Calculating the 2-variable polynomial for knots presented as closed braids

H.R. Morton and H.B. Short

I. Introduction

The recently developed 2-variable polynomial P_K of an oriented knot K (of one or more components), [FYHLMO], has widened the range of invariants available which can be used to distinguish between knots, (closed curves in space). In an attempt to investigate how far certain geometric properties of K are mirrored in algebraic features of P_K we were led to try to calculate P for some 48-crossing knots. The feature of exponential growth with number of crossings of the most direct algorithm then led to the development of an algorithm which, for knots presented as closed braids on at most 8 strings, grows only quadratically with the number of crossings.

This algorithm has been implemented on the Liverpool University IBM 3083 computer to allow interactive handling of braids on 7 strings, and the program will deal with 8-string braids of up to 150 crossings using less than 16 megabytes storage and under 200 seconds of computer time.

The limitation on the number of strings, although somewhat restrictive, still allows a wide variety of interesting examples to be studied, [M-S]. While extension to 9-string braids might be practicable, further attempts to increase the string index of braids handled will run up against the problem that storage requirements increase factorially with the number of braid strings.

In this paper we describe the algorithm used for calculation, starting with a development of the theory based on the approach of Ocneanu and

Jones [O], [J]. This is followed by an outline of the features in its practical implementation. An annotated copy of the Pascal program is available on request.

It should be possible to modify the approach so as to allow calculation of Kauffman's new 2-variable polynomial F_K with similar dependence on crossing number for knots presented as closed braids with few strings, or even in this case with a mixed braid and plat presentation.

II Theoretical background

Computations are based on the approach of Ocneanu and Jones, [O], to the 2-variable polynomial, in which a braid β on n strings closing to the given oriented knot K is represented as $\rho_v(\beta)$ in an algebra H. A linear trace function $Tr : H \to \mathbb{C}$ can be found so that after normalisation by a suitable constant μ the number $P(\beta) = \frac{1}{\mu^{n-1}} Tr(\rho_v(\beta))$ depends only on K, and not on the representing braid β. This number $P(\beta)$ is a polynomial with integer coefficients, $P_K(v, z)$, in two parameters $v^{\pm 1}$, $z^{\pm 1}$ which are involved in the construction of H and the representation ρ_v, and provides the invariant of K which is to be calculated from a given choice of β.

§1. The algebra H

The symmetric group S_n can be generated by the transpositions

$\tau_i = (i, i + 1), 1 \leq i \leq n - 1,$ with relations

$$1''. \quad \tau_i^2 = e$$

$$2''. \quad \tau_i\tau_j = \tau_j\tau_i, \quad |i - j| > 1$$

$$3''. \quad \tau_{i+1}\tau_i\tau_{i+1} = \tau_i\tau_{i+1}\tau_i, \quad 1 \leq i \leq n - 1.$$

The braid group B_n of braids on n strings has a similar presentation, with generators σ_i, $1 \leq i \leq n - 1$, represented by a braid


i i+1

which interchanges strings i and $i+1$, and relations,

$$2'. \quad \sigma_i\sigma_j = \sigma_j\sigma_i, \ |i - j| > 1$$

$$3'. \quad \sigma_{i+1}\sigma_i\sigma_{i+1} = \sigma_i\sigma_{i+1}\sigma_i, \quad 1 \le i \le n - 1.$$

The natural homomorphism $\pi : B_n \to S_n$ defined by $\pi(\sigma_i) = \tau_i$ takes a braid β to the permutation $\pi(\beta)$ induced on the ends of its strings.

Natural inclusions $S_n \subset S_{n+1}$, $B_n \subset B_{n+1}$ allow us to regard each as a subgroup of a group S_∞ or B_∞ generated by τ_i or σ_i, $1 \le i$, respectively, with relations as above.

We can construct, for each $z \in \mathbb{C}$, an algebra $H(z)$ with generators c_i, $1 \le i$, and relations

1. $c_i^2 = zc_i + 1$
2. $c_ic_j = c_jc_i$, $|i - j| > 1$
3. $c_{i+1}c_ic_{i+1} = c_ic_{i+1}c_i$, $1 \le i$,

a Hecke algebra, which becomes the group algebra $\mathbb{C}[S_\infty]$ when $z = 0$.

