Bryant Pepe
PHYS-265
Final Project

The great SWAMPE in the sky: Shallow Water Atmospheric Model for Exoplanets

1-4) Package name and function, selection reasoning, age, and maintenance

The Shallow Water Atmospheric Model for Exoplanets (SWAMPE V 1.0.0) uses a shallow-water model projected on a sphere to model upper-level zonal winds and geopotentials by latitude in the atmospheres of exoplanets. I chose this package as it reflects a special interest of mine, and pertains directly to the goals of my major(s). A relatively new development, no publication had used SWAMPE directly in its research until this year (Chen et al 2025). As the field of exoplanetary science expands and deepens, the usage of this package is likely to increase. In December 2022, SWAMPE appeared on JOSS in Landgren's "SWAMPE: A Shallow-Water Atmospheric Model in Python for Exoplanets", soon entering it's 3rd year of availability. The most recent activity on SWAMPE's github, however, occurred two years ago, indicating that regular updates have likely ceased at this time

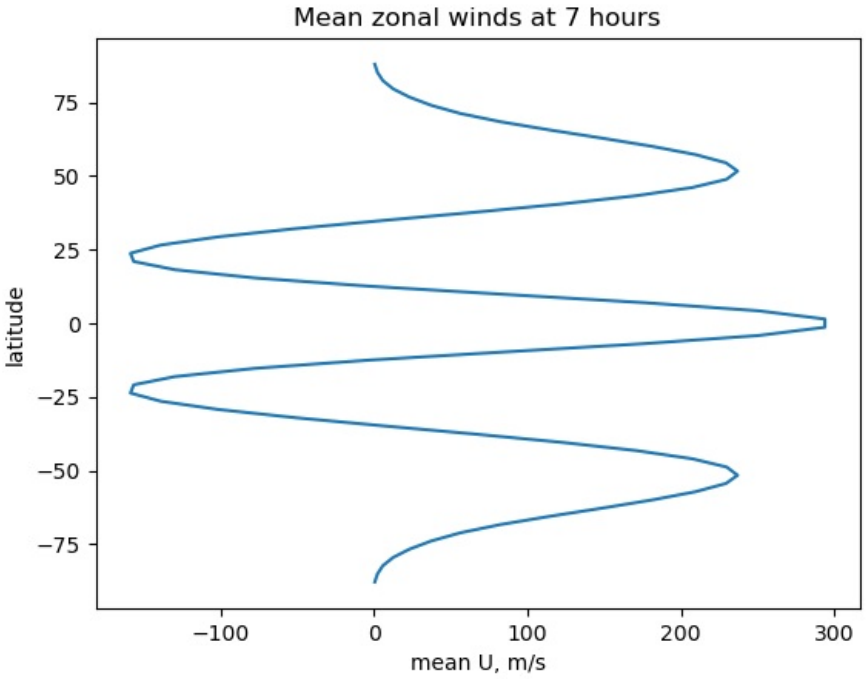
5-9) Ease of interface, installation, source code, usage in other packages, and code medium

