Department of Computer Science,
University of Otago

UNIVERSITY of OTAGO

Te Whare Wānanga o Otāgo

Technical Report OUCS-2004-08

Bounds For The Growth Rate Of Meander Numbers

Authors:
M . H. Albert
Department of Computer Science, University of Otago
M. S. Paterson
Department of Computer Science, University of Warwick

Status: presented at FPSAC 2004 (Vancouver)

Department of Computer Science,
University of Otago, PO Box 56, Dunedin, Otago, New Zealand

http://www.cs.otago.ac.nz/trseries/

Formal Power Series and Algebraic Combinatorics
Séries Formelles et Combinatoire Algébrique
Vancouver 2004

Bounds For The Growth Rate Of Meander Numbers

M. H. Albert and M. S. Paterson

Abstract. We provide improvements on the best currently known upper and lower bounds for the exponential growth rate of meanders. The method of proof for the upper bounds is to extend the Goulden-Jackson *cluster method*.

Limites au taux de croissance des nombres de méandres

Nous fournissons des améliorations aux meilleures bornes supérieures et inférieures actuellement connues pour le taux de croissance exponentiel des méandres. La méthode de preuve des bornes supérieures nécessite une extension de la méthode des "grappes" due à Goulden et Jackson.

1. Introduction

A *meander* of order n is a self-avoiding closed curve crossing a given line in the plane at $2n$ places, [**LZ93**]. Two meanders are equivalent if one can be transformed into the other by smooth deformations of the plane, which leave the line fixed (as a set). A number of authors have addressed the problem of exact and asymptotic enumeration of the number M_n of meanders of order n (see for instance [**FE02, Jen00**] and references therein). It is widely believed that an asymptotic formula

$$M_n \approx C M^n n^\alpha$$

applies, and some effort has been devoted to estimating the parameters M and α ([**DF00, DFGG00, DFGJ00, JG00**]). Broadly, these methods have relied on extrapolation from exact values of M_n, currently known for $n \leq 24$ (see [**JG00**]). A careful estimate, using differential approximants based on these values, yields [**JG00**] the approximate value

$$M \simeq 12.26287.$$

A presumed correspondence with certain field theories has yielded the amazing conjecture [**DFGG00**] that:

$$\alpha = \sqrt{29}(\sqrt{29} + \sqrt{5})/12 = 3.42013288\ldots\ .$$

Our, less ambitious, aim will be to provide rigorous upper and lower bounds on the exponential growth rate of M_n.

Consider the generating function:

$$M(t) = \sum_{n=0}^{\infty} M_n t^{2n}.$$

1991 *Mathematics Subject Classification.* Primary 05A15, 05A16; Secondary 57M99, 82B41.

Key words and phrases. meander, cluster method.

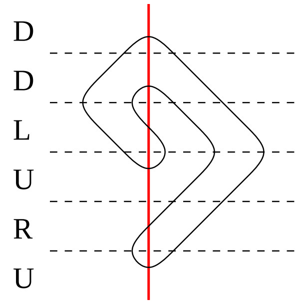

Figure 1. The meander $URULDD$.

It is easy to verify that $M_{a+b} \geq M_a M_b$ and so it is certainly the case that $M := \lim_{n\to\infty} M_n^{1/n}$ exists, and is the square of the reciprocal of the radius of convergence of this series. We will prove:

Theorem 1.1. *The following inequalities hold:*

$$11.380 \leq M \leq 12.901.$$

These bounds improve (on both sides) the best previous bounds due to Richard Stanley ($M > 10.0$) [1995, private communication] and Jim Reeds and Larry Shepp ($M \leq 13.002$) [1999, unpublished].

Our basic methodology is to represent meanders as a language over an alphabet consisting of four symbols. The bounds are then obtained by producing suitable sublanguages and superlanguages for which the growth rates can be computed explicitly. In principle our bounds could be improved by more detailed construction of these languages, and we include some indication in the final section of how much further progress might be possible by such means.

2. Definitions and notation

We begin by providing a combinatorial description of meanders which allows us to identify them with a language over a four letter alphabet. This interpretation is similar to the description of meanders by means of "configurations" in [**Jen00**].

