Rijksuniversiteit Groningen

Masters Thesis

Computational Fluid Dynamics of Hot Jupiters

Using Shallow Water Initial Value Problems with Dedalus 3

Abstract

We present a qualitative study of short-timescale atmospheric simulations of hot Jupiters, using an idealised model of a highly stratified atmosphere through the use of Shallow Water equations. To this end, we use Dedalus 3, a sparse tau spectral solver, to solve for the initial value problem of a hot Jupiter experiencing height forcing as an analogue to heating from its parent star. Testing the numerical framework Dedalus 3 on the benchmark for a mid-latitude perturbed jet from J. Galewsky et al. (2004), we find a close match, supporting the use of Dedalus 3 for Shallow Water Models. Constructed viscous Shallow Water Models models using zonal wind velocity profiles, as measured by JWST with the F164N filter, are ran over several spectral resolutions to test for numerical convergence when perturbed and report machine precision across differing timesteps for the vorticity field. Lastly, height forcing is introduced to the exoplanet's atmosphere to simulate close proximity to its parent star, running over 2000 h with timesteps of 20 s, with both the JWST wind profile and one without initial velocity parameters. We find that both cases result in strong polar vortices with opposing rotation between the two models, along with the JWST wind profile model displaying a more chaotic system with larger eddies.

rijksuniversiteit groningen

Author:
J.H. Bonhof
S4045742

Supervisors:
Q. Changeat
Dr. F.F.S van der Tak

Contents

1 Introduction

Exoplanetary studies are a relatively new field of science for astronomy as a whole, but many advancements have been made within this field to be able to observe and study planets outside our own solar system. One specific species of exoplanets called hot Jupiters is of particular interest due to its ease of detection compared to other types of planets. Even moreso, hot Jupiters are the first class of exoplanets to be detected around main sequence stars (M. Mayor & D. Queloz, 1995). While many exoplanets have been documented across various catalogues, such as the NASA exoplanets archive (J. L. Christiansen et al., 2025) and the Open Exoplanet Catalogue (H. Rein, 2012), specific properties of the planets themselves are still a major topic of discussion. Studying the atmosphere for one could be done by looking at the system's spectroscopy, a method that has been used ever since 2002, where the atmosphere was detected through transits around HD 209458 due to a deviation in sodium absorption within the planet's spectrum (D. Charbonneau et al., 2002). However, theoretical models can be used as well to construct simulations of the system.

One approach to do this would be to model from existing measurements of a planet we can directly observe such as Jupiter, and then adjust properties to model for a gas giant closely orbiting the star. Using these initial conditions, an Initial Value Problem (IVP) can be solved through a high precision numerical framework (J. Y. K. Cho & L. M. Polvani, 1996; J. Y.-K. Cho et al., 2003; J. Galewsky et al., 2004; T. D. Komacek & A. P. Showman, 2016; T. D. Komacek & X. Tan, 2018). With this, differing models can be tested for accuracy and in what manners they differ from each other. For this thesis, this approach is used using a spectral solver called Dedalus 3 (K. J. Burns et al., 2020) to test for the significance of the initial wind conditions of a planet that is then exposed to stellar radiation up close, contributing to existing atmospheric models and what assumptions can be made about the wind profile of planets.

1.1 Exoplanetary Physics

In order to successfully model an exoplanetary atmosphere, it is first necessary to understand what advances have been made for these models. J. Y.-K. Cho et al. (2003) already modelled for HD 209458b, a hot Jupiter. While typically for hot Jupiters the planet's temperature profile would be expected to have a clear day and night side(T. D. Komacek & X. Tan, 2018), the results from the aforementioned paper shows a larger gradient, visible in figure 1. The deviation from the expected strong day/night temperature difference would be due to more complex dynamics of the planet's atmosphere, which is additionally supported by more recent measurements of hot Jupiters (M. Zhang et al., 2018; L. Kreidberg et al., 2018).

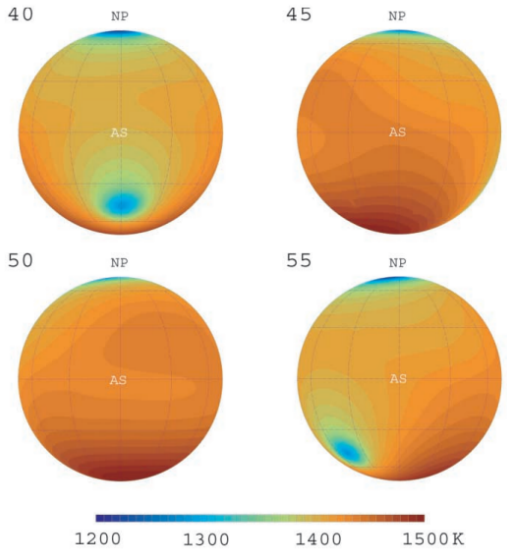

Figure 1: Temperature profiles of HD 209458b displayed over various days of the planet, where the amount of days is listed in the corner of each in the plots. Figure taken from (J. Y.-K. Cho et al., 2003)

To support these models, there would need to be grounds of observations of exoplanets as well such that these models can be used to predict measurements of said exoplanets. As mentioned earlier, spectroscopy can be used to gain several parameters from a planet's atmosphere, but the larger issue of measuring the planet persists. While methods like direct imaging do allow for direct measurements of the planet, this largely works for planets sufficiently distant from their parent star such that the star's point spread function does not pollute the signal of

the planet(J. T. Wright & B. S. Gaudi, 2013; M. Kaushik et al., 2024). As the interest of this paper is in Hot Jupiters and the atmospheric effect of that type of planet in particular, direct imaging would not be a good fit. The method of choice for measuring hot Jupiter exoplanets would be through studying phase curves from planetary transits. Exoplanets detected through transits would be biased for planets with short orbit around its parent star due to its increased probability of being detected (H. A. Knutson et al., 2012). Additionally, exoplanets at this small radius are often tidally locked due to the timescale for this tidal locking being far smaller than the typical age of systems with fully formed exoplanets (T. Guillot et al., 1996; P. Bodenheimer et al., 2001). As such, this method and its phase curves is highly favoured for Hot Jupiters. The following section will delve further into this specific field and its properties.

1.2 Phase Curves

Direct transits of exoplanets have been the method leading to the vast majority of detected exoplanets up to current day, as can been seen in figure 2. This method uses the exoplanet directly crossing the star as seen from earth, reducing the observed flux from the star as a result. This reduction can then be mapped as a phase curve, that displays the change in the observed flux over time, including both the direct transit of the planet in front of the star, as well as the eclipse of the planet through passing behind the star from earth's point of view. An example of a phase curve can be seen in figure 4, with both transit and eclipse visible. Of note in the example is the offset between the maximum radiative flux and the eclipse depth. For a symmetrical atmosphere on the exoplanet, the eclipse would have no offset with the maximum flux (J. C. Schwartz et al., 2017). due to the brightest section of the planet at that moment behind directly obscured by the star. In the case of asymmetry due to a planet's atmospheric dynamics however, an offset can be seen between the eclipse and maximum flux. It is possible to neglect this offset, though it will lead to a lowered flux of the nightside (D. Perez-Becker & A. P. Showman, 2013). Figure 3 shows this decreased flux as well as that the phase offset due to atmospheric dynamics has a significant impact on observations.

Comparing properties of the phase curve such as the shape of the transit, difference in measured temperature between the day- and nightside from the observed flux, and the shape of the phase curve with respect to the minimum and maximum radiative flux allows for large-scale properties of the exoplanet to be calculated. As a result however of the relative flux difference and time in-between transits and eclipses, phase curves can only indicate large-scale properties of exoplanetary atmospheres, and interference through gravitational effects between the planet and host star further increases uncertainties (A. Shporer & R. Hu, 2015). Values determined from this such as optical thickness, atmospheric pressure and radiative timescale can be used to construct Initial Value Problems to more robustly support theoretical models of the planets.

1.3 Initial Value Problems

Hydrodynamical systems as a whole are not easily solved to a high precision without numerical frameworks that can support them (S. A. Orszag, 1970). Instead of solving for an analytical solution of the equations, numerical frameworks are used to accurately approach the solution (S. A. Orszag, 1970; J. Y. K. Cho & L. M. Polvani, 1996; J. Y.-K. Cho et al., 2003; J. Galewsky et al., 2004; K. Julien & M. Watson, 2009). Within general uses of physics and applied mathematics, cases where the problem contains an Ordinary Differential Equation (ODE) can be solved numerically by solving a model for the ODE. Here a set initial condition is used to resolve the ODE problem over an evolving parameter (e.g. time being evolved over discrete steps). As these models rely on a strict definition of their initial conditions, these cases are defined as Initial Value Problems (IVP). Solving IVPs has been especially relevant across many fields of astronomy, not only for exoplanetary atmospheres(J. Y. K. Cho & L. M. Polvani, 1996; J. Y.-K. Cho et al., 2003; J. Galewsky et al., 2004; A. P. Showman et al., 2008; J. W. Skinner & J. Y.-K. Cho, 2022), but also for galactic models and simulations of cosmology (G. Efstathiou & J. W. Eastwood, 1981; D. Nelson et al., 2019).

For exoplanetary atmospheres, approaches for IVPs

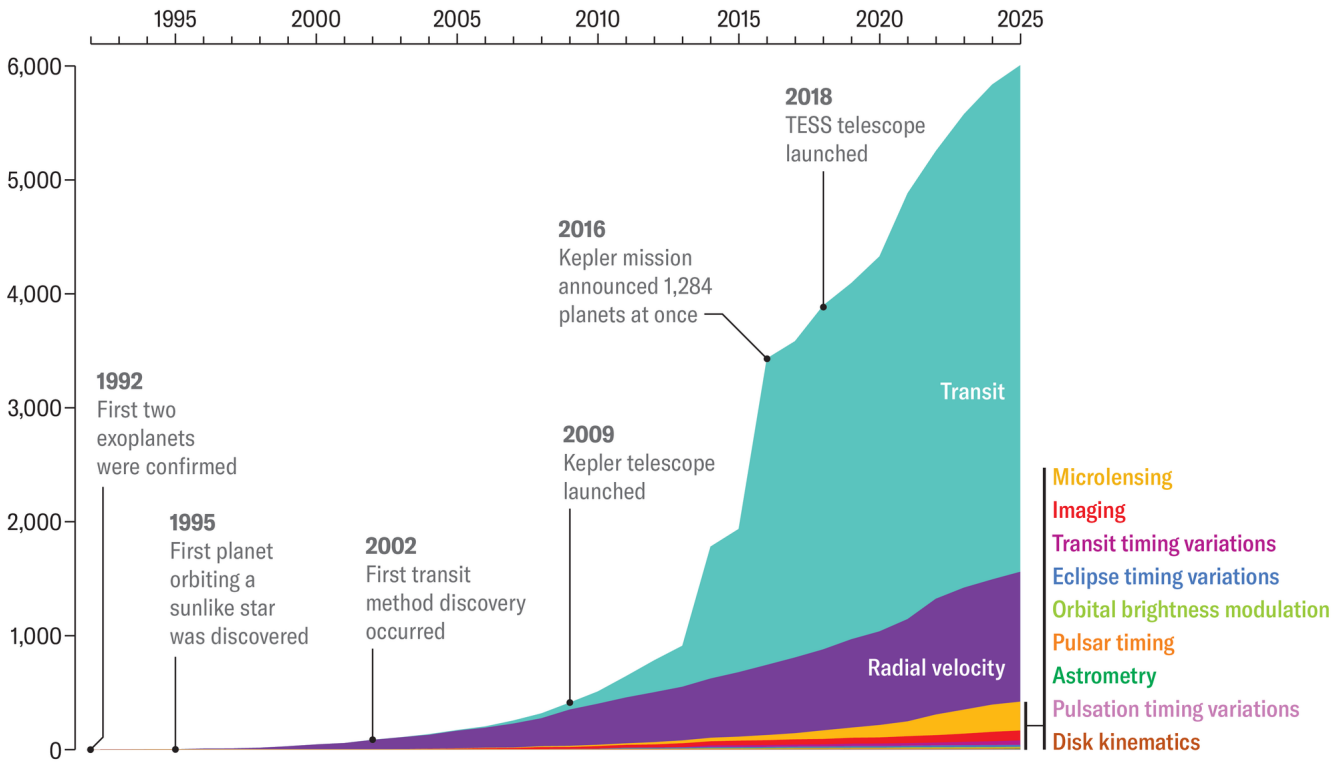

Figure 2: Cumulative Distribution of detected exoplanets, divided by the detection method. Taken from Amanda Montañez using the NASA Exoplanet Archive (J. L. Christiansen et al., 2025)

have both be done through studying atmospheric dynamics found on earth (M. Juckes & M. McIntyre, 1987), and by taking theoretical laws of hydrodynamics to constructs models from the ground up (J. Y.-K. Cho & L. M. Polvani, 1996; J. Galewsky et al., 2004). As this research is focused on models of hot Jupiters, extrapolating from earth would not work, due to the difference in composition and scale of the planets. As such, the approach of modelling from hydrodynamics to a feasible model is preferred, though the initial conditions still are to be estimated accordingly. The specifics to this will be explained within section 2.6 of the Methodology. The IVPs of exoplanetary atmospheres however are not expressed as Ordinary Differential Equations, and instead through Partial Differential Equations, as will be elaborated upon in the following section.

1.4 Partial Differential Equations

To define, Partial Differential Equations (PDEs) differ from ODEs through containing more than one independent variable, with partial derivatives being present only in PDEs. Because of these properties, ODEs are a type of simplified PDE, while many PDEs have intricate solutions or a complete lack of general solution. One such equation that does not have a general solution is the Navier-Stokes equation, especially relevant for this research due to its central use in fluid simulations:

$$\rho\frac{D\mathbf{u}}{Dt} = -\nabla p + \nabla\cdot(\mu[\nabla\mathbf{u} + (\nabla\mathbf{u})^T - \frac{2}{3}(\nabla\cdot\mathbf{u})\mathbf{I}]) + \nabla[\xi(\nabla\cdot\mathbf{u})] + \rho\mathbf{a}$$

Here ρ is density, $\frac{D}{Dt}$ is the material derivative, $\mathbf{u}$ is the flow velocity, p is pressure, μ is the dynamic

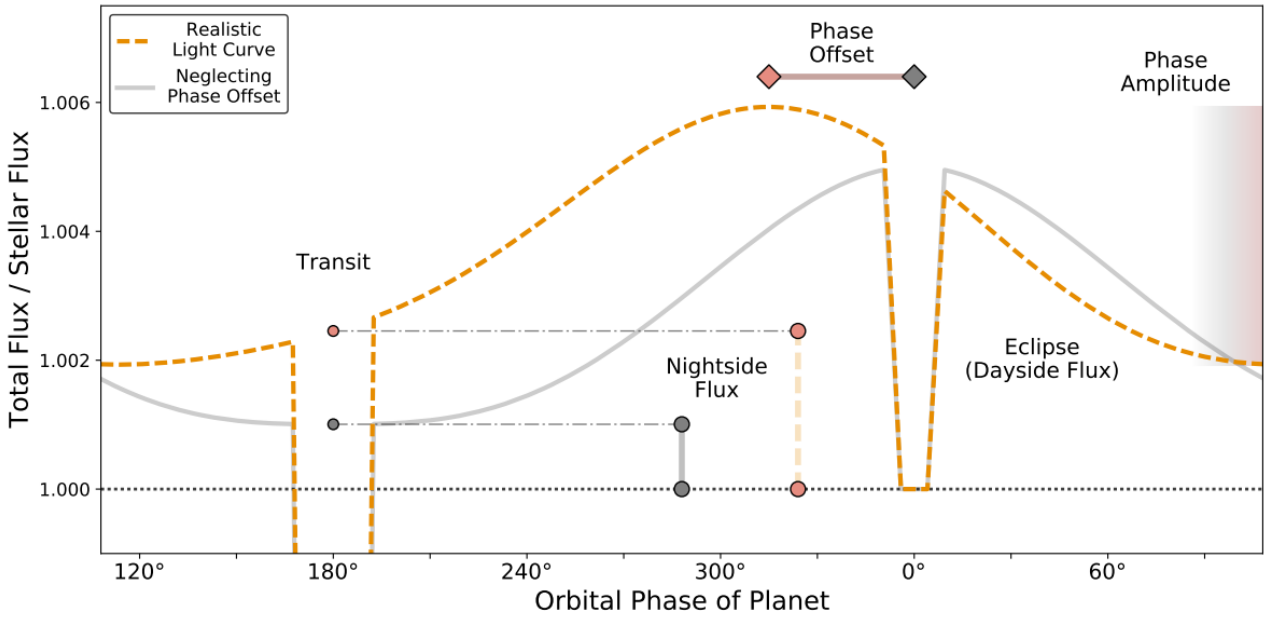

Figure 3: Phase Curve of a transition exoplanet in two cases: One without the phase offset included, and one with. Of note is the decreased total flux over stellar flux for the nightside for the case excluding the phase offset. Image taken from J. C. Schwartz et al. (2017).

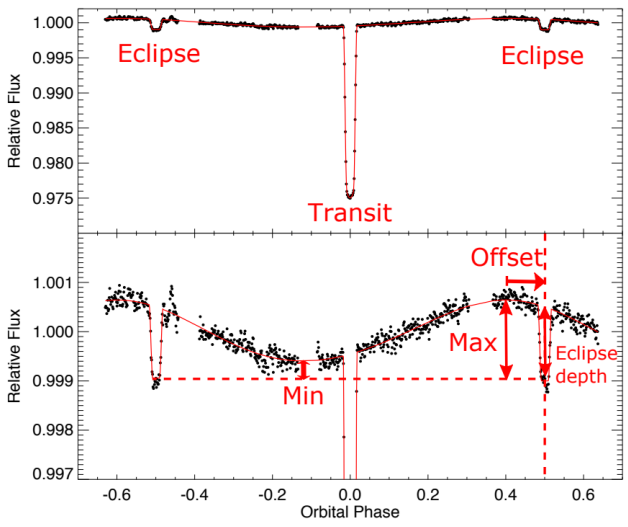

Figure 4: Phase Curve of HD189733b observed with the Spitzer Space Telescope (M. W. Werner et al., 2004), showing both the transit and eclipse of the planet around its parent star. Image taken from H. A. Knutson et al. (2012)

viscosity, $\mathbf{I}$ is the identity matrix, ξ is the bulk viscosity, and $\mathbf{a}$ are the accelerations of the system (G. K. Vallis, 2017). While the methodology will use a simplified model, that being the Shallow Water Model, both equations still require being solved numerically instead of achieving an analytical solution. These wave equations additionally are highly sensitive to their initial conditions, and thus will behave chaotically when evolved over time (M. H. Rafiq et al., 2023). As such, the formulation of the IVP is incredibly important for the behaviour of the model. For a system that is sensitive to its initial environment, it is important to qualitatively differentiate initial values that heavily affect the system's characteristics from ones that do not (as the Jupiter fluid simulations will show).

