Testing CLASS using Nosetests

Thomas Tram

Institute of Gravitation and Cosmology

October 31, 2014

Testing the code

When should we test our code?
- Every time we have made a big change.
- Before each release.

What we should test?
- Things we fear may not work?
- Things we are sure will work?
- Things we can not live without?

Testing the code

When should we test our code?
- Every time we have made a big change.
- Before each release.

What we should test?
- Things we fear may not work?
- Things we are sure will work?
- Things we can not live without?

Answer...
- All of the above!

The input is the devil

Optional inputs

- If a code takes 0 input parameters it can be tested by running it once.
- If a code takes N optional input parameters, we got 2^N possible combinations of input options!
- This number will increase further if each input parameter must be tested with more than one parameter!

Optional inputs

- This can not be done manually. (Wrapper!)

Automated testing

python/test_class.py

```
CLASS_INPUT['Mnu'] = (
    [{'N_ur': 0.0, 'N_ncdm': 1, 'm_ncdm': 0.06, 'deg_ncdm': 3.0},
     {'N_ur': 1.5, 'N_ncdm': 1, 'm_ncdm': 0.03, 'deg_ncdm': 1.5}],
    'normal')

CLASS_INPUT['Curvature'] = (
    [{'Omega_k': 0.01},
     {'Omega_k': -0.01}],
    'normal')

CLASS_INPUT['Output_spectra'] = (
    [{'output': 'mPk', 'P_k_max_1/Mpc': 10},
     {'output': 'tCl'},
     {'output': 'tCl pCl lCl'},
     {'output': 'mPk tCl lCl', 'P_k_max_1/Mpc': 10},
     {'output': 'nCl sCl'},
     {'output': 'tCl pCl lCl nCl sCl'}],
    'power')
```

How to add a test

Adding a test

If you have modified the code and want to add a test, you are in one of two possible situations:

1. The new input is compatible with any existing input. This is the standard case when adding a new species for instance. Add a new entry in `CLASS_INPUT`.

2. The new input is incompatible with one of the existing inputs. You should add a new entry in the incompatible entry.

How to add a test

Adding a test

1. Compatible. Add a new entry in `CLASS_INPUT`.
2. Incompatible. Add a new entry in the incompatible entry.

python/test_class.py

```python
CLASS_INPUT['Mnu'] = (
    [{'N_ur': 0.0, 'N_ncdm': 1, 'm_ncdm': 0.06, 'deg_ncdm': 3.0},
     {'N_ur': 1.5, 'N_ncdm': 1, 'm_ncdm': 0.03, 'deg_ncdm': 1.5}],
    'normal')

CLASS_INPUT['Curvature'] = (
    [{'Omega_k': 0.01},
     {'Omega_k': -0.01}],
    'normal')
```

What is being tested?

We test that...

- CLASS is not making a segmentation fault..
- CLASS will exit gracefully if the input is incompatible.
- CLASS will not throw an error during computation.
- All C_ℓ^{XX} agree between synchronous and Newtonian gauge to the advertised accuracy.
- $P(k)$ agree between synchronous and Newtonian gauge to the advertised accuracy.

You can try

You can try if you want (and have `nose` installed) by typing:
```
nosetests python/test_class.py
```