

Planning a Universe

Nested Designs and Input Transformations for the Dark Universe Simulations

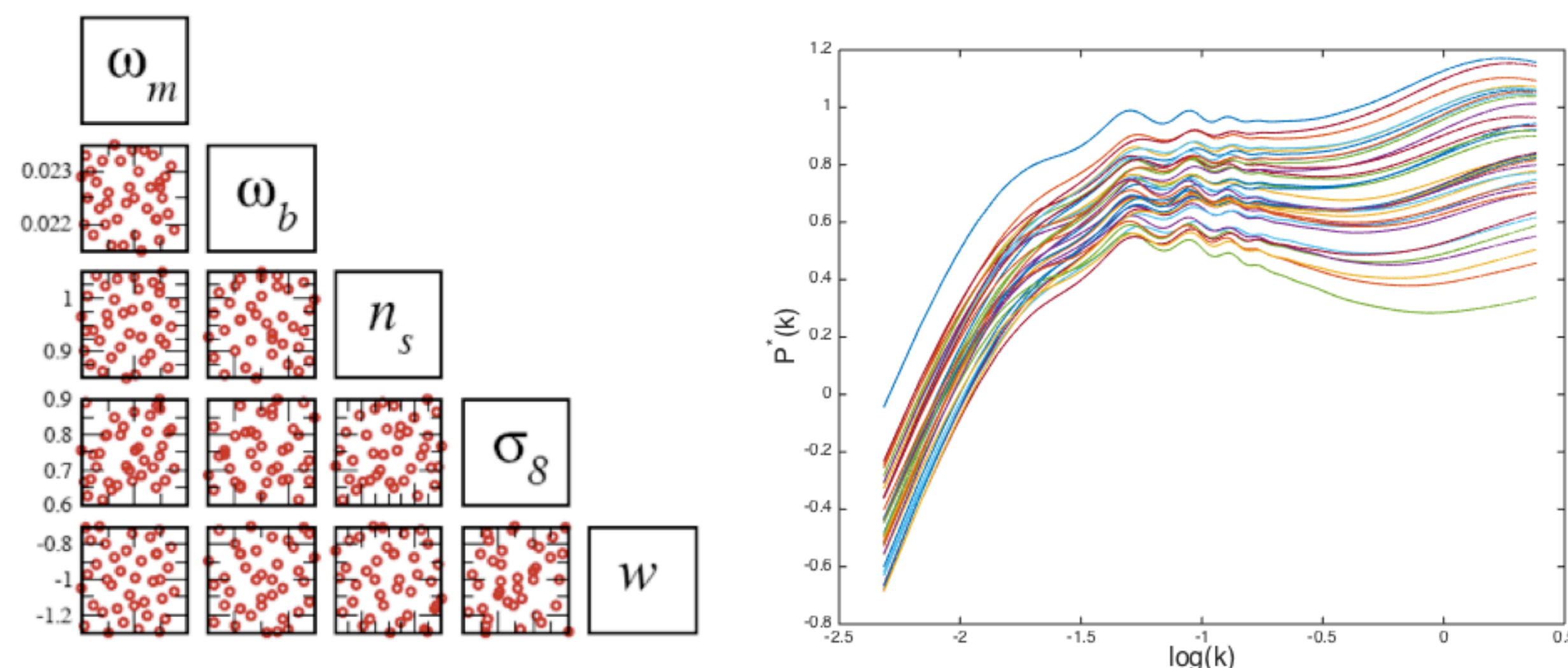
Earl Lawrence (LANL), Derek Bingham (Simon Fraser), David Higdon (Virginia Tech), Katrin Heitmann (ANL), Salman Habib (ANL)

