Ultra-fast Model Emulation with PRISM: Analyzing the Meraxes Galaxy Formation Model

Ellert van der Velden1,2, Alan R. Duffy1,2, Darren Croton1,2, and Simon J. Mutch2,3

1 Centre for Astrophysics and Supercomputing, Swinburne University of Technology, PO Box 218, Hawthorn, VIC 3122, Australia; evandervelden@swin.edu.au
2 ARC Centre of Excellence for All Sky Astrophysics in 3 Dimensions (ASTRO 3D)
3 School of Physics, University of Melbourne, Parkville, VIC 3010, Australia

Received 2020 November 22; revised 2021 January 4; accepted 2021 January 18; published 2021 April 1

1. Introduction

Recent years have seen the establishment of a concordance cosmological model, called ΛCDM, based on decades of increasingly sophisticated observations, with the universe being made up of ~75% dark energy and ~25% mass (dark matter, DM; and baryonic matter) (Planck Collaboration et al. 2016). Within the ΛCDM paradigm, a key transition occurs in the early universe, when sufficient ionizing photons have been produced to reionize the majority of the volume. This transition is called the Epoch of Reionization (EoR; Loeb & Barkana 2001). Studying the EoR is quite challenging because only a limited number of observations is available due to the very high redshifts at which this event occurs, and because the wall of neutral hydrogen is opaque along the line of sight. Also, the formation of large-scale structures during the EoR is affected by small-scale physics such as star formation, making modeling the EoR rather complex. In order to attempt to explain the EoR with galaxy formation, three different main types of approaches are used: N-body simulations, hydrodynamic simulations, and semianalytic models.

Following the descriptions given in Lacey (2001), Benson (2010), and Kuhlen et al. (2012), an N-body simulation models the physics of large-scale galaxy structure by directly evolving a large number of particles solely based on the fundamental equations of gravitation. Such simulations are very successful (like the Millennium simulation; Springel et al. 2005) because DM particles are known to only interact gravitationally and make up the majority of the total mass, leaving baryonic matter with only a small contribution.

Hydrodynamic simulations (e.g., Schaye et al. 2010, 2015; Genel et al. 2014) involve, as the name already suggests, explicit calculations of the hydrodynamic forces together with gravity. The inclusion of hydrodynamics means that baryons play a more explicit role in structure formation, especially at small scales because baryons can radiatively lose energy and thereby collapse to higher densities. Hence, to follow their evolution, the required resolution timescales are much smaller and are also more complex to simulate in general. Therefore hydrodynamic simulations tend to be much more computationally expensive than N-body simulations.

Finally, semianalytic models (SAMs; e.g., Cole et al. 2000; Croton et al. 2006, 2016; Henriques et al. 2013; Somerville et al. 2015; Mutch et al. 2016; Qin et al. 2017; Lagos et al. 2018) use approximations of the involved physics to model galaxy formation. SAMs are commonly applied to a static halo merger tree that is obtained from an N-body simulation, and the galaxy formation physics is treated in a phenomenological way. In practice, this means that there are typically far more parameters in the model that can be tuned than in hydrodynamic simulations, but SAMs are also far less computationally expensive. This allows SAMs to explore different evolution tracks much quicker, providing broader insight into the underlying physics. The disadvantage, however, is that a higher degree of approximation is used, which can give rise to unforeseen artifacts and unknown variances. In order to find out to what extent this has an effect on the accuracy and reliability of a SAM, the model needs to be thoroughly analyzed. While this paper explores a means to efficiently and thoroughly explore SAMs, we note that the technique could be similarly valuable in hydrodynamic simulations and their parameterizations.

Commonly, a combination of Markov Chain Monte Carlo (MCMC) methods and Bayesian statistics (e.g., Sivia & Skilling 2012; Gelman et al. 2014) is employed when
performing model parameter estimations, as performed by Henriques et al. (2013, 2019) and Martindale et al. (2017) for galaxy formation SAMs. This is used to obtain an accurate approximation of the posterior probability distribution function (PDF), which allows for regions in parameter space that compare well to the available data to be identified. As a consequence, MCMC and Bayesian statistics tend to be used for analyzing (exploring the behavior of) a model as well, even though obtaining the posterior PDF is an inherently slow process. However, Van der Velden et al. (2019), hereafter V19, argued that an accurate approximation of the posterior PDF is not required for analyzing a model, but the converging process toward the PDF is required instead. V19 therefore proposed the combination of the Bayes linear approach (Goldstein 1999; Goldstein & Wilkinson 2000; Goldstein & Wooff 2007) and the emulation technique (Sacks et al. 1989; Curri et al. 1991; Oakley & O’Hagan 2002; O’Hagan 2006) with history matching (Raftery et al. 1995; Craig et al. 1996, 1997; Kennedy & O’Hagan 2001; Goldstein & Rougier 2006) as an alternative. A combination of these techniques has been applied to galaxy formation SAMs before (Bower et al. 2010; Vernon et al. 2010, 2014; Rodrigues et al. 2017), showing their value.

In this work, we use an open-source PYTHON implementation of these techniques, called PRISM (Van der Velden et al. 2019; Van der Velden 2019), to explore one such semianalytic model: MERAXES (Mutch et al. 2016). Unlike other SAMs such as SAGE (Croton et al. 2016) and SHARK (Lagos et al. 2018), MERAXES walks through the aforementioned static halo trees of an N-body simulation horizontally instead of vertically. This means that for each snapshot (time instance), all halos at that snapshot within all trees are processed simultaneously, such that interactions between the different trees can be taken into account. For MERAXES, this is necessary, as it was purposefully built to self-consistently couple the galaxy formation to the reionization history at high redshifts ($z \geq 5$). This makes the model suitable to obtain more insight into reionization, but it also makes it much slower to explore because the coupling mechanism requires many additional calculations for each snapshot. It is therefore crucial that we use a framework such as PRISM to efficiently explore and analyze MERAXES.

This work is structured in the following way: In Section 2 we give a short overview of how the Bayes linear approach, emulation technique, and history matching can be used to analyze models. Then, using the PRISM framework, we analyze the MERAXES model and discuss our findings in Section 3. After this, we explore a modified version of MERAXES by Qiu et al. (2019) in Section 4, and compare our results with theirs. Finally, in Section 5 we give a summary of the results in this work and discuss the potential implications they may have.

