A users guide to Flipper

Mark Bell

March 24, 2014

Flipper is a program for computing the action of mapping classes on laminations on a punctured surface using ideal triangulation coordinates. Flipper is currently under development and this users guide will be based on Flipper 0.2.3. Eventually it will also be able to construct the canonical triangulation of the surface bundle associated to pseudo-Anosov mapping classes.

1 Getting and starting Flipper

Remark. For the commands to setup and start Flipper, see Appendix A.

Flipper has been tested on Python 2.7 and Python 3.3 on Windows 7 and Ubuntu 13.10. Some of its features require exact arithmetic. You can install Flipper as a Sage module (tested with Sage 5.12 and 6.1.1) and use Sages exact arithmetic libraries. These appear to be significantly faster.

1.1 Getting Flipper

You can get the latest copy of flipper from `https://bitbucket.org/Mark_Bell/Flipper` or straight from the mercurial repository with the command:

```
> hg clone https://bitbucket.org/Mark_Bell/Flipper
```

1.2 Installing and starting Flipper under Python

To install Flipper, in a terminal run the following command inside of its folder:

```
> python setup.py install
```

If that fails because you do not have the necessary permission, you can install it locally instead using the command:

```
> python setup.py install --user
```

If you want, you can test your install by using the command:

```
> python setup.py test
```

This will list out varous tests that it is running and if they passed.

Finally, you can now start Flipper by running the command:

```
> python -m Flipper.app
```

1.3 Installing and starting Flipper under Sage

To install Flipper as a Sage module, in a terminal run the following command inside of its folder:

```
> sage -python setup.py install
```

If that fails because you do not have the necessary permission, you can install it locally instead using the command:

```
> sage -python setup.py install --user
```

Again, if you want, you can test your install by using the command:

```
> sage -python setup.py test
```

You can then start Flipper from within Sage by using the command

```
> sage -python -m Flipper.app
```

Note: The Flipper application requires Tkinter. You can obtain this on Ubuntu by using the command:

```
> sudo apt-get install python-tk
```

If Sage does not recognise your Tkinter install you may initally see an error such as:

```
Error: no module named _tkinter
```

You can fix this by installing the tcl/tk development library and then rebuilding Sage's Python. On Ubuntu you can do this using the commands:

```
> sudo apt-get install tk8.5-dev
> sage -f python
```

1.4 Creating executables

Flipper also includes a freeze file. So if you have cx_Freeze installed as a Python module you can use the command:

```
> python freeze.py build
```

to create an executable within the build directory. As with most cx_Freeze distutils scripts, you can also build a Windows installer by doing:

```
> python freeze.py bdist_msi
```

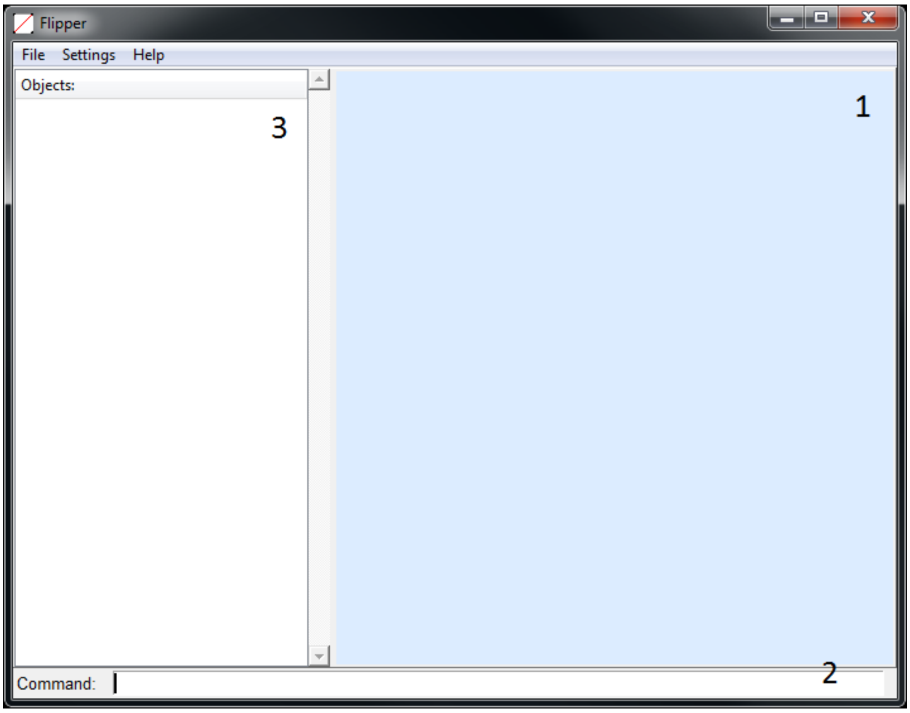
and a Mac disk image by doing:

```
> python freeze.py bdist_dmg
```

At some point the BitBucket repository may also include precompiled binaries.

2 Getting Started

This is the main window of Flipper. It has a canvas for drawing on (1), a command bar for entering instructions (2), a list of known laminations and mapping classes (3).



We now describe how to perform each of these steps in detail. First note that in each stage the currently selected object is highlighted in red. You can cancel your current selection at any time by clicking on the object again, pressing `Escape` or double clicking. Additionally you can delete the currently selected object by pressing either `Delete` or `Backspace`.

2.1 Creating a triangulation

To create a triangulation, click on the canvas to create vertices. Click on two in succession to connect them via an edge. You cannot add an edge if it would meet the interior of an existing edge. Click on two edges, each of which are part of exactly one triangle, in succession to identify them. Clicking on an identified edge will destroy the identification.