Introduction

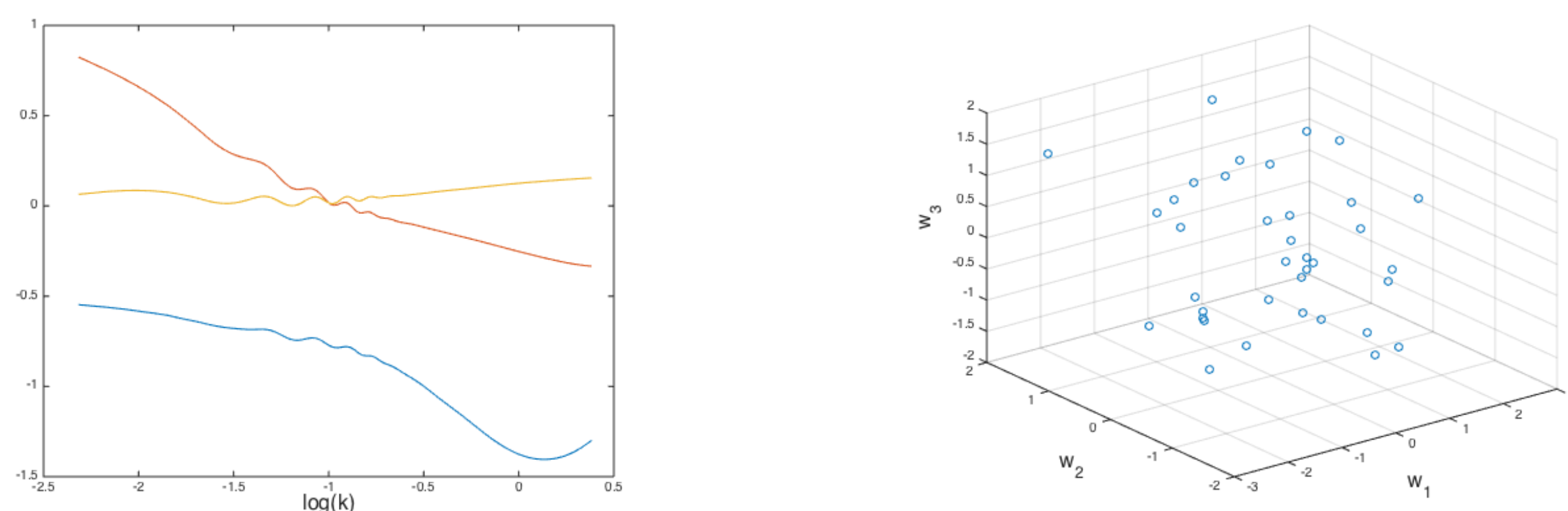
Gaussian process emulators are very effective at making fast predictions of complex scientific computations that usually require very computationally expensive simulations (Higdon et al. 2008). After training (described below), the emulator is able to predict the simulator output at new, untried settings of the inputs, but in a fraction of the time. This speed increase makes the emulator useful for tasks like parameter estimation and model validation. In cosmology, this scheme makes predictions of the nonlinear matter power spectrum that match the results computed from brute force n -body simulations to a few percent (Lawrence et al. 2010). We will describe some of our preparation for the next generation of cosmological emulators.