2. Model Analysis and Emulation

In this section, we give a short overview of how emulation in PRISM works and how it can be used to analyze scientific models efficiently. Note that this section is meant for those seeking a basic understanding of the methodology and terminology used throughout this work. We refer to Goldstein & Wooff (2007) and Vernon et al. (2010) for further details on the Bayes linear approach, emulation technique, and history matching, or to V19 for their specific use in PRISM.

The basics of the emulation technique is as follows. Suppose that we have a model that takes a vector of input parameters $\mathbf{x}$ and whose output is given by the hypothetical function $f(\mathbf{x})$. As $f(\mathbf{x})$ is both continuous and defined over a closed interval, we can apply the Stone–Weierstrass theorem (Stone 1948), which states that such a function can be uniformly approximated by a polynomial function as closely as desired. Using this, we can say that the output $i$ of $f(\mathbf{x})$, $f_i(\mathbf{x})$, is given by

$$f_i(\mathbf{x}) = \sum_j \beta_{ij} g_j(\mathbf{x}_{A,i}) + u_i(\mathbf{x}_{A,i}) + w_i(\mathbf{x}), \quad (1)$$

with $\beta_{ij}$ the unknown scalars, $g_j(\mathbf{x})$ the deterministic functions of $\mathbf{x}$, $u_i(\mathbf{x}_{A,i})$ a weakly stochastic process with constant variance, and $w_i(\mathbf{x})$ all remaining variance. The vector $\mathbf{x}_{A,i}$ represents the vector of active input arguments for output $i$; those values that are considered to have a non-negligible impact on the output.

The goal of an emulator is to find out what the form of this function $f(\mathbf{x})$ is for a given model. However, as model evaluations tend to be complex and very time-consuming, we would like to avoid evaluating the model millions of times. To solve this problem, we use the Bayes linear approach, which can be seen as an approximation of a full Bayesian analysis that uses expectations instead of probabilities as its main output. By using the Bayes linear approach, for a given vector of known model realizations $D_i = \{f_1(\mathbf{x}_{1,i}), f_2(\mathbf{x}_{2,i}), \ldots, f_n(\mathbf{x}_{n,i})\}$ we can calculate what the adjusted expectation and adjusted variance values are for a given output $i$.

$$E_{D_i}(f_i(\mathbf{x})) = E(f_i(\mathbf{x})) + Cov(f_i(\mathbf{x}), D_i) \cdot Var(D_i)^{-1} \cdot (D_i - E(D_i)), \quad (2)$$

$$Var_{D_i}(f_i(\mathbf{x})) = Var(f_i(\mathbf{x})) - Cov(f_i(\mathbf{x}), D_i) \cdot Var(D_i)^{-1} \cdot Cov(D_i, f_i(\mathbf{x})), \quad (3)$$

with $E(f_i(\mathbf{x})) = \sum_j E(\beta_{ij}) g_j(\mathbf{x}_{A,i})$ the prior expectation of $f_i(\mathbf{x})$, $Cov(f_i(\mathbf{x}), D_i)$ the vector of covariances between the unknown output $f_i(\mathbf{x})$ and all known outputs $D_i$, and $Var(D_i)$ the $n \times n$ matrix of covariances between all known outputs with elements $Var_{ij}(D) = Cov(f_i(\mathbf{x}_{1,i}), f_j(\mathbf{x}_{2,i})).$

Equations (2) and (3) can be used to update our beliefs about the function $f_i(\mathbf{x})$ given a vector of known outputs $D_i$. With our updated beliefs, we can determine the collection $\lambda^*$, which contains those input parameters $\mathbf{x}$ that give an acceptable match to the observations $z^*$ when evaluated in $f(\mathbf{x})$, including the unknown best vector of input parameters $\mathbf{x}^*$. The process of obtaining this collection $\lambda^*$ is called history matching, which can be seen as the equivalent to model calibration when performing a full Bayesian analysis. History matching is achieved through evaluating functions called implausibility measures (Craig et al. 1996, 1997), which have the following form:

$$f_i^2(\mathbf{x}) = \frac{(E_{D_i}(f_i(\mathbf{x})) - z_i)^2}{Var_{D_i}(f_i(\mathbf{x})) + Var(\epsilon_{md,i}) + Var(\epsilon_{obs,i})}. \quad (4)$$

with $E_{D_i}(f_i(\mathbf{x}))$ the adjusted emulator expectation (Equation (2)), $Var_{D_i}(f_i(\mathbf{x}))$ the adjusted emulator variance (Equation (3)), $Var(\epsilon_{md,i})$ the model discrepancy variance, and $Var(\epsilon_{obs,i})$ the observational variance. Here, the model discrepancy variance (Kennedy & O’Hagan 2001; Vernon et al. 2010) is a measure of

---

4 Not to be confused with redshift.
the (un)certainty that the output of the model is correct, given its intrinsic implementation and functionality (e.g., approximations, stochasticity, and missing features) As this variance can be rather challenging to properly calculate, it is usually estimated using a justified assumption.

For a given input parameter set \( x \), the corresponding implausibility value \( I(x) \) tells us how (un)likely it is that we would view the match between the model output \( f(x) \) and the observational data \( z \) as acceptable or plausible in terms of the standard deviation \( \sigma \). The higher the implausibility value, the more unlikely it is that we would consider \( x \) to be part of the collection \( \Lambda^* \). Because the implausibility measure is both unimodal and continuous, we can use the 3\( \sigma \) rule given by Pukelsheim (1994) to show that 95\% of its probability must lie within \( \pm 3\sigma (f(x) \leq 3) \). Values higher than 3 would usually mean that the proposed input parameter set \( x \) should be discarded, but we show later in this work that this is not always necessary in order to heavily reduce parameter space.

To show how the theory above can be applied to a model, we have created an emulator of a simple Gaussian function, given in Figure 1(a). On the left in the figure, we show the model output function \( f(x) \) as the dashed line, which is usually not known but shown here for convenience. This model has been evaluated a total of five times, represented by the black dots. Using these evaluations, we have created an emulator using Equations (2) and (3), given by the solid line and the shaded area. This shaded area shows the 3\( \sigma \) confidence interval, with \( \sigma = \sqrt{\text{Var}(f(x))} \). Finally, we added a single comparison data point, given as the horizontal line with its 2\( \sigma \) confidence interval.

