Super-parameterisation of ocean deep convection

Y. Andrianakis and P. Challenor

December 1, 2011

Abstract

Sub-grid scale processes play an important role in ocean and climate modelling. Typical examples include clouds in atmospheric models, flows over restricted topographies or the resolution of convective plumes in ocean models. Detailed numerical models of these sub-grid scale processes exist, but embedding them in a Global Circulation Model (GCM) for example, would be computationally prohibitive. In the present work we investigate the applicability of emulators for representing the sub-grid scale processes within a GCM simulation. Emulators can be thought as encapsulating our beliefs about the sub-grid dynamical model, derived from a designed computer experiment using a Bayesian framework. In particular, we propose to employ an emulator for parameterising the sub-grid scale process, and embed this within a GCM as a surrogate for the actual sub-grid scale model. The result of combining the GCM with the emulator will be a Super-parameterised model, which will also be computationally efficient, since the emulator incurs a very small computational overhead. The example we chose to illustrate the proposed methodology is deep ocean convection. The sub-grid scale dynamical model simulates deep convective plumes, while the large scale dynamics simulate the geostrophic eddy scale. We present details on building the emulator of the convective plumes and its coupling with the large scale process model. We also discuss whether the emulator should be run as a deterministic or stochastic parameterisation.

1 Introduction

We study the problem of ocean deep convection in a volume of water that is $40 \times 40 \times 2$ km. On the surface, a circular area of radius 10 km is cooled with a heat loss function of 800 Wm^{-2}. The water is initially stratified with a uniform temperature drop of 0.005°C per 100 m. The surface cooling creates plumes of cooled water that penetrate into the deeper layers. This creates a baroclinic instability, which results in the formation of geostrophic eddies. The latter create a horizontal mixing between the cooled area water and the surrounding stratified water. The formation of cooled water plumes represents the small (sub-grid) scale process in our problem and the geostrophic eddies represent the large scale process.

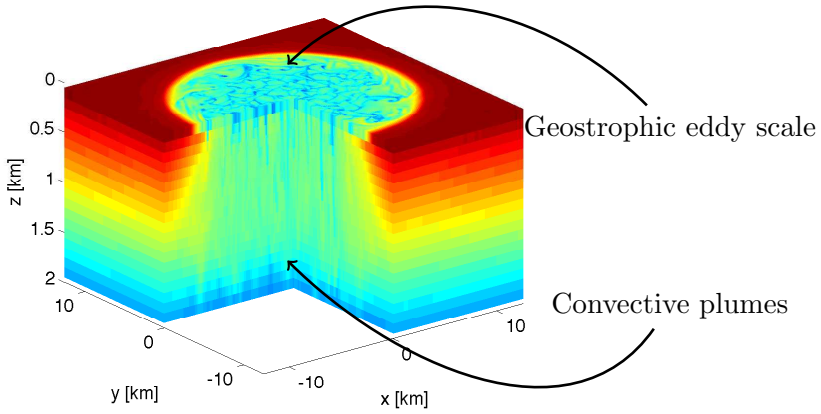


Figure 1: Results of the Reference model after 4 days of cooling.

The above processes are modelled with a high resolution configuration of the MITgcm model. The cubic grids have an edge size of 100 m yielding a model with $400 \times 400 \times 20$ cells. The high resolution of the model and its non-hydrostatic configuration allow the resolution of the convective plumes; the extent of its domain in the vertical and horizontal directions can reproduce the geostrophic eddies. The above model is the *reference* model (RM) in our work and its configuration is the same as that of the reference model in Campin et al. (2011). Its results after 4 days of cooling are shown in Figure 1, where the convective plumes and the geostrophic eddies can also be seen.

Although a high resolution model, as the reference model described above, can resolve both the small and large scale processes, such models can be quite expensive computationally. The computational load can be reduced by approximating the small scale process. Such an approach is the convective adjustment scheme of Klinger et al. (1996), which is a 1-d convection parameterisation that mixes unstable water through enhanced vertical diffusivity. Campin et al. (2011) proposed a Super-parameterisation scheme, that employs 2-D models that resolve rather than parameterise convection and embedded these in a coarse geostrophic model. The computational savings of this approach are due to the convection sub-models having 2 instead of 3 dimensions and because they can be massively parallelised.

The work we propose here has similarities with the work of Campin et al. (2011), in that we use separate models for the small and large scale processes. The major difference is that the small scale process is resolved by *emulated* three dimensional convection models. An emulator is a statistical representation of a computer model (simulator), which can be trained on model generated data and make predictions about the simulator's output for untested input configurations. A key benefit of an emulator, is that once trained it can make predictions about the model almost instantaneously.

2 Super-parameterised model

In this section we describe the Super-parameterised model (SP) that does not make use of emulators. This model is an intermediate step to developing the emulator-based Super-parameterised model

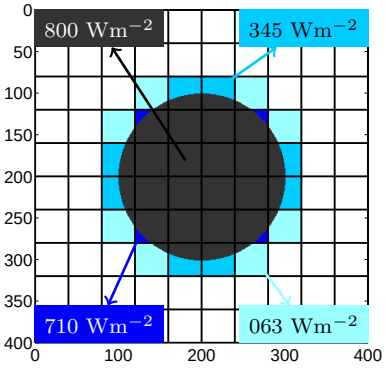


Figure 2: Segmentation of the Reference model's forcing function according to the Coarse model's grid. The different levels are represented with different colors. The numbers represent the heat loss of each level if the cooling surface were uniform within each cell.

