# A Transfer Matrix approach to the Enumeration of Knots

J. L. Jacobsen *and* P. Zinn-Justin

*Laboratoire de Physique Théorique et Modèles Statistiques*
*Université Paris-Sud, Bâtiment 100*
*91405 Orsay Cedex, France*

We propose a new method to enumerate alternating knots using a transfer matrix approach. We apply it to count numerically various objects, including prime alternating tangles with two connected components, up to order 18–22, and comment on the large-order behavior in connection with one of the authors' conjecture.

02/2001

## 1. Introduction

The classification of knots is a subject with a long history. More than a hundred years ago, Tait, Kirkman and Little tried to draw "by hand" the first knots (up to 10 crossings, but with some mistakes); nowadays, it is possible with the modern tools of knot theory and the use of computers to create programs that generate all possible (prime) knots up to 16 crossings [10]. Here we shall consider the simplest problem, namely the *enumeration* of knots (or similar objects). Furthermore we shall concentrate on so-called *alternating* objects which have much simpler properties than their generic counterparts. Even though they are probably asymptotically subdominant (this has been proved in the case of links [11]), they are quite interesting to study if only because one can prove much more about them than about general knots. In fact, the number of prime alternating *tangles* is known exactly [11]; however, a general tangle has an arbitrary number of connected components, and in the present paper we want to be able to control this number, that is to count knots or similar objects (e.g. tangles with exactly 2 connected components).

The key concept that will be used is that of a *transfer matrix*. A standard object of statistical mechanics, where it describes the discrete time evolution of a system, the transfer matrix is also a very efficient numerical tool for the combinatorial enumeration of discrete objets. This approach has recently been used to investigate the properties of 2D Lorentzian gravity [1], the enumeration of plane meanders [2,3], and the coupling of matter fields to 2D quantum gravity equipped with a Hamiltonian circuit [4]. In contradistinction to the standard situation, where the transfer matrix is used to construct the partition function for a statistical mechanics system on a semi-infinite strip of finite width, these combinatorial applications possess a state space which is different in each time slice. Accordingly the underlying physical models are not defined on a regular lattice, but belong to the realm of two-dimensional quantum gravity. In all these examples, the common feature that ensures the existence of a transfer matrix is that the objects under consideration permit a time ordering. For the meanders (resp. the Hamiltonian circuits) this was attained by "reading" each object as one moves along the river (resp. the circuit), adding one intersection (vertex) at each time step. Clearly, this stategy in also appropriate for enumerating knots with *one* connected component.

This type of enumeration problems have many interesting connections with mathematical physics [1–9]. In particular, in [5,6,7] the counting of links, knots or tangles was reduced to the evaluation of integrals over $N \times N$ hermitian matrices, in the limit $N \to \infty$.

Though these integrals cannot be computed explicitly in general, the subject of matrix models and its well-known connection to 2D quantum gravity [12] provide some information on the universal quantities of the model. This led to conjectures of the *asymptotic* behavior of the number of such objects when the number of crossings goes to infinity; and one motivation of the present work is to check the conjecture concerning the asymptotic number of prime alternating knots.

The paper is organized as follows: after some brief definitions in section 2, which enable us to give a more precise meaning to the problem that we are addressing, the basic principles underlying our transfer matrix approach are described in section 3. A number of practical details concerning our implementation of this algorithm are then given in section 4. Finally, section 5 gives the results of the numerical enumeration of prime alternating knots. As a byproduct we also count several other objects of combinatorial interest.

## 2. Basic definitions

A *knot* is a smooth circle embedded in $\mathbb{R}^3$, considered up to homeomorphisms of $\mathbb{R}^3$. In this paper we shall study slightly different objects, namely knots with "external legs"; a knot with $2n$ external legs is a a collection of $n$ intervals embedded in a ball $B$ and whose endpoints are given distinct points on the boundary $\partial B$, considered up to orientation preserving homeomorphisms of $B$ that reduce to the identity on $\partial B$. These knots with external legs are nothing but tangles in which no closed loops are allowed. Knots with 2 and 4 external legs will be of special interest to us. When we refer to standard knots we shall from now on call them closed knots. Clearly, each knot with 2 external legs can be transformed into a closed knot by joining its endpoints through a smooth curve outside $B$. And conversely, by cutting a closed knot once it may be turned into a knot with 2 external legs, but in general this transformation is not unique, in the sense that the topological properties of the resulting knot with 2 external legs depend on the point where the closed knot was cut. This means that counting knots with external legs does not help count closed knots. However we shall always count the former and never the latter; those are the objects whose generating series will have "nice" properties.[1]

[1] At the level of diagrams (see below for a definition of diagrams associated to knots), one can go slightly further by saying that, as the Feynman rules of perturbative quantum field theory tell us, counting planar diagrams with external legs is equivalent to counting closed planar diagrams weighted by a symmetry factor. This also explains why counting closed objects with a weight of 1 is unnatural.

It is common to represent such objects by their projections on a plane; we consider regular projections with only double points where lines cross. To avoid redundancies, we shall concentrate on *prime* knots, whose diagrams cannot be decomposed as a sum of pieces connected to each other by only two lines, and on *reduced* diagrams that contain no irrelevant crossings (Fig. 1). We shall need another related concept: a diagram is said to be *$r$-particle reducible (resp. irreducible)* ($r$PR, resp. $r$PI), $r$ positive integer, if it can (resp. cannot) be divided into disconnected components by cutting $r$ edges. For closed knots and knots with 2 external legs, the notion of a prime knot coincides with the 2-particle irreducibility of its diagram(s).[2]

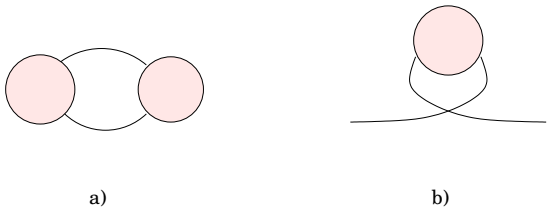


**Fig. 1:** a) a non prime link; b) an irrelevant (or "nugatory") crossing.

A diagram is called *alternating* if one meets under- and over-crossings alternatingly as one travels along each loop. A remarkable property (see e.g. [9], page 21) is that in the case of alternating diagrams there is no need to "remember" which crossings are under/over. In other words, two alternating knot diagrams have the same underlying planar diagram if and only if they are identical, or related by an overall flip under $\leftrightarrow$ over in the case of closed knots. This greatly simplifies their enumeration.

To a given knot can correspond several diagrams. In fact, in the case of alternating diagrams, two alternating reduced knot diagrams represent the same object if and only if they are related by a sequence of moves acting on tangles called "flypes" (see Fig. 2) [13]. This is of course an essential distinction when one is interested in counting such objects, and we shall briefly discuss it now. The general idea is the same as in [7]; however, the actual equations shown used here are simpler than those in [7], and their proof will be given elsewhere [14] in the more general framework of colored links.

