

# User's manual of “*Teruaki for Mathematica 7*” (T4M7)

Takuya SAKASAI      Masaaki SUZUKI

October 7, 2013

## 1 Preface

This manuscript is user's manual of the program “*Teruaki for Mathematica 7*” (T4M7 for short). This program was made under the guidance and the help of Professor Kazushi Ahara, who produced the original software “*Teruaki*”[2]. The authors would like to thank him.

For the theoretical part of this program, please refer to Ahara's article [1].

Please do NOT distribute this program to others. When you find some bugs in the program, it is helpful if you inform the authors about it. Finally, please use this program at your own risk.

## 2 Abstract

T4M7 enables us to draw simple closed curves on a surface, apply Dehn twists to them and compute some invariants of the mapping class given by Dehn twists in a computer. That is, T4M7 serves as a “mapping class group calculator”. The game as we can enjoy in Ahara's original software [2] is not programmed. We can choose the surface of any genus inside the capacities of our computer and *Mathematica*.

There are two types of the program

```
T4M7_*****.nb,                   ***** = Date of update  
T4M7_*****_Sg1.nb
```

where the first one corresponds to the mapping class group of a closed surface and the latter corresponds to that of a compact surface with one boundary.

## 3 How to use the program

We explain by using T4M7\_\*\*\*\*\*.nb mainly. However the explanation is valid also for T4M7\_\*\*\*\*\*\_Sg1.nb.

### 3.1 Preliminaries

After starting *Mathematica version 7*, open the file `T4M7_*****.nb` and input<sup>1</sup> the only cell in it. Then a notebook in which `drawcurve[]` is written and two kinds of palettes are opened. One of the palettes, which we call the *main palette*, contains some buttons labeled by arrows and another one, which we call the *twist palette*, is used when we apply Dehn twists to curves.

### 3.2 Choosing and changing the genus of a surface

We begin by choosing the genus of a closed surface. The default value of the genus is 3. If we want to change it to  $g \geq 1$ , input

```
genuschange[g]
```

directly. Then `genus=g` is outputted, and the genus is changed.

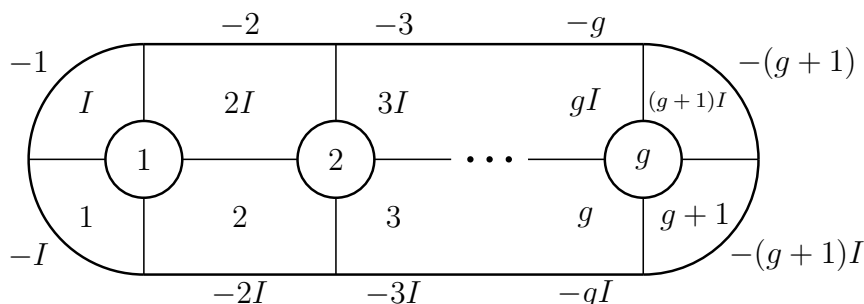
**Remark 3.1** We can use this command not only first, but anytime you prefer. Note that this command also initializes the data of simple closed curves you preserved (see Section 3.5).

### 3.3 Drawing curves

First, input

```
drawcurve[n]
```

directly or by using the button `drawcurve` on the palette. Here  $n$  is the label of the 2-cell of  $\Sigma_g$  from which a curve starts. In the program, the following cell decomposition is given to  $\Sigma_g$ .



Cell decomposition of  $\Sigma_g$  and the labels of 2-cells  
(The labels around the picture mean those of 2-cells of the backside)

After inputting the start point, we *move* around  $\Sigma_g$  by using buttons labeled by arrows on the main palette. At the endpoint of the curve, which is expected to be the same as the starting point, click the `setthecurve` button on the main palette. Then the program checks that the starting point and the endpoint are the same, and judges whether or not the loop we input given by the word is homotopic to a simple closed curve (SCC for short) by using Ahara's algorithm in [1]. If our loop is not homotopic to any non-trivial SCC, then the warning is displayed and the drawing process is quited, so that we must return to the start.

If our loop is homotopic to a SCC, then

---

<sup>1</sup>Type Shift-Return.

This defines a SCC.  
`current:`

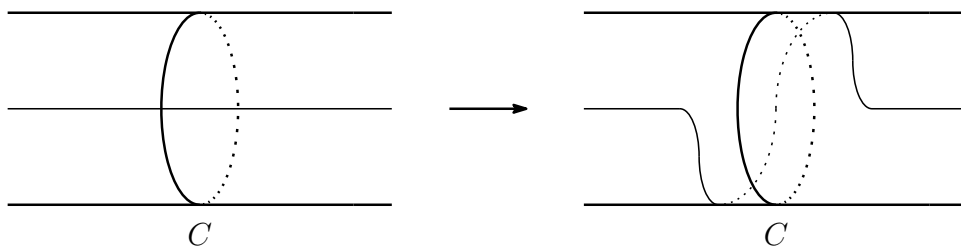
is outputted and the corresponding SCC, which may have a different starting point, is displayed.