The subalgebra H_n, generated by $c_1, \ldots, c_{n-1}$ (and corresponding to S_n) can be shown, following the exercises in Bourbaki, [Bo, p53-56], to have dimension $n!$ as a vector space, provided that z avoids certain values. Choosing z to be transcendental (or reformulating with z an indeterminate) will guarantee this. In section 4 we give an explicit set of $(n + 1)!$ generators for H_{n+1} which correspond bijectively to the elements of S_{n+1}.

It follows from relation 1 that c_i is invertible with $c_i^{-1} = c_i - z$, a linear combination of 1 and c_i. Then B_n can be represented in H_n, for any choice of v, by a homomorphism ρ_v, where $\rho_v(\sigma_i) = vc_i$.

§2. The trace function Tr.

Assuming that $\dim(H_n) = n!$ for almost all z, [Bo], it is then possible to construct, for any given $T \in \mathbb{C}$, a linear function

$Tr : H \to \mathbb{C}$ with the following properties:

4. $Tr(1) = 1$

5. $Tr(ab) = Tr(ba)$

6. $Tr(w\,c_n) = T\,Tr(w)$ for all $w \in H_{n-1}$.

A brief sketch of the construction, given in detail by Jones in his seminar notes [J], follows.

First use a dimension count to observe that the subspace $H_{n-1} \subset H_n$ has a complement K_{n-1} isomorphic to $H_{n-1} \otimes_{H_{n-2}} H_{n-1}$, with isomorphism α defined by $\alpha(y_1 \otimes y_2) = y_1 c_n y_2$. The linear function Tr can then be defined inductively on H_n by its implicit definition on K_{n-1} as $Tr(\alpha(y_1 \otimes y_2)) = T\,Tr(y_1 y_2)$, assuming that it is already defined on H_{n-1}, starting with $Tr(1) = 1$. Condition 6 is then guaranteed.

Since H_n is generated as an algebra by H_{n-1} and c_n, condition 5 will follow in H_n once it is proved for $b \in H_{n-1}$ and $b = c_n$. The first follows readily from the definition, and the second is established by showing that $Tr(y_1 c_n y_2 c_n) = Tr(c_n y_1 c_n y_2)$ for $y_1, y_2 \in H_{n-1}$. This in turn follows from the decomposition $H_{n-1} = H_{n-2} \oplus K_{n-2}$ and the fact that c_n and H_{n-2} commute.

It follows from 4 and 6 that $Tr(w\,y) = Tr(y)Tr(w)$ for all y in the 2-dimensional subalgebra generated by c_n and all $w \in H_{n-1}$.

Given v and z we may choose T so that

$Tr(\rho_v(\sigma_i)) = vT = Tr(\rho_v(\sigma_i^{-1})) = v^{-1}Tr(c_i^{-1}) = v^{-1}(T - z) = \mu$ say, by taking

$$T = \frac{v^{-1}z}{v^{-1}-v} = \frac{z}{1-v^2}.$$

For $\beta \in B_n$ we then have $Tr(\rho_v(\beta\sigma_n^{\pm 1})) = Tr(\rho_v(\beta))Tr(\rho_v(\sigma_n^{\pm 1})) = \mu Tr(\rho_v(\beta))$

§3. The polynomial P_K.

We now define $P_K(v, z)$ for the oriented knot K by representing K as the closure of an n-braid (β, n), and setting $P_K(v, z) = Tr(\rho_v(\beta)).\ \mu^{1-n}$.

This is independent of the choice of representative β, since it is invariant when β is altered by any of the Markov moves $(\beta, n) \sim (\gamma\beta\gamma^{-1}, n) \sim (\beta\sigma_n^{\pm 1}, n+1)$ which generate the set of braids with the same closure K. The choice of normalising factor μ^{1-n} guarantees independence of the second of these moves and ensures, with 4, that $P_K = 1$ when K is the unknot, while property 5 gives independence under the first move, conjugacy in B_n.