1.5 Jupiter Fluid Simulations

For the past two decades, research has been done on numerical simulations of hot Jupiters to match observed day- and nightside contrasts of these exoplanets found in transits (C. S. Cooper & A. P. Showman, 2005; J. Langton & G. Laughlin, 2007; J. Langton & G. Laughlin, 2008; A. P. Showman et al., 2008; I. Dobbs-Dixon & D. N. C. Lin, 2008;

K. Menou & E. Rauscher, 2009). Following early simulations done in 2002 by A. P. Showman & T. Guillot (2002), hydrodynamical simulations using equations such as the primitive equations or shallow water equations have allowed high-resolution models to capture precise dynamics such as atmospheric variability on hot Jupiters (D. J. Armstrong et al., 2016; B. Jackson et al., 2019; J. Wilson et al., 2021; Q. Changeat et al., 2024). While primitive equations can provide three-dimensional simulations of the exoplanetary atmosphere and thus allow modelling of dynamics between different layers of the atmosphere, the shallow water equations do provide a sufficient approximation for simulations of exoplanets and are more easily computed (J. Y. K. Cho & L. M. Polvani, 1996; J. Y.-K. Cho et al., 2003; J. Langton & G. Laughlin, 2007). This section will cover several large-scale properties found on hot Jupiter atmospheric simulations.

As shown earlier in figure 1, one large characteristic found within hot Jupiter simulations are the thermal dipoles found close to (but not exactly on) the poles of the planet when experiencing thermal forcing from their parent star. Matching these spots are polar vortices that move over long timescales around the poles of the planet, which for observed transits would be varying offsets of the phase curves (J. Y.-K. Cho et al., 2003; J. C. Schwartz et al., 2017). Circumpolar vortices of the same simulation can be seen in figure 5, providing a detailed global map of the atmosphere and some visual characteristics of note. While this figure uses potential vorticity, roughly proportional to the dot product of the vorticity with the stratification of the atmosphere, we will forgo plots of potential vorticity, as the shallow water model we will use assumes a fully stratified atmosphere. Nevertheless, the polar vortices visible in figure 5 are expected for hot Jupiters, with Rossby waves additionally being visible in the figure through the large-scale wave near the equator.

While Jupiter as in our solar system visibly has multiple tight bands around the entirety of the planet, hot Jupiter simulations lack this characteristic and instead have fewer and larger bands across their latitude (J. Y.-K. Cho et al., 2003; J. Y.-K. Cho et al., 2008; H. A. Knutson et al., 2012; J. W. Skinner & S. Wei, 2025). Exoplanetary atmospheres depend on the Rossby deformation scale L_D to be sufficiently small, such that the bands form similarly to those found on Jupiter. L_D is inversely proportional to the Coriolis parameter f, which itself scales to the planet's rotation rate Ω. As such, hot Jupiter which often exhibit synchronous rotation will have a large L_D due to their far smaller Coriolis parameter compared to what Jupiter exhibits, and qualitatively lack smaller bands across the atmosphere (J. Y.-K. Cho et al., 2008). This characteristic difference between hot Jupiters and Jupiter as we find it in our solar system does raise a matter: Does a planet that begins similar in atmospheric structure to Jupiter, and migrates closer to the star, start to exhibit global characteristics the same as hot Jupiters and lose its original structure?

1.6 Relevance

In this thesis, we will determine whether the initial zonal wind profile is a qualitatively significant property for exoplanetary atmospheres for the characteristic structure of the atmosphere when evolved numerically over time. To this end, the question of this research is:

Does the initial wind profile of a hot Jupiter simulation have a significant impact on the atmosphere's structure when evolved over time?

To specify, the method that will be used to answer this question is to simulate two versions of a hot Jupiter: One using a constructed initial wind profile based on Jupiter's zonal wind profile, and one with the initial wind profile set to be zero. If there is a large-scale qualitative difference between the two cases, the wind profile is an important and sensitive initial condition to solve the IVP and should be further researched for future studies on exoplanetary atmospheres. If the two cases instead were not to differ largely, this initial condition is not a sensitive parameter to an evolved planet's characteristics. However, exoplanetary atmosphere studies should take care in making conclusions based on atmospheric simulations early in their evolution, as patterns found in the atmosphere at such point do not yet reflect observations found in phase curves of transit planets. This should not be taken as if

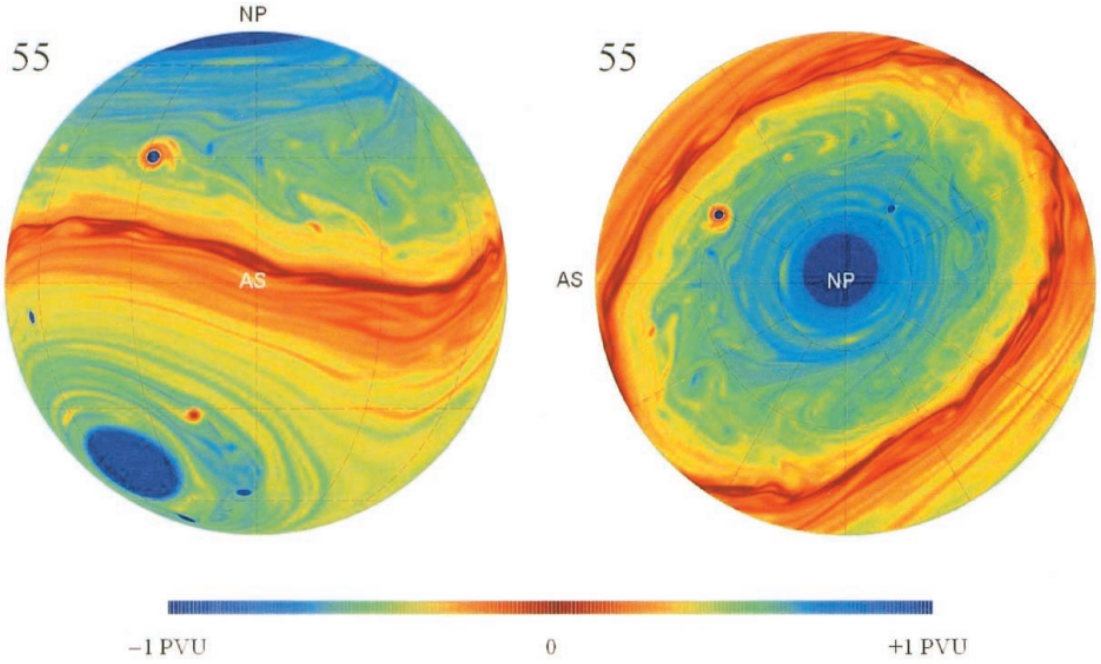

Figure 5: Potential vorticity fields of HD 209458b at t = 55 days and spectral resolution T341 with 1 PVU $= 4 \times 10^{-27}\,\mathrm{sm}^{1/K}$ taken from J. Y.-K. Cho et al. (2003).

this intermediate state is not important for the final shape of the planet, and care should be taken in atmospheric evolutions on short timescales for the longer-scale structure of the planet.

Furthermore, this research also serves to test the numerical framework Dedalus 3 for numerical stability of atmospheric simulations, offering insight into using Dedalus 3 for IVP simulations within this field, and if it can be reliably used alongside other verified solvers. The upcoming methodology section will detail further into the workings of Dedalus 3 and its use for solving atmospheric models.

As such, this thesis will both serve as a qualitative look into atmospheric solutions and their dependence on initial wind profiles, and as a verification of Dedalus 3 for high-resolution numerically convergent atmospheric models.

2 Methodology

2.1 Dedalus 3

This section will explain the spectral method solver Dedalus 3 as a method to solve initial value problems, along with the relevant parts of creating a working use case for solving PDE IVPs. Spectral methods will be explained later on in section 2.3, and the tau method that Dedalus 3 employs in section 2.4.

Dedalus 3 is a numerical solver, focused on solving PDEs to high precision levels (K. J. Burns et al., 2020). As PDEs in cases with multiple variables that evolve continuously can be intensive to solve, a computational method is often applied to numerically solve the problem and test the accuracy of various models. To achieve this, Dedalus 3 uses sparse spectral methods to evolve these models. This application is especially significant for initial value problems and their sensitivity to the initial conditions, where a numerical framework can then be used to qualitatively compare one model to benchmarks and known observations and possibly tuned or adapted for future research.

In order to initialise an initial value problem, the coordinate system of the simulation has to be constructed. Dedalus 3 offers various coordinate systems such as a cartesian grid, spherical coordinates, or S2 sphere coordinates. In case of using paralellisation the distributor method should be applied before constructing the different fields of the variables. To make a numerical solution of the PDEs possible, discrete bases for the fields should additionally be constructed, the specific application of this to pseudo-spectral methods is covered in section 2.3.1. For using the S2 sphere coordinate system, the SphereBasis method of Dedalus 3 is used, taking in a quadrature grid, of which its resolution is proportional to the spectral resolution. For example, a spectral resolution of T341 is approximately equal to a spherical quadrature grid of size 512 by 1024. Given a set of continuous variables, and optionally constants defined within the script (such as the timestep and end time for the simulation), a set of fields can be constructed using the distribution and bases defined. Global fields not explicitly defined across the coordinate system are constructed without a use of the basis and will evolve at a constant rate (e.g. for using time as a variable the d3.Field() method is used).

Given these fields, the equations of the problem can be defined with strings. As more complex operations on the fields such as cross products and derivatives need to be defined, Dedalus 3 offers substitutions to be put into the string using operator methods defined by the module. The fields themselves can furthermore have initial conditions defined by modifying either the grid space using the 'g' layout, or the 'c' layout for coefficient space. For example, a velocity field **u** can be defined in grid space with **u**['g'].

Lastly, the iterative method is defined with an end time for the simulation time set. Analysis tasks can be constructed if so desired to store calculated variables from the simulation in HDF5 files within the specified directory.

The simulation is stopped when either the simulation time has been exceeded, the real-time computation time has been exceeded, or the number of iterations has been exceeded. The computation time is relevant in the case of using High-Performance clusters where maximum allotted times per job are defined. If the computation time within Dedalus 3 is not exceeded, but the wall time of the High-Performance cluster is exceeded, the job is terminated without the possibility of post-processing and analysis of the data within the program. As a result, it is important to set the wall time of the job greater than the simulation time.

The solver stores snapshots at a specified interval of simulated time within a directory such that the data can be used for analysis, with the additional option to store newly defined variables within the snapshot such as vorticity or energy density.

If the computation time is unknown prior to running the solver, there is the option to run the simulation initially until computation time is potentially reached, and then resume the simulation in additional runs using a checkpoint system. This system is set up on a set time interval akin to the snapshots method and is advised to keep at a larger time inter-

val than the snapshots in order to save storage. As an additional note, either new directories should be specified to store snapshots when continuing from a checkpoint, or the old directory should be backed up in a separate location, as the then new simulation will write snapshots again as if initialised from t=0 with the initial conditions set by the checkpoint.

2.1.1 High-Performance Computation

Dedalus 3 uses parallelisation to return precise solutions with short computation time. In order to achieve this, parallelisation is employed through the MPI module. MPI, a Message Passing Interface, allows for computationally intensive processes to be computed on multiple CPU cores at once, reducing overall computation time (M. Rogowski et al., 2023). This is done through partitioning the workload between the different cores, distributing the initial fields across them. There are multiple methods of distribution for the fields, but for Dedalus 3 the distributions follow a relatively intuitive method. In the case of the S2 IVP sphere, MPI divides the latitude field into equally thick horizontal bands equal in amount to the total amount of cores assigned for the simulation. The lowest rank processor (which can also be interpreted as the first core in the series of assigned cores) assigned by MPI occupies the bottommost slice of the latitude field, and each subsequent processor occupies a horizontal slice directly above the previous processor.

In order to properly use parallelisation to decrease computation time, High-Performance Computing (HPC) clusters are used to run the program across multiple cores. For this research the Hábrók HPC cluster is used of the University of Groningen. Hábrók uses the SLURM resource scheduler, where jobs are submitting through shell scripts and queued across one or multiple CPU nodes to run the code or other scripts specified within the file. Using multiple nodes allows for the jobs the become more quickly available to run as SLURM assigns available cores across all nodes up to the amount of nodes specified. Within the shell script the amount of nodes, total amount of CPU cores, wall-time of the script, memory size of each CPU, type of partition, and name for the job are specified. Regular partition type is used for all simulations unless specified, which has a limit of 512 GB total CPU memory size and a maximum of 128 CPU cores. This amount is sufficient for most simulations used here, and it is advised to decrease the memory in cases where it is possible to reduce queue times. A maximum wall-time of 10 days is set for this partition type, however this wall-time is not reached for the simulations used.

For the simulations used in this thesis, wall-time was tested at low levels such as 6 to 12 hours before submitting jobs with a wall-time of 1 to 5 days. As such, scripts that can be ran efficiently or are focused on the initialisation of the model do not face long queue times and are used for the majority when fixing errors or crashes during construction of the simulations. Once a working case on the short term is set, a job then is submitted with the longer scale wall-time, estimated based on a linear extrapolation of the simulated time against real-time (with a margin of 6 hours incorporated to account for potential slower iterations during the process). For cores and nodes a balance is made of 2 nodes with 32 CPU cores, minimising the amount of nodes necessary while still allowing the cores to be distributed across different nodes in case of heavy workload periods on Hábrók. For the majority of simulations in the thesis a resolution of T341 is used, for which 8 GB per CPU core provides a good balance between the necessary amount of memory and minimising the total required memory for the job (once again, done such that queue times remain short). The efficiency and total memory use of the job can be seen in the SLURM log file that is stored in the directory that the job script is in, which in turn allows to fine-tune memory allocation to best suit the simulation if it is ran at higher or lower resolution. Additionally care should be taken in organising both the Dedalus 3 scripts and the relevant job script in the same directory when submitting jobs through the Hábrók interactive shell access, as relative pathing within scripts will be based on the directory the shell access node is in at the time of submitting the script.

2.2 The Shallow Water Model

Multiple models can be chosen for simulating planetary atmospheres, using equations like the primitive equations or the shallow water equations, where we elect to use the shallow water equations here for the

entirety of the thesis. This model is exceedingly simple compared to using the primitive equations, but it does provide a close analogue to strongly stratified atmospheres (M. L. Salby, 1989). Furthermore, its simplified nature in turn allows for far more efficient calculations exactly due to it not needing to account for complex dynamics such as stratification and convection (G. K. Vallis, 2017). For atmospheres with high stratification the Shallow Water Model can be used as an approximation of the dynamics of the planet's atmosphere.

2.2.1 Shallow water equation derivations

In order to derive the shallow water equations, assume a hydrodynamical region of a thin layer with constant density. The Shallow Water Equations (SWE) describe a structure of a rigid bottom surface, a large horizontal layer of fluid, and a free surface on top. SWE can be used in order to describe multiple fluid layers that do not mix, however for the derivations only a single fluid layer is described. The distance h of the rigid and free surface is proportionally small compared to the horizontal width, so we can approximate the vertical hydrostatic equilibrium, but not the horizontal momentum density.

We can describe the vertical hydrostatic equilibrium approximately by:

$$\frac{\delta p}{\delta z} = -\rho_0 g$$

Where g is the effective gravity and ρ_0 is the density at the boundary of the free surface. We now integrate this equation of the vertical direction z:

$$\int \frac{\delta p}{\delta z} dz = \int -\rho_0 g dz$$

Density is taken as constant, so:

$$p(x,y,z,t) - p_0 = -\rho_0 g z$$
$$p(x,y,z,t) = -\rho_0 g z + p_0$$

By setting the height of the rigid surface boundary to be η_b and the free surface boundary to be η, we can set the boundary condition of where $z = \eta$ to have the pressure $p = 0$, leading to p_0 being able to be described as such:

$$p_0 = 0 = \rho_0 g \eta(x,y,t)$$

Substitute this into the equation describing pressure:

$$p(x,y,z,t) = \rho_0 g(\eta(x,y,t) - z)$$

So the pressure gradient with respect to x and y is:

$$\nabla_z p(x,y,z,t) = \rho_0 g \nabla_z \eta(x,y,t)$$

Use the horizontal momentum equation $\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho_0}\nabla_z p(x,y,z,t)$ and substitute in the above equation. Here $\mathbf{u}$ has an x and y component.

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho_0}\rho_0 g \nabla_z \eta(x,y,t) = -g\nabla_z \eta(x,y,t)$$

Add rotation as a final term to the equation as the cross product $\mathbf{f} \times \mathbf{u} = f\mathbf{k} \times \mathbf{u}$. Using this addition we get the first SWE:

$$\frac{D\mathbf{u}}{Dt} + f\mathbf{k} \times \mathbf{u} = -g\nabla_z \eta(x,y,t)$$

For the second SWE we start from the mass continuity equation. Due to the density being constant, the fluid is incompressible and thus:

$$\nabla \cdot \mathbf{u} = \frac{\delta v_x}{\delta x} + \frac{\delta v_y}{\delta y} + \frac{\delta v_z}{\delta z}$$

$$\frac{\delta v_z}{\delta z} = -(\frac{\delta v_x}{\delta x} + \frac{\delta v_y}{\delta y}) = -\nabla_z \cdot \mathbf{u}$$

Next, integrate the equation from η_b to η:

$$\int_{\eta_b}^{\eta} \frac{\delta v_z}{\delta z} dz = \int_{\eta_b}^{\eta} -\nabla_z \cdot \mathbf{u} dz$$

$-\nabla_z \cdot \mathbf{u}$ is independent of z, so:

$$v_z(\eta) - v_z(\eta_b) = (\eta - \eta_b)(-\nabla_z \cdot \mathbf{u})$$

Remember that h is the distance between η and η_b, so:

$$v_z(\eta) - v_z(\eta_b) = -h\nabla_z \cdot \mathbf{u}$$

For a set height of a set fluid element within a fluid, its material derivative $\frac{Dz}{Dt}$ is equal to its velocity v_z at said height. So for height η, its vertical velocity is equal to the time dependence and advection rate in the horizontal direction. By applying the same to η_b we get the following equation:

$$\frac{D\eta}{Dt} - \frac{D\eta_b}{Dt} = -h\nabla_z \cdot \mathbf{u}$$

Combining the material derivatives we end up with the second SWE:

$$\frac{Dh}{Dt} = -h\nabla_z \cdot \mathbf{u}$$

Altogether the two SWE can be written as:

$$\frac{\delta \mathbf{u}}{\delta t} + \mathbf{u} \cdot \nabla \mathbf{u} = -g\nabla h - f\mathbf{k} \times \mathbf{u}$$
$$\frac{\delta h}{\delta t} + \mathbf{u} \cdot \nabla h = -h\nabla_z \cdot \mathbf{u}$$

Thus we have the derivations for single-layered shallow water approximations for the inviscid case. We will use both the case without explicit diffusion, and one with diffusion incorporated, for which we add standard order 2 diffusion of the following form:

$$\nu\nabla^2\mathbf{X}$$

Where ν is the numerical explicit diffusion, which is used to dissipate and as a result control small-scale flows. $\mathbf{X}$ here is what parameter the diffusion acts on, where adding this for both the velocity and height vector fields yields the viscous SWE:

$$\frac{\delta \mathbf{u}}{\delta t} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu\nabla^2\mathbf{u} = -g\nabla h - f\mathbf{k} \times \mathbf{u}$$
$$\frac{\delta h}{\delta t} + \mathbf{u} \cdot \nabla h - \nu\nabla^2 h = -h\nabla_z \cdot \mathbf{u}$$

These equations are then used for further derivations and applications in order to describe the observable section of exoplanetary atmospheres. To numerically solve the SWE however, the methodology of the spectral method needs to be set first. The coming section will go into detail on the specific workings of the spectral method and its variations.

