Horizon Europe (HORIZON)
Marie Skłodowska-Curie
Actions Doctoral Networks
(MSCA-DN)

UNIVERSITY OF LEEDS

MISSION-CCS: Material Science Innovation for Accelerated, Sustainable and Safe Implementation of Carbon Capture and Storage

Introduction to Python Programming, Flow Modelling and Optimisation

Harvey Thompson

Summary

UNIVERSITY OF LEEDS

- Aims of the training course
- Timescales
- What will you do?
- How do you get support?
- Summary

Aims

Scientific roles increasingly reliant on confident use of programming and data science techniques

- ❑ Develop skills in Python programming (2nd most popular language) if you don’t already have them
- ❑ Awareness of scientific modelling using differential equations
- ❑ Awareness of data science and optimisation methods

BUT …. at your own pace

Timescales

- To get the most out of the course you will have to read the materials, work through the practical programming, modelling and optimization examples and do the assignment.

- You have been given worked solutions so that you can work independently

- *Aim to complete the course by the end of year 2 so you can put what you have learned into practice.*

BUT it is ultimately up to you to take as long as you want…

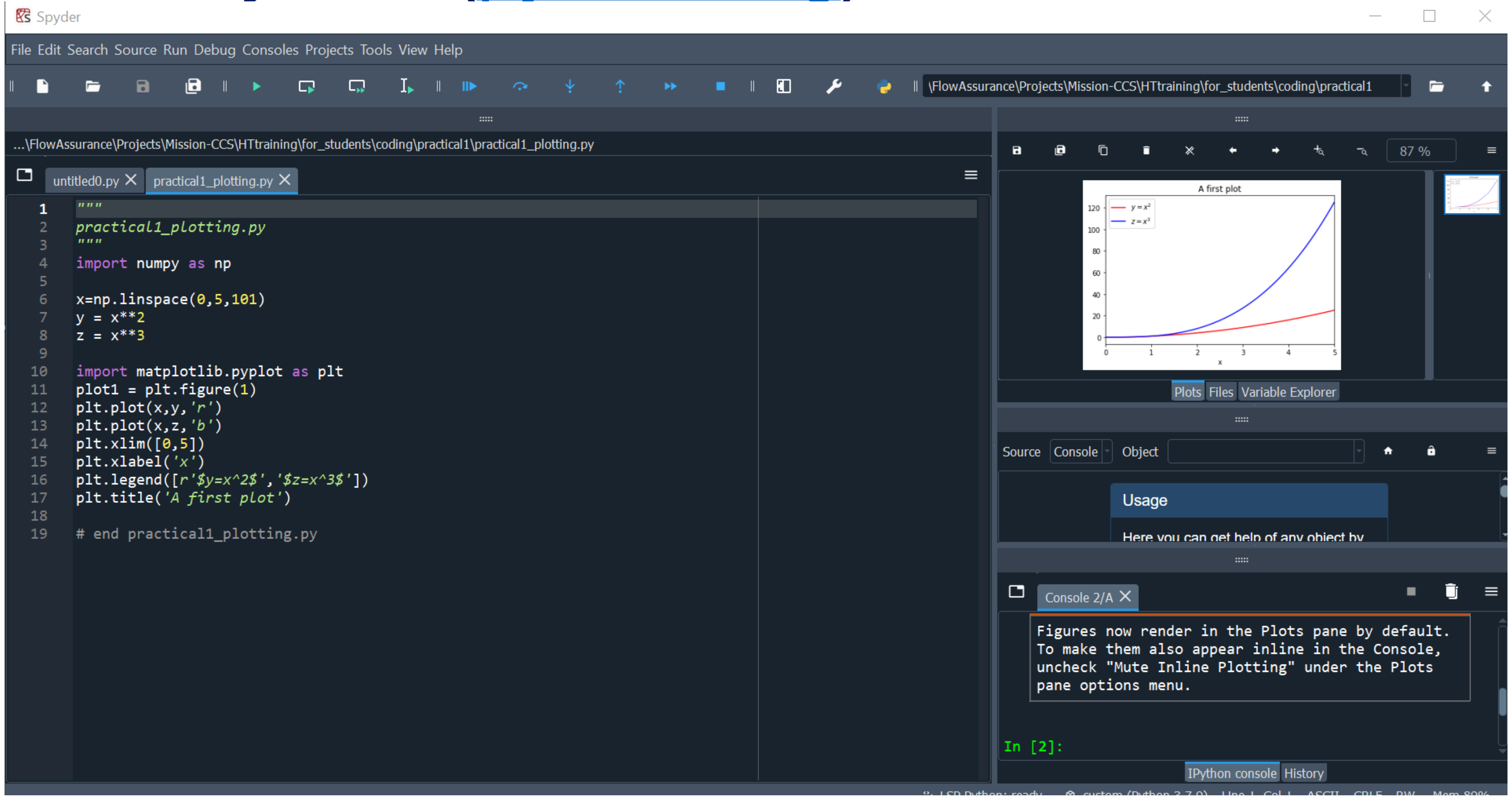
What will you do?

Setup Python and GUI Editor on your computer

You will need to download Python (python.org) and a GUI editor.

Lots of GUI editors available: Spyder, IDLE etc..

I tend to use **Spyder** (https://www.spyder-ide.org)



What will you do?

I will give you access to 3 directories: ***coding***, ***numericalmodelling*** and ***optimisation***

The **coding** directory: provides a simple introduction to Python programming that you can work through at your own pace.

Brief Introduction to Python.pdf : introduces you to Python and some simple programming statements and simple programs. The Python syntax is given in the accompanying file *BriefIntroductiontoPython.py*

The **coding** directory

Practical 1: programming exercises showing you how to:

- Manipulate vectors and matrices
- Calculate eigenvalues of matrices
- Control program flow
- Plot simple functions
- Use Python functions
- Solve a simple programming project based on Carbon Dating

Python programs and full solutions are provided.

The **numericalmodelling** directory:
Engineering systems are often modelled analysed theoretically using differential equations.

Practical 2: will show you how to solve linear and nonlinear systems of differential equations in Python. Worked solutions and Python programs are provided.

A comprehensive introduction to numerical modelling using Python is also given.

UNIVERSITY OF LEEDS

The **numericalmodelling** directory:
The introduction to numerical modelling using Python introduces methods by solving simple problems including carbon dating and spring-mass systems then applies them to:

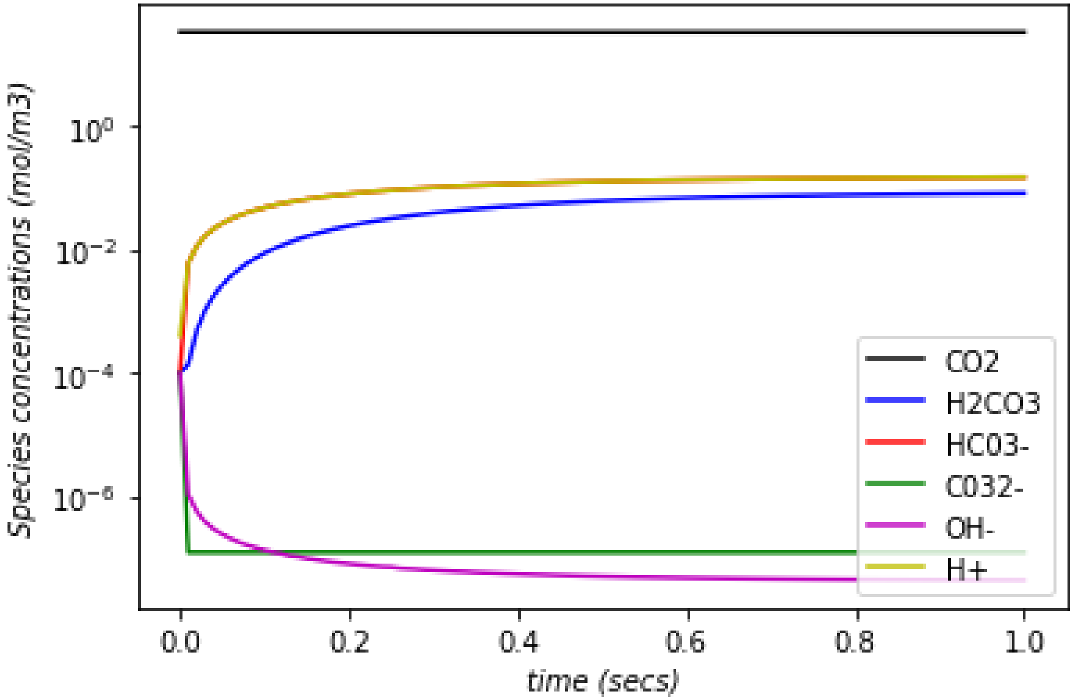
… then solves two more complex problems:

- Modelling chemical reactions in CO_2 pipeline corrosion
- Flow-induced vibration in oil and gas pipelines