**Fig. 2:** The flype of a tangle.

[2] Note however that this is not the case for knots with $2n$ external legs, $n \geq 2$; see for example [5] for a discussion of 2PI tangles.

Since the diagrams we shall work with most of the time are diagrams of knots with 2 external legs, we shall simply call them knot diagrams. Let us start by defining the generating series $G(g)$

$$G(g) = \sum_{p=0}^{\infty} a_p g^p \tag{2.1}$$

where $a_p$ is the number of knot diagrams with $p$ crossings, or equivalently, the number of topologically inequivalent open curves in the plane going from $(-\infty, 0)$ to $(+\infty, 0)$ with $p$ regular self-intersections. We similarly define $\Sigma_1(g)$ to count 1PI knot diagrams, and $\Sigma_2(g)$ to count 2PI knot diagrams (with the trivial diagram excluded and the two diagrams with one crossing included). The following relations hold: $\Sigma_1(g)$ is simply given by

$$G(g) = \frac{1}{1 - \Sigma_1(g)} \tag{2.2}$$

whereas $\Sigma_2(g)$ is given by the implicit equation

$$1 + \Sigma_2(g) = G\left(\frac{g}{(1 + \Sigma_2(g))^2}\right) \tag{2.3}$$

(see [5] for details).

Next, we want to take into account the flyping equivalence in order to count the actual objects and not diagrams. The data of $G(g)$ is insufficient for this purpose; we need a more general object, a double generating series $G(g_1, g_2)$

$$G(g_1, g_2) = \sum_{p_1, p_2=0}^{\infty} a_{p_1,p_2} g_1^{p_1} g_2^{p_2} \tag{2.4}$$

where $a_{p_1,p_2}$ is the number of topologically inequivalent open curves in the plane (going from $(-\infty, 0)$ to $(+\infty, 0)$) with $p_1$ regular self-intersections and $p_2$ *tangencies*, see Fig. 3.

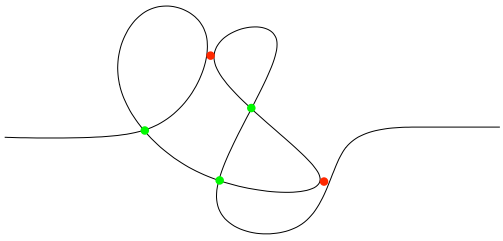

**Fig. 3:** Open curve with self-intersections (green dots) and tangencies (red dots).

Also, note that we have $a_{p,0} = a_p$ or $G(g) = G(g_1 = g, g_2 = 0)$, so that there is more information in $G(g_1, g_2)$ than in $G(g)$.

Since flypes act on tangles, we are led to the introduction of a few more generating functions which are related to the counting of tangles, $G_1$, $G_2$, $\Gamma_1$ and $\Gamma_2$; they are all expressible in terms of $G$ alone via:

$$G = 1 + 2g_1 G_2 + 2g_2(G_1 + G_2) \tag{2.5a}$$

$$\frac{\partial}{\partial g_2} G_2 = \frac{\partial}{\partial g_1}(G_1 + G_2) \tag{2.5b}$$

$$\Gamma_1 = G_1 \tag{2.5c}$$

$$\Gamma_2 = G_2 - G^2 \tag{2.5d}$$

We also introduce for our convenience an extra parameter $t$ which counts the number of edges of the diagram; it is easy to show that the following formulae take care of it: $G(g_1, g_2, t) \equiv \frac{1}{t} G(g_1/t^2, g_2/t^2)$ and $\Gamma_i(g_1, g_2, t) \equiv \frac{1}{t^2}\Gamma_i(g_1/t^2, g_2/t^2)$.

The parameters $t$, $g_1$ and $g_2$ must then be chosen as a function of $g$ according to the following *renormalization procedure* (see [14]):

$$1 = G(g_1(g), g_2(g), t(g)) \tag{2.6a}$$

$$g_1(g) = g(1 - 2H_2'(g)) \tag{2.6b}$$

$$g_2(g) = -g(H_1'(g) + V_2'(g)) \tag{2.6c}$$

where $H_1'(g)$, $H_2'(g)$ and $V_2'(g)$ are auxiliary quantities defined by:

$$H_2' \pm H_1' = \frac{(1 \mp g)(\Gamma_2 \pm \Gamma_1) \mp g}{1 + (1 \mp g)(\Gamma_2 \pm \Gamma_1) \mp g} \tag{2.7a}$$

$$V_2' = (1 - g)\Gamma_2(1 - H_2' - H_1')^2 \tag{2.7b}$$

where we have omitted all arguments for the sake of brevity; in particular $\Gamma_i \equiv \Gamma_i(g_1(g), g_2(g), t(g))$.

This ensures that the flypes are appropriately taken into account and for example that $\Gamma_1(g_1(g), g_2(g), t(g))$ and $\Gamma_2(g_1(g), g_2(g), t(g))$ are the desired generating functions for the number of tangles with 2 connected components of type 1 and 2 respectively (see [7] for a definition of type; the total number of tangles is given by $\Gamma_1 + 2\Gamma_2$). Similarly one could define other generating functions of the 2 variables $g_1$ and $g_2$ (higher correlation functions in the matrix model language) which would count objects with more external legs.

## 3. Space of states and transfer matrix

We now come to the description of the transfer matrix approach. The latter requires first that the knot diagrams be represented in an appropriate way (3.1); next we have to define the space of states on which the matrix acts (3.2); and finally define the transfer matrix itself (3.3). At first we shall concentrate on the usual knot diagrams with self-intersections only; a direct application of the first 3 subsections leads to the enumeration of alternating knot diagrams, but to count actual alternating knots, one has to introduce a refined procedure (addition of tangencies) to which subsection 3.4 is devoted.

### *3.1. Representation of a knot diagram using time slices*

A basic ingredient of the transfer matrix approach is the ability to cut the object one is studying into slices, which represent the state of the system at fixed (discrete) time. If we apply this to knots a complication arises. The naive idea would be to draw the knot diagrams on the plane in such a way that time would correspond to one particular coordinate of the plane, that is to read the knot diagrams "from left to right". Here, this idea does not work directly, and one is led to a slightly more sophisticated notion of slices, which we shall explain using the example of Fig. 4.

