSTRACT. We present a new derivation of the distance-dependent two-point function of random planar triangulations. As it is well-known, this function is intimately related to the generating functions of so-called slices, which are pieces of triangulation having boundaries made of shortest paths of prescribed length. We show that the slice generating functions are fully determined by a direct recursive relation on their boundary length. Remarkably, the kernel of this recursion is some quantity introduced and computed by Tutte a long time ago in the context of a global enumeration of planar triangulations. We may thus rely on these old results to solve our new recursion relation explicitly in a constructive way.

1. INTRODUCTION

The combinatorics of planar maps, i.e. connected graphs embedded on the sphere, is making constant progress since the seminal work of Tutte in the 60'th. In the more recent years, a growing interest was shown for metric properties of maps endowed with their graph distance, and especially for the corresponding distance statistics within ensembles of random maps. An emblematic result was the computation, for several families of maps, of the *distance-dependent two-point function* which, so to say, measures the profile of distances between two points (vertices or edges) picked at random on the map. Explicit expressions for this two-point function were obtained for ensembles of planar maps with controlled face degrees [2, 4] as well as for maps (or hyper-maps) with arbitrary face degrees but with controlled edge (or hyper-edge) and face numbers [1, 3]. A first way to solve these questions was the use of bijections between maps and decorated trees, as first discovered by Schaeffer [9] (upon reformulating a bijection by Cori and Vauquelin [5]). In a second, intimately related, approach, the problem of computing the distance-dependent two-point function was reduced to that of enumerating *slices*, which are particular pieces of maps bordered by shortest paths of prescribed length, meeting at some "apex". In a first stage, the computation of either decorated trees or slice generating functions relied on finding the solutions of particular *integrable systems of equations* satisfied by the generating functions at hand. No general technique however was developed to solve these equations and all the explicit expressions obtained in this way were the result of a *simple guessing* of the solution. The recourse to decorated trees or slices took on its full dimension when it was later discovered that their generating functions could be obtained mechanically as coefficients in suitable continued fraction expansions for standard map generating functions. This property was exploited in [4] to obtain a constructive derivation of the distance dependent two-point function for maps with controlled face degrees.

In this paper, we revisit the problem of computing the *distance-dependent two-point function of random planar triangulations*, i.e planar maps whose all faces have degree 3. These maps were extensively studied in the past as they form one of the simplest natural families of maps. Their two-point function was first obtained in [6] by guessing the solution of the associated integrable system. It was then re-obtained in [4] as a particular example of the general continued fraction formalism. Here, we present a *new recursive approach* which consists in directly relating the generating function of slices whose border has (maximal) length k to that of slices whose border has (maximal) length $k - 1$ (see eq. (3) below). Remarkably

enough, the “kernel” of our recursion relation is some particular generating function of triangulations, already introduced by Tutte as early as in his first paper [10] on triangulations. We may thus directly use the old results of [10] to solve our new recursion relation in a constructive way, without recourse to any guessing.

The paper is organized as follows: in Section 2, we recall the definition of slices and their connection with the distance-dependent two-point function of random planar triangulations. We also recall the standard integrable system obeyed by the slice generating functions, whose solution was guessed in [6]. Section 3 is devoted to the derivation of our new recursion relation between the generating function T_k for slices with (maximal) border length k and T_{k-1} . This new recursion is based on the existence of some particular *dividing line* which, so to say, delimits in the slice a region whose vertices are at distance strictly larger than $k - 1$ from the apex of the slice. As just mentioned, the kernel of our recursion is some particular generating function computed by Tutte in its seminal paper [10] on triangulations. In order to stick to Tutte’s original results, we make in Section 4 a detour to the family of *simple triangulations*, i.e. triangulations with neither loops nor multiple edges. As shown, a simple substitution procedure makes the correspondence between this simplified family and the family of all triangulations that we are interested in. We then use in Section 5 the explicit form given by Tutte for the kernel of our recursion relation to rewrite this recursion in a particularly simple and classical form (see eq. (20) below), whose solution is easily obtained by classical techniques. We finally return in Section 6 to the case of general (not necessarily simple) triangulations by performing the required substitution. This leads us to our final explicit expressions for T_k and for the distance-dependent two-point function. We gather our concluding remarks in Section 7.

2. SLICE GENERATING FUNCTIONS: REMINDERS

2.1. Definitions. The distance-dependent two-point function of planar triangulations may be expressed in terms of the generating functions R_k and S_k for *R-slices* and *S-slices* of maximal size k (see eq. (1) below). Slices are particular families of *triangulations with a boundary*, namely planar rooted (i.e. with a marked oriented edge, the root-edge) maps whose all faces have degree 3, except for the outer face (i.e. the face lying on the right of the root-edge) which may have arbitrary degree. The inner faces form what it called the *bulk* while the edges incident to the outer face (visited, say clockwise around the bulk) form *the boundary* whose *length* is the degree of the outer face. *R-* and *S-*slices are defined as follows:

- *R*-slices have a boundary of length 2ℓ ($\ell \geq 1$) and satisfy (see figure 1):
 - the (graph) distance from the origin of the root-edge (the root-vertex) to the *apex*, which is the vertex reached from the root-vertex by making ℓ elementary steps *along the boundary* clockwise around the bulk, is ℓ . In other words, the left boundary of the slice, which is the part of its boundary lying between the root-vertex and the apex clockwise around the bulk is a shortest path between its endpoints within the map;
 - the distance from the endpoint of the root-edge to the apex is $\ell - 1$. In other words, the right boundary of the slice, which is the part of the boundary lying between the endpoint of the root-edge and the apex counterclockwise around the bulk is a shortest path between its endpoints within the map;
 - the right boundary is the *unique* shortest path between its endpoints within the map;
 - the left and right boundaries do not meet before reaching the apex.

We call $R_k \equiv R_k(g)$ ($k \geq 1$) the generating function of *R*-slices with $1 \leq \ell \leq k$, enumerated with a weight g per inner face.

- *S*-slices have a boundary of length $2\ell + 1$ ($\ell \geq 0$) and satisfy (see figure 1):

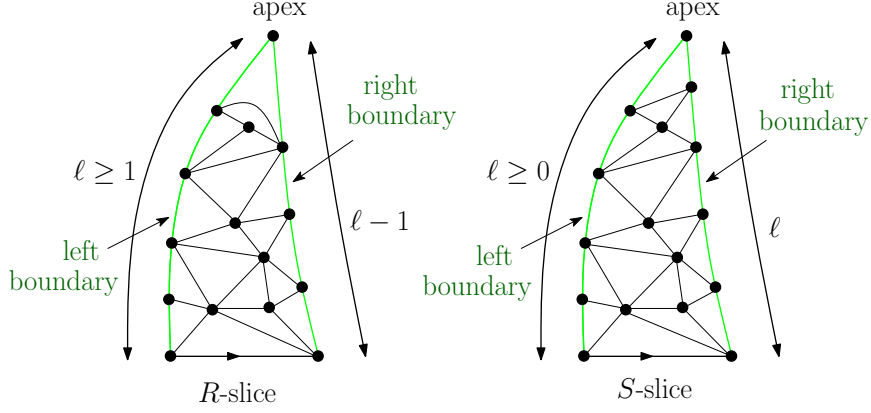


FIGURE 1. An example of R -slice (left) and S -slice (right). In both cases, the left boundary is a shortest path within the map between the root-vertex and the apex and has length ℓ . As for the right boundary, it is in both cases the unique shortest path within the map between the endpoint of the root-edge and the apex, with length $\ell - 1$ for the R -slice ($\ell \geq 1$), and ℓ for the S -slice ($\ell \geq 0$).

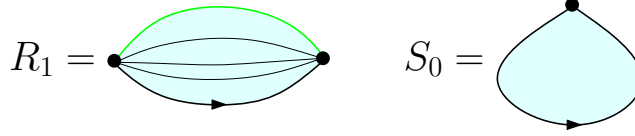


FIGURE 2. A schematic picture of maps enumerated by R_1 and S_0 .

- the distance from the root-vertex to the apex, which is the vertex reached from the root-vertex by making ℓ elementary steps *along the boundary* clockwise around the bulk, is ℓ . In other words, the left boundary of the slice (which is the part of the boundary lying between the root-vertex and the apex clockwise around the bulk) is a shortest path between its endpoints within the map;
- the distance from the endpoint of the root-edge to the apex is ℓ . In other words, the right boundary of the slice (which is the part of the boundary lying between the endpoint of the root-edge and the apex counterclockwise around the bulk) is a shortest path between its endpoints within the map;
- the right boundary is the unique shortest path between its endpoints within the map;
- the left and right boundaries do not meet before reaching the apex.

We call $S_k \equiv S_k(g)$ ($k \geq 0$) the generating function of S -slices with $0 \leq \ell \leq k$, enumerated with a weight g per inner face.

Note that the map reduced to a single root-edge and an outer face of degree 2 is an R -slice with $\ell = 1$ and contributes a term 1 to R_k for any $k \geq 1$.

