Super-parameterisation of oceanic deep convection using emulators

Ioannis Andrianakis and Peter Challenor

SIAM conference on Uncertainty Quantification

April 2, 2012
Outline

- Ocean convection.
- What is super-parameterisation?
- The super-parameterised model of ocean convection.
- Emulating the super-parameterised model.
- Results.
- Conclusions.
We study convection in a $40 \times 40 \times 2$ km volume of water, whose surface is cooled on a radius of 10 km.
Multiscale modelling

- Ocean convection.
- Flows over restricted topographies.
- Clouds in atmospheric models.
Multiscale modelling

- Ocean convection.
- Flows over restricted topographies.
- Clouds in atmospheric models.
Multiscale modelling

- Ocean convection.
- Flows over restricted topographies.
- Clouds in atmospheric models.
A high resolution model can resolve both processes but it can be computationally prohibitive. We may be interested in averages of the small scale process.
A high resolution model can resolve both processes but it can be computationally prohibitive. We may be interested in averages of the small scale process.
High resolution model

- 100 m cell edge size
- $400 \times 400 \times 20$ cells
- Non-hydrostatic

- A high resolution model can resolve both processes but it can be computationally prohibitive
- we may be interested in averages of the small scale process.
The high resolution model is replaced by a coarse model that resolves the large scale processes.

- 4 km × 4 km × 100 m cell size
- 10 × 10 × 20 cells
- Hydrostatic

Parameterisation

Coarse model
The high resolution model is replaced by a coarse model that resolves the large scale processes. A parameterisation algorithm that describes the small scale process in terms of the large scale state. The convection algorithm is applied to each of the coarse model’s cells.
Parameterisation

Parameterisation algorithms are computationally cheap but they do not solve the physics of the problem which are normally well known.
The coarse model is the same as before.
The coarse model is the same as before.

The convection parameterisation is replaced with an array of fine scale models that resolve the small scale process.
Super-parameterisation can be slow, if the fine scale models are (even moderately) slow themselves.

but it can benefit from parallelisation.
Computational savings can be achieved by using fast surrogates for the fine scale models.

Campin (2011) used 2 dimensional (sliced) models, to reduce the computational cost.
We propose the replacement of the full 3-D fine models with *emulators*. 
Four approaches

High resolution - *Reference*
Four approaches

High resolution - Reference

Parameterised - Baseline
Four approaches

High resolution - Reference

Parameterised - Baseline

Super-parameterised
Four approaches

High resolution - Reference

Parameterised - Baseline

Super-parameterised

Emulated super-parameterised
Four approaches

High resolution - Reference

Parameterised - Baseline

Super-parameterised

Emulated super-parameterised
Four approaches

High resolution - Reference

Parameterised - Baseline

Super-parameterised

Emulated super-parameterised
Four approaches

High resolution - *Reference*

Parameterised - *Baseline*

All the data are generated with the *MITgcm* model.

Super-parameterised

Emulated super-parameterised

*I. Andrianakis (NOC)*
Super-parameterised model
Super-parameterised model

### Coarse model
- $4 \text{ km} \times 4 \text{ km} \times 100 \text{ m cell size}$
- $10 \times 10 \times 20 \text{ cells}$
- Hydrostatic

### Fine model
- $100 \text{ m cell edge size}$
- $40 \times 40 \times 20 \text{ cells}$
- Non-Hydrostatic
- Array of 100 fine models
Coupling scheme
Coupling scheme

Each of the fine models is propagated for time $t_c$ and their state is stored.
Coupling scheme

Each of the fine models is propagated for time $t_c$ and their state is stored.

The FM temperatures are averaged along the x and y axes, and the resulting profiles are passed on to the coarse model.
Coupling scheme

Each of the fine models is propagated for time $t_c$ and their state is stored.

The coarse model is propagated for time $t_c$, without forcing.

The FM temperatures are averaged along the x and y axes, and the resulting profiles are passed on to the coarse model.
Coupling scheme

Each of the fine models is propagated for time $t_c$ and their state is stored. The coarse model is propagated for time $t_c$, without forcing.

The FM temperatures are averaged along the x and y axes, and the resulting profiles are passed on to the coarse model.

The (stored) FM temperatures are adjusted so that their mean matches that of the coarse model.
The propagation time $t_c$

- Both fine and coarse models are propagated at each iteration for $t_c = 12$ hours. (The process is simulated for a total of 4 days.)

- This interval was selected so that the results of the fine models do not depend on other initial conditions apart from temperature (e.g. velocities, tendencies, pressure).

- This will allow us to build emulators that only depend on the temperature at the location of each fine model.

- And will eliminate the need for storing all the other intermediate values.
The propagation time $t_c$

- Both fine and coarse models are propagated at each iteration for $t_c = 12 \text{ hours}$. (The process is simulated for a total of 4 days.)