Modelling chemical reactions in CO_2 pipeline corrosion

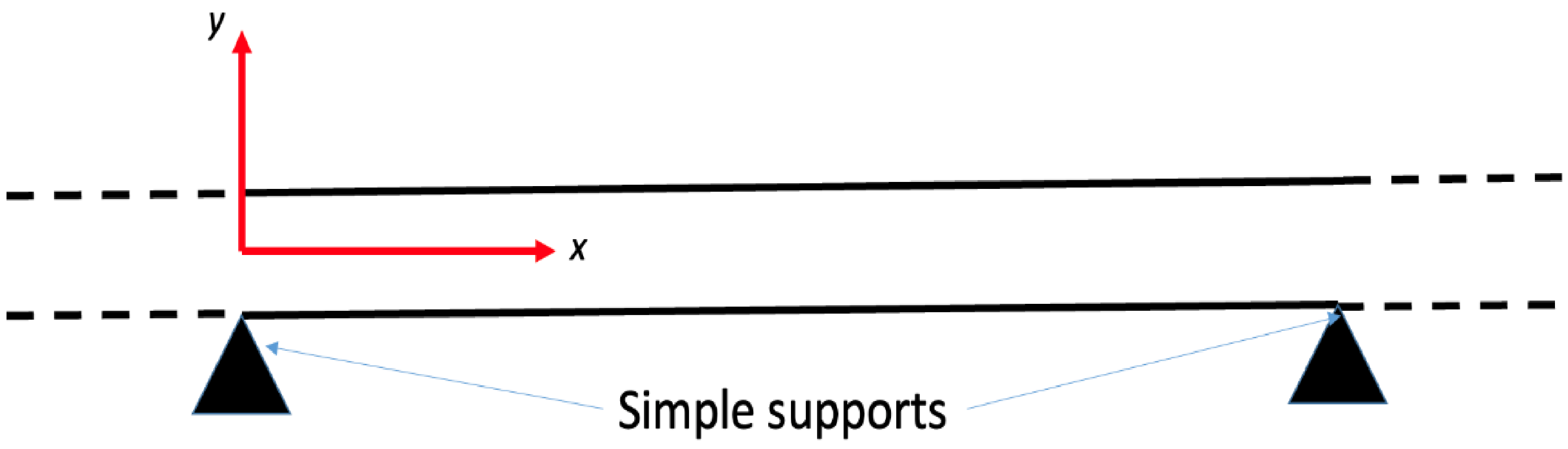
Shows how chemical reactions can be modelled and solved numerically in Python – *will require you to spend a bit more time understanding the equations and how they are solved!*

Working Python programs are provided



Flow-induced vibration in oil and gas pipelines

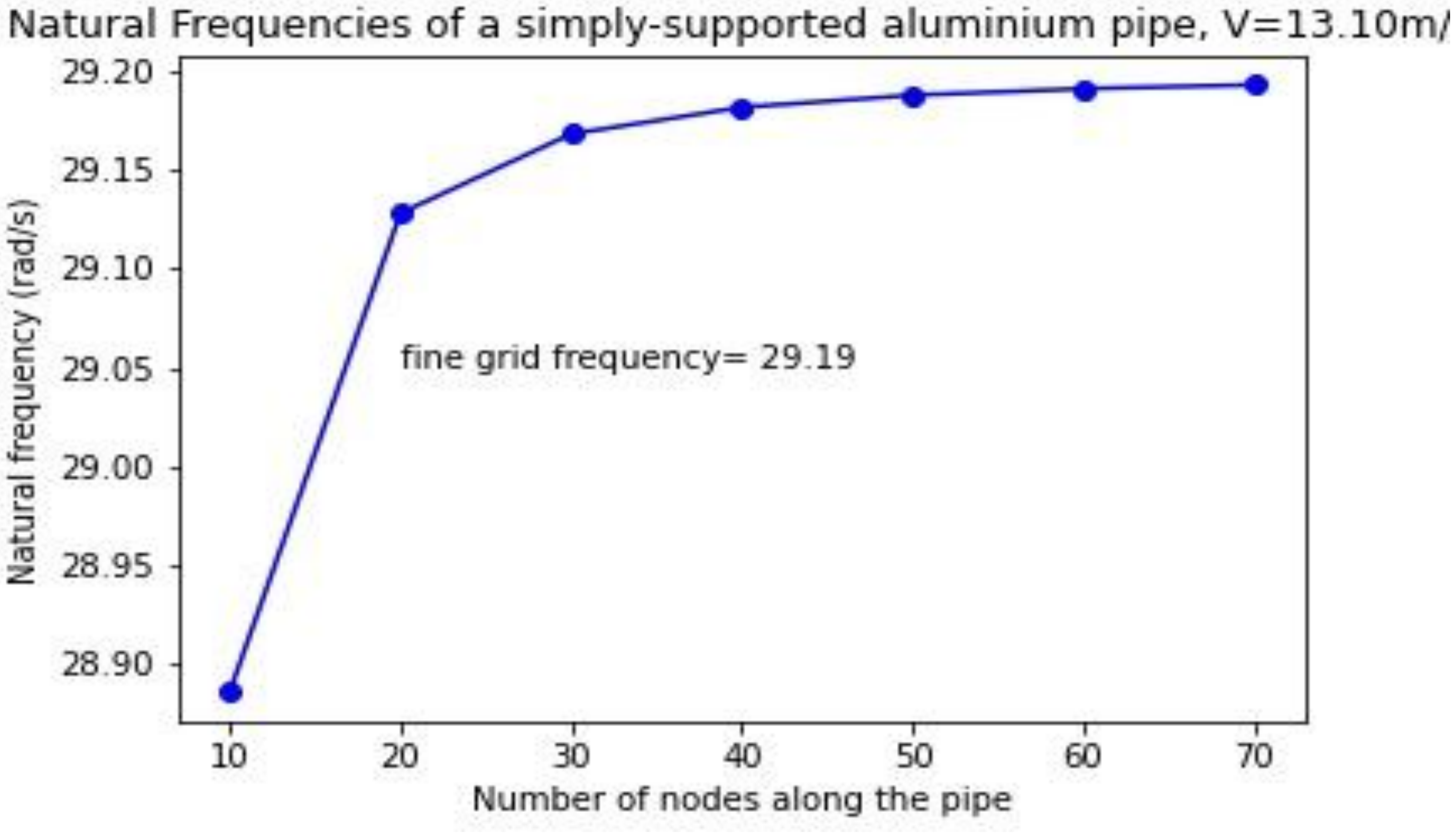
Develops a general modelling framework for analysing pipeline stability and failure – by analysing simply-supported pipelines.



$$y(0,t) = \frac{\partial y^2}{\partial x^2}(0,t) = 0 \qquad y(L,t) = \frac{\partial y^2}{\partial x^2}(L,t) = 0$$

Flow-induced vibration in oil and gas pipelines

Explains the importance of *verification*



UNIVERSITY OF LEEDS

Flow-induced vibration in oil and gas pipelines

And *validation* against previous studies in the literature

Flow velocity m/s	Experiment (Dodds & Runyan (1965) rad/s	Finite Element (Dangal & Ghimire (2019) rad/s	Finite Difference model rad/s
13.10	26.10	29.00	29.19
23.485	24.11	24.73	25.25
29.722	19.93	20.47	21.11

Table 1: Validation of effect of flow velocity on natural frequency for a simply supported-simply supported pipe.

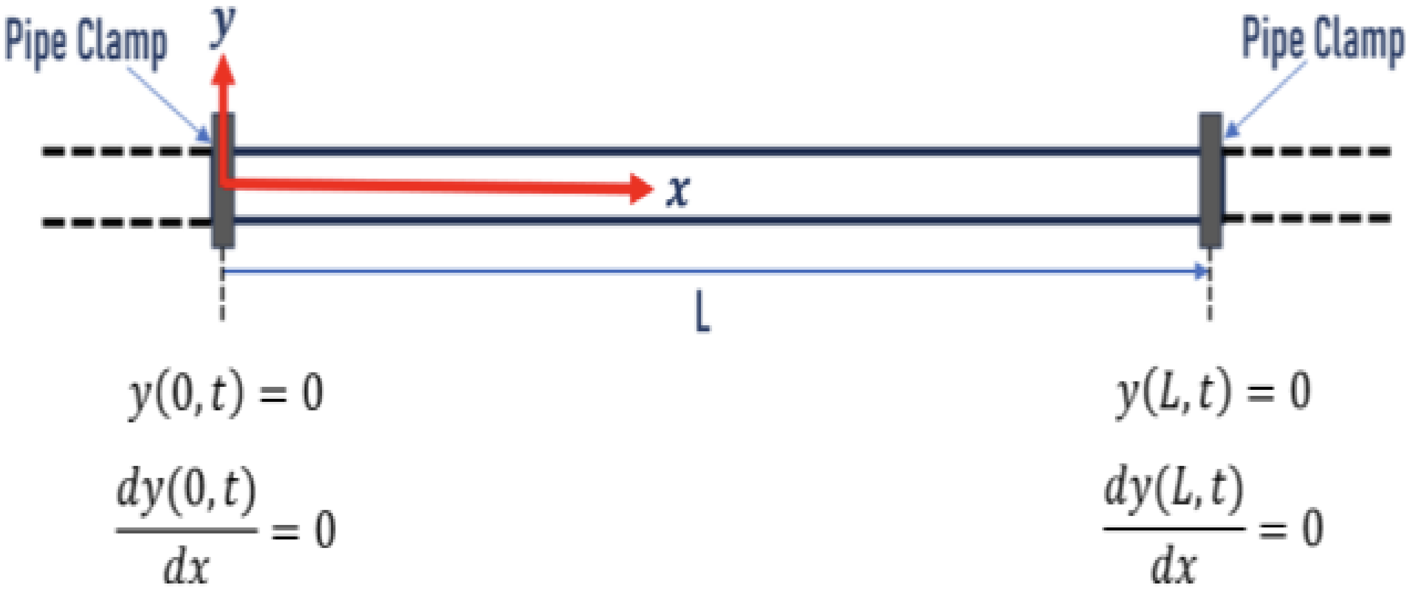
Working Python codes are provided to aid understanding.

UNIVERSITY OF LEEDS

Flow-induced vibration in oil and gas pipelines

Your assignment will be to modify the mathematical equations and Python programs to solve the more practically-important clamped pipeline case.

Worked solutions will be provided after you have attempted these!