(SPe). Considering that the latter model will be an approximation of SP, we would like SP to match the Reference model as closely as possible. The SP model we propose consists of two separate parts: the first is a *Coarse* model, that captures the large scale process and the second is an array of *Fine* models, that resolve the small scale processes. Details about the two sub-models of SP are given in Table 1. This table contains also information about the Reference model as well as about the Coarse model with the Convective Adjustment scheme that we will use for comparisons at later sections.

	Coarse	Fine	Reference	Coarse + C.A.
Dimensions (km)	$40 \times 40 \times 2$	$4 \times 4 \times 2$	$40 \times 40 \times 2$	$40 \times 40 \times 2$
Dimensions (cell)	$10 \times 10 \times 20$	$40 \times 40 \times 20$	$400 \times 400 \times 20$	$10 \times 10 \times 20$
Non-hydrostatic	✗	✓	✓	✗
Time step (sec)	1	1	1	1
Conv. Adjust.	✗	✗	✗	✓

Table 1: Main characteristics of the Fine and Coarse parts that comprise the Super-parameterised model, in addition to the Reference and Coarse model with the Convective Adjustment.

The Coarse model spans the same area as the Reference model, but has a much lower resolution. The model operates in *hydrostatic* mode, as it is intended to model only the geostrophic eddies, and there is no external forcing (cooling). The Fine models are a hundred identical 40×40 segments of the Reference model. Their forcing function varies with their position within the Coarse model's grid. Figure 2 shows the partitioning of the Reference model's forcing function to 100 segments that correspond to each of the Fine models. The symmetry of the problem yields 4 distinct forcing levels.

With the Coarse and Fine models in place, we now proceed to describe how these are coupled in order to create the Super-parameterised model. The coupling procedure consists of 4 stages, which are the propagation (stepping) of the Fine and Coarse models in time and passing the data between

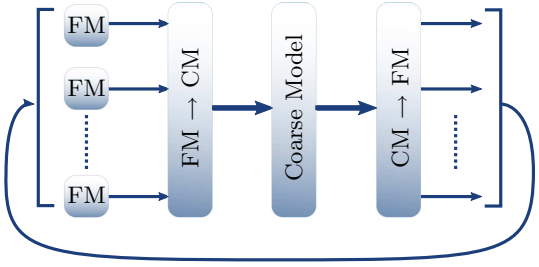


Figure 3: Coupling the Fine and Coarse models.

each of the two submodels. These steps are described in the following and illustrated in Figure 3.

2.1 Propagating the Fine models

The first step of the coupling scheme is to propagate the array of Fine models forward. The Fine models are propagated for $t_c = 12$ hrs, which is also the interval at which the Fine and Coarse models communicate with each other. At the end of the run, the internal state of the Fine models is stored for restarting them in the next iteration.

The reason for selecting such a large communication interval is that we want to reduce the Fine models' dependency on all initial conditions apart from the temperature. This choice was made because we would like the Fine model emulators to depend only on the input depth-temperature (DT) profile and the forcing but not on any other initial conditions (e.g. velocities, tendencies, pressures). This strategy will eliminate the need for storing the Fine models' state in the emulator based Super-parameterised model that is described in Section 3.

2.2 Mapping the Fine models to the Coarse

Each of the 100 Fine models produces a 3 dimensional temperature field after 12 hrs of cooling. This field is averaged along the x and y dimensions, to produce a 20×1 DT profile. The collection of the 100 DT profiles (one from each Fine model) are used to initialise the temperature of the coarse Model.

2.3 Propagating the Coarse model

The Coarse model is initialised with the DT profiles obtained in the previous step. The model is run for 12 hrs without forcing, as this is only applied to the Fine models. The Coarse model's internal state is saved at the end of the run, so that it can be used for restarting it in the next iteration, after the temperature fields have been updated with the results of the Fine models.

2.4 Mapping the Coarse model to the Fine

The results of the Coarse model are finally used to adjust the mean temperature levels of the Fine models. The adjustment is carried out with the following scheme (Campin et al., 2011). Let θ_f be the 3 dimensional temperature field of a Fine model, and $[\theta_f]_c$ the DT profile that would result from a spatial average. Let also θ_c be the DT profile of the Coarse model at the location of the Fine. The Fine model's temperature is updated with the following rule:

$$\begin{aligned} \theta_f &\leftarrow T_\theta + (\theta_f - T_\theta)\frac{\theta_c - T_\theta}{[\theta_f]_c - T_\theta} \\ \text{with } T_\theta &= \min(\theta_c, \min_{\text{sp}}(\theta_f)), \text{ if } \theta_c < [\theta_f]_c \\ \text{and } T_\theta &= \max(\theta_c, \max_{\text{sp}}(\theta_f)), \text{ if } \theta_c > [\theta_f]_c \end{aligned} \tag{1}$$

where $\max_{\text{sp}}$ and $\min_{\text{sp}}$ are operate spatially along the x and y dimensions of the 3 dimensional temperature field θ_f. The above averaging scheme was proposed in Campin et al. (2011) to avoid the appearance of false extrema in the temperature fields.

