Bayesian Emulation of Complex Computer Models with Structured Partial Discontinuities

Jonathan Owen

Supervisors: Dr Ian Vernon & Prof Michael Goldstein

Durham University

iCASE PhD sponsored by Emerson

Collaborator: Prof Jonathan Carter, Coventry University

jonathan.owen@durham.ac.uk

April 28, 2021

Durham University

Table of Contents

1. Introduction and Motivation
2. Bayesian Emulation of Functions with Structured Discontinuities
3. TNO OLYMPUS Well Placement Optimisation Challenge
4. Conclusion

Introduction

Aim

Develop **Bayesian emulation methodology** which can handle computer simulators (or functions) with **structured (partial) discontinuities**, for use within **decision support tasks**.

Introduction

Aim

Develop **Bayesian emulation methodology** which can handle computer simulators (or functions) with **structured (partial) discontinuities**, for use within **decision support tasks**.

Limitations to use of computer models include:

- Long simulator evaluation times.
- Complex structure of the models and underlying stochasticity.
- High-dimensional input and output spaces.
- Many sources of uncertainty.

Introduction

Aim

Develop **Bayesian emulation methodology** which can handle computer simulators (or functions) with **structured (partial) discontinuities**, for use within **decision support tasks**.

Limitations to use of computer models include:

- Long simulator evaluation times.
- Complex structure of the models and underlying stochasticity.
- High-dimensional input and output spaces.
- Many sources of uncertainty.

Structured Discontinuities

- Simulators may possess a set of structured discontinuities.
- These are poorly handled by standard emulation methodology, e.g. Gaussian Processes [Kennedy and O'Hagan, 2001].

Motivation

Well Placement Optimisation in the petroleum industry

Maximise the Net Present Value (NPV) as a function of well location which possesses **structured discontinuities** with respect to the **partial fault boundaries**.

Motivation

Well Placement Optimisation in the petroleum industry

Maximise the Net Present Value (NPV) as a function of well location which possesses **structured discontinuities** with respect to the **partial fault boundaries**.

TNO OLYMPUS Field Development Optimisation Challenge [ISAPP, 2018]

Aim is to encourage research and development of technology for optimising production systems under uncertainty.

Motivation

Well Placement Optimisation in the petroleum industry

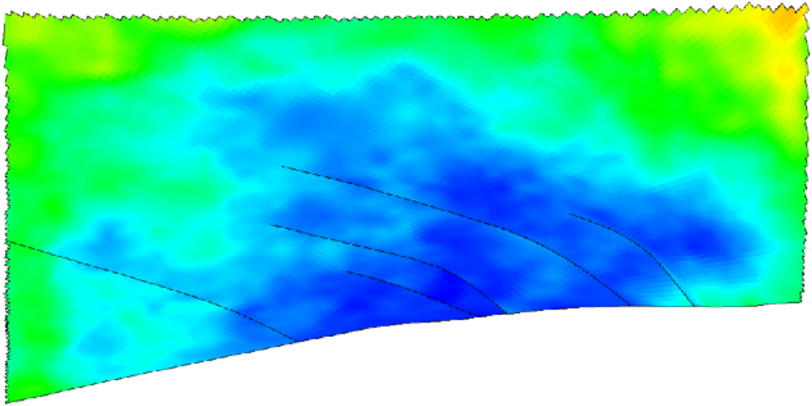
Maximise the Net Present Value (NPV) as a function of well location which possesses **structured discontinuities** with respect to the **partial fault boundaries**.

TNO OLYMPUS Field Development Optimisation Challenge [ISAPP, 2018]

Aim is to encourage research and development of technology for optimising production systems under uncertainty.

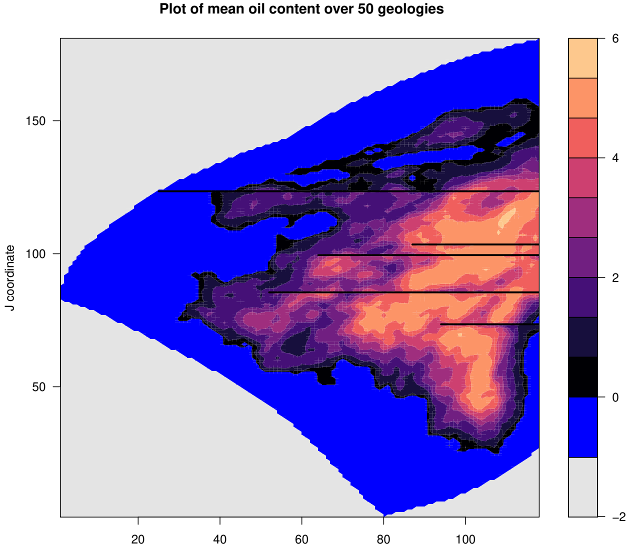
- Optimisation under uncertainty uses **ensembles of models**.
- Current industry approaches use **computationally expensive optimisation algorithms** [Tanaka et al., 2018].
- Decision support incorporating emulators has the ability to **fully explore the decision space**.

TNO OLYMPUS Reservoir Model [TNO, 2017]



- OLYMPUS is a fictitious oil reservoir model.
- Geological uncertainty is represented by an **ensemble of 50 model realisations**.
- Simulations are **computationally expensive**.

Transformed TNO OLYMPUS Map



Bayesian Emulation [Vernon et al., 2010]

Bayesian Emulation

A Bayesian emulator is a stochastic belief specification for a deterministic function that provides a fast and efficient statistical approximation, yielding predictions for as yet unevaluated parameter settings and a corresponding statement of the uncertainty.

Bayesian Emulation [Vernon et al., 2010]

Bayesian Emulation

A Bayesian emulator is a stochastic belief specification for a deterministic function that provides a fast and efficient statistical approximation, yielding predictions for as yet unevaluated parameter settings and a corresponding statement of the uncertainty.

Emulator Equation

$$f(\mathbf{x}) = \sum_j \beta_j g_j(\mathbf{x}_A) + u(\mathbf{x}_A) + w(\mathbf{x})$$

Bayesian Emulation [Vernon et al., 2010]

Bayesian Emulation

A Bayesian emulator is a stochastic belief specification for a deterministic function that provides a fast and efficient statistical approximation, yielding predictions for as yet unevaluated parameter settings and a corresponding statement of the uncertainty.

Emulator Equation

$$f(\mathbf{x}) = \sum_j \beta_j g_j(\mathbf{x}_A) + u(\mathbf{x}_A) + w(\mathbf{x})$$

The **squared exponential covariance function** for $\mathbf{x} \in \mathbb{R}^p$:

$$\mathrm{Cov}[u(\mathbf{x}_A), u(\mathbf{x}'_A)] = \sigma_u^2 \exp\left\{-\left(\mathbf{x}_A - \mathbf{x}'_A\right)^{\mathrm{T}} \Sigma_{p\mathrm{D}}^{-1} \left(\mathbf{x}_A - \mathbf{x}'_A\right)\right\}$$

Bayesian Emulation [Vernon et al., 2010]

Bayesian Emulation

A Bayesian emulator is a stochastic belief specification for a deterministic function that provides a fast and efficient statistical approximation, yielding predictions for as yet unevaluated parameter settings and a corresponding statement of the uncertainty.

