Centro de Informática U·F·P·E

Sostenes Lins <sostenes@cin.ufpe.br>

your challenge

Craig Hodgson <craigdh@unimelb.edu.au> 24 de abril de 2013 04:23
Para: sostenes@cin.ufpe.br, llins@research.att.com
Cc: Craig Hodgson <craigdh@unimelb.edu.au>

Dear Sostenese and Lauro,
I just saw your recent preprint on the arXiv.

Assuming your framings 0, -1 and 0, 3 are with respect to the standard homological longitude and meridian, your manifolds both have hyperbolic volume approximately 7.36429600733. Is that correct?

If I'm using the correct framings, then these manifolds can be distinguished using SnapPea (or SnapPy or Snap) since the lengths of the shortest closed geodesics in their hyperbolic metric are different: approximately
1.01521038256 for manifold U[1466]
and
0.935920660503 for manifold U[1563].
(See output below.)

In general, the length spectrum is often a very good way of distinguishing hyperbolic 3-manifolds of equal volume.

SnapPea's algorithms for finding the shortest geodesics and for testing for isometry between closed hyperbolic 3-manifolds are described in my paper with Jeff Weeks:

Symmetries, isometries and length spectra of closed hyperbolic three-manifolds
Experiment. Math. Volume 3, Issue 4 (1994), 261-274.
available at http://projecteuclid.org/euclid.em/1048515809.

For your examples, SnapPy's initial triangulation starting from a link projection diagram has some negatively oriented tetrahedra for the Dehn fillings giving your manifolds. But some random retriangulations produced the positively oriented triangulations attached below. This guarantees that the correct hyperbolic structure has been found by SnapPy. These triangulations can also be used with Snap to find exact descriptions of the hyperbolic structure, see:

D. Coulson, O. Goodman, C. Hodgson C.D. and W. Neumann,
Computing arithmetic invariants of 3-manifolds,
Experimental Mathematics 9 (2000), 127--152.

Regards,
Craig Hodgson

--------------some output from SnapPy-------------------

```
In [1]: M1=Manifold("U1466good.tri")

In [2]: M1.solution_type()
Out[2]: 'all tetrahedra positively oriented'
```

In [3]: M1.volume()
Out[3]: 7.36429600733

In [4]: M1.length_spectrum(2) (Note: this gives the closed geodesics of length <2)
Out[4]:

mult	length	topology	parity
1	(1.0152103824828331+0.39992347315914334j)	circle	orientation-preserving
1	(1.1673837107362062+0.19109819097990566j)	circle	orientation-preserving
1	(1.2321098393958962+1.6534739471522792j)	circle	orientation-preserving
1	(1.2597293633113396-1.4083335638789467j)	circle	orientation-preserving
1	(1.3697334734542512+2.479103935573692j)	circle	orientation-preserving
1	(1.8886967845490497-1.7320420117261583j)	circle	orientation-preserving
1	(1.929726071471829+2.1906185915603116j)	circle	orientation-preserving

In [5]: M2=Manifold("U1563good.tri")

In [6]: M2.solution_type()
Out[6]: 'all tetrahedra positively oriented'

In [7]: M2.volume()
Out[7]: 7.36429600733

In [8]: M2.length_spectrum(2) (Note: this gives the closed geodesics of length <2)
Out[8]:

mult	length	topology	parity
1	(0.9359206605025168+2.333526236965665j)	circle	orientation-preserving
1	(1.0152103825669705+0.39992347317483584j)	circle	orientation-preserving
1	(1.0293408996867977-2.1393601309230634j)	circle	orientation-preserving
1	(1.3697334734585518+2.479103935579211j)	circle	orientation-preserving
1	(1.8136237783604314+1.8659212289429898j)	circle	orientation-preserving
1	(1.8190461159026123+2.816213866723708j)	circle	orientation-preserving
1	(1.8467211339821399+0.7409198883807969j)	circle	orientation-preserving
1	(1.8778314147303738+1.7595474114031422j)	circle	orientation-preserving
1	(1.9053043023591834-2.21205635261203j)	circle	orientation-preserving

----

2 anexos

U1563good.tri
5K

U1466good.tri
4K