- This interval was selected so that the results of the fine models do not depend on other initial conditions apart from temperature (e.g. velocities, tendencies, pressure).

- This will allow us to build emulators that only depend on the temperature at the location of each fine model.

- And will eliminate the need for storing all the other intermediate values.
The propagation time $t_c$

- Both fine and coarse models are propagated at each iteration for $t_c = 12$ hours. (The process is simulated for a total of 4 days.)

- This interval was selected so that the results of the fine models do not depend on other initial conditions apart from temperature (e.g. velocities, tendencies, pressure).

- This will allow us to build emulators that only depend on the temperature at the location of each fine model.

- And will eliminate the need for storing all the other intermediate values.
The propagation time $t_c$

- Both fine and coarse models are propagated at each iteration for $t_c = 12$ hours. (The process is simulated for a total of 4 days.)

- This interval was selected so that the results of the fine models do not depend on other initial conditions apart from temperature (e.g. velocities, tendencies, pressure).

- This will allow us to build emulators that only depend on the temperature at the location of each fine model.

- And will eliminate the need for storing all the other intermediate values.
Segmentation of the cooling function

- The cooling function of each fine model depends on its position in the coarse model’s grid.
- The geometry of the problem gives rise to 4 cooling function levels.
- The emulators that we will build later on, will be conditioned on these 4 levels.
The cooling function of each fine model depends on its position in the coarse model’s grid.

The geometry of the problem gives rise to 4 cooling function levels.

The emulators that we will build later on, will be conditioned on these 4 levels.
The cooling function of each fine model depends on its position in the coarse model’s grid.

The geometry of the problem gives rise to 4 cooling function levels.

The emulators that we will build later on, will be conditioned on these 4 levels.
Emulation
The emulators accept a Depth-Temperature (DT) profile and a forcing level as inputs and estimate the output DT profile of the fine model after 12 hours of cooling.
We run a small section of the reference model that contains all three levels for 4 days.

For each level we save the spatially averaged temperature data every 3 hours (33 input data points per level).

This allows the lateral process to influence the input data.
We run a small section of the reference model that contains all three levels for 4 days.

For each level we save the spatially averaged temperature data every 3 hours (33 input data points per level).

This allows the lateral process to influence the input data.
We run a small section of the reference model that contains all three levels for 4 days.

For each level we save the spatially averaged temperature data every 3 hours (33 input data points per level).

This allows the lateral process to influence the input data.
Output training data

- The data stored in the previous step are fed to the respective fine models to yield 33 **input - output** pairs.
Dimension reduction

- The DT profiles are $20 \times 1$ vectors.

- They are also fairly smooth (correlated).

- In order to decrease the emulators’ dimensions we represent the DT profiles with a small number of basis functions.

- These are provided by PCA.

- 5 principal components can represent more than 0.999 of the variance.
A multi-input multi-output emulator of the fine models is built for each cooling function level according to \cite{Conti2010}.

The emulator accepts a DT profile as input.

And returns a posterior distribution of the fine model’s output after 12 hours of cooling, conditioned on the cooling function.
A multi-input multi-output emulator of the fine models is built for each cooling function level according to [Conti 2010].

The emulator accepts a DT profile as input.

And returns a posterior distribution of the fine model’s output after 12 hours of cooling, conditioned on the cooling function.
A multi-input multi-output emulator of the fine models is built for each cooling function level according to [Conti 2010].

The emulator accepts a DT profile as input.

And returns a posterior distribution of the fine model’s output after 12 hours of cooling, conditioned on the cooling function.
A multi-input multi-output emulator of the fine models is built for each cooling function level according to [Conti 2010].

The emulator accepts a DT profile as input.

And returns a posterior distribution of the fine model’s output after 12 hours of cooling, conditioned on the cooling function.
We validate the emulators using ‘leave one out’ and examine the Individual Prediction Errors [Bastos 2009].

\[ IPE = \frac{y(x) - m(x)}{\sqrt{u(x, x)}} \]

Diagnostics for the most significant principal component only.
We validate the emulators using ‘leave one out’ and examine the Individual Prediction Errors [Bastos 2009].

\[ \text{IPE} = \frac{y(x) - m(x)}{\sqrt{u(x,x)}} \]

Diagnostics for the most significant principal component only.
We validate the emulators using ‘leave one out’ and examine the Individual Prediction Errors [Bastos 2009].

\[ IPE = \frac{y(x) - m(x)}{\sqrt{u(x, x)}} \]

Diagnostics for the most significant principal component only.
- We validate the emulators using ‘leave one out’ and examine the Individual Prediction Errors [Bastos 2009].

\[ \text{IPE} = \frac{y(x) - m(x)}{\sqrt{u(x, x)}} \]

- Diagnostics for the most significant principal component only.
Results
Results

- We present results from the super-parameterised model with and without emulation.

