

Towards an agnostic algorithm for sampling empirical structure models

The case of Uranus and Neptune

Stefano Wirth^{1,2,*}, Luca Morf^{2,**}, and Ravit Helled²

¹ Department of Mathematics, Swiss Federal Institute of Technology Zurich (ETHZ), Rämistrasse 101, 8092 Zürich, Switzerland

² Department of Astrophysics, University of Zurich, Winterthurerstrasse 190, 8057 Zürich, Switzerland

Received 8 October 2025 / Accepted 18 December 2025

ABSTRACT

We present an algorithm to efficiently sample the full space of planetary interior density profiles. Our approach uses as few assumptions as possible to pursue an agnostic algorithm. The algorithm avoids the common Markov chain Monte Carlo method and instead uses an optimisation-based gradient-descent approach designed for computational efficiency. In this work, we use Uranus and Neptune as test cases and obtain empirical models that provide density and pressure profiles consistent with the observed physical properties (total mass, radius, and gravitational moments). We compared our findings to other work and find that while other studies are generally in line with our findings, they do not cover the entire space of solutions faithfully. Furthermore, we present guidance for modellers that construct Uranus or Neptune interior models with a fixed number of layers. We provide a statistical relation between the steepness classifying a density discontinuity and the resulting number of discontinuities to be expected. For example, if one classifies a discontinuity as a density gradient larger than 0.02 kg m^{-4} , then most solutions should have at most one such discontinuity. Finally, we find that discontinuities, if present, are concentrated around a planetary normalised radius of 0.65 for Uranus and 0.7 for Neptune. Our algorithm to efficiently and faithfully investigate the full space of possible interior density profiles can be used to study all planetary objects with gravitational field data.

Key words. planets and satellites: gaseous planets – planets and satellites: interiors – planets and satellites: individual: Uranus – planets and satellites: individual: Neptune

1. Introduction

Uranus and Neptune have always been key to understanding the formation and evolution history of the Solar System. As the investigation of our own planetary system continues to expand, the study of exoplanets is gaining more attention. Although the characterisation of exoplanets remains limited, it is now clear that in terms of mass and size, many exoplanets in the galaxy resemble Uranus and Neptune (Zhu & Dong 2021). By revealing the internal structures of Uranus and Neptune, we can improve our knowledge of these two planetary objects in our own neighbourhood and the many observed intermediate-mass exoplanets orbiting distant stars.

To study the internal structure of a planet, we need to know its density profile. In the absence of direct measurements, the profile can be inferred from other observations. Density profiles can be inferred by varying all values constituting a profile until the result matches the available observational data (for example total mass, radius, and the gravitational field). The number of such observational parameters is typically small; for Uranus and Neptune, there are fewer than ten. In contrast, density profiles typically consist of orders of magnitude more data points. Inferring density profiles is therefore severely degenerate, as many density profiles can fit the same observational data.

To address this problem, one can simplify the approach used to infer density profiles by relying on a multitude of assumptions.

However, caution must be taken. Making unsound simplifying assumptions could exclude a large portion (or all) of the valid density profiles. Therefore, obtaining a complete overview of all possible solutions is paramount before simplifications are even considered.

To gain such an overview, we must sample profiles with an approach that is as unbiased as possible. This kind of unbiased investigation is called empirical structure modelling. Empirical structure modelling has been used before to investigate the internal structure of Uranus and Neptune. The approach harkens back to earlier foundational works such as Marley et al. (1995) and Podolak et al. (2000). More recently, Neuenschwander & Helled (2022) presented empirical structure models parametrised by polytropes that directly relate density to interior pressure. In comparison, Movshovitz & Fortney (2022) used models that employed a combination of polynomials and sigmoid functions.

The current numerical method of choice for empirical structure modelling is often the Markov chain Monte Carlo (MCMC) method (for example Marley et al. 1995; Movshovitz & Fortney 2022). The MCMC approach starts by generating initial density profiles and then randomly alters their parameters until they fit the observational data. These poses challenges due to the aforementioned large number of data points that make up a density profile. Therefore, one must either tolerate the slow convergence of the MCMC method when handling countless parameters or reduce the dimensionality of the parameter space to accelerate the search. Previous studies have taken the latter route by parametrising density profiles with polytropes, polynomials, or

* These authors contributed equally to this work.

** Corresponding author: luca.morf@uzh.ch

sigmoid functions. This reduces the number of parameters to an order of magnitude comparable to the number of observational parameters, but it can introduce biases, which is precisely what we aim to avoid here.

We replace the common MCMC method with a new optimisation-based approach that uses an algorithm to efficiently investigate the full space of possible interior density profiles that fit the known observational data of a planet. The assumptions made for the profiles are reduced to an absolute minimum: We only require that there are no density plateaus or inversions and that the maximal density does not exceed the density of any feasible material. Otherwise, (with a resolution of $N = 1024$) the N density values are all allowed to change freely. Because we treat all N density values as free parameters, the method avoids biases introduced by the parametrisations of previous MCMC studies. However, this alone does not ensure that the results are free from bias. We must further ensure that the generation of starting profiles is unbiased, and – because we no longer use an MCMC algorithm – we must additionally ensure that the way we find the density profiles is unbiased. For the first point, if the generation of starting profiles favours certain types of density profiles, for instance, by initially under-representing profiles with sharp density discontinuities, an adjustment that increases the sampling of such profiles is required. For the second point, we need to verify that our optimisation approach does not overlook harder-to-find local solutions in favour of dominant global ones that are easily spotted.

Our paper is structured as follows. Section 2 provides an overview of the algorithm. In Section 3, we present and interpret the results. In Section 4, we discuss the results, highlight caveats, and provide an outlook for future work. Section 5 presents a summary of the key findings of this study. We note that our methodology is based on the theory of figures (ToF, Zharkov & Trubitsyn 1978). For the purposes of notation, we follow the elaboration presented in Appendix A of Morf et al. (2024). We employed the open-source code PyToF¹ (Morf 2025) to implement the ToF numerically.

2. Methods

2.1. Algorithm

Planets are flattened due to their rotation. Surfaces of both constant density and potential are not spherical shells defined by a single radius, r , but are rather spheroidal in shape. Spheroidal surfaces are described by a function, $r_l(\vartheta)$, that depends on the polar angle, ϑ . The variable l uniquely identifies a spheroid and represents its volumetric mean radius:

$$\frac{4\pi}{3}l^3 = 2\pi \int_{-1}^1 d \cos \vartheta \int_0^{r_l(\vartheta)} r'^2 dr'. \quad (1)$$

We also refer to l as the level surface of a spheroid. A planet with such a spheroidal structure has a multipolar (non-spherical) external gravity field

$$V_{\text{ext}}(r, \vartheta) = \frac{Gm}{r} \left(1 - \sum_{n=1}^{\infty} \left(\frac{R_{\text{ref}}}{r} \right)^{2n} J_{2n} P_{2n}(\cos \vartheta) \right), \quad (2)$$

where R_{ref} is a normalisation constant, m denotes the planetary mass, and P_{2n} are the Legendre polynomials. The J_{2n} are called gravitational moments and can be calculated with the ToF.

¹ See <https://doi.org/10.5281/zenodo.17592269>

For the numerical implementation, density profiles consist of N density values, ρ_k , at evenly spaced level surfaces, l_k . We restricted the maximum density to some threshold, ρ_{max} , and enforced a minimum density increase per step, Δ_{min} . The density values are derived from $N - 1$ parameters, p_i , as

$$\rho_k = \sum_{i=0}^{k-1} \frac{e^{p_i}}{w_i} \alpha. \quad (3)$$

The weights, w_i , were added to normalise the parameter values and are defined as

$$w_i = \sum_{j=i+1}^{N-1} l_j^2. \quad (4)$$

The scaling factor α was added to ensure the density profile mass, m_{calc} , agrees with the planetary mass, m . It is defined as

$$\alpha = \frac{m}{m_{\text{calc}}}, \quad (5)$$

where the calculated mass, m_{calc} , was obtained by numerically integrating the density values over the level surfaces via

$$m_{\text{calc}} = 4\pi \int_{l_{\text{min}}}^{l_{\text{max}}} \sum_{i=0}^{k-1} \frac{e^{p_i}}{w_i} l^2 dl. \quad (6)$$

Listing 1 gives a high-level overview of the optimisation algorithm.

Listing 1: The optimisation algorithm.

```

input: int epochs
output: tuple results
begin
  params ← create_start_params()
  for epoch in epochs do
    check convergence
    for steps in epochsize do
      params ← project(opt_step(params))
  return results
end

```

Given the number of epochs to run the algorithm for, we first created some suitable starting parameters. Then, for each epoch, we checked whether the optimisation converged to a stable solution. If not, we performed epochsize for many optimisation steps. For each optimisation step, we calculated the gradient and moved along it according to the Adam gradient descent scheme (Kingma & Ba 2017). Then, we projected the new parameters to enforce our minimum density increase. Once convergence was detected, the final result and other data, was returned. Further details on the algorithm can be found in the Appendix A. Explanations regarding the requirements of an unbiased generation method and optimisation method are given in Appendix C. The method's limitations are discussed in Section 4.

2.2. Simulation setup

The simulation was performed on an AMD EPYC 7702 Hypervisor CPU with 96 cores over 120 hours. We dedicated 64 cores to Uranus and 32 towards Neptune. We used twice as many cores for Uranus in order to run correlated ($\text{cov}(J_2, J_4) = 0.9861\sigma_{J_2}\sigma_{J_4}$; see Table 1) and uncorrelated ($\text{cov}(J_2, J_4) = 0$)

Table 1. Physical values for Uranus and Neptune.

	Uranus		Neptune	
Mass	$8.68099 \cdot 10^{25}$ kg	Jacobson & Park (2025)	$1.02409 \cdot 10^{26}$ kg	Jacobson (2009)
Equatorial radius	25 559 km	Lindal et al. (1987)	24 766 km	Lindal (1992)
Reference radius	25 559 km	French et al. (2024)	25 225 km	Wang et al. (2023)
J_2	$(3509.291 \pm 0.412)10^{-6}$	French et al. (2024)	$(3401.655 \pm 3.994)10^{-6}$	Wang et al. (2023)
J_4	$(-35.522 \pm 0.466)10^{-6}$	French et al. (2024)	$(-33.294 \pm 10)10^{-6}$	Wang et al. (2023)
cov(J_2, J_4)	$0.9861\sigma_{J_2}\sigma_{J_4}$	French et al. (2024)	–	–
Rotational period	62 064 s	Desch et al. (1986)	57 479 s	Karkoschka (2011)

simulations in parallel. All physical values for Uranus and Neptune are given in Table 1. The number of layers, N , was set to 1024, and the epoch size was 25 steps. We chose a maximum of 120 epochs (3000 steps). The scalar factor, c , introduced in Equation (A.2) was set to 100. This factor ensures that the optimisation cost function does not consider extreme values for α (far away from $\alpha = 1$) while optimising for gravitational field data. As a convergence criterion within the ToF algorithm PyToF, we employed $|J_{2n}^{i+1} - J_{2n}^i| \leq 10^{-10} J_{2n}^i$ to check if convergence was achieved after the i -th iteration. The maximum acceptable density ρ_{max} was set to $20\,000 \text{ kg m}^{-3}$. The minimum acceptable density increase per unit length, Δ_{min} , was set to $1 \Delta\ell^{-1} \text{ kg m}^{-3}$, where $\Delta\ell = R/N$. Explicitly, this resulted in a value of $\Delta\ell = 24\,960 \text{ m}$ for Uranus and $\Delta\ell = 24\,186 \text{ m}$ for Neptune (rounded to the nearest meter). This yielded $\Delta_{min} = 4.01 \cdot 10^{-5} \text{ kg m}^{-4}$ for Uranus and $\Delta_{min} = 4.13 \cdot 10^{-5} \text{ kg m}^{-4}$ for Neptune (rounded to two significant figures).

Overall, 694 465 (791 901) runs were completed for the correlated (uncorrelated) Uranus data, and 731 407 runs were completed for Neptune. For Uranus, 90.39% (90.48%) of the runs respected ρ_{max} , and 18.77% (18.56%) achieved gravitational moments, J_{2n} , within the multivariate three-standard-deviation threshold, which is 20.77% (20.56%) of those that respected ρ_{max} . For Neptune, the numbers (in the same order) are 89.22% and 88.83%, which is 99.57% of those that respected ρ_{max} . Our optimisation-based approach and the developed algorithm are highly efficient: The average time to complete a run was approximately 20 seconds.

We note that throughout all subsequent results, the parameter space of the possible density profiles for Neptune is larger in comparison to Uranus. This is because the measurements we have for the gravitational moments of Uranus are about one order of magnitude more precise than those for Neptune (Table 1). Consequently, Uranian runs had a $\sim 19\%$ success rate, compared to $\sim 89\%$ of the Neptunian runs.

3. Results

Below we present the inferred density profiles and their characteristics. For simplicity, we refer to the normalised volumetric mean radius of a spheroid as ‘the normalised radius r/R ’.

3.1. Density profiles

First, we present an aggregate view of the inferred profiles from the successful runs. A run is considered to be successful if its resulting density profile did not exceed ρ_{max} , and it achieved the observed gravitational moments within the multivariate three standard deviation thresholds. All runs within this threshold

were then weighted according to their multivariate Gaussian likelihood to recover a distribution compatible with the data of Table 1. Figure 1 shows the gravitational moment distributions, and Figures 2 and 3, respectively show the density profiles as a distribution in logarithmic scale and contour. Figure 4 shows their central densities, while Figure 5 shows the size of the confidence intervals (68% and 95%).

The gravitational moment distributions shown in Figure 1 demonstrate that our results cover the whole J_2 – J_4 space faithfully, especially for Uranus. For Neptune, the uncertainties are so large in comparison that our algorithm struggled to find a significant amount of solutions outside the multivariate threshold of one standard deviation, especially for J_4 . Since French et al. (2024) also provide a covariance estimate in their results, we additionally show a correlated distribution (orange) in Figure 1a for Uranus. While the change between the uncorrelated and correlated distribution looks significant in this panel, the impact on the subsequent results is negligible. All subsequent plots hence exclusively display results for the uncorrelated distributions to enable a fair comparison between Uranus and Neptune. The correlated Uranus results produce visually indistinguishable plots.