Emulator Equation

$$f(\mathbf{x}) = \sum_j \beta_j g_j(\mathbf{x}_A) + u(\mathbf{x}_A) + w(\mathbf{x})$$

The **squared exponential covariance function** for $\mathbf{x} \in \mathbb{R}^p$:

$$\text{Cov}[u(\mathbf{x}_A), u(\mathbf{x}'_A)] = \sigma_u^2 \exp\left\{-\frac{\|\mathbf{x}_A - \mathbf{x}'_A\|^2}{\theta^2}\right\}$$

Bayesian Emulation with Structured Discontinuities

Challenges emulating functions with discontinuities

- Diagnostics will be severely violated in the vicinity of the discontinuity.
- Emulator parameter estimation may produce strange results leading to global issues.

Bayesian Emulation with Structured Discontinuities

Challenges emulating functions with discontinuities

- Diagnostics will be severely violated in the vicinity of the discontinuity.
- Emulator parameter estimation may produce strange results leading to global issues.

Solution: Embed the input parameter space in higher dimensions

- Define the embedding function $\mathbf{v}(\mathbf{x})$ with the form chosen to characterise the discontinuities, e.g. torn along the discontinuities.
- Embed $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p$ in higher dimensional space $\mathbf{v} = \mathbf{v}(\mathbf{x}) \in \mathcal{V} \subset \mathbb{R}^q$, where $q > p$.
- Emulation proceeds but with input $\mathbf{v}(\mathbf{x})$ and variance matrix, $\Sigma_{q\mathrm{D}}$.

Bayesian Emulation with Structured Discontinuities

Challenges emulating functions with discontinuities

- Diagnostics will be severely violated in the vicinity of the discontinuity.
- Emulator parameter estimation may produce strange results leading to global issues.

Solution: Embed the input parameter space in higher dimensions

- Define the embedding function $\mathbf{v}(\mathbf{x})$ with the form chosen to characterise the discontinuities, e.g. torn along the discontinuities.
- Embed $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p$ in higher dimensional space $\mathbf{v} = \mathbf{v}(\mathbf{x}) \in \mathcal{V} \subset \mathbb{R}^q$, where $q > p$.
- Emulation proceeds but with input $\mathbf{v}(\mathbf{x})$ and variance matrix, $\Sigma_{q\mathrm{D}}$.

The **squared exponential covariance function** is modified to:

$$\mathrm{Cov}[u(\mathbf{v}), u(\mathbf{v}')] = \sigma_u^2 \exp\left\{-\left(\mathbf{v} - \mathbf{v}'\right)^{\mathrm{T}} \Sigma_{q\mathrm{D}}^{-1} \left(\mathbf{v} - \mathbf{v}'\right)\right\}$$

2D Example – Setup

Let $(x, y) \in [0,2] \times [0,2]$.

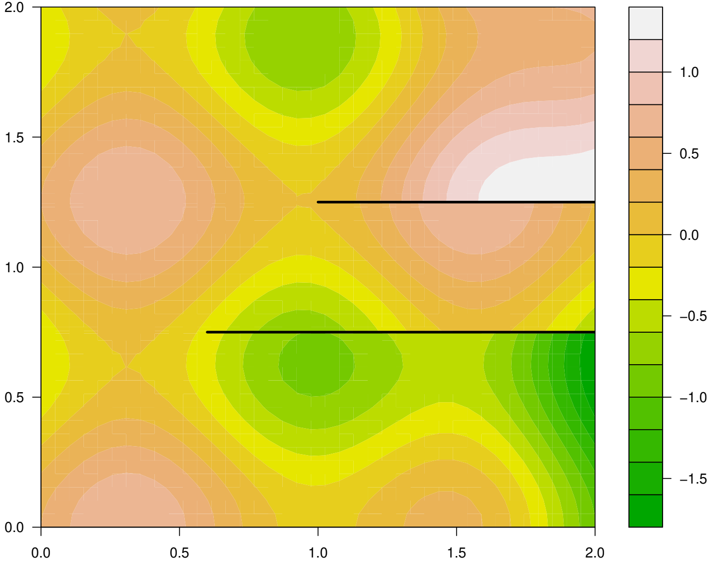
$$\begin{aligned} f(x,y) &= 0.4\sin(5x) + 0.4\cos(5y) \\ &\quad + 1.2(x-1)^2 \mathbb{1}_{\{x>1\}} \mathbb{1}_{\{y>1.25\}} \\ &\quad - 0.6(x-0.6)^2 \mathbb{1}_{\{x>0.6\}} \mathbb{1}_{\{y<0.75\}} \end{aligned}$$

Define the embedding function for $z = v(x,y)$ to be:

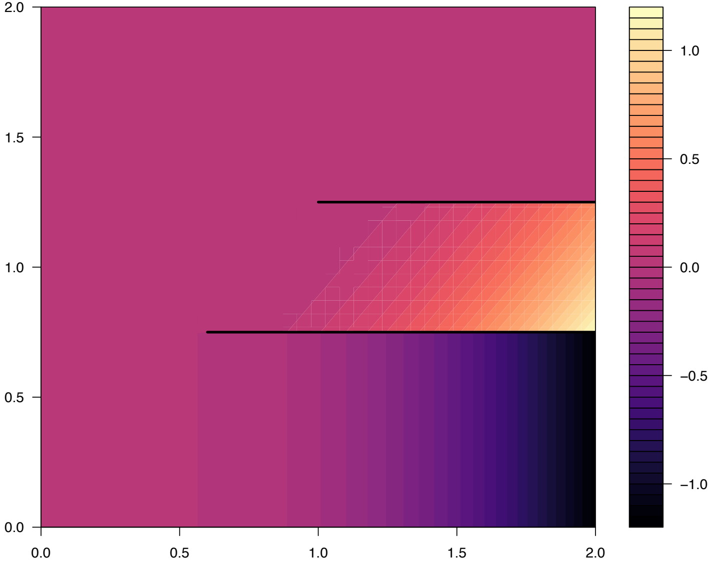
$$v(x,y) = \begin{cases} 0 & \text{if } y \geq 1.25 \\ 0.6\left(x - \left(0.6 + 0.8\left(y - 0.75\right)\right)\right)^2 \cdot \\ \qquad \mathbb{1}_{\{x>0.6+0.8(y-0.75)\}} & \text{if } 0.75 \leq y < 1.25 \\ -0.6\left(x-0.6\right)^2 \cdot \mathbb{1}_{\{x>0.6\}} & \text{if } y < 0.75 \end{cases}$$

$$\Sigma_{\mathrm{3D}} = \begin{pmatrix} \theta^2 & 0 & 0 \\ 0 & \theta^2 & 0 \\ 0 & 0 & \theta^2 \end{pmatrix}$$