Of particular interest are the generating functions R_1 and S_0 , with the following interpretations: by definition, R_1 enumerates R -slices with $\ell = 1$, therefore with a boundary of length 2. The right boundary has length 0, hence the apex is the endpoint of the root edge. The left boundary, of length 1, connects the extremities of the root-edge, which are

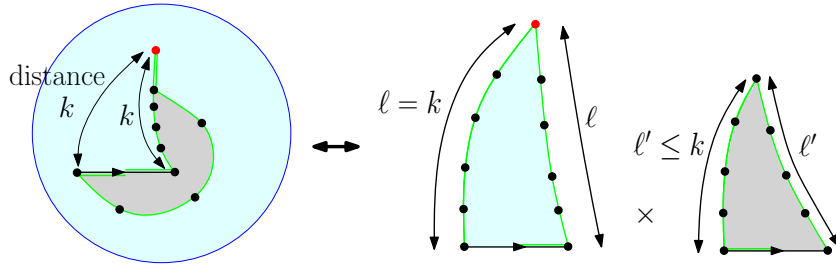


FIGURE 3. A schematic picture of the one-to-one correspondence between pointed rooted triangulations whose root-edge has its both extremities at distance k from the marked vertex (in red) and a pair of S -slices with left-boundary lengths ℓ and ℓ' satisfying $\max(\ell, \ell') = k$. The green lines on the left side are the leftmost shortest paths to the marked vertex from the middle of the root-edge in both directions. Cutting along these paths creates the two slices on the right. The choice of leftmost shortest paths ensures that the right boundaries of the slices are the unique shortest paths between their endpoints within the slice. Note that we reversed the original orientation of the root-edge to obtain the root-edge of the grey slice.

necessarily distinct. The function R_1 may therefore be interpreted as the *generating function of rooted triangulations with a boundary of length 2 connecting two distinct vertices* (the extremities of the root-edge). Note that in such maps, the edges connecting the extremities of the root edge within the map form in general what we shall call a *bundle* of p edges for some $p \geq 1$ (see figure 2). The function R_1 therefore enumerates bundles of edges. As for S_0 , it enumerates S -slices with $\ell = 0$, in which case both extremities of the root-edge coincide with the apex. In particular, the root-edge forms a loop. The function S_0 may thus be interpreted as the *generating function of rooted triangulations with a boundary of length 1* (see figure 2).

For later use, we also introduce the generating function

$$T_k \equiv S_k - S_0$$

for $k \geq 0$. This generating function enumerates S -slices with a left-boundary length ℓ satisfying $1 \leq \ell \leq k$. These are precisely the S -slices contributing to S_k and *whose root-edge does not form a loop* (recall indeed that the left and right boundary of an S -slice are required to meet only at the apex, so that the root-edge forms a loop if and only if $\ell = 0$). Note that $T_0 = 0$ by definition.

2.2. The distance dependent two-point function. For $k \geq 0$, we define the distance-dependent two-point function of planar triangulations as the generating function G_k for pointed (i.e. with a marked vertex) rooted (i.e. with a marked oriented edge) planar triangulations (i.e planar maps whose all faces have degree 3) for which the marked vertex is at graph distance k from the root-vertex (i.e. the origin of the root-edge). Let us show the following identity for $k \geq 1$:

$$(1) \quad G_k = (S_k^2 - S_{k-1}^2) + (R_k - R_{k-1} - \delta_{k,1}) + (R_{k+1} - R_k) = S_k^2 - S_{k-1}^2 + R_{k+1} - R_{k-1} - \delta_{k,1}$$

with the convention $R_0 = 0$. Maps enumerated by G_k may indeed be classified into three classes according to the distance from their marked vertex to the endpoint of the root-edge. This distance may be k , $k - 1$ or $k + 1$ and the three terms in the middle expression in (1) above correspond to the enumeration of the three classes. If the two extremities of the root-edge are at distance k from the marked vertex, we draw the *leftmost* shortest paths to the marked vertex, starting from the middle of the root-edge in both directions (see figure 3).

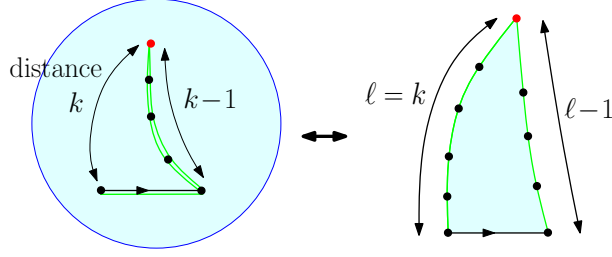


FIGURE 4. A schematic picture of the one-to-one correspondence between a pointed rooted triangulation whose root-edge has its origin and endpoint at respective distance k and $k - 1$ from the marked vertex (in red) and an R -slices with left-boundary lengths $\ell = k$. The green line on the left side is the leftmost shortest paths to the marked vertex from the the root-vertex (with the root-edge as first step). Cutting along this path creates the desired R -slice on the right.

Cutting along these paths results into two S -slices, whose root-edge is the original root-edge with its original orientation for one slice and with the reversed orientation for the other. Note that the choice of leftmost shortest paths ensures that the right boundary of each piece is the unique shortest path between its endpoints within the piece. As for the left boundaries, they are also shortest paths between their endpoints, with lengths less than or equal to k (corresponding to situations enumerated by S_k^2) and with at least one of these lengths equal to k , hence are enumerated by $S_k^2 - S_{k-1}^2$ (since the situations where both lengths are strictly less than ℓ are enumerated by S_{k-1}^2). If the endpoint of the root-edge is at distance $k - 1$ from the marked vertex, we draw the leftmost shortest path form the root-vertex to the marked vertex, taking the root-edge as first step (see figure 4). Cutting along this path results into an R -slice whose root-edge is the original root-edge, with left boundary length equal to k , and moreover different from the single root-edge if $k = 1$. Such slices are enumerated by $(R_k - 1) - (R_{k-1} - 1) = R_k - R_{k-1}$ for $k \geq 2$ and by $R_1 - 1$ for $k = 1$, i.e. by $R_k - R_{k-1} - \delta_{k,1}$ with our convention that $R_0 = 0$. Finally, if the endpoint of the root-edge is at distance $k + 1$ from the marked vertex, we reverse the orientation of the root-edge to get back to the previous situation with $k \rightarrow k + 1$. Such maps are thus enumerated by $R_{k+1} - R_k$, hence the relation (1). As already mentioned, the knowledge of both R_k and S_k leads immediately via (1) to a expression for the distance-dependent two-point function G_k . As a final remark, for $k = 0$, G_0 enumerates rooted triangulations and adapting the above argument immediately yields

$$G_0 = S_0^2 + R_1 - 1 .$$

2.3. Classical relations for slice generating functions. A first set of relations for the slice generating functions may be obtained by classifying the slices according to the nature of the inner face lying immediately on the left of the root-edge. In the case of an R -slice not reduced to a single root-edge, this triangular face is incident to the two extremities of the root-edge, at respective distances ℓ and $\ell - 1$ from the apex, and to an intermediate vertex at distance $\ell - 1$ or ℓ (note that this vertex may possibly be identical to one of the two others). Drawing the leftmost shortest path from this intermediate vertex to the apex separates the map into two slices: an R -slice and an S -slice (see figure 5). As for S -slices, the triangular face on the left of the root-edge has its intermediate vertex at distance ℓ or $\ell \pm 1$ from the apex. In the latter case, removing the triangle directly results into an R -slice while, in the former case, drawing the leftmost shortest path from the intermediate vertex to the apex separates the map into two S -slices (see figure 6). Taking into account the boundary length

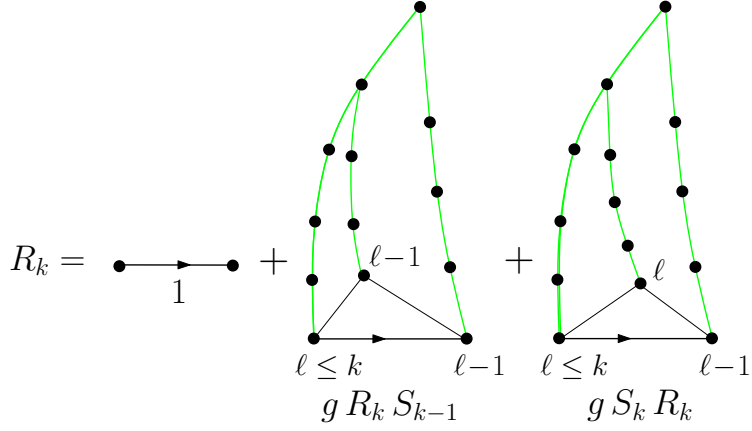


FIGURE 5. A schematic picture explaining the first line of eq. (2). If not reduced to a single edge, the R -slice with $1 \leq \ell \leq k$ is decomposed by removing the triangle immediately on the left of the root-edge, whose intermediate vertex is at distance $\ell - 1$ or ℓ from the apex, and by cutting along the leftmost shortest path from this vertex to the apex.