Set the orientation of the line which the meander crosses as vertical. We allow a meander to evolve as we move upwards along the line. Each step in this evolution is marked by a place where the meander crosses the line, and we allow these crossings to be of four types: U where a new segment of the meander is created, D where two previous segments are merged into one (or as a final step the meander is completed), and R or L where a segment crosses the line from left to right, or right to left respectively. Figure 1 illustrates this encoding of meanders.

The *meander language*, $\mathcal{M}$, is the set of words in these four letters that represent meanders. It is immediately clear that distinct words in the meander language represent distinct meanders, and only slightly less clear that every meander is represented by a single word in the meander language.

We digress briefly to recapitulate some standard notation and terminology concerning words and languages. A *word* is simply a finite sequence of symbols from some alphabet Σ. This sequence may be empty, and the empty word is denoted ϵ. The set of all words over Σ is denoted Σ^* and can be identified with the free monoid over Σ by considering juxtaposition as the monoid operation. So, a word v is said to be a *factor* of a word w if $w = xvy$ for some words x and y. If we can take $x = \epsilon$ then we say that v is a *prefix* of w while if we can take $y = \epsilon$ then we say that v is a *suffix* of w. A *language* over Σ is simply a subset of Σ^*. The $()^*$ notation is extended to languages, or even words, so that X^* simply means the language which consists of all possible juxtapositions (including the empty one) of elements of X. The length of a word w, that is, the number of symbols in the sequence w, is denoted $|w|$. Hence M_n, the number of meanders with $2n$ crossings is simply

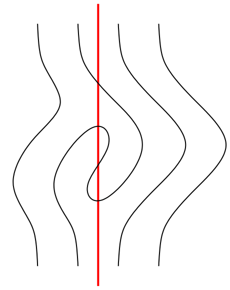

FIGURE 2. $URDL$ has no effect on the environment

the number of words in $\mathcal{M}$ of length $2n$ (since each symbol in a meander word accounts for a single crossing).

In our interpretation of meanders it makes sense to speak of the environment that exists as we scan prefixes of a word. This environment is simply the collection of segments in their appropriate order on either side of the line. Further, we adopt the convention that when two segments are merged, the newly merged segment is identified in the environment with the older of the two (in a meander the only time we will merge two segments of the same age is at the final D).

Sometimes it is useful to imagine that we have available an extended environment consisting initially of an infinite family of labelled and completely unmatched segments on either side of the line. This allows the effect of any word to be interpreted within this environment. For our purposes, words whose only effect is to shift some segments from one side of the line to the other are particularly significant. In Figure 2 we illustrate how the factor $URDL$ has no effect on the surrounding environment. In particular this means that if $w = uv$ is a meander, and if the environment following u contains a segment to the left of the line, then $uURDLv$ is also a meander. On the other hand, it is also clear that no meander (aside from UD) can have UD as a factor, and so neither can it have $UURDLD$ as a factor. From observations of the former kind we obtain sublanguages of $\mathcal{M}$ by building up words which must be meanders. From observations of the latter kind we obtain superlanguages of $\mathcal{M}$ by requiring words to avoid certain factors.

Throughout the remaining sections we identify languages over U, D, R, L with their generating function in the power series ring over U, D, R, L. Generally we work in this context to obtain relationships between (the generating functions of) various languages, and then specialize to a single variable t when we wish to obtain numerical estimates.

3. Shifts and lower bounds

Consider a state of the extended meander environment, such as might be achieved after executing some prefix p of a meander word. There are now various continuations which will have the same effect *on the environment* as R^k would for some k. Trivially any sequence of R's and L's which has k more R's than L's is such a continuation. However, it is also the case that URD has the same effect on the environment as R, and $UURRDD$ has the same effect as RR. Furthermore these constructions can be recursively combined and therefore:

$$U(UURRDD)LD$$

has the same effect as $URRLD$, hence as URD and finally as R.

DEFINITION 3.1. A *shift* is a word whose effect on the extended meander environment is the same as that of R^k or L^k for some non-negative integer k. The *displacement* of a shift is k in the former case, and $-k$ in the latter. A *jump* is a shift having no proper shift prefix[1]. A shift whose

[1] We apologize to the sensitive reader for using "shift" both as a noun and an adjective

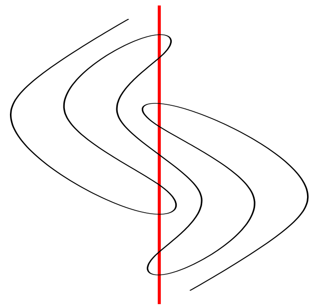

FIGURE 3. $URUL^3DRD$ is a jump of displacement -1.

only proper shift factors are in R^* or L^* is called *primitive*.