- We compare it with the high resolution (reference) model and the convective adjusted (parameterised) coarse model.

- The results shown are DT profiles in different coarse model cells, after 4 days of cooling.
Results

- We present results from the super-parameterised model with and without emulation.

- We compare it with the high resolution (reference) model and the convective adjusted (parameterised) coarse model.

- The results shown are DT profiles in different coarse model cells, after 4 days of cooling.
Results

- We present results from the super-parameterised model with and without emulation.

- We compare it with the high resolution (reference) model and the convective adjusted (parameterised) coarse model.

- The results shown are DT profiles in different coarse model cells, after 4 days of cooling.
Results

![Graph showing temperature and depth with lines representing different scenarios: Ref, C.A., SP, SP+E.](Image)

- **Ref**
- **C.A.**
- **SP**
- **SP+E**

**Temperature (°C)**

**Depth [km]**

- 0
- 0.5
- 1
- 1.5
- 2
- 19.9
- 19.905
- 19.91
- 19.915
- 19.92

---

I. Andrianakis (NOC)
Results

![Graph showing temperature and depth with lines representing different scenarios: Ref, C.A., SP, and SP+E.](image-url)
Results

![Graph showing temperature vs depth for different scenarios: Ref, C.A., SP, SP+E.](image)

- **Ref**
- **C.A.**
- **SP**
- **SP+E**

- **Temperature in °C**
- **Depth in km**

- I. Andrianakis (NOC)
Results

I. Andrianakis (NOC)
Results
Results

Temperature [°C]

Depth [km]

Ref
C.A.
SP
SP+E

0  0.5  1  1.5  2

19.9  19.92  19.94  19.96  19.98

I. Andrianakis (NOC)
Results

![Graph showing temperature vs depth with different curves for Ref, C.A., SP, and SP+E]
Computational effort

- The computational overhead of the emulated super-parameterised model is negligible.
- Its run time was around 1 minute, same as the convective adjusted model’s, when the high resolution model took almost a day to run.
- There is some overhead associated with obtaining the data needed to train the emulators
- and to actually train them.
Computational effort

- The computational overhead of the emulated super-parameterised model is negligible.

- Its run time was around 1 minute, same as the convective adjusted model’s, when the high resolution model took almost a day to run.

- There is some overhead associated with obtaining the data needed to train the emulators.

- and to actually train them.
Computational effort

- The computational overhead of the emulated super-parameterised model is negligible.

- Its run time was around 1 minute, same as the convective adjusted model’s, when the high resolution model took almost a day to run.

- There is some overhead associated with obtaining the data needed to train the emulators

- and to actually train them.
Computational effort

- The computational overhead of the emulated super-parameterised model is negligible.

- Its run time was around 1 minute, same as the convective adjusted model’s, when the high resolution model took almost a day to run.

- There is some overhead associated with obtaining the data needed to train the emulators.

- and to actually train them.
Conclusion - future work

- We investigated the application of emulators in the super-parameterisation of ocean convection.

- The proposed model approximates reasonably the reference model in the centre of the cooling region.

- The results are poorer on the edge - but so are the results of the convective adjustment scheme.

- More work is needed in reducing the discrepancies between the super-parameterised and the reference models.

- The emulated model matches closely the results of the super-parameterised model.
Conclusion - future work

- We investigated the application of emulators in the super-parameterisation of ocean convection.

- The proposed model approximates reasonably the reference model in the centre of the cooling region.

- The results are poorer on the edge - but so are the results of the convective adjustment scheme.

- More work is needed in reducing the discrepancies between the super-parameterised and the reference models.

- The emulated model matches closely the results of the super-parameterised model.
Conclusion - future work

- We investigated the application of emulators in the super-parameterisation of ocean convection.

- The proposed model approximates reasonably the reference model in the centre of the cooling region.

- The results are poorer on the edge - but so are the results of the convective adjustment scheme.

- More work is needed in reducing the discrepancies between the super-parameterised and the reference models.

- The emulated model matches closely the results of the super-parameterised model.
Conclusion - future work

- We investigated the application of emulators in the super-parameterisation of ocean convection.

- The proposed model approximates reasonably the reference model in the centre of the cooling region.

- The results are poorer on the edge - but so are the results of the convective adjustment scheme.

- More work is needed in reducing the discrepancies between the super-parameterised and the reference models.

- The emulated model matches closely the results of the super-parameterised model.
Conclusion - future work

- We investigated the application of emulators in the super-parameterisation of ocean convection.

- The proposed model approximates reasonably the reference model in the centre of the cooling region.

- The results are poorer on the edge - but so are the results of the convective adjustment scheme.

- More work is needed in reducing the discrepancies between the super-parameterised and the reference models.

- The emulated model matches closely the results of the super-parameterised model.
References