Figures 2 and 3 show the density solution space for Uranus and Neptune, respectively. We find that successful density profiles are least constrained in the planetary deep interior. The outermost regions are highly constrained, with many distributions being affected by the minimum density increase. Around $\sim 85\%$ of the planetary radius ($r/R = 0.85$), a more diverse regime takes over for both planets. Still, the general shape continues to show a relatively tight constriction of the possible space up to a normalised radius of $\sim 0.4R$. This is most apparent in Figure 5. Notably, there is a bump at around $r/R = 0.65$ for Uranus (Figure 2) and $r/R = 0.7$ for Neptune (Figure 3). In the innermost region, a large increase in the space of possible distributions develops. The logarithmic scale is somewhat misleading, as can be seen by consulting the contour plots Figures 2b and 3b. Though many distributions fall outside the 95% region (especially above it), their relative frequency is small, on the order of 10^{-4} or less. The median density (as a function of radius) for Uranus and Neptune shows gradual increases in density, consistent with composition gradients. We note that the median density shown in Figures 2b and 3b is neither an actual density profile from the solution ensemble nor a maximum-likelihood indicator.

Next, we compare our results to previous work, focusing first on models in the literature that are based on equations of state and hence infer composition and temperature alongside density (and pressure). We find that the central densities of profiles from Helled et al. (2011) and Vazan & Helled (2020) are concentrated around the peak of the distribution, not its median. Profiles from Nettelmann et al. (2013) have core densities higher than the 84th

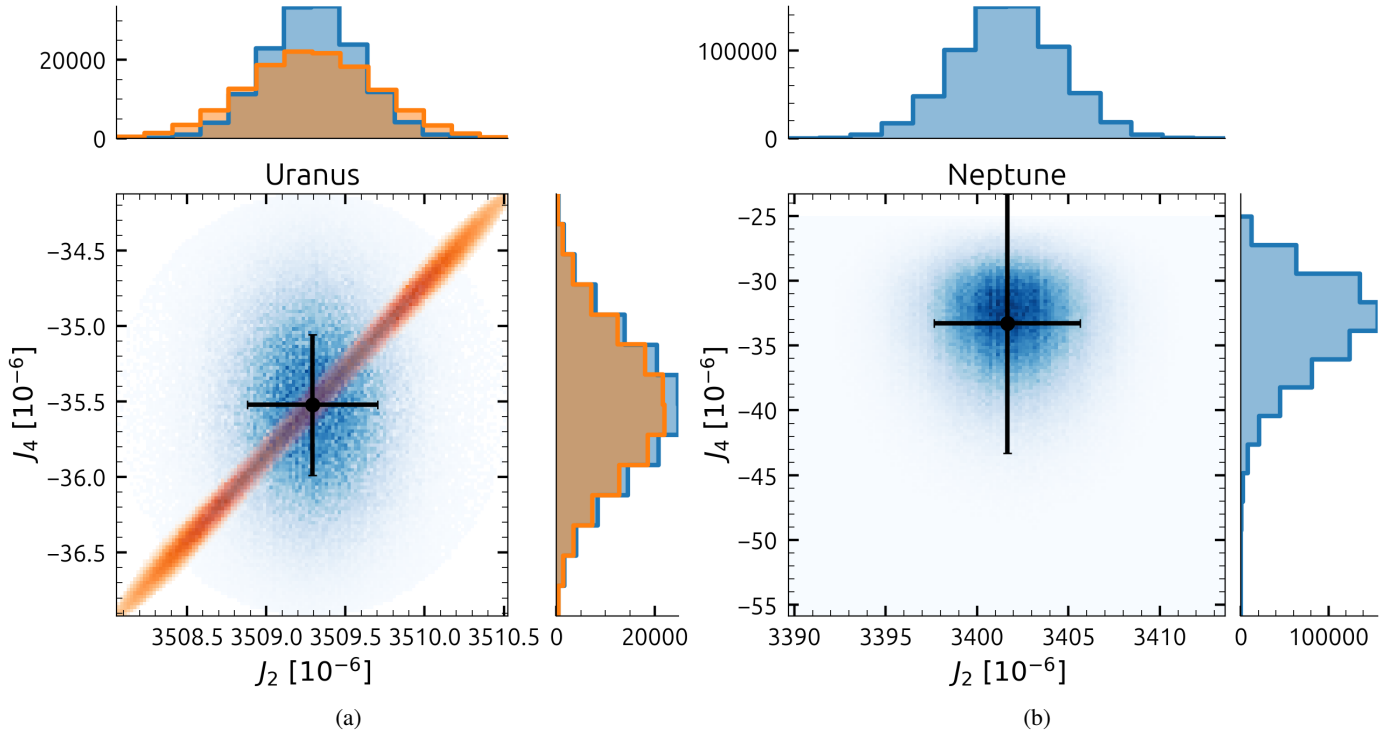


Fig. 1. Gravitational moment distribution for successful Uranus (a) and Neptune (b) profiles. Both panels include a 2D histogram where bins more frequently occupied by solutions are shown with a relatively darker colour. Additionally, 1D histograms display the weighted number of solutions along each axis. The weights were assigned to solutions based on a multivariate Gaussian likelihood, with means, standard deviations, and covariance taken from Table 1. In the left panel a, an orange distribution is shown in addition to the blue distribution. Both blue distributions in (a) and (b) assume $\text{cov}(J_2, J_4) = 0$, whereas the orange distribution for Uranus takes $\text{cov}(J_2, J_4) = 0.9861 \sigma_{J_2} \sigma_{J_4}$.

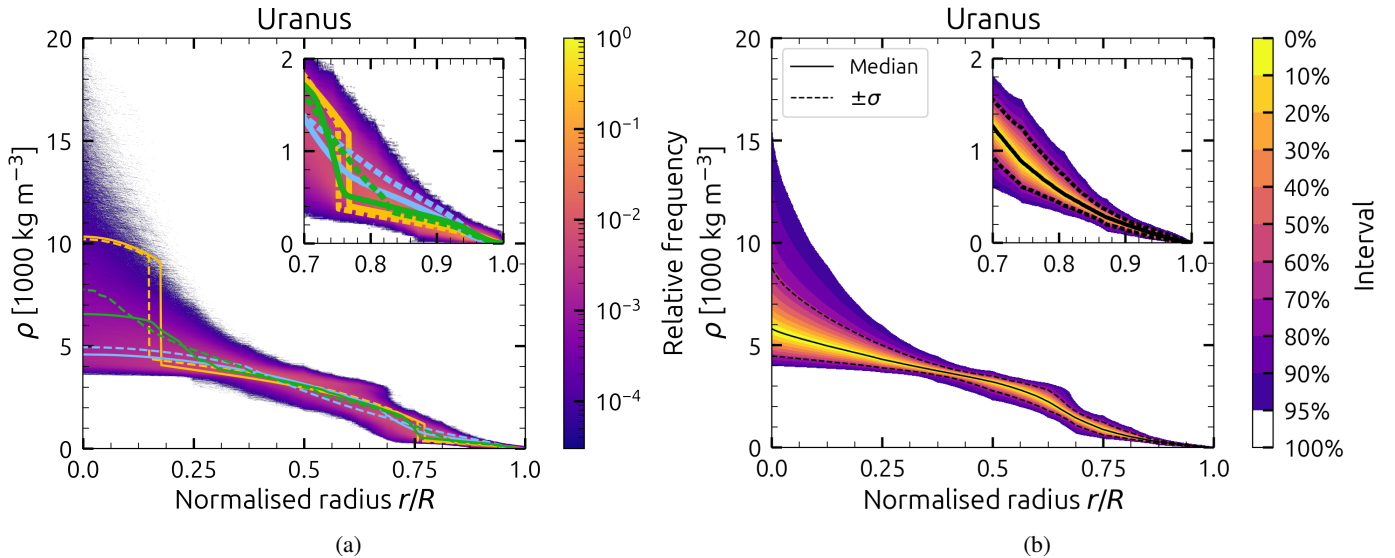


Fig. 2. Solution space distribution (a) and contour of the distribution (b) for Uranus. The left panel a shows the weighted distribution of all successful uncorrelated Uranus density profiles. The inset shows a zoomed-in view of the outermost region, from $r/R = 1$ to $r/R = 0.7$. Note that the zoom is not aspect-preserving. The colour scale shows the relative frequency over all distributions of a certain density value at a given radius. Coloured lines show density profiles from previous studies for comparison. Solid and dashed lines correspond to models U1 and U2 from [Nettelmann et al. \(2013\)](#) (orange); U1 and U3 from [Morf & Helled \(2025\)](#) (green); and V2 and V3 from [Vazan & Helled \(2020\)](#) (blue), respectively. The colour scale in the right panel b shows the percentile intervals in 10% increments, except the last interval, where a 95% interval was added. The black line shows the median. The dashed lines show the 16th (lower) and 84th (upper) percentiles.

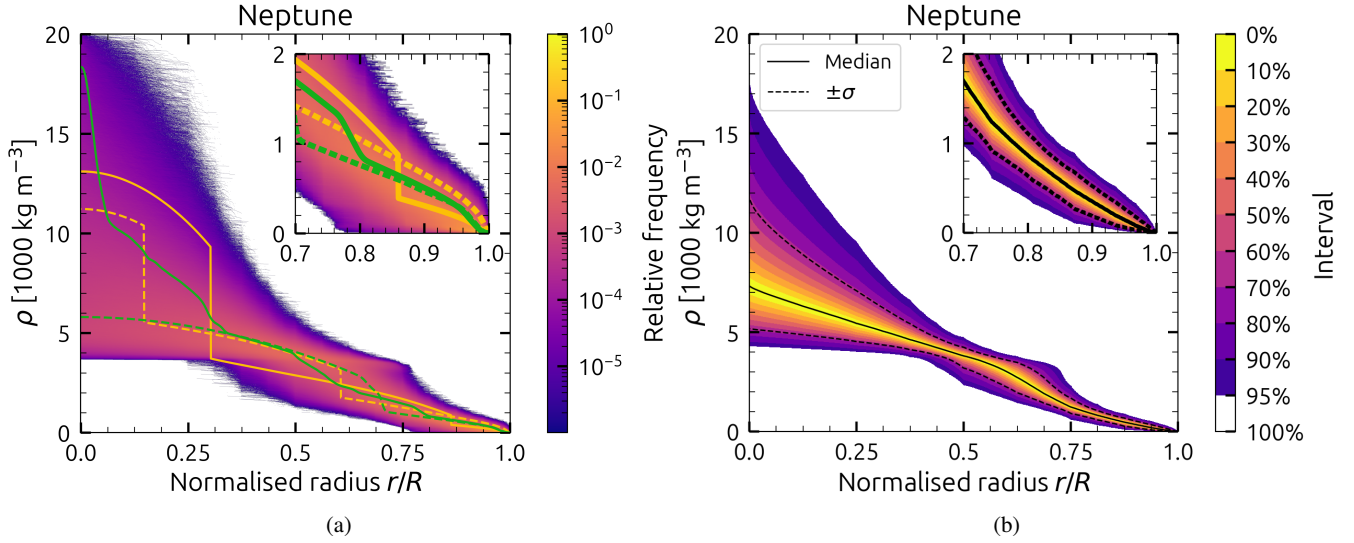


Fig. 3. Same as Figure 2 but for Neptune. The solid and dashed line in the left panel a correspond to the models N1 and N2b from [Nettelmann et al. \(2013\)](#) (orange) and to models N1 and N3 from [Morf & Helled \(2025\)](#) (green), respectively.

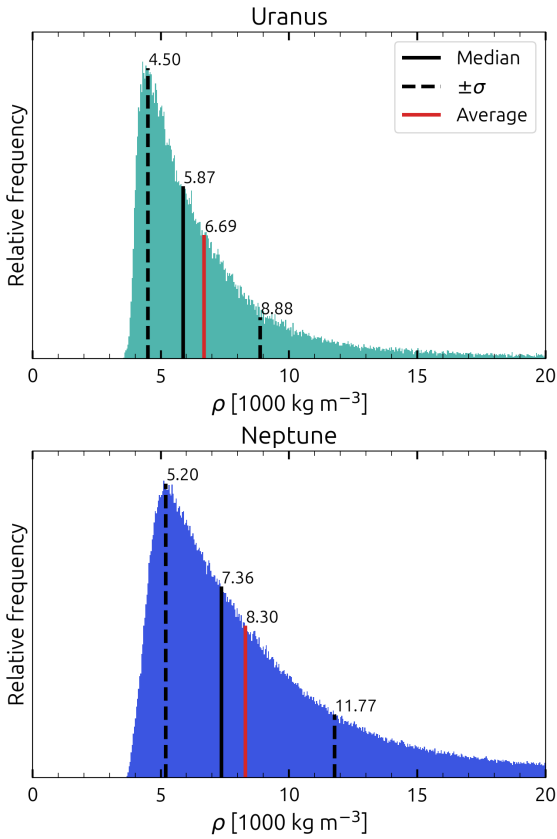


Fig. 4. Central densities. The black lines are as before in Figures 2b and 3b. The red line denotes the average. This figure is equivalent to a side-on view of the density profile distributions (Figures 2a and 3a) at radius 0, except in linear scale. Note that the distribution is cut off at $\rho_{max} = 20\,000\text{ kg m}^{-3}$.

percentile. The more recent work by [Militzer \(2024\)](#) is similar, predicting core densities that are close to those inferred by [Nettelmann et al. \(2013\)](#). The Uranus models U1 and U3 with extensive composition gradients from [Morf & Helled \(2025\)](#) are close to the median result. Their convective models U2 and U4