2.3 Spectral Method

For simulation cases using Partial Differential Equations (PDE), computation can prove to be complex due to their multiple variables and continuous derivatives. While many approaches to solving PDEs exist, they often rely on translating continuous variables into discrete ones to achieve an approximate solution to the equations. This does work in simple use cases where few variables are present, but in more complex cases such as hydrodynamical models, higher precision and accuracy become slow to solve.

One approach to efficiently solve PDEs computationally is through spectral methods. In this section the methodology of spectral methods is explained, with a focus on the implementation within Dedalus 3, a spectral solver for PDEs. More detail on Dedalus 3 is focused on in section 2.4.

The general construction of a spectral method is done through representing a specified function through a finite sum of n orthogonal basis functions ϕ_n which are scaled by coefficients $\mathcal{F}_n^\phi$. The basis functions can be of any form so long as they meet the requirements of converging to the specified function, being able to have its derivatives expressed through the same basis functions with a different set of coefficients $\mathcal{F}_n^\phi$, and it should be efficient to convert between the original dataset from the specified function and the set of basis functions. As such, the full representation of the original function is spectrally represented as:

$$\mathcal{F}(x) = \sum_{n=0}^{\infty} \mathcal{F}_n^\phi \phi_n(x)$$

Here $\mathcal{F}_n^\phi$ is furthermore defined as:

$$\mathcal{F}_n^{\phi} = \frac{(\phi_n, \mathcal{F})_{\phi}}{(\phi_n, \phi_n)_{\phi}}$$

Where $(,)_{\phi}$ defines the inner product of the terms specified. This representation of the original function is exact, but in order to have a computationally viable representation an approximate sum is used. This sum is truncated at N modes, and $\mathcal{F}_n^{\phi}$ is instead defined as:

$$\mathcal{F}_n^{\phi} = \sum_{i=0}^{N-1} w_i \mathcal{F}(x_i)$$

Where w_i are the weights of each of the coefficients and x_i are the collocation points used to numerically define the PDE. This approximation constitutes the discrete spectral method, and the error tends to the same order as the last coefficient in the series. As spectral representations of smooth functions have exponentially smaller coefficients for an increase in n, this method gives accurate representations of PDEs through the use of basis functions, which in turn provide an efficient computation of complex functions and problems.

The method of finding the coefficients largely varies between different frameworks and offer differing levels of complexity with regards to creating the models and computing multiple connected equations at once. In the following section the Galerkin method, a widely used technique for pseudo-spectral solvers, is explained, whereas the specific technique Dedalus 3 uses is explained in section 2.4.

2.3.1 Pseudo-Spectral Method

A clean spectral method would ideally be used to fully converge to the IVP PDEs, but non-linear terms drastically lower the efficiency of computation compared to linear ones (L. Greengard, 1991; K. Julien & M. Watson, 2009). Given the non-linear terms within the SWE, pseudo-spectral methods instead are favoured for solving non-linear problems. To illustrate the how pseudo-spectral methods are constructed, the collocation and Galerkin method provide well-posed general methods for pseudo-spectral solvers.

To explain the collocation and subsequently Galerkin method for solving for the coefficients, the trial and test functions are created by first taking the general form of a linear PDE (N. Mendes et al., 2019):

$$Lu(x) = \mathcal{F}(x)$$

Here L is a linear operator acting on function $\mathcal{U}(x)$, resulting in function $\mathcal{F}(x)$. Representing the functions with their finite expansions into basis functions results into an approximation with residuals R as follows:

$$R(x) = L\phi_n \mathcal{U}_n^{\phi} - \phi_n \mathcal{F}_n^{\phi}$$

As minimising the residuals will result in a convergent solution to the PDE, the PDE can be enforced across the domain by using the aforementioned weight functions w_i, also named test functions:

$$\int_X R(x) w_i dx = 0$$

For the classical pseudo-spectral method, also named the collocation method, these weight functions are defined as a set of Dirac delta functions across an N-sized quadrature grid of collocation points x_i. As such, the integral can be rewritten as:

$$\int_X R(x)\delta(x - x_i)dx = 0$$

$$R(x_i) = 0$$

Using the definition of the residuals, the solution to the PDE can be defined:

$$R(x_i) = L\mathcal{U}(x_i) - \mathcal{F}(x_i)$$

$$L\mathcal{U}(x_i) = \mathcal{F}(x_i), \quad with \quad i = 0, ..., N-1$$

The boundary conditions of the model replace the endpoints of the quadrature grid for the solutions to avoid singularities within computation.

The Galerkin method incorporates the boundary conditions into the trail and test functions themselves instead of using a quadrature grid of collocation points, and uses basis functions for both the trial (ϕ_n) and test functions (ψ_i) (N. Mendes et al., 2019). As such, the residual integral and subsequent solution can be rewritten as such:

$$\int_X R(x)\psi_i dx$$

$$\sum_n \langle \psi_i | L\phi_n | \psi_i | L\phi_n \rangle \, u_n^\phi = \sum_n \langle \psi_i | \phi_n | \psi_i | \phi_n \rangle \, \mathcal{F}_n^\phi$$

This allows for diagonalised solutions for models with periodic boundary conditions and Fourier basis functions, which in turn allow for efficient computation of these models. However, this method does require basis functions which themselves satisfy the boundary condition and as such limit the scope to which this can be applied without additional changes.

2.4 Dedalus 3 Spectral Method

Dedalus 3 uses a sparse spectral method based on the tau method to solve general equations. The aim for a sparse matrix of basis equations for a specified problem follows from sparse methods providing quicker and more stable solutions compared to densely defined matrices (L. Greengard, 1991; K. Julien & M. Watson, 2009; S. Olver & A. Townsend, 2013). These sparse matrices are achieved through changing the tau method with regards to its trial and test bases. These trial and test bases operate similarly to the Galerkin method, with the change of perturbing the PDE in the following form:

$$L\mathcal{U}(x) + \tau P(x) = \mathcal{F}(x)$$

Where τ is a constant determined by the boundary conditions of the problem, and $P(x)$ is classically defined as $P(x) = \phi_{n-1}(x)$. This formulation however is sensitive to the choice of boundary conditions and can return dense matrices as a result. The sparse spectral method avoids this through using basis recombination, which sets Dirichlet boundary conditions as the ordinary boundary conditions to create sparse matrices even with basis functions that would ordinarily create dense matrices. For further documentation on the basis recombinations that Dedalus 3 uses, see K. J. Burns et al. (2020).

An additional challenge to have a tau spectral method that uses sparse matrices, is that spatially Non-Constant Coefficients (NCC) can result in non-diagonalised or dense matrices (S. Olver & A. Townsend, 2013). This can be solved through expanding the NCC through basis functions and then truncating all coefficients within said expansion through a threshold amplitude, after which the coefficients are interleaved with the sparse matrices. This does in turn require the NCC to be smooth to preserve the sparseness of the matrices. The resulting banded matrix will have a bandwidth of an order of the amount of variables present within the PDEs, allowing efficient computation with a large range of basis functions and coupled PDEs.

2.4.1 Dedalus 3 Methodology

Dedalus 3 expands the aforementioned functions into the spectral functions, using coefficients $\mathcal{F}_n^\phi$ for example of a basis of ϕ_n, truncated after N modes. Due to the exponential decay in coefficients for smooth functions, this method is used to solve for SWE numerically. Given that the SWE is highly differentiable, the problem is more favourable to compute through Dedalus 3 opposed to other methods, providing good reason to use this method and module to calculate SWE cases for an S2 sphere.

2.5 Case 1: Numerical Benchmark using Viscous Shallow Water Models

Before the Dedalus 3 spectral solver can be used to create a model of Jupiter, it is first necessary to test the accuracy and precision of the model compared to other spectral solvers. One benchmark test for numerically solved shallow water initial value problems is the benchmark of J. Galewsky et al. (2004). This benchmark uses the viscous SWE to test for convergence on the height and vorticity field across different spectral resolutions. To restate, the equa-

tions of this viscous shallow water model are as follows:

$$\frac{D\mathbf{u}}{Dt} = -f\mathbf{k} \times \mathbf{u} - g\nabla h + v\nabla^2\mathbf{u}$$
$$\frac{Dh}{Dt} = -h\nabla \cdot \mathbf{u} + \nu\nabla^2 h$$

Where $\mathbf{u} = \mathbf{i}u + \mathbf{j}v$ is the velocity vector of the field across the surface of the sphere (specified as this adopts the S2 sphere geometry). All other variables are identical to the equations listed in section 2.2.1. The following section details the setup of the benchmark test, which has its values of the vorticity field and height field compared to two other numerical models for solving the problem spectrally.

2.5.1 The Zonal Jet Benchmark

For initial conditions, a zonal jet is created at latitude $\phi = \pi/4$ radians. The zonal jet is bound by ϕ_0 as the southern jet boundary in radians, and ϕ_1 as the northern jet boundary. Outside of these boundaries the jet is set to be 0. Within these boundaries the full expression of this jet is as follows:

$$u(\phi) = \frac{u_{max}}{e_n} e^{\frac{1}{(\phi-\phi_0)(\phi-\phi_1)}}$$

Here u_{max} is the maximum zonal velocity and e_n is the normalisation factor of the jet. Alongside this, the expression for the height field is given as follows:

$$gh(\phi) = gh_0 - \int_{\phi} ru(\phi')(f + \frac{tan(\phi')}{r}u(\phi'))d\phi'$$

Where h_0 is the mean height of the shallow water layer and r is the radius of the S2 sphere. Given these expressions and the equations for the viscous shallow water model, the initial values are set for the simulation to run. These values are: g = $9.806\,16\,\mathrm{m/s^2}$, $u_{max} = 80\,\mathrm{m\,s^{-1}}$, r = $6.371\,22\times10^6$ m, $h_0 = 1 \times 10^4$ m, and $\nu = 1 \times 10^5\,\mathrm{m^2/s}$ (analogous to Earth's parameters). Using the expression $f = 2\Omega sin\phi$, Ω additionally is set to $7.292 \times 10^{-5}\,\mathrm{s^{-1}}$.

This initial condition is then perturbed by introducing a bump to the height field defined between longitudes of $-\pi < \theta < \pi$ radians. This bump is defined as:

$$h'(\theta, \phi) = \hat{h}cos(\phi)e^{-(\theta/\alpha)^2}e^{-((\phi_2-phi)/\beta)^2}$$

With $\phi_2 = \pi/4$, $\alpha = 1/3$, $\beta = 1/15$, and the perturbed height $\hat{h} = 120$ m. For the benchmark order 2 diffusion is used, and for consistency order 2 diffusion is used as well for the other cases.

Using a timestep of 30 seconds and spectral resolution of T341 (roughly equal to a quadrature grid size of 512 x 1024), the model is run for 150 hours with data snapshots stored every 0.5 hours. The data snapshots contain the values of the height field, along with the vorticity field calculated as $\zeta = \nabla \times \mathbf{u}$.

For the first simulation case, the model is ran without viscosity, which can be done through simply setting $\nu = 0$. The results acquired by Galewsky using the Geophysical Fluid Dynamics Laboratory FMS-SWM model can be seen in figure 11 (J. Galewsky et al., 2004).

The second simulation case is ran with viscosity at the aforementioned value. The results from Galewsky for this case can be seen in figure 12.

For the script used to test Dedalus 3 for the Galewsky benchmark, initial values are set equivalent to those used in J. Galewsky et al. (2004). Dedalus provides a general framework for constructing IVP PDE models with an S2 sphere, with which we build on to rigorously design test cases for both the viscous and inviscid SWE, matching in formulation to the benchmark.

2.5.2 Model Comparison

To verify whether or not a given framework is accurate as a solver for the shallow water IVP, J. Galewsky et al. (2004) used two independent models to solve the IVP for the height and vorticity. Using the FMS-SWM, along with the NCAR BOB Shallow-Water Model (L. Rivier et al., 2002), matching values between the different numerical models are visible in the table of figure 6. Important note is that the Galewsky benchmark instructs to compare to values in the table for convergent values before adding explicit diffusion, while NCAR

BOB is only used in the explicit case in the paper. As a result, values for the minimum and maximum height and vorticity field values are to be taken for the case with explicit diffusion of $\nu = 1 \times 10^5\,\mathrm{m}^2/\mathrm{s}$ and compared to values from the table. Numerical frameworks that closely match values (ideally to machine precision) in the tables are numerically convergent, but further tests for convergence are taken later on in section 2.6.6 to verify models with more complex initial states. The results of the benchmark for Dedalus 3 can be found in section 3.1, but do not feature values of the divergence field.

	Value
At $t = 4$ h	
$l_2(\delta)$	$4.0 \times 10^{-7}\ \mathrm{s}^{-1}$
$\max(\delta)$	$3.7 \times 10^{-6}\ \mathrm{s}^{-1}$
$\min(\delta)$	$-2.0 \times 10^{-6}\ \mathrm{s}^{-1}$
$l2(h)$	9778 m
$\max(h)$	10 182 m
$\min(h)$	9052 m
At $t = 144$ h	
$l_2(\zeta)$	$2.1 \times 10^{-5}\ \mathrm{s}^{-1}$
$\max(\zeta)$	$9.3 \times 10^{-5}\ \mathrm{s}^{-1}$
$\min(\zeta)$	$-7.3 \times 10^{-5}\ \mathrm{s}^{-1}$

Figure 6: Table taken from J. Galewsky et al. (2004), listing vorticity, divergence and height field values for set moments in time.

2.6 Case 2: Simulating Zonal Jet Profile of Jupiter

2.6.1 Transition to Jupiter-based Initial Values

As the full zonal jet profile for this section as a whole has complex initial conditions, intermediary steps are made to introduce characteristics of Jupiter gradually. This way, any issue with the implementation of Jupiter can be isolated and resolved (e.g. opposing zonal jets, tightly packed jets over a small interval of latitudes).

Firstly, the script used in the Benchmark without perturbations is ran with identical jet properties and initial parameters of Jupiter. The values of these parameters can be seen in table 1.

Radius	r	$71.492 \times 10^6\,\mathrm{m}$
Rotation Rate	Ω	$1.758\,518\,1 \times 10^{-4}\,\mathrm{s}^{-1}$
Surface Gravity	g	$25.92\,\mathrm{m/s}^2$
Scale Height	h	$1 \times 10^4\,\mathrm{m}$
Maximum Velocity	u_{max}	$150\,\mathrm{m\,s}^{-1}$
Explicit Diffusion	ν	$1 \times 10^5\,\mathrm{m}^2/\mathrm{s}$

Table 1: Initial values for Jupiter for Case 2 (P. K. Seidelmann et al., 2007; R. Hueso et al., 2023).

2.6.2 JWST Zonal Data

Following this, the zonal jet is moved to the equator of the planet with u_{max} increased to the maximum observed zonal velocity from R. Hueso et al. (2023), of which the total zonal profile is publicly available and displayed in figure 7. For the zonal profile, the F164N filter is used, which captures methane absorption bands in the upper atmospheric regime of the planet. This zonal profile additionally is the intermediary observed profile of the planet compared to the two other observed profiles of F212N and F335M with regards to the maxima of zonal velocity. The F164N wind profile has an error of $2\sigma = 15\,\mathrm{m\,s}^{-1}$ and maximum zonal velocity of approximately $u_{max} = 150\,\mathrm{m\,s}^{-1}$.

On top of the simulation using the equatorial band with the maximum velocity as given by JWST, another simulation should be constructed to test for opposing bands. To do this, the Galewsky band from previous tests is used at Jupiter's scale on the same colatitude as the benchmark, and mirrored below the equator. With this, two strong bands in opposing directions can be tested for instabilities. If there is no perturbations and the bands are smoothly defined, precision should be maintained over the simulation, and the bands should run identical over long periods to machine precision.

If both the opposing bands and the maximum zonal velocity equatorial band work smoothly and stay identical without perturbation, the next step will involve implementing the full JWST profile. The coming section will detail issues with automated implementation.

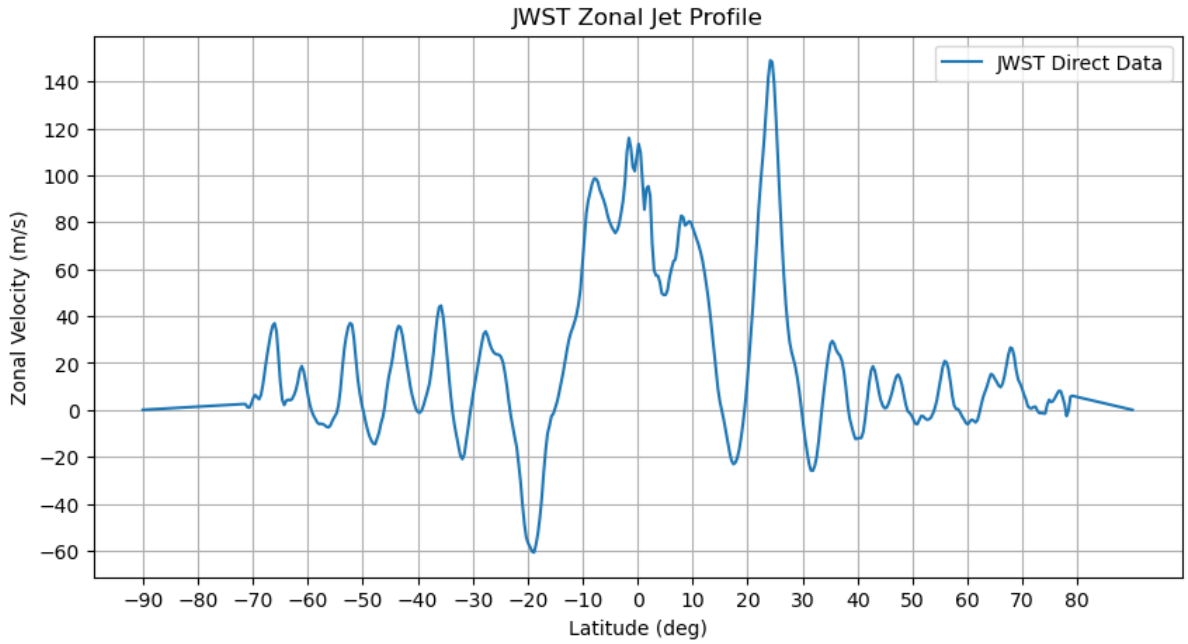

Figure 7: Zonal Velocity against latitude of Jupiter according to JWST Observations with F164N, linearly interpolated to 512 data points.

2.6.3 Issues with Fourier Construction

This section discusses the setup of the Fourier approximation function for the JWST zonal wind profile, but this method will not be used to get results, nor will it be used for further use cases. However, this section is still included due to it functioning as proof of concept of automated implementation of data. For the actual approximation used for this research, see section 2.6.4.

