

# JupiterMag

---

Jupiter magnetic field model

## Installation

---

(not implemented yet) Install using `pip3` :

```
pip3 install JupiterMag --user
```

Or using this repo:

```
git clone https://github.com/mattkjames7/JupiterMag.git
cd JupiterMag
python3 setup.py bdist_wheel
pip3 install dist/JupiterMag-x.x.x-py3-none-any.whl --user
```

## Usage

---

### Internal Field

A number of internal field models are included (see [here](#) for more information) and can be accessed via the `JupiterMag.Internal` submodule, e.g.:

```
import JupiterMag as jm

#configure model to use VIP4 in polar coords (r,t,p)
jm.Internal.Config(Model="vip4",CartesianIn=False,CartesianOut=False)
Br,Bt,Bp = jm.Internal.Field(r,t,p)

#or use jrm33 in cartesian coordinates (x,y,z)
jm.Internal.Config(Model="jrm33",CartesianIn=True,CartesianOut=True)
Bx,By,Bz = jm.Internal.Field(x,y,z)
```

All coordinates are either in planetary radii ( `x,y,z,r` ) or radians ( `t,p` ).

### External Field

Currently the only external field source included is the Con2020 field (see [here](#))

This works in a similar way to the internal field, e.g.:

```
#configure model
jm.Con2020.Config(equation_type='analytic')
Bx,By,Bz = jm.Con2020.Field(x,y,z)
```

## Tracing

There is an object for field tracing:

```
#be sure to configure external field model prior to tracing
jm.Con2020.Config(equation_type='analytic')
#this may also become necessary with internal models in future, e.g.
#setting the model degree

#create trace object, pass starting position(s) x0,y0,z0
T = jm.TraceField(x0,y0,z0,IntModel='jrm09',ExtModel='Con2020')

#plot a trace (ind is the index of the trace to plot)
T.PlotRhoZ(ind)
```