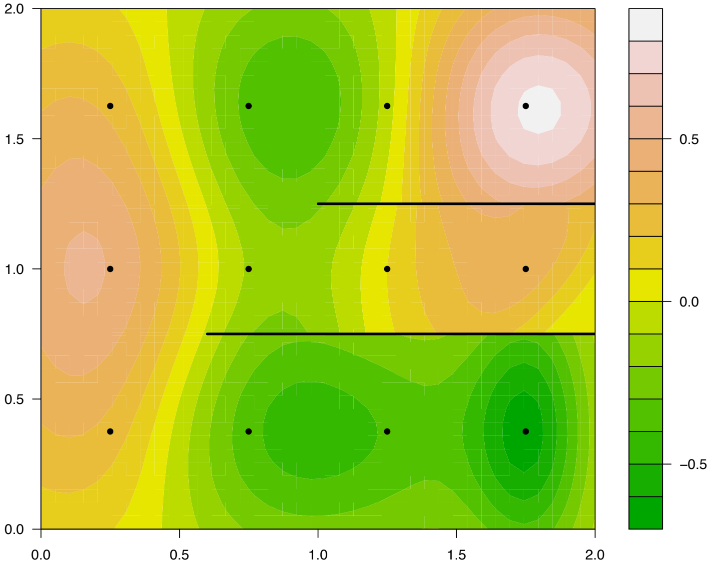
2D Example – True Function



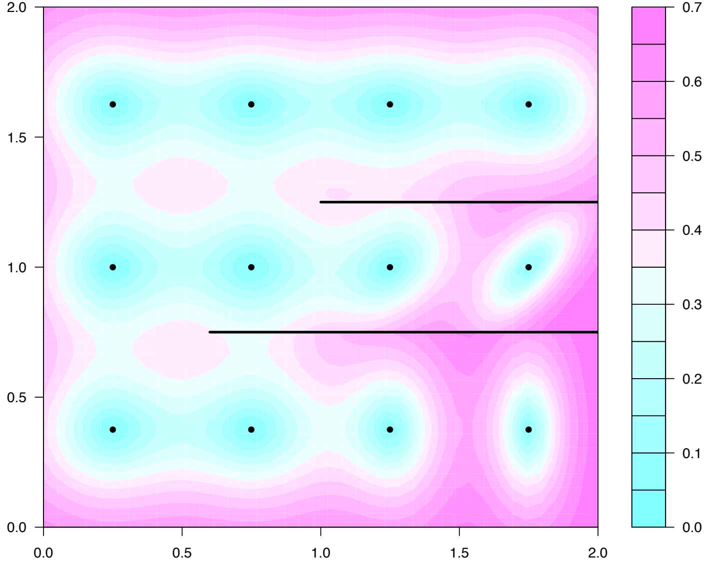
2D Example – Embedding Surface



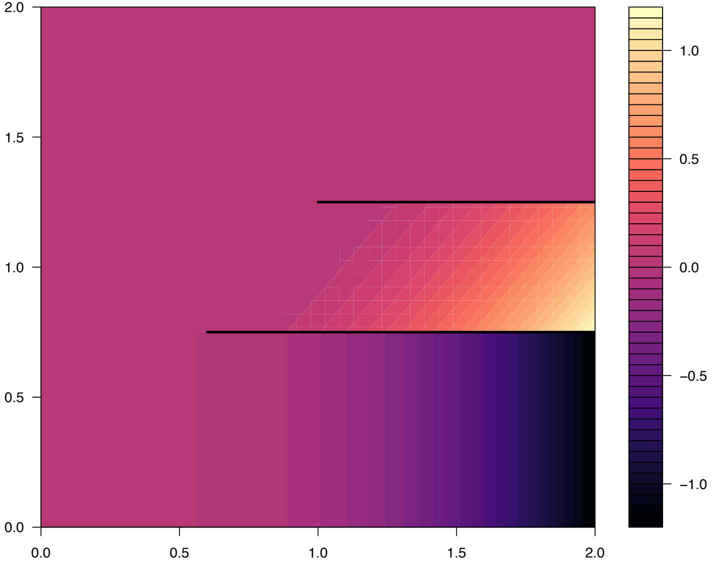
2D Example – Emulator Expectation



2D Example – Emulator Variance



2D Example – Embedding Surface



Embedding a 2D input parameter space in 3D

- Let $\mathbf{x} = \binom{x}{y} \in \mathcal{X} \subset \mathbb{R}^2$ and $\Sigma_{2\mathrm{D}} = \begin{pmatrix} \theta^2 & 0 \\ 0 & \theta^2 \end{pmatrix}$ for all $\mathbf{x} \in \mathcal{X}$.
- Define the embedding $\mathbf{v} = \mathbf{v}(\mathbf{x}) = \begin{pmatrix} x \\ y \\ v(x,y) \end{pmatrix} \in \mathcal{V} \subset \mathbb{R}^3$.
- Consider a point $\mathbf{x}_0 = (x_0, y_0) \in \mathcal{X}$.
- The partial derivatives are: $v_x = \frac{\partial v}{\partial x}\Big|_{\mathbf{x}=\mathbf{x}_0}$ and $v_y = \frac{\partial v}{\partial y}\Big|_{\mathbf{x}=\mathbf{x}_0}$, and can be used to construct the tangent plane at $v(x_0, y_0)$.

Embedding a 2D input parameter space in 3D

- Let $\mathbf{x} = \left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right) \in \mathcal{X} \subset \mathbb{R}^2$ and $\Sigma_{\mathrm{2D}} = \begin{pmatrix} \theta^2 & 0 \\ 0 & \theta^2 \end{pmatrix}$ for all $\mathbf{x} \in \mathcal{X}$.
- Define the embedding $\mathbf{v} = \mathbf{v}(\mathbf{x}) = \left(\begin{smallmatrix} x \\ y \\ v(x,y) \end{smallmatrix}\right) \in \mathcal{V} \subset \mathbb{R}^3$.
- Consider a point $\mathbf{x}_0 = (x_0, y_0) \in \mathcal{X}$.
- The partial derivatives are: $v_x = \frac{\partial v}{\partial x}\Big|_{\mathbf{x}=\mathbf{x}_0}$ and $v_y = \frac{\partial v}{\partial y}\Big|_{\mathbf{x}=\mathbf{x}_0}$, and can be used to construct the tangent plane at $v(x_0, y_0)$.

Aim

Construct Σ_{3D} which induces Σ_{2D}, using the tangent plane to approximate $v(x, y)$.

Embedding a 2D input parameter space in 3D

- Let $\mathbf{x} = \binom{x}{y} \in \mathcal{X} \subset \mathbb{R}^2$ and $\Sigma_{\mathrm{2D}} = \begin{pmatrix} \theta^2 & 0 \\ 0 & \theta^2 \end{pmatrix}$ for all $\mathbf{x} \in \mathcal{X}$.
- Define the embedding $\mathbf{v} = \mathbf{v}(\mathbf{x}) = \begin{pmatrix} x \\ y \\ v(x,y) \end{pmatrix} \in \mathcal{V} \subset \mathbb{R}^3$.
- Consider a point $\mathbf{x}_0 = (x_0, y_0) \in \mathcal{X}$.
- The partial derivatives are: $v_x = \frac{\partial v}{\partial x}\Big|_{\mathbf{x}=\mathbf{x}_0}$ and $v_y = \frac{\partial v}{\partial y}\Big|_{\mathbf{x}=\mathbf{x}_0}$, and can be used to construct the tangent plane at $v(x_0, y_0)$.

Aim

Construct Σ_{3D} which induces Σ_{2D}, using the tangent plane to approximate $v(x, y)$.