The Conway relation $v^{-1}P_{K^+} - vP_{K^-} = zP_{K^0}$ between the polynomials of knots whose diagrams differ as shown, 

,

then follows by presenting K^+, K^- and K^0 as the closures of $\beta\sigma_i$, $\beta\sigma_i^{-1}$ and β respectively for some $\beta \in B_n$. Then $c_i - c_i^{-1} = z$ from 1, so $v^{-1}\rho_v(\sigma_i) - v\rho_v(\sigma_i^{-1}) = z$. Multiply by $\rho_v(\beta)$ to get a corresponding relation between $\rho_v(\beta\sigma_i^{\pm 1})$ and $\rho_v(\beta)$ and hence $v^{-1}\mathrm{Tr}(\rho_v(\beta\sigma_i)) - v\mathrm{Tr}(\rho_v(\beta\sigma_i^{-1})) = z\mathrm{Tr}(\rho_v(\beta))$.

In later calculations we will suppose that β is given as a word $w(\sigma_1^{\pm 1}, \ldots, \sigma_{n-1}^{\pm 1})$, so that $\rho_1(\beta) = w(c_1^{\pm 1}, \ldots, c_{n-1}^{\pm 1})$ and $\rho_v(\beta) = v^k\rho_1(\beta)$, where $k = \tilde{c}(\beta)$ is the algebraic number of crossings in β, or equally the exponent sum of the generators in the monomial w. We can then work almost entirely with $\rho_1(\beta)$ in calculating P_K.

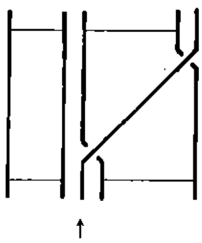
§4. Generators of H

We shall list $(n + 1)!$ generators of H_{n+1} as b_g, $1 \le g \le (n + 1)!$. Each one is $\rho_1(\beta_g)$ for some positive braid β_g, whose corresponding permutations $\pi(\beta_g)$ range over all of S_{n+1}; in fact β_g will be the unique positive braid of minimum length which realises the permutation $\pi(\beta_g)$.

For $r = 1, \ldots, n$ put $\sigma(r, n) = \sigma_{n-r+1}\sigma_{n-r+2}\cdots\sigma_n$ and take $\sigma(0, n) = 1$. The braid $\sigma(r, n)$ then makes r crossings as illustrated in Fig. 1.

Figure 1

The braid $\sigma(r, n)$



1 n-r+1 n+1

Write $c(r, n) = \rho_1(\sigma(r, n)) \in H_{n+1}$.

Now suppose that, starting with $b_1 = 1$, we have already constructed generators b_g, $g \leq n!$, for H_n. We extend this to a definition of b_h, $h \leq (n + 1)!$ by writing $h = g + r.(n!)$ with $g \leq n!$ and putting $b_h = c(r, n)\, b_g$. The braids β_h can be defined similarly using $\sigma(r, n)$ in place of $c(r, n)$.

The fact that H_{n+1} is generated by these elements b_h can be seen from our subsequent calculations of $b_h c_i^{\pm 1}$ as linear combinations of b_h and b_g for some other g.

Given g it is easy to write down b_g explicitly using the 'factorial expansion' of g. We may write g uniquely as $g = 1 + \sum_{j=1}^{n} g_j\,(j!)$ where $0 \leq g_j \leq j$. Then $b_g = c(g_n, n) \ldots c(g_1, 1)$, having length $\ell(g) = \sum_{j=1}^{n} g_j$ as a monomial in the algebra generators c_i.

III The practical algorithm

When the knot K is presented as the closure of a braid $\beta \in B_{n+1}$ the procedure to calculate $Tr(\rho_1(\beta))$, and hence very quickly to find $P_K(v, z)$, has two stages, called 'multiply' and 'wrap-up'.

The braid β is given as a monomial $w(\sigma_1, \ldots, \sigma_n)$. The principal procedure, 'multiply', expresses $\rho_1(\beta) = w(c_1, \ldots, c_n)$, which is initially a word in the algebra generators $c_r^{\pm 1}$, as a linear combination $\rho_1(\beta) = \sum_{g=1}^{(n+1)!} w_g b_g$ of the vector space generators b_g of H_{n+1}. The coefficients $w_g = \sum_{j=0}^{d(g)} a(g, j)\, z^j$ are integer polynomials, whose degree is at most the length of the monomial w.