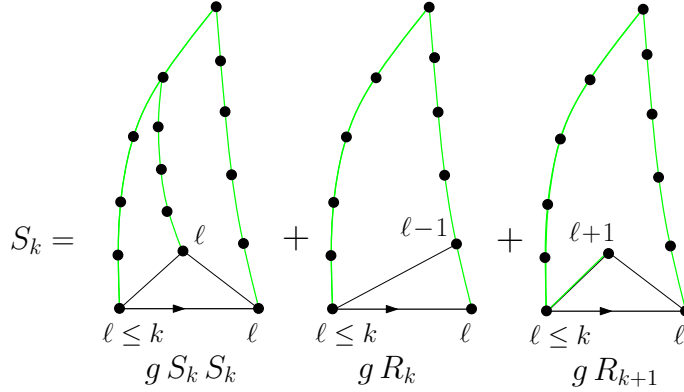


FIGURE 6. A schematic picture explaining the second line of eq. (2). The S -slice with $0 \leq \ell \leq k$ is decomposed by removing the triangle immediately on the left of the root-edge, whose intermediate vertex is at distance ℓ or $\ell - 1$ (in which case it lies on the right boundary) or $\ell + 1$ from the apex, and, in the first case, by cutting along the leftmost shortest path from this vertex to the apex.

constraints, we immediately arrive at the classical system

$$(2) \quad \begin{cases} R_k = 1 + g R_k (S_{k-1} + S_k), & k \geq 1 \\ S_k = g (S_k^2 + R_k + R_{k+1}), & k \geq 0 \end{cases}$$

with again our convention $R_0 = 0$. Note that this system is not *stricto sensu* recursive since we don't know the value of S_0 . Still it is recursive order by order in g if we impose $R_k = 1 + O(g^2)$ for all $k \geq 1$ and $S_k = O(g)$ for all $k \geq 0$, as required by the definition of R_k and S_k as slice generating functions. The solution of this system was first found in [6] by simple guessing, leading to explicit expressions for R_k and S_k , hence for the distance

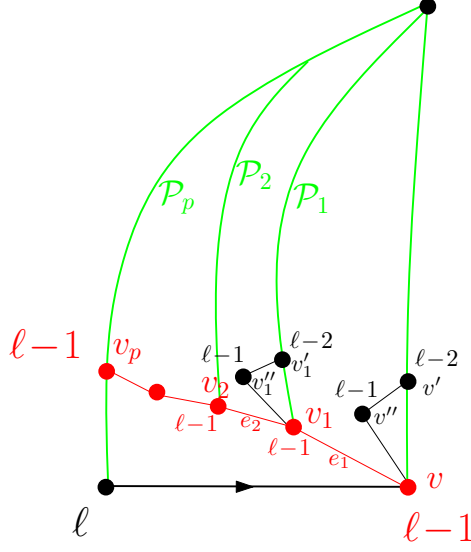


FIGURE 7. Construction of the dividing line (in red) in an R -slice (see text).

dependent two-point function G_k via eq. (1). Later on, these explicit expressions were re-derived in a constructive way *without recourse to the system* (2) or to any other recursion relation, but by instead relating R_k and S_k to the distance-independent generating functions of triangulations with a boundary of fixed length [4].

In the next section, we shall introduce a new set of recursion relations for R_k and S_k , which we shall then solve explicitly in a constructive way.

3. A NEW APPROACH BY RECURSION

3.1. Construction of a dividing line. We shall now derive a new set of recursion relations for R_k and S_k (or more precisely T_k) based on a new decomposition of the slices. This decomposition makes use of a particular *dividing line* drawn on the slice, which we define now. We start for convenience with an R -slice whose left boundary has some length $\ell \geq 2$. Its dividing line will then be made of a sequence of edges linking the right and left boundaries of the slice and connecting only vertices at distance $\ell - 1$ from the apex. It is defined recursively as follows (see figure 7): consider the face on the left of the first edge of the right boundary, i.e. the edge linking the endpoint v of the root-edge, at distance $\ell - 1$ from the apex to its neighbor v' along the right boundary, at distance $\ell - 2$ from the apex. The third vertex incident to this face, v'' , is necessarily different from v and v' as otherwise, we would have a second edge linking v and v' within the map, hence a second shortest path from v to the apex, lying strictly on the left of the right boundary. Moreover, v'' is necessarily at distance $\ell - 1$ from the apex: indeed the only allowed values for the distance are $\ell - 2$ and $\ell - 1$ but the value $\ell - 2$ is forbidden as again it would lead to the existence of another shortest path from v to the apex, strictly on the left of the right boundary. We conclude that v is incident to at least one edge *leading to a distinct neighbor at distance $\ell - 1$ from the apex*. Let us pick the leftmost such edge e_1 and call v_1 its endpoint. Assuming that v_1 does not belong to the left boundary, we draw the *leftmost* shortest path \mathcal{P}_1 from v_1 to the apex and call v'_1 the vertex on \mathcal{P}_1 at distance $\ell - 2$ from the apex. Consider again the face on the left of the first edge of \mathcal{P}_1 , linking v_1 to v'_1 . It is incident to a third vertex v''_1 , necessarily different from v_1 and v'_1 and at distance $\ell - 1$ from the apex (for the same reasons as above). We thus conclude that v_1 is incident to at least one edge leading to a distinct neighbor at

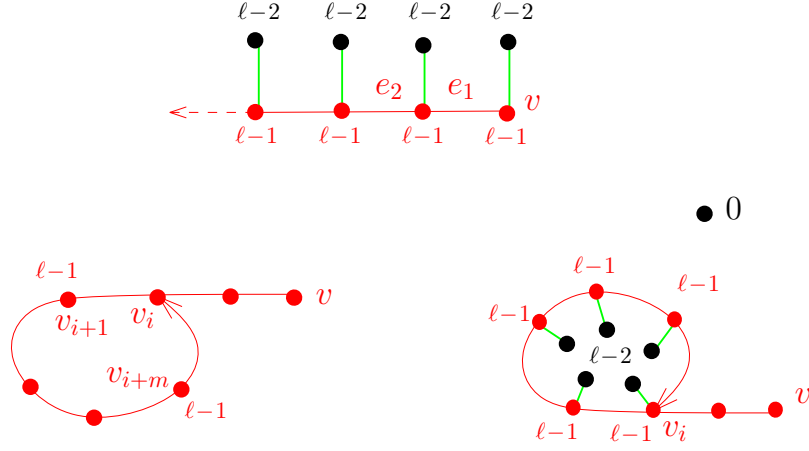


FIGURE 8. Top: a schematic picture of the dividing line. Bottom: if this line forms a loop, we end up with a contradiction at the first revisited vertex v_i : if the loop closes from the left (bottom left), it means that, at step i , we should have picked the edge leading to v_{i+m} , not to v_{i+1} ; if the loop closes from the right (bottom right), it encloses vertices at distance $\ell - 2$ from the apex (marked 0), which is inconsistent with the fact that all the vertices on the dividing line are at distance $\ell - 1$ from the apex.

distance $\ell - 1$ from the apex. As before, we pick the leftmost such edge e_2 and call v_2 its endpoint. We may repeat the procedure as long as we do not reach the left boundary, thus creating an oriented line (e_1, e_2, \dots) linking only vertices at distance $\ell - 1$ from the apex with moreover, on the right of each vertex along the line, an edge linking this vertex to a vertex at distance $\ell - 2$ from the apex (see figure 8-top). It is easy to see that this line cannot make a loop. Indeed, let us assume that the line revisits some already visited vertex and pick the first such vertex. If this vertex is reached from the left, this contradicts the fact that, in our construction, we always picked the leftmost edge to a neighbor at distance $\ell - 1$ from the apex (see figure 8-bottom left). If it is reached from the right, this creates a closed region surrounded by vertices at distance $\ell - 1$ from the apex, which does not contain the apex and which contains vertices at distance $\ell - 2$ from the apex, a contradiction (see figure 8-bottom right). The line thus necessarily ends after a finite number p of steps at the vertex v_p lying on the left boundary at distance $\ell - 1$ from the apex (note that \mathcal{P}_p is then the part of the left boundary lying between v_p and the apex). The open line (e_1, e_2, \dots, e_p) from v to v_p constitutes our dividing line with the following property: it is a simple curve linking the right and left boundaries, visiting only vertices at distance $\ell - 1$ from the apex, dividing *de facto* the map in two parts, an upper part containing the apex and a lower part containing the root-vertex. Finally, by construction, we have the following property:

Property 1. *Two vertices of the dividing line cannot be linked by an edge lying strictly inside the lower part.*

Indeed, violating Property 1 would contradict the fact that, in our construction, we always picked the leftmost edge leading to a neighbor at distance $\ell - 1$. We could similarly have started with an S -slice whose left boundary has some length $\ell \geq 2$. The dividing line would then be defined exactly in the same way, now starting from the vertex v of the right boundary at distance $\ell - 1$ from the apex (see figure 10 for an illustration).