UNIVERSITY OF LEEDS

<u>The **optimisation** directory:</u>

Introduces the related areas of *Data Science*, *Design Optimisation* and *Machine Learning* in a [comprehensive set of notes] including:

Introduction to Design Optimisation – e.g. bridges

:

UNIVERSITY OF LEEDS

Graphical solution methods

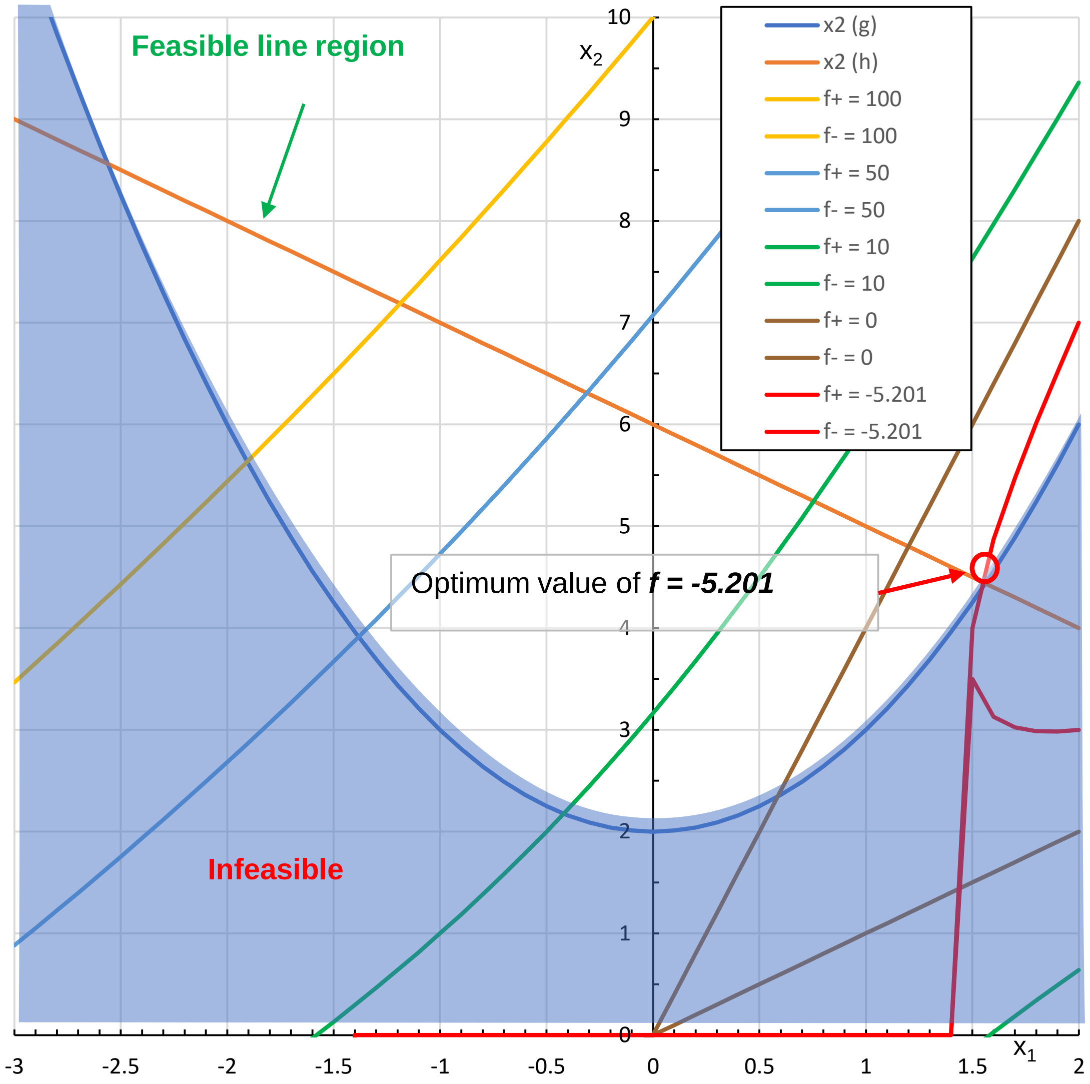
Minimize: $f\left(x_1,x_2\right)=4x_1^2-5x_1x_2+x_2^2$

Subject to:

$$g\left(x_1,x_2\right)=x_1^2-x_2+2\le 0$$

$$h\left(x_1,x_2\right)=x_1+x_2-6=0$$

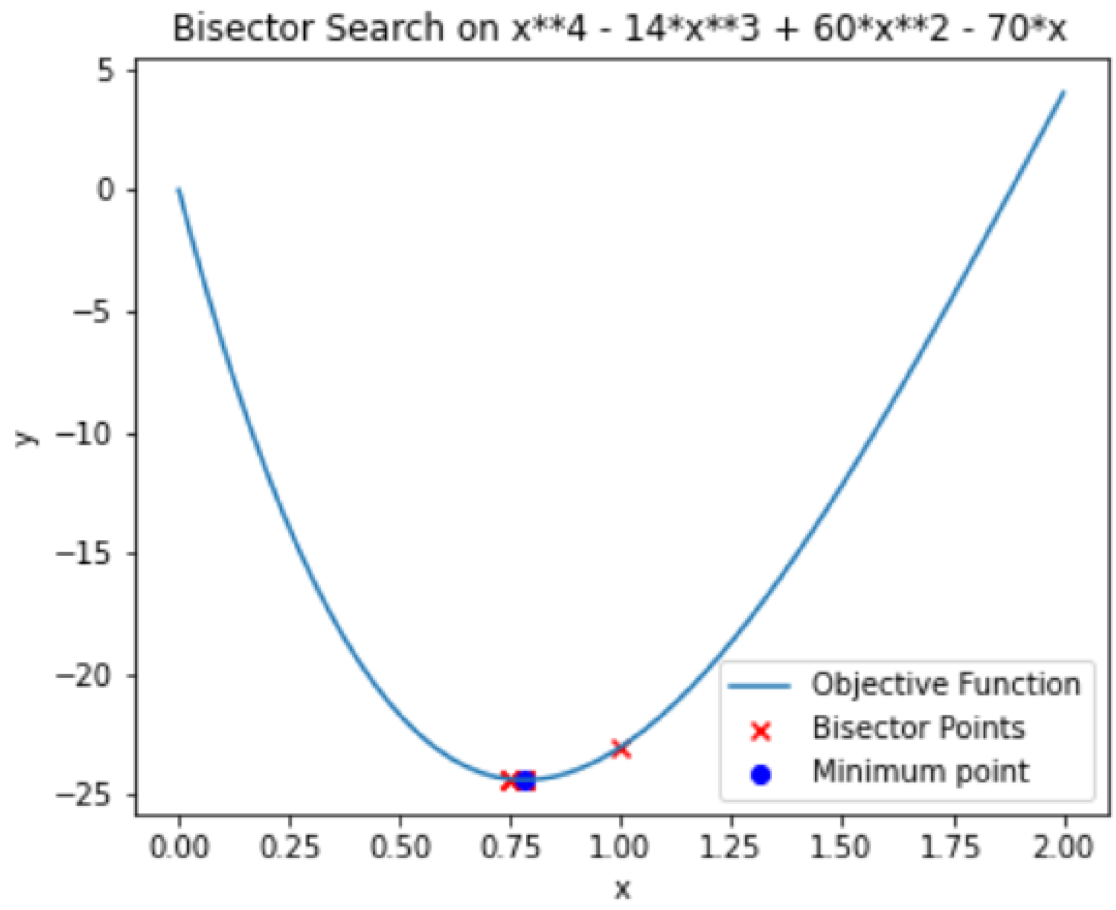
There are a set of examples, worked solutions and Python programs you can use to get familiar with the graphical solution method.



Uncontrained optimisation e.g. very simple search methods – *Bisector method*

Halves search interval each iteration

There are many more efficient search methods!