The simplest jumps are R and L. Next simplest are URD and ULD. A rather more complicated example is shown in Figure 3.

Every shift can be uniquely factored as a concatenation of jumps. In turn, every jump is created from some (uniquely determined) primitive shift by substitution of shifts for the blocks of R's and L's within the primitive shift. For example $UUURRDDLD$ is created from URD by substituting $UURRDD\,L$ (a shift of displacement 1 formed from a jump of displacement 2, and one of displacement -1) for R.

If $\mathcal{J}$ is the language of all jumps and $\mathcal{S}$ the language of all shifts, then of course

$$\mathcal{S} = \mathcal{J}^* = \frac{1}{1-\mathcal{J}}. \tag{3.1}$$

Introducing a new indexing variable x which commutes with the symbols of the language, and letting $\mathcal{J}_i$ (or $\mathcal{S}_i$) be the language of jumps (or shifts) of displacement i, we have slightly more generally that:

$$\sum_{i=-\infty}^{\infty} \mathcal{S}_i x^i = \frac{1}{1-\sum_{i=-\infty}^{\infty} \mathcal{J}_i x^i}.$$

Suppose that J is some primitive jump. Then the set of all jumps with primitive form J is obtained by replacing each (possibly null) block of L's or R's between consecutive occurrences of U or D by $\mathcal{S}_k$ where k is the displacement of the block. Denote the result of this replacement by $J^{\mathcal{S}}$. Then $\mathcal{J}_i$ is the sum over primitive jumps J of displacement i of the terms $J^{\mathcal{S}}$.

Let $s_i(t)$ be the generating function obtained from $\mathcal{S}_i$ by replacing all of U, D, L, and R by t. Since

$$t^{2i}s_0 < t^i s_i < s_0$$

all of the functions s_i have the same radius of convergence.

PROPOSITION 3.2. The radius of convergence of s_0 is not greater than that for the meander language.

PROOF. The result follows from the observation that $M(t) \le t^2 s_0$, since every meander is of the form USD where S is a shift of displacement 0. □

It seems clear that among all the shift words of length $2n$ only a vanishingly small proportion contain a prefix with a difference of at least $n^{3/4}$ between the numbers of R's and L's (here $n^{3/4}$ is an arbitrary value – larger than $\sqrt{n}$, by correspondence with a 1-dimensional drunkard's walk). Any such shift could then be built into a meander of $2n(1+o(1))$ crossings. This would establish that the shift language and the meander language have the same radius of convergence. The proof of this result is too involved to present here, but will appear in the full paper.

For our immediate computational purposes though it is superfluous as our shifts will be built up recursively from a set of primitive shifts whose excursions to the left or the right are of bounded size. Since we work with a symmetric (in R and L) set of shifts, the argument above then applies correctly to this situation. This observation is explained further at the end of subsection 6.1.

4. The cluster method

The *cluster method* is a method of enumerating words with a given finite set of forbidden factors. It was introduced in this form in [**GJ79**] and is also discussed in [**GJ83**]. Extensions of the cluster method are given in [**NZ99**] to handle certain cases where the forbidden set of factors is infinite. We need to supply a similar extension in an even more general setting.

Let Σ be an alphabet, and $\mathcal{B}$ a subset of Σ^+ (the non-empty words over Σ). We are concerned with the language consisting of those words which have no factor from $\mathcal{B}$, the *$\mathcal{B}$-factor-free words*, that is the complement in Σ^* of $\Sigma^*\mathcal{B}\Sigma^*$. If b is a factor of c and b does not occur as a factor of some word w, then of course neither does c. So, for any $\mathcal{B}$, the $\mathcal{B}$-factor-free words are the same as the $\mathcal{B}'$-factor-free words, where $\mathcal{B}'$ consists of the minimal elements of $\mathcal{B}$ in the factor ordering. Therefore we assume throughout that no word $b \in \mathcal{B}$ is a proper factor of any other word in $\mathcal{B}$.