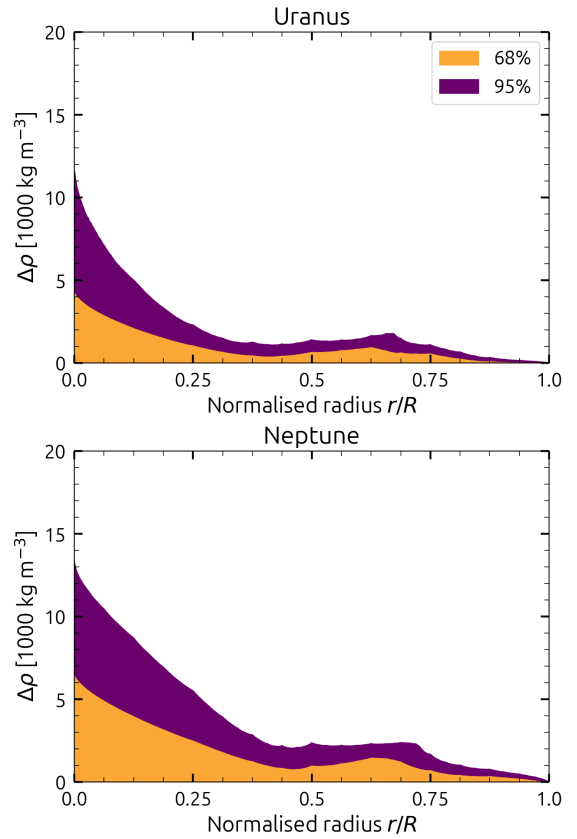


Fig. 5. Confidence intervals. Size of the 68% and 95% confidence intervals in Figures 2b and 3b for the successful density profile distributions. The plot is cumulative, where the size of the 95% confidence interval is the purple and the orange area combined.

are more similar to [Vazan & Helled \(2020\)](#) and are situated at the peak of the distribution. We note that although their U2 and U4 profiles possess the same comparatively low central density of $\sim 4.5\text{ g/cm}^3$, their inferred total rock-to-water mass fractions vary significantly (0.64 vs. 3.92). We can therefore conclude that the density-pressure relation alone is insufficient for determining

the planetary bulk composition. Interestingly, the N1 profile for Neptune by Morf & Helled (2025) falls outside the 95th percentile with a significantly higher central density. This occurs because their approach can also probe less standard solutions. On the other hand, their N3 solution is close to the median result.

Studies that, as in this work, analyse large ensembles of interior profiles enable a more direct and systematic comparison. For example, Figure 4 in Helled & Fortney (2020), Figure 10 in Movshovitz & Fortney (2022), and Figure 2 in Neuenschwander & Helled (2022) are directly comparable to Figures 2b and 3b. Helled & Fortney (2020) employed eighth-order polynomials, and Movshovitz & Fortney (2022) additionally incorporated sigmoid functions. They obtained profiles with generally higher densities in the deep interior for both Uranus and Neptune, especially in the case of Helled & Fortney (2020), below $0.3R$. Their median profiles also flatten off below $0.1R$, whereas our results show no such trend. Our medians maintain a consistent slope below $0.3R$ and even curve up slightly. Moreover, the distribution of Helled & Fortney (2020) is symmetric around the median, whereas our results show a wider spread for higher-density solutions. The inclusion of sigmoid terms in Movshovitz & Fortney (2022) produced extended flat regions that do not appear in our models due to the imposed minimum density increment per layer, Δ_{\min} . In the work of Neuenschwander & Helled (2022), interior structures were constructed using three consecutive polytropes. This imprinted distinct features on the confidence intervals, including sharp discontinuities and narrow choke points, which are absent in our results. Similar to our results, their approach yielded a distribution with a wider spread around the median for higher central densities.

The wider spread for higher central densities is further illustrated in Figure 4. We find that the central densities are very unequally distributed, rising to a peak rapidly and gradually descending as density increases for both planets. For Uranus, the peak is very tight at $\rho \sim 4500 \text{ kg m}^{-3}$, but the median is still somewhat higher. The solutions for Neptune are distributed more broadly. Hence, a significant portion of solutions exceeded the imposed maximum density ρ_{\max} and were excluded. The peak of the distribution is at $\rho \sim 5200 \text{ kg m}^{-3}$.

Figure 5 reflects the span of our results. The uncertainty of the gravitational moment measurements of Neptune is a factor of 10 larger than that of Uranus (Table 1). One could therefore expect that our results are considerably less constrained for Neptune as well. However, Figure 5 shows that this is not the case. Our density profile results are only slightly more widely spread for Neptune than they are for Uranus. Tighter bounds in the J_2 – J_4 space do not linearly translate to tighter bounds in the density solution space. Even if the J_2 and J_4 values were perfectly known, the density profile solution space would remain degenerate. Only a measurement of all the gravitational moments $\{J_{2n} \mid n \in \mathbb{N}\}$ would provide a unique solution for a north-south and azimuthally symmetric planet.

Finally, we note that in the contour (Figures 2b and 3b) and confidence interval (Figure 5) plots, small ‘spikes’ are visible concentrated around inverses of multiples of two, most notably at $r/R = 0.5$. These are artefacts of the initial density profile generation method and are addressed in Section 4 as a caveat.

3.2. Derived quantities

In this Section, we present quantities derived from the successful density profiles via the ToF. Figure 6 shows the distribution of pressure profiles with a logarithmic frequency scale. Figure 7

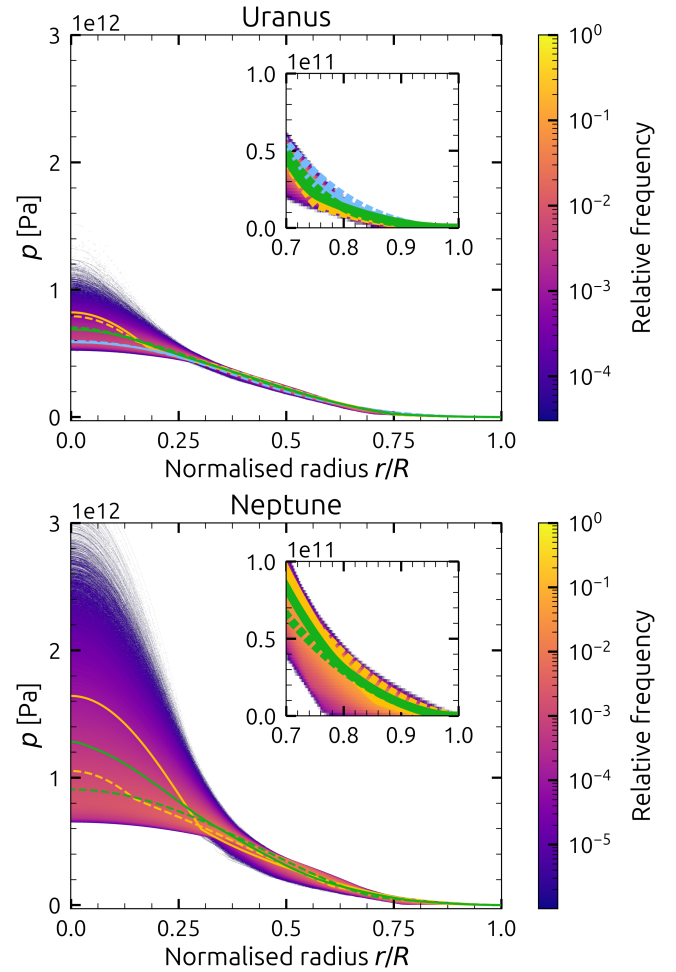


Fig. 6. Pressure profiles. Weighted distribution of pressure profiles derived from all successful density profiles. The inset shows a zoomed-in view of the outermost region, from $r/R = 1$ to $r/R = 0.7$. Note that the zoom is not aspect-preserving. The colour scale shows the relative frequency over all distributions of a certain pressure value at a given radius. The coloured lines are as before in Figures 2a and 3a.

shows the distribution of the normalised moments of inertia, and Figure 8 shows the distribution of the flattening ratios.

We find that the pressure profiles in Figure 6 agree well with previous studies and are more constrained for Uranus compared to Neptune. The pressure profiles were calculated from the density profiles under the assumption of hydrostatic equilibrium, $\nabla p = \rho \nabla U$, where U is the total (gravitational and rotational) potential of the planet.

The normalised moments of inertia depicted in Figure 7 are defined as $\lambda = I/(mR_{\text{mean}}^2)$, where I is the axial moment of inertia,

$$I = \int_V \rho r_{\perp}^2 dV, \quad (7)$$

and r_{\perp} denotes the perpendicular radial component with respect to the given axis, usually the axis of rotation. Further, R_{mean} is equal to l_{\max} , the outermost spheroid volumetric mean radius (see Section 2). We find that our average value for Uranus of 0.22595 agrees well with previous findings from Neuenschwander & Helled (2022), who found a range of 0.22594–0.22670. Our average value for Neptune of 0.23597 is lower than their finding of 0.23727–0.23900 for the standard

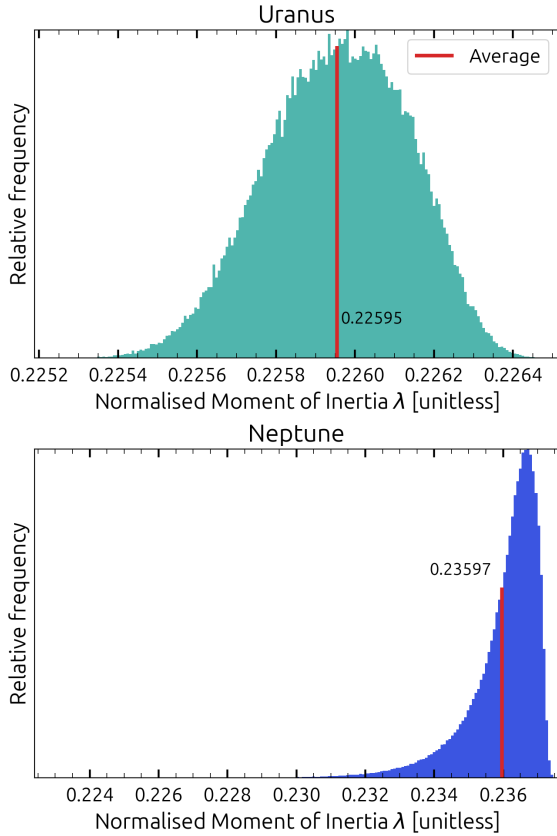


Fig. 7. Normalised moments of inertia. Weighted distribution of the normalised moments of inertia of successful density profiles. The red line denotes the average.

model, though our more frequent values agree with the lower end of their range.

The flattening ratios depicted in Figure 8 are defined as the ratio between the equatorial and mean radii:

$$r_f = \frac{R_{\text{eq}}}{R_{\text{mean}}}. \quad (8)$$

To compare our findings, we calculated the equatorial-to-mean volumetric radii ratios based on the values from Seidelmann et al. (2007). They are $(25\,559 \pm 4)/(25\,362 \pm 7)$ for Uranus and $(24\,764 \pm 15)/(24\,622 \pm 19)$ for Neptune. The resulting flattening ratios are $r_f = 1.0078 \pm 0.0003$ and $r_f = 1.0058 \pm 0.0010$ for Uranus and Neptune, respectively. Our inferred average value of 1.0068 in Figure 8 for Uranus is lower compared to Seidelmann et al. (2007), while our average of 1.0063 for Neptune is within their uncertainties. Our ToF implementation is benchmarked against Wisdom & Hubbard (2016), which shows that our flattening ratios should be accurate up to a relative error of 10^{-3} – 10^{-4} . This is significant given the uncertainties from Seidelmann et al. (2007) for Uranus and can therefore explain this deviation. Furthermore, the uncertainties given by Seidelmann et al. (2007) are quite conservative, especially given the uncertain rotation rates of these planets (for example Helled et al. 2010).

3.3. Discontinuities

In this section, we present an analysis on discontinuities of successful density profiles. First, a ‘discontinuity criterion’ (c_D)

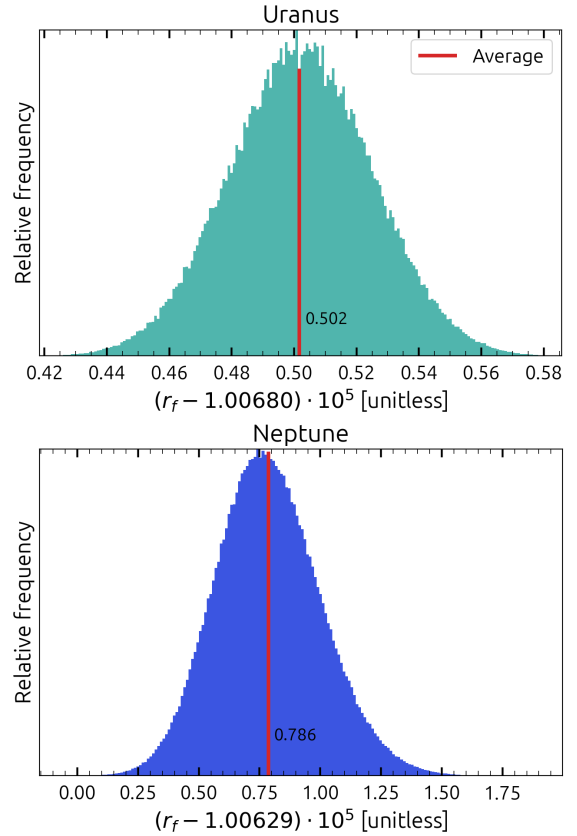


Fig. 8. Flattening ratios. Weighted distribution of the flattening ratios of successful density profiles. Note the scale is given as offsets from the constant 1.00680 for Uranus and 1.00629 for Neptune.

must be chosen in order to distinguish discontinuities from regular slopes. Unless otherwise specified, all results were obtained with $c_D = 100 \Delta\ell^{-1} \text{ kg m}^{-3}$. As a reminder, $\Delta\ell$ is equal to 24 960 m for Uranus and 24 186 m for Neptune (rounded to the nearest meter). Therefore, c_D has a value of $4.01 \cdot 10^{-3} \text{ kg m}^{-4}$ for Uranus and $4.13 \cdot 10^{-5} \text{ kg m}^{-4}$ for Neptune (rounded to two significant figures). As a second distinguishing feature, we only considered discontinuities that are ‘minimally maximal’. A precise definition of the term minimally maximal is given in Appendix D. In a nutshell, minimally maximal specifies that the (radial) width of a discontinuity can neither be extended (because it would add sections of the density profile below c_D) nor shortened (because it would remove sections of the density profile exceeding c_D).