3 Super-parameterised model with emulation

The emulation based Super-parameterised model is identical to the Super-parameterised model described in Section 2, with the exception that the array of Fine models is replaced by an array of emulators. According to Figure 2, segmenting the forcing function of the Reference model to the grid of the Coarse, yields 4 different forcing levels. A separate emulator is built for each level, which is then used as a substitute for the respective Fine model. An alternative approach would be to use the forcing level as an additional emulator input. We chose however to build separate emulators for each of the 4 different forcing levels, because of the simple geometry of the problem and in order to reduce the emulators' input space. No emulators are built for the Coarse model cells that have no forcing input, because we make the assumption that no convection occurs in these. The DT profiles of these cells are not modified between consecutive runs of the Coarse model.

The emulators we build in this section accept a DT profile as input and estimate the DT profile at the output of a Fine model after 12 hours of cooling, conditional on the forcing function. Substitution of the emulators in the Super-parameterisation scheme described in Section 2 yields the emulation based Super-parameterised model (SPe). Note that the emulators do not depend on any initialisation variables of the Fine models (e.g. velocities and pressures), apart from the temperature. This inevitably introduces some error in the results, but circumvents the need for storing all the intermediate variables. In the following, we describe the procedure for building an emulator for one Fine model and a given forcing level, a procedure which is then repeated for all available levels.

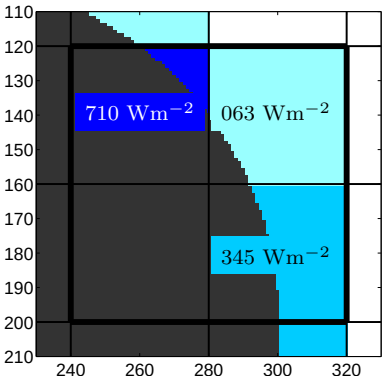


Figure 4: 80 × 80 part of the Reference model run for obtaining input training data.

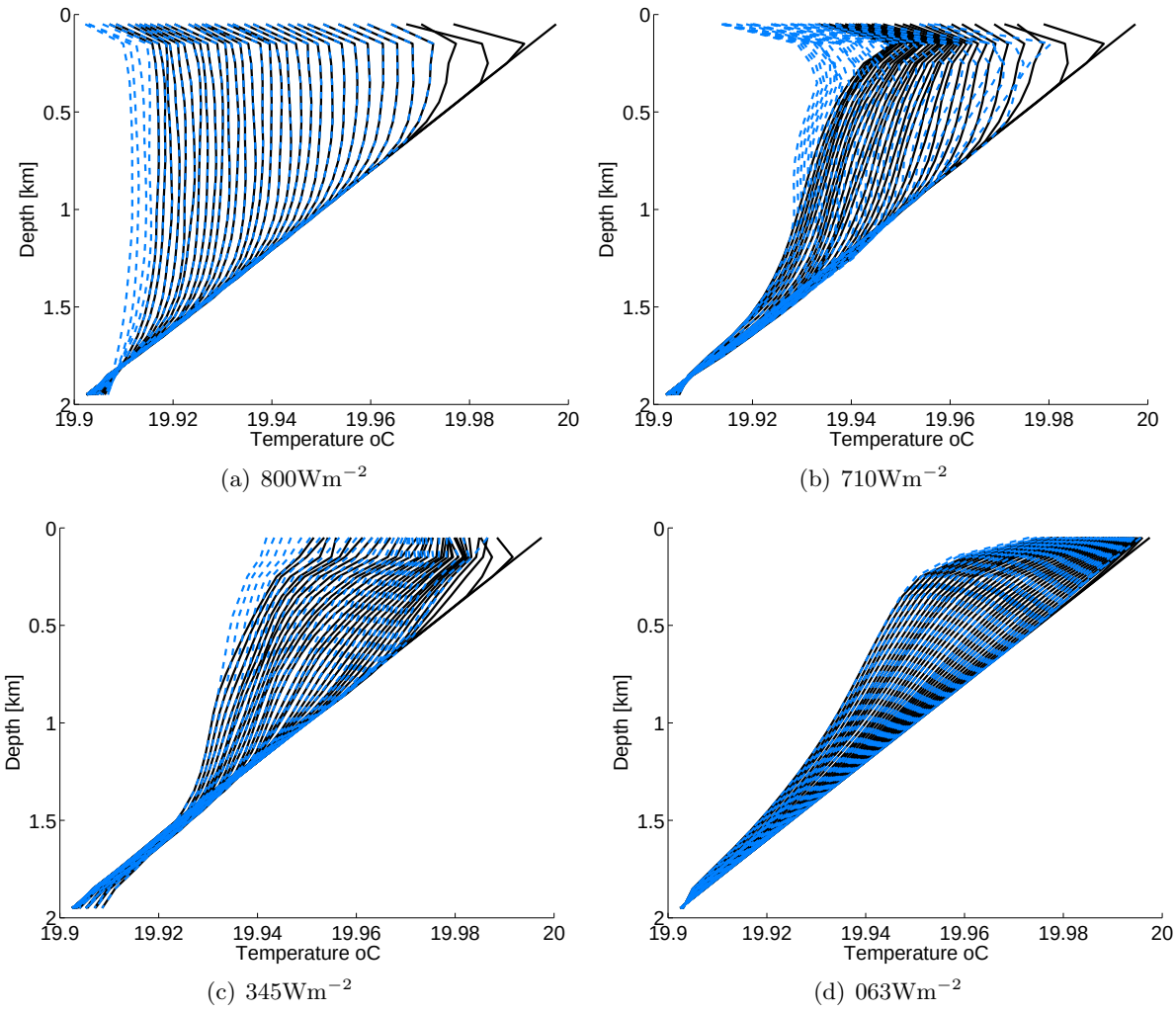
3.1 Training data

The first step in emulating a Fine model is to generate the data that will be used for training and validating the emulator. This data set will consist of a number of input DT profiles which will then be fed to the Fine model (with the given input forcing level) to produce the respective output DT profiles. This set of input - output DT profiles will serve as the training and validation data for the emulator.