Define the set of *overlaps*, $Ov(\mathcal{B})$ to be the collection of all triples (b, w, c) such that $b, c \in \mathcal{B}$, $w \in \Sigma^+$, such that $b \neq c$ and for some b^l and c^r, $b = b^l w$ and $c = wc^r$. Note that, owing to the assumption above, neither b^l nor c^r can be the empty word. The system of equations:

$$v_b = b - \sum \left\{ b^l v_c \,:\, (b, w, c) \in Ov(\mathcal{B}),\, b = b^l w \right\} \qquad \text{for } b \in \mathcal{B} \tag{4.1}$$

has a unique solution in the power series ring $\mathbb{Q}[[\Sigma]]$.

The following theorem generalises (to the case of infinite $\mathcal{B}$ and non-commuting variables) a specialisation (to the case of forbidding all occurrences of $\mathcal{B}$ rather than determining the type of the occurrences of $\mathcal{B}$ in a word) of Theorem 2.86 in [**GJ83**], often called the Goulden-Jackson cluster method. In [**Zei02**] an informal treatment of an equivalent method can also be found. A full generalisation of the original theorem could be obtained by adding tagging variables y_b (commuting with each other and with Σ) to the system (4.1), but the version below is adequate for our purposes.

THEOREM 4.1. *The generating function over $\mathbb{Q}[[\Sigma]]$ of $\Sigma^* \setminus \Sigma^*\mathcal{B}\Sigma^*$ is:*

$$\left(1 - \Sigma + \sum_{b \in \mathcal{B}} v_b\right)^{-1}$$

where $\{v_b \,:\, b \in \mathcal{B}\}$ are defined by (4.1).

PROOF. The proof of this theorem can be read off from the proof of the theorem cited above. However, at least in this form, it is really simply a restatement of the principle of inclusion/exclusion. Define a $\mathcal{B}$-marking of a word w in Σ^* to be a specific identification of certain factors of w which belong to $\mathcal{B}$ (not necessarily any or all such factors). If we assign the value $(-1)^k w$ to each $\mathcal{B}$-marking of w in which k factors from $\mathcal{B}$ are marked then the sum over all the $\mathcal{B}$ markings of a word w will be 0 if w contains a $\mathcal{B}$-factor, and w if it does not. By considering the expression above as a geometric series it is easy to see that the coefficient of w is exactly this sum over $\mathcal{B}$-markings of w, and hence the expression represents the generating function of $\mathcal{B}$-factor-free words. □

As remarked in [**Zei02**], in the case of infinite structureless $\mathcal{B}$ this does not give an equation for the generating function in any usual sense. However, in our application below, the language $\mathcal{B}$ will carry sufficient structure that we can make effective use of Theorem 4.1.

Note that if we turn to the ordinary generating function for the language of $\mathcal{B}$-factor-free words, then its radius of convergence is the smallest positive root of the equation:

$$1 - |\Sigma| t + \sum_{b \in \mathcal{B}} v_b(t)$$

where we also have:

$$v_b(t) = t^{|b|} - \sum \left\{ t^{|b^l|} v_c(t) \,:\, (b, w, c) \in Ov(\mathcal{B}),\, b = b^l w \right\} \qquad \text{for } b \in \mathcal{B}.$$

REMARK 4.2. In general it is not the case that the system of linear equations defined above has the required property to allow an iterative solution after specializing to a single variable, even if the value chosen for the variable lies inside the radius of convergence of the series which form its solution in $\mathbb{Q}[[t]]$. This fails, for example, in the case $\mathcal{B} = \{aaa, aba\}$ over the alphabet $\{a, b\}$.

5. Submeanders and upper bounds

We now apply the results of the preceding section in order to obtain upper bounds on the exponential growth rate of the meander language $\mathcal{M}$. Ideally, the language of forbidden words which we would like to consider consists of all words which define some closed loop, or submeander. That is, a word is forbidden if it is of the form $U \cdots D$ where the final symbol closes off the pair of segments created by the initial one. Let $\mathcal{B}$ be the language of such words. If an element of $\mathcal{B}$ occurs as a *proper* factor of a word m then $m \notin \mathcal{M}$. It is clear though that the growth rates for the languages of $\mathcal{B}$-factor-free words and proper $\mathcal{B}$-factor-free words are the same, so we do not need to worry about that distinction. Henceforth we fix the alphabet $\Sigma = \{U, D, R, L\}$.

