INSTRUCTIONS FOR SOAPY

This package is intended to facilitate computations involving Seifert fibered rational homology spheres (henceforth called ‘SFS’). For notations and conventions of the 3-manifolds and their invariants, see “Conventions” below. Part of the code is based on code written for MAGMA by Karakurt [Kar], which in turn is an implementation of Némethi’s algorithm to compute HF^+ [Ném05].

The package contains the class `soapy.SFS`, as well as its subclasses `soapy.Lens`, `soapy.Prism` and `soapy.Brieskorn`.

`soapy.SFS`

An object representing the SFS $Y(e; a_1/b_1, \ldots, a_n/b_n)$ (where, $a_i, b_i \neq 0$ for $i = 1, \ldots, n$.) is created by passing the tuple `(e;a_1/b_1,...,a_n/b_n)` to the constructor of `soapy.SFS`.

Alternatively, this can be done by passing the function `soapy.SFS.from_plumbing` the weights of a plumbing description of $Y(e; a_1/b_1, \ldots, a_n/b_n)$, as specified by a star-shaped tree with integer weights (see “Example” below).

If `Y` is a `soapy.SFS` object, calling `-Y` corresponds to orientation-reversal. Two objects are equal in this class if they represent orientation-preservingly homeomorphic 3-manifolds. An object is ‘less than or equal to’ another, if the pair passes the correction term obstruction to there existing a ribbon rational homology cobordism from the first object to the other (see “Obstruction” below). An object is ‘less than’ another if it is less than, but not equal, to the other.

The class contains the objects `soapy.three_sphere` and `soapy.poincare_sphere`, representing S^3 and the Poincaré homology sphere, respectively.

List of attributes:

`params`	List of integer coefficients representing the SFS specified. These do not necessarily coincide with the input parameters, but rather are normalized in such a way that the corresponding integer surgery diagram is definite.
`central_weight`	The weight of the central vertex of the normalized surgery description of the SFS specified.
`branch_weights`	Tuple containing the rational surgery coefficients of the exceptional fibers of the SFS specified.

List of methods:

`fractional_branch_weights()`	Returns the fractional parts of the framings on the exceptional fibers.
`euler_number()`	Returns the orbifold Euler number of the SFS specified.
`number_of_exceptional_fibers()`	Returns the number of exceptional fibers of the SFS specified.
`to_plumbing()`	Returns the definite plumbing (equivalently: an integral surgery description) corresponding to the SFS specified.
`seifert_invariants()`	Returns the Seifert invariants of the SFS specified.
`linking_matrix()`	Returns the linking matrix of the SFS specified.
`first_homology()`	Returns the first homology of the SFS specified.
`order_of_first_homology()`	Returns the order of the first homology of the SFS specified.
`spinc_to_HF()`	Computes HF^+ of the SFS specified in each spin^c-structure. The $\mathbb{Z}[U]$-module-structure of HF^+ is encoded as a dictionary of the format {'order of $\mathbb{Z}[U]$-module-summand' : 'list of bottommost gradings of all $\mathbb{Z}[U]$-module-summands of that order'}.
`print_HF()`	Prints HF^+ of the SFS specified in a more legible manner.
`correction_terms()`	Returns a list of the corrections terms of the SFS specified.
`is_lspace()`	Checks whether or not the SFS specified is a Heegaard Floer L-space.

`casson_walker()`	Computes the Casson-Walker invariant of the SFS specified.
`is_lens_space()`	Checks whether or not the SFS specified is homeomorphic to a lens space.
`to_lens_space()`	Transforms the SFS specified into the corresponding lens space.
`is_prism_mfld()`	Checks whether or not the SFS specified is homeomorphic to a prism manifold.
`to_prism_mfld()`	Transforms the SFS specified into the corresponding prism manifold.

List of functions:

`number_to_neg_cont_frac`	This computes the coefficients of the (-)-continued fraction expansion of a rational number p/q, for p, q coprime.
`number_to_pos_cont_frac`	This computes the coefficients of the (+)-continued fraction expansion of a rational number p/q, for p, q coprime.
`number_from_neg_cont_frac`	This computes (-)-continued fraction of an arbitrary list of coefficients.
`number_from_pos_cont_frac`	This computes (+)-continued fraction of an arbitrary list of coefficients.

soapy.Lens

This is a subclass of `soapy.SFS` representing lens spaces. In particular, any object belonging to this subclass inherits all methods defined for objects belonging to `soapy.SFS`.