Work in the orthonormal w-basis defined by the tangent plane:

$$\mathbf{w}_1 \propto v_x \mathbf{e}_x + v_y \mathbf{e}_y + (v_x^2 + v_y^2)\mathbf{e}_z$$

$$\mathbf{w}_2 \propto -v_y \mathbf{e}_x + v_x \mathbf{e}_y$$

$$\mathbf{w}_3 \propto -v_x \mathbf{e}_x - v_y \mathbf{e}_y + \mathbf{e}_z$$

Embedding a 2D input parameter space in 3D

Proposed form of Σ_{3D} by SVD:

$$\Sigma_{\mathrm{3D}} = \alpha_1^2 \mathbf{w}_1 \mathbf{w}_1^{\mathrm{T}} + \alpha_2^2 \mathbf{w}_2 \mathbf{w}_2^{\mathrm{T}} + \alpha_3^2 \mathbf{w}_3 \mathbf{w}_3^{\mathrm{T}}$$

Embedding a 2D input parameter space in 3D

Proposed form of Σ_{3D} by SVD:

$$\Sigma_{\mathrm{3D}} = \alpha_1^2 \mathbf{w}_1 \mathbf{w}_1^{\mathrm{T}} + \alpha_2^2 \mathbf{w}_2 \mathbf{w}_2^{\mathrm{T}} + \alpha_3^2 \mathbf{w}_3 \mathbf{w}_3^{\mathrm{T}}$$

Let $\mathbf{u} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ be in the tangent plane of $v(\mathbf{x})$ at $\mathbf{x}_0$ where $\mathbf{u} = A\mathbf{x} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ v_x & v_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$. Recentre at $\mathbf{x}_0 = \mathbf{0}$ and $\mathbf{u}_0 = \mathbf{v}(\mathbf{x}_0) = \mathbf{0}$.
Mahalanobis distance preservation requires: $\mathbf{x}^{\mathrm{T}} \Sigma_{\mathrm{2D}}^{-1} \mathbf{x} = \mathbf{u}^{\mathrm{T}} \Sigma_{\mathrm{3D}}^{-1} \mathbf{u}$.

Embedding a 2D input parameter space in 3D

Proposed form of Σ_{3D} by SVD:

$$\Sigma_{\mathrm{3D}} = \alpha_1^2 \mathbf{w}_1 \mathbf{w}_1^{\mathrm{T}} + \alpha_2^2 \mathbf{w}_2 \mathbf{w}_2^{\mathrm{T}} + \alpha_3^2 \mathbf{w}_3 \mathbf{w}_3^{\mathrm{T}}$$

Let $\mathbf{u} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ be in the tangent plane of $v(\mathbf{x})$ at $\mathbf{x}_0$ where $\mathbf{u} = A\mathbf{x} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ v_x & v_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$. Recentre at $\mathbf{x}_0 = \mathbf{0}$ and $\mathbf{u}_0 = \mathbf{v}(\mathbf{x}_0) = \mathbf{0}$.
Mahalanobis distance preservation requires: $\mathbf{x}^{\mathrm{T}} \Sigma_{\mathrm{2D}}^{-1} \mathbf{x} = \mathbf{u}^{\mathrm{T}} \Sigma_{\mathrm{3D}}^{-1} \mathbf{u}$.

$$\implies \qquad \Sigma_{\mathrm{2D}}^{-1} = A^{\mathrm{T}} \Sigma_{\mathrm{3D}}^{-1} A$$

$$\implies \quad \begin{pmatrix} \frac{1}{\theta^2} & 0 \\ 0 & \frac{1}{\theta^2} \end{pmatrix} = \frac{1 + r^2}{\alpha_1^2 r^2} \begin{pmatrix} v_x^2 & v_x v_y \\ v_x v_y & v_y^2 \end{pmatrix} + \frac{1}{\alpha_2^2 r^2} \begin{pmatrix} v_y^2 & -v_x v_y \\ -v_x v_y & v_x^2 \end{pmatrix}$$

where $r^2 = v_x^2 + v_y^2$

Embedding a 2D input parameter space in 3D

Proposed form of $\Sigma_{3\mathrm{D}}$ by SVD:

$$\Sigma_{3\mathrm{D}} = \alpha_1^2 \mathbf{w}_1 \mathbf{w}_1^{\mathrm{T}} + \alpha_2^2 \mathbf{w}_2 \mathbf{w}_2^{\mathrm{T}} + \alpha_3^2 \mathbf{w}_3 \mathbf{w}_3^{\mathrm{T}}$$

Let $\mathbf{u} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ be in the tangent plane of $v(\mathbf{x})$ at $\mathbf{x}_0$ where $\mathbf{u} = A\mathbf{x} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ v_x & v_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$. Recentre at $\mathbf{x}_0 = \mathbf{0}$ and $\mathbf{u}_0 = \mathbf{v}(\mathbf{x}_0) = \mathbf{0}$.
Mahalanobis distance preservation requires: $\mathbf{x}^{\mathrm{T}} \Sigma_{2\mathrm{D}}^{-1} \mathbf{x} = \mathbf{u}^{\mathrm{T}} \Sigma_{3\mathrm{D}}^{-1} \mathbf{u}$.

$$\Sigma_{3\mathrm{D}} = \begin{pmatrix} \theta^2 + \dfrac{\alpha_3^2 v_x^2}{1+r^2} & \dfrac{\alpha_3^2 v_x v_y}{1+r^2} & v_x\left(\theta^2 - \dfrac{\alpha_3^2}{1+r^2}\right) \\ \dfrac{\alpha_3^2 v_x v_y}{1+r^2} & \theta^2 + \dfrac{\alpha_3^2 v_y^2}{1+r^2} & v_y\left(\theta^2 - \dfrac{\alpha_3^2}{1+r^2}\right) \\ v_x\left(\theta^2 - \dfrac{\alpha_3^2}{1+r^2}\right) & v_y\left(\theta^2 - \dfrac{\alpha_3^2}{1+r^2}\right) & \theta^2 r^2 + \dfrac{\alpha_3^2}{1+r^2} \end{pmatrix}$$

Non-Stationary Emulators

- It is clear that the form of Σ_{3D} is **not constant over the 3D input space** for general embedding functions, $v(x, y)$.
- We need to construct **non-stationary emulators** since $\Sigma_{\mathrm{3D}}(\mathbf{x})$.
- Non-stationary emulators of [Dunlop et al., 2018] are employed to accommodate for this.

Non-Stationary Emulators

- It is clear that the form of Σ_{3D} is **not constant over the 3D input space** for general embedding functions, $v(x, y)$.
- We need to construct **non-stationary emulators** since $\Sigma_{\mathrm{3D}}(\mathbf{x})$.
- Non-stationary emulators of [Dunlop et al., 2018] are employed to accommodate for this.