Flipper automatically adds triangles between any triple of vertices each of which is pairwise connected via an edge.

The triangulation is *complete* if each edge is either contained in two triangles or is contained in one triangle and is identified with another. Once the triangulation is complete Flipper will switch to interpret clicks as drawing a lamination. You can force Flipper to place a vertex or select an edge, even if the triangulation is complete, by holding `Shift` while clicking.

2.2 Adding laminations

Once the surface drawn is complete you can start drawing laminations on it. Click on the canvas to start drawing. Click on the canvas again to extend it through the current point. To finish drawing a section of curve press `Escape` or double click.

You can remove the last point currently being drawn by either pressing `Delete` or `Backspace`. You should make sure to draw transverse to the underlying triangulation.

The currently drawn lamination can be added to the list of known laminations by using the `lamination <name>` command.

2.3 Adding mapping classes

There are currently three different basic types of mapping classes that can be created.

- Dehn twist - Use the command `twist [name]` to create a twist about the currently drawn lamination. This can only be done if the lamination is actually a curve. Alternatively, if this curve is in the list of known laminations and is listed as half twistable then double click on `Twistable: True`.
- Half twist - Use the command `half [name]` to create a half twist about the currently drawn lamination. This can only be done if the lamination is actually a curve. Alternatively, if this curve is in the list of known laminations and is listed as twistable then double click on `Half twistable: True`.
- Isometry - Use the command `isometry <from>:<to> <from>:<to> <from>:<to> [name]` to create a mapping class induced by the symmetry of the underlying triangulation which sends each edge numbered `<from>` to the paired `<to>`.

Finally, you can use the command `compose <composition> [name]` to create a new mapping class which is the composition of several existing mapping classes

Remark. Currently Flipper is only capable of performing Dehn twists and half twists about *good curves*, where every complementary region contains at least one puncture. Hence, for example, it cannot perform a half twist on a twice marked surface. Additionally, these also adds the twisting curve to the list of known laminations under the same name.

2.4 Object properties

Stored laminations and mapping classes appear in the object list. Clicking on a object will show more information about it and actions involving it. If a property can be computed in polynomial time then it is automatically listed. If a property can be computed in exponential time with known exponent then it is listed as `?`. If a property can be computed in exponential time but the exponent is unknown then it is listed as `??`. You can force Flipper to compute any unknown property by double clicking on it.

2.4.1 Lamination properties

The properties and methods of a known lamination are:

- Show - Renders this lamination on the current triangulation.
- Multicurve: True / False
- Twistable: True / False
- Half twistable: True / False
- Filling: True / False

2.4.2 Mapping class properties

The properties and methods of a known mapping class are:

- Apply - Applies this mapping class to the currently drawn lamination.
- Apply inverse - Applies the inverse of this mapping class to the currently drawn lamination.
- Order: Infinite / $\mathbb{N}$
- Type: Periodic / Reducible / Pseudo-Anosov
- Invariant lamination - Finds a lamination which is projectively invariant under this mapping class.

A Quick start

These are the quickest ways to get, install, test and start Flipper under various systems.

A.1 Under Python

```
> hg clone https://bitbucket.org/Mark_Bell/Flipper
> cd ./Flipper
> python setup.py install --user
> python setup.py test
> python -m Flipper.app
```

A.2 Under Sage

```
> hg clone https://bitbucket.org/Mark_Bell/Flipper
> cd ./Flipper
> sage -python setup.py install
> sage -python setup.py test
> sage -python -m Flipper.app
```

B All commands

The main Flipper commands are:

- `lamination [name]` - Stores the currently drawn lamination.
- `twist [name]` - Stores a left Dehn twist about the currently drawn lamination. This lamination must be a curve.
- `half [name]` - Stores a left half twist about the currently drawn lamination. This lamination must be a curve bounding a twice marked disk.
- `isometry <from>:<to> <from>:<to> <from>:<to> [name]` - Stores the mapping class induced by the isometry of the underlying triangulation.
- `apply <mapping class>` - Applies `<mapping class>` to the currently drawn lamination.
- `order <mapping class>` - Computes the order of `<mapping class>`.
- `type <mapping class>` - Computes the Nielsen–Thurston type of `<mapping class>`.
- `invariant_lamination <mapping class>` - Computes a lamination which is projectively invariant under `<mapping class>` along with its dilatation. This functionality requires a symbolic computation library.
- `bundle <mapping class>` - Builds the canonical bundle associated to the maximal splitting sequence of `<mapping class>`. This functionality requires a symbolic computation library.

Flipper also has commands for quickly creating standard triangulations:

- `ngon <specification>` - Creates a regular ngon with either the number of sides or side gluings given by `<specification>`.
- `rngon <specification>` - Creates a regular ngon with either the number of sides or side gluings given by `<specification>` where the underlying triangulation has rotational symmetry.

The following commands are also available:

- `information` - Reports information about the underlying surface such as genus and Euler characteristic.
- `tighten` - Tightens the currently draw curve to the underlying triangulation.

- `erase` - Erases the current curve.
- `zoom` - Scales the current drawing to fill the canvas.

Arguments given to Flipper commands must meet the Conventions:

- `<name>` must be an string of letters, numbers and underscores starting with a letter.
- `<mapping class>` must be a string of mapping class names and inverse names, given by their swapcase, each separated by a '.'. This is read left to right to agree with the order of composition.
- `<specification>` must either be a number or a string of letters.