Emulation and the Original CosmicEmu

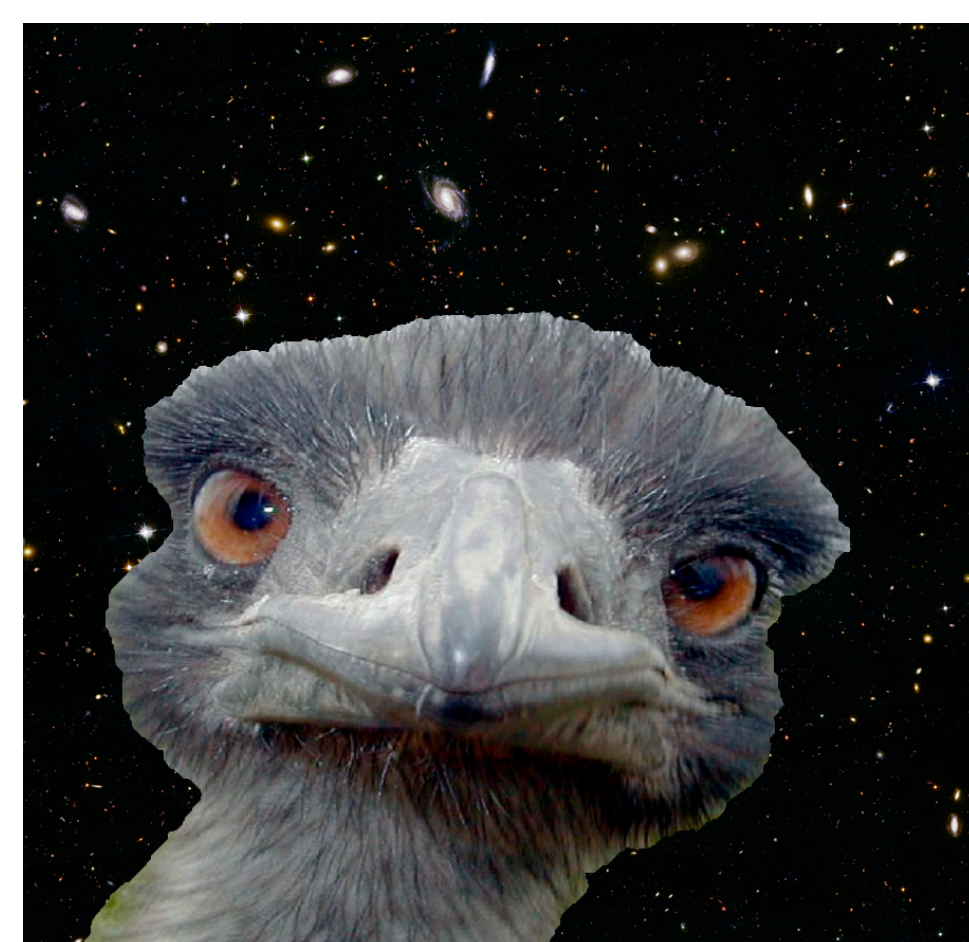
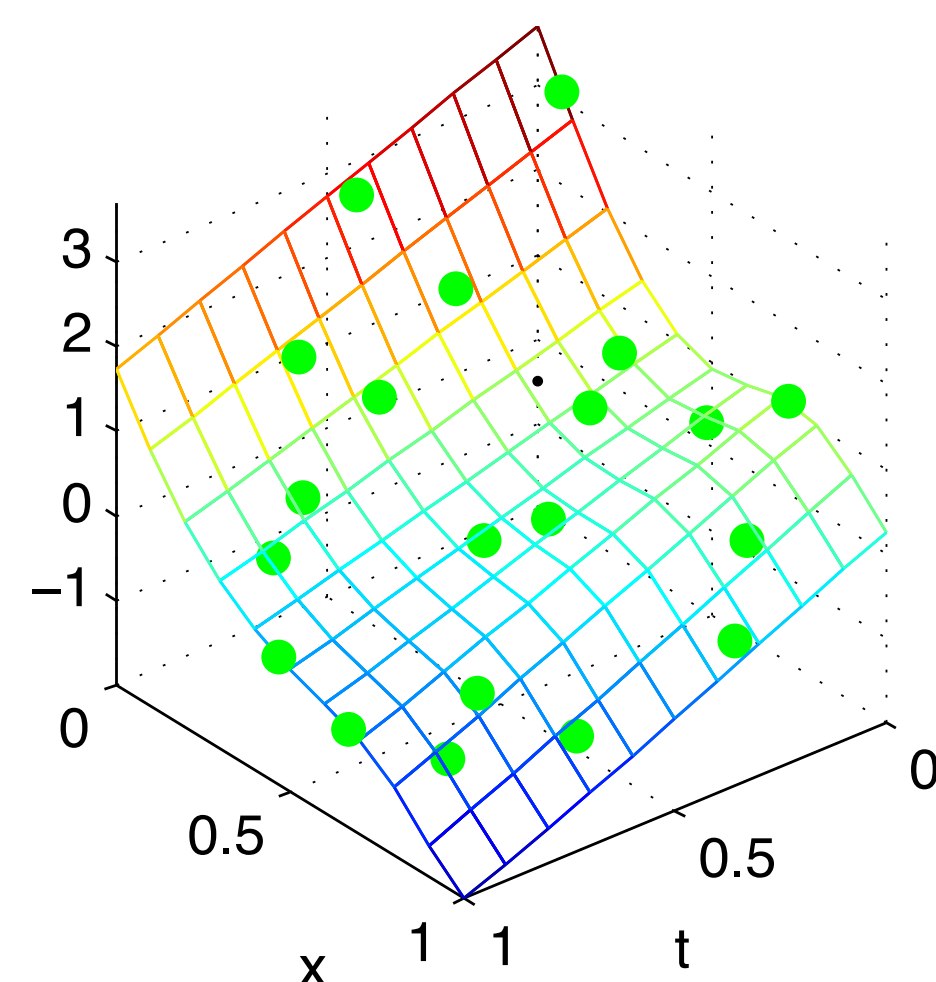
We choose 37 settings of the inputs (left below) at which to run the expensive simulation. Each simulation produces a quantity of interest, usually functional (right below).



