A users guide to flipper

Mark Bell

September 20, 2014

Flipper is a program for computing the action of mapping classes on laminations on a punctured surface using ideal triangulation coordinates. It can decide the Nielsen–Thurston type of a given mapping class and, for pseudo-Anosov mapping classes, construct a layered, veering triangulation of their mapping torus, as described by Agol [3]. Flipper is currently under development and this users guide will be based on flipper 0.5.2.

Flipper can be run as a Python 2, Python 3 or Sage Python module [5]. It has been tested on:

- Python 2.7.8 on Windows 7,
- Python 3.3.4 on Windows 7,
- Python 3.4.1 on Windows 7,
- Python 2.7.6 on Ubuntu 14.04,
- Python 3.4.0 on Ubuntu 14.04, and
- Sage 6.3 on Ubuntu 14.04.

Some of its features require exact arithmetic. Although these calculations can be done in pure Python, Sage's libraries appear to be significantly faster.

1 Getting and starting flipper

1.1 Quick start

To get, install, test and start the flipper application under Python using Mercurial [2].

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

1.2 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
```

1.3 Dependencies

The flipper kernel has no required dependencies. However, some of the examples using the kernel require SnapPy [4]. Additionally, the flipper application requires Tkinter. You can obtain this on Ubuntu by using the command:

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

If you are running flipper as a Sage Python module then your Tkinter install may not be recognised and you may see an error such as: `Error: no module named _tkinter`. To fix this install the tcl/tk development library and then rebuild Sage's Python. On Ubuntu you can do this using the commands:

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

Note that there are several known issues with tcl/Tk on Mac OS X, see `https://www.python.org/download/mac/tcltk`.

1.4 Using Setup.py

The easiest way to install flipper is to use the included setup.py Python script.

To install flipper use the command:

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

To install flipper locally use the command:

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

This can be used if you do not have the necessary permission to use the previous command.

To test the installed version of flipper use the command:

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

This will list out varous tests run and if they passed. Any failed tests will be listed at the end.

To profile the installed version of flipper use the command:

```
> python setup.py profile
```

To use the setup.py script under sage use the `sage -python` option. For example, to install flipper use the command:

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

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

```
> python -m flipper.app
```

1.5 Using freeze.py

Flipper includes a freeze file for building binaries. The following commands all require you to have cx_Freeze [1] installed as a Python module.

To build an executable within the build directory use the command:

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

To build a Windows installer use the command:

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

To build a Mac disk image use the command:

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

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

2 Getting Started

Once installed the flipper application can be started by using the command:

```
> python -m flipper.app
```

The main window of flipper is shown in Figure 1. It has a canvas for drawing on (1) and a list of known laminations and mapping classes (2).

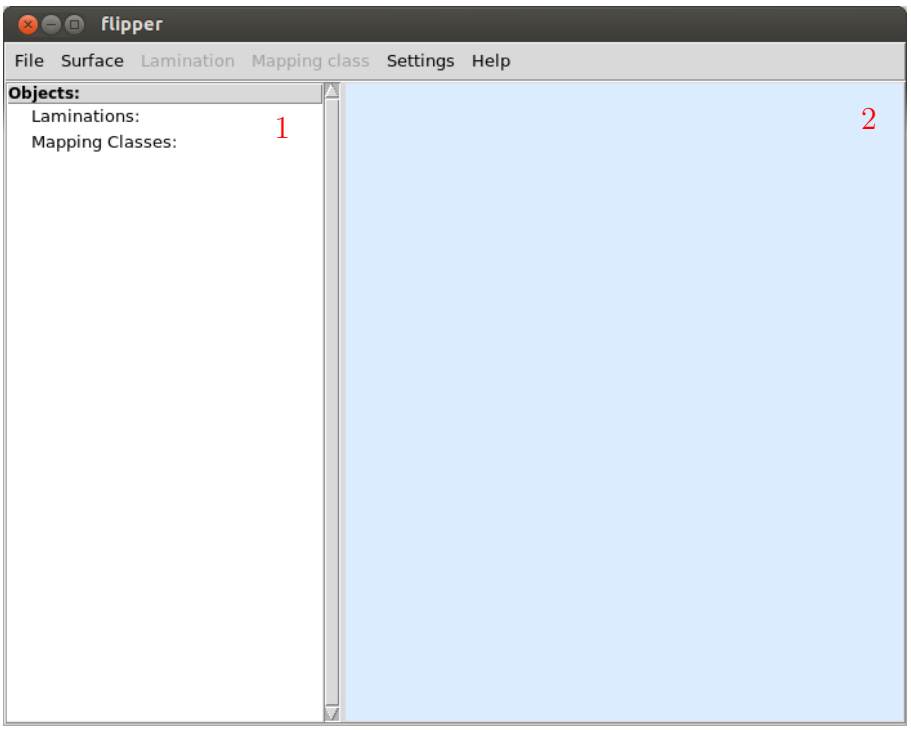

Figure 1: The main flipper window.

Remark 2.1. 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 action `Laminations > Store`.

2.3 Adding mapping classes

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

- Dehn twist - A Dehn twist about the currently drawn lamination can be created by using the action `Mapping classes > Store... > Twist`. 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 `Twistable: True`.
- Half twist - A half twist about the currently drawn lamination can be created by using the action `Mapping classes > Store... > Half twist`. 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 `Half twistable: True`.
- Isometry - An isometry of the underlying triangulation can be created by using the action `Mapping classes > Store... > Isometry`. The isometry should be specified by a string of the from `<from>:<to> <from>:<to> <from>:<to>` indicating which edges should be sent to which edges. If this string does not specify a unique isometry then an arbitary one will be chosen.
- Composition - A composition of existing mapping classes can be created by using the action `Mapping classes > Store... > Composition`. The composition should be specified by a string of mapping class names and inverse names separated by periods.

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.

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 otherwise it will be listed as `?`. You can ask flipper to compute any unknown property by double clicking on it. Some properties prevent other actions from being taken and so are listed as `x`. For example, flipper cannot compute the invariant lamination of a periodic mapping class.

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 - The Nielsen–Thurston type of this mapping class.
- Invariant lamination - Finds a lamination which is projectively invariant under this mapping class.

References

[1] cx_Freeze, for cross platform freezing of python scripts. Available at `http://cx-freeze.sourceforge.net/`. [2]

[2] Mercurial, a distributed source control management tool. Available at `http://mercurial.selenic.com/`. [1]

[3] Ian Agol. Ideal triangulations of pseudo-Anosov mapping tori. In *Topology and geometry in dimension three*, volume 560 of *Contemp. Math.*, pages 1–17. Amer. Math. Soc., Providence, RI, 2011. [1]

[4] Marc Culler, Nathan M. Dunfield, and Jeffrey R. Weeks. SnapPy, a computer program for studying the topology of 3-manifolds. Available at `http://snappy.computop.org` (20/09/2014). [1]

[5] W. A. Stein et al. *Sage Mathematics Software (Version 6.3.1)*. The Sage Development Team, 2014. `http://www.sagemath.org`. [1]