We could at this stage calculate $Tr(\rho_1(\beta))$ from a list of $Tr(b_g)$ for $1 \leq g \leq (n + 1)!$ However it proves very wasteful of space to store such a list for $n \geq 6$, bearing in mind that $8! > 40{,}000$, and that each $Tr(b_g)$ requires storage of up to $\frac{1}{2}n^2$ integers. The 'wrap-up' procedure uses instead the conjugacy property, 5, of Tr to find successively simpler elements of H, and eventually a linear combination of $n + 1$ basis elements, with the same trace as $\rho_1(\beta)$.

§1. 'Multiply' procedure

This procedure, to calculate a monomial $w(c_1, \ldots, c_n)$ as a linear combination of basis elements of H, is based on a simple calculation of $b_g c_r^{\pm 1}$ for each r. For a given choice of r the basis elements of H can be paired so as to pair b_g and b_h whose permutations are related by $\pi(\beta_g)\tau_r = \pi(\beta_h)$. The elements $b_g c_r^{\pm 1}$ turn out to be always a linear combination of the pair b_g and b_h. The exact form of $b_g c_r^{\pm 1}$ can best be described using the factorial coordinates $(g_1, \ldots, g_n)$ for g mentioned earlier.

Lemma 1 If $g_r \le g_{r-1}$ then $b_g c_r = b_h$, where $h_{r-1} = g_r$, $h_r = g_{r-1} + 1$, $h_j = g_j$ otherwise. Then $h_r > h_{r-1}$ and $h > g$.

This lemma enables us to locate, for a given r, the elements b_g and b_h paired by τ_r, for if we find h with $h_r > h_{r-1}$ then we can clearly find g with $g_r \le g_{r-1}$ and $b_g c_r = b_h$. Indeed if we work through the basis elements in order we will always meet the element b_g with $g_r \le g_{r-1}$ before its pair.

From Lemma 1, when b_g and b_h are paired by τ_r with $g < h$ then

$$b_g c_r = b_h \quad \text{and} \quad b_h c_r = b_g c_r^2 = z\, b_g c_r + b_g$$
$$= z\, b_h + b_g.$$

If we are given $w = w_h b_h + w_g b_g$ with g, h as above then

$$wc_r = (w_g + zw_h)b_h + w_h b_g \quad \text{and similarly}$$
$$wc_r^{-1} = w_g b_h + (w_h - zw_g)b_g.$$

As a consequence, $Wc_r^{\pm 1}$ can be readily calculated from $W = \sum_{g=1}^{(n+1)!} w_g b_g$ as a linear combination of the basis elements. For each g simply calculate enough factorial coordinates to decide if it is the first or second in a pair, and then alter the coefficients of both elements of the pair as dictated above when the first element is reached.

Given $\beta = w(\sigma_1, \ldots, \sigma_n) \in B_{n+1}$ we can then express $\rho_1(\beta) = w(c_1, \ldots, c_n) \in H_{n+1}$ in the form $\rho_1(\beta) = \sum_{g=1}^{(n+1)!} w_g b_g$ by successive multiplications by $c_r^{\pm 1}$, starting with $1 = b_1 \in H_{n+1}$.

The coefficients w_g will be integer polynomials in z of degree at most the length of w. Initially take $w_1 = 1$, $w_g = 0$, $g \neq 1$. After each successive multiplication by $c_r^{\pm 1}$ the current coefficients $a(g, j)$ for $w_g = \sum_j a(g,j)z^j$ may be stored in an $(n + 1)! \times (\text{length } w)$ matrix. To avoid repeated handling of many zero entries it is convenient to store $d(g)$, the current degree of w_g, which is easily estimated from the previous degree of w_g and w_h. As a refinement, to save on storage in many cases, the coefficients of each polynomial w_g may be held as a linked list using only the non-zero coefficients.

The quadratic growth of the algorithm with the length $\ell(w)$ in B_{n+1}, for fixed n, can be seen from the number of operations, proportional to $\ell(w)$, needed to alter the matrix (a_{gj}) of coefficients when w is altered to $w\, c_r^{\pm 1}$.