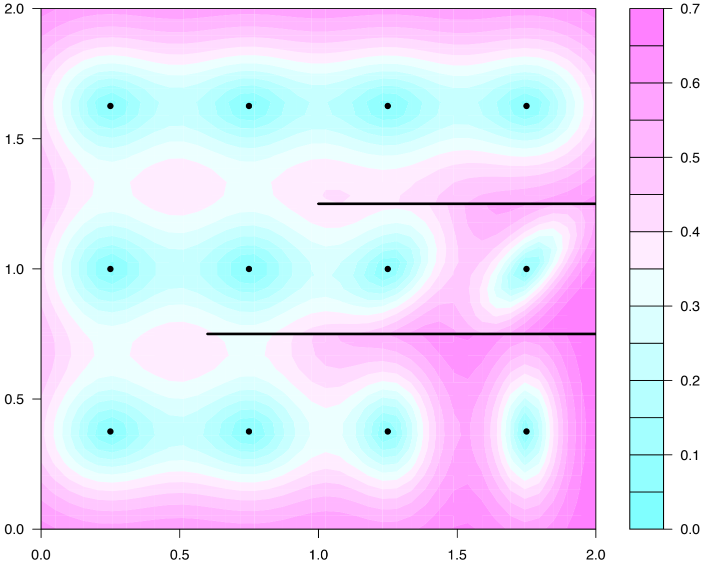
Quadratic Form, where $\mathbf{v} = \mathbf{v}(\mathbf{x})$ and $\Sigma(\mathbf{x}) = \Sigma_{\mathrm{3D}}(\mathbf{x})$

$$Q(\mathbf{v}, \mathbf{v}') = (\mathbf{v} - \mathbf{v}')^{\mathrm{T}} \left(\frac{\Sigma(\mathbf{x}) + \Sigma(\mathbf{x}')}{2} \right) (\mathbf{v} - \mathbf{v}')$$

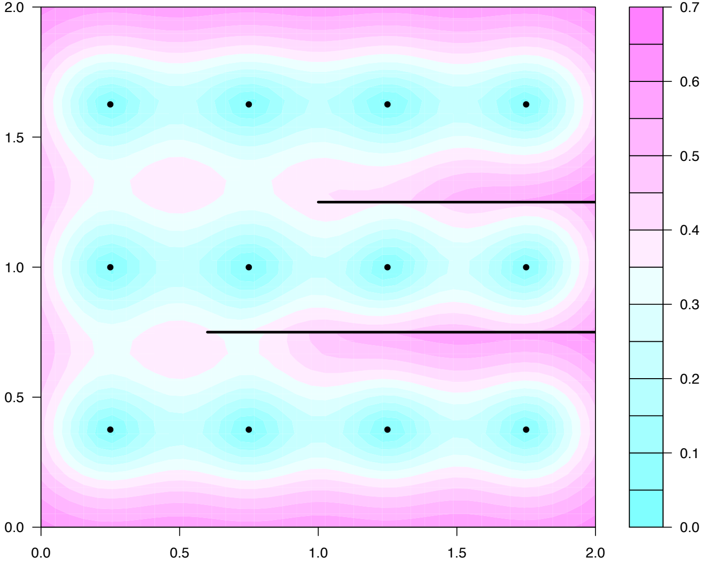
Non-Stationary Squared Exponential Covariance Function

$$\mathrm{Cov}[u(\mathbf{v}), u(\mathbf{v}')] = \sigma_u^2 \frac{2^{\frac{3}{2}} \left|\Sigma(\mathbf{x})\right|^{\frac{1}{4}} \left|\Sigma(\mathbf{x}')\right|^{\frac{1}{4}}}{\left|\Sigma(\mathbf{x}) + \Sigma(\mathbf{x}')\right|^{\frac{1}{2}}} \exp\left\{-Q(\mathbf{v}, \mathbf{v}')\right\}$$

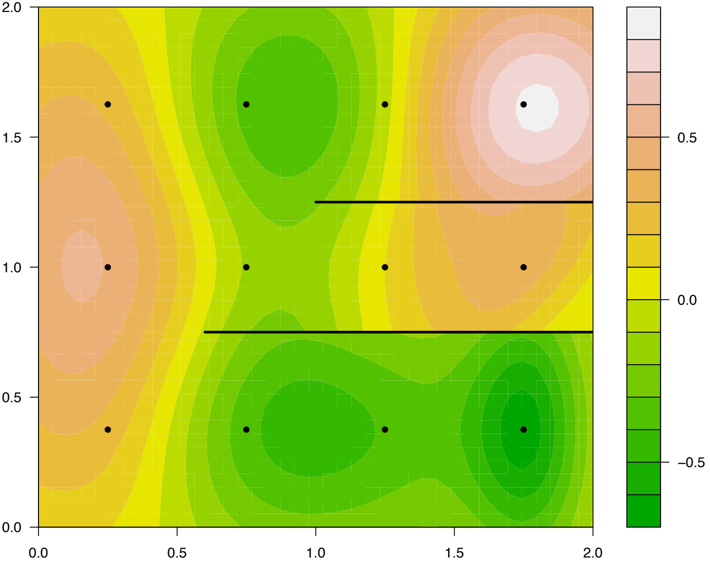
2D Emulator Variance NOT using embedding correction



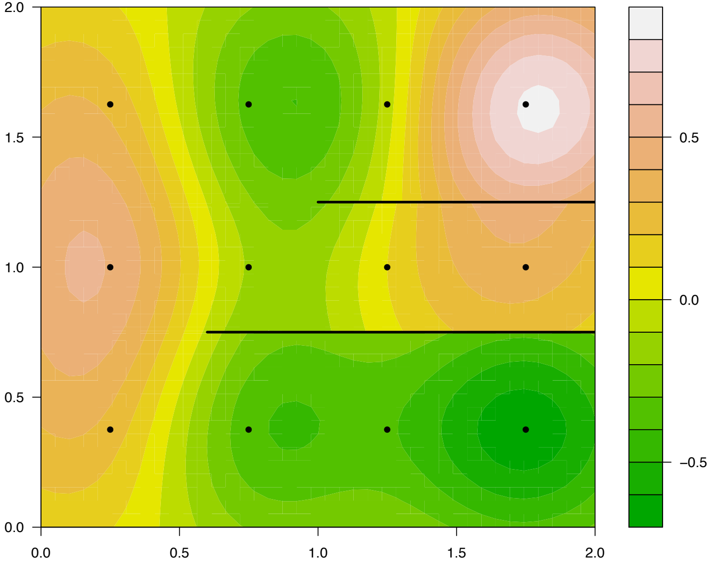
2D Emulator Variance using embedding correction



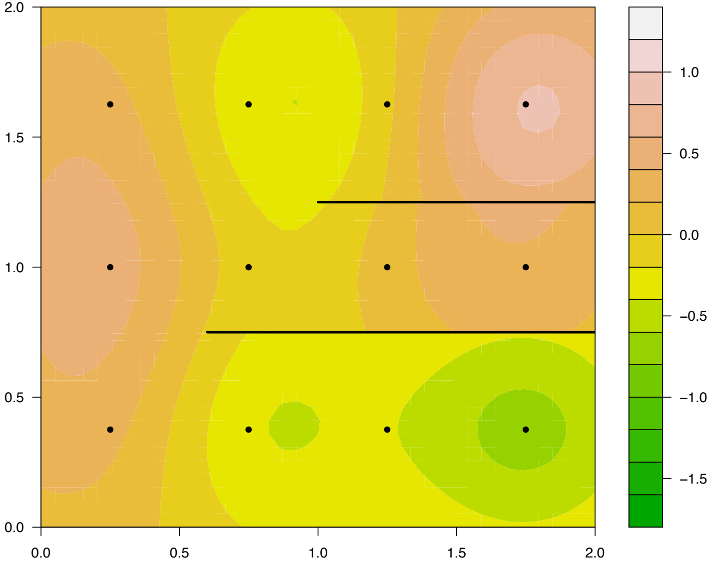
2D Emulator Expectation NOT using embedding correction