The generation of training data for the 800 Wm^{-2} emulator is fairly straightforward. A single Fine model with a uniform forcing of 800 Wm^{-2} is run for 4.5 days and the DT profiles are stored every 3 hours. This creates 33 pairs of input - output (i.e. after 12 hours) training data sets. The 'sequential' approach in acquiring these data implies that some input and output DT profiles will be identical (see Figure 5(a)).

The generation of data for the other 3 emulators is slightly more involved. In the Coarse model's cells that correspond to the 800 Wm^{-2} forcing (i.e. the central cells) the DT profiles will be largely determined by the external forcing. However, in the cells that correspond to the edge of the cooling region, the DT profiles will also be influenced by the horizontal diffusion of the temperature, as the heat loss function has less power in those cells. Running a single Fine model with the 710, 345 or 063 Wm^{-2} forcings of Figure 2 will not yield DT profiles that are representative of those found in the respective regions of a Reference model run. That would result in training the emulator with DT profiles that are unlikely to be encountered during the run of the SPe model and would lead to poor estimates. To circumvent this problem we implement the following strategy.

We build a 80 × 80 portion of the Reference model, that corresponds to a part on the edge of the cooling surface that contains all the three lower levels of forcing. This model is shown in Figure 4. We run the model for 4 days, saving the DT profiles, for each 3 of the 4 regions of interest every 3 hours. The profiles from each region are then fed to a Fine model with the respective forcing. The result will again be 33 input - output pairs of DT profiles, which should be more representative of the DT profiles that would be encountered in the Reference model at the edge of the cooling circle. The training data for all 4 emulators are shown in Figure 5.



(a) 800Wm^{-2} (b) 710Wm^{-2}

(c) 345Wm^{-2} (d) 063Wm^{-2}

Figure 5: Training data for the 4 emulators. The black lines represent the input DT and the dashed blue lines the output (after 12 hours) DT profiles. Note that 29 out of the 33 input-output profiles of the 800 Wm^{-2} emulator are identical, as they come from the same model run.

3.2 Dimension reduction

The DT profiles θ, as they are shown in Figure 5, are 20×1 vectors, i.e. $\theta \in \mathbb{R}^{20}$. Considering that the elements of θ seem to be strongly correlated, we wish to decrease their dimension by representing them using a small number of bases. A suitable transformation that can achieve this is Principal Component Analysis (PCA).

Let θ_{i}^i and θ_{o}^i, $i \in [1, 33]$, represent the input and output training data for a given emulator. We aggregate these data in a matrix $\Theta \in \mathbb{R}^{20,66}$, subtract their row-wise mean μ_θ and find the covariance matrix $C_\theta = \mathrm{Cov}[\Theta]$. The eigenvalues and eigenvectors of the covariance matrix C_θ are then calculated, yielding the equation

$$C_\theta = \tilde{V}\tilde{D}\tilde{V}^{\mathrm{T}} \tag{2}$$

where $\tilde{V}$ is a matrix of eigenvectors and $\tilde{D}$ is the diagonal matrix of eigenvalues. Out of the 20 eigenvalues of C_θ, the first five represent more than 0.999 of the variance; therefore we can represent the DT profiles using only the 5 first eigenvalues in $\tilde{D}$ and the 5 first eigenvectors in $\tilde{V}$. We represent these reduced matrices as V and D.

We now define the forward and inverse transforms that map the space of DT profiles to the reduced space of principal components and vice versa. We define the forward transform as

$$x = \mathcal{F}(\theta) = \mathcal{U}\left[D^{-1/2}V^{\mathrm{T}}(\theta - \mu_\theta)\right] \tag{3}$$

and the inverse transform as:

$$\theta = \mathcal{I}(x) = VD^{1/2}\mathcal{U}^{-1}[x] + \mu_\theta \tag{4}$$

The operator $\mathcal{U}[\cdot]$ is a linear transformation that maps the coefficients of Θ to the $[0, 1]$ interval; $\mathcal{U}^{-1}[\cdot]$ is the inverse transform. This transformation is applied so that the data used for training the emulator lie in the unit interval.

3.3 Emulator

The procedures described in the previous two sections provide us with a training set for building an emulator for the Fine model and a given forcing function. Let us denote the input data as $X \in \mathbb{R}^{p,n}$, with $X = \{x^{(i)} = \mathcal{F}(\theta_{\mathrm{i}}^{(i)})\}$ and the output data as $Y \in \mathbb{R}^{p,n}$ with $Y = \{y^{(i)} = \mathcal{F}(\theta_{\mathrm{o}}^{(i)})\}$, for $i \in [1, n]$. The index $p(= 5)$ denotes the number of inputs and $n(= 33)$ denotes the number of training data. A multivariate separable emulator is built according to the paradigm of Conti and O'Hagan (2010). The emulator is described by the conditional posterior distribution of the DT profile at the output of the Fine model, given the DT profile at its input and a number of emulator hyperparameters. The posterior distribution is the following multivariate t-student distribution with $n - q$ degrees of freedom

$$p(\eta(\cdot)|Y, \delta, \nu) = \mathcal{T}_{n-q}(m(\cdot), v(\cdot, \cdot)) \tag{5}$$

where the posterior mean is given by

$$m(x) = h(x)^{\mathrm{T}}\hat{\beta} + t(x)^{\mathrm{T}}A^{-1}(Y^{\mathrm{T}} - H\hat{\beta}) \tag{6}$$

and the posterior covariance is given by

$$v(x, x') = \hat{\Sigma} \left\{ c(x, x') - t(x)^{\mathrm{T}} A^{-1} t(x') + R(x)(H^{\mathrm{T}} A^{-1} H)^{-1} R(x')^{\mathrm{T}} \right\} \tag{7}$$