The functions are decomposed in a set of basis vectors (left below) and weights (right below). These are usually based on an SVD (i.e. principal components).



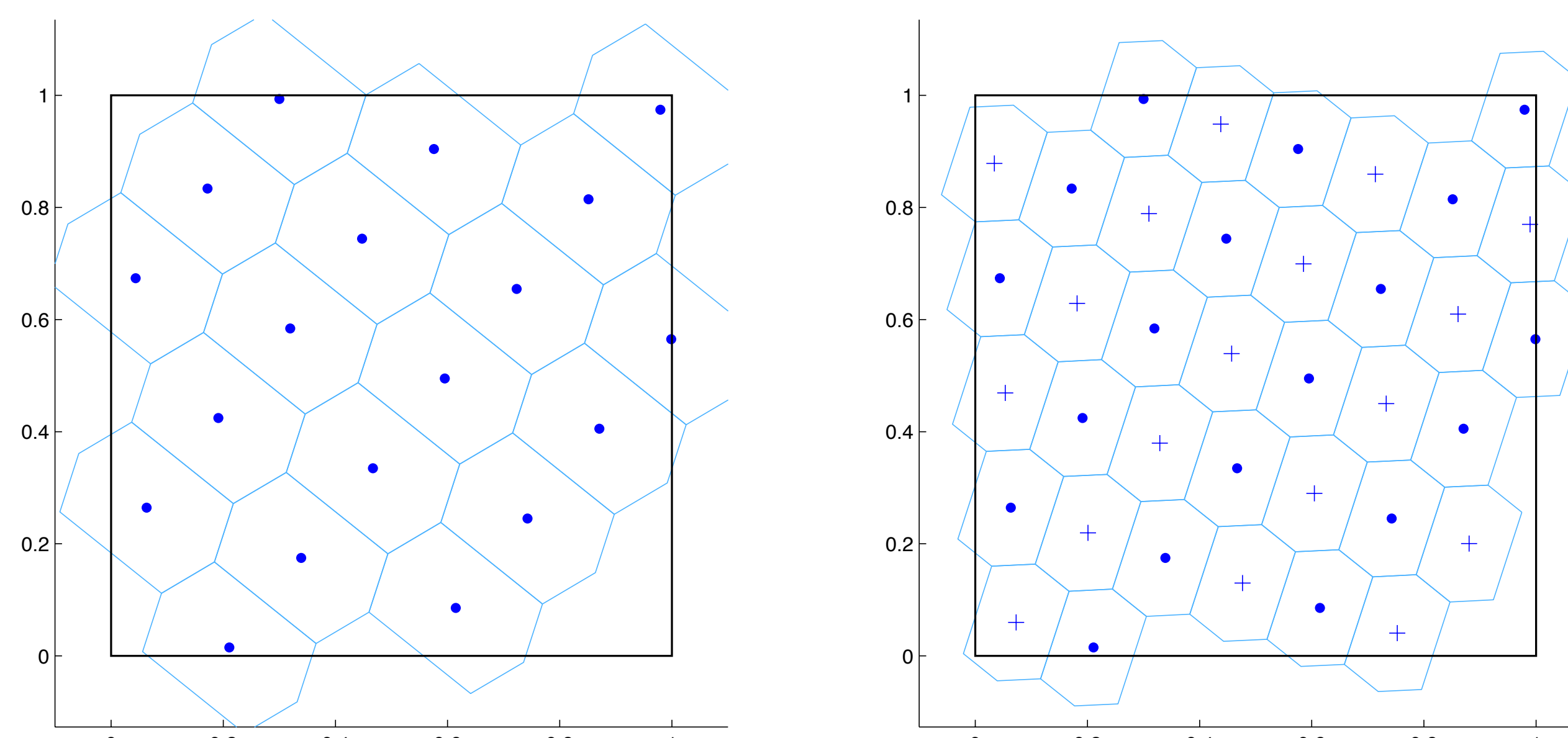
Each set of weights is independently modeled as a function of the original inputs using a Gaussian process (left below). These learned functions can predict the weights, and thus the function of interest, at untried inputs. The trained functions for the original cosmology simulations were coded in C and released to the astro community as the Cosmic Emu (right below).