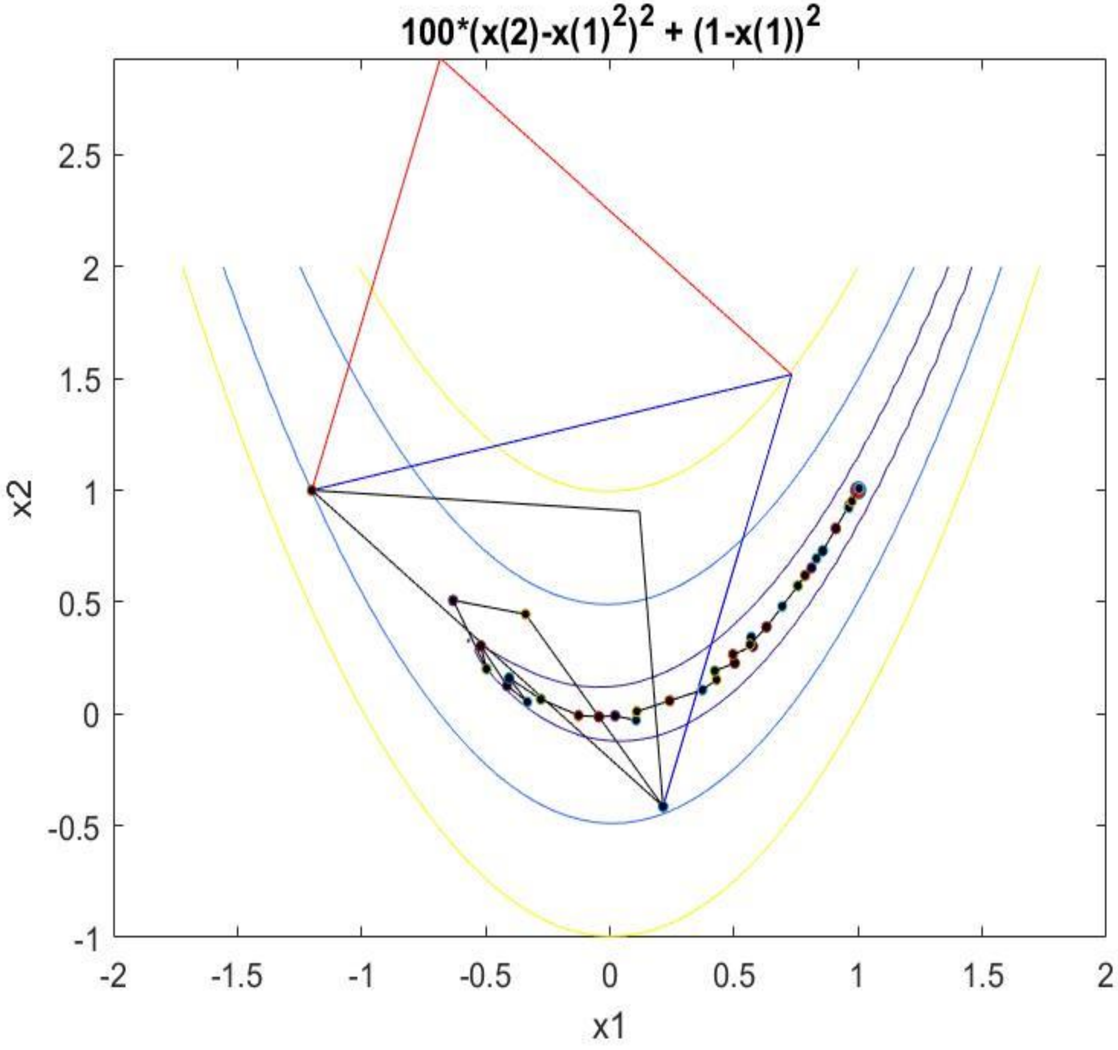


Uncontrained optimisation

E.g. Nelder-Mead simplex

Navigates through design space by comparing values at vertices

Works well for up to ~10 variables



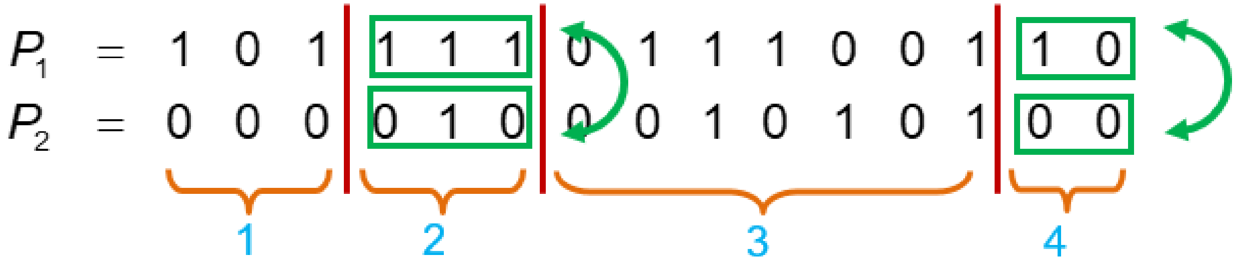
:
https://neldermead-jscfbwbryzlpf6pdjhyltj.streamlit.app/

The **optimisation** directory

Uncontrained optimisation: Random search methods needed to avoid local minima.

Stochastic methods based on biological or physical processes.

e.g. Genetic Algorithms



To produce these two children

$$O_1 = 1\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 0$$

$$O_2 = 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0$$

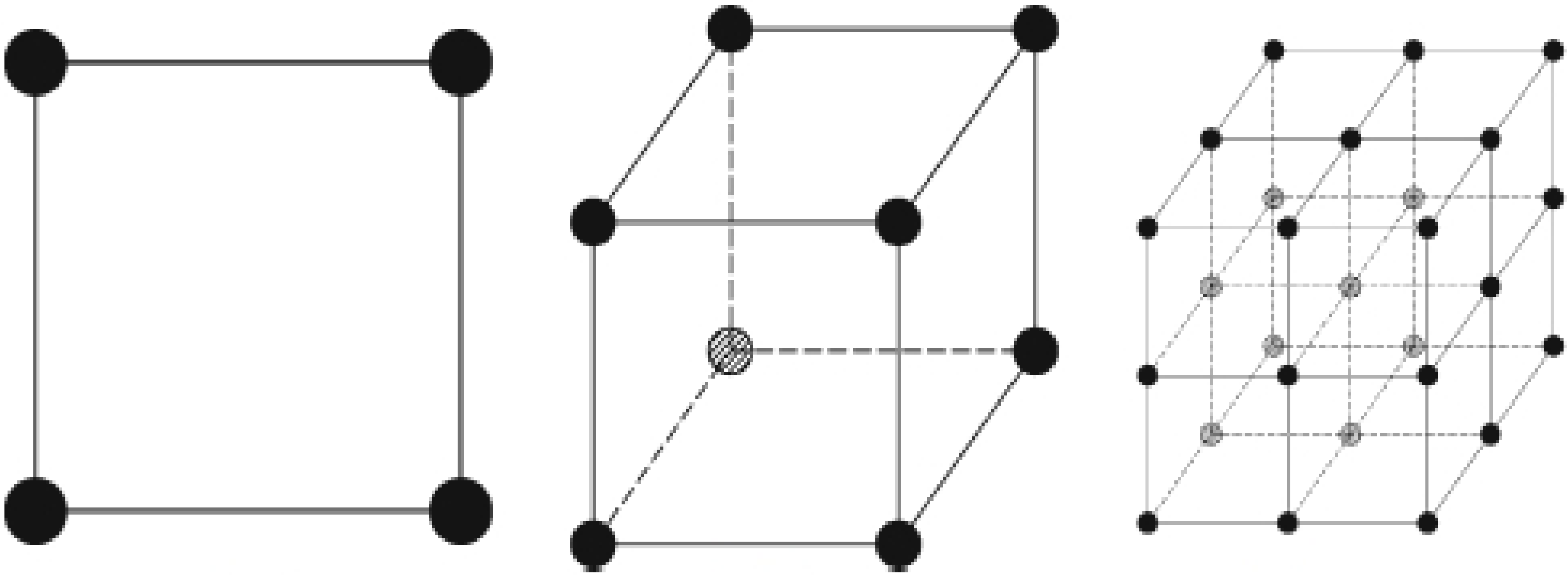
The **optimisation** directory

Design of Experiments: Often expensive to get data so need to get maximum useful information from minimum sampling points.

- Use Design of Experiments (DoE) methods to minimise sampling.

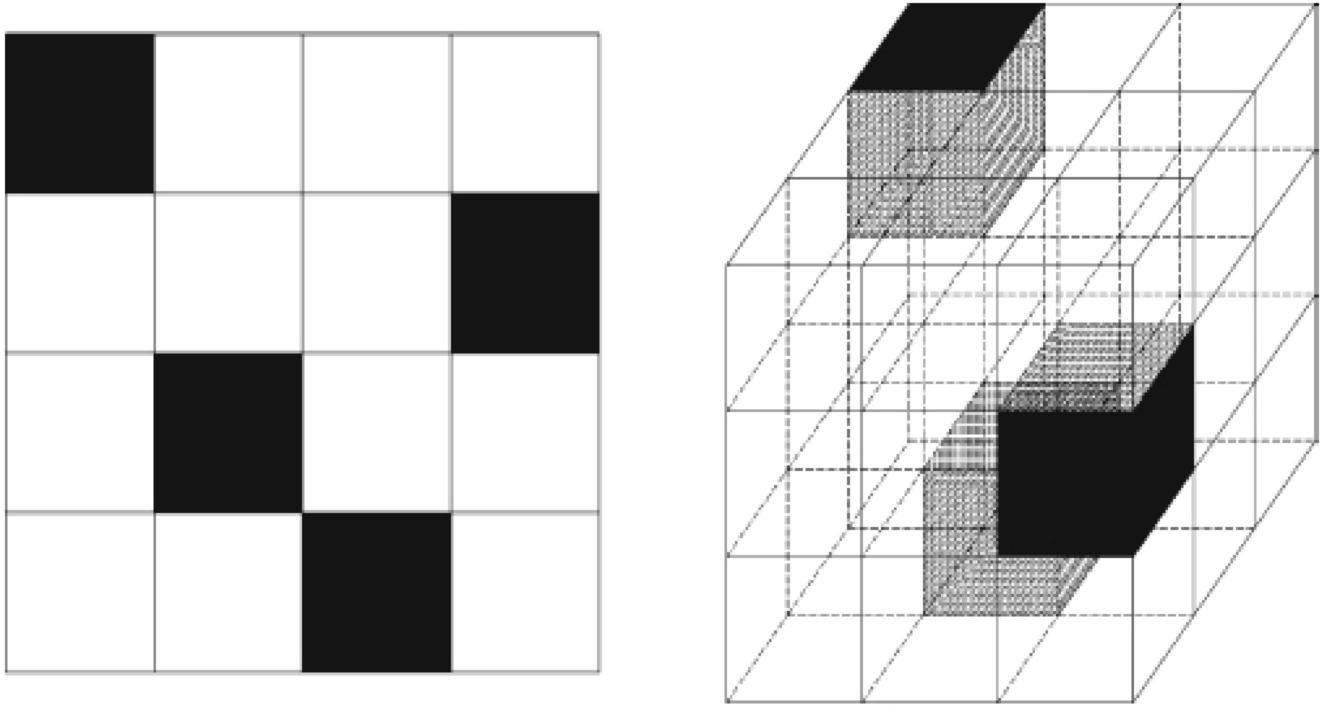
- Provides a summary of the most popular classical (*full factorial, fractional factorial, central composite, box-behnken*,…) and non-classical (*Latin Hypercube, Hammersley*,…) DoE methods

Full factorial DoEs - the following figure shows graphical representations of the 2^2, 2^3 and 3^3 full factorial designs (Cavazutti, 2013)