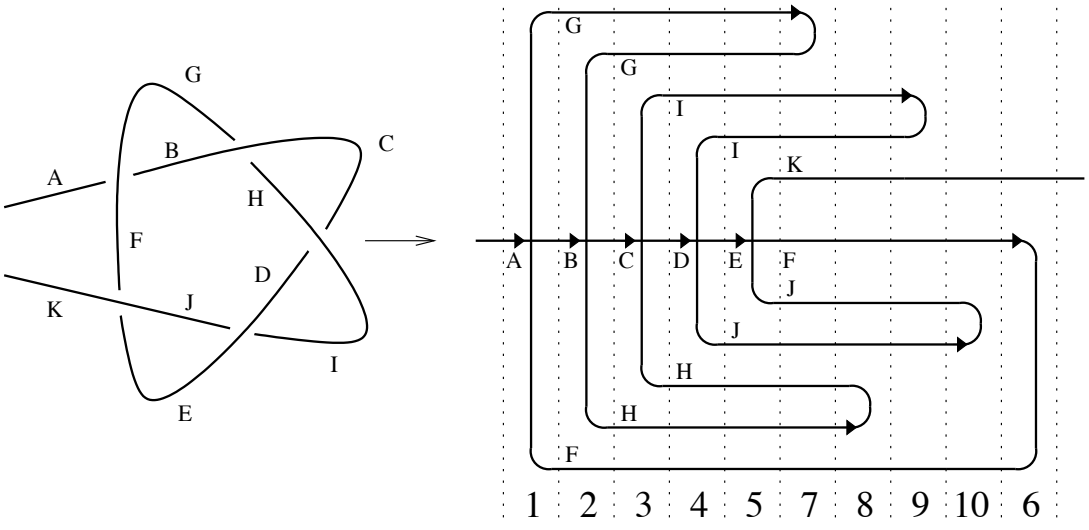


**Fig. 4:** A knot diagram and its corresponding "sliced" diagram. The steps are ordered by their number below the diagram. At each step, the active line is distinguished by an arrow.

The general idea is to follow the knot as it winds around itself from one "incoming" external leg to the other "outgoing" leg, and write down step by step the crossings and the lines that are crossed. We shall call the edge of the diagram that we are currently following the active line. Of course this is a step-dependent concept since every edge of the diagram will at some point become the active line. The edges of the diagram have been labelled from A to K in the order in which we encounter them. The precise recipe is as follows. At each step there are two possibilities: 1) The label of the active line does not appear anywhere else in the picture drawn this far. We then proceed to the next crossing and draw it. 2) The active line already appears somewhere. We connect the active line to its other appearance. We then follow this new line until it reemerges as an open line: this will be the new active line. In the case of Fig. 4, steps 1–5 are of type 1) whereas steps 6–10 are of type 2). In general, steps of type 1) and 2) can appear in any order, except that at any stage the number of type 2) steps performed cannot be greater than the number of type 1) steps. In a more algebraic language, the sequence of 1s and 2s forms a Dyck word [15].

After step 5, for the first time the active line, which is now the edge F, is already present (AB–FG crossing). When we reconnect the two occurrences of the edge F, we notice that some open lines are "imprisoned" inside the new arch we have created, and therefore we cannot draw step 6 right after step 5. Instead we must allow lines inside the new arch to continue to evolve (steps 8 and 10), keeping in mind that they cannot have any contact with the lines outside the arch.[3]

Using this procedure, to each knot diagram we can associate a "sliced" diagram; and it is easy to show that two "sliced" diagrams are topologically equivalent (in the sense of graphs) if and only if they consist of the same sequence of steps. We shall now show how to generate all knots with such diagrams using a transfer matrix.

### *3.2. Space of states*

The vector space on which the transfer matrix acts will be spanned by the intermediate states created in the process described in the previous section. The important point to remember is that at each step, we had to take into account the following information: a) the current (open) lines, including the active line; b) the existing connections of the lines

[3] Note that the order in which we *draw* the steps from left to right can contain some arbitrariness for more complex knots; however the sequence of steps is unique.

from the left: pairs of lines are connected by what we call *left arches*; c) the different groups of lines which can still be connected to each other from the right; the lines are divided in multiple connected components by what we call *right arches*. All this information has to be included in the state of the system.

A basis state will therefore be described by a series of left and right arches and the position of the active line. As an illustration, we show all the intermediate states of the example of Fig. 4 on Fig. 5.

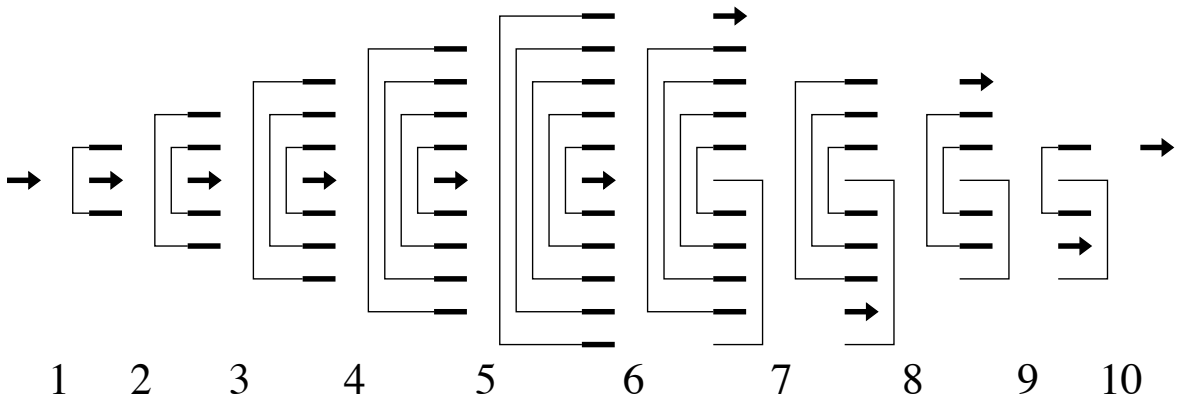


**Fig. 5:** A sequence of intermediate states. The active line is denoted by an arrow.

For practical applications, it is important to notice that some configurations should be forbidden. Firstly, we have states that cannot evolve into knots (Fig. 6 a) and b)). Secondly, there are redundant states that are equivalent to simpler states (Fig. 6 c) and d)); we shall describe a systematic simplification procedure in section 4 below.

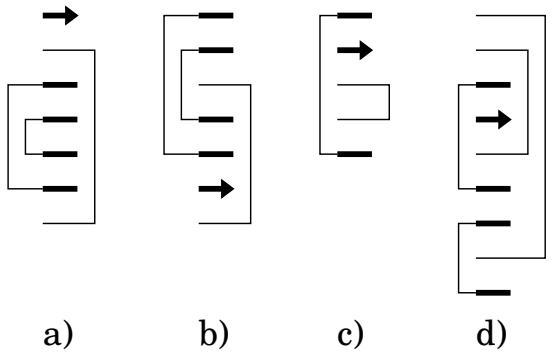


**Fig. 6:** Examples of forbidden configurations. a) A region enclosed within a right arch which has no connections with the outside. b) A region enclosed within a right arch with an odd number of lines. c) An empty right arch. d) Two consecutive opening right arches.

*3.3. Transfer matrix*

We now describe the transfer matrix $T$. Its entries $T_{ab}$, where $a$ and $b$ are two basis states of the kind defined in the previous section, are 0 unless $a$ is a descendant of $b$, in which case $T_{ab}$ is the number of ways $a$ can be obtained from $b$. An allowed state $a$ is a descendant of $b$ if it can be obtained via a transformation of one of the two types shown on Fig. 7, followed by an arbitrary number of simplifications (Fig. 8).