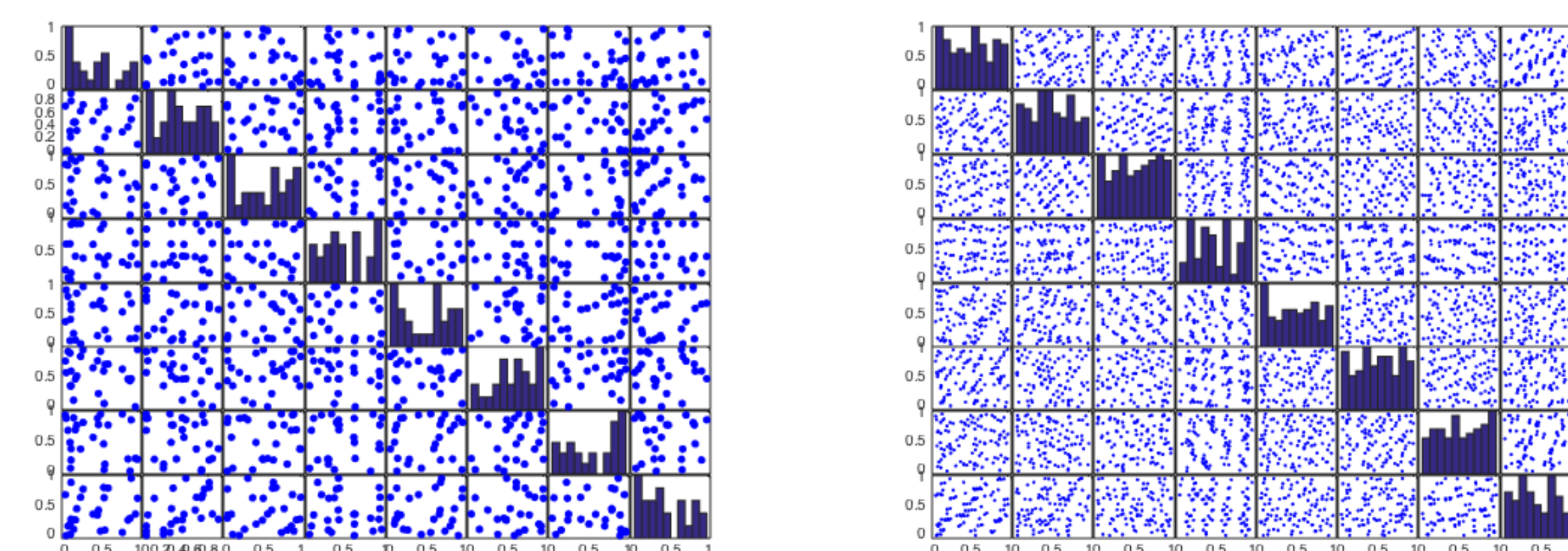
Lattice Designs for Partial-Result Emulation

The suite of simulations, the Mira Universe, for the next generation emulator will take a number of years of to complete. In total, we will produce about 100 runs over an input space of 8 parameters. An emulator based on this expanded set of parameters will be a valuable addition to the astro community. Therefore, we would like to release new software without waiting for the entire suite to be ready. However, each release needs to be based on a set of inputs that fill the space uniformly. To solve this problem, we use an input design based on *lattices* (Bergner 2011).

This design strategy is easily explained in the figure below. The left panel shows a 19-run design in two dimensions. These 19 points fall on the intersection of a regular lattice (transformation of the 2-d integers) and the design region of interest (unit square). The right panel shows 37 runs that are the intersection of the region of interest and a *dilation* of the same lattice. This dilation results in a set of 37 runs that include the 19 runs from the left panel.