Ideally it would be possible to use the fast Fourier transformation to get the coefficients for the JWST full profile data. This way a smooth global function can be defined through said Fourier expansion. If alternatively implementing the data directly is desired, there are some factors that cause immediate errors to appear. As the sparse tau method requires a smooth expansion of basis functions from the original function, raw data as initial velocity field conditions do not result in a stable simulation. This happens due to the data being a discrete array, where even using linear interpolation to the grid size used (T341), will immediately return math overflow errors.

Fourier approximations be a favourable approach, as using globally smooth and analytical functions fit well for finding the spectral expansion. An alternative way to interpret this would be that the coefficient space is found for the data, turned into a grid-based function, and then Dedalus 3 automatically finds the coefficient space depending on the basis function chosen. Direct input on the coefficient space would limit the choice of basis functions, so this approach would be preferred. The coefficients of the wind profile are achieved through using the Numpy fast Fourier transform (i.e. numpy.fft.fft) method and stored in a local array, then loaded in the following function:

$$S(\theta) = \sum_{j=0}^{N} \frac{1}{\pi} \mathcal{F}_j e^{2i\pi(j/\pi)\theta}$$

Where $\mathcal{F}_j$ is the j-th coefficient of the fft array, and N is the total amount of expansion terms (possible range of up to 512 total coefficients).

With the Fourier approximation function defined, there is the additional issue of the poles of the S2 sphere. If the simulation were to be a Cartesian grid, the boundaries on the latitude would not create issues, but as the simulation is on the

shell of a spherical coordinate system, the poles in Fourier space are singularities. This would logically cause errors within the computation of the model, and the documentation of Dedalus 3 as well recommends creating no-slip boundaries to the poles of the sphere. To create this initial region that is explicitly defined to have no non-zero values, the error function can be used in the following form:

$$erf(z) = \frac{2}{\pi}\int^{z} e^{-t^2} dt$$

This function is then used to create an error function window that over a smooth interval sets the Fourier approximation function to 0 outside certain colatitudes. The full window is then defined as:

$$W_{erf}(\theta) = \frac{1}{2}(erf((\theta - \theta_W)/a)) - \frac{1}{2}(erf((\theta - \pi - \theta_W)/a))$$

Where θ_W is the truncation latitude chosen as $\theta_W = \pi/7$, and a is the width parameter of the error function. The width parameter in the test is chosen as 0.05 to minimise the loss of data for the wind profile while still preserving smoothness of the curve. This window is then multiplied with the Fourier approximation function, where the product of this is used to define the initial velocity field.

The issues with this model can be seen in figure 8, which occurs a few iterations after being initialised. For time scaled on the order of magnitude larger than 5 second, a math overflow error occurs after 4 timesteps. When taking smaller scales in mind, the simulation produces errors on timescales ranging from 20 milliseconds to 9 seconds, depending on how complex the Fourier approximation function is and how truncated the window is. The direct cause to this is not certain, but based on similar issues found by other users of Dedalus 3 it is possible that the approximation function features too steep features, causing Gibbs-like ringing in spectral space. This ringing is then worsened by the present steep slopes in the fields and causes math overflow errors to occur. Creating an approximation with smoother slopes would likely result in simulations that do not experience these errors, and additionally allow for any arbitrary data to be used in shallow water models with Dedalus. One possible approach for future cases could be convolution of the wind profile with a wide normal distribution, as if to blur the approximated function.

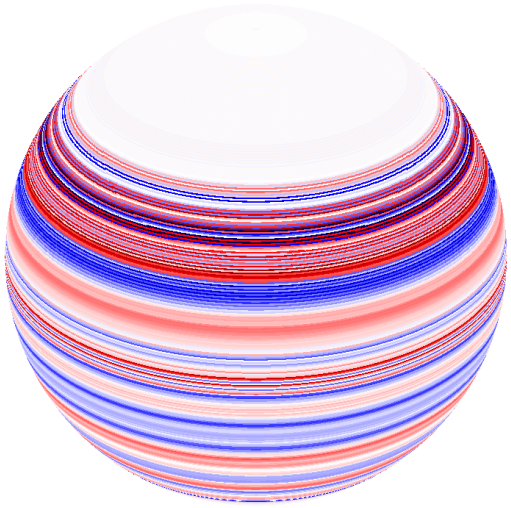

Figure 8: Vorticity plot of the Fourier construction using JWST zonal jet profile data. Gibbs-like ringing can be seen at mid- and upper latitudes with strong vorticity at the banded points.

2.6.4 Gaussian Band Approximation

In order to approximate the JWST zonal wind profile through a global function, but not result in math overflow errors as with the Fourier approximation function, the jet formulation from the Galewsky benchmark is used to construct a global profile similar to the input data. This is achieved simply through using a sum over all jets, with the following form:

$$u(\lambda) = \sum_i \frac{u_{max,i}}{e_{n,i}} e^{\frac{1}{(\lambda-\lambda_{0,i})(\lambda-\lambda_{1,i})}}$$

Where λ is the colatitude defined as $\lambda = \frac{\pi}{2} - \theta$, λ_0 and λ_1 are the upper and lower colatitudes of the band respectively. For constructing these bands using the centre of the Gaussian bands and the width

of said bands, the upper and lower bounds can be defined as:

$$\lambda_0 = (\lambda_m + \lambda_w)\frac{\pi}{180}$$
$$\lambda_1 = (\lambda_m - \lambda_w)\frac{\pi}{180}$$

Where λ_m is the mean of the jet band in degrees, and λ_w is half the width of the band in degrees. This half-width of these bands cannot be smaller than 10 degrees, as smaller widths cause overflow errors to occur. This does again support the idea that the Fourier approximation is not suitable due to strong slopes occurring within the function. The full assembled profile of the wind profile can be seen in figure 9, with the Gaussian approximation function plotted over the JWST zonal jet profile data. Small features, especially near the outer regions of the field, are not reproduced due to instabilities from too strongly sloped features and avoiding non-zero values near the poles of the shell. This provides a near complete initial condition for the velocity field, and the following section will provide a method to add stirring mechanisms to the simulation in order to provide a model sufficient to use in the subsequent case. To indicate the difference between velocity profile, vorticity field, and the visual bands of Jupiter, figure 9 shows the dissimilarities between them. However, the vorticity field and velocity field can be visually compared through strong slopes in the velocity profile being matched with strong values in vorticity, where negative slopes respectively match to negative vorticity values. For testing stability of the unperturbed simulation, a timestep of 100 seconds and total simulation time of 240 hours is used.

2.6.5 Stirring

A pure smooth and solid bottom layer below the shallow water layer would not suffice as an analogue for the lower regions of Jupiters atmosphere, as it would imply a fully stratified and stable interface between the two different sections of the atmosphere. To create a closer analogue to a gas giant's atmosphere, initial stirring is applied to the shallow water layer on the height and velocity field. For the height field, a random noise perturbation is applied with an amplitude of 1×10^3 m, whereas for the velocity field the amplitude of the noise is $10\,\mathrm{m\,s^{-1}}$. Through this, the large-scale features of the simulation are not changed, while interactions from lower layers in the atmosphere are mimicked through random motion. For reproducibility *ideally* a seed for the random noise would be used, but due to the process being parallelised and the structure of Dedalus 3 distributing the fields across cores, using a seed would result in multiple horizontal slices across the surface having identical noise. Due to this, no seed is used, and stirring cannot be mimicked identically. However, for the large-scale features of the atmosphere, which are a focus for this research, the lack of reproducible pattern does not detract from the results.

2.6.6 Convergent Perturbed Solutions

Given the model with Gaussian approximation function and stirring present, testing numerical convergence for this new, more complex structure, is necessary. This is necessary as the final case involves direct height forcing (see case 3 for this). For proven convergence, this then allows for definitive statements on tests with the Gaussian approximation function. In the case of numerical convergence with the Galewsky benchmark, this test should additionally have numerical convergence.

The resolutions used to test for convergence are T170, T341, and T683, where for T341 convergence is additionally tested for a timestep of 30 seconds, 50 seconds, 150 seconds, and 300 seconds. This can be done by simply changing the initial parameter of the timestep, and for spectral resolution adopting quadrature grid sizes of 256x512, 512x1024, and 1024x2048 respectively. Due to the scale of Jupiter, using a maximum time of 144 is not sufficient to see significant motion. For the convergence tests we use as a result a simulation end time of 2400 hours and $\nu = 1 \times 10^{-10}\,\mathrm{m^2/s}$ for testing a minimally viscous model.

The perturbation used is an equatorial height bump using the perturbation from the Galewsky benchmark at 0 degrees latitude and longitude, with a total height of 12000 meters and with parameters set of $\alpha = \frac{1}{3}$ and $\beta = \frac{1}{15}$.

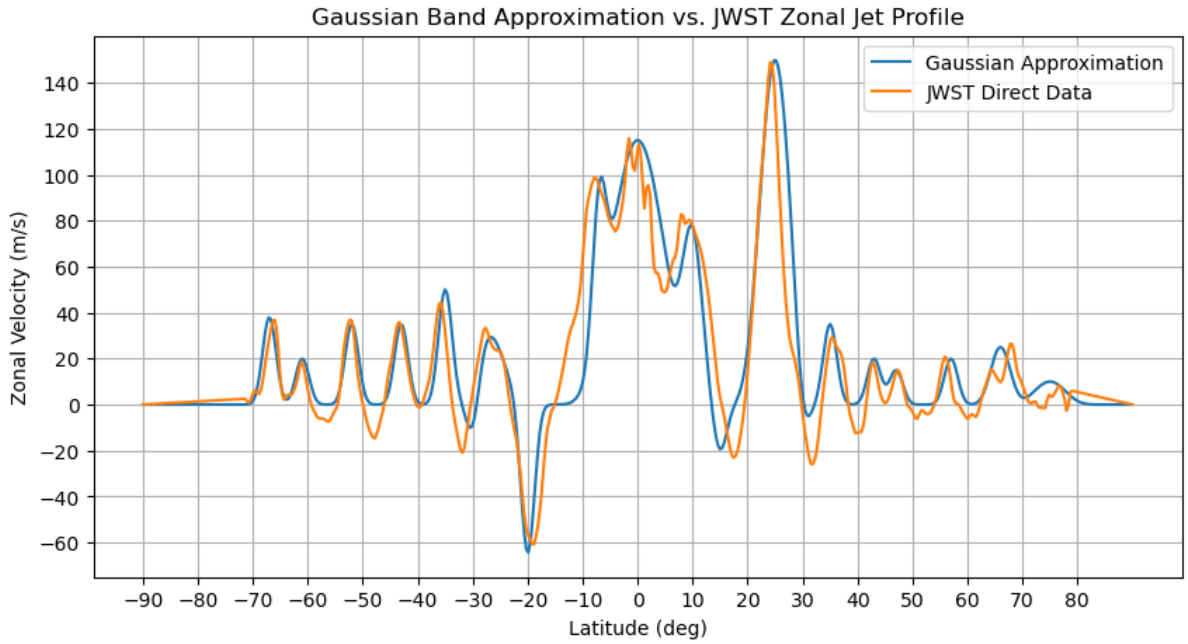

Figure 9: JWST zonal jet profile data plotted over the zonal profile constructed through the superimposed Gaussian bands. Note the further bands close to the boundaries of the latitude being discarded as to avoid non-zero values near singularities.

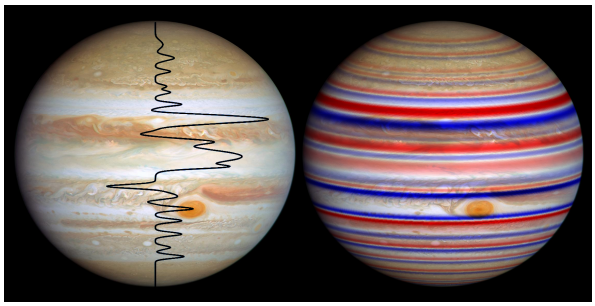

Figure 10: Visual plots of the data from figure 9. Left: The Gaussian Approximation zonal velocity profile plotted over Jupiter. Right: The constructed vorticity field based on the Gaussian Approximation, plotted over Jupiter. Image of Jupiter is taken from NASA, ESA, Amy Simon (NASA-GSFC); Image Processing: Joseph DePasquale (STScI)

Additionally, long term stability is tested by running a T341 100 second timestep simulation with identical perturbation and simulation stop time of 24000 hours. As the third case of height forcing requires a large simulation time size, this allows for tuning the system in case longer term simulations require additional programming. In case the simulation length exceeds the wall-time, a checkpoint system is implemented that stores additional snapshots every 480 hours, and can load the system state of the snapshot to initialise subsequent runs. The full code is available within the Appendix.

2.7 Case 3: Hot Jupiter Simulation using Heigh Forcing

The final case uses the Jupiter simulation acquired in case 2 with all zonal bands present, simulated with one side of the planet radiated by its parent star. As the goal is to compare between wind profiles of a hot gas giant with no initial conditions imposed on the wind profile, and one where a wind profile is initialised, this case will allow a qualitative analysis on whether or not the initial conditions of the wind profile are relevant.

Using the case 2 script for T341 and timestep of 20 seconds without perturbation, some additional initial values are required before the simulation can be ran. In the coming sections the details of these values and additional changes to the equations for height field forcing are described.

2.7.1 Tidal Locking

While tidal locking or synchronous rotation would be a property that is present on hot Jupiters, the strength of the height forcing outweighs the effects from planetary rotation. As a result, planetary rotation is still incorporated.

2.7.2 Height Forcing

In order to create a case that simulates thermal heating one side of the planet, we apply height forcing to the planet. This uses the following function to deviate the height:

$$h_\Delta(\lambda, \phi) = h_{dev} cos(\lambda) cos(\phi)$$

Where h_Δ is the deviating height field across the surface and h_{dev} is the maximum deviation of the height field. The maximum deviation $H_\Delta$is set to $H_\Delta = 0.1 H_{ave}$, where the averaged height is calculated as follows:

$$H_{ave} = 5 \cdot H_{scale}$$

Where H_{scale} is the scale height calculated through the following equation:

$$H_{scale} = \frac{k_b T_{eq}}{\mu_\sigma g}$$

Where k_b is the Boltzmann constant set to $1.380\,649 \times 10^{-23}\,\mathrm{J\,K^{-1}}$, $T_{eq} = 1.5 \times 10^3\,\mathrm{K}$ is the average temperature across the model, μ_σ $=3.321\,078\,1 \times 10^{-24}\,\mathrm{g}$ is the mean molecular mass of H_2, and g $=10\,\mathrm{m/s^2}$. The height field as well is adjusted to a larger radius of the hot Jupiter, where the increase in radius is justified from the gas heating up through being in close proximity to its parent star. As such, the total radius of the planet is set to $R = 1.2r$ $=8.579\,04 \times 10^7\,\mathrm{m}$.

The height forcing term using this is as follow:

$$\frac{h_\Delta - h}{\tau_{rad}}$$

Where τ_{rad} is the radiative timescale set to 12.0 hours.

While this is not used in the simulations for the third case here, additional smoothing is possible for those concerned with sudden height forcing on the planet. As an additional factor, a saturation curve can be added to the right hand side by multiplying it with the height forcing term:

$$S(t) = \frac{1}{1 + e^{-0.05(t-60)}}$$

Using this, the height forcing is applied slowly over time instead of instantly at the start of the simulation. This will in turn avoid shockwaves forming from the immediate strong forcing, which would cause instabilities in the model that should not exist naturally. If desired, the scale height of the simulation can be evolved from an initial low temperature similar to Jupiter to the full average temperature for the height forcing case through the same means. However, for the case used in this paper, the saturation curve is left out due to computational issues with the existing script (specifically, math overflow errors akin to the Fourier case). For stability of the model, diffusion is increased as well to $\nu = 1 \times 10^5\,\mathrm{m^2/s}$. This model is then evolved over 2000 hours with a timestep of 20 seconds, saving data snapshots every 50 hours.

2.7.3 Rest Case Model

To compare qualitatively if there is a significant factor in added the zonal wind profile for creating accurate atmospheric simulations, a control comparison is added to compare with the zonal jet profile case. This does require minimal setup, as the velocity field can be simply set to 0. This simulation is then ran for the same time parameters as the jet profile case.

3 Results

3.1 Results Case 1: Numerical Benchmark and Deviation

Runs by Dedalus 3 without perturbation prove to be identical when evolved over time, both in the inviscid and viscous case, over 144 hours. For the inviscid case, results are plotted in figure 11, whereas the viscous case can be seen in figure 12. For the inviscid case, strong vortices can be seen evolving over time, with ringing artifacts becoming visible in the t=120 hours and t=144 hours snapshots. These ringing artifacts occur due to small floating point errors that cause instabilities, which then evolve into large-scale vortices. Compared to the results presented by Galewsky, differences within these artifacts could be attributed to a difference in datatype, as those simulation used float32 instead of float64.

On a larger scale, the Dedalus 3 vorticity fields dissipate quicker, and strong vorticity fronts (regions with tightly packed contour lines visible in figure 11 and 12) appear along the entirety of the planet. An additional issue when comparing the plots is a different in scaling, where the Dedalus 3 Test Results appear slightly more compressed in the longitudinal direction. This is likely due to a different in plotting methods, but cannot be easily verified due to the benchmark lacking direct data of the simulations. Because of this, visual comparison of the plots is only viable on large scales, with a focus on the distribution of contour lines and ringing artifacts. For an exact test of accuracy and precision of Dedalus 3 compared to the benchmark, the timestepper would be set to be leapfrog instead of Runge Kutta 2nd order, and the datatype tested for both float32 and float64 as it is unknown which datatype was used. For the simulations of this paper however, the numerical integration is taken with Runge Kutta 2nd order and float64 to reduce the sensitivity to floating point errors.

3.1.1 Deviation with Dedalus

Dedalus 3 does not match the values within the table if using the inviscid case, but the values do match closely with explicit diffusion taken into account, with values notes in table 2. A visual comparison between the values from NCAR BOB and FMS-SWM against Dedalus 3 can be seen in figure 11. Using strictly the instructions of Galewsky would result in the inviscid values being compared, meaning that Dedalus 3 does not match the standard of the benchmark. This is supported by the fact that Galewsky asks for comparing the vorticity and height field values to the tabular values before adding explicit diffusion, but the values within the table are ones that match for both NCAR BOB and FMS-SWM. NCAR BOB in the description of how the benchmark was constructed is only used in the viscous cases, and no for the inviscid cases. This would logically mean that the values found within the table in fact use the data from the case with explicit diffusion. Additionally, the ringing artifacts present in inviscid simulations are not present in the viscous case as can been seen in figure 12, and as a result are significantly less sensitive to floating point errors. Finding identical values between different models would then require matching datatypes, timesteppers, and spectral method computation. Because of this, it would be not a well-posed benchmark to be applied to models that use other datatypes or methods for computation. Correcting the instructions for the benchmark to have explicit diffusion applied first, and then looking for numerical convergence between a model and the benchmark is what is done here for testing Dedalus 3 on numerical convergence. Thus, as Dedalus 3 does match using this correction, it does work as numerical framework for a shallow water model on S2 spheres.