In the following, we shall recourse to the dividing line to decompose R -slices enumerated by R_k and S -slices enumerated by T_k . Both families of slices have a left-boundary length ℓ

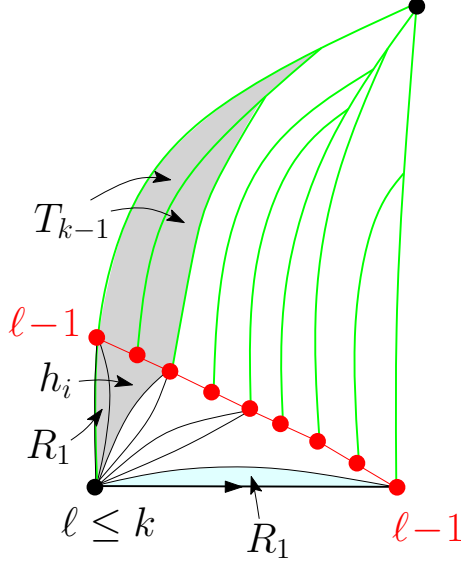


FIGURE 9. The decomposition of an R -slice enumerated by R_k into a sequence of blocks. The first block is indicated in gray. Each block is formed of a bundle (enumerated by R_1), a rooted triangulation with a boundary of some arbitrary length $i \geq 3$ and with particular properties (see text), enumerated by h_i , and a set of $i - 2$ attached S -slices enumerated by T_{k-1} . The generating function of a block is thus $R_1 \sum_{i \geq 3} h_i T_{k-1}^{i-2}$. The sequence of blocks is to be completed by a final bundle (enumerated by R_1 , here in light blue) connecting the extremities of the root-edge.

satisfying $1 \leq \ell \leq k$. So far we defined the dividing line only for $\ell \geq 2$. For $\ell = 1$, we take the convention that the dividing line is reduced to a single vertex equal to the apex.

3.2. A new set of recursion relations. We shall now derive a new set of recursion relations for R_k and T_k , based on a new decomposition of slices intimately linked to their dividing line. We start again for convenience with an R -slice and draw its dividing line, as defined above. We then note that the root-vertex, at distance ℓ from the apex, is adjacent, in all generality, to a number of vertices of the dividing line. These include the two extremities of the line, plus possibly some of its internal vertices. In general, such adjacency with a given vertex of the dividing line is moreover achieved by a bundle of edges, as we defined it, which we view as a rooted triangulation with a boundary of length two (the boundary being made of the two outermost edges performing the connection), hence which is enumerated by R_1 . For each vertex of the dividing line adjacent to the root-vertex, we cut the map along the *leftmost* edge of the associated bundle and along the leftmost shortest path from this vertex to the apex. This cutting decomposes the slice into a sequence of blocks (see figure 9), each of which being formed of (1) a bundle enumerated by R_1 , (2) a triangulation with a boundary of some arbitrary length $i \geq 3$, lying in the lower part of the slice in-between two successive bundles, and enumerated by a generating function $h_i \equiv h_i(g)$ that we shall analyze just below, and (3) a set of $i - 2$ S -slices *whose root-edge does not form a loop*. If we start with an R -slice enumerated by R_k , hence with $1 \leq \ell \leq k$, then the $\ell = 1$ contribution yields the empty sequence of blocks (since the dividing line is reduced to the apex in this case), while the $\ell \geq 2$ contribution yields non-empty sequences of blocks whose S -slice components have some arbitrary left-boundary length between 1 and $\ell - 1$, hence between 1 and $k - 1$, as

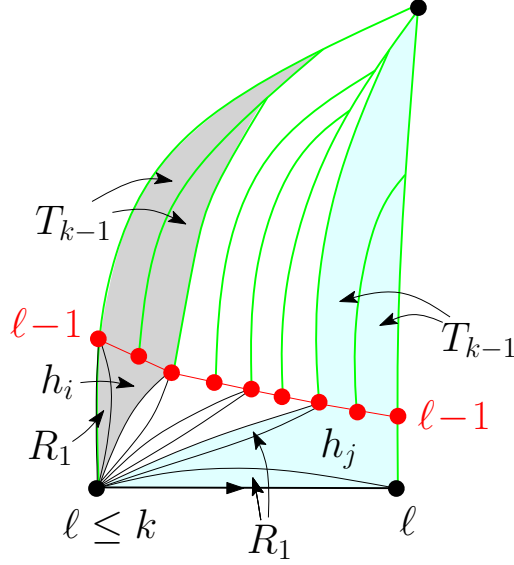


FIGURE 10. The decomposition of an S -slice enumerated by T_k into blocks similar to those of figure 9, each enumerated by $R_1 \sum_{i \geq 3} h_i T_{k-1}^{i-2}$. The sequence of blocks is now to be completed by a final portion (in light blue) formed of two bundles (each enumerated by R_1), a rooted triangulation with a boundary of some arbitrary length $j \geq 3$, enumerated by h_j , and a set of $j-3$ S -slices enumerated by T_{k-1} . The generating function of the last portion is thus $R_1^2 \sum_{j \geq 3} h_j T_{k-1}^{j-3}$.

enumerated by T_{k-1} (since their root-edge cannot form a loop). To summarize, each block of the (possibly empty) sequence is enumerated by $R_1 \sum_{i \geq 3} h_i T_{k-1}^{i-2}$. Finally, we are left with a final bundle connecting the two extremities of the root-edge and enumerated by R_1 (see figure 9). If we now start instead with an S -slice enumerated by T_k , hence satisfying $1 \leq \ell \leq k$, a similar decomposition produces a (possibly empty) sequence of the same blocks, now completed by a final portion of map enumerated by $R_1^2 \sum_{i \geq 3} h_i T_{k-1}^{i-3}$ (see figure 10). We arrive at the relations

$$(3) \quad \begin{cases} R_k = \frac{R_1}{1 - R_1 \sum_{i \geq 3} h_i T_{k-1}^{i-2}} = \frac{R_1}{1 - R_1 T_{k-1} \Phi(T_{k-1})} \\ T_k = \frac{R_1^2 \sum_{i \geq 3} h_i T_{k-1}^{i-3}}{1 - R_1 \sum_{i \geq 3} h_i T_{k-1}^{i-2}} = \frac{R_1^2 \Phi(T_{k-1})}{1 - R_1 T_{k-1} \Phi(T_{k-1})} \end{cases}$$

with

$$(4) \quad \Phi(T) \equiv \Phi(T, g) = \sum_{i \geq 3} h_i(g) T^{i-3} .$$

Note in particular the relation

$$(5) \quad R_k - R_1 = T_{k-1} T_k .$$

So far we did not discuss the precise definition of $h_i \equiv h_i(g)$ ($i \geq 3$). By construction, the maps enumerated by h_i correspond to a part of the slice lying below the dividing line and in-between two consecutive bundles. This part forms a rooted triangulation with a boundary

of length i made of a segment formed by $i - 2$ consecutive edges of the dividing line and of 2 extra edges connecting the extremities of this segment to the root-vertex (we may then decide for instance to root the map at its rightmost edge from the root vertex to the dividing line, oriented away from the root-vertex). This boundary forms by construction a simple curve. Moreover, we have the property:

Property 2. *In the maps enumerated by h_i , two vertices of the boundary cannot be linked by an edge lying strictly inside the map.*

For a pair of boundary vertices belonging to the dividing line, this property follows immediately from Property 1 above. As for the root vertex, the only boundary vertices to which it is connected are the two extremities of the segment of the dividing line and each of this connection is performed by a single boundary edge.

To summarize, $h_i \equiv h_i(g)$ is the generating function of rooted triangulations with a boundary of length i ($i \geq 3$) forming a simple curve, and *with the property 2* above. It is interesting to note that, even if, in the construction, the root-vertex and the vertices of the dividing line play very different roles, all boundary vertices eventually play *symmetric roles* in the maps enumerated by h_i .

Assuming that we know $\Phi(T)$, the second line of (3) is a direct recursion on k which fixes T_k for all $k \geq 0$ recursively from the initial condition $T_0 = 0$. Then the first line of (3) gives access to R_k for all $k \geq 1$. Strictly speaking, we also need as input the knowledge of R_1 and, if we eventually want an expression for $S_k = T_k + S_0$ and for the two-point function G_k , we also need the value of S_0 . We will explain in Section 6 how to get rid of this problem.

3.3. Back to Tutte's seminal paper. The natural question at this stage is: do we have an expression for $\Phi(T)$? Remarkably, the answer is yes, as shown in Tutte's seminal paper [10] on the enumeration of triangulations. There Tutte introduces precisely the same notion of triangulations having a boundary forming a simple curve of arbitrary length at least 3 and *satisfying Property 2* above. To be precise, Tutte considers what are called *simple triangulations*, i.e. triangulations required to have no loops nor multiple edges. The quantity considered by Tutte is therefore a slight reformulation of $\Phi(T)$, called $\psi(x, y)$ in [10], but the passage from $\psi(x, y)$ to $\Phi(T)$ is straightforward and can be obtained via a simple substitution procedure. This correspondence will be made explicit in Section 4 below. With this correspondence, Tutte's result immediately translates into the following equation

$$(6) \quad R_1 T^2 \Phi^2(T) + (g R_1^2 + g R_1^3 h_3 T - T - g R_1 T^2) \Phi(T) + (g T - g R_1^2 h_3) = 0$$

which, as shown in [10], *entirely fixes* $\Phi(T)$ as a function of T , g and R_1 (see Section 5 below for an explicit expression). We thus have at our disposal all the ingredients to solve our new recursion relation, a task that will be performed explicitly in the next sections.

