PHOEBEB 2 - Introduction

SpaceInn Binary Workshop
KU Leuven – April 13, 2015
Kyle Conroy

Steven Bloemen, Jonas Debosscher, Angela Kochoska, Michel Hillen
Andrej Prsa, Bert Pablo, Joe Giammarco, Jana Nemravova,
Pieter Degroote, Gal Matijevic, Kelly Hambleton
Outline

- PHOEBE 1 (legacy) vs PHOEBE 2
- Forward Model
- Other random not well-tested cool things PHOEBE 2 can do
- Fitting with MCMC
<table>
<thead>
<tr>
<th>PHOEBE 1.0 (Legacy)</th>
<th>PHOEBE 2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binaries (OC, SD, D, ELV)</td>
<td>Doesn't handle OCs well (yet)</td>
</tr>
<tr>
<td>Binaries only</td>
<td>Single stars, binaries, planetary systems, hierarchical multiples</td>
</tr>
<tr>
<td>Simple grid (limited precision)</td>
<td>Mesh with adaptive subdivision</td>
</tr>
<tr>
<td>Library, scripter, GUI</td>
<td>Backend (python, C, fortran)</td>
</tr>
<tr>
<td></td>
<td>Frontend (python)</td>
</tr>
<tr>
<td></td>
<td>No GUI (yet)</td>
</tr>
<tr>
<td>DC, simplex, gradient fitting only (built-in)</td>
<td>MCMC, Imfit, leastsq, etc</td>
</tr>
<tr>
<td>LC, RV</td>
<td>Multiple observables</td>
</tr>
<tr>
<td>Eclipses, reflection, spots</td>
<td>Eclipses, reflection, spots, pulsations, beaming/boosting, ltte</td>
</tr>
<tr>
<td>Stable</td>
<td>Alpha-release</td>
</tr>
</tbody>
</table>
PHOEBE 1: Grid
PHOEBE 1: Grid
PHOEBE 2:
Mesh and subdivision
PHOEBE 2

Data modelling tool + laboratory

Planet
Multiple system
Rotation
Thick disk
Binary
Spot
Magnetic field
Pulsation
...

Data
spectropolarimetry
light curve
multicolour photometry
astrometry
radial velocity curve
interferometry
image

pymc
emcee
Imfit
minuit
genetic
grid
PHOEBE 2: Additional Physics

- Doppler Effects (boosting/beaming)
  - Relativistic beaming (+ only)
  - Photon arrival rate (+ or -)
  - Passband-dependence (+ or -)
- Pulsations (including tidally induced)
- Roemer (Light-time) delay
- Other system objects (planets, disks, rings)
- Multiple systems (hierarchical and not)
- Other observables besides just LCs and RVs
Getting Started: PHOEBE 2.0

Backend:
- Written in C, Fortran, and Python (interface in Python)
- Parameters and ParameterSets
- Low-level functionality

Frontend:
- Python module
- Single picklable object (bundle) to hold all options (system, data, computing options)
- Hides some non-intuitive structure required by the backend
- High-level functionality and convenience functions
Getting Started: Forward Model

```python
>>> import phoebe
>>> import numpy as np
>>> import matplotlib.pyplot as plt
```
Getting Started: Forward Model

```python
>>> b=phoebe.Bundle()
>>> print b
>>> print b.summary()
>>> print b.list()
```

```
new_system (BodyBag)
|  
| +----------> primary (BinaryRocheStar)
|  
| +----------> secondary(BinaryRocheStar)
```
Getting Started: Forward Model

```python
>>> print b['incl']
80.0
>>> b['incl'] = 90
>>> print b['incl']
90.0
>>> print b['teff@primary']
6000.0
>>> print b['teff{TAB}']
b['teff@component@primary@system@Bundle'
b['teff@component@secondary@system@Bundle
```
Getting Started: Forward Model

```python
>>> b.lc_fromarrays(
    time=np.linspace(0, 3, 100),
    dataref='mylc')
>>> b.run_compute()
>>> b.plot_syn('mylc', fmt='b-')
>>> plt.show()
```
Rotating Single Star

```python
>>> b = phoebe.Bundle('sun')
>>> b.lc_fromarrays(time=[0], dataref='mylc')
>>> b.plot_mesh('mylc', time=0, select='teff')
>>> plt.show()
```
Rotating Single Star

```python
>>> b.plot_mesh('mylc', time=0, select='teff',
               cmap='blackbody_proj')
```
Rotating Single Star

```python
>>> b.plot_mesh('mylc', time=0, select='logg')
```
Rotating Single Star

```python
>>> b.plot_mesh('mylc', time=0, select='rv')
```
Binary System

![Graphs and diagrams related to binary system analysis.](binary_anim.py)
>>> b = phoebe.Bundle()

>>> b.rv_fromarrays(
    time=np.arange(0,1,0.01),
    objref=["primary","secondary"],
    dataref='rvs')