3.2 Results Case 2: Band Approximation Jupiter

Using the JWST data of the zonal wind profile, and the Gaussian approximation of said data, the vorticity fields of Jupiter are visible in figure 13. All simulations are able to run without errors appearing, proving the Gaussian Approximation to be a proper way to describe the wind profile instead of the Fourier Approximation (although as stated earlier, a working model using the Fourier Approximation would have the additional benefit of not needing manual construction).

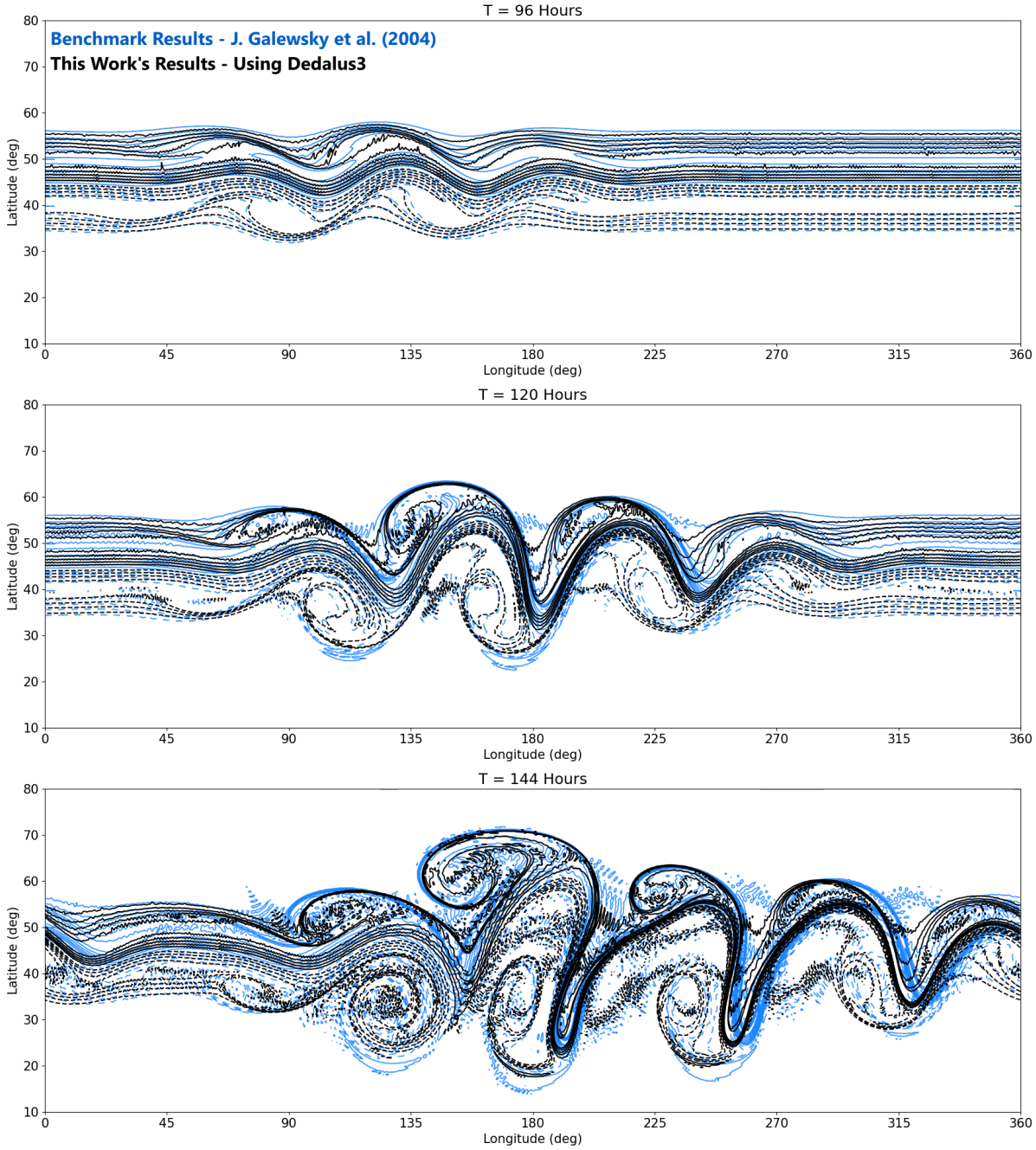

Figure 11: Time evolution of the vorticity field for the inviscid benchmark test using the Dedalus 3 spectral solver at a resolution of T341 with a timestep of 30 seconds. Displayed in blue are the results from FMS-SWM taken from J. Galewsky et al. (2004). Contour lines are defined at intervals of $2 \times 10^{-5}\,\mathrm{s}^{-1}$. Positive contours are solid while negative contours are dashed

3.2.1 Stability

The zonal wind profile simulation of Jupiter with the 30 second unperturbed simulation of Jupiter with spectral resolution T341 visible in figure 13. This model retains the same structure in the velocity profile over the span of 150 hours, with no eddies or instabilities arising during this period. For the model with stirring, the lower bands of Jupiter especially become unstable quickly as can be seen

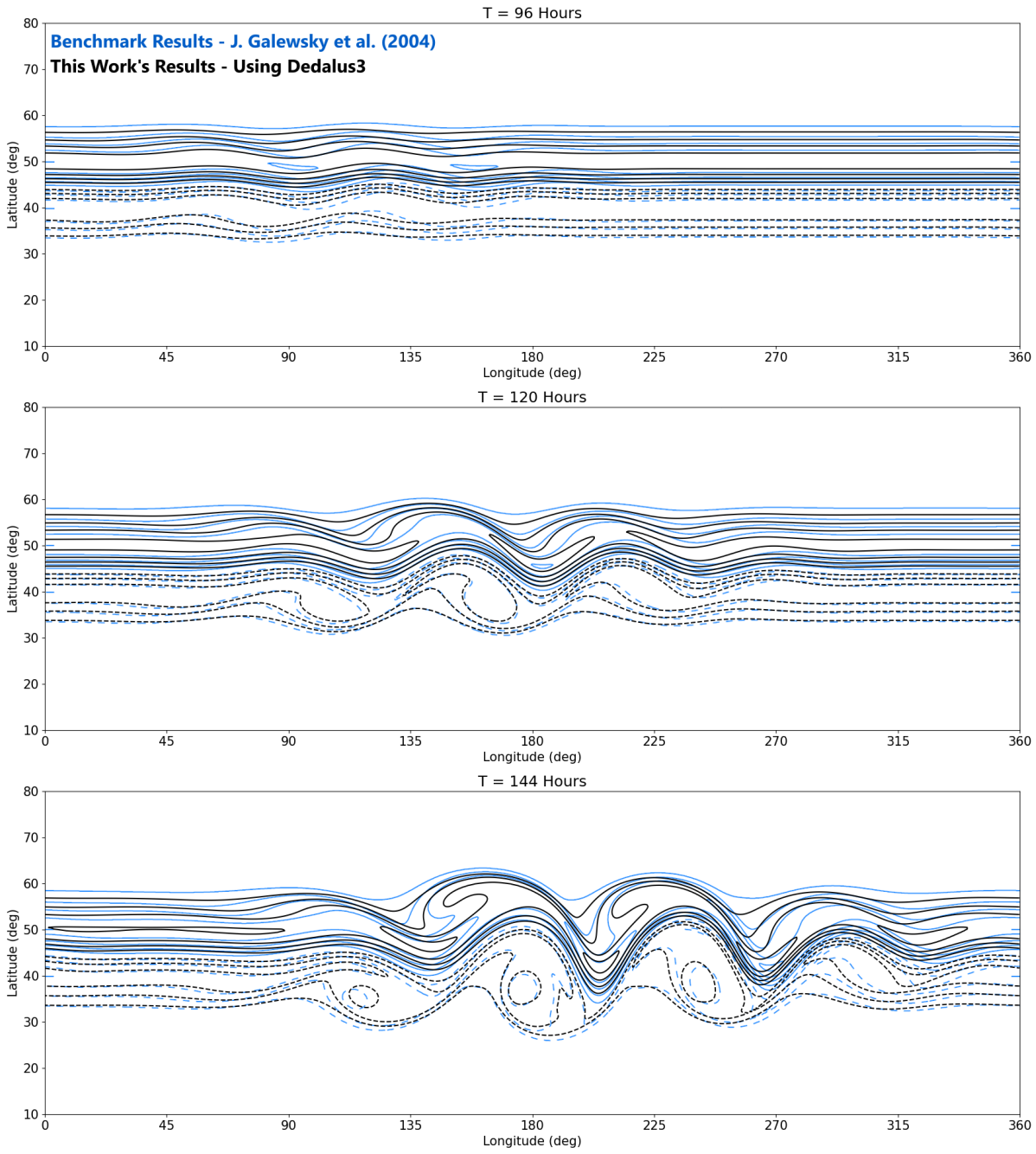

Figure 12: Time evolution of the vorticity field for the viscous benchmark test, using the same properties as in figure 11, with added viscosity of $\nu = 1.0 \times 10^5\,\mathrm{m}^2/\mathrm{s}$. Displayed in blue are the results from FMS-SWM taken from J. Galewsky et al. (2004).

in figure 14, with Kelvin-Helmholtz instabilities destroying the region featuring both prograde and retrograde bands. Both of these models use a reduced viscosity of $1 \times 10^{-10}\,\mathrm{m}^2/\mathrm{s}$ to approach an inviscid case, where convergence here will be convergent for more viscous cases. This same viscosity is used for the convergence tests of model to properly verify whether or not this is a well-posed simulation for Jupiter, especially given the simplification from the Gaussian bands.

One major feature of Jupiter that has not been dis-

	Galewsky	Dedalus 3	Difference
$t = 4$ h			
h_{min}	9052 m	9057 m	5 m (0.06 %)
h_{max}	10 182 m	10 166 m	16 m (0.16 %)
$t = 144$ h			
ζ_{min}	$-7.3 \times 10^{-5}\,\mathrm{s}^{-1}$	$-7.35 \times 10^{-5}\,\mathrm{s}^{-1}$	$0.05 \times 10^{-5}\,\mathrm{s}^{-1}$ (0.68 %)
ζ_{max}	$9.3 \times 10^{-5}\,\mathrm{s}^{-1}$	$9.17 \times 10^{-5}\,\mathrm{s}^{-1}$	$0.13 \times 10^{-5}\,\mathrm{s}^{-1}$ (1.4 %)

Table 2: Benchmark values between the Galewsky benchmark and Dedalus 3, values from Galewsky are taken from J. Galewsky et al. (2004).

cussed is the Red Spot on the lower hemisphere of the planet. This large-scale storm features multiple eddies across the entire region and has been left out of the zonal wind profile for simplification. The region with opposing bands on the lower hemisphere that destabilises quickly with stirring is at approximately the same latitude as where the Red Spot would be. While height forcing would quickly destroy this storm, future simulations that do add the Red Spot would allow for verifying whether or not the storm is a major factor in the structure of the atmosphere's upper layer.

3.2.2 Convergence

For the convergence cases, simulations provide results without any overflow errors for the T170 and T341 cases, but math overflow and 'out of memory' errors are returned for T682 and would need memory outside the available range. Experimentation with high memory partitions could allow higher resolution models to be ran, but for numerical convergence T341 is a sufficiently high resolution, as supported by the Galewsky benchmark. Tests with a timestep exceeding 150 seconds resulted in math overflow errors as well, which likely is due to timestepping exceeding the possible size for its resolution. Specifically this means that the Courant-Friedrichs-Lewy condition is violated, which requires that a fluid element in the simulation cannot travel to an adjacent grid point for a given timestep (R. Courant et al., 1967).

Convergence is done over different resolutions and timesteps, with the different timesteps of 30 seconds, 50 seconds, and 150 seconds displayed in figure 15. Vorticity and height values of each of the simulations at t = 500 h can be seen in table 3 showing full numerical convergence across the differing timesteps of the simulation. Of note is that this level of matching precision in fact exceeds what would be expected, as the Galewsky benchmark from Case 1 has a far simpler atmospheric structure and required timesteps of 30 seconds for numerical convergence. This implies Dedalus 3 simulations potentially being too stable, but due to a lack of insight into the inner code of Dedalus 3 this cannot be confirmed. As such, other frameworks like FMS-SWM or NCAR BOB can be tested with the same initial conditions for numerical convergence, and whether or not Dedalus 3 artificially stabilises simulations.

3.2.3 Computational time

Opposed to Case 1 simulations, simulations of Jupiter take a significantly longer period of time to finish. Simulations running at a timestep of 50 seconds will take approximately 1.8 days on spectral resolution T170 and 5.8 days on spectral resolution T341 for a total simulation time of 24000 hours. Some variation is possible dependent on the complexity of the simulation, such as running a perturbed or unperturbed simulation. Regarding potential exploration of T682 resolutions, the increase in computation time from T170 to T341 would imply that increasing to T682 would rapidly increase the computational needed to run a single simulation, given that a total CPU memory allocation of 1 TB would not return 'out of memory' errors. Due to this, T341 proves to be best for models that are ran for analysis, while T170 models are preferred for experimentation with initial values. If a T170 model proves to result in well-posed simulations, a T341 simulation using the same initial parameters

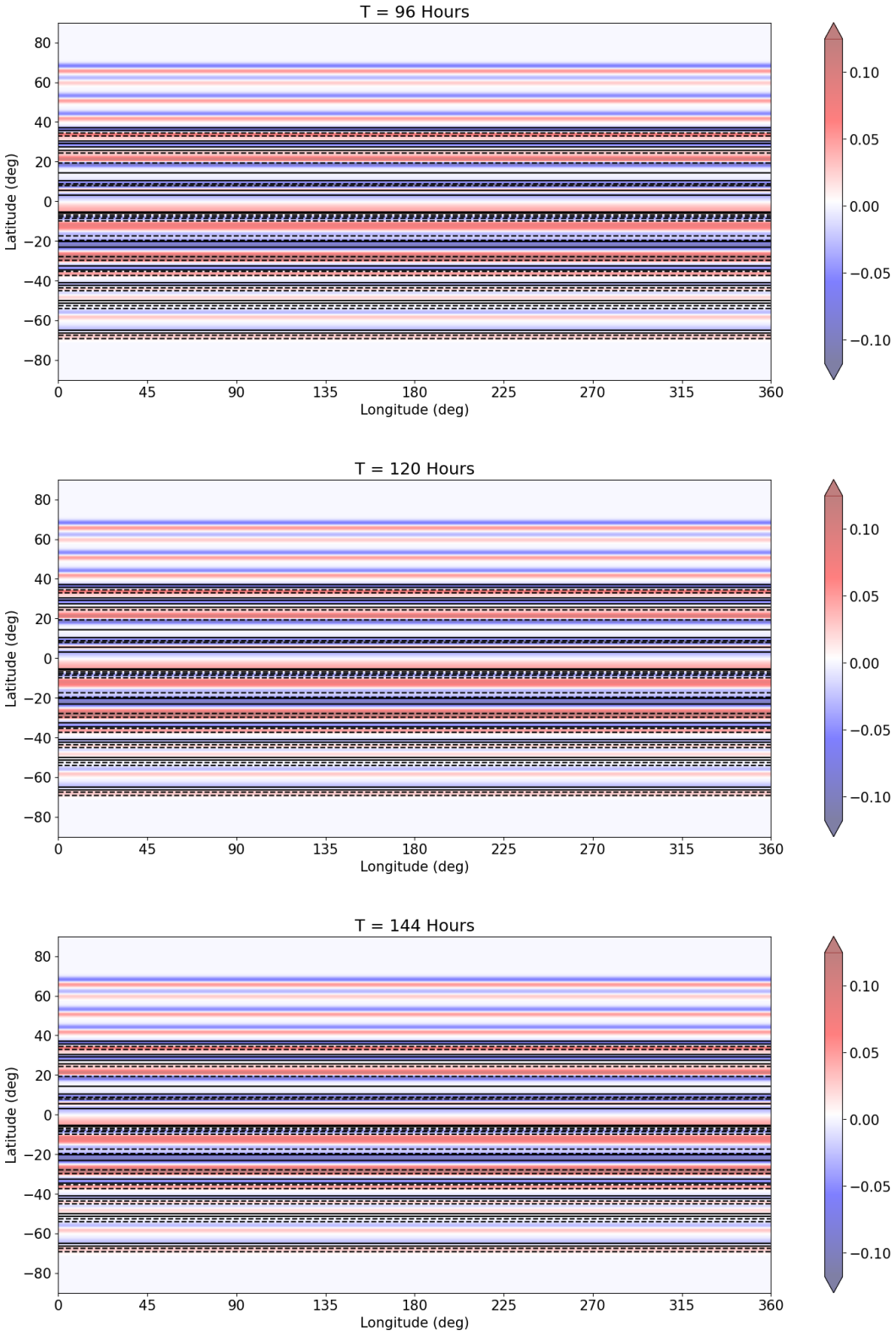

Figure 13: Working case of Jupiter with the Gaussian band approximation as in figure 9 with no stirring or perturbation. Contours are displayed on top at intervals of $\zeta = 2 \times 10^{-5}\,\mathrm{s}^{-1}$. For full animations of the simulation, see the video linked in the Appendix.