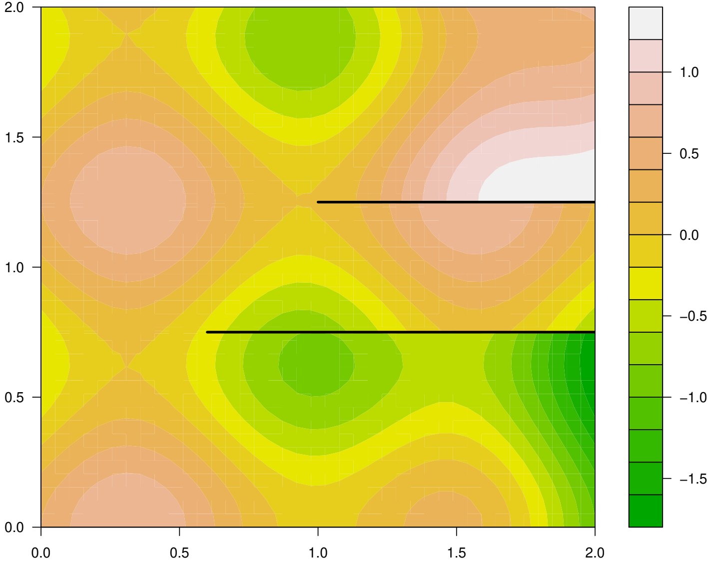
2D Emulator Expectation using embedding correction



2D Emulator Expectation using embedding correction



2D Example – True Function



Well Placement Optimisation

Aim

Design a well placement strategy including each well location, trajectory and type, as well as the number of wells and drilling sequence, for the TNO OLYMPUS oil reservoir model which maximises the expected NPV over the field lifetime, under the uncertainty captured by the 50 geological realisations.

Well Placement Optimisation

Aim

Design a well placement strategy including each well location, trajectory and type, as well as the number of wells and drilling sequence, for the TNO OLYMPUS oil reservoir model which maximises the expected NPV over the field lifetime, under the uncertainty captured by the 50 geological realisations.

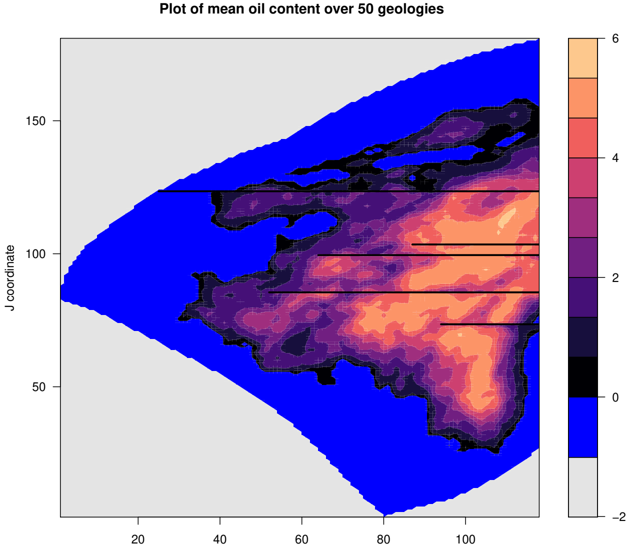
Objective Function

$$\mathrm{NPV}_j(\mathbf{x}) = \sum_{i=1}^{N_t} \frac{R_j(\mathbf{x}, t_i)}{(1+d)^{\frac{t_i}{\tau}}}$$

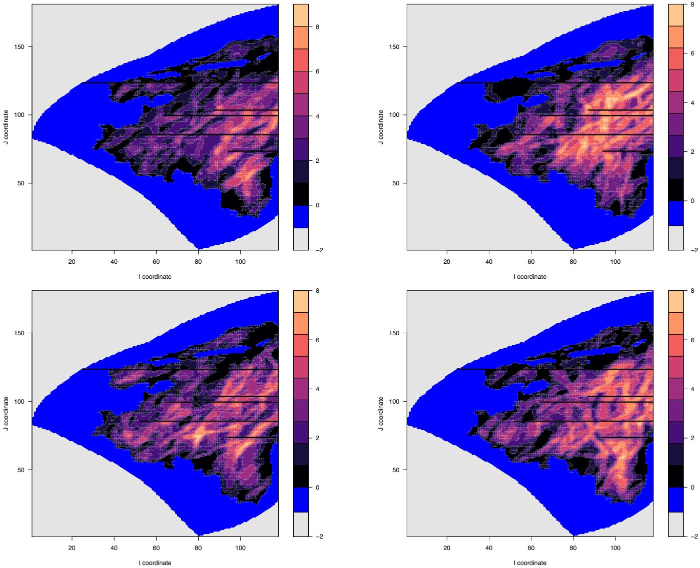
$$\begin{aligned}\mathbb{E}[\mathrm{NPV}](\mathbf{x}) &\approx \overline{\mathrm{NPV}}(\mathbf{x}) \\ &= \frac{1}{n}\sum_{j=1}^{n} \mathrm{NPV}_j(\mathbf{x})\end{aligned}$$

$$\begin{aligned}R_j(\mathbf{x}, t_i) = {} & Q_{j,op}(\mathbf{x}, t_i) \cdot r_{op} - Q_{j,wp}(\mathbf{x}, t_i) \cdot r_{wp} - Q_{j,wi}(\mathbf{x}, t_i) \cdot r_{wi} \\ & - P_j(\mathbf{x}, t_i) - D_j(\mathbf{x}, t_i)\end{aligned}$$

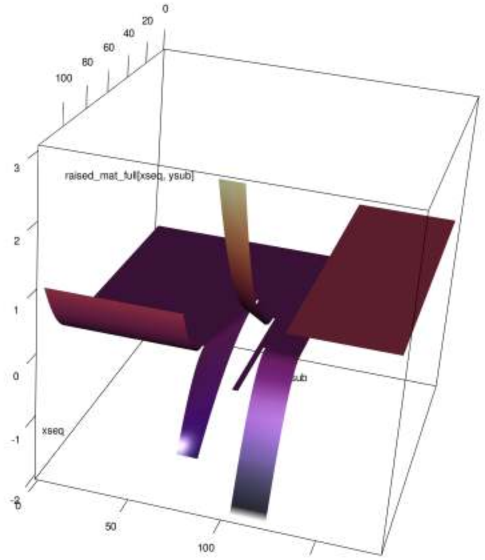
Transformed TNO OLYMPUS Map



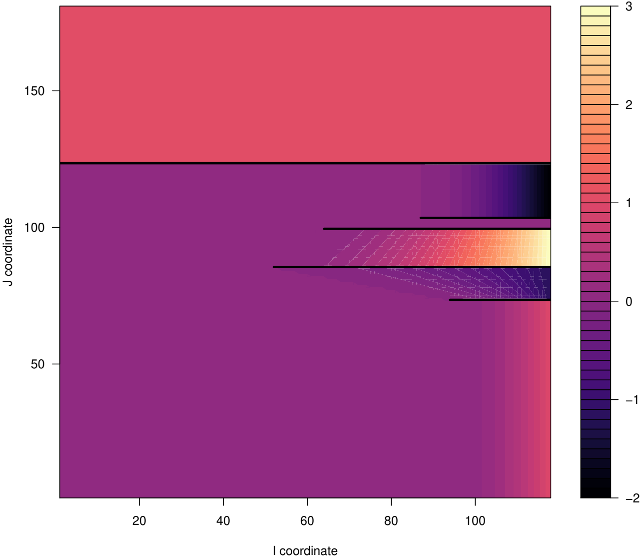
OLYMPUS Model Ensemble – Oil Content Maps