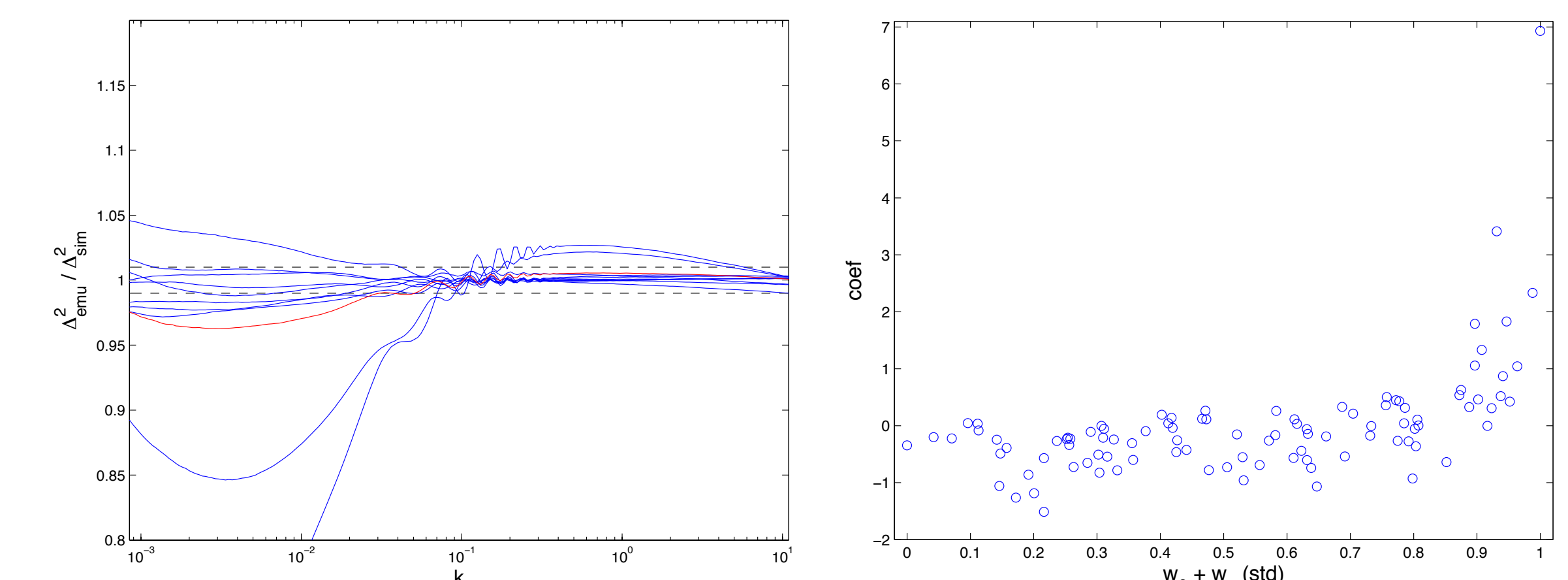
This process can be repeated, doubling the size of the design each time. At each stage, there is an approximately uniform distribution. Our design has three stages with complete lattices at 26, 55, and 101 runs (26 and 101 are shown below). When the simulations for each stage are complete, we will release an emulator based on the results.