The shortest word in $\mathcal{B}$ is UD. However, this single word is really a representative of a much wider family of forbidden words. Among these are $URLD$, and $UURDLD$. Generally if S is any shift of displacement 0, then USD is a forbidden word. It is worth noting that there is no requirement that the words in $\mathcal{B}$ be balanced with respect to U and D. For example, the word $URULLD$ is in $\mathcal{B}$, since the final D forms a submeander with the original U, and so if this word occurs as a factor of some longer word w then w cannot represent a meander.

There is an equivalence relation defined on words by taking the transitive closure of the relation obtained by allowing the replacement of a shift, by any other shift of the same displacement. Each equivalence class of this relation contains a representative with the property that any maximal shift factor lies in L^* or R^*. Let us call these representatives the *standard representatives* of their classes. Note also that $\mathcal{B}$ is closed under this equivalence relation.

LEMMA 5.1. *Let a word w be given. Its standard representative is obtained by replacing the maximal shift factors of w by blocks of L's or R's of the same displacement.*

PROOF. This follows immediately from the observation that two shift factors of w cannot overlap unless their overlap is also a shift. This is because a proper suffix of a shift which is not a shift and begins with U contains more D's than U's, and no prefix of a shift word has this property. Since shifts are closed under concatentation, the maximal shift factors of w are disjoint and properly separated, and so the standard representative is obtained in the manner described. □

Using this result we obtain:

PROPOSITION 5.2. Let $b, c \in \mathcal{B}$ have an overlap w. Then the standard representatives of b and c also have an overlap, which is the image of w under the replacement described in Lemma 5.1.

PROOF. The word w has the form UuD. Moreover in b the terminal D closes the segments formed by the initial U of b so, interpreted in isolation, it does not close any segment created within u and so cannot be part of any shift factor of w. The same idea applies to the observation that the initial U of c is matched by its final D and so shows that the original U of w can also not be part of any shift factor of w. So the shift factors of b and c which occur within w, occur within u. Therefore the reduction of Lemma 5.1 affects w in the same way in both b and c. □

Let $\mathcal{B}_{\mathsf{rep}}$ be the sublanguage of $\mathcal{B}$ consisting of the standard representatives of the elements of $\mathcal{B}$. For any word w let $\bar{w}$ be the generating function of its equivalence class. Now consider a modification of the system of equations (4.1)

$$x_b = \bar{b} - \sum \left\{ \bar{b^l} x_c \,:\, (b, w, c) \in Ov(\mathcal{B}_{\mathsf{rep}}),\, b = b^l w \right\} \qquad \text{for } b \in \mathcal{B}_{\mathsf{rep}}. \tag{5.1}$$

Then, it follows directly from Proposition 5.2 that:

$$\sum_{b \in \mathcal{B}} v_b = \sum_{b \in \mathcal{B}_{\mathsf{rep}}} x_b$$

(where v_b is defined by the system of equations (4.1)).

Thus we may use the latter form in computations arising from Theorem 4.1. For instance, we could use a finite subset of the original language $\mathcal{B}$, and also place some restrictions on the shift words used in constructing $\bar{w}$ from w.

For example, take as forbidden language $\mathcal{B}_0$, the single forbidden word UD, and its expansions USD where $S \in \{R, L\}^*$ has displacement 0. Then the generating function for $\mathcal{B}_0$-factor-free words is:

$$\frac{1}{1 - 4t + \frac{t^2}{\sqrt{1-4t^2}}}.$$

The radius of convergence of this generating function is the smallest positive solution of

$$65t^4 - 32t^3 - 12t^2 + 8t - 1 = 0$$

whose approximate value is 0.272054. Since $\mathcal{B}_0$ represents a subset of the actual words forbidden to appear as factors in a meander word, this gives an upper bound of 13.5111 on M.

In the next section we will describe in greater detail how these results can be used to provide bounds for M in situations where we cannot analytically solve the equations for the radius of convergence.