Before we solve our recursion relation, it is interesting to explore what we may learn by simply *making the system (3) consistent with the more classical system (2)*. The first line in (3) may be rewritten as

$$T_{k-1} \Phi(T_{k-1}) = \frac{1}{R_1} - \frac{1}{R_k}$$

while the first line of (2) leads to

$$\frac{1}{R_k} = 1 - g(S_k + S_{k-1}) = 1 - 2g S_0 - g(T_k + T_{k-1})$$

and in particular

$$\frac{1}{R_1} = 1 - 2g S_0 - g T_1 = 1 - 2g S_0 - g R_1^2 h_3$$

since $T_0 = 0$ and $T_1 = h_3 R_1^2$. This later relation is easily understood by noting that T_1 enumerates S -slices with $\ell = 1$, which are rooted triangulations with a boundary of length

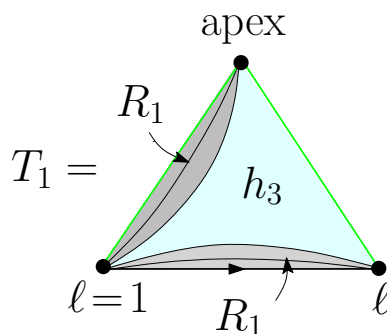


FIGURE 11. A schematic explanation of the relation $T_1 = R_1^2 h_3$.

3 forming a simple curve (see figure 11). On the other hand, h_3 also enumerates rooted triangulations with a boundary of length 3 and the only difference is that, in T_1 , both the root edge and the left boundary edge may be doubled by a bundle of edges (the right boundary edge cannot be doubled by a bundle as it is the unique shortest path to the apex). In other words, to get T_1 , we must multiply h_3 twice by the bundle generating function R_1 , hence the relation.

Combining the above equations, we deduce

$$T_{k-1}\Phi(T_{k-1}) = g(T_k + T_{k-1}) - gR_1^2 h_3 = g\left(\frac{R_1^2 \Phi(T_{k-1})}{1 - R_1 T_{k-1} \Phi(T_{k-1})} + T_{k-1}\right) - gR_1^2 h_3$$

where we have used the second line of (3) to express T_k in terms of T_{k-1} . Equating the left and right terms above the leads precisely to equation (6) for the specific value $T = T_{k-1}$. This equation being valid for any positive integer k , we may reasonably infer that it holds for any T (small enough so that $\Phi(T)$ is well-defined). Indeed the explicit dependence of T_{k-1} in k (see below) allows to formally extend T_{k-1} to non-integer values of k , so that T_{k-1} now varies continuously with real k . To summarize, making the systems (3) and (2) consistent is yet another way to understand Tutte's equation (6).

4. A DETOUR VIA SIMPLE TRIANGULATIONS

4.1. Substitution. As already mentioned, the analysis of [10] deals with simple triangulations, i.e. triangulations *with neither loops nor multiple edges* (here a loop stands of an edges with identical extremities). Let us therefore introduce the generating functions r_k , t_k ($k \geq 1$) and \tilde{h}_i ($i \geq 3$), defined as R_k , T_k and h_i respectively, but with the constraint that the map contains neither loops nor multiple edges. It is easy to see that, for maps enumerated by R_k , T_k and h_i , the presence of a loop automatically implies the presence of a multiple edge. Indeed, a loop separates the map into two regions, its exterior, which contains the outer face and its interior. One loop is then said to be included in another if it lies in its interior. This inclusion defines a partial ordering of the loops and we may consider one of the largest elements for this ordering, i.e. a loop which is not contained in the interior of any other loop. The face incident to the edge forming this loop and lying in its exterior is necessarily a triangle of the bulk. Indeed, the boundary of maps enumerated by R_k , T_k or h_i cannot contain loops. The two remaining edges of this triangle necessarily form a multiple edge surrounding the loop by connecting the endpoint of the loop to a distinct vertex in the exterior of the loop (the other possibility, namely that the two remaining edges of the triangle form two loops, is ruled out as, if so, one of these two loops would encircle the supposedly largest loop, a contradiction).

To suppress both loops and multiple edges in maps enumerated by R_k , T_k and h_i , it is thus sufficient to *suppress multiple edges only*. This translates into a simple substitution procedure to go from r_k , t_k , \tilde{h}_i to R_k , T_k and h_i : maps in the second family are directly obtained by simply replacing the edges in the maps of the first family by bundles of edges, as enumerated by R_1 . Note that the substitution is to be performed for each edge except for some of the boundary edges: in the case of r_k and t_k , edges of the right boundary must be left untouched as duplicating some of them would create a shortest path strictly on the left of the right-boundary. In the case of \tilde{h}_i , none of the boundary edges can be duplicated since we have to enforce Property 2 on maps enumerated by h_i .

Let us now describe in details the consequences of the substitution procedure. Consider a simple triangulation with a simple boundary of length L and call E , V and F its numbers of *inner* edges, vertices and *inner* faces respectively. From the relation $3F = 2E + L$ and the Euler relation $F + V - E - L = 1$, we immediately deduce

$$(7) \quad V = \frac{L + F}{2} + 1, \quad E = \frac{3F - L}{2}.$$

The case of h_i vs \tilde{h}_i : In this case, we have $L = i$ and the substitution requires a weight R_1 per inner edge of the simple triangulation. These edges are in number $(3F - i)/2 = 3F/2 - i/2$, hence we should give an extra weight $R_1^{3/2}$ per face, resulting in a total face weight

$$(8) \quad G = g R_1^{3/2}$$

together with a global factor $R_1^{-i/2}$. In other words, we have

$$(9) \quad h_i(g) = R_1^{-i/2} \tilde{h}_i(G).$$

The case of T_k vs t_k : For a slice enumerated by t_k , of arbitrary left-boundary length ℓ (between 1 and k), we have $L = 2\ell + 1$ and the substitution requires a weight R_1 for each inner edge of the simple slice, as well as for the base edge and for the edges of the left boundary (as already mentioned, there is no substitution attached to the edges of the right boundary as this boundary must be the unique shortest path between its extremal vertices). These edges are in number $(3F - (2\ell + 1))/2 + \ell + 1 = 3F/2 + 1/2$, hence we should give an extra weight $R_1^{3/2}$ per face as before, hence a total weight G , together with a global factor $R_1^{1/2}$. In other words, we now have

$$T_k(g) = R_1^{1/2} t_k(G).$$

The case of R_k vs r_k : For a slice enumerated by r_k , of arbitrary left-boundary length ℓ (between 1 and k), we have $L = 2\ell$ and the substitution requires a weight R_1 for each inner edge of the simple slice, as well as for the base edge and for the edges of the left boundary. These edges are in number $(3F - 2\ell)/2 + \ell + 1 = 3F/2 + 1$, hence we should again give a total weight G to each face, together with a global factor R_1 . In other words, we have

$$R_k(g) = R_1 r_k(G).$$

Introducing

$$\tilde{\Phi}(t) \equiv \tilde{\Phi}(t, G) = \sum_{i \geq 3} \tilde{h}_i(G) t^{i-3},$$

we read from (9) the correspondence

$$(10) \quad \Phi(T) = R_1^{-3/2} \tilde{\Phi}(t), \quad T = R_1^{1/2} t.$$

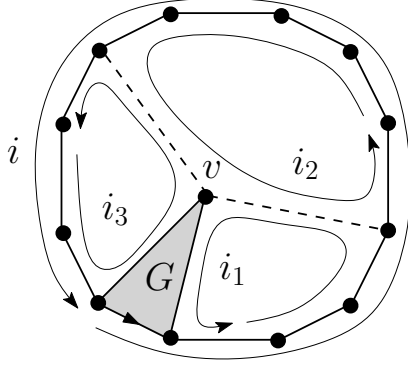


FIGURE 12. The decomposition of a map enumerated by h_i ($i \geq 3$) into $(p+1)$ domains (here $p=2$) enumerated by $h_{i_1}, h_{i_2}, \dots, h_{i_{p+1}}$ respectively. The boundary lengths of the domains satisfy $i_m \geq 3$ for $m=1, \dots, p+1$ and $\sum_{m=1}^{p+1} (i_m - 2) = i - 1$.

With the above correspondence, the system (3) is equivalent to the relations

$$(11) \quad \begin{cases} r_k = \frac{1}{1 - \sum_{i \geq 3} \tilde{h}_i t_{k-1}^{i-2}} = \frac{1}{1 - t_{k-1} \tilde{\Phi}(t_{k-1})} \\ t_k = \frac{\sum_{i \geq 3} \tilde{h}_i t_{k-1}^{i-3}}{1 - \sum_{i \geq 3} \tilde{h}_i t_{k-1}^{i-2}} = \frac{\tilde{\Phi}(t_{k-1})}{1 - t_{k-1} \tilde{\Phi}(t_{k-1})} \end{cases}$$

which determine r_k and t_k recursively from the initial condition $t_0 = 0$, while (5) becomes

$$(12) \quad r_k = 1 + t_{k-1} t_k .$$

Note that all these latter equations could have been obtained directly by applying the decompositions used in Section 3 directly to maps enumerated by r_k and t_k . Note also that considering simple triangulations is a way to get rid of R_1 (as well as of S_0), which disappeared from our recursion relations. As a final remark, it would be tempting to believe that the distance-dependent two-point function of simple triangulations is given by a formula as simple as eq. (1), say by simply replacing R_k and S_k by r_k and t_k . This is however not true since, in the cutting procedure illustrated in figures 3 and 4, the requirement of having no loop nor multiple edge *encircling the marked vertex* introduces non-trivial constraints on the associated slices, which are difficult to handle. So our detour in the ensemble of simple triangulations should here be viewed as a simple trick to simplify our recursion and to directly use the results of [10].