Using all of this information, we can calculate what the implausibility values are for all values of \( x \) using Equation (4), which is shown on the right in Figure 1(a). If we state that we consider values of \( I(x) \leq 3 \) to be acceptable, as indicated by the dashed line, then we can see that there are only a few regions in parameter space that satisfy this condition and thus require further analysis. After evaluating the model six more times in these regions and updating our beliefs, we obtain the plots in Figure 1(b). We can now see that the emulator solely focused on the important parts of parameter space, where it has been greatly improved, and that the implausibility values are only acceptable for \( x = 1.5 \) and \( x = 2.5 \), which is as expected.

When making emulators of models that are more complex than the one shown in Figure 1, it is very likely that the model has more than a single output (and thus multiple comparison data points). In this scenario, every model output has its own defined implausibility measure, which are evaluated and combined together. In order for an input parameter set \( x \) to be considered plausible, all implausibility measures must meet a specific criterion. This criterion, called the implausibility curve, is given by

\[
I_{\max,n}(x) \leq I_{\text{cut,n}},
\]

with \( I_{\text{cut,n}} \) being the maximum value that the \( n \)th highest implausibility value \( I_{\max,n}(x) \) is allowed to have for a given \( x \).

In cases where there are many model outputs, it can sometimes be desirable to ignore the first few highest implausibility measures regardless of their values. In this scenario, we apply so-called implausibility wildcards to the emulator, which allows us to analyze a model in a slower fashion by ignoring as many implausibility measures as there are wildcards. As a higher implausibility value implies a more constraining data point, this effectively removes the heavier constraints from the analysis.

By performing history matching iteratively, a process called refocusing, we can remove parts of parameter space based on the implausibility values of evaluated input parameter sets. This in turn leads to a smaller parameter space to evaluate the model in, and thus a higher density of parameter sets to update our beliefs with. As this leads to a more accurate emulator, we can progressively make a better approximation of the model with each iteration.

The concepts discussed in this section allow the PRISM framework to quickly make accurate approximations of models in those parts of parameter space that are important, and
### Table 1  
The Parameters Used by PRISM to Calculate the SMF in MERAXES

| Name                      | Min | Max | Estimate |
|---------------------------|-----|-----|----------|
| MergerBurstFactor         | 0.0 | 1.0 | 0.57     |
| SfCriticalSDNorm          | 0.0 | 1.0 | 0.2      |
| SfEfficiency              | 0.0 | 1.0 | 0.03     |
| SnEjectionEff             | 0.0 | 1.0 | 0.5      |
| SnEjectionNorm            | 0.0 | 100.0 | 70.0   |
| SnEjectionScaling         | 0.0 | 10.0 | 2.0     |
| SnReheatEff               | 0.0 | 10.0 | 6.0     |
| SnReheatNorm              | 0.0 | 100.0 | 70.0   |
| SnReheatScaling           | 0.0 | 10.0 | 0.0     |

*Note*. All other (potentially important) parameters were set to their default values at all times. The boundaries (where possible) and estimates were obtained from Table 1 in M16, with missing boundary values being chosen within reason. The names in the first column indicate the name that these parameters have in MERAXES.

Table 2  
Statistics for the MERAXES Emulator with Mock Data

| f_{\text{eval}} | n_{\text{eval}} | f_{\text{true}} | n_{\text{wild}} | f_{\text{space}} |
|-----------------|-----------------|-----------------|-----------------|-----------------|
| 1               | 500             | [4.0, 3.5, 3.2, 3.0] | 3               | 1.32%           |
| 2               | 712             | [4.0, 3.5, 3.2, 3.0] | 2               | 0.144%          |
| 3               | 1036            | [4.0, 3.5, 3.2, 3.0] | 1               | 0.0748%         |
| 4               | 808             | [4.0, 3.5, 3.2, 3.0] | 2               | 0.0262%         |

*Note*. The first column specifies the emulator iteration \( f_{\text{eval}} \), that this row is about. The next three columns provide the number of model evaluations \( n_{\text{eval}} \), the non-wildcard implausibility cutoffs \( f_{\text{true}} \), and the number of implausibility wildcards \( n_{\text{wild}} \) used for this emulator iteration. Finally, the last column gives the fraction of parameter space remaining \( f_{\text{space}} \) after this emulator iteration was analyzed. Therefore, all three of these parameters directly influence the stellar mass of a galaxy or halo. On the other hand, the SnEjection... and SnReheat... parameters each are a group of three free parameters that determine a single physical quantity in MERAXES: the supernova energy coupling efficiency and the mass loading factor, respectively. Since the supernova feedback heavily influences the amount of (cold) gas that is available for star formation, they are very important for the model predictions.

### 3. Analyzing the Meraxes Model

In this section, we discuss the results of our analysis of the MERAXES galaxy formation model by Mutch et al. (2016), hereafter M16. These results are based on two emulators made with PRISM. The emulators only differ in the constraining observational data that was used, and are described below.

The emulator uses a MODELLINK subclass (a PYTHON class that wraps a model such that PRISM can interface with it) that maps the stellar mass function (SMF) number density values from its logarithmic scale to an arctan function, while also mapping the associated SMF data errors accordingly as well. This results in all SMF data values to be in the range \([-0.5, 0.5] = [-1.571, 1.571]\), which was used for all subsequent emulators in this work as well. The reason we use an arctan function here is to limit the range of orders of magnitude for the data values, decreasing the possibility of artifacts due to floating point errors in the emulation process. The MODELLINK subclass uses the SMF parameter values as described in Table 1 in M16 with appropriate ranges, which are given in Table 1. We used the SMFs presented in Song et al. (2016) at redshifts \( z = [7, 6, 5] \) to constrain the parameters for a combined total of 24 data points. Note that because of this, in the following, whenever we refer to the SMF, we specifically mean the SMF at these redshifts. As the MERAXES model is expected to be reasonably well constrained in this redshift range, we assumed a static model discrepancy variance \( \text{Var}(\epsilon_{\text{md},i}) \) of \( 10^{-4} \) for all data points in this section.