The picture displayed as `current:` is important. It means that we are now considering this curve. We can refer to it by the variable `current`. In what follows, we call it simply the *current curve*. Note that the variable `current` is rewritten whenever we obtain a new data of a simple closed curve. To preserve the date of the current curve, the command `preserve` is used (See Section 3.5).

### 3.4 Dehn twists I (Twists along standard simple closed curves)

Here we explain how to apply Dehn twists along standard SCC's to the current curve. As for the method for twisting along SCC's we preserved, see Section 3.7.

**Remark 3.2** The *positive* Dehn twist along a simple closed curve  $C$  means the following.

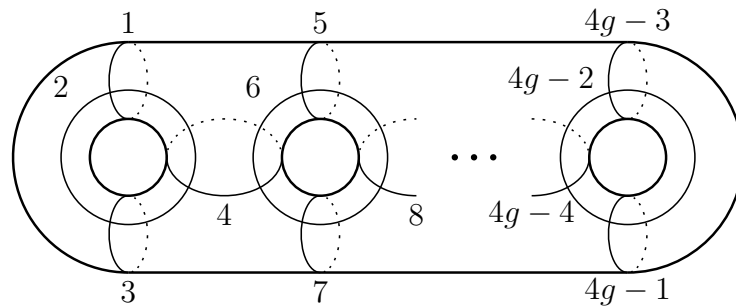


The positive Dehn twist along a simple closed curve  $C$

If we want to apply the positive Dehn twist along the SCC labeled by  $k$  ( $1 \leq k \leq 4g - 1$ ), to the current curve, input

`tw[k]`

directly, where the labels of standard SCC's are given as follows.



The labels of standard simple closed curves

We do not use the button `twist` in the main palette here. Note that if we apply the Dehn twist, the current curve is rewritten.

The negative Dehn twist is done by the command:

`tw[-k]`

**Remark 3.3** In the cases of  $g = 3$ , we can use the twist palette instead of inputting the commands.

**Remark 3.4** We can refer to the SCC having the index  $n$  in the above picture by `lic[n]`.

### 3.5 Preserving simple closed curves

To preserve the data of current curve, input

```
preserve
```

directly or by using the button `preserve` in the main palette. Then

```
This SCC is named simple[[n]]
```

is outputted, and the data of the current curve is preserved. We can refer to it by the variable `simple[[n]]`, where  $n$  is the number of the SCC assigned when we preserved it. Note that when we change the genus of a surface, the preserved data of curves are initialized.

### 3.6 Changing the current curve

If we input

```
current=simple[[n]]
```

then the current curve is changed into the curve preserved as `simple[[n]]`.

### 3.7 Dehn twists II (Twists along preserved simple closed curves)

If we want to apply the positive Dehn twist along the preserved SCC `simple[[n]]` to the current curve, input

```
twistCurrentAlongSimple[n]
```

directly or by using the button `twist` in the main palette. The negative twist is done by replacing  $n$  by  $-n$ .

### 3.8 Pictures of curves

To display the picture of the current curve, input

```
picture[current]
```

directly or by using the button `picture` in the main palette. We can also display a preserved SCC `simple[[n]]` by using the command

```
picture[simple[[n]]]
```

We can display a set of SCC's simultaneously by inputting

```
drawpicture[{simple[[n_1]], simple[[n_2]], ..., simple[[n_k]]}]
```

where we can put `current` (resp. `lic[n]`) among `simple[[n_i]]`'s if we want to display the current curve (resp. `lic[n]`). The curves are colored by three fashions, so that the color of the fourth curve is the same as that of the first, for example.

### 3.9 Hyperelliptic involution

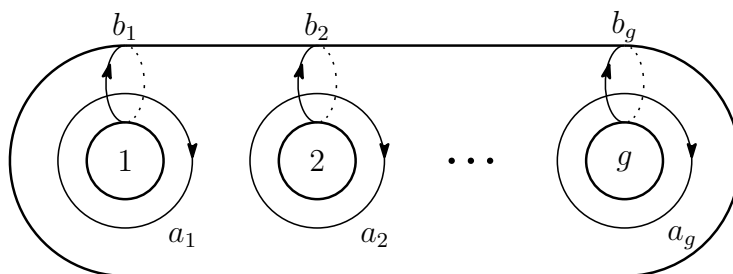
We can apply the hyperelliptic involution (with respect to the standard position) to the current curve by using the command `involution`.