4.2. Equation for $\tilde{\Phi}(t)$. With the correspondence (8), (9) and (10), eq. (6) is fully equivalent to the following equation for $\tilde{\Phi}$:

$$(13) \quad t^2 \tilde{\Phi}^2(t) + (G + G \tilde{h}_3 t - t - G t^2) \tilde{\Phi}(t) + (G t - G \tilde{h}_3) = 0$$

where $\tilde{h}_3 = R_1^{3/2} h_3$ from (9). Let us recall here how to derive this equation, following Tutte's argument in [10]. Consider a map enumerated by \tilde{h}_i ($i \geq 3$): for $i=3$, the bulk of the map may possibly be reduced to a single triangle, contributing G to \tilde{h}_3 . In all the other cases, in order to guarantee Property 2, the triangle on the left of the root-edge of the map is incident to a vertex v lying strictly inside the bulk (see figure 12). This vertex is in all generality connected to a number $p \geq 0$ of the $i-2$ boundary vertices different from the extremities of

the root-edge. These connections are moreover performed by p single edges. Removing the triangle on the left of the root-edge and cutting along these p single edges gives rise to $p + 1$ domains which are triangulations with boundaries of lengths i_1, i_2, \dots, i_{p+1} , all larger than 3, satisfying

$$\sum_{m=1}^{p+1} (i_m - 2) = i - 1 .$$

The boundaries of these triangulations are simple curves and two boundary vertices cannot be linked by an internal edge. So the m -th piece is enumerated by \tilde{h}_{i_m} . This leads to the identity

$$\tilde{h}_i = G \delta_{i,3} + G \sum_{p \geq 0} \sum_{\substack{(i_1, \dots, i_{p+1}) \\ i_m \geq 3, m=1, \dots, p+1 \\ \sum_{m=1}^{p+1} (i_m - 2) = i - 1}} \tilde{h}_{i_1} \cdots \tilde{h}_{i_{p+1}} .$$

Multiplying by t^{i-3} and summing over $i \geq 3$, this rewrites as

$$\begin{aligned} \tilde{\Phi}(t) &= G + \frac{G}{t^2} \sum_{p \geq 0} \sum_{\substack{(i_1, \dots, i_{p+1}) \\ i_m \geq 3, m=1, \dots, p+1 \\ \sum_{m=1}^{p+1} (i_m - 2) \geq 2}} (\tilde{h}_{i_1} t^{i_1-2}) \cdots (\tilde{h}_{i_{p+1}} t^{i_{p+1}-2}) \\ &= G + \frac{G}{t^2} \left(\sum_{p \geq 0} \left(\sum_{i \geq 3} \tilde{h}_i t^{i-2} \right)^{p+1} - \tilde{h}_3 t \right) . \end{aligned}$$

Note the subtracted term corresponding to $p = 0$ and $i_1 = 3$, which, for arbitrary $p \geq 0$ and $i_m \geq 3$ is the only case for which the condition $\sum_{m=1}^{p+1} (i_m - 2) \geq 2$ is not satisfied. We end up with

$$\tilde{\Phi}(t) = G + \frac{G}{t^2} \left(\frac{t \tilde{\Phi}(t)}{1 - t \tilde{\Phi}(t)} - \tilde{h}_3 t \right) = G + \frac{G}{t} \left(\frac{\tilde{\Phi}(t)}{1 - t \tilde{\Phi}(t)} - \tilde{h}_3 \right)$$

which immediately leads to (13), and after substitution to the announced equation (6).

5. USING TUTTE'S SOLUTION

5.1. Tutte's generating function $\psi(\mathbf{x}, \mathbf{y})$. We shall now rely on Tutte's solution of the equation (13) to solve our new recursion relation. In order to directly use Tutte's expressions in [10], we need a slight (and harmless) reformulation of the generating function $\tilde{\Phi}(T)$. First, as noted in [10] for triangulations enumerated by \tilde{h}_i , the numbers E, V, F and $L = i$ of, respectively, inner edges, vertices, inner faces and boundary-edges (satisfying (7)) may be written as

$$E = 3n + m, \quad V = n + m + 3, \quad F = 2n + m + 1, \quad L = i = m + 3$$

for some $m, n \geq 0$ (since $i \geq 3$ and $V \geq i$). Using the variables m and n (instead of F and i), Tutte introduces (instead of $\tilde{\Phi}(t) = \tilde{\Phi}(t, G)$) the generating function

$$\psi(x, y) \equiv \sum_{m, n \geq 0} \psi_{m, n} x^n y^m, \quad \psi_{m, n} \equiv [G^{2n+m+1}] \tilde{h}_{m+3} .$$

We immediately read the correspondence

$$\begin{aligned}
\psi(x, y) &= \sum_{m \geq 0} y^m \sum_{n \geq 0} x^n [G^{2n+m+1}] \tilde{h}_{m+3} \\
&= \sum_{m \geq 0} y^m \frac{\tilde{h}_{m+3}(G)}{G^{m+1}} \quad \text{for } x = G^2 \\
&= \frac{1}{G} \sum_{m \geq 0} t^m \tilde{h}_{m+3}(G) \quad \text{for } y = Gt \\
&= \frac{\tilde{\Phi}(t)}{G}
\end{aligned}$$

or, in short

$$\psi(x, y) = \frac{\tilde{\Phi}(t)}{G}, \quad x = G^2, \quad y = Gt.$$

Note in particular that, setting $t = 0$, we have

$$g_3(x) \equiv \psi(x, 0) = \frac{\tilde{h}_3}{G}.$$

Setting the correspondence

$$\begin{array}{c}
\underbrace{y = Gt, \quad x = G^2, \quad g_3 = \tilde{h}_3/G, \quad \psi = \tilde{\Phi}/G}_{(14)} \\
\Updownarrow \\
\underbrace{t = y/\sqrt{x}, \quad G = \sqrt{x}, \quad \tilde{h}_3 = g_3 \sqrt{x}, \quad \tilde{\Phi} = \psi \sqrt{x}}
\end{array}$$

equation (13) is equivalent to

$$(15) \quad y^2 \psi^2(x, y) + (x + xy g_3(x) - y - y^2) \psi(x, y) + y - x g_3(x), \quad g_3(x) = \psi(x, 0)$$

which is precisely the form given by Tutte in [10].

As explained in [10], the solution of this equation is best expressed upon parametrizing x and y as

$$\begin{cases} x = \theta (1 - \theta)^3 \\ y = (1 - \theta)^3 \sigma. \end{cases}$$

With this parametrization, we have [10]

$$x g_3(x) = \theta (1 - 2\theta)$$

while $\psi(x, y)$ is fixed by

$$(16) \quad \psi(x, y) = \frac{1}{(1 - \theta)^3 \sigma} \left(\frac{\theta \sigma}{Y(\theta, \sigma)(1 + Y(\theta, \sigma))^2} + 1 \right)$$

where $Y(\theta, \sigma)$ is the solution of the quadratic equation

$$(17) \quad Y^2(\theta, \sigma) + (1 - \sigma + \theta \sigma) Y(\theta, \sigma) + \theta \sigma = 0$$

such that $Y(\theta, \sigma) \sim -\theta \sigma$ for small θ or small σ .

5.2. Writing the recursion in terms of Tutte's variable. Using the correspondence (14), θ and σ are to be considered as parametrizations of G and t via

$$\begin{cases} G^2 = \theta (1 - \theta)^3 \\ Gt = (1 - \theta)^3 \sigma. \end{cases}$$

In particular, eqs. (16) and (17) translate into

$$\begin{cases} \tilde{\Phi}(t) = \frac{C^3}{Y(t)(1+Y(t))^2} + \frac{1}{t} \\ Y(t) = \frac{1}{2} \left(Ct - 1 + \sqrt{(Ct - 1)^2 - 4C^3 t} \right) \\ C = \sqrt{\frac{\theta}{1-\theta}} = \sum_{n \geq 0} \frac{2}{3n+2} \binom{4n+1}{n} G^{2n+1}, \end{cases}$$

from which (together with the relation $G = g R_1^{3/2}$) we may obtain, as announced, an explicit expression of $\Phi(T) = R_1^{-3/2} \tilde{\Phi}(R_1^{-1/2} T)$ as a function of g and R_1 only.