Figure 9 shows the average discontinuity ‘intensity’ as a function of radius for all successful density profiles for both Uranus and Neptune. More precisely, it shows the average discontinuity height, $\overline{\Delta\rho}$, divided by the average discontinuity width, $\overline{\Delta r}$, for discontinuities that are minimally maximal and in agreement with c_D at their respective average normalised radius, r/R . For example, Figure 9a shows that minimally maximal and c_D -consistent discontinuities in Uranus with an average location of $r/R = 0.5$ possess an average slope of $\overline{\Delta\rho}/\overline{\Delta r} \sim 0.5\Delta\ell^{-1} \text{ kg m}^{-3}$. We note that profiles that have no minimally maximal and c_D -consistent discontinuity at any given location also contribute to the averages with a discontinuity intensity value of zero. Separately, we show $\overline{\Delta\rho}$ and $\overline{\Delta r}$ at their respective average normalised radii, r/R , in Figures D.1 for Uranus and D.2 for Neptune.

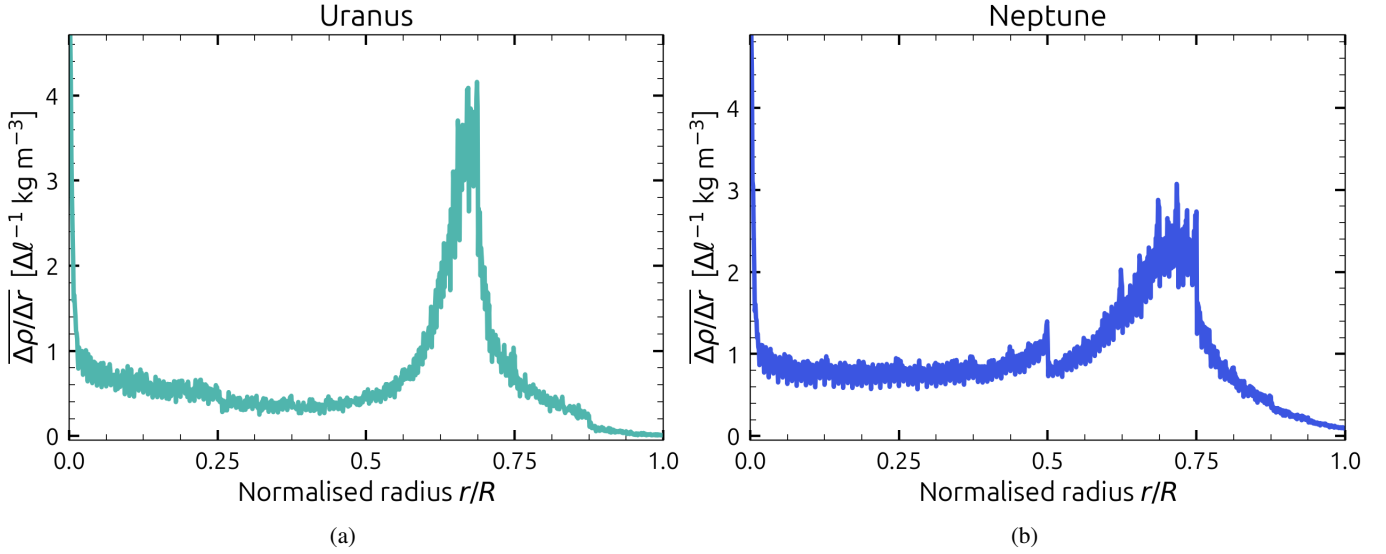


Fig. 9. Weighted discontinuity ‘intensity’ for Uranus (a) and Neptune (b). The figures depict the average height, $\overline{\Delta\rho}$, divided by the average width, $\overline{\Delta r}$, of the discontinuities. For example, Uranus discontinuities with an average location of $r/R = 0.5$ possess $\overline{\Delta\rho/\Delta r} \sim 0.5\Delta\ell^{-1} \text{ kg m}^{-3}$. Profiles with no discontinuities also contribute to the averages with a value zero.

For Uranus (Figure 9a), we observed a discontinuity intensity peak at $\sim 0.65R$, which explains the bump in the density profile distribution seen in Figure 2a. A discontinuity in this location could mark the transition point between the outer atmosphere dominated by hydrogen and helium and deeper layers dominated by, for example, water or rocks. There are no other significant discontinuity regions, although the innermost region tends to have bigger slopes in general, and thus the discontinuity intensity also increases inward. In the same plot for Neptune (Figure 9b), the effects are reduced in visibility, but the same general trends exist, with the peak being located slightly further out at $\sim 0.7R$. We note that here as well artefacts of the initial density profile generation method are visible, but this time as small cliffs around the inverses of multiples of two (we discuss this further in Section 4).

Modellers often assume a fixed number of density discontinuities (for example Nettelmann et al. 2013; Neuenschwander & Helled 2022; Morf et al. 2024; Militzer 2024). However, this leads to the question of how many of these discontinuities should be chosen to account for all possible density profiles (given a certain criterion c_D). To answer this, we analysed the number of discontinuities compatible with a given c_D . Figure 10 shows the distribution of the total number of (minimally maximal and c_D -consistent) discontinuities in a given profile over all successful density profiles. We find that the number of discontinuities in one solution can be quite large, particularly in comparison to the above-mentioned published models, which set the number of discontinuities to three. However, the results in Figure 10 clearly depend on the discontinuity criterion, c_D . We hence investigated the influence of c_D on the number of discontinuities in Figure 11. We show the average and upper 95th percentile number of discontinuities detected per successful density profile as a function of the criterion c_D . In addition, Table D.1 lists all values from this figure numerically. Figure 11 and Table D.1 should be read as follows: If we consider, for example, $c_D = 100\Delta\ell^{-1} \text{ kg m}^{-3}$ to be a discontinuity, then the average number of discontinuities on Uranus is four, and 95% of all solutions have eight discontinuities or fewer. Alternatively, the statement ‘95% of Uranian density profiles have at most one discontinuity’ is only correct

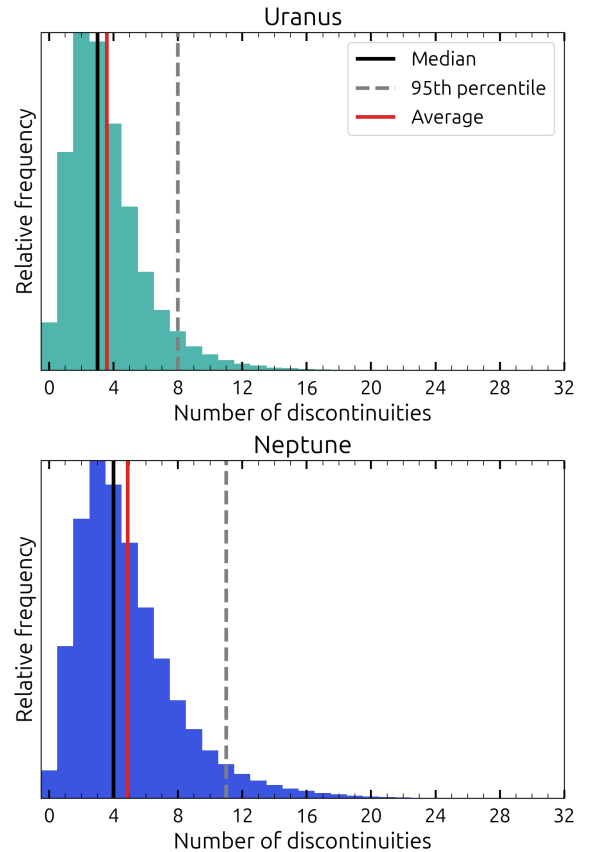


Fig. 10. Number of discontinuities. Weighted distribution of the total number of discontinuities detected in a successful density profile for $c_D = 100\Delta\ell^{-1} \text{ kg m}^{-3}$.

if we set $c_D \gtrsim 550\Delta\ell^{-1} \text{ kg m}^{-3}$. Finally, we note that we consistently tend to find slightly more jumps for Neptune than Uranus, probably due to the larger uncertainty in Neptune’s gravity data, which allows for a larger parameter space of solutions.

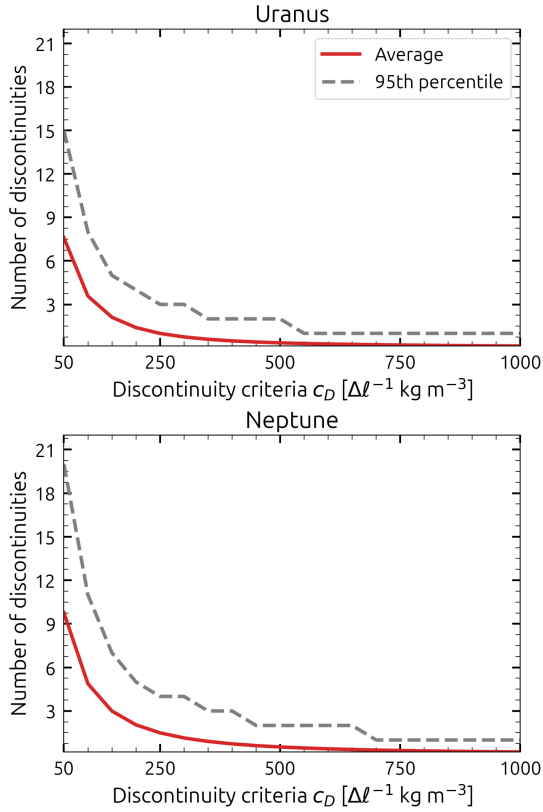


Fig. 11. Number of discontinuities versus c_D . The real valued average (red) and the integer valued upper 95th percentile (black) are depicted. All criteria from $50 \Delta\ell^{-1} \text{ kg m}^{-3}$ to $1000 \Delta\ell^{-1} \text{ kg m}^{-3}$ in steps of $50 \Delta\ell^{-1} \text{ kg m}^{-3}$ were considered.

4. Discussion

We begin by noting that our results depend on the measured values listed in Table 1. Some of these values remain uncertain. The rotation rates and the contributions of zonal winds, for example, remain unclear. Such fundamental uncertainties persist regardless of the method used. Below we focus on the limitations exclusive to our approach.

While our algorithm is able to efficiently and faithfully investigate the full space of planetary interior density profiles with an absolute minimum amount of assumptions, some caveats remain. Most importantly, our method for generating starting profiles has a higher variance (or degree of exploration) at some normalised radii compared to others. Listing 2 shows our generation method: the binary subdivision algorithm. The binary subdivision algorithm starts by setting the density value at $r/R = 1/2$. It does so by uniformly choosing at random from the entire range of possible values and subdividing the range into two halves in the process. It then recursively calls itself on those halves.

Listing 2: The binary subdivision algorithm.

```

input: int pivot, int step, float  $\underline{y}$ , float  $\bar{y}$ 
output: array rand_decr
define subdivide_binary(pivot, step,  $\underline{y}$ ,  $\bar{y}$ ):
     $y \leftarrow \text{uniform}(\underline{y}, \bar{y})$ 
    rand_decr[pivot]  $\leftarrow y$ 
    if step = 0 do
        break # base case; no further subdivision
    subdivide_binary(pivot-step, step//2,  $y$ ,  $\bar{y}$ )
    subdivide_binary(pivot+step, step//2,  $\underline{y}$ ,  $y$ )

```

As a result, some binary points in the range from $r/R = 0$ to $r/R = 1$ (for example 1/2, 1/4, 3/4) are decided earlier, and their bounds, \underline{y} and \bar{y} , for the (uniform) random generation are larger. Therefore, the variance at these points is higher than at others, and the generated starting density profiles tend to have kinks at these locations. These kinks can influence the final results, as can be seen in the small spikes that appear in Figure 5 and the cliffs in Figure 9. We nonetheless decided to use this method because it generates profiles sampled from the space of all possible density profiles such that the likelihood of generating a profile similar to another reference profile is equal everywhere. Appendix C.1 provides more details and a mathematically sound justification of the former statement. More practically speaking, we find that this limitation is acceptable because many of the starting profiles do not have these kinks and because the algorithm is able to remove the kinks after the initial generation. This is most apparent in Figure 9a, where the algorithm has almost completely removed the cliffs for Uranus because of the tighter gravitational moment constraints that are incompatible with density discontinuities at the inverse multiples of two in units of r/R . This can also be seen by directly comparing Figure D.1 (Uranus), where there are fewer kinks or cliffs visible at inverse multiples of two, with Figure D.2 (Neptune), where these kinks or cliffs are very prominent. We provide a discussion on the bias possibly introduced through the gradient method in Appendix C.2, which shows that our gradient descent method can be considered unbiased for the purpose of this work.

While the linear approximation (see Appendix B) for the gradient was sufficient here, we find that there are limitations in performance on harder problems. This is especially true if there are more measured gravitational moments or more precise measurements. There are several ways in which the algorithm could be modified accordingly. First, the optimisation scheme is always readily exchangeable. We used the Adam (Kingma & Ba 2017) scheme, but any number of other descent schemes could be implemented. Due to the structure of the ToF, a full linear gradient was the most amenable choice, but depending on the method used to calculate gravitational moments, the problem could be approached with a stochastic gradient. This would alter the requirements, though Adam is still likely to perform very well. Second, the density profile parametrisation is also a viable candidate for modifications. Our parametrisation has the advantage of being simple to implement and easy to analyse for the gradient. However, it cannot structurally change the density profile in a meaningful way, as the gradient cannot ‘see’ directions of changes very well. Density profiles that are very similar to each other can have parameters (p_i in Equation (3)) that are very different. Different parametrisations could be preferred for this task, such as using orthogonal bases of polynomials, for example.