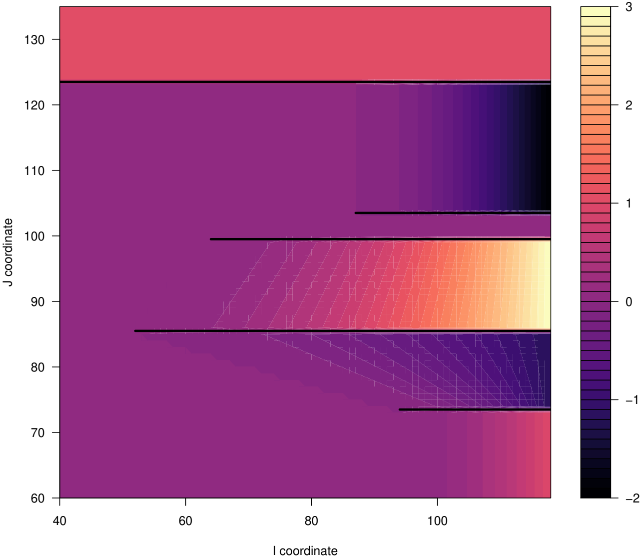
OLYMPUS – Embedding Surface



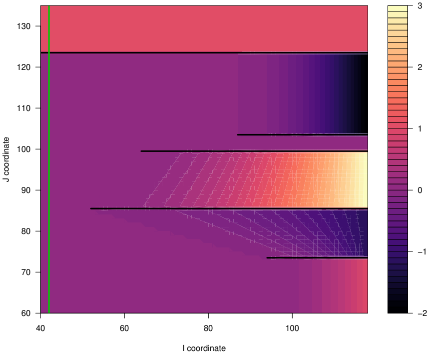
OLYMPUS – Embedding Surface

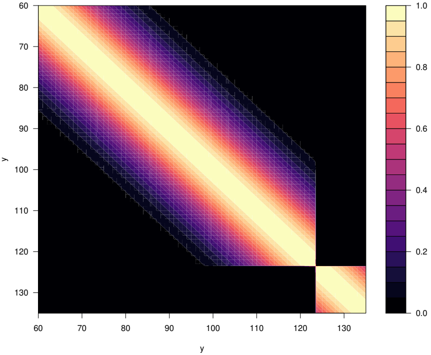


OLYMPUS – Embedding Surface

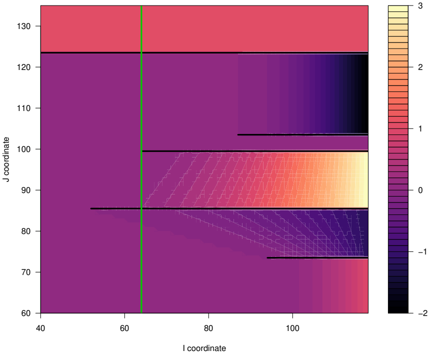


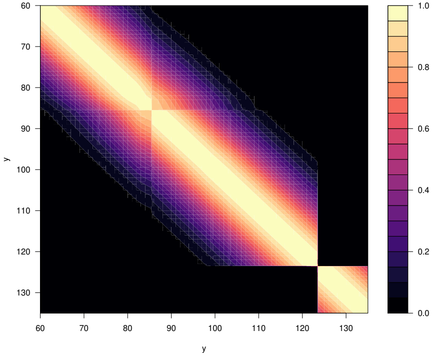
Covariance Matrix for Vertical Slices through the Location Parameter Space



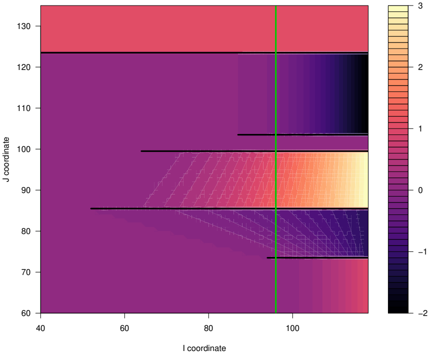


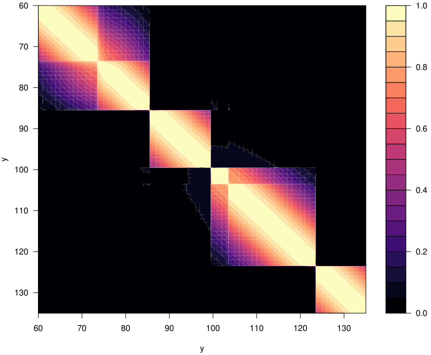
Covariance Matrix for Vertical Slices through the Location Parameter Space



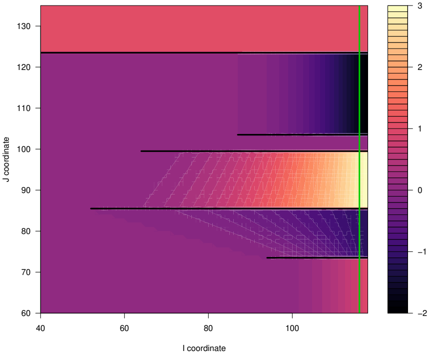


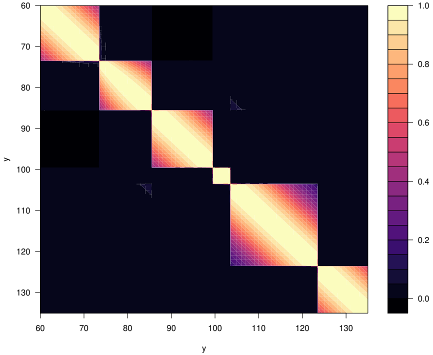
Covariance Matrix for Vertical Slices through the Location Parameter Space