<u>Proof of Lemma 1.</u>

Write $b_g = I\, c(g_r, r)\, c(g_{r-1}, r-1) J$ where $J \in H_{r-1}$ and I is the product of $c(g_j, j)$ for $j > r$. Now c_r commutes with J, so $b_g c_r = I\, c(g_r, r)\, c(g_{r-1}, r-1)\, c_r\, J$.

It is then enough to show that

$c(g_r, r)\, c(g_{r-1}, r-1) c_r = c(g_{r-1} + 1, r)\, c(g_r, r-1)$ when $g_r \le g_{r-1}$.

Write $p = r - g_r$, $s = r - 1 - g_{r-1}$. The condition $g_r \le g_{r-1}$ then becomes $s < p$. Now for $s < k \le r$ we have

$c_k . c_s c_{s+1} \ldots c_r = c_s c_{s+1} \ldots c_r . c_{k-1}$ from the relations in H.

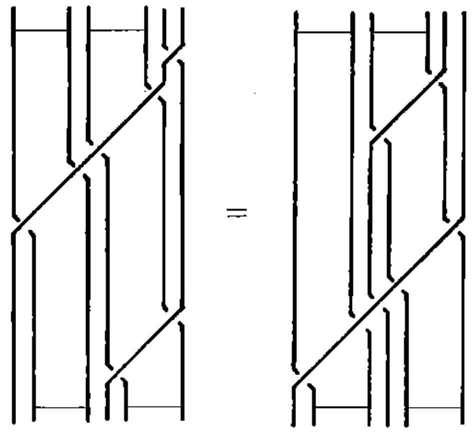
Then $c(g_r, r)\, c(g_{r-1}, r-1)\, c_r = c_p c_{p+1} \ldots c_r . c_s c_{s+1} \ldots c_{r-1} . c_r$

$$= c_s c_{s+1} \ldots c_r . c_{p-1} \ldots c_{r-1}$$

$$= c(g_{r-1} + 1, r)\, c(g_r, r-1).$$

The corresponding braids are illustrated in fig. 2, showing the strings from s to $r + 1$.

Figure 2



§2. 'Wrap-up' procedure

The inductive definition of the basis elements of H_{n+1} allows them to be described as $c(r, n)\ b_h$, $1 \leq h \leq n!$, $0 \leq r \leq n$. When the element $W = \sum_{g=1}^{(n+1)!} w_g b_g$ is rewritten by grouping together $n + 1$ successive blocks of $n!$ terms we have

$$W = \sum_{r,h} w_{r.n!+h}\ c(r, n)\ b_h = \sum_{r=0}^{n} c(r, n)\ W_r$$

$$\text{where } W_r = \sum_{h=1}^{n!} w_{r.n!+h}\ b_h \in H_{n-1}.$$

For $r \geq 2$ we may write $c(r, n) = c_{n-r+1}\ c(r-1, n)$, and use the trace relation 5 to calculate $\mathrm{Tr}(c(r, n)W_r) = \mathrm{Tr}(c(r-1, n)\ W_r c_{n-r+1})$, with $W_r\ c_{n-r+1} \in H_n$. Then calculation of $W_r\ c_{n-r+1}$ by one application of the multiplication procedure to $W_r \in H_n$, whose coefficients we know explicitly, gives an element of H_{n-1}. We may alter W by replacing W_{r-1} by $W_{r-1} + W_r c_{n-r+1}$ and setting $W_r = 0$ (or in practice simply omitting W_r) to find a new element of H having the same trace as W but using $n!$ fewer coefficients. Following this procedure successively from $r = n$ down to $r = 2$ gives $V = V_0 + c(1, n)V_1$ with $V_0, V_1 \in H_n$ and $\mathrm{Tr}V = \mathrm{Tr}W$; V_0 in fact remains as W_0 in this process.

The procedure is then repeated, with $n - 1$ in place of n, on V_0 and V_1, to get $U = (U_{00} + c(1, n - 1) U_{01}) + c(1, n) (U_{10} + c(1, n - 1) U_{11})$ with the same trace as before, and $U_{ij} \in H_{n-1}$.