(a) 2^2 (b) 2^3 (c) 3^3

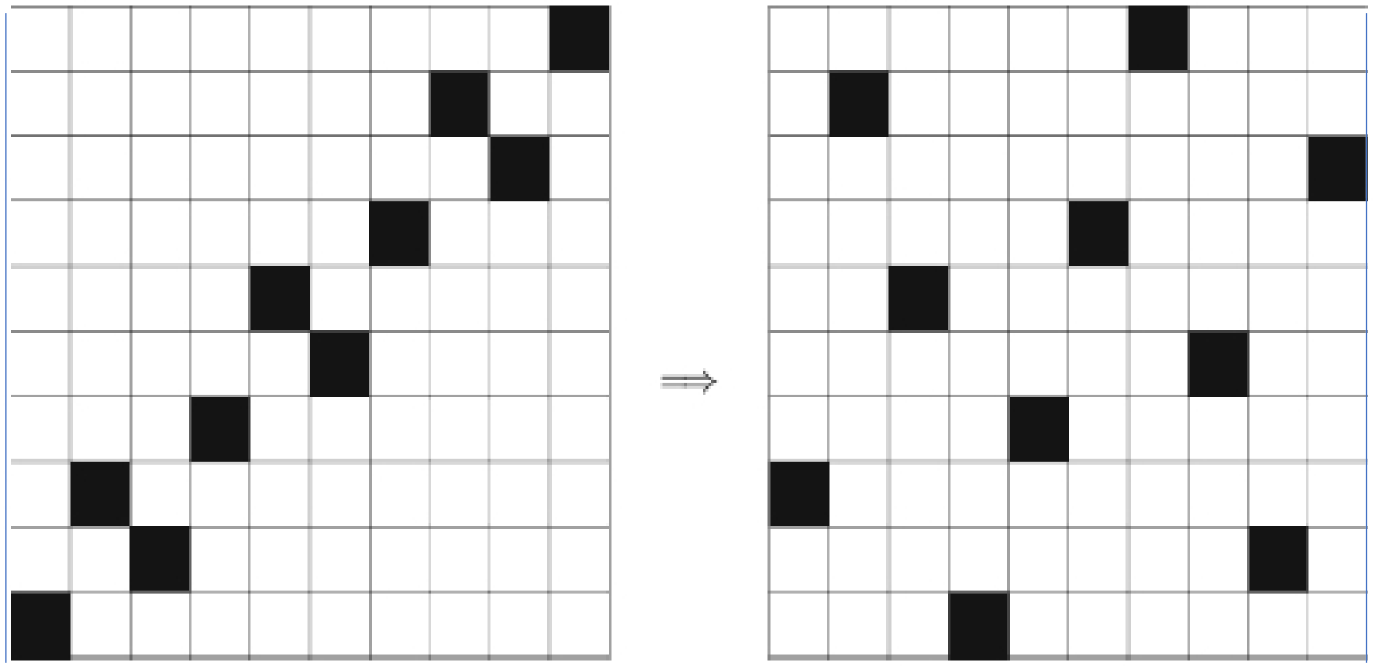
Latin Hypercube DoEs - Each element is populated by **one and only one** sample point, chosen randomly within each element



(a) $k = 2$, $N = 4$ (b) $k = 3$, $N = 3$

The **optimisation** directory

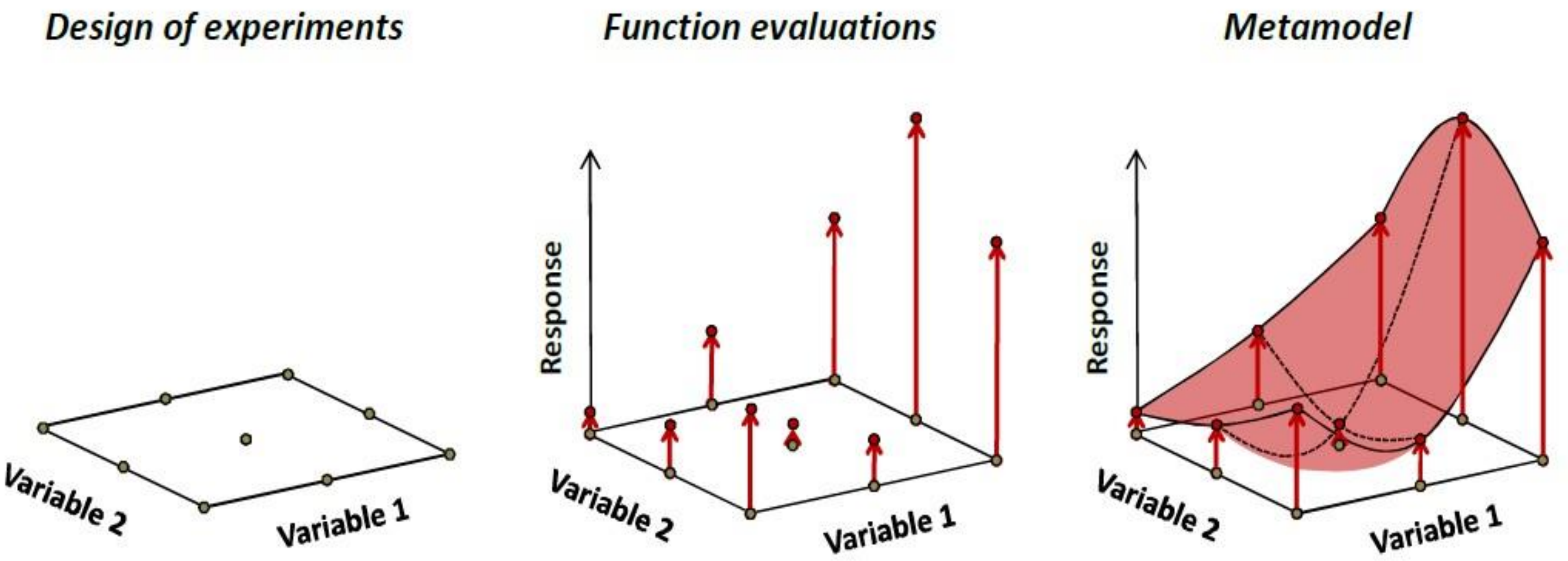
Latin Hypercube DoEs - sub-volumes must be chosen to spread the samples all over the design space.
E.g. N=10 and k=2 design variables.



Copyright © 2024 University of Leeds UK. All rights reserved

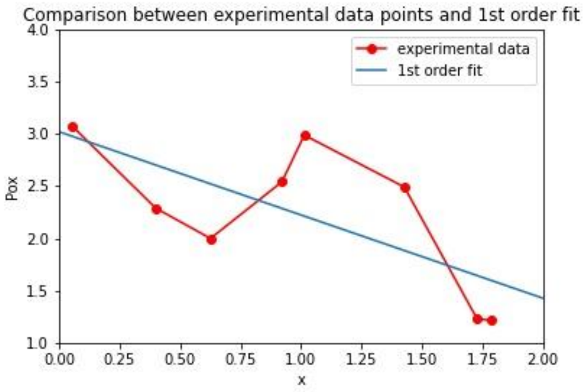
UNIVERSITY OF LEEDS

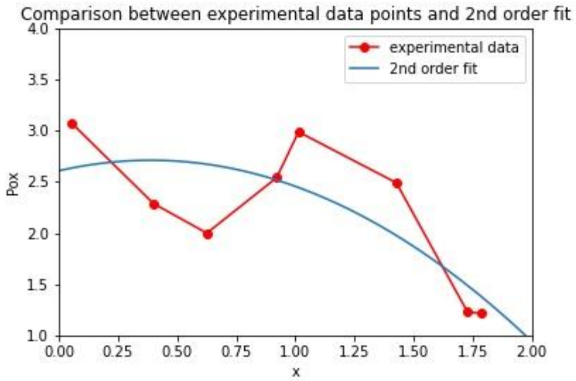
Surrogate Modelling: the surrogate modelling process for two design variables is illustrated in the following figure

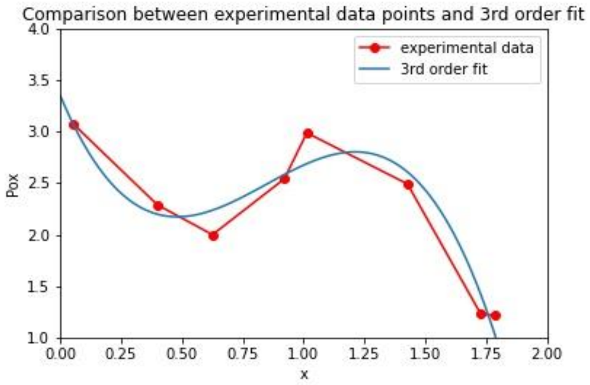


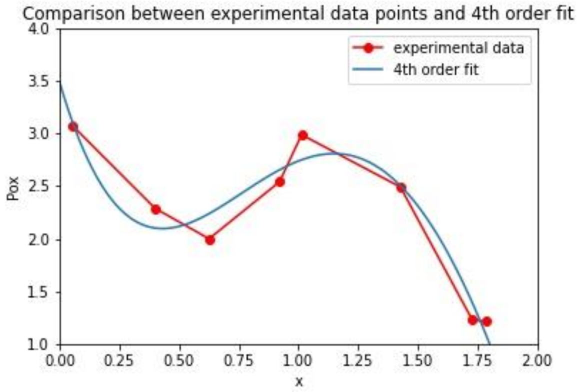
Copyright © 2024 University of Leeds UK. All rights reserved

Surrogate Modelling: e.g. polynomial (least squares) approximations of phosphorus concentration in photo-bioreactors. https://curvefitting-l8dvsd6bbqeeuh4uua2zxo.streamlit.app/

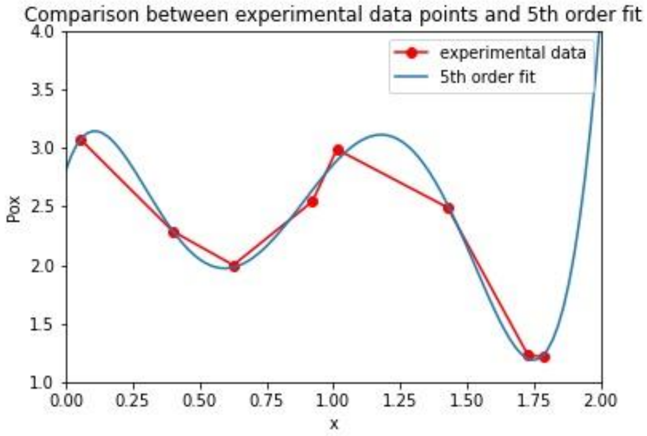


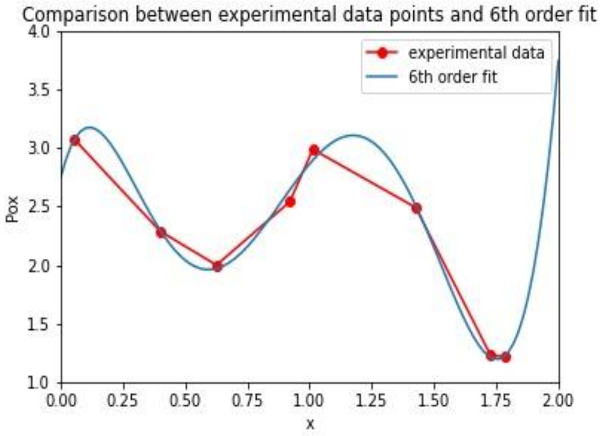






Surrogate Modelling: e.g. polynomial (least squares) approximations of phosphorus concentration in photo-bioreactors.