Finally, we note that a more comprehensive approach could eliminate the difficulties with the ToF and the need for a linear approximation entirely. Our algorithm was performance-bound exclusively by the implementation of the ToF we used; therefore, a reduction in time usage there would have direct equal benefits. If a differentiable model could be trained to learn the ToF to within a certain acceptable loss, one would gain significant advantages in the accuracy of the gradients (and, presumably, the calculation of a single gradient would be quicker too). Alternatively, a non-differentiable machine model of the ToF would still have benefits in performance time but would require zero-order methods. Additionally, an iterative approach where the average starting point moves towards the best performant point every so often could mean more time is spent resolving the space of acceptable solutions. However, this would not have served this

work since our goal was the exploration of the entire space, not just the successful portion.

Regarding future work, the analysis done on the solution space could be expanded further. One could include higher-order or odd-gravitational moments, especially in connection to dedicated Uranus and/or Neptune missions such as in Parisi et al. (2024); Mankovich et al. (2025). Chiefly, an analysis of the compositional and thermal structure of the planets based on the found density profiles could be performed. Special attention should be given to investigating the compositional structure around the hypothesised transition point between the hydrogen-helium rich outer layer and the below layer consisting of denser elements such as water or rocks. Morf & Helled (2025) perform such a compositional and thermal analysis for profiles of Uranus and Neptune based on random starting profiles generated in a fashion similar to our models. However, their methods are limited by computational cost and hence yield only a handful of models (four for each planet). Hence, the key challenge would lie within ensuring a high enough computational efficiency that does not come at the cost of physical self-consistency.

5. Summary and conclusions

We have presented a novel algorithm for efficiently and faithfully sampling planetary interior density profiles. Unlike conventional MCMC approaches, our method employs a new optimisation-based strategy. We used this algorithm to construct interior models of Uranus and Neptune, and we analysed the results from an extensive suite of simulations and compared them to prior studies.

Our results are broadly consistent with previous studies, whose solutions typically fall within the 95% confidence interval of our model ensemble. However, the earlier models do not capture the full diversity of possible interior configurations, particularly in the deep interior. We also discussed the limitations related to uneven sampling variance in our generation method and outlined potential future directions, including refining the algorithm in various ways and expanding the model scope to include compositional analyses. Our key conclusions are summarised as follows:

- For Uranus and Neptune, literature models with sharp density discontinuities are biased towards a high central density. Literature models without these features are biased towards the peak of the solution space. Models around the median central density have only recently been proposed by Morf & Helled (2025);
- The median density (as a function of radius) for Uranus and Neptune shows gradual increases in density consistent with composition gradients. In particular, the medians maintain a consistent slope towards the centre and do not flatten off, with moderate central densities of $\sim 6 \text{ g cm}^{-3}$ for Uranus and $\sim 7.5 \text{ g cm}^{-3}$ for Neptune;
- The solution space for Uranus and Neptune is least well constrained for the innermost region of the planets, with the 95% confidence interval reaching sizes of $\Delta\rho \sim 11 \text{ g cm}^{-3}$ for Uranus and $\Delta\rho \sim 13 \text{ g cm}^{-3}$ for Neptune;
- Accounting for the covariance between J_2 and J_4 , such as given in French et al. (2024) for Uranus, has a negligible effect on our results;
- Discontinuities with a slope greater than 0.02 kg m^{-4} are generally rare;
- Discontinuities with more moderate slopes ($0.002\text{--}0.02 \text{ kg m}^{-4}$) are frequent. The literature generally tends to only consider three or fewer discontinuities, which excludes a significant part of the solution space. If discontinuities are considered, we recommend adopting the values provided in Figure 11 or equivalently those in Table D.1;
- If at least one discontinuity is present, it is most likely to be found around $\sim 0.65R$ for Uranus and $\sim 0.7R$ for Neptune.

By embracing uncertainty rather than constraining it, our approach faithfully captures the complexity of planetary interiors. Our ensemble of solutions does not converge on a single solution but defines the space within which physical reality must lie. This empirical and agnostic perspective reaffirms that progress in planetary science depends not only on more precise data but also on broader, more flexible ways of interpreting it.

Acknowledgements. We thank the anonymous referee for valuable comments that greatly improved the manuscript. We thank user Kaolay for making their Python implementation of Adam available for use on GitHub. This work was supported by the Swiss National Science Foundation (SNSF) through a grant provided as a part of project number 215634: <https://data.snf.ch/grants/grant/215634>.

References

- Desch, M. D., Connerney, J. E. P., & Kaiser, M. L. 1986, *Nature*, 322, 42
- French, R. G., Hedman, M. M., Nicholson, P. D., Longaretti, P.-Y., & McGhee-French, C. A. 2024, *Icarus*, 411, 115957
- Helled, R., & Fortney, J. J. 2020, *Philos. Trans. Roy. Soc. Lond. Ser. A*, 378, 20190474
- Helled, R., Anderson, J. D., & Schubert, G. 2010, *Icarus*, 210, 446
- Helled, R., Anderson, J. D., Podolak, M., & Schubert, G. 2011, *ApJ*, 726, 15
- Jacobson, R. A. 2009, *AJ*, 137, 4322
- Jacobson, R. A., & Park, R. S. 2025, *AJ*, 169, 65
- Karkoschka, E. 2011, *Icarus*, 215, 439
- Kingma, D. P., & Ba, J. 2017, *Adam: A Method for Stochastic Optimization*
- Lindal, G. F. 1992, *AJ*, 103, 967
- Lindal, G. F., Lyons, J. R., Sweetnam, D. N., et al. 1987, *J. Geophys. Res.*, 92, 14987
- Mankovich, C. R., Parisi, M., Landau, D. F., & Dewberry, J. W. 2025, *Planet. Sci. J.*, 6, 276
- Marley, M. S., Gómez, P., & Podolak, M. 1995, *J. Geophys. Res.*, 100, 23349
- Militzer, B. 2024, *PNAS*, 121, e2403981121
- Morf, L. 2025, *PyToF: a numerical implementation of the Theory of Figures algorithm (4th, 7th, 10th order) including barotropic differential rotation*
- Morf, L., & Helled, R. 2025, *A&A*, 704, A183
- Morf, L., Müller, S., & Helled, R. 2024, *A&A*, 690, A105
- Movshovitz, N., & Fortney, J. J. 2022, *Planet. Sci. J.*, 3, 88
- Nettelmann, N., Helled, R., Fortney, J. J., & Redmer, R. 2013, *Planet. Space Sci.*, 77, 143
- Neuenschwander, B. A., & Helled, R. 2022, *MNRAS*, 512, 3124
- Parisi, M., Friedson, A. J., Mankovich, C. R., et al. 2024, *Planet. Sci. J.*, 5, 116
- Podolak, M., Podolak, J. I., & Marley, M. S. 2000, *Planet. Space Sci.*, 48, 143
- Seidelmann, P. K., Archinal, B. A., A'hearn, M. F., et al. 2007, *Celest. Mech. Dyn. Astron.*, 98, 155
- Vazan, A., & Helled, R. 2020, *A&A*, 633, A50
- Wang, B., Yan, J., Gao, W., et al. 2023, *A&A*, 671, A70
- Wisdom, J., & Hubbard, W. B. 2016, *Icarus*, 267, 315
- Zharkov, V. N., & Trubitsyn, V. P. 1978, *Physics of Planetary Interiors (Astronomy and Astrophysics Series)* (Tucson: Pachart)
- Zhu, W., & Dong, S. 2021, *ARA&A*, 59, 291

Appendix A: Algorithm

We give further details regarding the implementation of the optimisation algorithm. For the reader's convenience, listing 1 presenting a high-level overview of the optimisation algorithm is reproduced here.

Listing 3: The optimisation algorithm (reprinted).

```

input: int epochs
output: tuple results
begin
    params ← create_start_params()
    for epoch in epochs do
        check convergence
        for steps in epochsize do
            params ← project(opt_step(params))
    return results
end
    
```

create_start_params()

A random decreasing function is generated to serve as the starting distribution. To do this, we epacke a recursive subdivision method described in listing 2 which is reprinted below for the reader's convenience.

Listing 4: The binary subdivision algorithm. (reprinted)

```

input: int pivot, int step, float  $\underline{y}$ , float  $\bar{y}$ 
output: array rand_decr
define subdivide_binary(pivot, step,  $\underline{y}$ ,  $\bar{y}$ ):
     $y \leftarrow \text{uniform}(\underline{y}, \bar{y})$ 
    rand_decr[pivot] ←  $y$ 
    if step = 0 do
        break # base case; no further subdivision
    subdivide_binary(pivot-step, step//2,  $y$ ,  $\bar{y}$ )
    subdivide_binary(pivot+step, step//2,  $\underline{y}$ ,  $y$ )
    
```

Note that `uniform(a,b)` denotes a function that generates a number uniformly at random in the given interval and that `//` denotes floor integer division. Each call is supplied with a pivot (where the algorithm is currently operating), a step size, and bounding values. We generate a value y uniformly at random between \underline{y} and \bar{y} and set the value of the function at the pivot to y . These bounds ensure that the resulting array `rand_decr` is decreasing. If we have reached step size 0, the recursion breaks, as this is the base case and there are no other pivots to set. Otherwise, we must still process the y -values of the pivots that lay step to the left and right of the pivot (and all values in-between later on, recursively). In that case, we call the function again at those pivots, with halved step size, and with y as the new lower and upper bound respectively.

To generate a random decreasing function of N values between 0 and 1, the method is called as `subdivide_binary(N/2, N/4, 0, 1)`. The resulting array `rand_decr` is then rescaled by α to ensure the resulting density profile has the correct mass. Then, the profile is used to calculate the starting parameters.

opt_step

To calculate `opt_step`, we determine goal functions F_1 and F_2 and a gradient. The first goal function we wished to minimise

was the sum of the costs of being far away from the d observed gravitational moments J_{2n} :

$$F_1(\rho) = \frac{1}{d} \left[\sum_{n=1}^d \frac{1}{2} \frac{(J_{2n, \text{ToF}}(\rho) - J_{2n}^*)^2}{\sigma J_{2n}^2} + \sum_{\substack{n,m=1 \\ n \neq m}}^d \frac{1}{2} \text{cov}(J_{2n}^*, J_{2m}^*) (J_{2n, \text{ToF}}(\rho) - J_{2n}^*) (J_{2m, \text{ToF}}(\rho) - J_{2m}^*) \right]. \quad (\text{A.1})$$

The gravitational moments calculated by the ToF algorithm PyToF (Morf 2025) are denoted as $J_{2n, \text{ToF}}(\rho)$. The observed gravitational moments are denoted as J_{2n}^* and their standard deviations as σJ_{2n} . For numerical stability, we include a second cost function for deviating from the planet's mass:

$$F_2(\rho) = c \frac{1}{2} \left(\frac{m_{\text{calc}}}{m} - 1 \right)^2. \quad (\text{A.2})$$

The numerical constant c is chosen so that the mass term is not dominated by the cost term and was set to 100 based on experimental experience.

We set the gradient for the optimisation $\nabla_p F$ as

$$\nabla_p F = \nabla_p F_1 + \nabla_p F_2 \|\nabla_p F_1\|, \quad (\text{A.3})$$

where p denote the parameters introduced in Equation 3. This choice ensures that $\nabla_p F_2$ never gets dominated by $\nabla_p F_1$, but makes it hard to find an explicit form for F . The gradients $\nabla_p F_1$ and $\nabla_p F_2$ are calculated in Appendix B. For the case of $\text{cov}(J_{2n}^*, J_{2m}^*) = 0$ for all m, n , the results read

$$\begin{aligned} \nabla_p F_1 &= \frac{1}{d} \sum_{i=1}^d \nabla_p J_{2n, \text{ToF}}(\rho) \frac{(J_{2n, \text{ToF}}(\rho) - J_{2n}^*)}{\sigma J_{2n}^2} \\ &= \frac{1}{d} \sum_{i=1}^d \frac{(J_{2n, \text{ToF}}(\rho) - J_{2n}^*)}{\sigma J_{2n}^2} \left[- \left(\frac{R_m}{R_{eq}} \right)^{2n} (S_{2n, \text{ToF}}^{(-1)} - S_{2n, \text{ToF}}^{(-2)}) \right. \\ &\quad \left. \left(\frac{\mathbf{e}^p}{\mathbf{w}} \odot \sum \frac{\mathbf{w}}{\mathbf{e}^p} - \gamma \cdot \mathbf{e}^p \cdot \left(\left(\sum \frac{\mathbf{e}^p}{\mathbf{w}} \right)^\top \frac{\mathbf{w}}{\mathbf{e}^p} \right) \right) \right], \end{aligned} \quad (\text{A.4})$$

$$\nabla_p F_2 = c \nabla_p m_{\text{calc}} \left(\frac{m_{\text{calc}}}{m} - 1 \right) = c 4\pi \Delta l \mathbf{e}^p \left(\frac{m_{\text{calc}}}{m} - 1 \right),$$

where R_m and R_{eq} are the mean and equatorial radius, respectively. $S_{2n, \text{ToF}}^{(-1)}$ denotes the last (innermost) and $S_{2n, \text{ToF}}^{(-2)}$ the second to last result of $S_{2n}(z)$ as defined by the ToF (see Appendix A of Morf et al. 2024) and calculated by PyToF. \mathbf{e}^p denotes the element-wise exponential of the parameter vector. \mathbf{e}^p/\mathbf{w} and \mathbf{w}/\mathbf{e}^p are defined analogously. $\overleftarrow{\Sigma} \mathbf{e}^p/\mathbf{w}$ denotes the cumulative sum vector of \mathbf{e}^p/\mathbf{w} and $\overleftarrow{\Sigma} \mathbf{w}/\mathbf{e}^p$ denotes the reverse cumulative sum vector of \mathbf{w}/\mathbf{e}^p (a cumulative sum starting with a sum of all elements and ending with the last element only). Lastly, $^\top$ denotes the usual vector product and \odot the element-wise product.

