forward

October 22, 2021

[1]:
```
import phoebe
from phoebe import u,c
import numpy as np
```

[2]:
```
logger = phoebe.logger(clevel='WARNING')
```

[3]:
```
b = phoebe.default_binary()
```

[4]:
```
import numpy as np
times_kepler, fluxes_kepler, sigma_kepler = np.loadtxt('data/lcflux_kepler_ltf.
 ↪txt', unpack=True)
```

[5]:
```
times_rvA, rvA, sigma_rvA = np.loadtxt('data/rvA.dat', unpack=True)
times_rvB, rvB, sigma_rvB = np.loadtxt('data/rvB.dat', unpack=True)
```

[6]:
```
compute_times = phoebe.linspace(np.min(times_kepler), np.max(times_kepler),␣
 ↪10000)

b.add_dataset('lc', times=times_kepler, fluxes=fluxes_kepler,␣
 ↪sigmas=sigma_kepler, dataset='lc_kepler', passband='Kepler:mean',␣
 ↪compute_times=compute_times)
b.add_dataset('rv', times={'primary': times_rvA, 'secondary': times_rvB},␣
 ↪rvs={'primary': rvA, 'secondary': rvB}, sigmas={'primary': sigma_rvA,␣
 ↪'secondary': sigma_rvB}, dataset='rv01', compute_times=compute_times)
```

[6]:
```
<ParameterSet: 81 parameters | contexts: dataset, compute, constraint, figure>
```

[7]:
```
b.filter(qualifier='pblum_mode', dataset='lc_kepler').
 ↪set_value_all(value='decoupled')
b.filter(qualifier='pblum', dataset='lc_kepler').set_value_all(value=4*np.pi)
b.filter(qualifier='pblum', dataset='lc_kepler', component='secondary').
 ↪set_value(value=4*np.pi * 0.0170083351)
print(b.filter(qualifier='pblum'))
```

```
ParameterSet: 2 parameters
   pblum@primary@lc_kepler@dat…: 12.566370614359172 W
   pblum@secondary@lc_kepler@d…: 0.2137330423998137 W
```

Setup approximations:

[8]:
```
b.set_value(qualifier='distortion_method', component='secondary', value='roche')
b.set_value(qualifier='distortion_method', component='primary', value='roche')
b.set_value(qualifier='rv_method', component='secondary', value='flux-weighted')
b.set_value(qualifier='rv_method', component='primary', value='flux-weighted')
```

Set paramaters for system:

[9]:
```
b.flip_constraint(qualifier='mass', component='primary', solve_for='q')
b.flip_constraint(qualifier='mass', component='secondary', solve_for='period')
b.flip_constraint(qualifier='t0_perpass', component='binary',␣
 ↪solve_for='t0_supconj')

b.set_value(qualifier='mass', component='primary', value=1.2544743268)
b.set_value(qualifier='mass', component='secondary', value=0.8137981554)
b.set_value(qualifier='requiv', component='primary', value=7.5322533155)
b.set_value(qualifier='requiv', component='secondary', value=0.7539093043)
b.set_value(qualifier='sma', component='binary', value=85.1857110174)
b.set_value(qualifier='per0', component='binary', value=349.35517451582)      ␣
 ↪# using perilong instead of arg of peri
b.set_value(qualifier='ecc', component='binary', value=0.2564703004)
b.set_value(qualifier='incl', component='binary', value=89.2064583383)
b.set_value(qualifier='t0_perpass', component='binary', value=54998.2349211194␣
 ↪- 0.1972987964*63.32709529966483)

b.set_value(qualifier='vgamma', context='system', value=16.025)
```

[10]:
```
print(b.get_value(qualifier='period', component='binary', context='component'))
```

```
63.32709529966483
```

[11]:
```
b.set_value(qualifier='teff', component='primary', value=5042)
b.set_value(qualifier='teff', component='secondary', value=5660)
```

[12]:
```
b.set_value('l3_mode', dataset='lc_kepler', value='fraction')
b['l3_frac@lc_kepler@dataset'] = 0.0029667000
```

Check compute

[13]:
```
b.add_compute(compute='initial', overwrite=True)
b.run_checks(compute='initial')
b.compute_pblums(compute='initial', pbflux=True)
```

[13]:
```
{'pblum@primary@lc_kepler': <Quantity 12.56637061 W>,
 'pblum@secondary@lc_kepler': <Quantity 0.21373304 W>,
 'pbflux@lc_kepler': <Quantity 1.01700834 W / m2>,
 'pblum@primary@rv01': <Quantity 1.3313406e+27 W>,
```

```
 'pblum@secondary@rv01': <Quantity 2.34975751e+25 W>,
 'pbflux@rv01': <Quantity 1.07814597e+26 W / m2>}
```