The nine parameters shown in Table 1 can be subdivided into two different groups, namely those related to the star formation (MergerBurstFactor, SfCriticalSDNorm, and SfEfficiency) and the supernova feedback (SnEjection... and SnReheat...). The star formation parameters determine the star formation burst that happens after a galaxy or halo merger (MergerBurstFactor), the critical surface density required for star formation (SfCriticalSDNorm), and the efficiency with which available cold gas is converted into stars (SfEfficiency).
per grid point that is independent of the non-plotted parameters. In every projection figure, the top subplot shows the minimum implausibility value (i.e., \( \min(i_{\text{max}}) \) with \( n \) the first non-wildcard cutoff) that can be reached, whereas the bottom subplot shows the fraction of samples (line-of-sight depth) that is plausible. An easy way to distinguish these plots is that the former shows where the best plausible samples can be found, while the latter shows where the most plausible samples are. A 2D projection is similar to a 3D projection, but only a single model parameter is plotted, and is more comparable to a marginalized likelihood/density plot often used in Bayesian statistics.

A combination of 2D and 3D projections can be used to study many properties of the MERAXES model. On the left in Figure 2, we show the 2D projections of the three star formation parameters. These three projections show us quite clearly where the best parameter values can be found, especially for the star formation efficiency \( SfEfficiency \), which all agree very well with their parameter estimates (the dashed lines). Although we know that the MERAXES model is capable of generating a model realization that corresponds to the data, as mock data was used instead of observational data, this result is still somewhat surprising. After all, almost the entire parameter range is still considered to be plausible for the \( \text{MergerBurstFactor} \) and \( SfCriticalSDNorm \) parameters, while the favored parameter values are very obvious. This shows that while PRISM is quite conservative with removing plausible space despite low variances, it can still reach a result quickly, only requiring 3056 model evaluations to reduce plausible parameter space by almost a factor of 4000.

On the right of Figure 2, we show the 3D projections of the free parameters related to the supernova feedback. As each of these supernova feedback parameters is comprised of three free parameters, we expect the free parameters to be heavily

---

**Figure 2.** Projection figures of the MERAXES emulator with mock data at \( i_{\text{emul}} = 4 \). The dashed lines show the estimated value of the corresponding parameter as given in Table 1. Left: three 2D projection figures showing the behavior of the parameters related to the star formation. Right: three 3D projection figures showing the behavior of and correlations between the parameters related to the supernova feedback.

---
correlated with each other. As demonstrated by these figures, this is indeed the case. While PRISM can come to a good estimate of where the best parameter values are, it cannot pinpoint them nearly as easily as the star formation parameters. This would either require more model evaluations or more data points, where the latter would hopefully provide more constraining information for these free parameters.

### 3.2. Investigating the Meraxes Model Calibration

Now that we have established that PRISM is indeed capable of retrieving the proper parameters with reasonable accuracy and a low number of model evaluations, we use the observational data from Song et al. (2016). Except for the different comparison data values, we kept all other variables, such as the implausibility cutoffs and model discrepancy variance, as similar as possible. This allows us to compare the results of this emulator with the mock data one we studied previously. The statistics of this emulator are shown in Table 3.

When comparing the emulator statistics given in Table 3 with those in Table 2, we can see that both emulators behaved very similarly in iterations 1 and 2. However, because we noted that the mock data emulator barely changed when we did not decrease the number of implausibility wildcards, we decided to use the same implausibility parameters for iteration 3 as we did for iteration 4 in Table 2. However, unlike the mock data emulator, this emulator vastly reduced its plausible parameter space, doing so by more than a factor 50. This can be an indication that either there are a select few data points that constrain the model heavily (as they would be ignored when using more implausibility wildcards), or that the model cannot produce any realizations that can explain the data reasonably well. Given that the former trend was not observed in the mock data emulator, even though it would apply there as well, we expect that the latter plays a role here.

In Figure 3 we show the projection figures of this emulator for iteration 2. We show the iteration 2 projections here instead of those of iteration 3 because the latter show much less structure than the former. This is another indication that the MERAXES model might be incapable of producing realizations that fit the data, as a lack of structure is usually caused by no areas of parameter space being favored over others, thus creating a tiny hypercube shell of plausible samples.

From the iteration 2 projections, we can see that the star formation parameters, specifically the MergerBurstFactor and SfCriticalSDNorm, do not agree with their own estimates. Instead, it seems that they prefer much lower values than their estimates indicate. The star formation efficiency SfEfficiency does agree with its estimate, but not nearly with the same confidence as it did in Figure 2. We kept in mind here that this specific emulator iteration has a plausible space that is five times larger than the aforementioned emulator’s plausible space, but the plausible samples are still much more spread out. This is especially evident when comparing the 2D projection for SfEfficiency in both figures with each other.

However, when we examine the 3D projections in Figure 3, we can clearly see that the range of plausible values is very large. The minimum implausibility plots show barely any structure (which decreases even further for iteration 3, which is why we do not show these figures here), unlike the corresponding projections in Figure 2. Furthermore, the parameters are not at all consistent with their estimates. Because we noted earlier that the supernova feedback parameters used here are in fact groups of three free parameters that combined determine a single physical process, this might be an indication that the estimates are significantly perturbed due to correlation errors.

Using the results that we have collected thus far, mainly that the parameters do not converge, we conclude that the manually calibrated parameter values, as given by M16 and shown in Table 1, are likely to be biased and thus non-optimal. This conclusion is indeed implied by M16, whose caption of the parameter estimates in their Table 1 states that the “values were constrained to visually reproduce the observed evolution in the galaxy stellar mass function between $z=5$ and $z=7$.” Furthermore, even though we have shown that PRISM is very capable of constraining MERAXES quickly, the free parameters for the energy coupling efficiency $\text{SnEfficiency}$... and the mass loading factor $\text{SnReheat}...$ are very hard to constrain properly using the SMF at $z=[7, 6, 5]$, as shown by the 3D projections in Figure 2. Qiu et al. (2019) came to a similar conclusion and decided to make some modifications to the MERAXES model and use luminosity data in order to attempt to constrain the star formation and supernova feedback parameters. In the next section, we use their modified MERAXES model with PRISM to see whether this resolves the problem, and discover what else we can learn from its analysis.

### 4. Studying a Modified Meraxes

In this section, we discuss the results of the emulators that were made using the methodology and parameter values for MERAXES as reported in Qiu et al. (2019), hereafter Q19. In the following, to avoid confusion, we refer to the two different versions of MERAXES using the corresponding work in which they are described (e.g., M16 MERAXES and Q19 MERAXES), or simply refer to it as “MERAXES” when we describe the model in general.