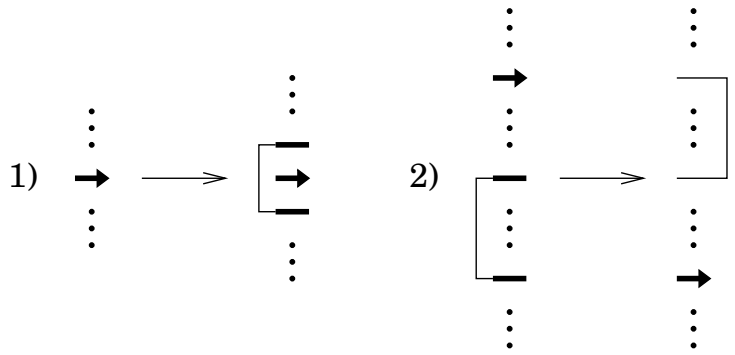


**Fig. 7:** The two types of transformations.

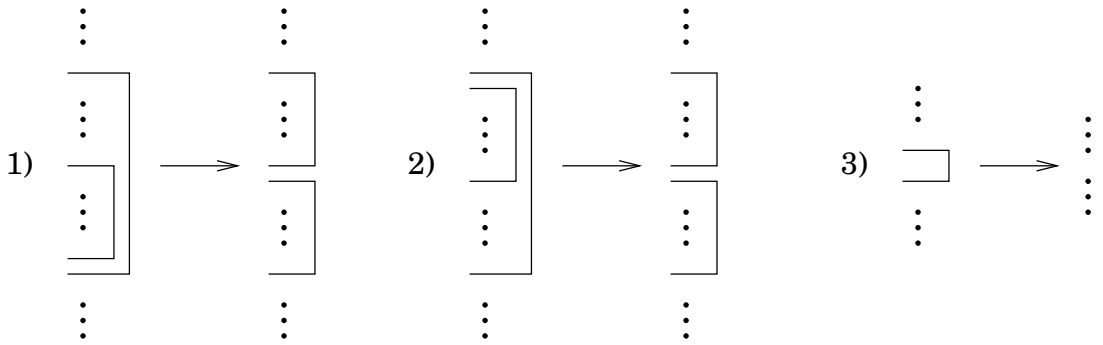


**Fig. 8:** Possible simplifications.

The two transformations of Fig. 7 reproduce the procedure described in section 3.1, but we have now reformulated it in terms of states.

Note that in transformation 2), the relative positions of the active line and the line onto which it connects are not arbitrary. The screening role of the right arches should be respected, so that the active line can only connect onto lines belonging to the same block; and the relative distance between the active line and the line that it connects onto must be *odd*. For each of the allowed connection, the position of the new active line is found by following the line just connected along the left arch to which it belongs.

Transformation 1) (resp. 2)) increases (resp. decreases) the number of lines by 2. This means that at step $p$, for any intermediate state, the number of crossings $n$ and the number of lines $l$ (excluding the active line) are related by $l = 2(p - 2n)$. In particular, we claim that the number of knot diagrams with $n$ crossings is given by $\langle 0 | T^{2n} | 0 \rangle$, where $|0\rangle$ is the state with the active line only. In this notation, $|0\rangle$ is understood to be assigned the weight one, and $\langle 0|$ acts as a projection operator: $\langle 0|0\rangle = 1$. Formally we have

$$G(g) = \langle 0 | \frac{1}{1 - gT^2} | 0 \rangle \tag{3.1}$$

In Appendix A we show explicitly the action of $T$ in the case of 1PI diagrams with at most four crossings.

*3.4. Inclusion of tangencies in the transfer matrix*

In order to count knots and not knot diagrams, it was explained in section 2 that one must start with more general objects than standard diagrams: one must count curves with both self-intersections and tangencies which produce a double generating series $G(g_1, g_2)$. This can be easily included in the transfer matrix as follows.

Firstly, the space of states must be slightly extended to take into account the fact that we have a double generating series. Typically a state must contain the information concerning the number of previous tangencies in the diagram. Therefore the new space of states will be the tensor product of the old space of states and of the space of polynomials in a variable $x$ which can be defined as $x = g_2/g_1$.

Secondly, the transfer matrix itself must be modified to allow for the creation of such tangencies: the new transfer matrix $\tilde{T}$ is of the form

$$\tilde{T}(x) = T + xT' \tag{3.2}$$

where $T$ is the old transfer matrix defined by the transformations of Fig. 7, and $T'$ is the additional transformation described on Fig. 9 (plus, in each case, an arbitrary number of simplifications of the type of Fig. 8).

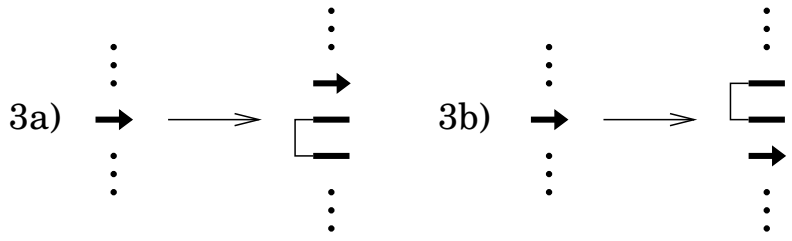


**Fig. 9:** The transformations that generate tangencies.

We can finally write the following formal expression for $G(g_1, g_2)$:

$$G(g_1, g_2) = \langle 0| \frac{1}{1 - g_1 \tilde{T}^2(x)} |0\rangle \tag{3.3}$$

where it is recalled that $x = g_2/g_1$.

## 4. Practical details

Having described the principles underlying the transfer matrix algorithm we now turn to a number of important practical details concerning its implementation. Section 5.1 describes the data structures needed to encode the states and their corresponding weights. Several remarks on the algorithm will be made in section 5.2, and in section 5.3 we explain a number of different implementations that we have made, in the aim of obtaining a reasonable balance between the time and memory needs of the algorithm.

### *4.1. Data structures*

The state space, formulated in terms of left and right arches and the active line, has been described in section 3.2. Ideally, in order to obtain a highly efficient transfer matrix algorithm, one would like to introduce a *ranking* among the states. By this we mean a bijective mapping from the $N$ different states to the set of integers $\{1, 2, \ldots, N\}$. With a ranking at hand, the integer representation can be used to label the entries of the transfer matrix, and the arch representation is then used to produce the descendants of a given state, as described in section 3.3. In a previous publication, one of the authors has shown how to obtain this goal in the case of meanders [3]; however, due to the very complicated interplay between left and right arches we have not been able to make similar progress in the case of knots.