Need cross-validation methods from machine learning to determine ‘best’ polynomial fit and avoid ***over-fitting***.

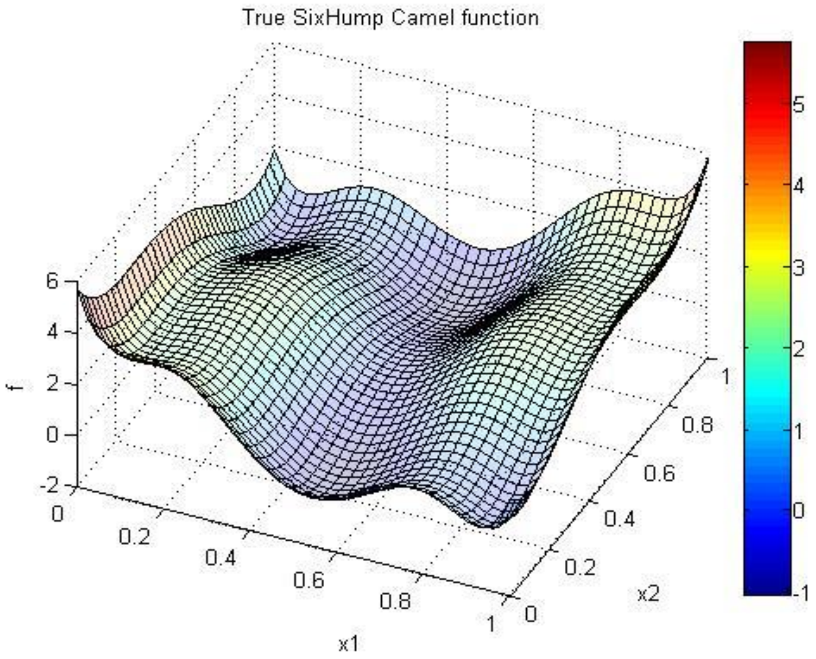
Copyright © 2024 University of Leeds UK. All rights reserved

UNIVERSITY OF LEEDS

Surrogate Modelling: Six Hump Camel Back (SHCB) Function with the global minima $f = -1.0316$.

$$X_1 = 4x_1 - 2;\ X_2 = 2x_2 - 1;\ 0 \leq x_1, x_2 \leq 1$$

$$f(x_1, x_2) = (4 - 2.1X_1^{\ 2} + X_1^{\ 4}/3)\ X_1^{\ 2} + X_1X_2 + (-4 + 4X_2^{\ 2})\ X_2^{\ 2}$$

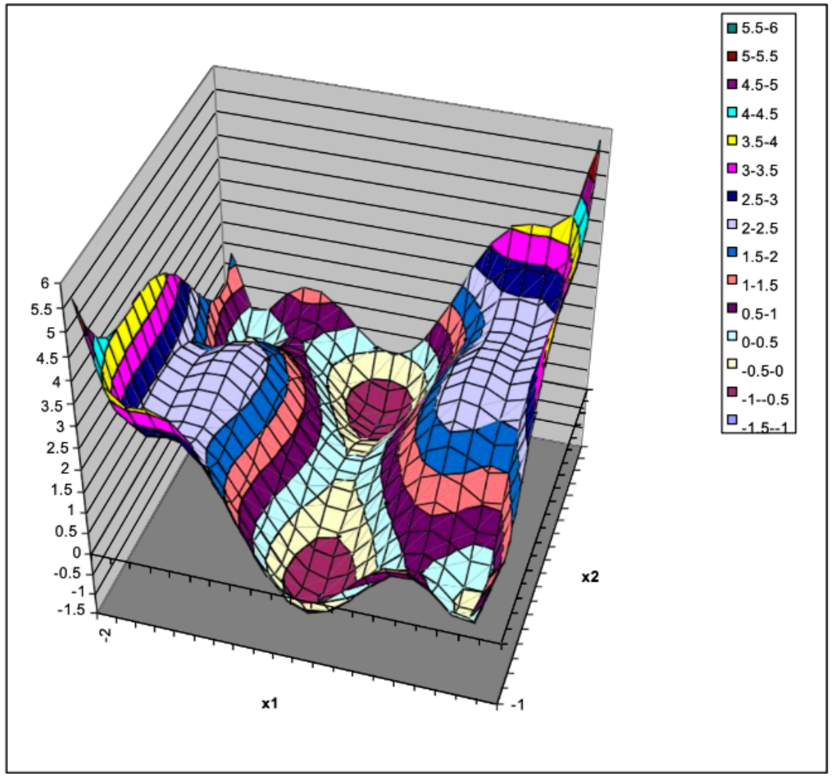


Copyright © 2024 University of Leeds UK. All rights reserved

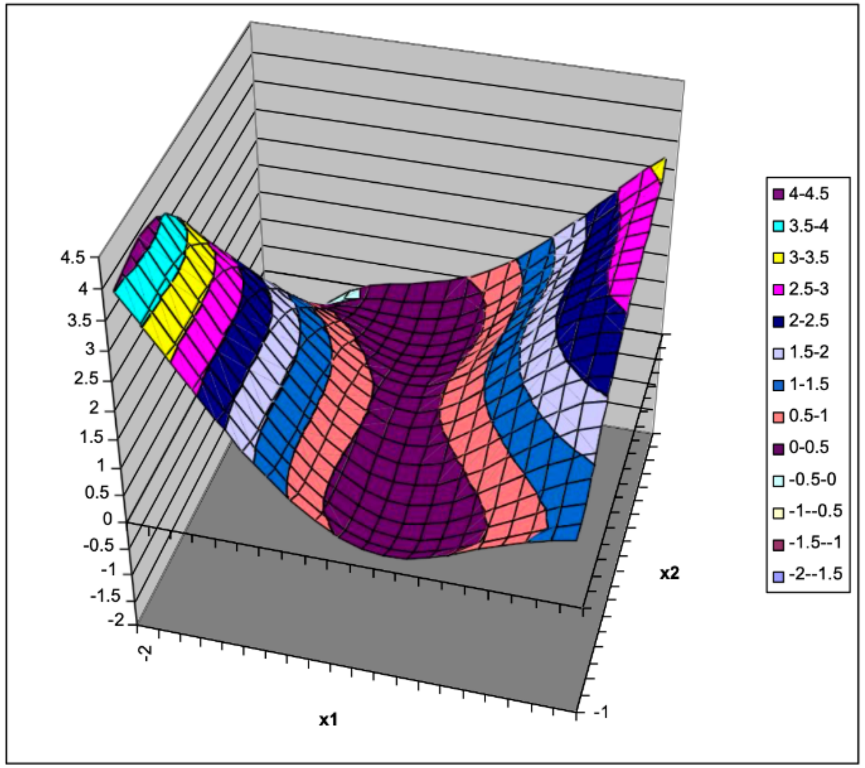
UNIVERSITY OF LEEDS

Surrogate Modelling: (SHCB) Function - Moving Least Squares

Analytical Function



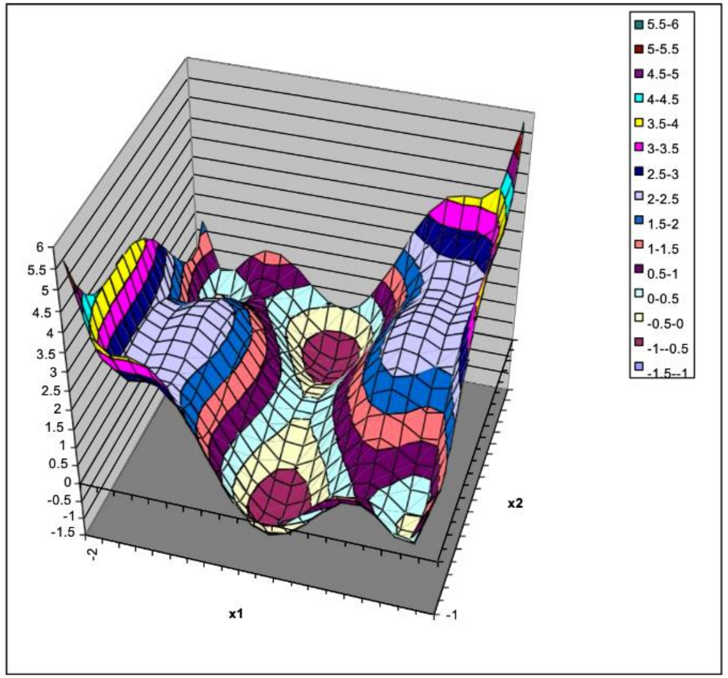
20 DoE points, β=10.0



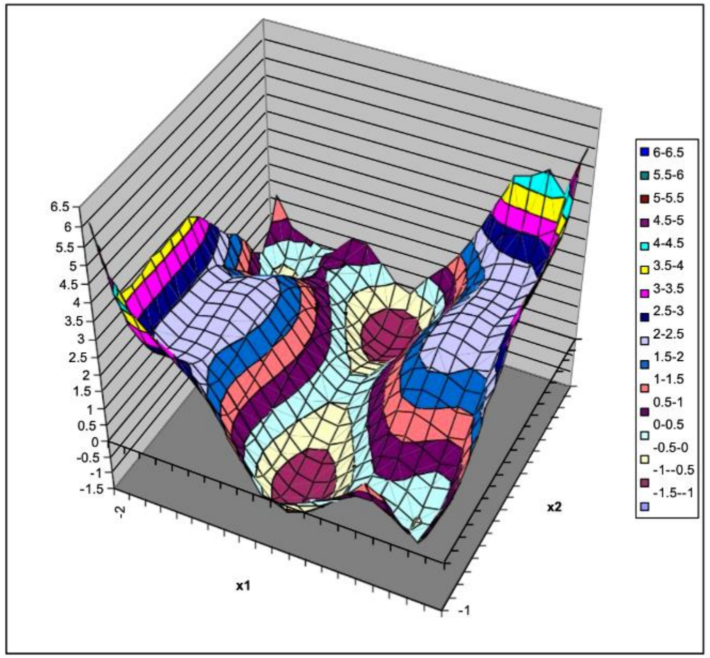
Copyright © 2024 University of Leeds UK. All rights reserved