The Q19 MERAXES model is substantially modified from the M16 MERAXES model we explored in Section 3. The most significant difference between the two MERAXES models is that the original (M16) uses three free parameters to calculate the mass loading factor and energy coupling efficiency (given by Equations (14) and (13) in M16, respectively), whereas the modified version (Q19) only uses a single free parameter for both quantities (Equations (9) and (10) in Q19, respectively). Not only does this reduce the number of parameters that need to be constrained by four (five, when the MergerBurstFactor is counted as well) compared to the previous emulators, it also removes many correlation patterns between all the parameters.

| $i_{\text{emul}}$ | $n_{\text{eval}}$ | $i_{\text{cut,n}}$ | $f_{\text{space}}$ |
|------------------|------------------|------------------|------------------|
| 1                | 500              | [4.0, 3.5, 3.2, 3.0] | 3              | 2.29%            |
| 2                | 1189             | [4.0, 3.5, 3.2, 3.0] | 2              | 0.149%           |
| 3                | 878              | [4.0, 3.5, 3.2, 3.0] | 1              | 0.00270%         |

Note: The first column specifies the emulator iteration $i_{\text{emul}}$ that this row is about. The next three columns provide the number of model evaluations $n_{\text{eval}}$, the non-wildcard implausibility cutoffs $i_{\text{cut,n}}$, and the number of implausibility wildcards $n_{\text{wild}}$ used for this emulator iteration. Finally, the last column gives the fraction of parameter space remaining $f_{\text{space}}$ after this emulator iteration was analyzed.
Furthermore, the Q19 MERAXES model uses UV luminosity function (LF) and color–magnitude relation (CMR) data from Bouwens et al. (2014, 2015) as observational constraints, not the SMF data. To be able to handle this, Q19 implemented three different parameterizations of the dust model proposed by Charlot & Fall (2000) in MERAXES. We chose to use their dust-to-gas ratio (DTG) parameterization in this work as it seemed to be the least constrained, giving the emulation process more freedom. Despite this, the priors for all parameters as reported in Q19 are still much more heavily constrained than the priors we used earlier (as given in Table 1).

Finally, the initial mass function (IMF) used in Q19 MERAXES (Kroupa; Kroupa 2002) is different from the one used in M16 MERAXES (Salpeter; Salpeter 1955). Due to this change in the used IMF, the SMF data from Song et al. (2016) needs to be adjusted. According to Equation (2) in Speagle et al. (2014), the SMF masses can be shifted from a Salpeter IMF to a Kroupa IMF by applying the following offset:

$$M_{*,K} = 0.62M_{*,S},$$

with the K and S subscripts referring to the Kroupa and Salpeter IMFs, respectively. This translates into an offset of $-0.21$ dex for logarithmic masses. For the emulators reported in this section, we have applied this offset to the SMF data from Song et al. (2016).

Since the Q19 MERAXES model uses different parameters than the original M16 MERAXES, the parameter ranges and estimates given in Table 1 are no longer correct. Therefore we opted for using the same ranges and estimates for both the SMF and dust optical depth parameters, as described in Table 2 in Q19, which are given in Table 4. For the data, we used the same data as used in M16 and Q19, which is the observational data given by
The parameter ranges and estimates in Q19 were calibrated for them. The statistics of this emulator are shown in Table 5.

| Name                | Min  | Max  | Estimate |
|---------------------|------|------|----------|
| SfCriticalISDNorm   | 0.001| 0.25 | 0.01     |
| SfEfficiency        | 0.05 | 0.18 | 0.1      |
| SnEjectionEff       | 0.8  | 2.2  | 1.5      |
| SnReheatEff         | 2.0  | 15.0 | 7.0      |
| a                   | 0.1  | 0.65 | 0.34     |
| n                   | −2.5 | −0.8 | −1.6     |
| s1                  | 0.4  | 2.2  | 1.2      |
| tauBC               | 0.0  | 1000.0 | 381.3   |
| tauISM              | 0.0  | 50.0 | 13.5     |

Note. All other (potentially important) parameters were set to their default values at all times. The boundaries and estimates were obtained from Table 2 in Q19. The names in the first column indicate the name that these parameters have in Q19 MERAXES.

Song et al. (2016; SMF), adjusted for a Kroupa IMF (Kroupa 2002) and Bouwens et al. (2014, 2015) (LF/CMR), respectively. The parameter ranges and estimates in Q19 were calibrated manually by repeatedly using an MCMC estimator, which we kept in mind during the analysis.

4.1. Comparing the Two Meraxes Models

As the number of free parameters used for the mass loading factor and energy coupling efficiency in Q19 MERAXES was reduced to one and the used IMF was changed, we first decided to make an emulator using SMF data alone. We had concluded earlier that M16 MERAXES was very difficult to constrain properly using only SMF data at $z = [7, 6, 5]$, and we wished to determine whether the parameter reduction and/or change in the IMF has any positive effects on the convergence behavior of the parameter space. For this emulator, we assumed a model discrepancy variance $\text{Var}(\epsilon_{\text{mod}})$ set to $(z_i/100)^2$, with $z_i$ being the value of the corresponding observational data point. This results in values that are very similar to the static $10^{-4}$ we used earlier (as the SMF data values are mapped to an arctan function and thus are on the order of unity), but makes them dependent on the data value. We did this in order to prepare for the LF/CMR data points we use later, as their values are on the order of $10^{-4}$ and thus such a model discrepancy variance would not work well for them. The statistics of this emulator are shown in Table 5.

Looking at Table 5, we can see that the parameters in this emulator are constrained much more heavily than in previous emulators. As a reminder, every implausibility wildcard causes the next highest implausibility measure to be ignored when evaluating the emulator. Despite using no wildcards at all, this emulator still had 19.6% of parameter space remaining in iteration 1. This may seem like much compared to the statistics of previous emulators (that do use wildcards), but this can be easily explained by noting that the parameter ranges are far smaller than those used before. Therefore only a potentially more interesting or plausible space was available to begin with.

We also note that there is a large reduction in plausible space between iterations 1 and 2, corresponding to a factor of about 100. Given that the implausibility cutoffs $I_{\text{cut,n}}$ were not reduced by a large amount for iteration 2 and neither had any implausibility wildcards involved, this implies that many comparison data points were just barely plausible in iteration 1. This would mean that a majority of the data points have a high constraining power in Q19 MERAXES and similar implausibility values. The projection figures will allow us to see whether this is the case.