[14]:
```
b['ntriangles@primary@initial'] = 7000
b['ntriangles@secondary@initial'] = 4000
```

[15]:
```
b.run_compute(compute='initial')
```

```
100%|      | 10000/10000 [1:16:25<00:00,  2.18it/s]
```

[15]:
```
<ParameterSet: 9 parameters | kinds: lc, rv>
```

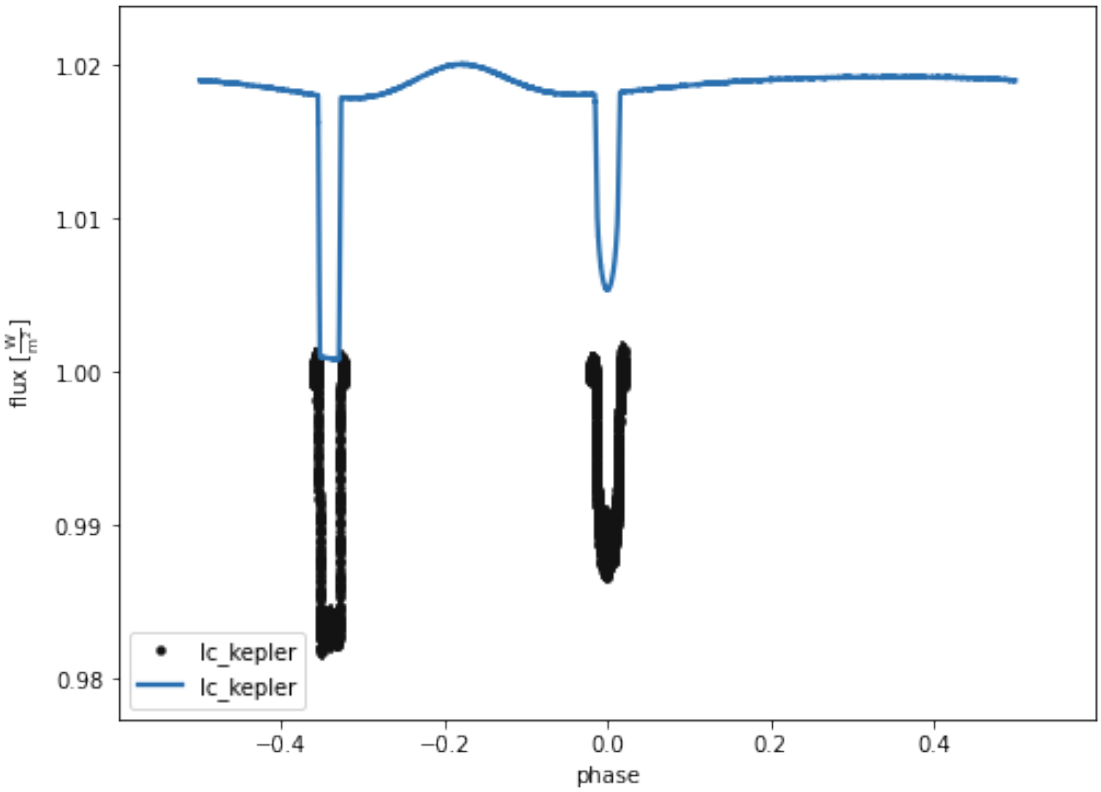
Run compute using MPI multiprocessing

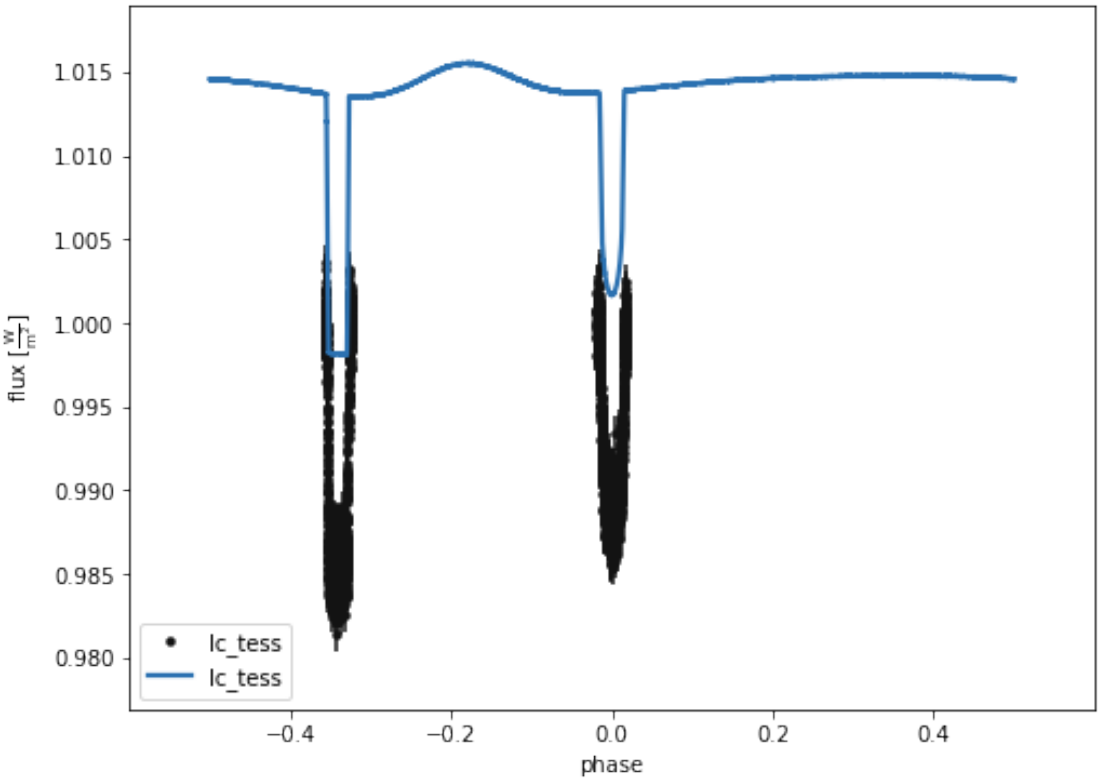
[16]:
```
b.export_compute('compute_script.py', compute='initial')
! mpirun -np 1 python -m mpi4py compute_script.py
b.import_model('compute_script.py.out')
```

```
Traceback (most recent call last):
  File "/usr/lib/python3.6/runpy.py", line 193, in _run_module_as_main
    "__main__", mod_spec)
  File "/usr/lib/python3.6/runpy.py", line 85, in _run_code
    exec(code, run_globals)
  File "/home/sinkbaek/default-py-env/lib/python3.6/site-
packages/mpi4py/__main__.py", line 7, in <module>
    main()
  File "/home/sinkbaek/default-py-env/lib/python3.6/site-
packages/mpi4py/run.py", line 196, in main
    run_command_line(args)
  File "/home/sinkbaek/default-py-env/lib/python3.6/site-
packages/mpi4py/run.py", line 47, in run_command_line
    run_path(sys.argv[0], run_name='__main__')
  File "/usr/lib/python3.6/runpy.py", line 263, in run_path
    pkg_name=pkg_name, script_name=fname)
  File "/usr/lib/python3.6/runpy.py", line 96, in _run_module_code
    mod_name, mod_spec, pkg_name, script_name)