The above equations introduced a number of new quantities that will be detailed in the following. Firstly, $h(x)$ is a predefined set of basis functions of the input x, which in our case were $h(x)^{\mathrm{T}} = [1, x]$. The matrix H is defined as $H = [h(x^{(1)}), h(x^{(2)}), \cdots, h(x^{(n)})]$. The correlation between two points $c(x, x')$ is defined as

$$c(x, x') = \exp\left(- \sum_{i=1}^{p} \frac{|x_i - x'_i|^2}{\delta_i^2} \right) \tag{8}$$

where $\delta = [\delta_1, \delta_2, \cdots, \delta_p]$ are the hyperparameters of the correlation function known as correlation lengths. The correlation matrix A is defined as

$$(A)_{i,j} = c(x^{(i)}, x^{(j)}) + \nu(I)_{i,j} \tag{9}$$

where I is the identity matrix and ν a hyperparameter known as the nugget. The i^{th} element of $t(x) \in \mathbf{R}^{1,n}$ is defined as $(t(x))_i = c(x^{(i)}, x)$. The $n \times r$ matrix $\hat{\beta}$ has the form

$$\hat{\beta} = (H^{\mathrm{T}} A^{-1} H)^{-1} H^{\mathrm{T}} A^{-1} Y^{\mathrm{T}} \tag{10}$$

and $R(x)$ is given by

$$R(x) = h(x)^{\mathrm{T}} - t(x)^{\mathrm{T}} A^{-1} H \tag{11}$$

Finally, $\hat{\Sigma}$ is given by

$$\hat{\Sigma} = (n - q)^{-1} (Y^{\mathrm{T}} - H\hat{\beta})^{\mathrm{T}} A^{-1} (Y^{\mathrm{T}} - H\hat{\beta}) \tag{12}$$

In order to complete the emulator we need to marginalise the hyperparameters δ and ν, which is only possible using numerical integration methods, such as MCMC. We opt however for a more viable alternative, which is the estimation of hyperparameter values that maximise their likelihood, and consider them as the true values. This approach does not account for the uncertainty about the true values of ν and δ, but is computationally more efficient. The likelihood of the hyperparameters is given by

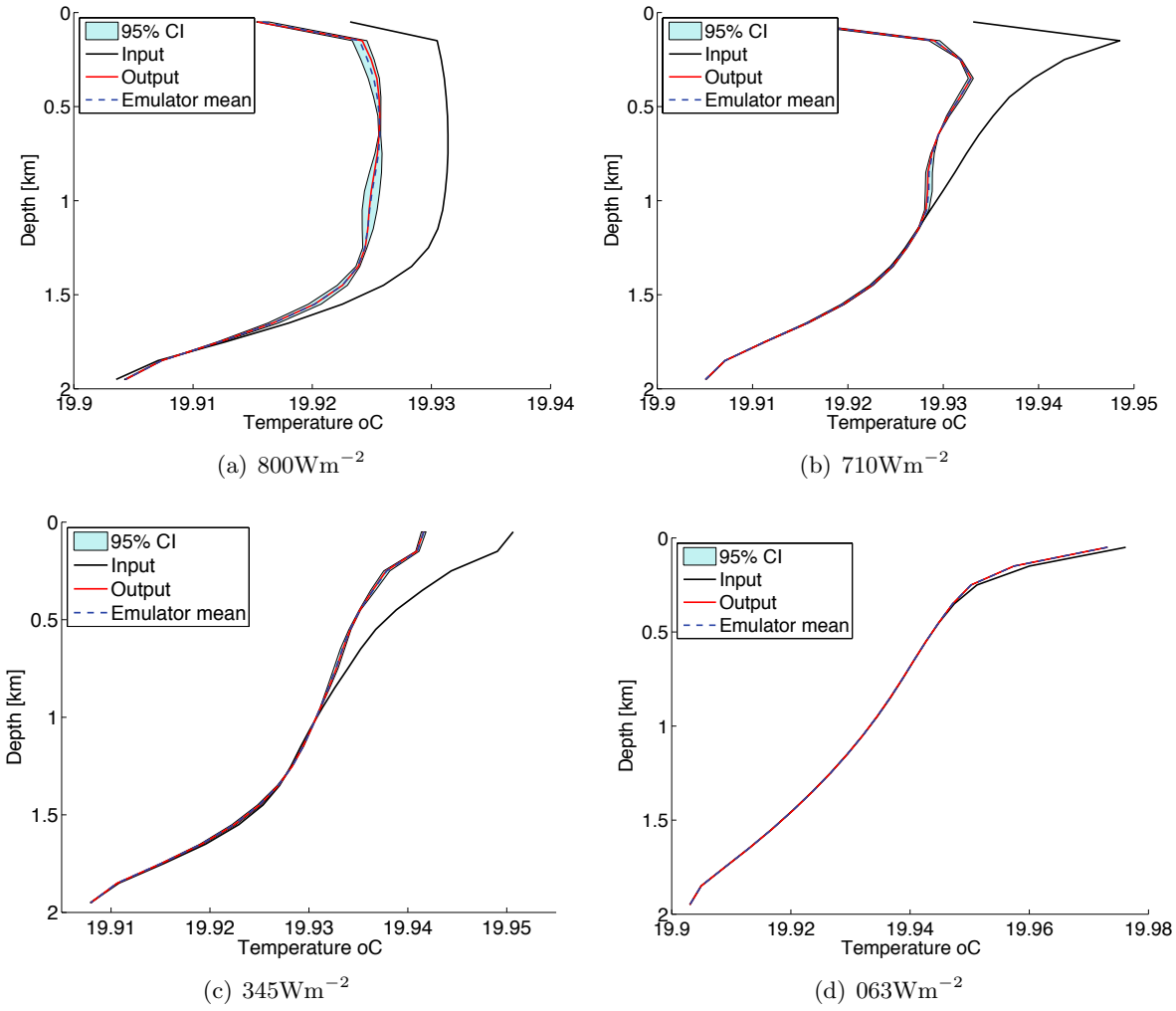
$$\mathcal{L}(\delta, \nu | Y) \propto |\hat{\Sigma}|^{-(n-q)/2} |A|^{-r/2} |H^{\mathrm{T}} A^{-1} H|^{-r/2} \tag{13}$$