Surrogate Modelling: (SHCB) Function - Moving Least Squares

Analytical Function



100 DoE points, β=120.0



Copyright © 2024 University of Leeds UK. All rights reserved

Surrogate Modelling: (SHCB) Function – Neural Networks

The first layer is called the input layer and the last one is the output layer. The layers between these two are the hidden layers.

The total number of layers is called the network's depth.

Deep Neural Networks have many layers, enabling very complex behaviour to be represented accurately.

The first and last layers can be considered to be the inputs and outputs of the surrogate model.

Each node in the hidden layers represents a function.

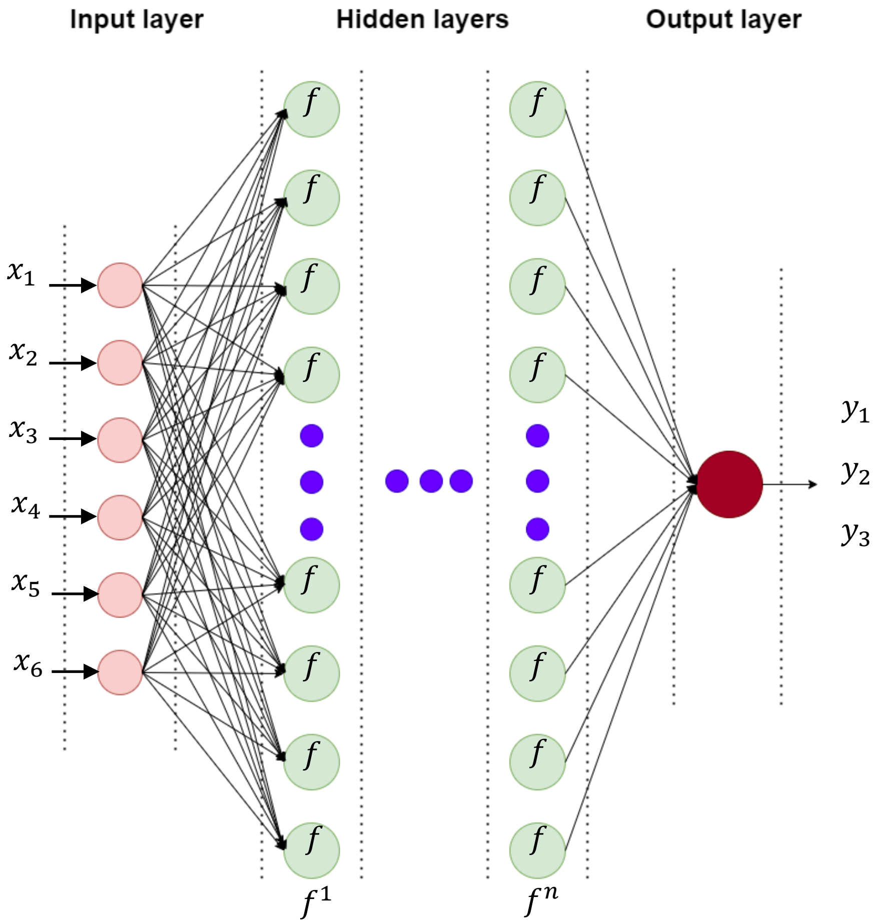
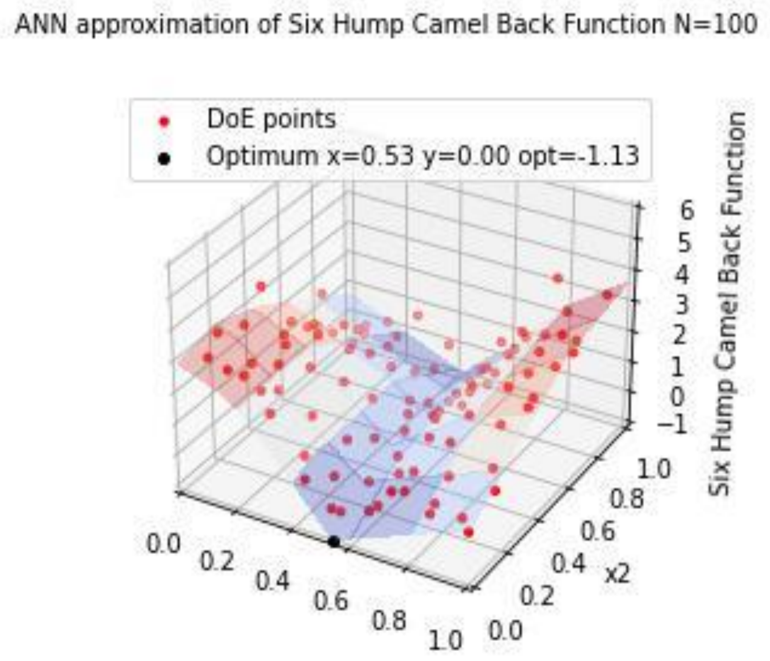
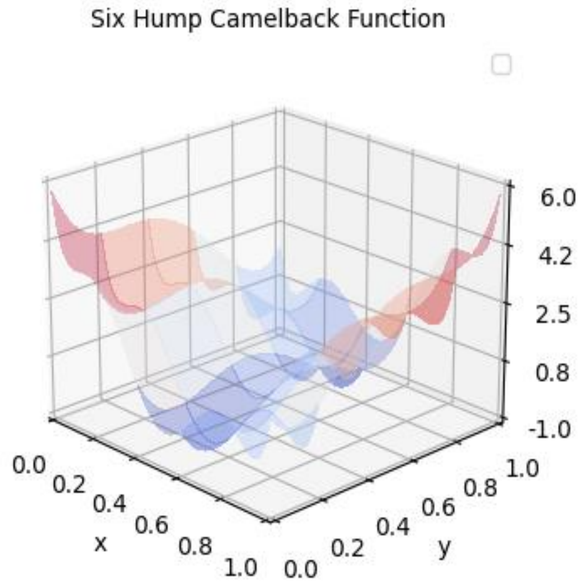


Figure due to M. Raihan

Surrogate Modelling: (SHCB) – Neural Networks



Exact Function



Neural Networks work much better for much larger datasets!

Copyright © 2024 University of Leeds UK. All rights reserved

UNIVERSITY OF LEEDS

Surrogate Modelling: SHCB function

Other important surrogate modelling/Machine Learning methods are introduced:

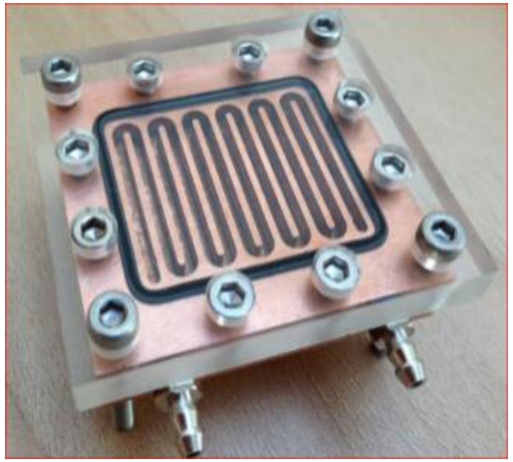
- Radial Basis Functions
- Random Forests
- Gaussian Processes

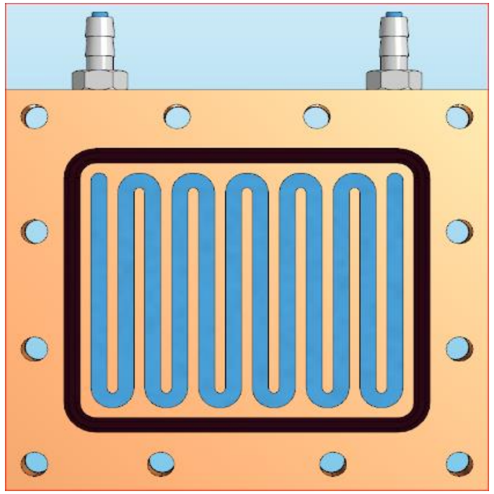
Copyright © 2024 University of Leeds UK. All rights reserved

Multi-objective Optimisation: Liquid cooled heat sinks for high-density electronics cooling => minimise both thermal resistance and pressure drop – multi-objective optimisation!