The 2D projections of the parameters related to the star formation and supernova feedback processes in MERAXES are shown in Figure 4 for all three emulator iterations. The dashed lines in the figures show the estimated value of the corresponding parameter as given in Table 4. Note that the parameter ranges in the projection figures differ between emulator iterations because the emulator was only defined over that specific range. For comparison, we show the full parameter range for the parameters in emulator iteration 3 in the final row.

Using this information, we can see some interesting trends in the projection figures. For example, all parameters except for the energy coupling efficiency, SnEjectionEff, seem to be moving away from their estimate. This could either mean that the parameter estimates are potentially biased, as we encountered earlier with the estimates given in Table 1, or that the LF/CMR data constrains these parameters differently. Despite this, however, the parameters are much more well-behaved than the projections in Figures 2 and 3, confirming that the parameter reduction has a positive effect on the convergence behavior.

Furthermore, we can see that our observation from earlier is indeed correct: most data points have similar implausibility values. This is particularly evident in the projection figures of $\epsilon_{\text{mod}}$ and the mass loading factor SnReheatEff, shows this as well. As the y-axis in the top subplot ranges from the lowest (reachable) implausibility value to the highest (plausible) implausibility value, it shows that there is little variation in the values it can take. Taking into account the implausibility cutoffs reported in Table 5, this behavior can only be obtained when most data points have similar implausibility values and thus similar constraining power as well.

In Figure 5 we show the 3D projections for the parameters related to the star formation and supernova feedback processes for all three emulator iterations. Because the star formation efficiency $SfEfficiency$ is commonly the most well constrained and most correlated with the other parameters, the first three projections show the correlations between $SfEfficiency$ and the other parameters. As the energy coupling efficiency SnEjectionEff and the mass loading factor SnReheatEff should be heavily correlated with each other, in addition to being modified in Q19 MERAXES, we show their 3D projections as well.
Looking at these projections, we can see that the star formation efficiency is indeed heavily correlated with the other parameters, revealing strong relations in all projection figures. Despite this, however, it seems that the energy coupling efficiency, \( \text{SnEjectionEff} \), cannot be effectively constrained. This is evident in both projection figures that show this parameter, which can be found in the second and fourth columns of Figure 5. In fact, it seems that nearly all values for this parameter except its estimate are equally plausible, which is particularly visible in the iteration 3 figure in the fourth column. This strongly implies that the SMF data cannot constrain this parameter, which might become important later.

Finally, we note that all parameters, with maybe the exception of the energy coupling efficiency \( \text{SnEjectionEff} \), seem to be limited by their own priors, implying that their best values lie outside of their parameter ranges. This is particularly clear in their 3D projection figures in the third column of Figure 5. This is another indication that the parameter priors might be too constrained or that the LF/CMR data constrains these parameters differently. To test this hypothesis, we checked what the SMF looks like when the best plausible samples found in the emulator are used.

In Figure 6 we show a comparison between the SMFs created by 50 plausible samples in the emulator and the SMF given by the parameter estimates from Table 4. Because an emulator consists of approximations, it is unable to provide a true best-parameter fit, and therefore we evaluated the emulator 2500 times within plausible space at iteration 3, and selected the 50 best plausible samples. These 50 samples were then evaluated in Q19 MERAXES to produce the solid lines shown in the figure.

From this figure, we can see that the emulator samples (solid lines) apparently fit the data much better than the parameter estimates from Table 4 (dashed line). However, even though the fits are better, the emulator still seems to be overestimating the number density of galaxies at low masses for all redshifts. Figure 4 shows that the lowest implausibility values are reached at either boundary for all parameters. This means that these 50 plausible samples all have values that are at the minimum or maximum value they can take. Therefore this figure confirms the hypothesis we made earlier: the parameter priors are too constrained, and/or that the LF/CMR data constrains these parameters differently. In order to test which of the two statements is true, we explore whether the trend is still apparent when using the LF/CMR data in the next section.
4.2. Exploring Meraxes Using Both SMF and Luminosity Data

Now that we have analyzed the Q19 MERAXES model and studied how it affects the behavior of the SMF parameters, it is time to analyze the model using all nine parameters. As mentioned before, we use the same data as used in M16 and Q19 (i.e., Bouwens et al. 2014, 2015; Song et al. 2016) and the same parameter ranges as given in Table 4. To allow for a better comparison with the previous emulator, this emulator also uses a model discrepancy variance $\text{Var}(\epsilon_{\text{mod},i})$ of $(\sigma_i/100)^2$. The statistics of this emulator are shown in Table 6.

As this emulator uses over three times the number of data points as the previous one (74 versus 24) and because we expect the LF/CMR data to constrain the model differently, we intentionally converged the emulator in a slower, more conservative fashion. We therefore have three implausibility wildcards for the first iteration. This can potentially also provide us with more information on the value of using the LF/CMR data as additional constraints for MERAXES.

Looking at Table 6, we can see that the emulator converged rather smoothly throughout all iterations. However, we note that there is a reduction of roughly a factor 50 in plausible space between iterations 3 and 4. Given that the difference in the implausibility parameters between these two iterations is not very large, this is implies that all data points have similar implausibility values and thus similar constraining power. As we found a similar trend for the SMF-only emulator (see Table 5), we investigate later using projection figures and emulator evaluations whether this is the case.

The 2D projections of the parameters related to the star formation and supernova feedback processes in MERAXES are shown in Figure 7 for all four emulator iterations. The dashed lines show the estimated value of the corresponding parameter as given in Table 4. When either parameter estimate is outside of the plotted value range, an arrow pointing in the direction of the intersection of the estimates is shown instead. First row: $i_{\text{emul}} = 1$. Second row: $i_{\text{emul}} = 2$. Bottom rows: $i_{\text{emul}} = 3$, with the final row showing the full parameter range instead of only the defined range, but it is otherwise equivalent to the third row.

Figure 5. 3D projection figures of the SMF-only Q19 MERAXES emulator at all three iterations, showing the four parameters related to the star formation and supernova feedback. The first three columns show the correlation between the star formation efficiency $\text{SFEfficiency}$ and the other parameters. The last column shows the correlation between the energy coupling efficiency $\text{SnEjectionEff}$ and the mass loading factor $\text{SnReheatEff}$. The dashed lines show the estimated value of the corresponding parameter as given in Table 4. When either parameter estimate is outside of the plotted value range, an arrow pointing in the direction of the intersection of the estimates is shown instead. First row: $i_{\text{emul}} = 1$. Second row: $i_{\text{emul}} = 2$. Bottom rows: $i_{\text{emul}} = 3$, with the final row showing the full parameter range instead of only the defined range, but it is otherwise equivalent to the third row.