Fortunately a simpler, and almost as efficient, alternative is available. Suppose that to each state $i$ we can assign a unique integer $k_i \in \mathbb{Z}_+$, and devise a function $f : \mathbb{Z}_+ \to \{0, 1, 2, \ldots, P-1\}$ that distributes the set of $k_i$'s more-or-less uniformly on the set $\{0, 1, 2, \ldots, P-1\}$. By inserting the states $i$ into an array of noded lists indexed by $f(k_i)$, we can retrieve a given state $k$ in a time proportional to the mean length of one of the pointer lists, $t \propto N/P$. This is a standard technique known as *hashing* [16]; the integer $k_i$ and the function $f$ are known respectively as the hash key and the hash function.

In the case at hand, a key $k_i$ can be defined by representing each arch state as a base-four number. Specifically, we read a configuration of left and right arches from top to

bottom, associating the digit 1 (resp. 0) with the opening (resp. closing) of a left arch, and 3 (resp. 2) with the opening (resp. closing) of a right arch. The 1s and the 0s (resp. the 3s and the 2s) thus form two interlaced Dyck words [15]. For the computation at order $p$ crossings there are at most $p$ arches, and the resulting key is at most $4^p$; we also need a few extra bits to specify the position of the active line. The hash function is simply $f(k) = k \bmod P$, where $P$ is a large prime which we choose such that $N/P \sim 10$.

For each state in the hash table, we store its key and its weight. The weight is an integer, but since the number of knots with $p$ intersections grows exponentially with $p$ the weights of the largest knots considered in this work cause overflow in a standard 32-bit integer arithmetic. Instead of wasting memory storing double-precision integers, we took advantage of modular arithmetic [17]. This means that the largest computations were done modulo various coprime numbers (typically $2^{32}$ and $2^{32} - 1$), and the full result was retrieved from the Chinese remainder theorem.

### *4.2. Algorithmic details*

It is possible to perform a number of reductions on the state space. Although these do not affect the correctness of the algorithm, they are nevertheless important to implement since they reduce the number of intermediate states needed in the transfer process, and thus enables us to go to larger system sizes.

After each of the two transformations shown on Fig. 6, the resulting arch state can be simplified using the reductions given in Fig. 7. The idea is to associate each inequivalent "screened" state with a unique configuration of right arches. Reductions 1) and 2) consists in sliding an exterior right arch over an adjacent interior arch, and 3) consists in removing right arches that do not screen any ingoing left line. In the algorithm, these simplifications are performed recursively until no further reduction is possible. The resulting state is then unique.

It may happen that after reduction a state is forbidden in the sense of Fig. 5 a) or b). A first example of this occurs at order $p = 4$, and is shown in Appendix A. Before inserting a descendant state in the hash table we therefore examine whether each right arch contains an even number of lines (including the active line), of which at least one is connected to the exterior.

A final algorithmic detail concerns the possibility of removing tadpole insertions in the knot diagrams. A tadpole is generated if and only if a type 1) transformation is immediately followed by a type 2) transformation in which the active line connects onto

an *adjacent* line (see Fig. 6). We can therefore forbid tadpoles if each state encompasses an extra sign signalling whether the previous transformation was of type 1). Superficially this would appear to double the number of states needed, but in fact this is not so, since an important number of states are only produced when tadpoles are allowed. In practice we found that the number of signed states without tadpoles, and the number of states with tadpoles only differ by a few percent. Of course, eliminating tadpoles directly in the algorithm carries no intrinsic interest, since it is a trivial matter to do so afterwards by manipulating the generating functions. However, since the number of tadpole diagrams is, very roughly, found to be the square of the number of diagrams without tadpoles, including tadpoles would mean that we would have to carry out twice as many runs in order to retrieve the full result from the Chinese remainder theorem. For this reason we opted for the algorithm without tadpoles.

### *4.3. Implementations*

Even though our transfer matrix method is much more efficient than a direct enumeration of the knot diagrams, it suffers from the drawback that the dimension of the state space, and thus the memory needs, grow exponentially with $p$. For a fixed size $p$, the evolution of the memory dynamically allocated by the hash table as a function of the discrete "time" steps is shown in Fig. 10. Near the beginning the number of states grows exponentially, reaches a maximum after roughly $3p/2$ steps, and then decreases exponentially towards the end. For practical reasons we only had about one gigabyte of memory available for our computations, and this turns out to be a more severe limitation for the obtainable system size than the computation time available. We have therefore experimented with several different implementations that limit the memory needs at the expense of using more time.

The most successful of these consist in, roughly speaking, using the transfer matrix approach until the available memory is exhausted. We then switch to a direct enumeration, which is carried out for a fixed number of steps, until the number of states needed by the transfer matrix approach has decreased to a level that again fits into the available memory. On Fig. 10 this could be represented by cutting the maximum of the memory profile by a horizontal line segment, representing the process of direct enumeration. The latter is based on the same recursive principle as the one defining the transfer process, but since states generated in intermediate time steps are not inserted into the hash table the process allocates no further memory.

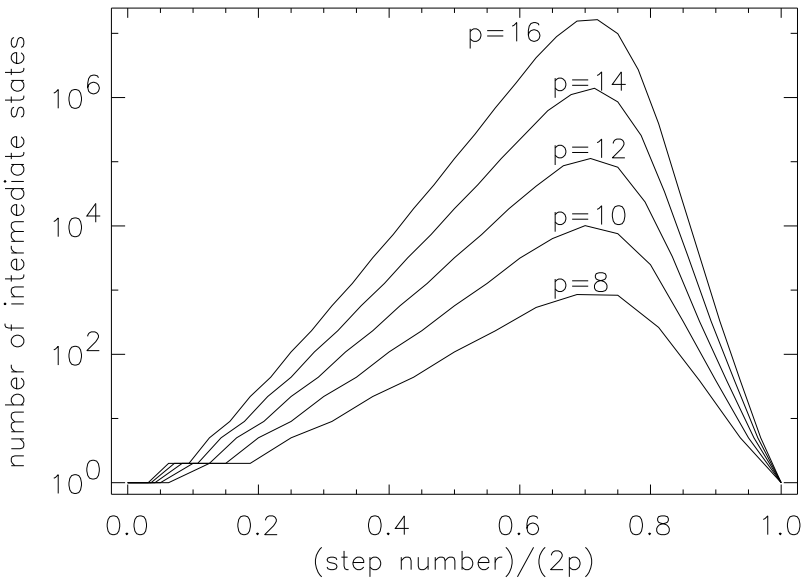


**Fig. 10:** The evolution of the number of intermediate states, as a function of 'time'. The different curves pertain to the enumeration of diagrams with $p = 8, 10, 12, 14$ and 16 crossings, no tangencies, and tadpoles included.

## 5. Results

In Table 1 we display the coefficients of the generating functions ($G$, $\Sigma_1$, $\Sigma_2$) up to order $p = 22$. We recall that these functions represent, respectively, the total number of knot diagrams with $p$ crossings and two outgoing strings, and the subsets of 1PI and 2PI diagrams.

The first 10 terms of $G$ have already been reported by Gusein-Zade and Duzhin [18], who called the corresponding diagrams 'long curves'. The algorithm used by these authors was however based on direct enumeration, and thus did not enjoy the advantages of the transfer matrix approach. Namely, in the latter, a multitude of diagrams can correspond to the same intermediate state at a given time step, and is therefore counted "simultaneously". The difference between the two approaches can readily be appreciated by comparing the number of diagrams (Table 1) with the number of intermediate states (Figure 10).