Since $c(1, r) = c_r$, the trace relation 6 shows that $\mathrm{Tr}\, U = \mathrm{Tr}\, U_{00} + T(\mathrm{Tr}\, U_{01} + \mathrm{Tr}\, U_{10}) + T^2(\mathrm{Tr}\, U_{11})$, and we can simplify the trace calculation further by using

$$U_{00} + c_{n-1}(U_{01} + U_{10}) + c_n c_{n-1} U_{11} \quad \text{in place of } U,$$

Now replace each of the three elements of H_{n-1} in a similar way, and continue, to finish with an element $x = x_0 + c_1 x_1 + c_2 c_1 x_2 + \ldots c_n c_{n-1} \ldots c_1 x_n$, with $\mathrm{Tr} W = \mathrm{Tr}\, x$, and each $x_j \in H_1$ being an integer polynomial in z, stored as the coefficient of b_g for $g = 1 + 1! + 2! + \ldots + j!$.

We then immediately have

$$\mathrm{Tr}\, W = \sum_{i=0}^{n} x_i(z)\, T^i$$

§3. Polynomial calculations

The calculation of $P_K(v,z)$ where $(\beta, n + 1)$ closes to K, then follows readily from $P_K(v, z) = v^{\tilde{c}(\beta)} \mu^{-n}\, \mathrm{Tr}\, W$

$$= v^{\tilde{c}(\beta)-n} \sum_{i=0}^{n} x_i(z) T^{i-n}$$

$$= v^{\tilde{c}(\beta)-n} \sum x_i(z)\, z^{i-n}(1 - v^2)^{n-i}.$$

The table of P_K is then calculated using the binomial expansion of $(1 - v^2)^{n-i}$, which does not require large binomial coefficients, and the coefficients of the polynomials $x_i(z)$, together with the algebraic crossing number $\tilde{c}(\beta)$ which is calculated from the initial presentation of the word β.

The Alexander and Jones polynomials are calculated from P_K by the substitution $z = \sqrt{t} - \frac{1}{\sqrt{t}}$ and $v = 1$ or t respectively. Binomial expansion of powers of z up to the length of the initial word w are required here, and can give errors where this length is > 100 arising from the calculation of large binomial coefficients, many of which may subsequently cancel.

The size of the coefficients in the polynomials $x_i(z)$ themselves becomes quite large, particularly for positive braids, and may exceed the largest integer permitted in the computer when the braid length is of the order of 100, unless special arrangements are made.

§4 Reliability

A variety of calculations have been made, agreeing with independent calculations by Thistlethwaite [T] where these were available.

Examples with 8-string presentations have been looked at with several properties in mind, [M-S]. Calculations using a braid β which are repeated using a conjugate of β have always given the same result (as they should), although the route through the algorithm in the two cases will involve very different intermediate calculations.

The most convincing evidence of the general reliability of the implementation comes in one pair of complicated examples, also described in [M-S], which give identical polynomials although the knots themselves differ significantly. In this case at least the chances of a faulty implementation producing by error the same polynomials with over 100 non-zero coefficients are extremely small, and our confidence in the other calculations is greatly increased.

The only significant danger of miscalculation appears to come from undetected integer overflow in the coefficients, and rounding error on the large binomial coefficients needed for the Alexander and Jones polynomials. These errors will almost inevitably show up in the Alexander polynomial which can often be calculated quite readily by other means; a check on this is a very good indicator that no overflow has happened in the earlier routines.

By way of illustration,

$\beta = (\sigma_4 \sigma_5 \sigma_3 \sigma_4)^3 (\sigma_2 \sigma_1 \sigma_3 \sigma_2)^{-1} \sigma_6 \sigma_5 \sigma_7 \sigma_6 \quad (\sigma_4 \sigma_3 \sigma_5 \sigma_4)^{-2} \sigma_6 \sigma_5 \sigma_7 \sigma_6$
$(\sigma_4 \sigma_5 \sigma_3 \sigma_4)^{-1} (\sigma_2 \sigma_1 \sigma_3 \sigma_2)^{-1} \sigma_6 \sigma_5 \sigma_7 \sigma_6 \sigma_7$ is shown in figure 3.