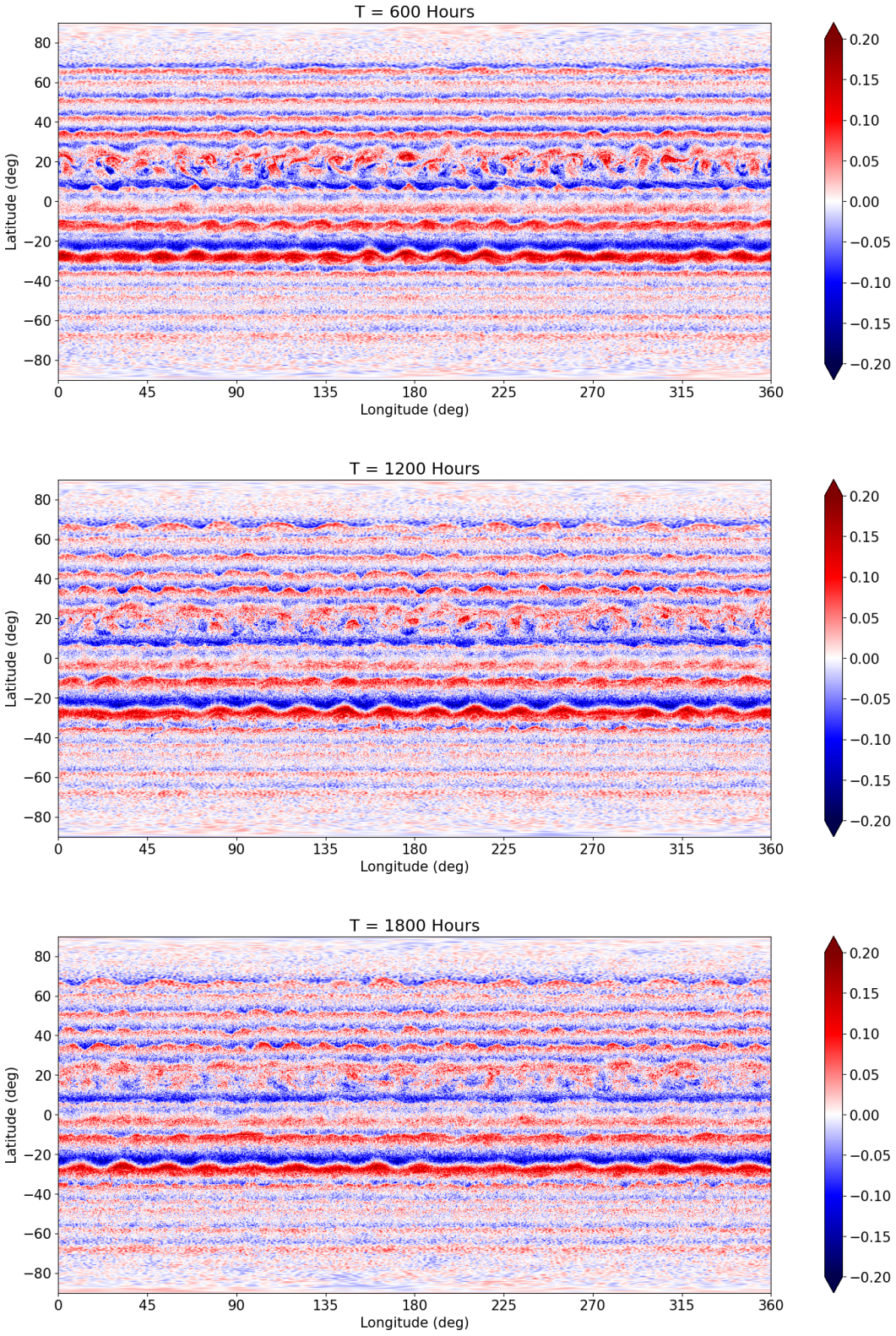

Figure 14: Vorticity plots of the working case of Jupiter with the Gaussian band approximation as in figure 9 with random noise stirring present in the height and velocity fields. For full animations of the simulation, see the video linked in the appendix.

Timestep	Minimum ζ	Maximum ζ
30 s	$-6.527\,66 \times 10^{-5}\,\mathrm{s}^{-1}$	$6.232\,55 \times 10^{-5}\,\mathrm{s}^{-1}$
50 s	$-6.527\,66 \times 10^{-5}\,\mathrm{s}^{-1}$	$6.232\,55 \times 10^{-5}\,\mathrm{s}^{-1}$
150 s	$-6.527\,66 \times 10^{-5}\,\mathrm{s}^{-1}$	$6.232\,55 \times 10^{-5}\,\mathrm{s}^{-1}$

Table 3: Vorticities of the convergence test ran at spectral resolution T341, with the snapshot taken at 500 hours simulated time.

would then allow for analysis of said model to verify the results.

3.3 Results Case 3: Height Forcing and Initial Value Problem

With the new initial conditions for the height forcing case of the Hot Jupiter, the simulation is ran with a timestep of 20 seconds up to 2000 hours. As can be seen in figure 16, the atmospheric banding of Jupiter is dissolved over a timescale of days, with the large scale structure already having collapsed at 90 hours. After this, the atmosphere behaves chaotically, with random asymmetries across the latitude having formed at T = 190 h. Of interest is the snapshot at T = 1000 h, as polar vortices found similar in structure as the ones observed in J. Y.-K. Cho et al. (2003) have emerged and thus match expected characteristics. Furthermore, the completely chaotic behaviour of the atmosphere results in the initial wind profile of the planet not being retained.

On top of the vorticity field following as results would be expected, the height field evolves smoothly over time, as can be seen in figure 17. Near the poles of the simulation a region of reduced height is visible, proportional to a region with reduced temperature. Given the results from J. Y.-K. Cho et al. (2003) of the temperature profile of HD 209458b in figure 1, gradual regions near the poles with reduced height would be expected, as to match the expected temperature profile. As these features are visible in both, Dedalus 3 manages to reproduce characteristics expected in hot Jupiters. Comparing figure 16 with figure 17 does seem to globally match the shape between the two fields as well, with sheer lines in vorticity being matched with sheer lines in the height field. Additional testing for height convergence between this model (i.e. Dedalus 3) with other models such as FMS-SWM and NCAR BOB is recommended to see if there is a significant difference in resulting height fields. Of note is that due to the chaotic evolution of the atmosphere, only large-scale characteristics would be considered.

For additional testing, analysis of the kinetic energy spectrum would be recommended such that any unstably evolving eddies within the simulation can be ruled out.

3.3.1 Comparison between JWST Profile Against Empty Profile

Comparing both models with the initial wind profile and the one without, strong global differences can be seen early in the simulation in both the vorticity and height profile, as seen in figure 16 and 17 respectively. For simplicity, the JWST zonal wind profile in this section will be referred to as the *JWST* profile, and the empty wind profile as *zero* profile. Due to stronger initial variation in vorticity in the JWST profile, eddies form as early as T = 190 h, and have a greater number of strong vortices across the surface at T = 1000 h, best seen in figure 18. Figure 18 additionally provides a clearer view of the north pole of the planet, along with the antistellar point (i.e. the center of where the height forcing is applied). Polar vortices with the JWST profile are strongly cyclonic, matching simulations from J. Y.-K. Cho et al. (2003); J. Langton & G. Laughlin (2007); J. Skinner & J.-K. Cho (2021). Ignoring small-scale dynamics of the atmosphere, the JWST profile additionally matches large-scale regions of vorticity and height with the zero profile early in the simulation. At long timescales the two cases diverge strongly, where the zero profile lacks the height profile features seen in J. Y.-K. Cho et al. (2003).

Of note are the anti-cyclonic polar vortices in the zero profile, characteristically different from other simulations. These polar vortices do not evolve

from large-scale dynamics, but instead are shaped by small-scale eddies early on. Due to the chaotic nature of these eddies, the orientation of the polar vortices is highly dependent on what small-scale dynamics dominate the poles. For the JWST profile, the large-scale structure of the planet, does seem influence the rotation of the polar vortices, making the orientation less sensitive to small-scale structure. However, large circumpolar storms can be seen at T = 1000 h for both models, and it is unclear whether the JWST or zero profiles retain their large-scale structures over longer timescales.

For both profiles strong prograde equatorial jets can be seen, with few vortices around said region. This is expected, and the zero profile follows the general structure of hot exoplanetary atmospheres (J. Y.-K. Cho et al., 2003; J. Langton & G. Laughlin, 2007; J. W. Skinner & J. Y.-K. Cho, 2025). The strong zonal jet with vorticity regions directed to the poles can be seen towards the western hemisphere in the zero profile at T = 1000 h, matching to the structure in T341 simulations from J. W. Skinner & J. Y.-K. Cho (2025). Of note is the symmetry retained here due to the explicit diffusion damping small-scale eddies, while simulations at greater orders of diffusion allow for far lower ν and more accurate small-scale evolution (J.W. Skinner for example uses up to ∇^{16}). As such, additional research for testing initial wind conditions should be ran at greater orders of diffusion to accurately simulate small-scale features, with a minimum of ∇^{12} recommended for accurate numerical convergence that is not over-dissipated (J. Skinner & J.-K. Cho, 2021). Tests for numerical convergence of Case 2 should be done first before attempting high order simulations of Case 3 to see whether Dedalus 3 maintains numerical conversion at higher orders.

Thus, both profiles are highly dependent on short timescale evolutions (i.e. t =20 s) at high resolutions for characteristics of hot Jupiter atmospheres and cannot be easily neglected, with the initial JWST wind profile strongly influencing the rotation of the atmosphere's polar vortices.

4 Conclusion

In this thesis, we have summarised the state of the field of exoplanetary atmospheres and the theoretical models used to predict and analyse dynamics found on hot Jupiters. Simulations of hydrodynamical IVPs are critical to understand the characteristics found in observations, such as the phase curve offsets to determine the convection between the day- and nightside. Initial conditions and the sensitivity of end states based on these conditions are vital components for the general structure and characteristics of exoplanetary atmospheres. As such, high resolutions and numerical convergence are required for numerical models to accurately predict behaviour of these atmospheres.

To test the sensitivity of initial wind profiles to the characteristics of hot Jupiters for high resolution models, the Shallow Water Model benchmark by J. Galewsky et al. (2004) is used to test Dedalus 3, a pseudo spectral solver using the sparse tau method. We find a close match to the benchmark values, with differences of $\zeta_{min} = 0.05 \cdot 10^{-5}$ (0.68 %), $\zeta_{max} = 0.13 \cdot 10^{-5}$ (1.4 %), $h_{min} = 5$ (0.06 %), $h_{max} = 16$ (0.16 %) between the values from FMS-SWM and NCAR BOB with the values from Dedalus 3. The height field is more closely resolved to what the values in the benchmark report, whereas the vorticity field values do have a stronger deviation. This is likely due to Dedalus 3 using a different numerical form of diffusion opposed to the other models, but lack of insight into the inner structure of the solver means that the direct cause for this difference is unknown. Nevertheless, Dedalus 3 proves to be an accurate solver for these idealised simulations and can be applied to use cases such as Jupiter-like planets.

The model for the benchmark is adjusted for a Jupiter-scaled planet, and zonal wind profile data is given as an initial condition for the velocity, constructed manually through a sum of jets modified from the Galewsky benchmark. This simulation is numerically stable over 240 hours, to which initial stirring to the velocity and height profile is applied with $\nu = 1 \times 10^{-10}$ m^2/s. This simulation is subsequently perturbed and evolved for 500 hours, and is convergent to 6 significant figures for timesteps of 30, 50, and 150 seconds. This is extremely sta-

ble, even moreso when compared to the deviation found in the benchmark. The cause to this as well is unknown due to the inner workings of Dedalus 3 being unknown. Results from both the benchmark and the use of Dedalus 3 for hot Jupiter simulations have been written as well as a research note, that is of the time of writing yet pending publication. This document can be read below in section 6.1.

Lastly, two hot Jupiters are simulated through height forcing at the equator, with one simulation ran with the JWST initial wind profile and one with a zero wind profile. Both are evolved to T = 2000 h with a timestep of 20 s and compared to existing research. The model with a nonzero wind velocity profile matches at T = 1000 h characteristically with height profile features and exhibits cyclonic polar vortices. Additionally, the complex structure of Jupiter's bands result in a more chaotic system, with strong vortices across the entirety of the planet. The model with no initial velocities instead has anticyclonic polar vortices, with fewer strong vortices present across the surface. However, small-scale features evolving into large-scale vortices do result in it being unclear whether or not these characteristics will remain the same over longer timescales. Similar to weather forecast models on earth, the state of the atmosphere changes rapidly over short periods of time due to numerical differences rapidly diverging from small changes in initial conditions. Because of this, the states found in both models might be a transient state and not a stable end state of the atmosphere, but it is not possible with this constructed idealised model to concretely conclude if it is transient or not.

To answer the initial question **"Does the initial wind profile of a hot Jupiter simulation have a significant impact on the atmosphere's structure when evolved over time?"** we find that the initial wind profile does influence the polar vortices and characteristics on a short timescale, but high dependence on small-scale eddies make the simulation sensitive to the resolution and order of diffusion. As such, the question has two parts as answers. *Yes*, the initial wind profile does have a significant impact on the structure of short timescale simulations, *however* for long timescale simulations it is unknown whether this will be preserved, or achieve a different structure that is indistinguishable from simulations that start from rest. Simulations ran for greater lengths of time at higher resolution and orders of diffusion would be able to study the dynamics of the atmosphere for long timescales, whereas this serves as a push for action with the given limited scope of a thesis.

If given the opportunity to do research further into this field, a promising direction to take this study would be to address the caveats of the model used here. The model uses a fully stratified upper layer of the atmosphere, and lacks any movement between potential lower layers. Along with this, the resolution and order of diffusion of the model are too low to accurately represent small-scale features, which do dictate the general structure over extended periods of time. Finally, the height forcing applied to the planet is a simplified version of an irradiated hot Jupiter, with many factors such as tidal forcing, synchronous rotation and varying optical thicknesses across the layers not being taken into account. Due to all of these shortcomings of an idealised model such as the one used here, research into the sensitivity of initial wind profiles with a more robust and complex model would be very promising. Equations such as the primitive equations would permit complex three-dimensional motion through different layers of the atmosphere, providing insight into not only the uppermost observable layer of the atmosphere, but also into deeper layers and the convection found within. This, of course, would require a complete new construction of the planet with greater resources spent on computation of the planet, but would be a research topic worth researching beyond the scope of a MSc thesis.

Scripts used, along with plots and animations of the data, are linked below in the Appendix.

5 Acknowledgements

We would like to thank the Center for Information Technology of the University of Groningen for having provided access to the Hábrók HPC. Additionally, we would like to thank Prof. James Y-K. Cho and Dr. Jack William Skinner for providing feedback throughout the thesis when verifying the

benchmark tests and constructing the Jupiter profile (alongside the many crash cases the program would have). Lastly, we would like to thank the members of the exoAIM group for the weekly meetings where intermediary results were able to be presented and commented on.

Neither ChatGPT nor any other LLM was used to write, make, or assist in making this thesis. This is stated explicitly as there has been a growing precedent in use of these programs in works (S. Astarita et al., 2024; M. Kaushik et al., 2024). As such it is important to note whenever solely human work has been done for research.

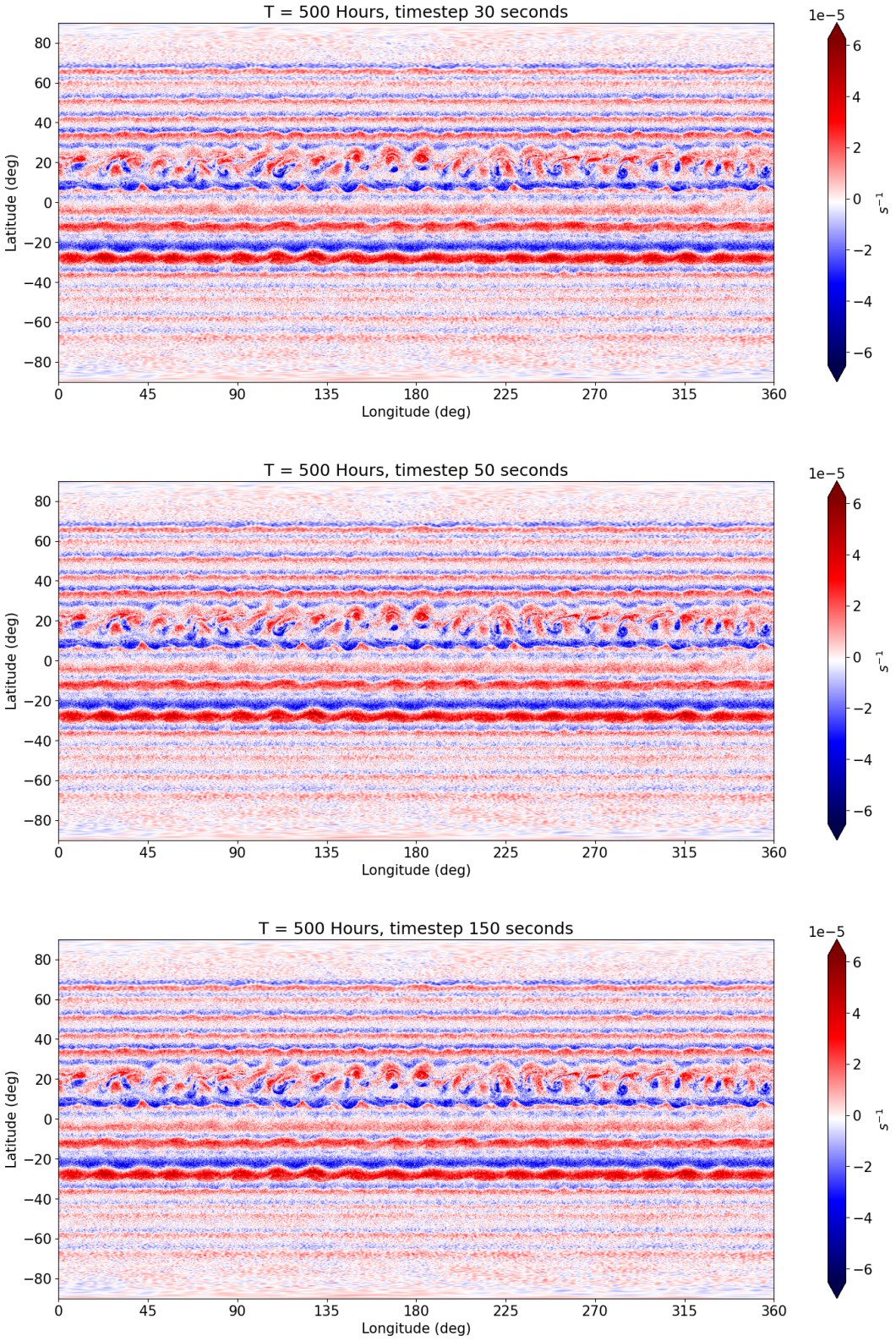

Figure 15: Vorticity plots of the working case of Jupiter with the Gaussian band approximation as in figure 9 with random noise stirring and strong perturbation in the height and velocity field. Displayed are the vorticity plots for the simulations with timesteps of 30 seconds, 50 seconds, and 150 seconds respectively from top to bottom. For full animations of the simulation, see the video linked in the appendix.