An object representing the lens space $L(p, q)$ is created by passing the parameters `p` and `q` to the constructor of `soapy.Lens`. Alternatively, this can be done by passing the function `soapy.Lens.from_linear_lattice` the weights of a linear lattice giving a plumbing description of $L(p, q)$.

List of attributes:

`p`	The first parameter of the lens space specified, normalized to be greater than zero.
`q`	The second parameter of the lens space specified, normalized so that $p > q > 0$.

List of methods:

`to_SFS()`	Transforms the lens space specified into a SFS.
`to_linear_lattice(epsilon=-1)`	Returns the weights of the linear lattice bounded by the lens space specified. By default, the negative definite linear lattice is returned, unless epsilon is set to 1.

soapy.Prism

This is a subclass of `soapy.SFS` representing prism manifolds. In particular, any object belonging to this subclass inherits all methods defined for objects belonging to `soapy.SFS`.

An object representing the prism manifold $P(p, q)$ is created by passing the parameters `p` and `q` to the constructor of `soapy.Prism`.

List of attributes:

`p`	The first parameter of the prism manifold specified, normalized to be greater than 0.
`q`	The second parameter of the prism manifold specified.

List of methods:

`to_SFS()`	Transforms the prism manifold specified into a SFS.

soapy.Brieskorn

This is a subclass of `soapy.SFS` representing Brieskorn homology spheres. In particular, any object belonging to this subclass inherits all methods defined for objects belonging to `soapy.SFS`.

An object representing the Brieskorn homology sphere $\Sigma(a_1, \ldots, a_n)$ is created by passing the parameters `a_1` through `a_n` to the constructor of `soapy.Brieskorn`. If all entries of the tuple are negative, an object representing $-\Sigma(a_1, \ldots, a_n)$ is created.

List of attributes:

`coeffs`	The coefficients of the Brieskorn homology sphere specified.

List of methods:

`to_SFS()`	Transforms the Brieskorn homology sphere specified into a SFS.

Conventions.

- The Seifert fibered space $Y(e; a_1/b_1, \dots, a_n/b_n)$ is the 3-manifold with surgery diagram given in Figure 1.

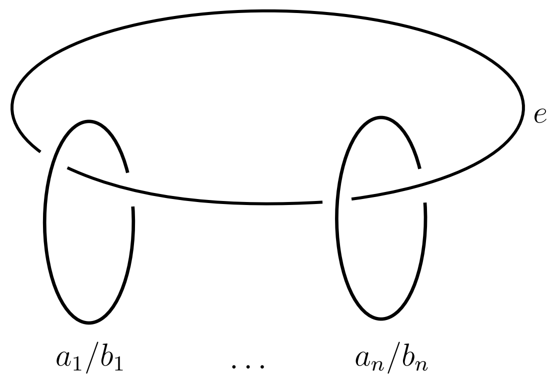

FIGURE 1

- The orbifold Euler number of $Y(e; a_1/b_1, \dots, a_n/b_n)$ is defined to be
$$e_{\mathrm{orb}}(Y) := e - \sum_{i=1}^{n} \frac{b_i}{a_i}.$$
- The lens space $L(p, q)$ is the oriented 3-manifold obtained by performing $-(p/q)$-framed Dehn surgery along the unknot in S^3. Defined for any pair p, q of non-zero coprime integers.
- The prism manifold $P(p, q)$ is defined to be $Y(-1; -2, -2, -p/q)$. Defined for any pair p, q of non-zero coprime integers. Contrary to some conventions, $p = \pm 1$ is allowed; note, however, that $P(1, n)$ is homeomorphic to the lens space $L(4n, 2n - 1)$.
- Brieskorn spheres are oriented such that $\Sigma(2, 3, 5)$ is the boundary of the negative definite plumbing along the E_8 Dynkin diagram (where each vertex has weight -2). $\Sigma(a_1, \dots, a_n)$ is defined for any sequence of positive, pairwise coprime integers.
- The Casson-Walker invariant is defined as in [Rus05], i.e.
$$\lambda(Y) = \frac{1}{|H_1(Y; \mathbb{Z})|} \sum_{\mathfrak{s} \in \mathrm{Spin}^c(Y)} \left(\chi(\mathrm{HF}^+_{\mathrm{red}}(Y)) - \frac{d(Y, \mathfrak{s})}{2} \right).$$