In particular, having available more terms of the generating functions enables us to examine the asymptotic behavior of the number of diagrams. Calling $a_p$ the coefficients of $G(g)$, as in Eq. (2.1), a first rough estimate yields $a_p \sim \mu^p$ with $\mu \simeq 11.4 \pm 0.1$. This corresponds to a singularity of the function $G(g)$ at $g_c = 1/\mu$. However, from the point of view of the underlying field theory it is the subleading corrections to the dominant

| $p$ | $G$ | $\Sigma_1$ | $\Sigma_2$ |
|---|---|---|---|
| 0 | 1 | 0 | 0 |
| 1 | 2 | 2 | 2 |
| 2 | 8 | 4 | 0 |
| 3 | 42 | 18 | 2 |
| 4 | 260 | 108 | 4 |
| 5 | 1796 | 748 | 12 |
| 6 | 13396 | 5648 | 60 |
| 7 | 105706 | 45234 | 226 |
| 8 | 870772 | 378300 | 1076 |
| 9 | 7420836 | 3271204 | 5156 |
| 10 | 65004584 | 29049824 | 24984 |
| 11 | 582521748 | 263656356 | 128548 |
| 12 | 5320936416 | 2436827328 | 663040 |
| 13 | 49402687392 | 22871937208 | 3514968 |
| 14 | 465189744448 | 217536523260 | 18918792 |
| 15 | 4434492302426 | 2092958991474 | 103123906 |
| 16 | 42731740126228 | 20341256951692 | 569877652 |
| 17 | 415736458808868 | 199471121367508 | 3180066004 |
| 18 | 4079436831493480 | 1971730006936240 | 17921451960 |
| 19 | 40338413922226212 | 19630246152650228 | 101842206548 |
| 20 | 401652846850965808 | 196703992506546352 | 583109887600 |
| 21 | 4024556509468827432 | 1982670344984596872 | 3361640932872 |
| 22 | 40558226664529024000 | 20091545428174220376 | 19499226668816 |

**Tab. 1:** Table of the total number of two-legged diagrams, as well as the subsets of 1PI and 2PI diagrams.

exponential behavior that are of paramount interest, the connective constant $\mu$ being non-universal. Based on a random-matrix model description of alternating knots as the $n \to 0$ limit of a generalized $O(n)$ symmetric action [5,6,7], one of us has conjectured that the detailed asymptotic behavior reads

$$a_p \approx \mu^p p^{-\alpha}, \tag{5.1}$$

with $\alpha = 3$. This value of $\alpha$ corresponds to a string susceptibility exponent $\Gamma \equiv 2 - \alpha = -1$ characteristic of the coupling of a conformal field theory with central charge $c = -2$ to two-dimensional quantum gravity, via the celebrated KPZ formula [19].

With the current data, it is difficult to estimate $\alpha$ in Eq. (5.1) without any knowledge of the subleading corrections. Indeed, a direct fit gives $\alpha \approx 2.76$ but usual convergence acceleration methods do not confirm this result. However, if we fit our data with $a_p = \mu^p p^{-\alpha}(a \log p + b + o(1))$ (the presence of logarithmic corrections being justified by possible marginally irrelevant operators in a $c = -2$ theory), the result is in good agreement with the conjecture:

$$\mu = 11.416 \pm 0.005 \quad \alpha = 2.97 \pm 0.06 \quad a = 0.04 \pm 0.02 \quad b = 0.1 \pm 0.03 \qquad (5.2)$$

In the case of $\Sigma_2$, the approach to the asymptotic regime appears to be less regular. This is probably due to another singularity of the function $G(g)$ around $g_{c2} \approx -0.3$ which causes oscillations that are very subdominant in $G(g)$ but less so in $\Sigma_2(g)$. However, the leading exponential behavior $\sim \mu_2^p$ can be readily extracted without analyzing numerically the coefficients of $\Sigma_2$, by using the following simple identity: $\mu_2 = \mu / G(g_c)^2$. Assuming the expansion above and using the fit (5.2), We find:

$$\mu_2 = 6.613 \pm 0.008$$

| $p_1$ \ $p_2$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 10 | 70 | 588 | 5544 | 56628 |
| 1 | 2 | 20 | 210 | 2352 | 27720 | 339768 | 4294290 |
| 2 | 8 | 174 | 2992 | 47820 | 742296 | 11376554 | 173401952 |
| 3 | 42 | 1504 | 37100 | 784672 | 15294006 | 283730240 | 5095814988 |
| 4 | 260 | 13300 | 433620 | 11515714 | 271846056 | 5947557516 | 123429078160 |
| 5 | 1796 | 120744 | 4928798 | 158295072 | 4403552940 | 111289501120 | 2626033507768 |
| 6 | 13396 | 1122198 | 55237824 | 2086803540 | 66981001600 | 1923315870960 | 50921564862176 |
| 7 | 105706 | 10638464 | 614451348 | 26737722400 | 973914284112 | 31351179461568 | 921163652161792 |
| 8 | 870772 | 102541428 | 6807871480 | 335676172480 | 13691869089168 | 488718870505840 | ... |
| 9 | 7420836 | 1002305040 | 75275707584 | 4150940757440 | 187548130544528 | ... | |
| 10 | 65004584 | 9914663308 | 831595048320 | 50739269522864 | ... | | |
| 11 | 582521748 | 99085515840 | 9185000522880 | 614607881444256 | ... | | |
| 12 | 5320936416 | 999104604784 | 101470031154352 | 7390867767651290 | ... | | |
| 13 | 49402687392 | 10153152363648 | 1121497913694390 | ... | | | |
| 14 | 465189744448 | 103892246982390 | 12403035430713344 | ... | | | |
| 15 | 4434492302426 | 1069610792999424 | 137266650274351716 | ... | | | |
| 16 | 42731740126228 | 11072575568623300 | ... | | | | |
| 17 | 415736458808868 | 115189593628215600 | ... | | | | |
| 18 | 4079436831493480 | 1203690675390892316 | ... | | | | |
| 19 | 40338413922226212 | ... | | | | | |
| 20 | 401652846850965808 | ... | | | | | |
| 21 | 4024556509468827432 | ... | | | | | |
| 22 | 40558226664529024000 | ... | | | | | |

**Tab. 2:** Table of the number of two-legged diagrams with $p_1$ self-intersections and $p_2$ tangencies.

We now turn to the inclusion of the flype equivalence. As described in Section 3.4 this can be done by enumerating also diagrams with tangencies. A power-counting argument reveals that in order to accomodate the flype equivalence at order $p$, we need to know the number of diagrams at order $p_1$ with (roughly) at most $(p_2)_{\max} \equiv \lfloor (p-p_1)/3 \rfloor$ tangencies, for all $p_1 = 0, 1, \ldots, p$. These data are shown up to order $p = 20$ in Table 2.