We give here the explicit expression of $\tilde{\Phi}(t)$ only for completeness. To solve our recursion, it is indeed much simpler to directly work with Tutte's variables. In order to write the relations (11), we must specialize t to the values t_k (and t_{k-1}) hence we define

$$\begin{cases} \sigma_k \equiv \frac{G t_k}{(1-\theta)^3} = \sqrt{\frac{\theta}{(1-\theta)^3}} t_k \\ Y_k \equiv Y(\theta, \sigma_k) \end{cases}$$

for $k \geq 0$. Note that, by inverting the relation (17) defining $Y(\theta, \sigma)$, we may write σ_k in terms of Y_k as

$$(18) \quad \sigma_k = \frac{Y_k(1+Y_k)}{(1-\theta)Y_k - \theta}.$$

Let us now show that our recursion relation translates into a *particularly simple recursion relation* for Y_k . Writing the second line of eq. (11) in terms of σ_k , we get immediately

$$(19) \quad \sigma_k = \frac{\theta \psi_{k-1}}{1 - \sigma_{k-1} (1-\theta)^3 \psi_{k-1}}, \quad \psi_{k-1} \equiv \psi\left(\theta(1-\theta)^3, (1-\theta)^3 \sigma_{k-1}\right).$$

Using the relation (18) and the expression (16), we have the expressions

$$\begin{cases} \sigma_{k-1} = \frac{Y_{k-1}(1+Y_{k-1})}{(1-\theta)Y_{k-1} - \theta} \\ \psi_{k-1} = \frac{1}{(1-\theta)^3 \sigma_{k-1}} \left(\frac{\theta \sigma_{k-1}}{Y_{k-1}(1+Y_{k-1})^2} + 1 \right) = \frac{(1-2\theta) + (1-\theta)Y_{k-1}}{(1-\theta)^3(1+Y_{k-1})^2} \end{cases}$$

which, incorporated in (19) lead to the expression of σ_k in terms of Y_{k-1}

$$\sigma_k = \frac{(\theta - (1-\theta)Y_{k-1})((1-2\theta) + (1-\theta)Y_{k-1})}{(1-\theta)^3(1+Y_{k-1})}.$$

Comparing with (18), this yields the relation

$$\frac{Y_k(1+Y_k)}{(1-\theta)Y_k - \theta} = \frac{(\theta - (1-\theta)Y_{k-1})((1-2\theta) + (1-\theta)Y_{k-1})}{(1-\theta)^3(1+Y_{k-1})}$$

which we may equivalently write as

$$\left(Y_k + Y_{k-1} + \frac{1-2\theta}{1-\theta} \right) \left(Y_k + \frac{\theta}{(1-\theta)^2} \frac{\theta - (1-\theta)Y_{k-1}}{1+Y_{k-1}} \right) = 0.$$

In order to choose which of the two factors we should cancel, we recall that both Y_k and Y_{k-1} should vanish for $\theta \rightarrow 0$, in which case only the second factor above vanishes. We are thus led to cancel the second factor in the above product, hence we eventually end up with the desired recursion relation for Y_k :

$$(20) \quad Y_k = -\frac{\theta}{(1-\theta)^2} \frac{\theta - (1-\theta)Y_{k-1}}{1+Y_{k-1}}.$$

This relation is equivalent to our initial recursion (3) for T_k . It fixes Y_k for all $k \geq 0$ from the initial condition $Y_0 = 0$ (since $t_0 = 0$, hence $\sigma_0 = 0$) and the knowledge of Y_k allows us to obtain σ_k , t_k and eventually T_k .

5.3. Solving the recursion relation. Getting the solution of the recursion relation (20) is a standard exercise and goes as follows: consider more generally the equation

$$Y_k = f(Y_{k-1}) \quad f(Y) \equiv \frac{aY + b}{cY + d}.$$

Introducing the two fixed points α and β of the function f (i.e. the two solutions of $f(Y) = Y$), then the quantity

$$W_k = \frac{Y_k - \alpha}{Y_k - \beta}$$

is easily seen to satisfy $W_k = \lambda W_{k-1}$, hence

$$W_k = \lambda^k W_0, \quad \lambda \equiv \frac{c\beta + d}{c\alpha + d}.$$

This immediately yields Y_k via $Y_k = (\alpha - \beta W_k)/(1 - W_k)$ (strictly speaking, we must have $\alpha \neq \beta$, which can be verified a posteriori in our case).

To solve eq. (20), we may take

$$a = \frac{\theta}{1 - \theta}, \quad b = -\frac{\theta^2}{(1 - \theta)^2}, \quad c = 1, \quad d = 1$$

and thus

$$f(Y) - Y \propto Y^2 + \frac{1 - 2\theta}{1 - \theta}Y + \frac{\theta^2}{(1 - \theta)^2} = (Y - \alpha)(Y - \beta)$$

so that

$$\alpha + \beta = -\frac{1 - 2\theta}{1 - \theta}, \quad \alpha\beta = \frac{\theta^2}{(1 - \theta)^2}.$$

We deduce

$$\frac{\lambda}{(1 + \lambda)^2} = \frac{\frac{\beta+1}{\alpha+1}}{\left(1 + \frac{\beta+1}{\alpha+1}\right)^2} = \frac{1 + \alpha + \beta + \alpha\beta}{(2 + \alpha + \beta)^2} = \frac{1 - \frac{1-2\theta}{1-\theta} + \frac{\theta^2}{(1-\theta)^2}}{\left(2 - \frac{1-2\theta}{1-\theta}\right)^2} = \theta.$$

In other word, we have the correspondence between θ and λ

$$\theta = \frac{\lambda}{(1 + \lambda)^2} = \frac{1}{\left(\frac{1}{\sqrt{\lambda}} + \sqrt{\lambda}\right)^2}$$

(note that since $\lambda = (\beta + 1)/(\alpha + 1)$, the condition $\alpha \neq \beta$ is equivalent to the condition $\lambda \neq 1$). In terms of λ , we have

$$\alpha + \beta = -\frac{1 + \lambda^2}{1 + \lambda + \lambda^2}, \quad \alpha\beta = \frac{\lambda^2}{(1 + \lambda + \lambda^2)^2}$$

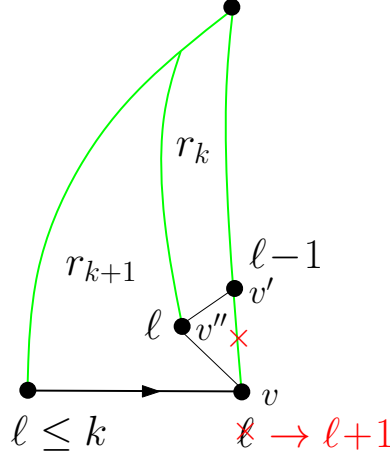
so that we may take

$$\alpha = -\frac{\lambda^2}{1 + \lambda + \lambda^2}, \quad \beta = -\frac{1}{1 + \lambda + \lambda^2}.$$

Since $Y_0 = 0$, we have $W_0 = \alpha/\beta = \lambda^2$, so that

$$W_k = \lambda^{k+2}, \quad Y_k = \frac{\alpha - \beta W_k}{1 - W_k} = -\frac{\lambda^2}{1 + \lambda + \lambda^2} \times \frac{1 - \lambda^k}{1 - \lambda^{k+2}}.$$

This solves our recursion relation (11). Returning to the slice generating functions, we

FIGURE 13. A schematic explanation of the relation $t_k = G r_k r_{k+1}$.

indeed have

$$\begin{aligned} t_k &= \sqrt{\frac{(1-\theta)^3}{\theta}} \sigma_k = \sqrt{\frac{(1-\theta)^3}{\theta}} \frac{Y_k(1+Y_k)}{(1-\theta)Y_k - \theta} \\ &= \sqrt{\frac{\lambda}{1+\lambda+\lambda^2}} \times \frac{(1-\lambda^k)(1-\lambda^{k+3})}{(1-\lambda^{k+1})(1-\lambda^{k+2})} \end{aligned}$$

and from (12)

$$r_k = 1 + t_{k-1} t_k = \frac{(1+\lambda)^2}{1+\lambda+\lambda^2} \times \frac{(1-\lambda^k)(1-\lambda^{k+2})}{(1-\lambda^{k+1})^2}$$

where λ may be viewed as parametrizing G via

$$(21) \quad G = \sqrt{\theta(1-\theta)^3} = \frac{\sqrt{\lambda(1+\lambda+\lambda^2)^3}}{(1+\lambda)^4}.$$

The condition $\lambda \neq 1$ limits the range of G between 0 and $3\sqrt{3}/16$, in agreement with the fact that the number of simple triangulations with n faces (and a boundary of finite length) has a large n exponential growth of the form $(16/(3\sqrt{3}))^n$ [10]. With their explicit values, we may easily verify the relation

$$(22) \quad t_k = G r_k r_{k+1}$$

which, for $k \geq 1$, can be explained combinatorially as follows: take a map enumerated by t_k , with left-boundary length ℓ ($1 \leq \ell \leq k$) and consider the triangle immediately on the left of the first edge of the right boundary linking the endpoint v of the root-edge to a vertex v' at distance $\ell - 1$ from the apex (see figure 13). The third vertex v'' incident to this triangle is distinct from the other two (as there are no loops) and is at distance ℓ from the apex. More generally, all neighbors of v but v' must be at distance ℓ from the apex as otherwise, we would have a shortest path from v to the apex different from the right-boundary. Removing the edge from v to v' (and the incident triangle), the distance from the vertex v to the apex therefore becomes $\ell + 1$. Cutting the resulting map along the leftmost shortest path from v'' to the apex creates two slices, one enumerated by r_k and the other by r_{k+1} (see figure 13) hence the relation (22).