Obstruction. Given a pair of SFS's Y_1 and Y_2, the package implements the following obstruction to there being a ribbon rational homology cobordism from Y_1 to Y_2. If Y_1 admits a ribbon rational homology cobordism to Y_2, then

- $|H_1(Y_2; \mathbb{Z})| = u^2 \cdot |H_1(Y_1; \mathbb{Z})|$, for some $u \in \mathbb{Z}$; and

- each correction term of Y_1 appears u times among the correction terms of Y_2 (counted with multiplicity)

Example: Let $Y_1 = L(2,1)$ and $Y_2 = L(8,5)$. The lists of correction terms of Y_1 and Y_2 are $[-1/4,\ 1/4]$ and $[-3/8,\ -3/8,\ -1/4,\ -1/4,\ 1/4,\ 1/4,\ 5/8,\ 5/8]$, respectively. Since

$$|H_1(Y_2;\mathbb{Z})| = 8 = 2^2 \cdot 2 = 2^2 \cdot |H_1(Y_1;\mathbb{Z})|,$$

and because each correction term of Y_1 appears 2 times among those of Y_2, the pair passes the obstruction.

Note that, while this implements just an obstruction, pairs of lens spaces (and connected sums thereof) that admit a rational ribbon cobordism from one to the other have been classified in [Hub21]

Example. Calling `Y = soapy.SFS(2,2,1,3,2,11,9)` creates an object representing $Y(2;2,3/2,11/9)$. Starting from a plumbing diagram, this object can be created by calling `soapy.SFS.from_plumbing(2,[2],[2,2],[2,2,2,2,3])` (generally, the lists of weights along the branches must be specified starting at the central vertex). The coefficients of this plumbing can be found by calling `Y.to_plumbing()`. One can verify that Y is homeomorphic to $-\Sigma(2,3,11)$ e.g. by calling `Y == -soapy.Brieskorn(2,3,11)` or `Y == soapy.Brieskorn(-2,-3,-11)`.

Calling `Y.first_homology()` returns an empty tuple, thus verifying that $H_1(Y;\mathbb{Z})$ has no non-trivial summands, and hence that Y is an integral homology sphere.

Calling `Y.print_HF()` (or, equivalently, `sp.Brieskorn(2,3,11).print_HF()`) prints

```
HF^+(Y(2; 2, 3/2, 11/9)):
spin^c-structure:  (0, 0, 0, 0)
HF^+ = T_(-2) + Z(1)_(-2)
```

This is to be interpreted as saying that $\mathrm{HF}^+(Y)$ in its unique spin^c-structure consists of a tower whose bottommost grading is -2, together with a $\mathbb{Z}[U]$-torsion summand of order 1 whose bottommost grading is -2 as well. In other words,

$$\mathrm{HF}^+(-\Sigma(2,3,11),\mathfrak{s}) \cong \mathcal{T}^+_{(-2)} \oplus \mathbb{Z}_{(-2)}.$$

Accordingly, calling `Y.correction_terms()` and `Y.is_lspace` returns `(-2,)` and `False`, respectively.

References

[Hub21] Marius Huber. Ribbon cobordisms between lens spaces. *Pacific J. Math.*, 315(1):111–128, 2021.

[Kar] Çağri Karakurt. MAGMA code. http://web0.boun.edu.tr/cagri.karakurt/Research.html.

[Ném05] András Némethi. On the Ozsváth-Szabó invariant of negative definite plumbed 3-manifolds. *Geom. Topol.*, 9:991–1042, 2005.

[Rus05] Raif Rustamov. *On Heegaard Floer homology of three-manifolds.* ProQuest LLC, Ann Arbor, MI, 2005. Thesis (Ph.D.)–Princeton University.