Maximising equation 13 can yield the estimates $\hat{\delta}$, $\hat{\nu}$, the substitution of which in eq. 5 provides the emulator's posterior distribution. An example of the emulators' predictions is shown in Figure 6.

3.4 Validation

The next step after building the emulator is to validate its predictions against the simulator's data. The validation scheme we apply is the 'leave one out' validation, in which we attempt to predict a single data point each time, using all the others. The validation criterion we use is the Individual prediction error Bastos and O'Hagan (2009), which is defined as

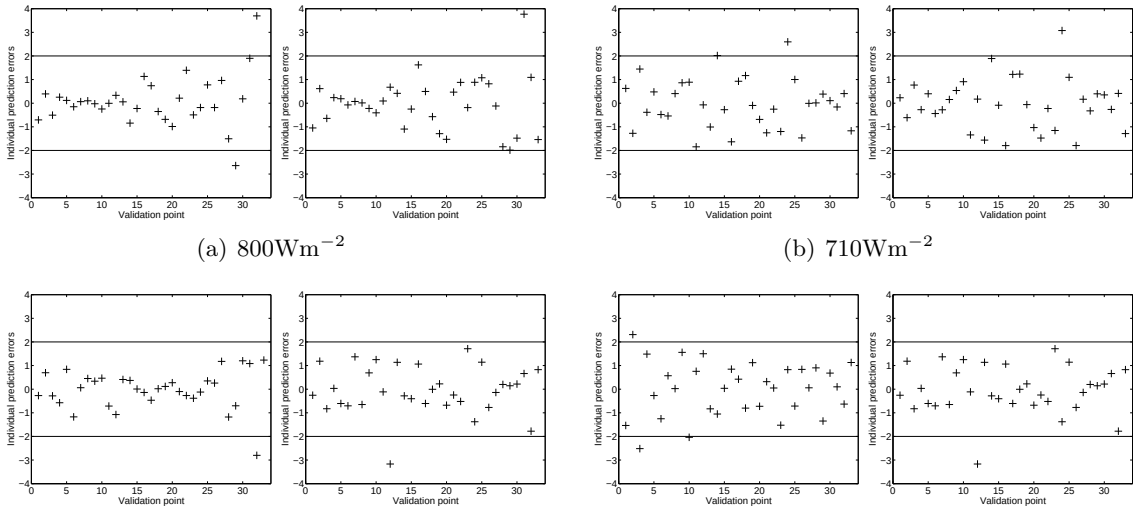
$$\text{IPE} = \frac{y_j^{(i)} - m_j(x^{(i)})}{\sqrt{u_j(x^{(i)}, x^{(i)})}} \tag{14}$$



(a) $800\mathrm{Wm}^{-2}$ (b) $710\mathrm{Wm}^{-2}$

(c) $345\mathrm{Wm}^{-2}$ (d) $063\mathrm{Wm}^{-2}$

Figure 6: Examples of the posterior distribution of the 4 emulators. The black line is the input DT profile and the red line is the output of the model with the respective forcing. The blue dashed line is the emulator's posterior mean the shaded area represents its 95% posterior variance.



(a) 800Wm^{-2} (b) 710Wm^{-2}

(c) 345Wm^{-2} (d) 063Wm^{-2}

Figure 7: Leave one out validation for the 4 emulators. The two sub-panels in each of the 4 panels correspond to the 2 most significant PCA components.