### 3.10 Computing the action of the composite of Dehn twists on $H_1(\Sigma_g)$

The command

```
spAction[{simple[[n_1]],k_1,simple[[n_2]],k_2,...,
         simple[[n_l]],k_l}]
```

computes the action of the composite  $\varphi$  of Dehn twists on  $H_1(\Sigma_g)$ . (We can put `current` or `lic[n]` among `simple[[n_i]]`'s if we want to use them.) The rule for the composite  $\varphi$  is given as follows. First we twist along the curve `simple[[n_1]]`  $k_1$  times<sup>2</sup>, then twist along the curve `simple[[n_{l-1}]]`  $k_{l-1}$  times and then  $\dots$ . The output of this command is the representation matrix of the action under the basis  $\{a_1, a_2, \dots, a_g, b_1, b_2, \dots, b_g\}$  given as follows:



## 4 The version for a surface with one boundary

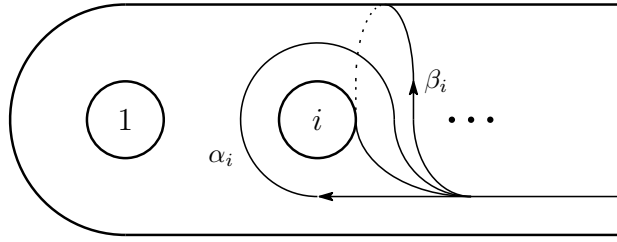
As mentioned in Section 2, we can run the version for a surface  $\Sigma_{g,1}$  with one boundary<sup>3</sup> by using `T4M7_*****_Sg1.nb`. The method for using this version is almost the same as the version for  $\Sigma_g$ . One of the differences is that we need not choose and change the genus of a surface.

### 4.1 Computing the action on $\pi_1(\Sigma_{g,1})$ by the composite of Dehn twists

In this version, we can compute the action of the composite of Dehn twists on the fundamental group  $\pi_1(\Sigma_{g,1})$ . In the program, the following set of generators for  $\pi_1(\Sigma_{g,1})$  is adopted.

<sup>2</sup>Negative integers mean negative twists.

<sup>3</sup>More precisely, we consider  $\lim_{g \rightarrow \infty} \Sigma_{g,1}$ .



The labels of the standard generators of  $\pi_1(\Sigma_{g,1})$

Note that  $\alpha_i$  (resp.  $\beta_i$ ) is sent to  $a_i$  (resp.  $b_i$ )  $\in H_1(\Sigma_g)$  by the abelianization map  $\pi_1(\Sigma_{g,1}) \rightarrow H_1(\Sigma_{g,1})$ . We order the generators of  $\pi_1(\Sigma_{g,1})$  by  $\alpha_1, \beta_1, \alpha_2, \dots, \beta_g$ . (This order is different from that of the basis of  $H_1(\Sigma_{g,1})$ )

The command

```
action[ {simple[ [n_1] ], k_1, simple[ [n_2] ], k_2, ...,
        simple[ [n_1] ], k_1 } ]
```

computes the action of the composite  $\varphi$  of Dehn twists on  $\pi_1(\Sigma_{g,1})$ . The rule for the composite  $\varphi$  is the same as in Section 3.10. The output of this command is the list

$$\{\varphi(\alpha_1), \varphi(\beta_1), \dots, \varphi(\alpha_g), \varphi(\beta_g)\}.$$

Each entry of the list consists of numbers corresponding to the generators of  $\pi_1(\Sigma_{g,1})$ . For example  $\{1, 2, 4, -3\}$  means  $\alpha_1\beta_1\beta_2\alpha_2^{-1}$ .

## 4.2 Computing some invariants of the composite of Dehn twists

The command

```
computations[ {simple[ [n_1] ], k_1, simple[ [n_2] ], k_2, ...,
              simple[ [n_1] ], k_1 } ],
```

computes the actions of the composite on  $\pi_1(\Sigma_{g,1})$  and  $H_1(\Sigma_{g,1})$  together with the image by the extended Johnson homomorphism<sup>4</sup> [3] simultaneously.

## References

- [1] K. Ahara, *How to draw a simple closed curve on a picture of Riemann surface*, Sūrikaiseikikenkyūsho Kōkyūroku 1022 (1997), 7–19
- [2] K. Ahara, *Teruaki*, you can download this software from <http://www.math.meiji.ac.jp/~ahara/teruaki.html>
- [3] S. Morita, *The extension of Johnson's homomorphism from the Torelli group to the mapping class group*, Invent. Math. 111 (1993), 197–224

<sup>4</sup>The extension of the Johnson homomorphism for the Torelli group to the mapping class group is *not* unique. The choice of the extension adopted in the program is different from that in [3].