The Astrophysical Journal Supplement Series, 253:50 (17pp), 2021 April van der Velden et al.
Figure 6. Realizations of the stellar mass functions at redshifts $z = [7, 6, 5]$ using the results of the SMF-only Q19 MERAXES emulator. All realizations were created by directly evaluating Q19 MERAXES. The solid lines use the 50 best plausible samples out of 2500 in the emulator at iteration 3. The dashed line uses the parameter estimates as given in Table 4. The dots show the SMF data from Song et al. (2016) adjusted for a Kroupa IMF, with corresponding standard deviations.

Table 6

| $i_{\text{emul}}$ | $n_{\text{eval}}$ | $I_{\text{true}}$ | $n_{\text{wild}}$ | $f_{\text{space}}$ |
|------------------|------------------|------------------|------------------|------------------|
| 1                | 500              | [4.0, 3.5, 3.2, 3.0] | 3                | 9.89%            |
| 2                | 1068             | [4.0, 3.5, 3.2, 3.0] | 1                | 1.39%            |
| 3                | 1519             | [4.0, 3.5, 3.2, 3.0] | 0                | 0.226%           |
| 4                | 1481             | [3.5, 3.0, 2.7, 2.5] | 0                | 0.00489%         |

Note. The first column specifies the emulator iteration $i_{\text{emul}}$ that this row is about. The next three columns provide the number of model evaluations $n_{\text{eval}}$, the non-wildcard implausibility cutoffs $I_{\text{true}}$, and the number of implausibility wildcards $n_{\text{wild}}$ used for this emulator iteration. Finally, the last column gives the fraction of parameter space remaining $f_{\text{space}}$ after this emulator iteration was analyzed.

...structure in them, which shows us that the parameters agree relatively well with their estimates.

In iteration 2 however, we note that this trend is starting to change for the star formation efficiency, SFEfficiency, and mass loading factor, SnReheatEff, which move away from their estimates. This becomes increasingly apparent over the next two iterations. When we compare the iteration 4 projections with the iteration 3 projections shown in Figure 7 to the iteration 3 projections in Figure 4, we can see that the parameters behave very similarly in both emulators.

Despite this, however, the parameter ranges are not nearly as well constrained as they are in Figure 4. Furthermore, whereas the minimum implausibility subplots showed strong correlations before, they do not do this in Figure 7. This is interesting because it suggests that the SMF data and the LF/CMR data constrain different regions of parameter space. It also implies, however, that it would be rather challenging to perform a global parameter estimation on these parameters because no region of parameter space is removed quickly. Instead, there is a tiny hypercube shell of plausible samples that stretches over parameter space.

In Figure 8 we show the 2D projection figures of the three free parameters used to calculate the dust optical depth. Note that the names of these parameters, as reported in the title of a projection figure, have an added LF_CMRCM_DTG_prefix. As these parameters are free parameters in a single equation, we do not expect them to show interesting details.

Looking at these projection figures, all the free parameters appear to agree with their estimates in all iterations, although $n$ is slightly questionable. However, with the exception of a small fraction for $n$, their plausible ranges are not reduced at all throughout the iterations. As these parameters are free parameters, they are expected to be heavily correlated with each other, and thus the behavior in the projection figures is logical. It would indicate a difficulty in finding their optimal values, however.

Similarly to the previous emulator, in Figure 9 we show the 3D projections for the parameters related to the star formation and supernova feedback processes for all four emulator iterations. As expected for the first emulator iteration, given its $f_{\text{space}}$ value of 9.89%, the first row of projections shows nothing of great interest. There are a few small indications of correlations between the parameters, but nothing significant.

The second row of projection figures, on the other hand, begins to exhibit some interesting behavior. Here, we can see that there are regions in parameter space where better (i.e., lower minimum implausibility) samples can be found, but the plausible samples are still very spread out over parameter space. The same can be observed for the third iteration. In the last iteration, however, there are actually significant parts of parameter space that are no longer considered to be plausible. We also note that, unlike the projections in Figure 5, the star formation efficiency SFEfficiency no longer dominates the other parameters as strongly. This further suggests that the SMF and LF/CMR data constrain different parts of parameter space, making it much harder to perform global parameter estimations.

Finally, we show in Figure 10 the 3D projection figures of the dust optical depth parameters. Similarly to the 3D projections shown in Figure 9, it takes until iteration 4 before the parameter space finally begins to be restricted, even though only 0.00489% of it is plausible. Even then, however, basically all values in the parameter ranges of the dust optical depth parameters are plausible, with no indication of converging. Here, it appears that the observational data is not enough to constrain these free parameters.

In Figure 11 we show a comparison between the SMFs, LFs, and CMRs created by 50 plausible samples in the emulator and the SMF, LF, and CMR given by the parameter estimates from Table 4. Similarly to Figure 6, we evaluated the emulator 2500 times within the plausible space at iteration 4 and selected the 50 best plausible samples. These 50 samples were then
evaluated in Q19 MERAXES to produce the solid lines shown in the figure.

From this figure, we can see that the SMF fits are very similar (if not the same) as those we showed in Figure 6. This is in agreement with the projection figures in Figures 4 and 7, which also showed similar parameter trends. However, whereas the emulator samples provide better fits for the SMF, it appears to provide roughly equal or slightly worse fits for the LF (the center row in Figure 11), while the CMR fits seem to be equal (bottom row). This implies that the SMF data constrains the LF much more heavily than the LF data itself does, but also that Q19 MERAXES might be unable to fit both the SMF data and the LF and CMR data simultaneously.

When we take the information shown in all projection figures and the two realization figures into account, we come to the conclusion that the prior parameter ranges as given in Table 4 are too restrictive and that the SMF and LF/CMR data constrain the MERAXES model differently. The former is clearly shown in the star formation and supernova feedback parameter projections in Figures 4 and 7. In addition to the energy coupling efficiency, SnEjectionEff, all parameters have their best values at or very near the boundaries of their priors for both emulators. The projection figures in Figure 7 and Figure 8, on the other hand, show that the parameters are much harder to constrain when the LF and CMR data is used as well.