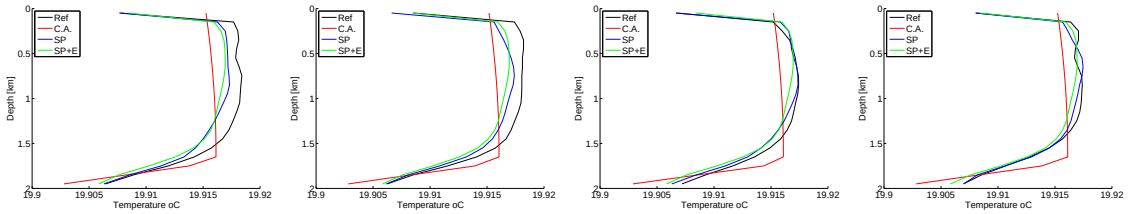
where $i \in [1, n]$ indexes the validation data point and $j \in [1, p]$ indexes one of the 5 PCA components. A large number of IPE values outside the [-2,2] interval denote a conflict between the simulator and the emulator. The results for the 4 emulators and the 2 most significant PCA components are shown in Figure 7. The IPE values largely lie within the desired [-2,2] interval, which suggests a good match between the emulator and the simulator. The IPE scores for the remaining three PCA components (not shown) are also similar.

4 Results

In this section we present the results from our simulation experiment. We compare 4 different models: the Reference model described in Section 1 is shown with a black line and the Coarse model with the Convective adjustment in red. The Super-parameterised model of Section 2 is shown in blue. The Super-parameterised model that uses the emulators is shown in cyan. The results shown are DT profiles for various cells of the Coarse model, obtained after 4 days of cooling.

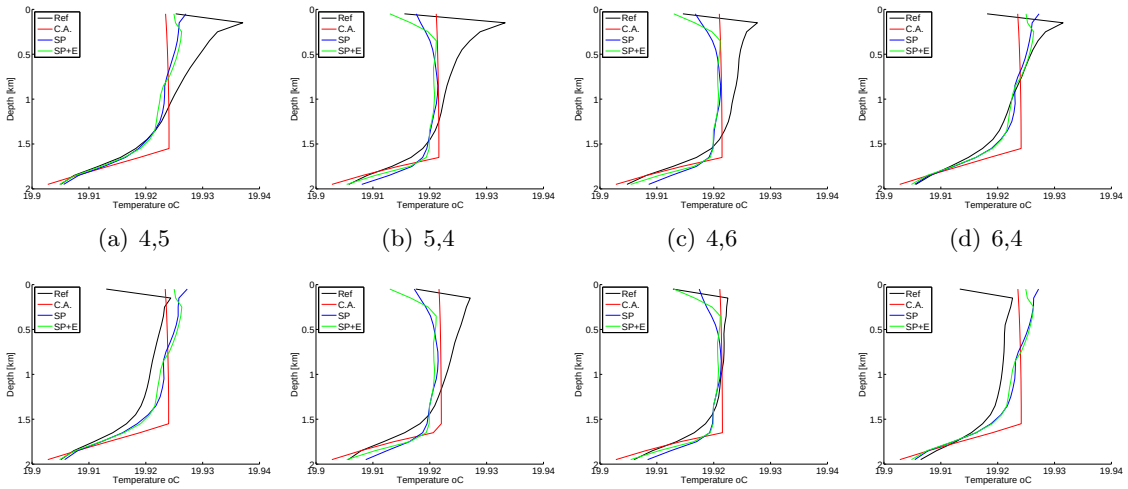
Figure 8 shows the results for the 4 central cells of Figure 2. The Super-parameterised models with and without the emulators capture the curved profiles of the reference model, with greater accuracy than the Coarse convective adjusted model. It's worth mentioning that the latter has an almost linear response for temperatures up to a certain depth.

Figures 9, 10, 11 and 12 show the results from the cells on the edge of the cooling surface. Figure 9 shows the 800 Wm^{-2} non-central cells, and the rest of the Figures show the 710, 345 and 063 Wm^{-2} cells respectively. The results show that reproducing the results of the Reference model in regions closer to the edge of the cooling surface proves a harder job for all approximate models. The



(a) 5,5 (b) 5,6 (c) 6,5 (d) 6,6

Figure 8: DT profiles for the 4 central cells of Figure 2.



(a) 4,5 (b) 5,4 (c) 4,6 (d) 6,4

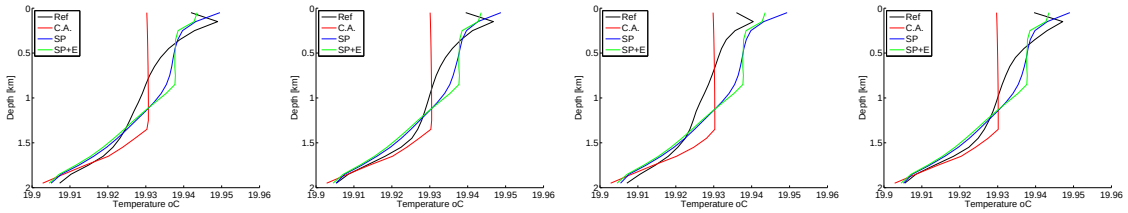
(e) 5,7 (f) 7,5 (g) 6,7 (h) 7,6

Figure 9: DT profiles for the non-central 800 Wm^{-2} cells of Figure 2.

two Super-parameterised models seem to be doing marginally better than the Coarse model, but there is plenty of room for improvement. The encouraging result is that the emulator based Super-parameterised model matches closely the results of the Super-parameterised model that makes no use of emulators, at a fraction of the computational cost. Therefore, future efforts should be directed in creating a Super-parameterised model that is a better approximation of the Reference, before introducing emulation.