:

The following figures shows the physical design (left), CAD model (right)

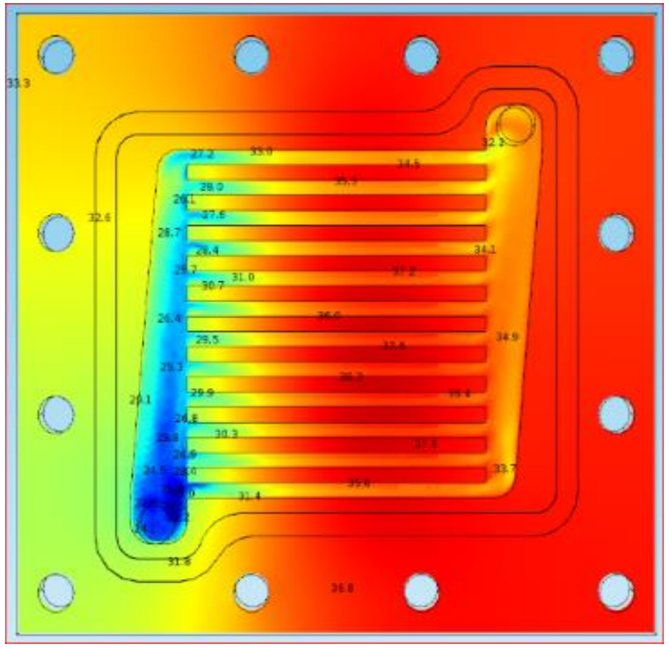


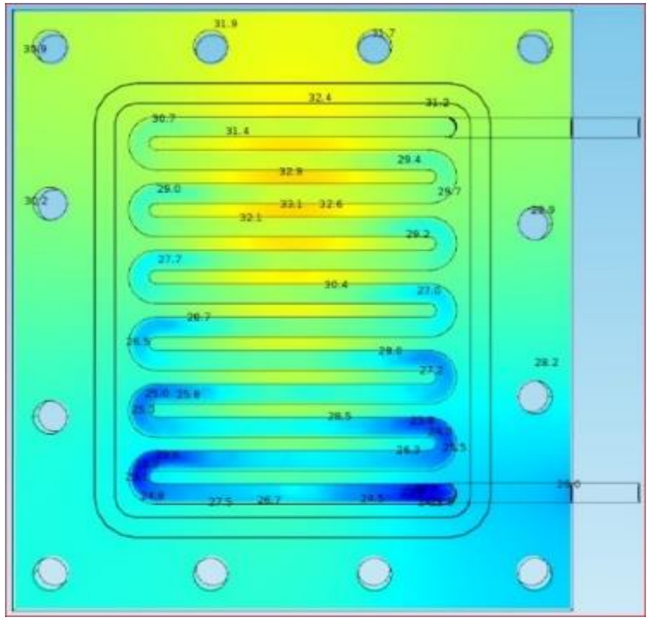


Copyright © 2024 University of Leeds UK. All rights reserved

UNIVERSITY OF LEEDS

Multi-objective Optimisation: Liquid cooled heat sinks for high-density electronics cooling – temperature distributions.





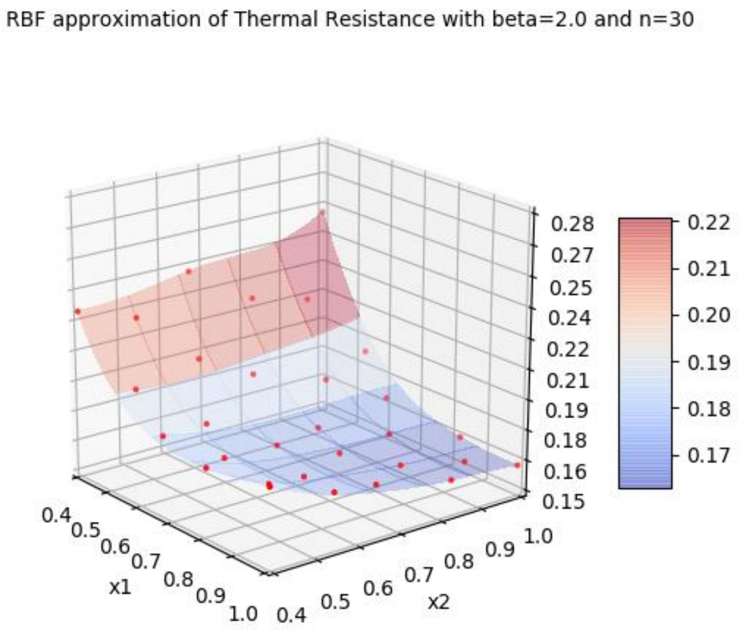
Serpentine channels reduce temperature but with much bigger pressure drop!

Copyright © 2024 University of Leeds UK. All rights reserved

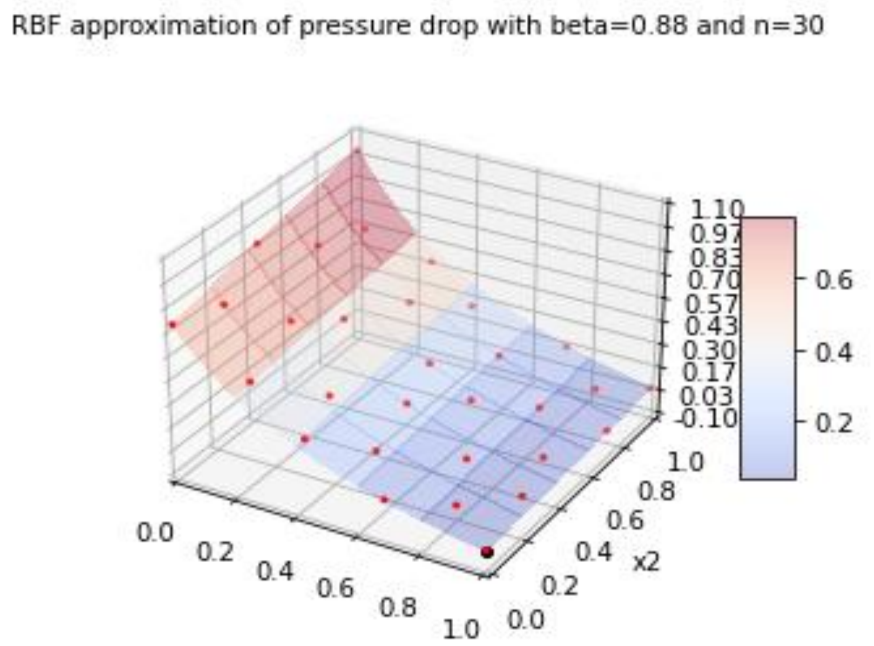
UNIVERSITY OF LEEDS

Multi-objective Optimisation: Surrogate modelling using Radial Basis Functions with 30 DoE points

thermal resistance



pressure drop



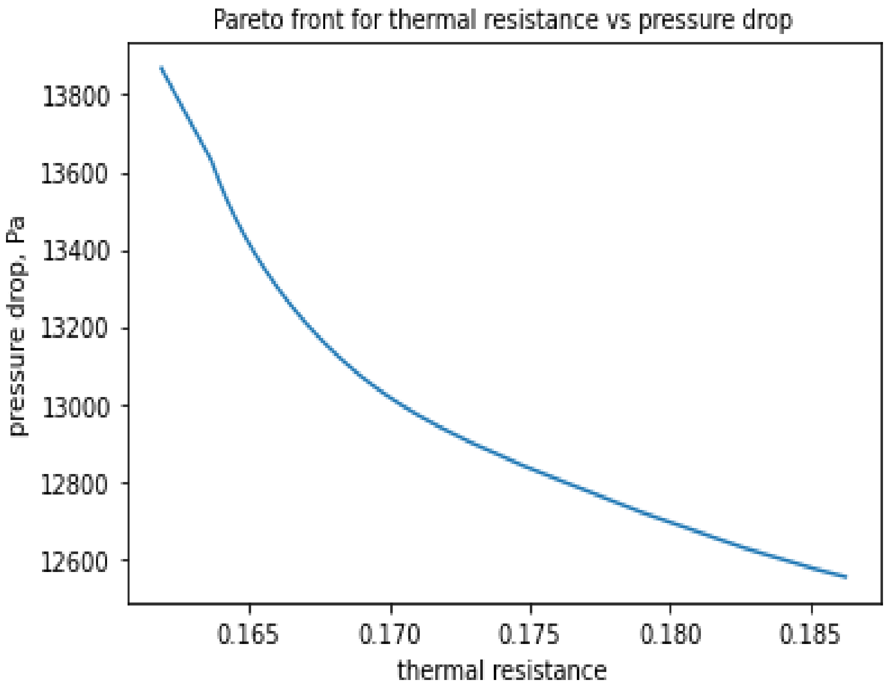
Copyright © 2024 University of Leeds UK. All rights reserved

UNIVERSITY OF LEEDS

Multi-objective Optimisation: Pareto Front

Shows the compromises that can be struck between minimising each of the objectives: reducing thermal resistance below 0.17 will require pressure drops > 13,000 Pa.

:



Copyright © 2024 University of Leeds UK. All rights reserved

How do you get further support?

Materials have been designed to enable you to study independently and at your own pace.

However, you may need further help so feel free to contact Harvey Thompson at H.M.Thompson@leeds.ac.uk any time.

Copyright © 2024 University of Leeds UK. All rights reserved

Summary

We have prepared a set of notes, worked examples and Python programs that you can work through at your own pace.

The learning objectives are that you will:

- Develop skills in python programming
- Understand and be able to solve mathematical models with differential equations in python
- Understand how to use data science, machine learning and optimisation methods for scientific analysis

Copyright © 2024 University of Leeds UK. All rights reserved

Summary

There is no set timescale for completing the material but advisable to finish in your first 2 years, particularly if material highly relevant for your PhD project.

:

Good luck and I hope you find the materials interesting!

Copyright © 2024 University of Leeds UK. All rights reserved

Horizon Europe (HORIZON)
Marie Skłodowska-Curie
Actions Doctoral Networks
(MSCA-DN)

UNIVERSITY OF LEEDS

Thanks for listening!

Any questions?

H.M.Thompson@leeds.ac.uk