Covariance Matrix for Vertical Slices through the Location Parameter Space

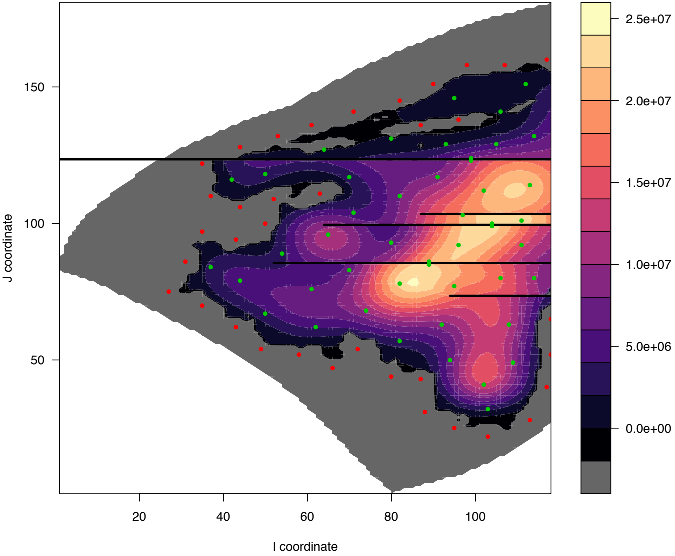




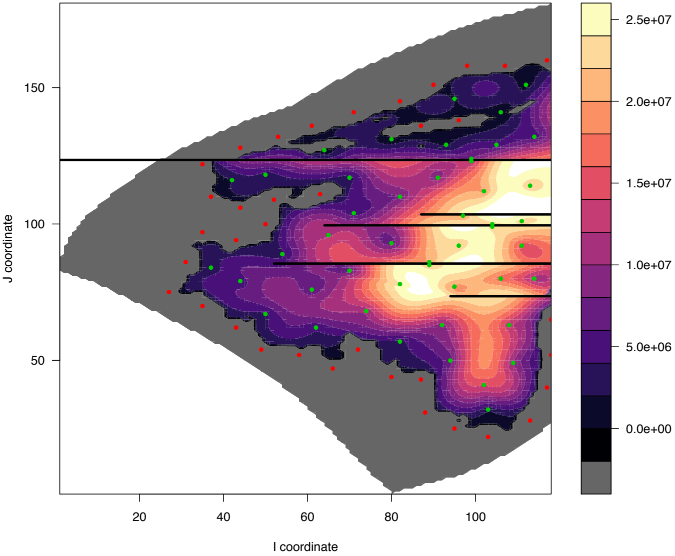
Design for Well 1 Wave 1

- Oil reservoir engineering insight suggests the first well should be a **producer**.
- An initial 41 point design is constructed by sequentially selecting points which **minimises the average emulator variance** over the input parameter space.
- This is based on an emulator for functions with structured discontinuities.
- An additional 6 points are added in 3 pairs to **explore the existence and magnitude of the (partial) discontinuities**.
- **Ghost runs with zero NPV** are manually added in the water region surrounding the field to pin down the oil-water boundary or the location parameter space.

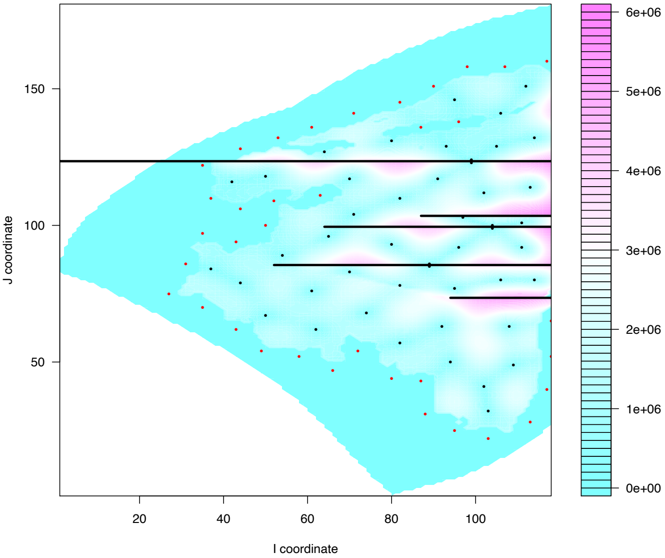
NPV Emulator Expectation for Well 1 Wave 1



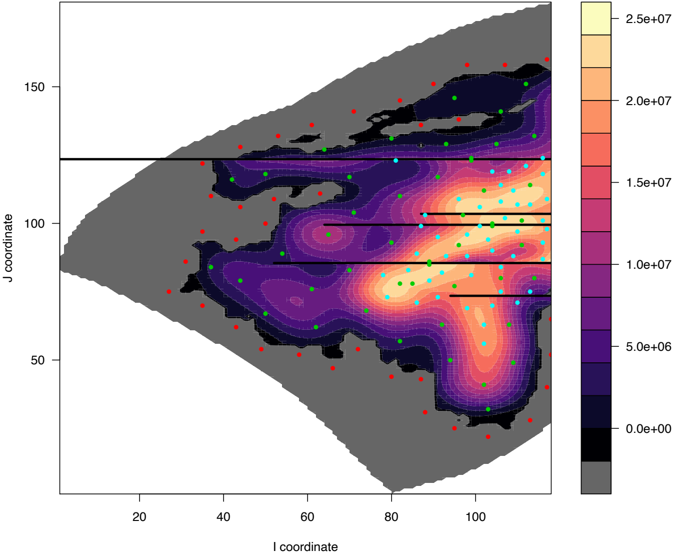
NPV Emulator Expectation plus 2 SD for Well 1 Wave 1



NPV Emulator Standard Deviation for Well 1 Wave 1



NPV Emulator Expectation for Well 1 Wave 2



Conclusion

- The proposed emulation methodology is able to **handle general forms of structured partial discontinuities** whilst still maintaining a flexible choice of emulator form.
- These emulators have been successfully demonstrated to help choose well locations for the **TNO OLYMPUS Challenge**.
- Emulators which can handle structured partial discontinuities can be incorporated within our **sequential decision support strategy** for the full well placement optimisation problem.
- An efficient strategy is to **simultaneously choose locations for groups of wells**. For example, 3 wells represents a 6D problem requiring an embedding of the location parameter space in 9D.
- The embedded emulator methodology is **fully generalisable and transferable** to many scientific and industrial applications possessing such structured discontinuities.

References I

Dunlop, M. M., Girolami, M. A., Stewart, A. M., and Teckentrup, A. L. (2018).
How deep are deep gaussian processes?
Journal of Machine Learning Research, 19(54):1–46.

EAGE, editor (2019).
EAGE/TNO Workshop on OLYMPUS Field Development Optimization. European Association of Geoscientists and Engineers (EAGE) and the Netherlands Organisation for Applied Scientific Research (TNO, EAGE Publications.

ISAPP (2018).
Integrated systems approach for petroleum production (isapp).
ISAPP research program website.

Kennedy, M. C. and O'Hagan, A. (2001).
Bayesian calibration of computer models.
Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63(3):425–464.

Kimpton, L., Challenor, P., and Daniel B, W. (2020).
Classification of computer models with labelled outputs.
In Submission.

Mohammadi, H., Challenor, P., Goodfellow, M., and Williamson, D. (2020).
Emulating computer models with step-discontinuous outputs using gaussian processes.

O'Hagan, A. (2006).
Bayesian analysis of computer code outputs: A tutorial.
Reliability Engineering and System Safety, 91:1290–1300.

References II

Pope, C. A., Gosling, J. P., Barber, S., Johnson, J. S., Yamaguchi, T., Feingold, G., and Blackwell, P. G. (2021).
Gaussian process modeling of heterogeneity and discontinuities using voronoi tessellations.
Technometrics, 63(1):53–63.

Santner, T. J., Williams, B. J., and Notz, W. I. (2003).
The Design and Analysis of Computer Experiments.
Springer Series in Statistics. Springer-Verlag, 1 edition.

Tanaka, S., Wang, Z., Dehghani, K., He, J., Velusamy, B., and Wen, X.-H. (2018).
Large scale field development optimization using high performance parallel simulation and cloud computing technology.
In *2018 SPE Annual Technical Conference and Exhibition*. Chevron Energy Technology Company, Society of Petroleum Engineers.

TNO (2017).
Olympus oil reservoir model input decks.
OLYMPUS Oil Reservoir Model Input Decks and documentation designed for use in the TNO OLYMPUS field development optimisation challenge.

Vernon, I., Goldstein, M., and Bower, R. G. (2010).
Galaxy formation: a bayesian uncertainty analysis.
Bayesian Analysis, 5(4):619–670.