This is one of the pair referred to above, and its closure K is a 2-cable about Conway's eleven-crossing knot. In the accompanying table 1 we give the coefficients p_{ij} of the polynomial $P_K = \Sigma\, p_{ij} v^i z^j$, with the corresponding degrees in v and z noted at the top and side. We also give its Alexander and Jones polynomials in powers of $\sqrt{t}$, with negative powers omitted in the case of the Alexander polynomial.

The other braid γ in the pair, which closes to the same type of cable about Kinoshita and Teresaka's mutant of Conway's knot is given by

$$\gamma = (\sigma_2\,\sigma_1\,\sigma_3\,\sigma_2)^2\,(\sigma_6\,\sigma_5\,\sigma_7\,\sigma_6)^3\,\sigma_4\,\sigma_3\,\sigma_5\,\sigma_4\,(\sigma_6\,\sigma_5\,\sigma_7\,\sigma_6)^{-2}\,(\sigma_2\,\sigma_3\,\sigma_1\,\sigma_2)^{-1}$$
$$\sigma_4\,\sigma_3\,\sigma_5\,\sigma_4\,(\sigma_6\,\sigma_5\,\sigma_7\,\sigma_6)^{-1}\,(\sigma_2\,\sigma_3\,\sigma_1\,\sigma_2)^{-1}\,(\sigma_4\,\sigma_3\,\sigma_5\,\sigma_4)^{-1}\,\sigma_7.$$

This braid and β give identical polynomials P.

Figure 3

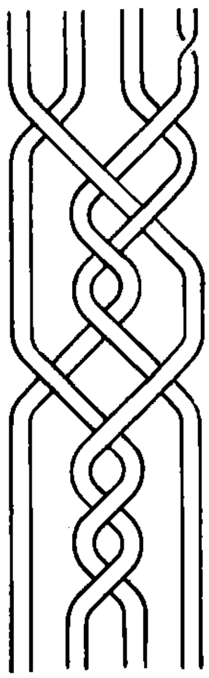


Table 1

Braid representing a 2-cable about Conway's 11-crossing knot

*** 43544354435421 32-6-5-7-6-4-3-5-4-4-3-5-42132-4-3-5-42132-6-5-7-61 ***

Braid representing a similar 2-cable about Kinoshita-Teresaka knot

*** 2132213221326576657643 54-6-5-7-6-2-1-3-2-2-1-3-24354-6-5-7-6
-2-1-3-2-4-3-5-41 ***

The polynomial P for both of these knots

-2	0	2	4	6	8	10	12	
15	-97	233	-252	101	33	-43	11	0
146	-861	1917	-1926	646	341	-344	82	2
688	-3533	7068	-6430	1815	1251	-1115	256	4
1831	-8531	15171	-12175	2879	2352	-1982	455	6
2921	-13081	20828	-14371	2800	2547	-2115	471	8
2870	-13145	19014	-10997	1714	1656	-1389	277	10
1757	-8781	11703	-5514	656	651	-562	90	12
667	-3908	4850	-1790	151	151	-136	15	14
152	-1142	1330	-361	19	19	-18	1	16
19	-210	231	-41	1	1	-1		18
1	-22	23	-2					20
	-1	1						22

Conway Polynomial

1(0)
1(2)

Alexander Polynomial

-1(0)
1(2)

Jones Polynomial

1(-24)	1(12)
-2(-22)	2(14)
2(-18)	-6(16)
-1(-16)	5(18)
-2(-12)	3(20)
1(-10)	-8(22)
5(-8)	3(24)
-5(-6)	4(26)
-2(-4)	-3(28)
7(-2)	-1(30)
-3(0)	2(34)
-2(2)	-2(38)
3(4)	1(40)
-1(8)	
-1(10)	

REFERENCES

[Bo] N. Bourbaki Groupes et algèbres de Lie, Chapters 4, 5, 6. Hermann, Paris.

[FYHLMO] P. Freyd, D. Yetter, J. Hoste, W.B.R. Lickorish, K. Millett, and A. Ocneanu, 'A new polynomial invariant of knots and links', Bull. A.M.S. 12(1985), 239-246.

[J] V.F.R. Jones An introduction to Hecke algebras, Seminar notes, MSRI 1984.