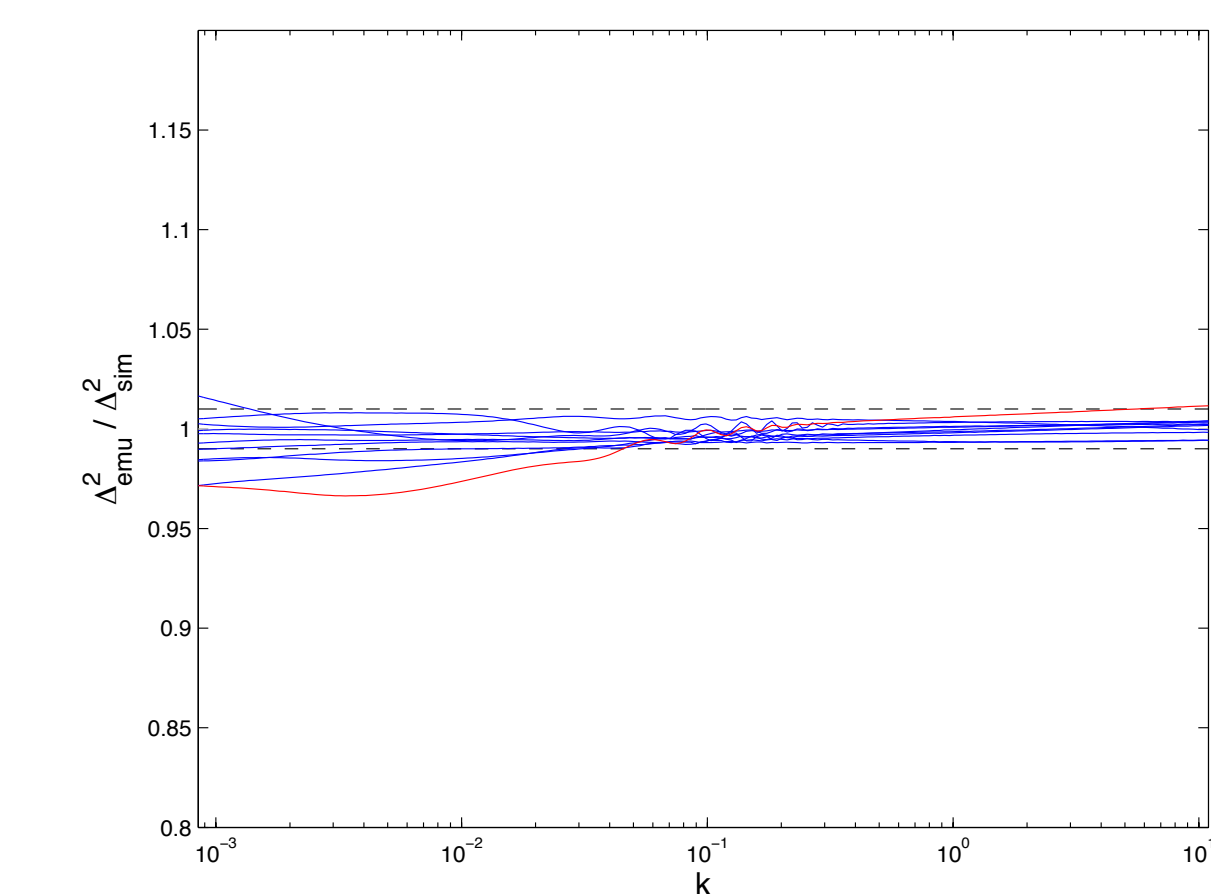
A smaller issue is that the current best fit value of the neutron mass is zero, on the boundary of its range. To improve prediction near this value, we supplement the simulation suite with 10 runs from a Latin hypercube over the other 7 parameters, with the neutron mass set to zero.

Transforming the Design Space

We also tested the overall strategy using a proxy model for the matter power spectrum. This allowed us to choose the best of several designs based on the performance on a test set. In the process, we noticed unexpectedly poor performance in a specific region of the design space. Our new parameter space replaces the single parameter for a constant dark energy equation of state with two parameters describing a linear change in time. The sum of these two parameters is bounded above by zero. The left panel in the figure below shows the prediction results for our test set. The scale is the ratio of the emulator prediction to the simulation result. The red line is the result when using the current estimate of the parameters from other sources. Two lines show errors of at least 10% for the lower part of the spectrum. These simulations have dark energy parameters whose sum approaches the theoretical boundary. These results suggest that the error becomes increasingly worse as the sum approaches the limit. Some exploration reveals the source. The right panel shows the weights for one of the basis vectors plotted against the sum of the dark energy parameters. This weight changes suddenly at the end of the range and our stationary Gaussian process models this type of behavior poorly.



In order to maintain stationarity but retain our modeling simplicity, we explore some transformations of our input parameters. The above results suggest a polynomial transformation of the sum of the parameters. In the end, the best results are achieved by transforming the original parameters, w_0 and w_a , into w_0 and $(-w_0 - w_a)^{0.25}$. On this scale, our test results are considerably improved, as shown below.



References

- Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008), "Computer Model Calibration Using High Dimensional Output," *Journal of the American Statistical Society*.
- Lawrence, E., Heitmann, K., White, M., Higdon, D., Wagner, C., Habib, S., Williams, B. (2010), "The Coyote Universe III: Simulation Suite and Precision Emulator for the Nonlinear Matter Power Spectrum," *The Astrophysical Journal*.
- Bergner, S. (2011), "Making choices in multi-dimensional parameter spaces," PhD Thesis, Simon Fraser University.