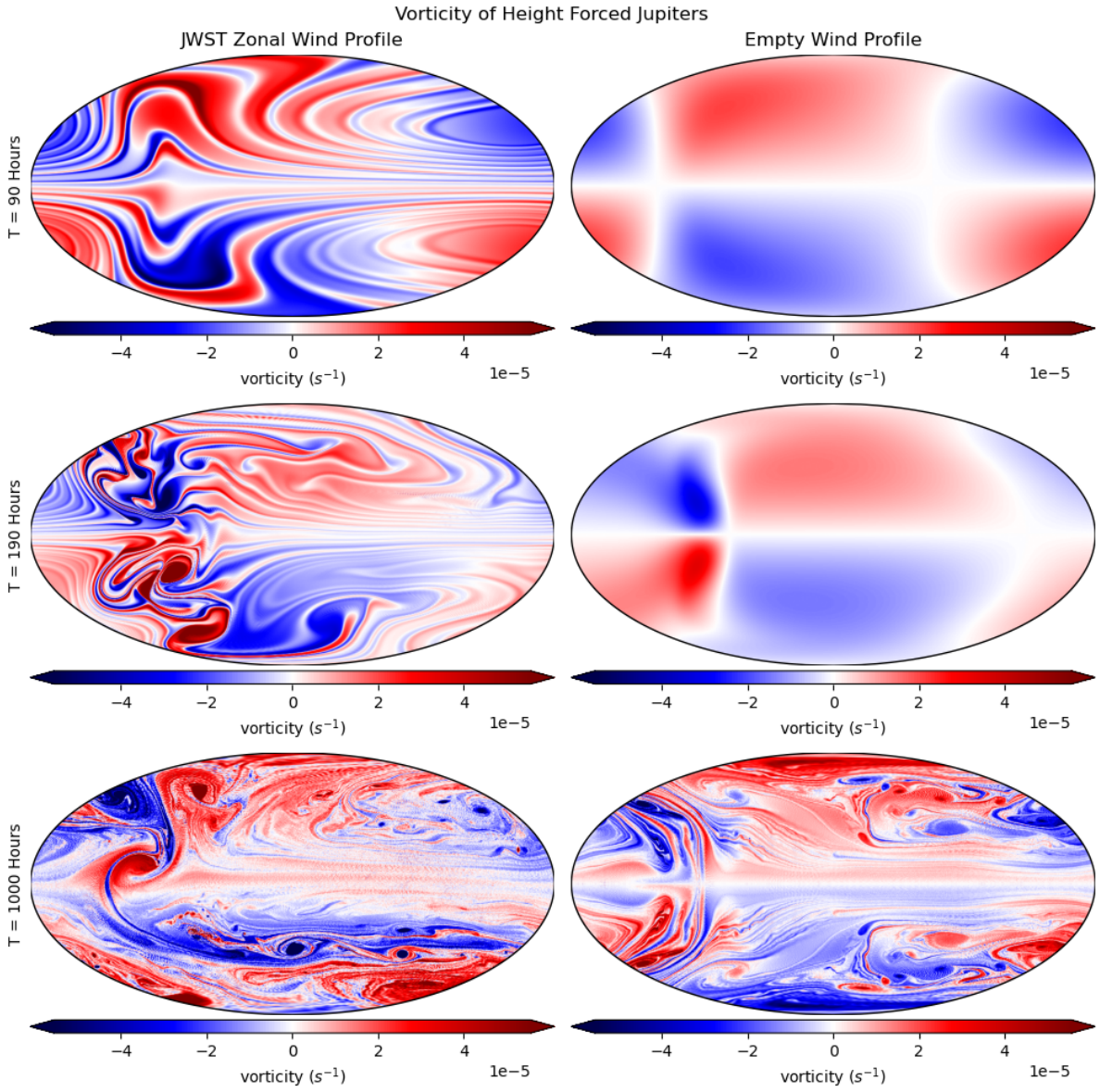

Figure 16: Mollweide projected vorticity plots of the working case of Jupiter with the Gaussian band approximation as in figure 9 and height forcing with random noise stirring in the height field. Note the chaotic structure of the atmosphere at 1000 hours. For full animations of the simulation, see the video linked in the appendix.

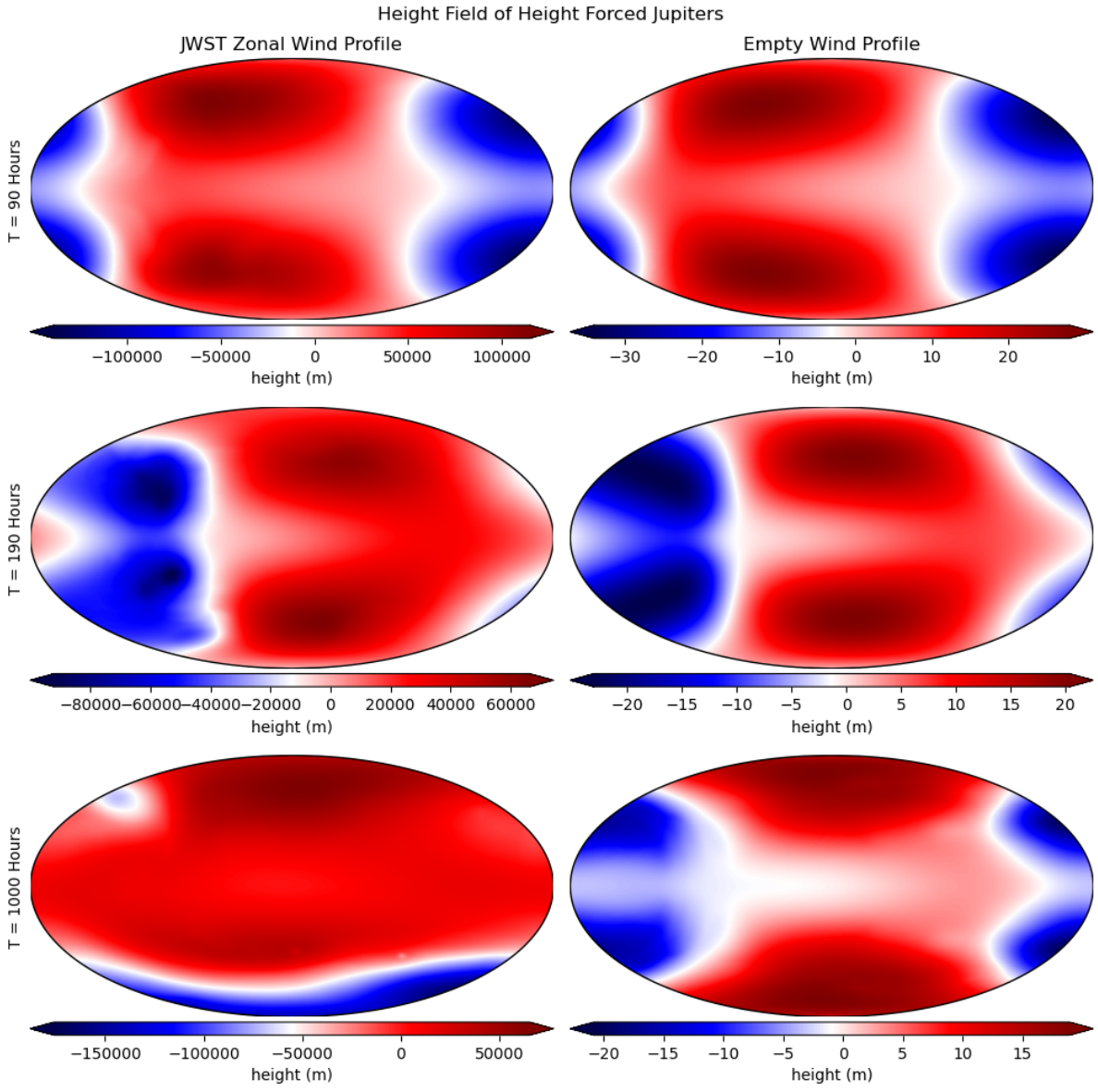

Figure 17: Mollweide projected height plots of the working case of Jupiter with the Gaussian band approximation as in figure 9 and height forcing with random noise stirring in the height field. For full animations of the simulation, see the video linked in the appendix.

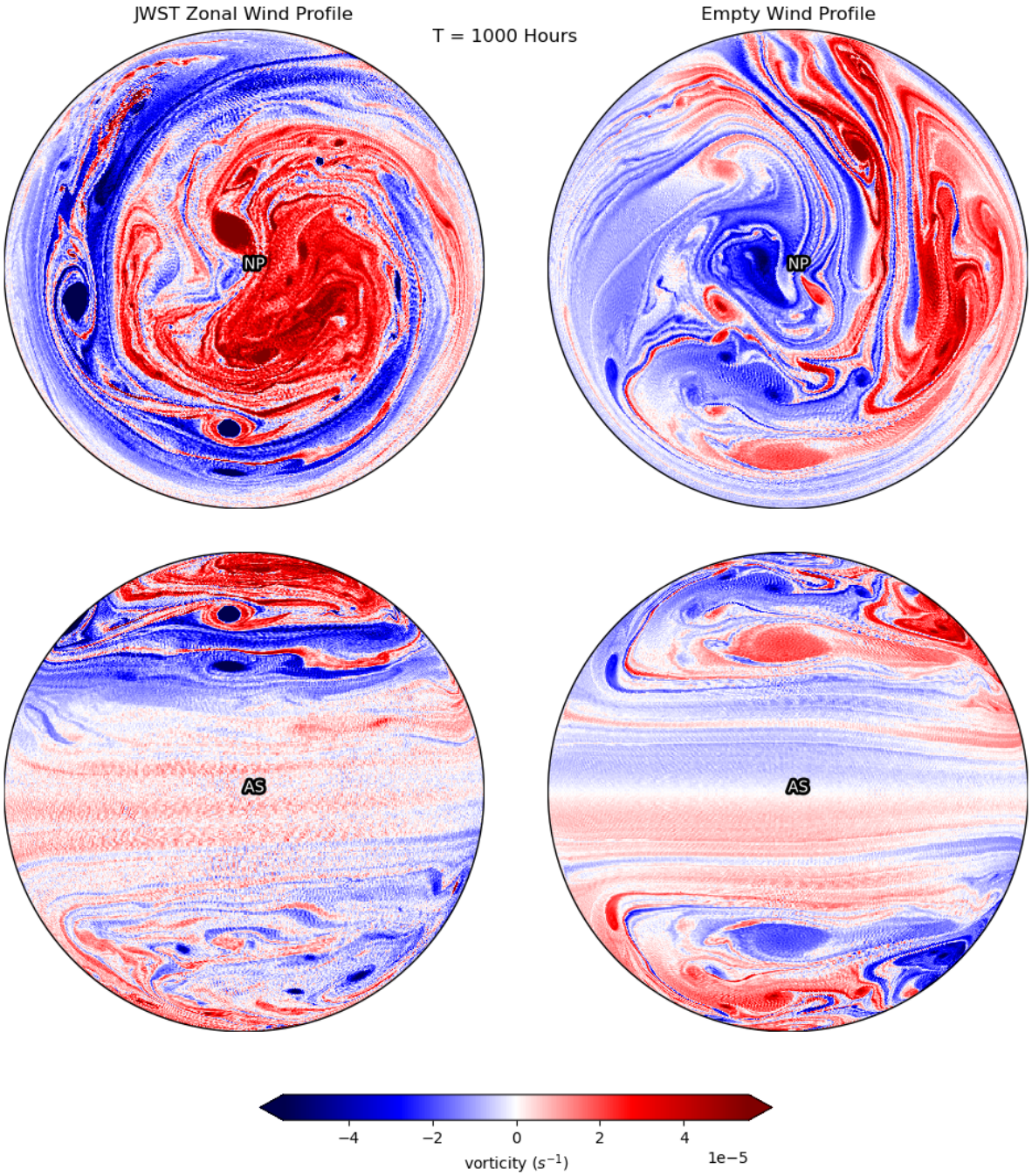

Figure 18: Vorticity plots of the height forced hot Jupiter, for both the JWST initial wind profile and empty profile, with orthographic projections from both the north poles and antistellar points.

References

Armstrong, D. J., de Mooij, E., Barstow, J., et al. 2016, Variability in the atmosphere of the hot giant planet HAT-P-7 b, Nature Astronomy, 1, 0004, doi: 10.1038/s41550-016-0004

Astarita, S., Kruk, S., Reerink, J., & Gómez, P. 2024, Delving into the Utilisation of ChatGPT in Scientific Publications in Astronomy, in Proceedings of SPAICE2024: The First Joint European Space Agency / IAA Conference on AI in and for Space, ed. D. Dold, A. Hadjiivanov, & D. Izzo, 241–246, doi: 10.5281/zenodo.13885577

Bodenheimer, P., Lin, D. N. C., & Mardling, R. A. 2001, On the Tidal Inflation of Short-Period Extrasolar Planets, , 548, 466, doi: 10.1086/318667

Burns, K. J., Vasil, G. M., Oishi, J. S., Lecoanet, D., & Brown, B. P. 2020, Dedalus: A flexible framework for numerical simulations with spectral methods, Physical Review Research, 2, 023068, doi: 10.1103/PhysRevResearch.2.023068

Changeat, Q., Skinner, J. W., Cho, J. Y.-K., et al. 2024, Is the Atmosphere of the Ultra-hot Jupiter WASP-121 b Variable?, , 270, 34, doi: 10.3847/1538-4365/ad1191

Charbonneau, D., Brown, T. M., Noyes, R. W., & Gilliland, R. L. 2002, Detection of an Extrasolar Planet Atmosphere, , 568, 377, doi: 10.1086/338770

Cho, J. Y.-K., Menou, K., Hansen, B. M. S., & Seager, S. 2003, The Changing Face of the Extrasolar Giant Planet HD 209458b, , 587, L117, doi: 10.1086/375016

Cho, J. Y.-K., Menou, K., Hansen, B. M. S., & Seager, S. 2008, Atmospheric Circulation of Close-In Extrasolar Giant Planets. I. Global, Barotropic, Adiabatic Simulations, The Astrophysical Journal, 675, 817, doi: 10.1086/524718

Cho, J. Y. K., & Polvani, L. M. 1996, The emergence of jets and vortices in freely evolving, shallow-water turbulence on a sphere, Physics of Fluids, 8, 1531, doi: 10.1063/1.868929

Cho, J. Y.-K., & Polvani, L. M. 1996, The Morphogenesis of Bands and Zonal Winds in the Atmospheres on the Giant Outer Planets, Science, 273, 335, doi: 10.1126/science.273.5273.335

Christiansen, J. L., McElroy, D. L., Harbut, M., et al. 2025, The NASA Exoplanet Archive and Exoplanet Follow-up Observing Program: Data, Tools, and Usage, The Planetary Science Journal, 6, 186, doi: 10.3847/PSJ/ade3c2

Cooper, C. S., & Showman, A. P. 2005, Dynamic Meteorology at the Photosphere of HD 209458b, , 629, L45, doi: 10.1086/444354

Courant, R., Friedrichs, K., & Lewy, H. 1967, On the partial difference equations of mathematical physics, IBM J. Res. Dev., 11, 215–234, doi: 10.1147/rd.112.0215

Dobbs-Dixon, I., & Lin, D. N. C. 2008, Atmospheric Dynamics of Short-Period Extrasolar Gas Giant Planets. I. Dependence of Nightside Temperature on Opacity, , 673, 513, doi: 10.1086/523786

Efstathiou, G., & Eastwood, J. W. 1981, On the clustering of particles in an expanding universe, , 194, 503, doi: 10.1093/mnras/194.3.503

Galewsky, J., Scott, R. K., & Polvani, L. M. 2004, An initial-value problem for testing numerical models of the global shallow-water equationsErratum printed in: Tellus A 58:4 p. 520 - 522doi:10.1111/j.1600-0870.2006.00192.x, Tellus Series A, 56, 429, doi: 10.3402/tellusa.v56i5.14436

Greengard, L. 1991, Spectral Integration and Two-Point Boundary Value Problems, SIAM Journal on Numerical Analysis, 28, 1071, doi: 10.1137/0728057

Guillot, T., Burrows, A., Hubbard, W. B., Lunine, J. I., & Saumon, D. 1996, Giant Planets at Small Orbital Distances, , 459, L35, doi: 10.1086/309935

Hueso, R., Sánchez-Lavega, A., Fouchet, T., et al. 2023, An intense narrow equatorial jet in Jupiter's lower stratosphere observed by JWST, Nature Astronomy, 7, 1454, doi: 10.1038/s41550-023-02099-2

Jackson, B., Adams, E., Sandidge, W., Kreyche, S., & Briggs, J. 2019, Variability in the Atmosphere

of the Hot Jupiter Kepler-76b, , 157, 239, doi: 10.3847/1538-3881/ab1b30

Juckes, M., & McIntyre, M. 1987, A high-resolution one-layer model of breaking planetary waves in the stratosphere, Nature, 328, doi: 10.1038/328590a0

Julien, K., & Watson, M. 2009, Efficient multi-dimensional solution of PDEs using Chebyshev spectral methods, Journal of Computational Physics, 228, 1480, doi: 10.1016/j.jcp.2008.10.043

Kaushik, M., Mattoo, A., & Rastogi, R. 2024, Exoplanet Detection : A Detailed Analysis, https://arxiv.org/abs/2404.09143

Knutson, H. A., Lewis, N., Fortney, J. J., et al. 2012, 3.6 and 4.5 μm Phase Curves and Evidence for Non-equilibrium Chemistry in the Atmosphere of Extrasolar Planet HD 189733b, , 754, 22, doi: 10.1088/0004-637X/754/1/22

Komacek, T. D., & Showman, A. P. 2016, ATMOSPHERIC CIRCULATION OF HOT JUPITERS: DAYSIDE–NIGHTSIDE TEMPERATURE DIFFERENCES, The Astrophysical Journal, 821, 16, doi: 10.3847/0004-637x/821/1/16

Komacek, T. D., & Tan, X. 2018, Effects of dissociation/recombination on the day-night temperature contrasts of ultra-hot Jupiters, https://arxiv.org/abs/1805.07415

Kreidberg, L., Line, M. R., Parmentier, V., et al. 2018, Global Climate and Atmospheric Composition of the Ultra-hot Jupiter WASP-103b from HST and Spitzer Phase Curve Observations, The Astronomical Journal, 156, 17, doi: 10.3847/1538-3881/aac3df

Langton, J., & Laughlin, G. 2007, Observational Consequences of Hydrodynamic Flows on Hot Jupiters, The Astrophysical Journal, 657, L113, doi: 10.1086/513185

Langton, J., & Laughlin, G. 2008, Hydrodynamic Simulations of Unevenly Irradiated Jovian Planets, , 674, 1106, doi: 10.1086/523957

Mayor, M., & Queloz, D. 1995, A Jupiter-mass companion to a solar-type star, , 378, 355, doi: 10.1038/378355a0

Mendes, N., Chhay, M., Berger, J., & Dutykh, D. 2019, Spectral Methods (Springer International Publishing), 167–209, doi: 10.1007/978-3-030-31574-0_8

Menou, K., & Rauscher, E. 2009, Atmospheric Circulation of Hot Jupiters: A Shallow Three-Dimensional Model, , 700, 887, doi: 10.1088/0004-637X/700/1/887

Nelson, D., Springel, V., Pillepich, A., et al. 2019, The IllustrisTNG simulations: public data release, Computational Astrophysics and Cosmology, 6, 2, doi: 10.1186/s40668-019-0028-x

Olver, S., & Townsend, A. 2013, A Fast and Well-Conditioned Spectral Method, SIAM Review, 55, 462, doi: 10.1137/120865458

Orszag, S. A. 1970, Transform Method for the Calculation of Vector-Coupled Sums: Application to the Spectral Form of the Vorticity Equation., Journal of the Atmospheric Sciences, 27, 890, doi: 10.1175/1520-0469(1970)027<0890:TMFTCO>2.0.CO;2

Perez-Becker, D., & Showman, A. P. 2013, Atmospheric Heat Redistribution on Hot Jupiters, , 776, 134, doi: 10.1088/0004-637X/776/2/134

Rafiq, M. H., Raza, N., & Jhangeer, A. 2023, Dynamic study of bifurcation, chaotic behavior and multi-soliton profiles for the system of shallow water wave equations with their stability, Chaos, Solitons Fractals, 171, 113436, doi: https://doi.org/10.1016/j.chaos.2023.113436

Rein, H. 2012, A proposal for community driven and decentralized astronomical databases and the Open Exoplanet Catalogue, arXiv e-prints, arXiv:1211.7121, doi: 10.48550/arXiv.1211.7121