With the gradient, we update the parameters in an optimisation step according to the Adam gradient descent scheme (Kingma & Ba 2017). We chose Adam because of its excellent performance in practice. Hyperparameters were tuned experimentally, the learning rate was set at 0.08 because a rapid convergence to a solution was desired. All other parameters were found to perform well at default values.

project()

After each optimisation step, the projection Π is applied to enforce the minimum density increase Δ_{\min} per step:

$$\Pi(p_i) = \max(p_i, \ln(w_i \Delta_{\min})). \quad (\text{A.5})$$

Note that the algorithm treats the parameters p_i , w_i , and Δ_{\min} as unitless, Equation 3 reads

$$\rho_k = \left(\sum_{i=0}^{k-1} \frac{e^{p_i}}{w_i} \alpha \right) \frac{\text{kg}}{\text{m}^3} \quad (\text{A.6})$$

with units. This implies that the minimal possible central density (where $p_i < \ln(w_i \Delta_{\min})$ for all $i \in 0, \dots, N-2$) is given by

$$\rho_{\min}^{\text{centre}} = \alpha N \Delta_{\min} \frac{\text{kg}}{\text{m}^3} = 1024 \alpha \frac{\text{kg}}{\text{m}^3} \quad (\text{A.7})$$

for $N = 1024$ and $\Delta_{\min} = 1$. α ensures that our density profiles always have the correct mass, regardless of the value of α . The introduction of the function F_2 ensures that $\alpha \neq 1$ comes with a cost, but does not force $\alpha = 1$. Consequently, even profiles that lie below the minimal profile linearly increasing to 1024 kg/m^3 in the centre are considered by the algorithm. This is best seen in the zoom panels of subfigures 2a and 3a, where some density profiles are quite flat in the outermost parts of the planets.

check convergence

Before each epoch, we check for convergence. We do this by comparing the change in cost. If the exponential moving average of the relative change in cost is close enough to one (difference of less than 0.0005), we terminate and return the following results:

- The starting density profile
- The final density profile
- The gravitational moments J_2 and J_4
- The pressure profile
- The normalised moment of inertia
- The flattening ratio
- Whether ρ_{\max} has been respected

Appendix B: Gradient calculation

Here, we calculate the gradients of F_1 and F_2 :

$$\nabla_p F_1 = \frac{1}{d} \sum_{i=1}^d \nabla_p J_{2n, \text{ToF}}(\rho) \frac{(J_{2n, \text{ToF}}(\rho) - J_{2n}^*)}{\sigma J_{2n}^2} \quad (\text{B.1})$$

$$\nabla_p F_2 = c \nabla_p m_{\text{calc}} \left(\frac{m_{\text{calc}}}{m} - 1 \right).$$

Since $J_{2n} = - \left(\frac{R_m}{R_{\text{eq}}} \right)^{2n} S_{2n}(1)$, we get

$$\nabla_p J_{2n, \text{ToF}}(\rho) = - \left(\frac{R_m}{R_{\text{eq}}} \right)^{2n} \nabla_p S_{2n, \text{ToF}}(\rho) \Big|_{z=1}. \quad (\text{B.2})$$

We avoid some notation for legibility and note that

$$\nabla_p S = \frac{dS}{dp} = \frac{d\rho}{dp} \frac{dz}{d\rho} \frac{dS}{dz} \Big|_{z=1}. \quad (\text{B.3})$$

Each of these terms can be readily calculated. By plugging in $z = 1$, the rightmost derivative is approximated as

$$(S_{2n, \text{ToF}}^{(-1)} - S_{2n, \text{ToF}}^{(-2)}) / \Delta z, \quad (\text{B.4})$$

where $S_{2n, \text{ToF}}^{(-i)}$ denotes the last/innermost ($i = -1$) and second to last ($i = -2$) value calculated by the ToF. The small change Δz does not have to be further specified, it vanishes in the end result. The middle derivative (denoted element-wise here, but actually a vector) is easily calculated by its inverse, which we can approximate as

$$\frac{d\rho_k}{dz} \approx \frac{e^{p_{k-1}}}{w_{k-1}} \alpha / \Delta z \quad (\text{B.5})$$

since the difference between ρ_k and ρ_{k-1} is exactly this term. We calculated the leftmost derivative element-wise: Write ∂_i for the partial derivative by p_i , and therefore $\frac{\partial}{\partial p_i}$. The k -th entry of ρ is $\rho_k = \sum_{j=0}^{k-1} \frac{e^{p_j}}{w_j} \alpha$. Thus,

$$\partial_i \rho_k = \partial_i \left(\sum_{j=0}^{k-1} \frac{e^{p_j}}{w_j} \right) \alpha + \sum_{j=0}^{k-1} \frac{e^{p_j}}{w_j} \partial_i \alpha. \quad (\text{B.6})$$

The first derivative is only non-zero if $k > i$, since p_i only appears in the sum $\sum_{j=0}^{k-1} \frac{e^{p_j}}{w_j} \alpha$ for $k > i$. Therefore,

$$\partial_i \left(\sum_{j=0}^{k-1} \frac{e^{p_j}}{w_j} \right) = \begin{cases} \partial_i \frac{e^{p_i}}{w_i} = \frac{e^{p_i}}{w_i} & k > i \\ 0 & k \leq i \end{cases} \quad (\text{B.7})$$

since w_i is constant with respect to p_i . The second derivative, $\partial_i \alpha$, is

$$\partial_i \alpha = \partial_i \frac{m}{m_{\text{calc}}} = - \frac{m}{m_{\text{calc}}} \partial_i m_{\text{calc}}. \quad (\text{B.8})$$

We remind that $m_{\text{calc}} = 4\pi \sum_{k=0}^{N-1} \tilde{\rho}_k l_k^2 \Delta l$, where $\tilde{\rho}_k = \sum_{j=0}^{k-1} \frac{e^{p_j}}{w_j}$ and note that l_k is constant with respect to the parameters. Thus,

$$\begin{aligned} \partial_i \alpha &= - \frac{m}{m_{\text{calc}}} \partial_i \left(4\pi \sum_{k=0}^{N-1} \sum_{j=0}^{k-1} \frac{e^{p_j}}{w_j} l_k^2 \Delta l \right) \\ &= - \frac{m}{m_{\text{calc}}} 4\pi \Delta l \sum_{k=0}^{N-1} l_k^2 \partial_i \sum_{j=0}^{k-1} \frac{e^{p_j}}{w_j}, \end{aligned} \quad (\text{B.9})$$

where again only terms with $k > i$ survive (always yielding e^{p_i}/w_i), so

$$\partial_i \alpha = - \frac{m}{m_{\text{calc}}} 4\pi \Delta l \sum_{k=i+1}^{N-1} l_k^2 \frac{e^{p_i}}{w_i}. \quad (\text{B.10})$$

Rewriting m/m_{calc} as α and arranging terms yields

$$\partial_i \alpha = -\alpha \underbrace{\frac{4\pi \Delta l}{m_{\text{calc}}}}_{=: \gamma} e^{p_i} \frac{\sum_{k=i+1}^{N-1} l_k^2}{w_i}. \quad (\text{B.11})$$

At this point, it becomes clear why the weights are defined as they are in Equation 4. If we remember $w_i = \sum_{k=i+1}^{N-1} l_k^2$, we can simplify the expression as

$$\partial_i \alpha = -\alpha \gamma e^{p_i}. \quad (\text{B.12})$$

Putting everything together, we arrive at

$$\partial_i \rho_k = \begin{cases} \sum_{j=0}^{k-1} \frac{e^{p_j}}{w_j} \cdot (-\alpha \gamma e^{p_i}), & k \leq i \\ \frac{e^{p_i}}{w_i} \alpha + \sum_{j=0}^{k-1} \frac{e^{p_j}}{w_j} \cdot (-\alpha \gamma e^{p_i}), & k > i. \end{cases} \quad (\text{B.13})$$

We can now combine everything to finally obtain $\nabla_p S$. In summary, we have

$$\begin{aligned} \frac{dS}{dz} \Big|_{z=1} &= S_{2n, \text{ToF}}^{(-1)} - S_{2n, \text{ToF}}^{(-2)} \quad (\text{B.14}) \\ \frac{d\rho}{dz} &= \alpha \begin{pmatrix} 0 \\ e^{p_0}/w_0 \\ \vdots \\ e^{p_{N-2}}/w_{N-2} \end{pmatrix} \Rightarrow \frac{dz}{d\rho} = \frac{1}{\alpha} \begin{pmatrix} 0 \\ w_0/e^{p_0} \\ \vdots \\ w_{N-2}/e^{p_{N-2}} \end{pmatrix}, \\ \frac{d\rho}{d\mathbf{p}} &= \alpha \begin{pmatrix} 0 & \frac{e^{p_0}}{w_0} & \cdots & \frac{e^{p_0}}{w_0} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \frac{e^{p_{N-2}}}{w_{N-2}} \end{pmatrix} - \alpha \gamma \underbrace{\begin{pmatrix} e^{p_i} \\ \sum_{j=0}^{k-1} \frac{e^{p_j}}{w_j} \end{pmatrix}}_{k \rightarrow} \Big\} \downarrow. \end{aligned}$$

We combine the above as vector and matrix products. They may be conveniently written if we define $\mathbf{e}^{\mathbf{p}}$ to be the element-wise exponentiation of the parameter vector, and analogously define $\mathbf{e}^{\mathbf{p}}/\mathbf{w}$ and $\mathbf{w}/\mathbf{e}^{\mathbf{p}}$. Then write $\overleftarrow{\Sigma} \mathbf{e}^{\mathbf{p}}/\mathbf{w}$ for the cumulative sum vector of $\mathbf{e}^{\mathbf{p}}/\mathbf{w}$ and $\overleftarrow{\Sigma} \mathbf{w}/\mathbf{e}^{\mathbf{p}}$ for the reverse cumulative sum vector of $\mathbf{w}/\mathbf{e}^{\mathbf{p}}$. With \top the usual vector product and \odot the element-wise product, we get

$$\begin{aligned} \nabla_p J_{2n, \text{ToF}}(\rho) &= - \begin{pmatrix} R_m \\ R_{eq} \end{pmatrix}^{2n} \left(S_{2n, \text{ToF}}^{(-1)} - S_{2n, \text{ToF}}^{(-2)} \right) \quad (\text{B.15}) \\ &\quad \left(\frac{\mathbf{e}^{\mathbf{p}}}{\mathbf{w}} \odot \overleftarrow{\Sigma} \frac{\mathbf{w}}{\mathbf{e}^{\mathbf{p}}} - \gamma \cdot \mathbf{e}^{\mathbf{p}} \cdot \left(\left(\overleftarrow{\Sigma} \frac{\mathbf{e}^{\mathbf{p}}}{\mathbf{w}} \right)^\top \frac{\mathbf{w}}{\mathbf{e}^{\mathbf{p}}} \right) \right). \end{aligned}$$

We are almost done and just need to consider the remaining $\nabla_p m_{\text{calc}}$. Luckily, during our calculation of $\partial_i \alpha$, we actually already calculated this expression. Since we calculated

$$\partial_i \alpha = - \frac{\alpha}{m_{\text{calc}}} \partial_i m_{\text{calc}} = - \alpha \frac{4\pi \Delta l}{m_{\text{calc}}} e^{p_i}, \quad (\text{B.16})$$

we can infer directly

$$\partial_i m_{\text{calc}} = 4\pi \Delta l e^{p_i} \quad (\text{B.17})$$

or in vector form $\nabla_p m_{\text{calc}} = 4\pi \Delta l \mathbf{e}^{\mathbf{p}}$. This finally yields the expressions presented in Appendix A, Equation A.4.

Appendix C: Bias

In this section, we address potential biases of our algorithm in detail.

The observational data together with their respective uncertainties allow us to decide whether a given density profile is plausible (fits the data) or not. We call the set of all plausible density profiles the ‘space of possible interior density profiles that fit their known observational data’ or the solution space for short. Our algorithm samples this solution space. This sampling is unbiased if all plausible density profiles in the solution space have equal likelihood of being generated by our algorithm.

Formally, we may call the solution space S . An algorithm A is a random variable that produces density profiles $\rho \in S$. It has a probability density function $f_A : S \rightarrow \mathbb{R}_{\geq 0}$, which fulfils

$$\Pr[A \in R] = \int_R f_A d\mu, \quad (\text{C.1})$$

for any observable event, R . The probability measure on S is μ . We call the algorithm A unbiased if its probability density function f_A is equal to 1 almost everywhere. This is the formal definition of all profiles having equal likelihood of being generated. Note that this definition is stronger than just demanding $\Pr[A = \rho]$ be equal for all $\rho \in S$. Every algorithm fulfils this weaker requirement, because the probability to generate exactly ρ is $\mu(\{\rho\}) = 0$. As a result, a more careful definition is needed: We also require that all regions $R \subseteq S$ have proportional likelihood $\int_R f_A d\mu = \mu(R)$.

To define a concept of distance between profiles, we employed the infinity norm

$$\|\rho - \rho'\|_\infty = \max_{i \in \{1, \dots, N\}} |\rho_i - \rho'_i| \quad (\text{C.2})$$

which allowed us to approximate any region in S as the countable union of distinct neighbourhoods N_ρ^ε of radius $\varepsilon/2$ around ρ :

$$N_\rho^\varepsilon = \left\{ \rho' \mid \|\rho - \rho'\|_\infty < \frac{\varepsilon}{2} \right\}. \quad (\text{C.3})$$

We call two density profiles, ρ and ρ' , similar or ε -close if $\rho' \in N_\rho^\varepsilon$. From now on, we refer to the measure of N_ρ^ε as its volume, and write $V(N_\rho^\varepsilon)$.