[M-S] H.R. Morton & H.B. Short The 2-variable polynomial of cable knots. Math.Proc.Camb.Phil.Soc (to appear)

[O] A. Ocneanu A polynomial invariant for knots: a combinatorial and an algebraic approach. Preprint MSRI 1984.

[T] M.B. Thistlethwaite Tables of knot polynomials. Polytechnic of South Bank, London, 1985.

January 1986

Department of Pure Mathematics
University of Liverpool
P.O. Box 147,
LIVERPOOL.
L69 3BX

Figure 1

The braid $\sigma(r, n)$

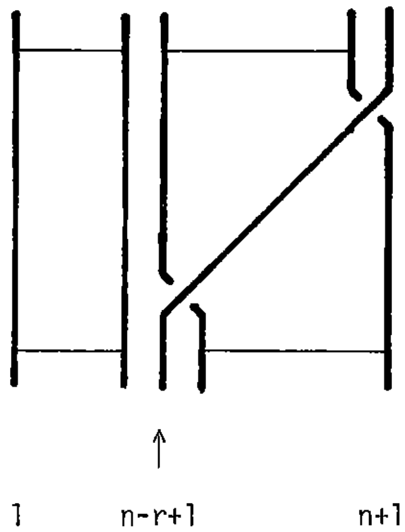


Figure 2

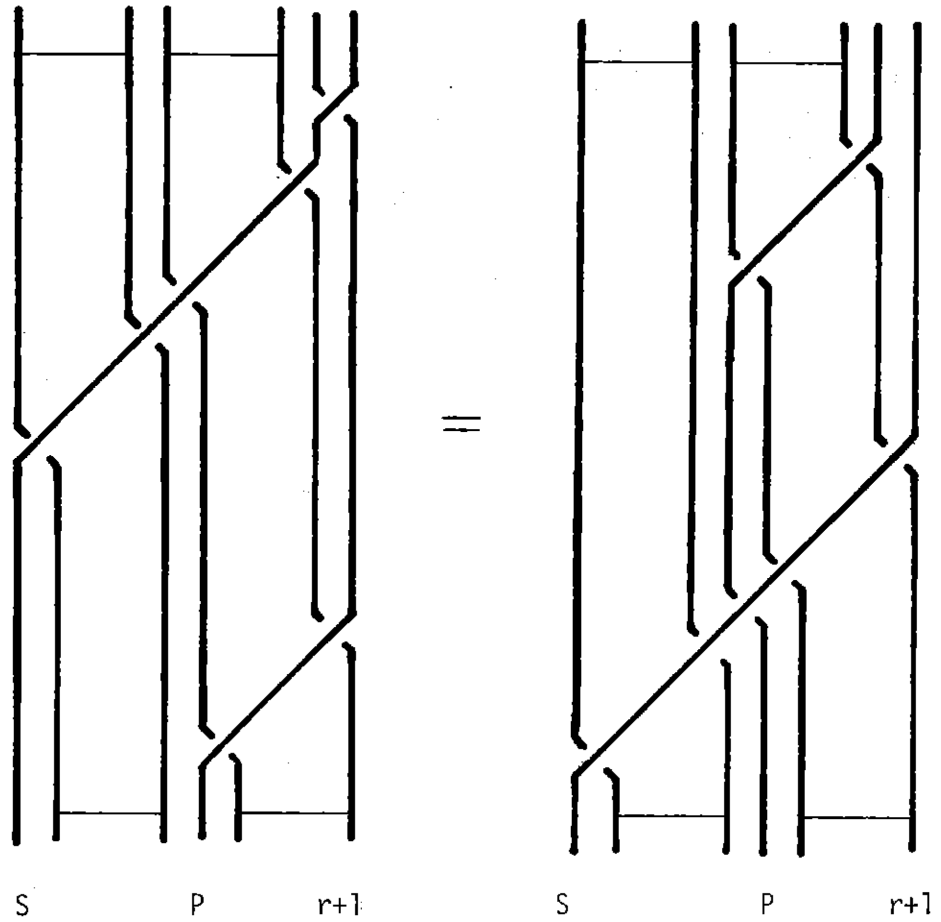


Figure 3