6. Computational methodology

In this section we give an overview of the computational methods used to evaluate lower and upper bounds on M.

6.1. Lower bounds. In computing lower bounds on the exponential growth rate for the meander generating function, we attempt to construct a generating function based on a subset of the set of shifts, built up from a subset of the primitive jumps. Generally, we make use of all the primitive jumps containing at most some preset number of symbols. These are constructed by simulating the extended meander environment and carrying out a depth-first search. The only extra information which must be maintained is a record of the new segments present when each U occurs. This must then be compared to the D which eliminates the segment created by the U in order to ensure that the only shift factors are in L^* and R^*.

The results quoted below are for primitive jumps containing a maximum of 24 symbols. There are 875,938 such primitive jumps with non-negative displacement. On the other hand, there are only 25,264 of length at most 20, and only the following 13 of length at most 10:

URD, $UURRDD$, $UULLDRRD$, $UURRDLLD$, $ULLURRDD$,
$URRULLDD$, $UUURRRDDD$, $ULURRRDLD$, $ULUURRDRDD$,
$URRULLDRRD$, $UURURRDDLD$, $UULURRRDDD$, $UUURRRDLDD$.

The basic computational scheme employed is a simple iterative one. We establish at the outset an arbitrary bound on the number of jumps which will be concatenated to form a shift (in practice 50 is more than adequate). Then we take an existing set of jumps and compute a new set of shifts by concatenating them in this way. These new shifts are in turn substituted into our supply of primitive jumps in order to compute a new set of jumps and so on.

All of this is handled numerically by passing at the outset to generating functions in a single variable t (which replaces each of the letters of the meander alphabet). For a fixed real value of t we can then carry out the computation described above. If the value of t lies outside of the radius of convergence of the generating function then the iteration will diverge. It is easy to establish strict divergence criteria for this iteration. For example, the RHS of the equation defining s_0 dominates the one which would define an ordinary one-dimensional drunkard's walk, that is, $1+t^2 s_0^2$. In particular, if ever $s_0 > 1/t^2$ then each successive iteration must increase s_0 by at least 1, and hence divergence is established. We can make use of a loose convergence criterion (no divergence through some fixed number of iterations), since lower bounds on M are determined by upper bounds on the radius of convergence of the generating function. Then a simple binary search on t allows us to determine rigorous upper bounds on the radius of convergence for $s_0(t)$.

Using jumps of length up to 24, we obtain an upper bound for the radius of convergence of $s_0(t)$ of 0.296431. This translates to a lower bound of 11.38 on M.

Given that our supply of primitive jumps is finite, there is a bound on the displacement of each jump. Using this it is possible to compute exact values for shifts made up of arbitrarily many such jumps using standard techniques from the enumeration of drunkard's walks. In practice this scheme suffers from a number of drawbacks. First, it is computationally much more expensive and complex than the simple iteration. Second, the results obtained are not significantly better than those obtained by simple iteration since the dominant terms for shifts will in any case be composed of relatively few jumps. Finally, allowing arbitrarily many jumps per shift would require verification that almost all such shifts still remain within the meander context. Since our primitive shifts are of bounded displacement, we can guarantee that the excursions away from the original centre of the meander context are "not large" except in a vanishing proportion of cases, and so almost all of the words which we (implicitly) enumerate through the recursive scheme are legitimate.

6.2. Upper bounds. In producing upper bounds for the growth rate of meander numbers we begin from a set $\mathcal{B}$ of standard representatives of words creating a submeander. Again, the most straightforward approach is simply to list all such words up to some predefined length. Doing this again involves a depth-first search in the extended meander environment. This time we must check that the final D joins the segments formed by the initial U, that no earlier D creates a sub-meander, and that no jumps occur as subwords other than L and R. All these tests are easily implemented within the meander environment.

After passing to a single variable t we use equation (5.1) in order to compute the quantities x_b. Rather than solving this large (but relatively sparse) system exactly we may use a simple iterative scheme since it is easily checked that for values of t in the range we are interested in there are no eigenvalues of the matrix representing the summations on the RHS of this equation whose modulus is greater than or equal to 1. Convergence is therefore guaranteed, with error bounds decreasing by a constant factor on each iteration. Having computed the values x_b, all that is necessary is to evaluate the sign of

$$1 - 4t + \sum_{b \in \mathcal{B}} x_b(t)$$

in order to determine whether t lies above or below the radius of convergence (below if the sign is positive, above if it is negative). Again a simple binary search can now be used to estimate the radius of convergence, and hence an upper bound on the exponential growth of the meander numbers.