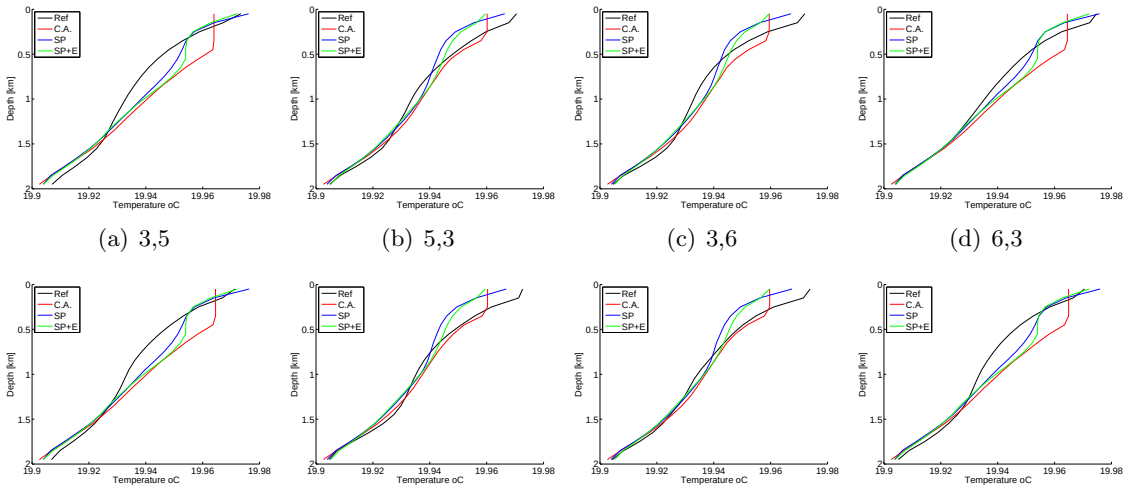
5 Conclusion - Further work

We have explored the application of Super-parameterisation of ocean deep convection using emulation. We proposed a Super-parameterised model that attempts to reproduce the results of a high resolution model at a lower computational cost. The Super-parameterised model consists of two sub-models: one that resolves the convective plumes and one that models the geostrophic eddies. Savings in the computational effort and the novelty in this work come from replacing the Fine models that resolve convection with emulators.



(a) 4,4 (b) 4,7 (c) 7,4 (d) 7,7

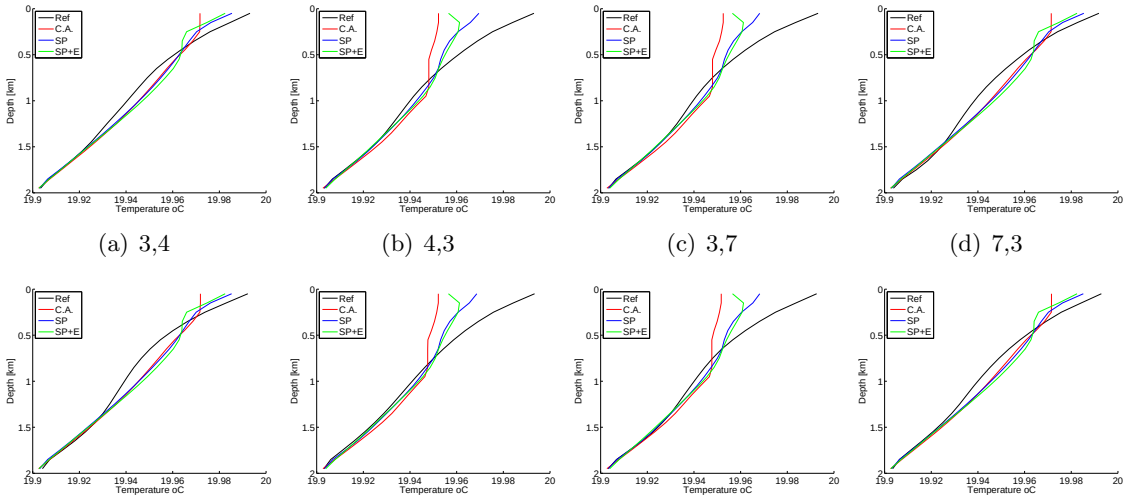
Figure 10: DT profiles for the 710 Wm^{-2} cells of Figure 2.



(a) 3,5 (b) 5,3 (c) 3,6 (d) 6,3

(e) 5,8 (f) 8,5 (g) 6,8 (h) 8,6

Figure 11: DT profiles for the 345 Wm^{-2} cells of Figure 2.



(a) 3,4 (b) 4,3 (c) 3,7 (d) 7,3

(e) 4,8 (f) 8,4 (g) 7,8 (h) 8,7

Figure 12: DT profiles for the 063 Wm^{-2} cells of Figure 2.

The results are encouraging in that the emulation based Super-parameterised model provides an improvement over the 1-d Convective Adjustment scheme that can be found in MITgcm. Especially in the centre of the cooling region the proposed model captures the Reference models DT profiles with relative accuracy. The results on the edge of the cooling surface on the other hand, are harder to reproduce and more work is needed. Further work should aim to improve the Super-parameterised model in these regions prior to introducing emulation.

References

L. S. Bastos and A. O'Hagan. Diagnostics for Gaussian process emulators. *Technometrics*, 51: 425–438, 2009.

Jean Michel Campin, Chris Hill, Helen Jones, and John Marshall. Super-parameterization in ocean modeling: Application to deep convection. *Ocean Modelling*, 36:90–101, 2011.

S. Conti and A. O'Hagan. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140:640–651, 2010.

B. A. Klinger, J. Marshall, and U. Send. Representation of convective plumes by vertical adjustment. *Journal of Geophysical Research*, 101:175–182, 1996.