Soap by Tesseral Spherical Harmonis

Aki Morooka

July 23, 2020

Abstract

This is a documentation on the derivatives of the SOAP spectrum.

Steps

In short:

step 1) Open real_spherical_harmonics.wxmx, and run the function definitions.

step 2) run the cell with

```
h[i,j] := rlY(i-1,-(i-j),x,y,z);
genmatrix(h,5,5*2-1);
```

where "5" is the l in spherical harmonics, and can be replaced to an arbitrary number.

step 3) Copy the matrix as Matlab/Octave, paste it into a plane text file.

step 4) Delete char "[" and "]". And replace char ";" to ","

step 5) In mat2c.py change the "test.txt" to the file name you used to copy paste the matrix, in line

```
mat = np.loadtxt("test.txt", dtype = "U16384") # Change me
```

step 6) Run python3 mat2c.py, this will produce files tesseral_mat_nopow.txt and tesseral_mat_nopow.txt.

step 7) Finally, run python3 printMat.py. This will produce the file

```
"finalSoapFunctionsWithoutSqrtPi3.txt"
```

CAUTION: This will erase the file first.

step 8) Copy paste the list of soap functions to a c or cpp source file. And multiplty them by

$$\sqrt{(\pi)^3} \tag{1}$$

The only thing missing now is the radial basis, but you only need to multiply the α and β terms, and the exponent term.

CAUTION: Do not forget to multiply $\pi^{3/2}$ to the functions.

Structure of the Final File

- the first line is the l=0 term, then the next 3 lines are the l=1 tern, then the next 5 are l=2 terms and so on.
- the m's run from $m_{\min}$ to $m_{\max}$ by 1 step every line startin with new l.
- The $rr = r^2$ and "to the power of" are just numbers after a variable, for example $x2 = x^2$, $rr^2 = r^4$ and so on. The reason for this is to precalculate the powers so it is not done in real time, which would significantly slow down the code if done.
- There are obvious, and not so obvoius patterns, where you can reuse the computations from before, but in general, just precalculating the power terms would be fast enough.
- In c or cpp, the x, y, z and "rr will be the distance from the soap-point to the atoms, which would be looped over i, so it would be x[i], rr[i], x2[i], rr3[i] and so on. You can run

  ```
  python3 putIs.py
  ```

  to get the file with the [i]'s,

  ```
  "finalSoapFunctionsWithoutSqrtPi3Is.txt"
  ```

Final notes

For the dericvatie forms, there are slight modifications that need to be made, but the basic idea is the same. Please read SoapDerivaties.pdf for the implementation.