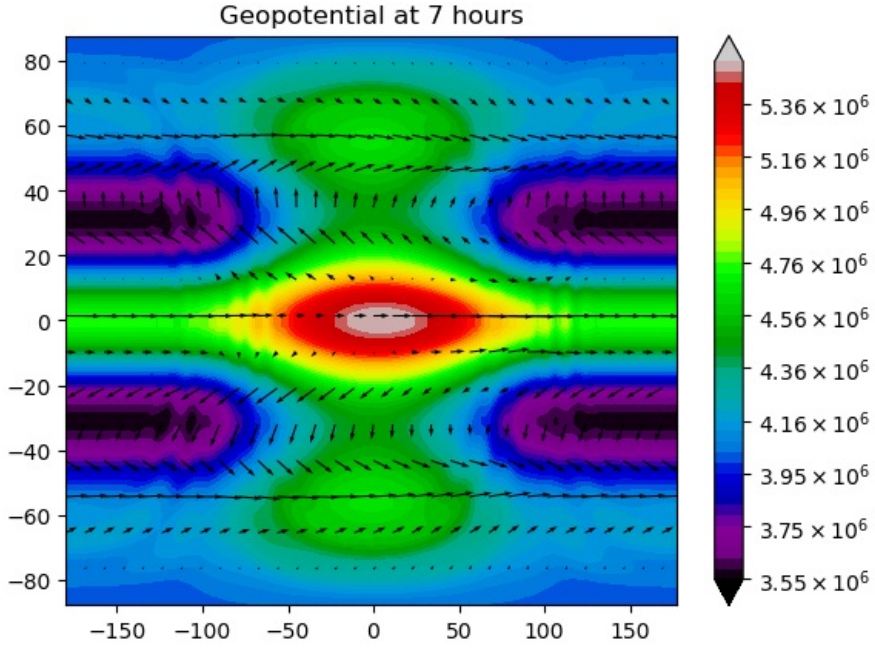
SWAMPE is an efficient, approachable plug-and-play package, wherein all inputs are fed into vector equations coded into the package and depicted by graphical modeling of the resulting fields. The package installs using a simple "!pip install SWAMPE" command, and checks for system requirements upon installation. The source code is available through the package's github account, found at https://github.com/kathlandgren/SWAMPE/tree/main/docs/source. SWAMPE appears to be a standalone package. As a purely surface-level evaluation using idealized conditions on a planetary body disregarding mass, it is not a perfect tool for atmospheric modeling, but offers a generalized look at idealized zonal upper-atmospheric circulation. This package can be used as python script in a notebook, but is encouraged to be used by commandline or web interface due to the long timescales required to generate most models. As such, the rms wind plot when run in a notebook does not reach a steady state oscillation before the model breaks down.