The algorithm A first generates a starting profile. It subsequently evolves this starting profile through optimisation towards a result profile, implausible results are later discarded. If our algorithm A generates profiles such that all regions N_ρ^ε of equal radius have equal volume $V(N_\rho^\varepsilon)$ (probability), and it then evolves these regions such that their outputs also all have the same volume, then the algorithm must be unbiased: The probability that our algorithm ends up in R depends only on the probability measure μ of R in S , not on the algorithm A , because it treats all regions equally. The relative frequency of successful runs in R during our simulation is then a valid proxy for the size (probability) of R in S , which allows us to calculate results such as confidence intervals.

We will investigate the above steps in the next two sections of this appendix in detail. The first step – generating profiles with equal volume everywhere – is treated analytically. The second step – evolving profiles without distorting volumes – is treated using empirical evidence obtained by monitoring the behaviour of the descent. Additionally, we analyse the distortions that do occur in the descent method to quantify their impact on our results.

C.1. Generation method

We can convert the space of all density profiles $\rho = (\rho_0, \dots, \rho_{N-1})$ to its corresponding $N - 1$ -dimensional space of increases $\Delta\rho_k = \rho_{k+1} - \rho_k$ between values: $\Delta\rho = (\Delta\rho_0, \dots, \Delta\rho_{N-2})$. If we then rescale this space, we obtain the standard $(N - 2)$ -simplex. The uniform distribution over the standard $(N - 2)$ -simplex is the flat Dirichlet distribution obtained from the general Dirichlet distribution

$$f(\Delta\rho_0, \dots, \Delta\rho_{N-2}) = \frac{\Gamma(\alpha(N-1))}{\Gamma(\alpha)^{N-1}} \prod_{k=0}^{N-2} \Delta\rho_k^{\alpha-1} \quad (\text{C.4})$$

by setting $\alpha = 1$. f is a probability density function. This forces $\Delta\rho_k^0 = 1$ and thus the distribution has constant value $f = (N - 2)!$ and is therefore uniform. This is natural, given that the volume of the simplex is $\frac{1}{(N-2)!}$. This distribution is not, however, uniformly dense in the space of density profiles. Profiles around

the average (with each increase $\sim 1/(N-1)$) have many more neighbours than sparse profiles (with most increases close to 0). This is because, given some small radius $\varepsilon/2$, the volume of the neighbourhood N_ρ^ε can be bounded as

$$\prod_{k=1}^{N-2} \left[\min\left(\frac{\varepsilon}{2}, \frac{\Delta\rho_{k-1}}{2}\right) + \min\left(\frac{\varepsilon}{2}, \frac{\Delta\rho_k}{2}\right) \right] \leq V(N_\rho^\varepsilon) \quad (\text{C.5})$$

$$V(N_\rho^\varepsilon) \leq \prod_{k=1}^{N-2} \left[\min\left(\frac{\varepsilon}{2}, \Delta\rho_{k-1}\right) + \min\left(\frac{\varepsilon}{2}, \Delta\rho_k\right) \right],$$

which is a bound tighter (and hence better) compared to a bound relying on ε alone. Using only half of each increase ($\Delta\rho_{k-1}/2$ and $\Delta\rho_k/2$) guarantees no overlap between increases but thus undercounts possible neighbours, whereas using the full increase ($\Delta\rho_{k-1}$ and $\Delta\rho_k$) both times has overlap and thus overcounts. For the average case, it holds $\varepsilon \leq \Delta\rho_k$ for sufficiently small ε , and thus

$$\min\left(\frac{\varepsilon}{2}, \frac{\Delta\rho_k}{2}\right) = \frac{\varepsilon}{2} \quad (\text{C.6})$$

always. In that case the volume therefore equals $V(N_\rho^\varepsilon) = \varepsilon^{N-2}$. For the sparse case however, this is smaller ($V(N_\rho^\varepsilon) < \varepsilon^{N-2}$), because in this case there exist $\Delta\rho_k$ smaller than $\varepsilon/2$.

We look for a probability density function \hat{f} for generating initial profiles that is uniformly dense in the space of density profiles:

$$V(\hat{f}(N_\rho^\varepsilon)) = \int_{N_\rho^\varepsilon} \hat{f}(\rho') dV = \text{const.}, \quad (\text{C.7})$$

where the integration over dV includes all $\rho' \in N_\rho^\varepsilon$. This is fulfilled by

$$\hat{f}(\Delta\rho_0, \dots, \Delta\rho_{N-2}) = \prod_{k=0}^{N-2} \frac{1}{\Delta\rho_k}, \quad (\text{C.8})$$

equivalent to the Dirichlet distribution if one were to set $\alpha = 0$ (which is impossible due to the gamma function scale factor). \hat{f} fulfils Equation C.7 because it is the exact inverse of the volume distortion.

We now show that our generation function f_{BD} (the binary subdivision algorithm) takes on a very similar shape. At the last phase of the recursion, conditioned on the fact that all higher-level calls successfully chose the correct density values, the probability to choose ρ_k out of all possible values for index k is

$$\frac{1}{\Delta\rho_{k-1} + \Delta\rho_k}. \quad (\text{C.9})$$

This is because the uniform distribution has probability density equal to one over the volume of the interval, which in this case must be $\Delta\rho_{k-1} + \Delta\rho_k$. At higher phases of the recursion, this is not the case. The interval (and thus the variance) is larger and the probability is smaller. This causes the artefacts discussed in Section 4. Barring these limitations, we find that f_{BD} takes on the shape

$$f_{\text{BD}}(\Delta\rho_0, \dots, \Delta\rho_{N-2}) = \prod_{k=0}^{N-2} \frac{1}{\Delta\rho_{k-1} + \Delta\rho_k}, \quad (\text{C.10})$$

which also fixes the volume distortion like \hat{f} does and is actually computable. With this, we find that our volume is now bounded as

$$\prod_{k=1}^{N-2} \frac{\min\left(\frac{\varepsilon}{2}, \frac{\Delta\rho_{k-1}}{2}\right) + \min\left(\frac{\varepsilon}{2}, \frac{\Delta\rho_k}{2}\right)}{\Delta\rho_{k-1} + \Delta\rho_k} \leq V(f_{\text{BD}}(N_\rho^\varepsilon)) \quad (\text{C.11})$$

$$V(f_{\text{BD}}(N_\rho^\varepsilon)) \leq \prod_{k=1}^{N-2} \frac{\min\left(\frac{\varepsilon}{2}, \Delta\rho_{k-1}\right) + \min\left(\frac{\varepsilon}{2}, \Delta\rho_k\right)}{\Delta\rho_{k-1} + \Delta\rho_k}.$$

For a given ε (that is, for a given level of stringency as to what two profiles are considered similar) and small enough $\Delta\rho_{k-1}, \Delta\rho_k < \varepsilon$, the volume becomes constant (const. $\in [0.5, 1]$), which is what was desired.

In summary, by correcting for the uneven distribution of possible density profiles in the simplex through the use of a sampling method with the inverse probability distribution, we attain a method of generating density profiles that samples from the space of possible density profiles without bias. We emphasise again that our generation method is not the uniform distribution of the space of density profiles. We conjecture, however, that a uniform and unbiased generation method (formally, a method with uniform probability distribution and uniformly distributed probability measure) is, in fact, impossible.

C.2. Gradient method

The usage of a gradient can result in local solutions being ignored in favour of a large and prominent global solution, which would introduce bias. For our method to be unbiased, the search method must return close, local results. To be more specific, call the local solution $\hat{\rho}$. The gradient method G transforms inputs into solutions. The target region for G is the neighbourhood $N_{\hat{\rho}}^\varepsilon$. Successful inputs are thus the set $G^{-1}(N_{\hat{\rho}}^\varepsilon)$. The size ratio of G at $\hat{\rho}$ with distance ε is thus

$$r_G(\hat{\rho}) = \frac{V(G^{-1}(N_{\hat{\rho}}^\varepsilon))}{V(N_{\hat{\rho}}^\varepsilon)}, \quad (\text{C.12})$$

which ideally should be independent of ε . The universal size ratio \bar{r}_G of G is equal to the ratio of the total input and output volumes and does not necessarily equal to one. We divide all size ratios by this universal size ratio to obtain the relative local performance of G at $\hat{\rho}$, which we call the discoverability \mathfrak{d} of $\hat{\rho}$ by G :

$$\mathfrak{d}(\hat{\rho}) = \frac{r_G(\hat{\rho})}{\bar{r}_G}. \quad (\text{C.13})$$

Our method is unbiased if and only if all solutions over all distances have discoverability 1. Otherwise, a preferred dangerous global solution $\hat{\rho}'$ would exist with larger input domain such that

$$V(G^{-1}(N_{\hat{\rho}'}^\varepsilon)) > V(G^{-1}(N_{\hat{\rho}}^\varepsilon)). \quad (\text{C.14})$$

To test our results, we drew 512 successful solutions from the correlated Uranus run at random and used these as references ρ^* . To calculate the discoverability of a given ρ^* , we considered all the n results $\rho^{(i)}$ that are ε -close to ρ^* for a given ε . We then calculated the average distance between all of them:

$$\binom{n}{2}^{-1} \sum_{\substack{i,j=1 \\ i < j}}^n \|\rho^{(i)} - \rho^{(j)}\|_\infty, \quad (\text{C.15})$$

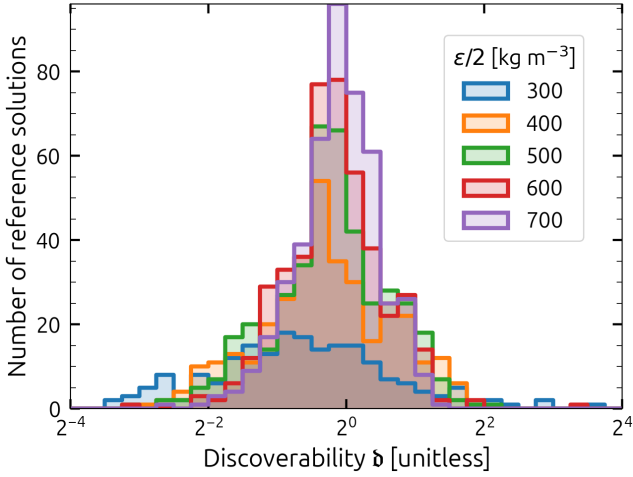


Fig. C.1: Discoverabilities. Distribution of the discoverabilities of randomly chosen reference solutions. Different colours indicate different ε distances according to equation C.3. A discoverability of $\delta = 1 = 2^0$ is equivalent to the employed gradient method being unbiased for the given reference solution.

which scales with the volume of $V(N_{\rho^*}^{\varepsilon})$ and hence was used as a proxy for it. For each reference solution ρ^* that a given ρ was close to, we added its input and output density profile to a list of neighbours of ρ^* . Due to numerical constraints, only the most recent 128 neighbours could be saved. Any additional ρ overwrote earlier neighbours and the new average distance was estimated by calculating the average distance to the most recent 128 neighbours while scaling up the result proportionally.

Figure C.1 shows the discoverability distribution of the reference solutions ρ^* for five different values of $\varepsilon/2 \in \{300, 400, 500, 600, 700\} \text{ kg m}^{-3}$. Both ends of this ε -range are motivated: At $\varepsilon/2 = 300 \text{ kg m}^{-3}$, over half of all references already had to be discarded for lack of neighbours.² At $\varepsilon/2 = 700 \text{ kg m}^{-3}$, some reference solutions had upwards of 30000 neighbours, a significant portion of the total number of runs.

Figure C.1 demonstrates that the discoverability distribution gets better (most reference solutions have $\delta \approx 1$) for bigger values of ε and loses its bias towards low discoverability. To ensure that the outliers with a discoverability δ not close to 1 are unproblematic, we compare the popularity of a solution (that is, its number of neighbours) against its discoverability. If low discoverability solutions ($\delta \ll 1$) were unpopular, but high discoverability solutions ($\delta \gg 1$) popular, our gradient method would suffer from severe bias.

Figure C.2 shows the relationship of the discoverability of the reference solution vs their popularity. We find Pearson correlation coefficients of $\{0.052, 0.137, 0.091, 0.058, 0.138\}$ for $\varepsilon/2 \in \{300, 400, 500, 600, 700\} \text{ kg m}^{-3}$. The p-values are $\{0.467, 0.011, 0.065, 0.224, 0.003\}$. The discoverabilities of the reference solution and their popularity are therefore only very weakly linearly correlated. However, the two values are clearly non-linearly correlated: The discoverability converges towards 1 as the popularity grows, our data is heteroscedastic. This behaviour is expected, as the popularity is simultaneously the sample size of the discoverability measurement which hence should converge to its true value as we increase the sample size.

² To be precise: $\{316, 166, 101, 73, 52\}$ reference profiles ρ^* had no neighbours for $\varepsilon/2 \in \{300, 400, 500, 600, 700\} \text{ kg m}^{-3}$. These were discarded because their average distances were undefined.

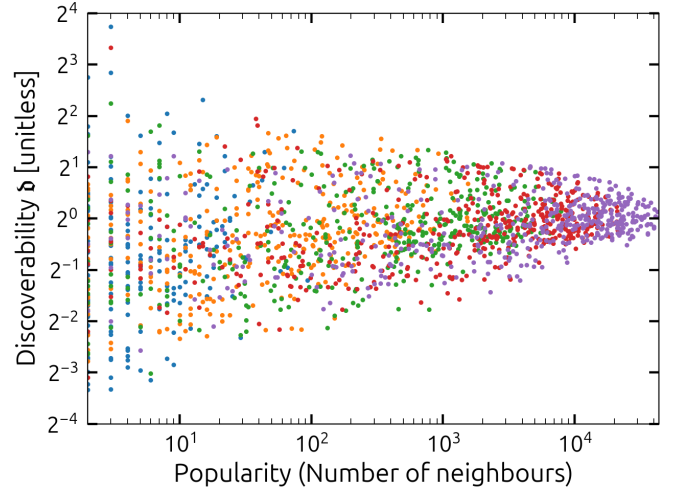


Fig. C.2: Discoverabilities vs popularities. Discoverabilities of the reference solutions (with neighbours) against their number of neighbours for different ε distances. The colours are analogous to Figure C.1.