Figure 7. 2D projection figures of the full Q19 MERAXES emulator at all four iterations, showing the four parameters related to the star formation and supernova feedback. The dashed lines show the estimated value of the corresponding parameter as given in Table 4. When the parameter estimate is outside of the plotted value range, an arrow pointing in the direction of the estimate is shown instead. First row: $i_{emul} = 1$. Second row: $i_{emul} = 2$. Third row: $i_{emul} = 3$. Bottom rows: $i_{emul} = 4$, with the final row showing the full parameter range instead of only the defined range, but it is otherwise equivalent to the fourth row.
These findings demonstrate two significant results: The importance of having the correct data points when performing parameter estimations, and the value of using PRISM for exploring models. Because only the LF and CMR data from Bouwens et al. (2014, 2015) was used by Q19 to constrain all nine parameters, the differing, stronger constraining power of the

**Figure 8.** 2D projection figures of the full Q19 MERAXES emulator at all four iterations, showing the three main free parameters related to the dust optical depth. The dashed lines show the estimated value of the corresponding parameter as given in Table 4. When the parameter estimate is outside of the plotted value range, an arrow pointing in the direction of the estimate is shown instead. First row: $i_{emul} = 1$. Second row: $i_{emul} = 2$. Third row: $i_{emul} = 3$. Bottom rows: $i_{emul} = 4$, with the final row showing the full parameter range instead of only the defined range, but it is otherwise equivalent to the fourth row.
SMF data was not noticed. As we can see in the 2D projections in Figure 7, the best values for the star formation and supernova feedback parameters are similar to those in Figure 4, implying that the SMF data constrains these parameters more heavily than the LF and CMR data. The two emulators discussed in this section required roughly 7000 evaluations of the MERAXES model, multiple orders of magnitude less than the average amount used by an MCMC approach. Despite this, however, the emulators still show us all this crucial information.

In addition to the importance of using the correct data points, the projections in Figure 9 highlight the potential dangers of having too many degrees of freedom in a model (or, alternatively, the dangers of not having enough constraining data points). Even though the SMF and the LF and CMR data constrain the MERAXES model differently, causing a wide range of parameter values to be plausible, we can see much more structure in these projections than in those we showed in Figure 3. We therefore also note the importance of choosing the degrees of freedom for a scientific model wisely.

5. Conclusions

We have analyzed the galaxy formation model MERAXES using the emulation-based framework PRISM. Two different
Figure 10. 3D projection figures of the full Q19 MERAXES emulator at all four iterations, showing the three free parameters related to the dust optical depth. The three columns show the correlation between the star formation efficiency $S\text{E}fficiency$ and the main free dust optical depth parameters. The dashed lines show the estimated value of the corresponding parameter as given in Table 4. When either parameter estimate is outside of the plotted value range, an arrow pointing in the direction of the intersection of the estimates is shown instead. First row: $\kappa_{\text{emul}} = 1$. Second row: $\kappa_{\text{emul}} = 2$. Third row: $\kappa_{\text{emul}} = 3$. Bottom rows: $\kappa_{\text{emul}} = 4$, with the final row showing the full parameter range instead of only the defined range, but it is otherwise equivalent to the fourth row.
versions of MERAXES have been studied: the original created by M16, and a modified one by Q19. Both versions have provided us with different, interesting results.

The M16 MERAXES model has shown to be difficult to constrain properly using the SMF observational data at \( z = [7, 6, 5] \) alone. As described in Section 3, several indications imply that the M16 MERAXES model has too many free parameters, mainly for the supernova feedback parameters SnEjection... (supernova energy coupling efficiency) and SnReheat... (mass loading factor) to be able to explain these SMF observations. We also come to the conclusion that the manually calibrated parameter values for M16 MERAXES are likely to be biased because the parameters do not converge, which is probably caused by the large correlations between the free parameters. PRISM has demonstrated that it can quickly perform a rough parameter estimation using only a few thousand model evaluations, which provided us with the mentioned results.

In Section 4 we analyzed the Q19 MERAXES model, which has a reduced parameter space for the star formation parameters, and uses only a single degree of freedom for both supernova feedback parameters. Our analysis has shown that this greatly improves the convergence rate of the MERAXES model and that it greatly facilitates constraining MERAXES, as shown in Table 5 and Figure 5. This figure also showed us the value of using the performance of acceptable parameter values...
(e.g., minimum implausibility) in addition to their density when performing parameter estimations.

Afterward, in Section 4.2, we constrained the Q19 MERAXES model using both SMF and LF and CMR data. This gave us two important results: The SMF and the LF and CMR data constrain the MERAXES model differently, and using heavily constrained parameter priors can bias the best parameter fits. The latter was shown in Figures 4 and 7, where the best parameter values can be found very near or at the boundary of their priors. Alternatively, when looking at this from a physics standpoint, this can imply that aspects of the MERAXES model itself are incomplete or incorrect if the used priors are determined by their physical meaning, which is a rather important detail to know.

The results in Sections 3 and 4 show us the importance of using more than one observational data type when constraining models, especially when appropriate observational data is scarce. Additionally, the choice of the degrees of freedom in a model should be made wisely. With our results, we have demonstrated the potential of using an emulation-based framework like PRISM for analyzing scientific models, which is a task that a full Bayesian analysis cannot perform quickly. The use of PRISM allows us to discover important details such as those mentioned above at an early stage, and we therefore recommend it as a core component in the development of scientific models.

E.v.d.V. would like to thank Michael Goldstein, Manodeep Sinha, and Ian Vernon for fruitful discussions and valuable suggestions. Parts of the results in this work make use of the rainforest and freeze color maps in the CMasher package (Van der Velden 2020). We are thankful for the open-source software packages used extensively in this work, including E13TOOLS, H5PY (Collette 2013), HICKLE (Price et al. 2018), MATPLOTLIB (Hunter 2007), MLXTEND (Raschka 2018), MPI4PY (Dalcin et al. 2005), NUMPY (Van der Walt et al. 2011), SCIKIT-LEARN (Pedregosa et al. 2011), SCIPY (Virtanen et al. 2020), and TQDM. Parts of this research were supported by the Australian Research Council Centre of Excellence for All Sky Astrophysics in 3 Dimensions (ASTRO 3D), through project number CE170100013. Parts of this work were performed on the OzSTAR national facility at Swinburne University of Technology. OzSTAR is funded by Swinburne University of Technology and the National Collaborative Research Infrastructure Strategy (NCRIS).