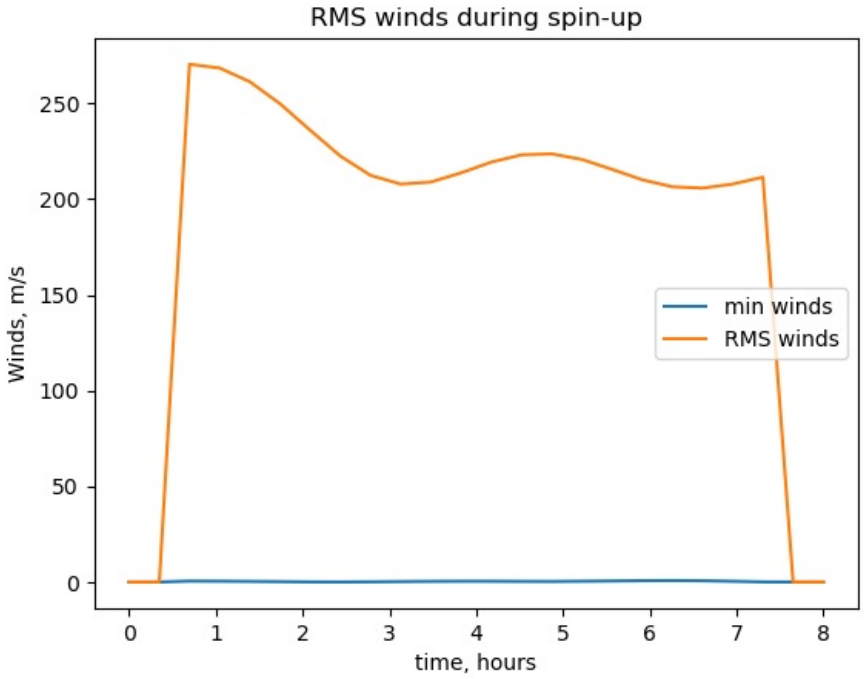
10/12) Code example & Figure example

In [1]:
```
import numpy as np
import SWAMPE as sw

a = 5.82*10**(7) #Saturn radius m

omega = 1.63*10**(-4) #rotation rate rad/s

Phibar = 4*10**(6) #ref geopotential height m^2/s^2

dt = 1200 #timestep (s)
tmax = 24 # number of timesteps for simulation

M=42 #spectral resolution

DPhieq = Phibar
taurad = 3600*24 #one day radiative timescale
taudrag = 10*3600*24 #ten day radiative timescale

plotflag=True
plotfreq = 22 #output: every n time steps

sw.run_model(M, dt, tmax, Phibar, omega, a, taurad=taurad, taudrag=taudrag, DPhieq=DPhieq, plotflag=plotflag, p
```

```
t=2, 8.333333333333334% complete
t=4, 16.666666666666668% complete
t=6, 25.0% complete
t=8, 33.333333333333336% complete
t=10, 41.666666666666664% complete
t=12, 50.0% complete
t=14, 58.333333333333336% complete
t=16, 66.66666666666667% complete
t=18, 75.0% complete
t=20, 83.33333333333333% complete
t=22, 91.66666666666667% complete
```


Mean zonal winds at 7 hours
latitude
mean U, m/s
75
50
25
0
−25
−50
−75
−100
0
100
200
300


Geopotential at 7 hours
80
60
40
20
0
−20
−40
−60
−80
−150
−100
−50
0
50
100
150
5.36 × 10^6
5.16 × 10^6
4.96 × 10^6
4.76 × 10^6
4.56 × 10^6
4.36 × 10^6
4.16 × 10^6
3.95 × 10^6
3.75 × 10^6
3.55 × 10^6




```
GCM run completed!
```

11-15) Figure generation, code purity, input & output parameters

Figures are automatically generated by the model based upon the parameters imposed on equations included in the package, and can be easily saved using "savemyfig=True" and coding custom paths, but can also be made individually using specific timesteps and data generated and pulled from the model. Figures of specific timestamps can furthermore be generated from any point in the simulation, and the package is capable of generating graphic image files.

Figures generated include a 1D plot of zonal winds by latitude, a colormap and vector plot of geopotential at the selected altitude, and the minimum and rms winds for the model as it initializes.

This code is pure Python but is based on Fortran 3D global circulation models utilizing curl and divergence of vector fields, among other computations. Package inputs are planetary radius in km, sidereal rate in radians per second, number of timesteps in seconds, maximum time of simulation, and radiative timescales in seconds. Output, along with the three plots, are lat-lon zonal wind component U, lat-lon meridional wind component V, lat-lon geopotential field , uniformly spaced longitudes , vorticy coefficient , and timestamps of simulation snapshots. Outputs can be saved externally and hardcoded into the simulation.

16-20) Unit tests, reliability, main package dependence, and citation/documentation

SWAMPE is benchmarked against end-to end tests from standard test sets for numerical shallow waters solovers and against strongly irradiated hot Jupiters (Williamson and Drake 1992, Perez-Becker and Showman 2013, Landgren 2022). The code is dependable as long as time stepping is set such that an overspin can be avoided in notebook settings, and the package will generate reasonably reliable results of a shallow water model, including a steady state oscillation of rms winds. SWAMPE requires numpy, scipy, imageio, matplotlib, and jupyter, among others appearing upon install to run correctly, and a comprehensive, easy to follow user guide is available. Authors request citation of their JOSS paper if SWAMPE is used in research.

21) References

> Landgren et al., (2022). SWAMPE: A Shallow-Water Atmospheric Model in Python for Exoplanets. Journal of Open Source Software, 7(80), 4872, https://doi.org/10.21105/joss.04872 https://swampe.readthedocs.io/en/latest/index.html

22) Package references

> Chen, Z., Ji, J., Chen, G., Yan, F., & Tan, X. (2025). Asymmetry and dynamical constraints in two-limbs retrieval of WASP-39 B inferring from JWST Data. The Astronomical Journal, 169(6), 294. https://doi.org/10.3847/1538-3881/adc803

> Landgren, E., & Strogatz, S. (2022). Models of varying complexity: From voter networks to extrasolar planets (dissertation). Models of varying complexity: from voter networks to extrasolar planets. ProQuest Information and Learning, Ann Arbor, MI.

23) New python method requirements

No new Python or coding techniques were required to achieve initial operational capability of this package apart from learning how the package itself operates.