6. FINAL EXPRESSIONS

Let us now return to our original problem and obtain expressions for R_k , S_k and eventually for the two-point function G_k . From the relations

$$(23) \quad r_k = \frac{R_k}{R_1}, \quad t_k = \frac{S_k - S_0}{R_1^{1/2}}, \quad G = g R_1^{3/2}$$

we can get R_k and S_k as functions of λ , R_1 and S_0 . Introducing $R_\infty \equiv \lim_{k \rightarrow \infty} R_k$, $S_\infty \equiv \lim_{k \rightarrow \infty} S_k$ and similarly r_∞ and t_∞ , we can write instead R_k and S_k as functions of λ , R_∞ and S_∞ via the correspondence

$$(24) \quad \begin{cases} r_\infty = \frac{(1 + \lambda)^2}{1 + \lambda + \lambda^2} = \frac{R_\infty}{R_1} \\ t_\infty = \sqrt{\frac{\lambda}{1 + \lambda + \lambda^2}} = \frac{S_\infty - S_0}{R_1^{1/2}}. \end{cases}$$

Eq. (23) reads indeed

$$\begin{cases} R_k = R_\infty \frac{r_k}{r_\infty} \\ S_k = S_\infty - \sqrt{\frac{R_\infty}{r_\infty}} t_\infty \left(1 - \frac{t_k}{t_\infty}\right) \\ g = G \left(\frac{r_\infty}{R_\infty}\right)^{3/2} \end{cases}$$

with, as a consequence of (22), $t_\infty = G r_\infty^2$, hence $\sqrt{R_\infty/r_\infty} t_\infty = g R_\infty^2$. We end up with the explicit relations

$$\begin{cases} R_k = R_\infty \frac{(1 - \lambda^k)(1 - \lambda^{k+2})}{(1 - \lambda^{k+1})^2} \\ S_k = S_\infty - g R_\infty^2 \lambda^k \frac{(1 - \lambda)(1 - \lambda^2)}{(1 - \lambda^{k+1})(1 - \lambda^{k+2})} \end{cases}$$

which reproduce the formulas found in [6, 4]. To end our calculation, we still have to express λ , R_∞ and S_∞ in terms of the weight g only. The quantities R_∞ and S_∞ are simply obtained as the solutions of the system obtained by letting $k \rightarrow \infty$ in eq. (2), namely

$$(25) \quad \begin{cases} R_\infty = 1 + 2g R_\infty S_\infty \\ S_\infty = g(S_\infty^2 + 2R_\infty). \end{cases}$$

The desired solution is entirely determined by the condition $R_\infty = 1 + O(g^2)$ and $S_\infty = 2g + O(g^3)$. As for λ , we note that, from the expression (21) of G and that, (24), of r_∞ , we can immediately write that

$$G^2 r_\infty^3 = \frac{\lambda}{(1 + \lambda)^2} = \frac{1}{\lambda + \frac{1}{\lambda} + 2}$$

while, from eqs. (23) and (24), $G^2 r_\infty^3 = g^2 R_\infty^3$. To summarize, λ is connected to g via

$$(26) \quad \lambda + \frac{1}{\lambda} + 2 = \frac{1}{g^2 R_\infty^3}$$

which is precisely the relation found in [4]. To be as explicit as in the case of simple triangulations, let us conclude this section by expressing R_∞ , S_∞ and g in terms of the parameter λ . Introducing the intermediate variable $s \equiv S_\infty/\sqrt{R_\infty}$, we may write (25) as

$$\begin{cases} R_\infty = 1 + 2\sqrt{g^2 R_\infty^3} s \\ R_\infty s = \sqrt{g^2 R_\infty^3} (s^2 + 2) \end{cases}$$

which, after eliminating s from the system, implies

$$R_\infty^2 = 1 + 8(g^2 R_\infty^3) .$$

From (26), this leads to

$$R_\infty = \left(1 + \frac{8}{\lambda + \frac{1}{\lambda} + 2}\right)^{1/2} = \frac{\sqrt{1 + 10\lambda + \lambda^2}}{1 + \lambda}$$

with λ parametrizing g via

$$g = \frac{1}{R_\infty^{3/2} (\lambda + \frac{1}{\lambda} + 2)^{1/2}} = \frac{\sqrt{\lambda(1 + \lambda)}}{(1 + 10\lambda + \lambda^2)^{3/4}} .$$

Note that, since $\lambda \neq 1$, g ranges from 0 to $1/(2 \cdot 3^{3/4})$. in agreement with the fact that the number of triangulations with n faces grows like $(2 \cdot 3^{3/4})^n$ (see for instance [7] for explicit formulas). Finally we have

$$S_\infty = \frac{R_\infty - 1}{2gR_\infty} = (1 + 10\lambda + \lambda^2)^{1/4} \frac{\sqrt{1 + 10\lambda + \lambda^2} - (1 + \lambda)}{2\sqrt{\lambda(1 + \lambda)}} .$$

Plugging the above formulas for R_k and S_k in eq. (1) and the expressions for g , R_∞ and S_∞ , we arrive at the remarkably simple expression of the distance-dependent two-point function G_k for $k \geq 1$:

$$G_k = \frac{(1 - \lambda^3)(1 + 10\lambda + \lambda^2)}{1 + \lambda} \frac{\lambda^{k-1}(1 + \lambda^{k+1})}{(1 - \lambda^k)(1 - \lambda^{k+1})(1 - \lambda^{k+2})} - \delta_{k,1} .$$

7. CONCLUSION

In this paper, we presented a new technique to compute the distance-dependent two-point function of planar triangulations by first deriving and then solving a new recursion for the intimately related slice generating functions. Although our method makes a crucial use of properties which are specific to triangulations, it is likely that it could be generalized to other families of maps. In particular, the case of *planar quadrangulations* seems promising for a similar treatment.

Our approach is based on the existence, in slices of left-boundary length ℓ , of a dividing line connecting the right and left boundaries of the slice via $\ell - 1 \rightarrow \ell - 1$ edges. Upon gluing, say the two boundaries of an R -slice with $\ell = k$, we produce via the equivalence displayed in figure 4 a pointed rooted triangulation whose root-edge is “of type” $k \rightarrow k - 1$ with respect to the marked vertex. After gluing, the dividing line creates a simple closed path made of edges connecting vertices at distance $k - 1$ from the marked vertex, and which separates the marked vertex from the root-vertex. By construction, all the vertices strictly outside the domain containing the marked vertex are at distance at least k from this vertex. We may thus interpret the dividing line as the boundary of the *hull of radius $k - 1$* centered at the marked vertex which, so to say, is formed of the ball of radius $k - 1$ together with *all the complementary connected domains* (thus containing vertices at distance at least k from the marked vertex) *except that containing the root-vertex*. In other words, we may decide to use the dividing line as a way to precisely *define* what we shall call the hull of radius $k - 1$ centered at the marked vertex, namely the domain delimited by this line and containing the marked vertex. Note that this definition is *not equivalent* to that given in [8] where the hull of radius $k - 1$ is constructed explicitly from the ball of radius $k - 1$, itself defined at the set of all triangles incident to at least one vertex at distance $\leq k - 2$ from the marked vertex (see [8]).

Finally, our recursive construction allows us to define dividing lines within each of the slices (enumerated by T_{k-1}) which appear in the slice decomposition of the hull (see figure 4). After gluing, the concatenation of these dividing lines creates a simple closed path made of edges connecting vertices at distance $k - 2$ from the marked vertex, and which separates the

marked edge from root-vertex. This line may be viewed as the boundary of the hull of radius $k - 2$ centered at the marked vertex. We may in this way define hulls of all radii between 1 and $k - 1$ and our recursion relations should in principle allow us to describe the statistics of the lengths of these hull boundaries, an analysis yet to be done.

REFERENCES

- [1] J. Ambjørn and T.G. Budd. Trees and spatial topology change in causal dynamical triangulations. *J. Phys. A: Math. Theor.*, 46(31):315201, 2013.
- [2] J. Bouttier, P. Di Francesco, and E. Guitter. Geodesic distance in planar graphs. *Nucl. Phys. B*, 663(3):535–567, 2003.
- [3] J. Bouttier, É. Fusy, and E. Guitter. On the two-point function of general planar maps and hypermaps. *Ann. Inst. Henri Poincaré Comb. Phys. Interact.*, 1(3):265–306, 2014. arXiv:1312.0502 [math.CO].
- [4] J. Bouttier and E. Guitter. Planar maps and continued fractions. *Comm. Math. Phys.*, 309(3):623–662, 2012.
- [5] R. Cori and B. Vauquelin. Planar maps are well labeled trees. *Canad. J. Math.*, 33(5):1023–1042, 1981.
- [6] P. Di Francesco. Geodesic distance in planar graphs: an integrable approach. *Ramanujan J.*, 10(2):153–186, 2005.
- [7] Z.-C. Gao. The number of rooted triangular maps on a surface. *Journal of Combinatorial Theory, Series B*, 52(2):236–249, 1991.
- [8] M.A. Krikun. Uniform infinite planar triangulation and related time-reversed critical branching process. *Journal of Mathematical Sciences*, 131(2):5520–5537, 2005.
- [9] G. Schaeffer. *Conjugaison d'arbres et cartes combinatoires aléatoires*. PhD thesis, Université Bordeaux I, 1998.
- [10] W. T. Tutte. A census of planar triangulations. *Canad. J. Math.*, 14:21–38, 1962.

INSTITUT DE PHYSIQUE THÉORIQUE, CEA, IPhT, 91191 GIF-SUR-YVETTE, FRANCE, CNRS, UMR 3681
E-mail address: emmanuel.guitter@cea.fr