Rivier, L., Loft, R., & Polvani, L. M. 2002, An Efficient Spectral Dynamical Core for Distributed Memory Computers, Monthly Weather Review, 130, 1384, doi: 10.1175/1520-0493(2002)130<1384:AESDCF>2.0.CO;2

Rogowski, M., Aseeri, S., Keyes, D., & Dalcin, L. 2023, mpi4py.futures: MPI-Based Asynchronous Task Execution for Python, IEEE Transactions on Parallel and Distributed Systems, 34, 611, doi: 10.1109/TPDS.2022.3225481

Salby, M. L. 1989, Deep circulations under simple classes of stratification, Tellus Series A, 41, 48, doi: 10.3402/tellusa.v41i1.11820

Schwartz, J. C., Kashner, Z., Jovmir, D., & Cowan, N. B. 2017, Phase Offsets and the Energy Budgets of Hot Jupiters, The Astrophysical Journal, 850, 154, doi: 10.3847/1538-4357/aa9567

Seidelmann, P. K., Archinal, B. A., A'hearn, M. F., et al. 2007, Report of the IAU/IAG Working Group on cartographic coordinates and rotational elements: 2006, Celestial Mechanics and Dynamical Astronomy, 98, 155, doi: 10.1007/s10569-007-9072-y

Showman, A. P., Cooper, C. S., Fortney, J. J., & Marley, M. S. 2008, Atmospheric Circulation of Hot Jupiters: Three-dimensional Circulation Models of HD 209458b and HD 189733b with Simplified Forcing, , 682, 559, doi: 10.1086/589325

Showman, A. P., & Guillot, T. 2002, Atmospheric circulation and tides of "51 Pegasus b-like" planets, , 385, 166, doi: 10.1051/0004-6361:20020101

Shporer, A., & Hu, R. 2015, Studying Atmosphere-dominated Hot Jupiter Kepler Phase Curves: Evidence that Inhomogeneous Atmospheric Reflection Is Common, , 150, 112, doi: 10.1088/0004-6256/150/4/112

Skinner, J., & Cho, J.-K. 2021, Numerical convergence of hot-Jupiter atmospheric flow solutions, Monthly Notices of the Royal Astronomical Society, 504, 5172–5187, doi: 10.1093/mnras/stab971

Skinner, J. W., & Cho, J. Y.-K. 2022, Modons on tidally synchronized extrasolar planets, , 511, 3584, doi: 10.1093/mnras/stab2809

Skinner, J. W., & Cho, J. Y.-K. 2025, Early Time Small-scale Structures in Hot Exoplanet Atmosphere Simulations, , 982, 64, doi: 10.3847/1538-4357/adb0ce

Skinner, J. W., & Wei, S. 2025, Bridging the Atmospheric Circulations of Hot and Warm Giant Exoplanets, arXiv e-prints, arXiv:2505.01397, doi: 10.48550/arXiv.2505.01397

Vallis, G. K. 2017, Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-Scale Circulation, doi: 10.1017/9781107588417

Werner, M. W., Roellig, T. L., Low, F. J., et al. 2004, The Spitzer Space Telescope Mission, , 154, 1, doi: 10.1086/422992

Wilson, J., Gibson, N. P., Lothringer, J. D., et al. 2021, Gemini/GMOS optical transmission spectroscopy of WASP-121b: signs of variability in an ultra-hot Jupiter?, , 503, 4787, doi: 10.1093/mnras/stab797

Wright, J. T., & Gaudi, B. S. 2013, Exoplanet Detection Methods, in Planets, Stars and Stellar Systems. Volume 3: Solar and Stellar Planetary Systems, ed. T. D. Oswalt, L. M. French, & P. Kalas, 489, doi: 10.1007/978-94-007-5606-9_10

Zhang, M., Knutson, H. A., Kataria, T., et al. 2018, Phase Curves of WASP-33b and HD 149026b and a New Correlation between Phase Curve Offset and Irradiation Temperature, , 155, 83, doi: 10.3847/1538-3881/aaa458

6 Appendix

Github Repository: Contains Plots and Scripts of the Simulations

Simulation Videos (Youtube)

6.1 Paper for Publication

Draft version December 5, 2025
Typeset using LaTeX **twocolumn** style in AASTeX7.0.1

Preliminary benchmarks of the Dedalus spectral solver for atmospheric dynamics of exoplanets

Rick Bonhof,[1] Quentin Changeat,[1] and James Y-K. Cho[2]

[1] *Kapteyn Institute, University of Groningen, 9747 AD Groningen, The Netherlands*
[2] *Martin A. Fisher School of Physics, Brandeis University, Waltham MA 02453, USA*

ABSTRACT

Studying the short-timescale variability of giant planets demands high-resolution, low-dissipation atmospheric simulations. Yet for exoplanets, very few dynamical cores remain numerically stable under these stringent physical conditions. In this research note, we evaluate the performance of the Dedalus 3 spectral solver in idealized planetary-atmosphere setups. We accurately reproduce the canonical Galewsky mid-latitude jet instability and qualitatively capture Jupiter-like flow structures with good conservation of the global circulation pattern. Although our tests focus on relatively gentle dynamical regimes, the results indicate that Dedalus 3 is a promising tool for rigorous investigations of planetary-atmosphere dynamics.

Keywords: Exoplanets modeling — Atmospheric dynamics — Shallow Water — Jupiter

1. INTRODUCTION

With the influx of new observations from the James Webb Space Telescope (JWST) and the anticipated contributions of upcoming facilities like the Extremely Large Telescope (ELT) and the Ariel mission, exoplanet research is increasingly focusing on atmospheric characterization. Observations now constrain atmospheric properties with unprecedented precision, enabling detailed investigations of the climate and weather patterns on distant worlds (T. J. Bell et al. 2024; Q. Changeat et al. 2024; A. M. McCarthy et al. 2025). These observations, probing spatial and temporal variabilities, are often interpreted using atmospheric dynamics models. However, recent studies (I. Polichtchouk et al. 2014; J. W. Skinner & J. Y.-K. Cho 2021) have shown that current exoplanet global circulation models (GCMs)—often originally designed for the Earth climate—use dynamical cores at too low-resolution and with high numerical dissipation that prevent them to capture the key dynamical phenomena on extreme exoplanets reliably. Specifically, warm to hot-Jupiters have vigorous dynamics with important small-scale eddies and wave–mean flow interactions that are driving strong jets, instabilities, and shock-like features. This vigorous dynamics can only be captured using high-resolution (T170+) dynamics solvers that conserve energy and potential vorticity, and do not numerically smear out small and mesoscale dynamics.

Corresponding author: Quentin Changeat
Email: q.changeat@rug.nl

In this context, the recently developed Dedalus3 code (K. J. Burns et al. 2020) appears as a promising dynamical core for planetary atmospheric studies. Dedalus 3 solves partial differential equations using high-order pseudo-spectral sparse tau methods. The framework is designed for parallel execution on modern HPC architectures and has been applied extensively to astrophysical problems. We aim to evaluate the performance of Dedalus 3 on planetary atmosphere application. In this quick research note, we restrict ourselves to simple problems exhibiting only *gentle* dynamics. As such, we model the planetary atmospheres using an hydrostatically balanced shallow layer of frictionless gas, moving only under the influence of gravitational and Coriolis accelerations. The motion of such atmospheric layer is given by the Equivalent Barotropic Shallow-Water Equations (EB-SWE: M. L. Salby 1989):

$$\frac{\partial \mathbf{u}}{\partial t} - \nu\nabla^2\mathbf{u} + g\,\nabla h + 2\Omega\,\mathbf{k}\times\mathbf{u} = -\,(\mathbf{u}\cdot\nabla)\mathbf{u}, \qquad (1)$$

$$\frac{\partial h}{\partial t} - \nu\nabla^2 h + \bar{H}\,\nabla\cdot\,\mathbf{u} = -\,\nabla\cdot\,(h\,\mathbf{u}), \qquad (2)$$

where $\mathbf{u}$ is the horizontal velocity and h is the deviation from the mean layer thickness $\bar{H}$. The parameters of the simulation are ν: the *numerical* diffusion (i.e., this is not a physical parameter); g: the gravitational acceleration; Ω: the planetary rotation rate; and $\mathbf{k}$: the vertical unit vector. Note that terms on the left-hand side of these equations are treated in spectral space for implicit eval-

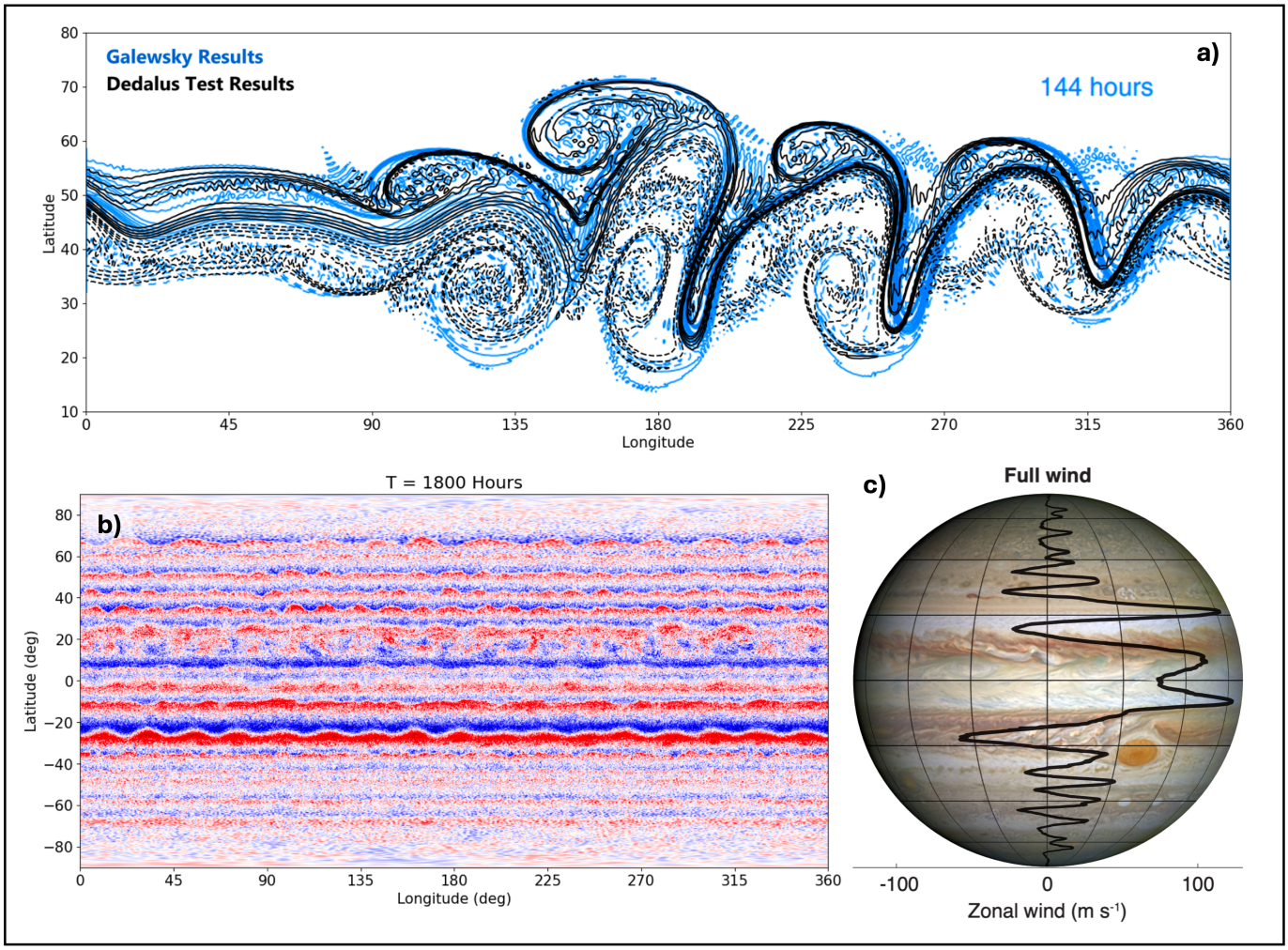

Figure 1. Summary of the simulations performed for this article. **Top panel a)** Comparison of the vorticity field obtained after 144 h of simulations in the J. Galewsky et al. (2004) mid-latitude jet test case (inviscid scenario at T341 with contours at intervals of $2 \times 10^{-5}\,\mathrm{s}^{-1}$). Our results well align with the FMS-SWM results in their Figure 4. **Bottom Left panel b):** Vorticity map of Jupiter, obtained from a dynamics simulation with Dedalus at T341 that solves for the freely evolving SWE, with the observed Jupiter zonal wind profile as initial conditions and small ($\mathbf{u}$, h) stirring. **Bottom Left panel c):** Real image of Jupiter with the observed zonal wind profile from Y. Kaspi et al. (2018) superimposed. The figure illustrates the relevance of the Dedalus atmospheric dynamic core for simulations of planetary atmospheres.

uation, while the right-hand side terms are treated in physical space for explicit evaluation.

We solve the EB-SWE with Dedalus, exploring a) the well-known Galewsky test case (J. Galewsky et al. 2004); and b) a freely evolving Jupiter simulation initialized with the JWST observed zonal wind profile. More complex forced cases exhibiting stronger dynamics will be studied in the future.

2. GALEWSKY TEST OF DEDALUS 3

We perform a numerical check of the Dedalus 3 solver on a well-known initial value problem from J. Galewsky et al. (2004). This well-defined test aims at evaluating the numerical convergence of atmospheric dynamics cores for planetary atmosphere applications. It models an analytically specified, barotropically unstable mid-latitude jet. Initially balanced, a simple perturbation is introduced to initiate the jet instability. We refer to J. Galewsky et al. (2004) for further details on the test. We perform the simulations at resolutions T42, T85, T170, and T341 using timestep $\Delta t = 30\,\mathrm{s}$. We ran inviscid (i.e., $\nu = 0$) and explicit diffusion ($\nu = 10^5$ $\mathrm{m}^2\mathrm{s}^{-1}$) cases as in the original paper. Looking at the evolution of the instability after $t = 144\,\mathrm{h}$, we find that our results mostly match those of J. Galewsky et al. (2004). For instance, Panel a) of Figure 1 shows the vorticity field after 144 h for the inviscid T341 case. We also record the maximum/minimum vorticity (ζ_{max} / ζ_{min}) at t = 144 h and the maximimum/minimum height values (h_{max} / h_{min}) at t = 4 h, which can be

compared to Table 1 of J. Galewsky et al. (2004) for the explicit diffusion run [3]. We find a close match: $\zeta_{min} = 0.05 \cdot 10^{-5}$ (0.68 %), $\zeta_{max} = 0.13 \cdot 10^{-5}$ (1.4 %), $h_{min} = 5$ (0.06 %), $h_{max} = 16$ (0.16 %) to the values stated in J. Galewsky et al. (2004). Some differences exist: see e.g., the slightly more chaotic eddies of FMS-SWM in Figure 1, and < 1 % differences in height/velocity reference values. These differences most likely come from the treatment of the numerics but their exact origins could not be established during our exploration. Possible avenues are: differences in time-step algorithms (leapfrog vs Runge-Kutta 4), floating point errors (we utilize float64, but the original paper does not specify this), internal numerical dissipation in the codes. Our results, however, suggest that a close match could be obtained and that Dedalus can be used for the modeling of planetary atmospheres.

3. A SIMPLE JUPITER EXAMPLE

We provide an example use case for Dedalus by solving the EB-SWE for a Jupiter's atmosphere simulation. We model a layer of thickness $\bar{H} = 2.7 \times 10^4$ m in the inviscid scenario and at resolution T341 (equivalent to 1024×512 lon-lat grid). The simulation is initialized with the observed JWST wind profile from Jupiter R. Hueso et al. (2023) and a corresponding globally balanced height anomaly field (h). A small additional random stirring of magnitude 100 m is added to h. The solution is evolved over 2400 h using timesteps of $\Delta t = 30$ s. Despite the lack of explicit diffusion, the simulation is extremely stable: the bands of Jupiter maintain a coherent structure for the entire duration of the simulation. We show in the panel b) of Figure 1 the vorticity field obtained after 75 days. Jupiter's bands are accurately maintained, closely tracing the original zonal flow field shown in panel c) .

4. CONCLUSIONS

We test the Dedalus 3's pseudo-spectral solver to evolve the EB-SWE, following the well-known Galewsky test with and without explicit diffusion. We find maximum/minimum vorticity and height field values close to those of the reference article. Overall, this close match provides confidence that Dedalus 3 can be used for numerically solving planetary atmospheres fluid dynamics problems. However, small differences exist, with progressively differing evolution of eddies compared to FMS-SWM. We could not establish the exact reason for these differences, and suggest that further investigations should be performed to further reinforce our conclusions. We also provide an application example of Dedalus, solving the inviscid EB-SWE with Jupiter's initial conditions. We utilize the zonal wind profile measured through JWST from R. Hueso et al. (2023), producing a realistic simulation of Jupiter's flow. The solution remains stable over our entire calculation (100 days), retaining the large characteristics of the original wind profile.

DATA AVAILABILITY

This article makes use of the observations of Jupiter interpreted in R. Hueso et al. (2023). For the different data products, we refer to that paper.

Software: Dedalus3[4](K. J. Burns et al. 2020).

REFERENCES

Bell, T. J., Crouzet, N., Cubillos, P. E., et al. 2024, Nature Astronomy, 8, 879, doi: 10.1038/s41550-024-02230-x

Burns, K. J., Vasil, G. M., Oishi, J. S., Lecoanet, D., & Brown, B. P. 2020, Physical Review Research, 2, 023068, doi: 10.1103/PhysRevResearch.2.023068

Changeat, Q., Skinner, J. W., Cho, J. Y.-K., et al. 2024, ApJS, 270, 34, doi: 10.3847/1538-4365/ad1191

Galewsky, J., Scott, R. K., & Polvani, L. M. 2004, Tellus A, 56, 429,
doi: https://doi.org/10.1111/j.1600-0870.2004.00071.x

Hueso, R., Sánchez-Lavega, A., Fouchet, T., et al. 2023, Nature Astronomy, 7, 1454,
doi: 10.1038/s41550-023-02099-2

Kaspi, Y., Galanti, E., Hubbard, W. B., et al. 2018, Nature, 555, 223, doi: 10.1038/nature25793

McCarthy, A. M., Vos, J. M., Muirhead, P. S., et al. 2025, ApJL, 981, L22, doi: 10.3847/2041-8213/ad9eaf

Polichtchouk, I., Cho, J. Y.-K., Watkins, C., et al. 2014, Icarus, 229, 355, doi: 10.1016/j.icarus.2013.11.027

Salby, M. L. 1989, Tellus Series A, 41, 48,
doi: 10.3402/tellusa.v41i1.11820

Skinner, J. W., & Cho, J. Y.-K. 2021, MNRAS, 504, 5172,
doi: 10.1093/mnras/stab971

[3] note that Table 1 of Galewsky provides the values for ν=1.0 × 10^5 m^2/s instead of ν=0

[4] https://dedalus-project.org/