The contents of the first column ($p_2 = 0$) is of course just the coefficients of $G$, cf. Table 1. The first line ($p_1 = 0$) gives the number of two-legged diagrams with $p_2$ tangencies and no self-intersections,

$$a_{0,p} = \frac{(2p)!(2p+2)!}{p!(p+1)!^2(p+2)!} \tag{5.3}$$

This formula is a corollary of the exact solution of the standard O($m \to 0$) model on random tetravalent graphs. It can also be shown in a straightforward way. First, represent each tangency by a dotted line, as in Figure 11.a. In this way, the problem becomes that of rooted Hamiltonian circuits on a random trivalent graphs [20]. Next, straighten out the full line, as in Figure 11.b. The dotted lines now form two independent Catalan arch configurations, one above and one below the full line. Clearly, the number of such configurations is

$$\sum_{j=0}^{p} \binom{2p}{2j} c_{p-j} c_j$$

where $c_k = \frac{(2k)!}{k!(k+1)!}$ are the Catalan numbers. The result (5.3) follows immediately.

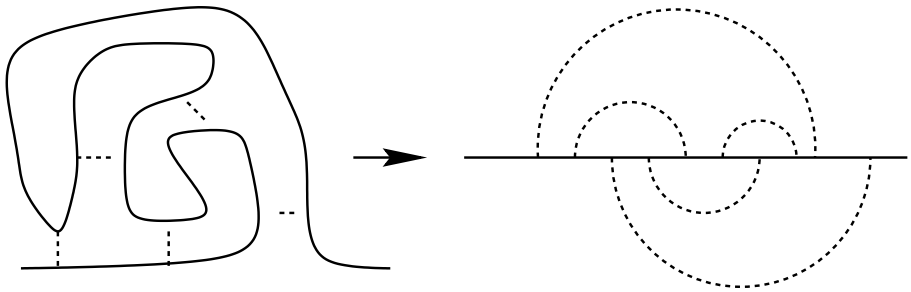

**Fig. 11:** Counting of pure tangency diagrams. a) Transform each tangency into a pair of trivalent vertices. b) Stretch the full line.

In a similar fashion, we can give an explicit formula for the second line:

$$a_{1,p-1} = p\, a_{0,p} = \frac{(2p)!(2p+2)!}{(p-1)!(p+1)!^2(p+2)!}$$

Indeed, the diagrams with one self-intersection and $p-1$ tangencies are obtained from diagrams with $p$ tangencies by replacing one tangency with a crossing.

| $p$ | $\Gamma_1$ | $\Gamma_2$ |
|---|---|---|
| 1 | 1 | 0 |
| 2 | 0 | 1 |
| 3 | 2 | 1 |
| 4 | 2 | 3 |
| 5 | 6 | 9 |
| 6 | 30 | 21 |
| 7 | 62 | 101 |
| 8 | 382 | 346 |
| 9 | 1338 | 1576 |
| 10 | 6216 | 7040 |
| 11 | 29656 | 31556 |
| 12 | 131316 | 153916 |
| 13 | 669138 | 724758 |
| 14 | 3156172 | 3610768 |
| 15 | 16032652 | 17853814 |
| 16 | 80104192 | 90220450 |
| 17 | 408448012 | 460221672 |
| 18 | 2105616701 | 2365627740 |
| 19 | ... | 12300901598 |

**Tab. 3:** Table of the number of prime alternating tangles with two connected components.

The results for the the number of prime alternating tangles with two connected components can now be found from the procedure outlined in Section 2; see Table 3. The reader is reminded that the total number of tangles is given by $\Gamma_1 + 2\Gamma_2$.

The first 8 orders were previously given in [7]. The number of tangles seem to approach their asymptotic behavior in a less regular fashion than the objects discussed above. In particular there seems to be a strong dependance on the parity of $p$. Even the leading term in the large $p$ limit, of the form $\tilde{\mu}_2^p$, is hard to isolate, with $\tilde{\mu}_2$ roughly given by $6.0 \pm 0.1$; some serious numerical analysis is required to determine it more accurately. Note that this leading behavior should be the same for prime alternating knots, with the same constant $\tilde{\mu}_2$. Similary, the critical exponent of knots should be one plus the exponent of tangles. Although we have not been able to extract reliable values of the exponent, physical insight suggest that these exponents are most likely to the same as those of diagrams (conjecturally, $\alpha = 3$). It is therefore very likely that the conjecture made in [6] is valid, though we have no definite evidence at the moment.

In a future publication [21] we shall show how to generalize our transfer matrix approach to allow for the enumeration of connected knot diagrams with an *arbitrary* fixed number of components. As a first application, this will allow us to enumerate alternating *links*, and to extend the generating functions given in [7] by several orders. Another interesting goal that we are currently pursuing is the enumeration of multi-component meander diagrams [3]. There are many other applications related to the possibility of counting planar Feynman diagrams.

## Appendix A. Knot diagrams up to 4 crossings.

As an illustration we show the first 8 iterations of the transfer matrix. We restrict ourselves to states which generate 1PI diagrams and with at most 4 crossings. Reading off the weight of the vacuum state (containing only the active line) after step $2p$, we deduce the number of diagrams with $p$ crossings. However, this state is not allowed to evolve in subsequent steps, since otherwise one-particle reducible diagrams would be generated. We have *not* excluded tadpoles in this example.

Note also that when connecting the active line of the third diagram at step 6 to the uppermost line by means of a right arch, we generate (after reduction of the two right arches) a forbidden state of the type shown in Fig. 5 a), which is therefore not shown.