  File "/usr/lib/python3.6/runpy.py", line 85, in _run_code
    exec(code, run_globals)
  File "compute_script.py", line 2, in <module>
    import phoebe; import json;
  File "/home/sinkbaek/default-py-env/lib/python3.6/site-
packages/phoebe/__init__.py", line 217, in <module>
    mpi = MPI()
  File "/home/sinkbaek/default-py-env/lib/python3.6/site-
packages/phoebe/__init__.py", line 121, in __init__
    raise ImportError("need more than 1 processor to run with mpi")
ImportError: need more than 1 processor to run with mpi
application called MPI_Abort(MPI_COMM_WORLD, 1) - process 0
```

[16]: <ParameterSet: 11 parameters | kinds: lc, rv>

[17]:
```
b.filter(dataset='lc_kepler').plot(show=True, legend=True, x='phases',␣
 ↪s={'dataset': 0.01}, fmt={'dataset': 'k.'}, elinewidth={'dataset': 0.05})
b.filter(dataset='lc_tess').plot(show=True, legend=True, x='phases',␣
 ↪s={'dataset': 0.01}, fmt={'dataset': 'k.'}, elinewidth={'dataset': 0.05})
```





[17]:
```
(<autofig.figure.Figure | 1 axes | 2 call(s)>,
 <Figure size 576x432 with 1 Axes>)
```

[16]:
```
# print(b.filter(dataset='lc_kepler'))
# print(b['fluxes@lc_kepler@latest@model'].get_value())
save_array = np.array([b['compute_phases@lc_kepler@dataset'].get_value(),␣
 ↪b['fluxes@lc_kepler@latest@model'].get_value(),␣
 ↪b['compute_times@lc_kepler@dataset'].get_value()]).T
np.savetxt('fwmodel.txt', save_array)
```