We conclude that our gradient method is unbiased in practice based on the above presented empirical arguments. We find that for large-scale statistics (like the confidence interval), where large distances ε apply, we do not underestimate relatively infrequent solutions. However, caution must be taken in making statements about small subsections of our sample cohort. The bias there is larger and its direction unknown.

Appendix D: Discontinuities

D.1. Discontinuity detection

The following is a description of how density discontinuities are detected and categorised given a discontinuity criterion c_D . Note that the base working environment here is an array of discrete entries listing the increase from one level surface to the next.

Our task is to find every discontinuity of any arbitrary width. For this, we must first define what a discontinuity is. For that, we use the following definitions.

Definition 1 (Jump). A jump is defined as an interval of size k such that the total increase over that interval is more than $k \cdot c_D$, that is, the total average increase over this interval is more than the discontinuity criterion.

Definition 2 (Maximally steep jump). A jump is considered maximally steep if shrinking it by one on the left or right would lower its steepness or the removed entry exceeds the discontinuity criterion.

Definition 3 (Jump cluster). A jump cluster is defined as the recursive union of a jump with all other jumps that intersect it.

Note the jump cluster need not necessarily be a jump.

Definition 4 (Discontinuity). A discontinuity is defined as the union of all maximally steep jumps in a jump cluster.

A discontinuity therefore consists entirely of maximally steep jumps and is ‘minimally maximal’. It cannot be extended, since then at least one jump within the discontinuity would no longer be maximally steep. Neither can it be shortened, since

Table D.1. Same as Figure 11 but as a table for ease of reading.

(a) Uranus																				
c_D	50	100	150	200	250	300	350	400	450	500	550	600	650	700	750	800	850	900	950	1000
Avg	8	4	2	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
95%	15	8	5	4	3	3	2	2	2	2	1	1	1	1	1	1	1	1	1	1
(b) Neptune																				
c_D	50	100	150	200	250	300	350	400	450	500	550	600	650	700	750	800	850	900	950	1000
Avg	10	5	3	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
95%	20	11	7	5	4	4	3	3	2	2	2	2	2	1	1	1	1	1	1	1

Notes. Table of the number of discontinuities detected per successful density profile over some selection criteria c_D (in units of $\Delta\ell^{-1} \text{ kg m}^{-3}$), average (rounded to the nearest integer) and upper 95th percentile.

then either its steepness would be lowered or sections above the discontinuity criterion would be removed.

We provide some illustrative examples for $c_D = 100$. We list the increases, not the values directly, and the first listed value has index 0. The brackets denote a jump and the numbers within the brackets denote the indices. For example [1,1] denotes a jump of size one with a height of the value with index 1. [1,3] denotes a jump of size three with a height equal to the sum of the values with indices 1, 2, and 3.

- ..., 0, 101, 99, 101, 0, ... corresponds to one discontinuity at [1,3] because [1, 1], [3, 3], and [1, 3] are maximally steep jumps.
- ..., 0, 100, 0, 100, 0, ... corresponds to two discontinuities because [1, 3] is not a jump.
- ..., 0, 100, 200, 100, 0, ... corresponds to one discontinuity at [1,3].
- ..., 0, 99, 200, 99, 0, ... corresponds to one discontinuity at [2,2] because the jumps [1, 2], [2,3], and [1,3] are not maximally steep.
- ..., 0, 100, 0, 200, 0, 100, 0, ... corresponds to one discontinuity at [1,5] because [1,3] and [3,5] are maximally steep jumps, resulting in a cluster spanning [1,5].
- ..., 0, 100, 0, 100, 0, 100, 0, ... corresponds to three discontinuities.

Regarding the numerical detection of a discontinuity, one could iterate over all interval sizes and find all corresponding jumps and then find their largest intersection of maximally steep jumps for each cluster. However, this method is very slow. We thus use an approximate solution using a convolution (sliding window). An averaging sliding window (for example [0.25, 0.25, 0.25, 0.25]), would only guarantee to find discontinuities of the size of the window. This is because smaller discontinuities would get squashed (in our example by the factor 0.25) and larger discontinuities could erroneously be split. For example ..., 125, 125, 0, 125, 125, ... should be one discontinuity but would get identified as two with [0.5, 0.5] being the sliding window. A constant sliding window (for example [1, 1, 1, 1]) is also not usable, since it would misidentify increases consistently below the density criterion as discontinuities, especially at large window sizes. Therefore, we used a sliding window with varying values: We chose a window size of $n = 9$ and want to guarantee that we will find all discontinuities of any size up to $n = 9$. The window hence must contain one entry ≥ 1 to find discontinuities of length one, two entries ≥ 0.5 to find discontinuities of length two, and so on. $n = 9$ requires nine entries $\geq 1/9$ which ultimately yields [1/8, 1/6, 1/4, 1/2, 1, 1/3, 1/5, 1/7, 1/9] as a sliding window. Since it is highly likely that discontinuities larger

than this window will have subsections within this window that are captured, this essentially guarantees all discontinuities are found. In summary, this sliding window ensures that a discontinuity even of only size 1 will certainly at some point be detected by exceeding the discontinuity criterion. The same holds true for a discontinuity spread over nine steps. However, this method is still too generous for the same reason as the above problem with the constant sliding window of [1, 1, 1, 1]. Therefore, some post-processing has to be done, where we iteratively cut potential ‘discontinuities’ on both sides as long as the smaller ‘discontinuity’ has better steepness and as long as the cut itself is not steeper than the discontinuity criterion. If the final resulting ‘discontinuity’ has a total height $\geq k \cdot c_D$, we have successfully detected a true discontinuity.

D.2. Further results

We show the values of Figure 11 in Table D.1. Figures D.1 (Uranus) and D.2 (Neptune) separately show the average discontinuity height $\overline{\Delta\rho}$ and width $\overline{\Delta r}$ combined in Figure 9 at their respective average normalised radii. Note that profiles that have no discontinuity at any given location also contribute to the averages with value zero. Artefacts from the starting model generation (see Section 4) are clearly visible in Figure D.2. Finally, we show the ‘raw’ average density increase for all successful profiles in Figure D.3. This Figure differs from the left panels of Figures D.1 and D.2 because we do not show the average discontinuity height, but rather the general average density increase as a function of normalised radius for all successful density profiles. However, increases below the discontinuity criterion still only contribute with value zero to the average.

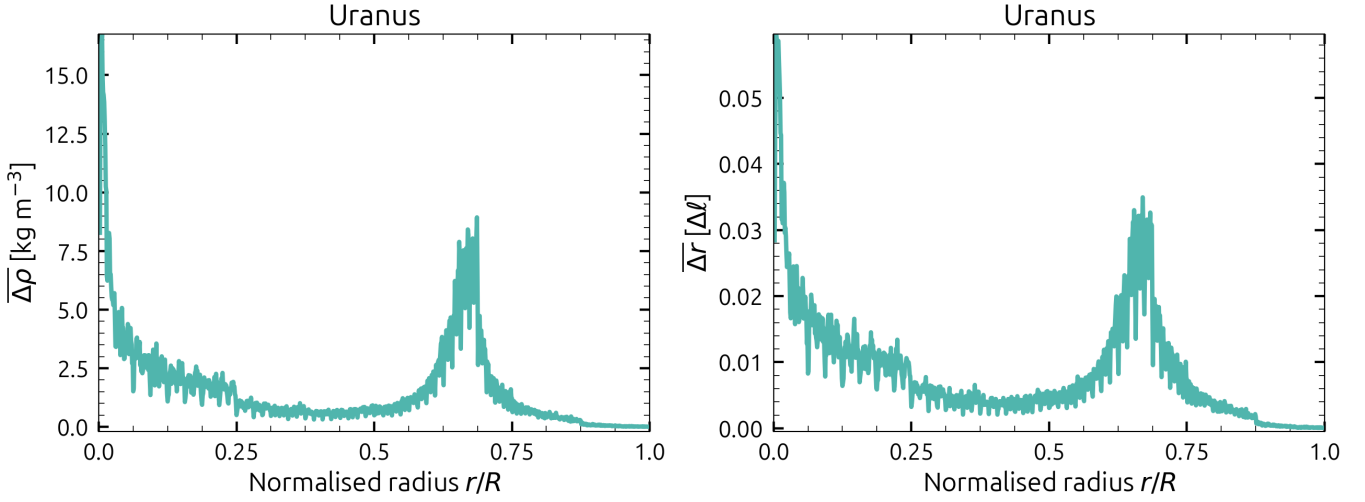


Fig. D.1: Average discontinuity heights (left) and widths (right) of successful density profiles. For example, discontinuities with an average location of $r/R = 0.5$ possess an average width of $\Delta r \sim 0.005\Delta\ell$. Profiles with no discontinuities also contribute to the averages with value zero.

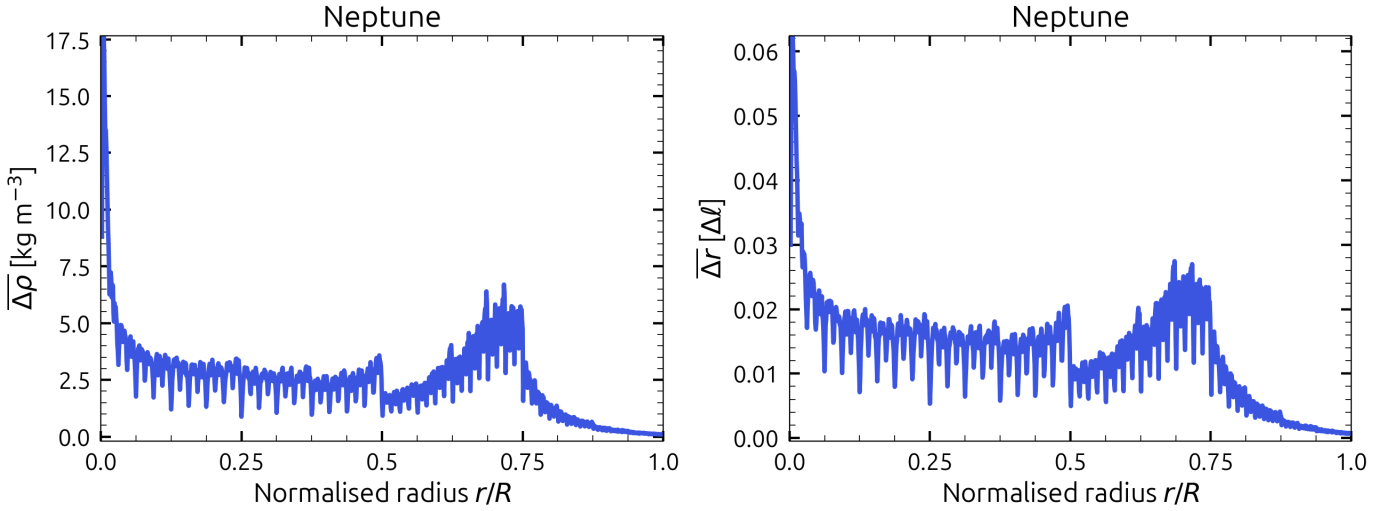


Fig. D.2: Same as Figure D.1, but for Neptune.

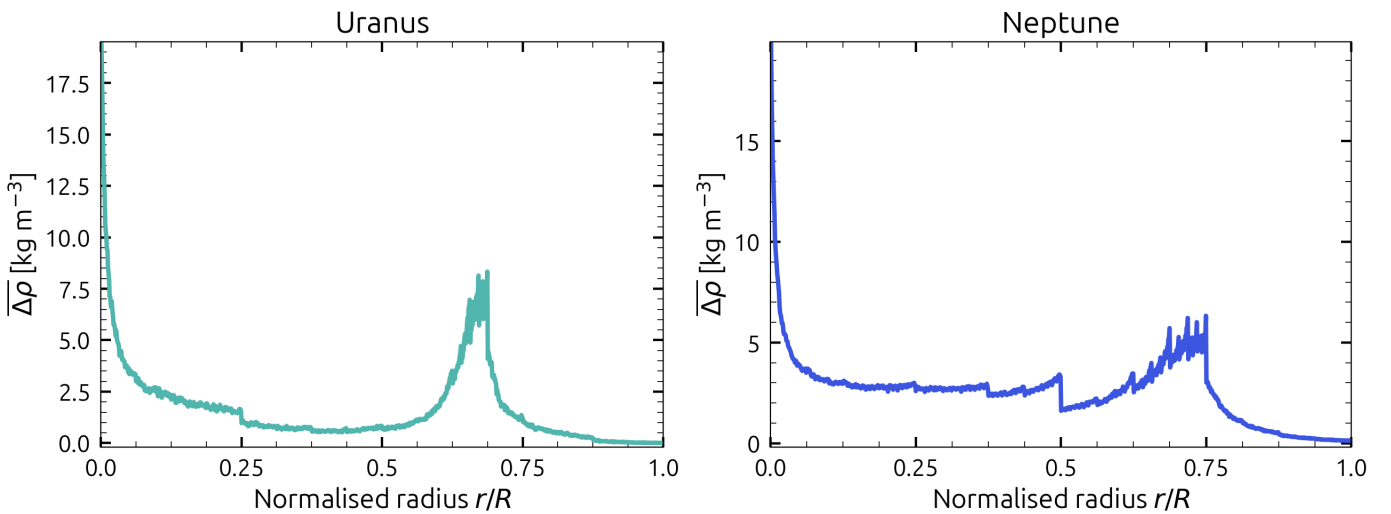


Fig. D.3: Average density increase as a function normalised radius for all successful density profiles. Increases below the discontinuity criterion contribute with value zero to the average.