Using the 20509 words of length 16 which are standard representatives of words creating a submeander for $\mathcal{B}$ produces an estimate of 0.2784 for the radius of convergence of $\mathcal{B}$-factor-free words, and hence an upper bound of 12.901 on M.

7. Summary and conclusions

Obviously the methods which we have applied could be extended to obtain better bounds through more extensive computation using longer words as primitive jumps, or as the standard representatives of submeander words. Some indication of how far this might or might not progress is shown in Table 1.

Lower bounds		Upper bounds	
10	10.749	6	13.171
12	10.928	8	13.086
14	11.023	10	13.018
16	11.114	12	12.970
18	11.188	14	12.931
20	11.249	16	12.901
22	11.301		
24	11.380		

TABLE 1. Lower and upper bounds on M based on maximum length of jumps, and submeanders.

A simple extrapolation based on this data suggests a limiting lower bound of approximately 11.6, and an upper bound of approximately 12.8. However, the final lower bound which we have computed (from jumps up to length 24) represents a better than expected improvement on the previous value. Put another way, there are more jumps of length 24 than one would expect based on simple extrapolation of previous values. So, it may be that better improvements on the lower bound are possible.

References

[DF00] P. Di Francesco, *Exact asymptotics of meander numbers*, Formal power series and algebraic combinatorics (Moscow, 2000), Springer, Berlin, 2000, pp. 3–14. MR **2001i:**05016

[DFGG00] P. Di Francesco, O. Golinelli, and E. Guitter, *Meanders: exact asymptotics*, Nuclear Phys. B **570** (2000), no. 3, 699–712. MR **2001f:**82032

[DFGJ00] P. Di Francesco, E. Guitter, and J. L. Jacobsen, *Exact meander asymptotics: a numerical check*, Nuclear Phys. B **580** (2000), no. 3, 757–795. MR **2002b:**82024

[FE02] Reinhard O. W. Franz and Berton A. Earnshaw, *A constructive enumeration of meanders*, Ann. Comb. **6** (2002), no. 1, 7–17. MR **2003f:**05004

[GJ79] I. P. Goulden and D. M. Jackson, *An inversion theorem for cluster decompositions of sequences with distinguished subsequences*, J. London Math. Soc. (2) **20** (1979), no. 3, 567–576. MR **83b:**05005

[GJ83] ———, *Combinatorial enumeration*, A Wiley-Interscience Publication, John Wiley & Sons Inc., New York, 1983, With a foreword by Gian-Carlo Rota, Wiley-Interscience Series in Discrete Mathematics. MR **84m:**05002

[Jen00] Iwan Jensen, *A transfer matrix approach to the enumeration of plane meanders*, Journal of Physics A: Mathematical and General **33** (2000), no. 34, 5953–5963.

[JG00] Iwan Jensen and Anthony J Guttmann, *Critical exponents of plane meanders*, Journal of Physics A: Mathematical and General **33** (2000), no. 21, L187–L192.

[LZ93] S. K. Lando and A. K. Zvonkin, *Plane and projective meanders*, Theoret. Comput. Sci. **117** (1993), no. 1-2, 227–241, Conference on Formal Power Series and Algebraic Combinatorics (Bordeaux, 1991). MR **94i:**05004

[NZ99] John Noonan and Doron Zeilberger, *The Goulden-Jackson cluster method: extensions, applications and implementations*, J. Differ. Equations Appl. **5** (1999), no. 4-5, 355–377. MR **2000f:**05005

[Zei02] Doron Zeilberger, *The umbral transfer-matrix method. V. The Goulden-Jackson cluster method for infinitely many mistakes*, Integers **2** (2002), Paper A5, 12 pp. (electronic). MR **2003f:**05010

Department of Computer Science, University of Otago, Dunedin, New Zealand
E-mail address: `malbert@cs.otago.ac.nz`

Department of Computer Science, University of Warwick, Coventry, UK
E-mail address: `msp@dcs.warwick.ac.uk`