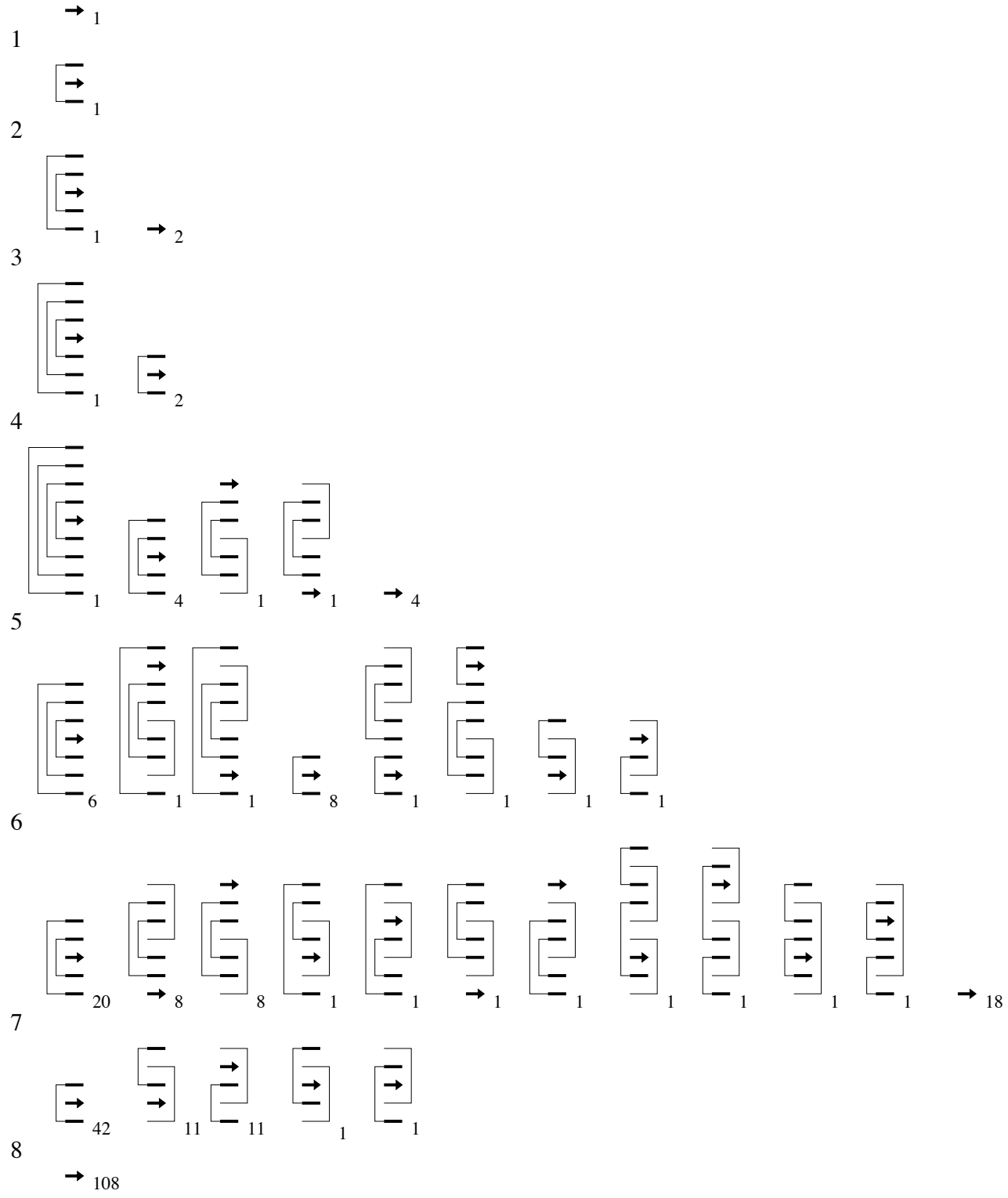


**Fig. 12:** The list of all intermediate steps for 1PI knot diagrams up to 4 crossings. The number in subscript is the weight of the state.

## References


[1] P. Di Francesco, E. Guitter and C. Kristjansen, *Integrable 2D Lorentzian Gravity and Random Walks*, *Nucl. Phys.* **B 567** (2000) 515–553 (preprint `hep-th/9907084`).

[2] I. Jensen, *Enumerations of Plane Meanders* (preprint `cond-mat/9910313`); *A Transfer Matrix Approach to the Enumeration of Plane Meanders*, *J. Phys. A*, to appear (preprint `cond-mat/0008178`).

[3] P. Di Francesco, E. Guitter and J.L. Jacobsen, *Exact Meander Asymptotics: a Numerical Check*, *Nucl. Phys.* **B 580** (2000) 757–795 (preprint `cond-mat/0003008`).

[4] P. Di Francesco, E. Guitter and J.L. Jacobsen, work in progress.

[5] P. Zinn-Justin and J.-B. Zuber, *Matrix Integrals and the Counting of Tangles and Links*, to appear in the proceedings of the 11th International Conference on Formal Power Series and Algebraic Combinatorics, Barcelona June 1999 (preprint `math-ph/9904019`).

[6] P. Zinn-Justin, *Some Matrix Integrals related to Knots and Links*, proceedings of the 1999 semester of the MSRI "Random Matrices and their Applications", MSRI Publications Vol. 40 (2001) (preprint `math-ph/9910010`).

[7] P. Zinn-Justin and J.-B. Zuber, *On the Counting of Colored Tangles*, *Journal of Knot Theory and its Ramifications* 9 (2000) 1127–1141 (preprint `math-ph/0002020`).

[8] S. Nechaev, *Statistics of knots and entangled random walks*, lectures at Les Houches 1998 summer school (preprint `cond-mat/9812205`).

[9] L.H. Kauffman, *Knots and physics*, World Scientific Pub Co (1994).

[10] J. Hoste, M. Thistlethwaite and J. Weeks, *The First 1,701,936 Knots*, *The Mathematical Intelligencer* **20** (1998) 33–48.

[11] C. Sundberg and M. Thistlethwaite, *The rate of Growth of the Number of Prime Alternating Links and Tangles*, *Pac. J. Math.* **182** (1998) 329–358.

[12] P. Di Francesco, P. Ginsparg and J. Zinn-Justin, *2D Gravity and Random Matrices*, *Phys. Rep.* **254** (1995) 1–133.

[13] W.W. Menasco and M.B. Thistlethwaite, *The Tait Flyping Conjecture*, *Bull. Amer. Math. Soc.* **25** (1991) 403–412; *The Classification of Alternating Links*, *Ann. Math.* **138** (1993) 113–171.

[14] P. Zinn-Justin, *The General O(n) Quartic Matrix Model and its application to Counting Tangles and Links* (preprint `math-ph/0106005`).

[15] M.-P. Delest and G. Viennot, *Theor. Comp. Sci.* **34** (1984) 169.

[16] R. Sedgewick, *Algorithms in C* (Addison-Wesley, 1990).

[17] D. E. Knuth, *The Art of Computer Programming*, vol 2: *Seminumerical Algorithms* (Addison-Wesley, 1969).

[18] S. M. Gusein-Zade, *Adv. Soviet Math.* **21**, 189–198 (1994); S. M. Gusein-Zade and F. S. Duzhin, *On the number of topological types of plane curves*, *Uspekhi Math. Nauk.* **53**, 197–198 (1998) [English translation in *Russian Math. Surveys* **53**, 626–627 (1998)].

[19] V. G. Knizhnik, A. M. Polyakov and A. B. Zamolodchikov, *Fractal structure of 2D quantum gravity*, *Mod. Phys. Lett. A* **3**, 819–826 (1988); F. David, *Conformal field theories coupled to 2D gravity in the conformal gauge*, *Mod. Phys. Lett. A* **3**, 1651–1656 (1988); J. Distler and H. Kawai, *Conformal field theory and 2D quantum gravity*, *Nucl. Phys.* **B 321**, 509 (1989).

[20] W. T. Tutte, *A census of Hamiltonian polygons*, *Canad. J. Math.* **14** 402-417 (1962).

[21] J. L. Jacobsen and P. Zinn-Justin, *A Transfer Matrix approach to the Enumeration of Colored Links* (preprint `math-ph/0104009`).