>>> b.sp_fromarrays(
    time=np.arange(0,1,0.01),
    wavelength=np.linspace(399,401,500),
    dataref='spectra')
>>> b.attach_plot_syn('rvs@primary',
    fmt='b-', axesref='rvs',
    axesloc=(2,2,4), highlight_fmt='bo')

>>> b.attach_plot_syn('rvs@secondary',
    fmt='r-', highlight_fmt='ro')

>>> for t in b['time@spsyn']:
    ...    b.draw('fig', time=t)
Binary System
>>> print b['detailed@compute']

mesh_rescale 1.0 -- Scaling factor for mesh densities
heating True -- Allow irradiators to heat ...
refl True -- Allow irradiated Bodies to reflect ...
refl_num 1 -- Number of reflections
ltt True -- Correct for light time travel effects
subdiv_alg edge -- Subdivision algorithm
subdiv_num 3 -- Number of subdivisions
eclipse_alg binary -- Type of eclipse algorithm
boosting_alg local -- Type of boosting algorithm
irradiation_alg point_source -- Type of irradiation algorithm
mpilabel None -- Label of the MPI params to use

>>> b['refl@detailed@compute'] = False
>>> b.run_compute('detailed', heating=False)
Forward Model: Current Support

**Supported and should work:**
- D/SD binaries *(known heating/reflection bug)*
- Lightcurves and Radial Velocities

**Implemented but not well tested and/or limited frontend support:**
- Single stars, hierarchical multiples
- Spectra, interferometry, ETVs, polarimetry, …
- Spots, pulsations, rings & accretion disks

**Not implemented yet:**
- Non-hierarchical multiples (N-body)
- Tidally induced pulsations
- Migrating spots
Forward Model: Pulsations

```python
>>> b.attach_ps(
    phoebe.PS(
        context='puls',
        label='modepulsations'),
    'Sun')

>>> b['freq@puls'] = 3.97
>>> b['l@puls'] = 4
>>> b['m@puls'] = 4
```
Forward Model: Pulsations

pulse_mode.py
Forward Model: Pulsations
Accretion Disks
Hierarchical Systems
Just for fun
Just for fun
Just for fun
Getting Started: Fitting

- Multiple integrated (wrapped) fitting algorithms
- emcee (MCMC) has been tested the most and is preferred
- Bayesian approach – always deal with priors and posteriors
- Determine posteriors by simultaneously fitting model to all observables
- Frontend: framework consistent with compute
>>> b.set_prior('incl',
    distribution='uniform',
    lower=80,
    upper=90)

>>> b.set_adjust('incl', True)

>>> b.set_adjust('offset@fakedata')

>>> b.set_adjust('scale@fakedata')
Fitting:
Setup

```python
>>> b.lc_fromfile('fakedata.lc',
    columns=["time", "flux"],
    dataref='fakedata')

>>> b.rv_fromfile('fakedata.rv',
    columns=["time", "rv", "rv"],
    objref=[None, 'primary',
             'secondary'],
    dataref='fakerv')
```
>>> b.add_fitting(context='fitting:emcee',
                 label='mcmc')

>>> print b['mcmc@fitting']

erats 1000            --   Number of iterations
burn 0                --   Burn-in parameter
thin 1                --   Thinning parameter
walkers 6             --   Number of walkers
init_from prior       --   Initialize walkers from priors, posteriors or
                         previous run
incremental False     --   Add results to previously computed chain file
acc_frac 0.0          --   Acceptance fraction
label emcee           --   Fit run name
computelabel preview  --   Label of the compute params to use
mpilabel None         --   Label of the MPI params to use or blank for None

>>> b.add_mpi(context='mpi:slurm', label='server',
             time=20, np=16)

>>> b['mpilabel@mcmc'] = 'server'

>>> b.save_pickle('pre_fitting.bundle')

>>> b.run_fitting('mcmc')

fit_mcmc.py
Fitting: load results

```python
>>> b = phoebe.Bundle('pre_fitting.bundle')
>>> b.feedback_fromfile(
    'mcmc.mcmc_chain.dat',
    ongoing=True,
    accept_feedback=True,
    lnproblim=None)
```
Fitting: plot results

```python
>>> b['mcmc@feedback'].plot_summary()
```
Fitting: plot results

```python
>>> b['mcmc@feedback'].plot_logp()
```
Fitting: plot results

```python
>>> b.plot_syn('fakedata')
```
Fitting: Current Support

Supported and should work:
- emcee (MCMC)
- MPI, MPI:torque
- MPI:slurm (need latest update for frontend support)

Implemented but not well tested:
- pymc
- lmfit (leastsq, nelder, lbfgsb, anneal, powell, cg, newton, cobyla, slsqp)
- grid

Not implemented yet:
- Differential Corrections (soon!)
Conclusion

● PHOEBE 2 is powerful but not fully tested (alpha-release)

● Some known (and probably unknown) science bugs

● For simple binaries (or trustable science) – consider using PHOEBE 1 for now

● Submit BUG REPORTS
  sourceforge.net/p/phoebe/bugs

● Feel free to email (or join) the development team:
  phoebe-devel@lists.sourceforge.net
  phoebe-project.org