Oral presentation | Higher order methods

Higher order methods-IV

Tue. Jul 16, 2024 4:30 PM - 6:30 PM Room C

[6-C-01] Online Bayesian Optimization of Polynomial-Multigrid Cycles for Flux Reconstruction

*Freddie Witherden[1], Will Trojak[2], Sambit Mishra[1] (1. Texas A&M University, 2. IBM Research UK)

Keywords: flux reconstruction, multigrid, Bayesian optimization

Online Bayesian Optimisation of Polynomial-Multigrid Cycles for Flux Reconstruction

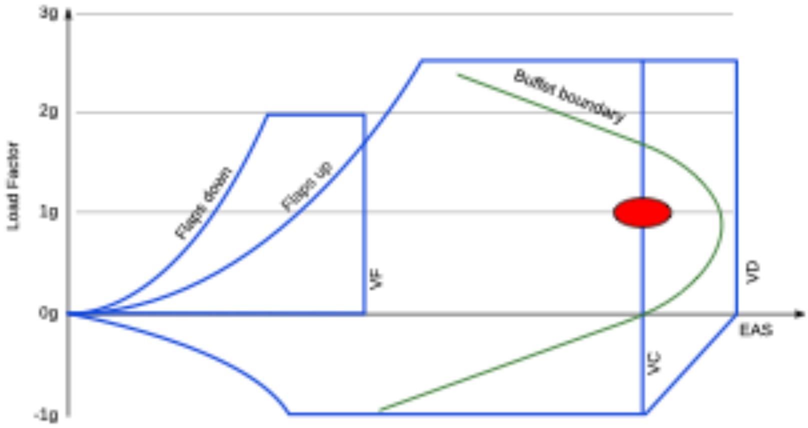
F.D. Witherden, W. Trojak, S. Mishra

Department of Ocean Engineering
Texas A&M University

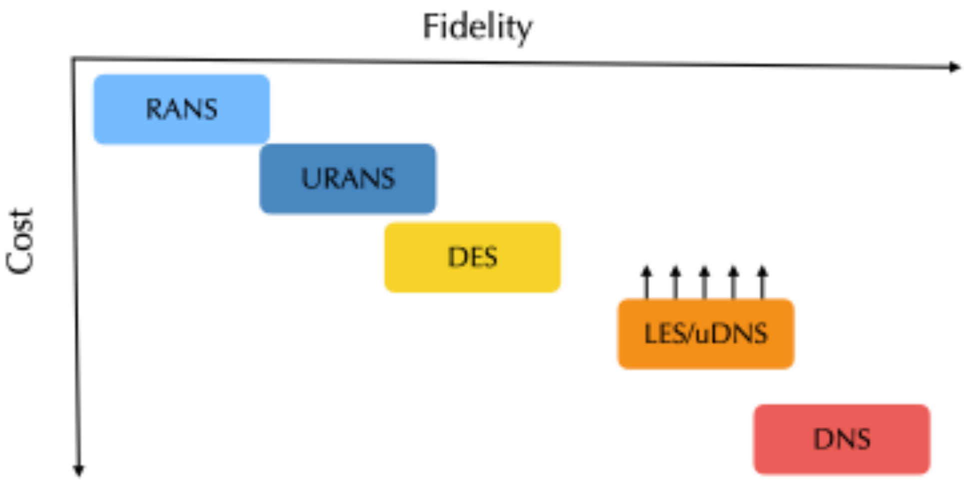
Motivation

- Computational fluid dynamics (CFD) is the bedrock of several high-tech industries.

Motivation



Motivation

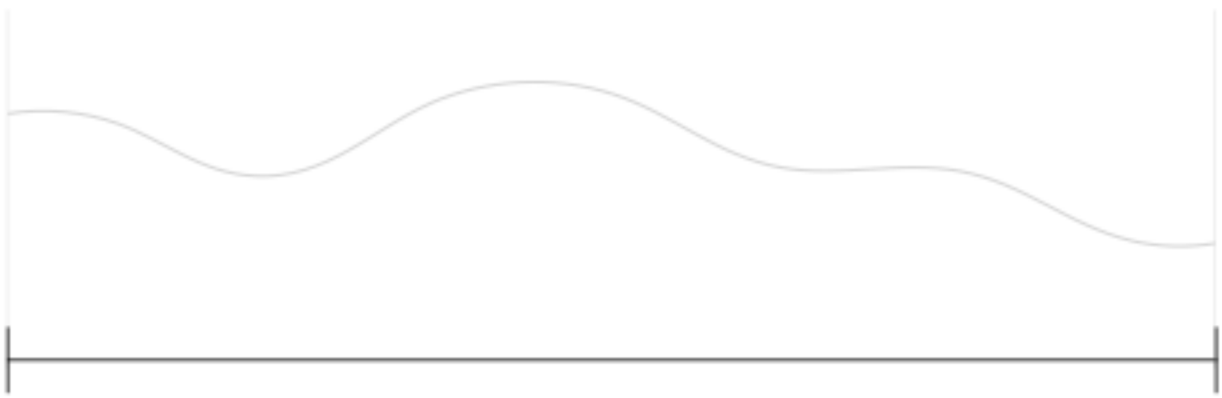


High-Order Methods

- Our choice of method is the high-order accurate **Flux Reconstruction** (FR) approach of Huynh.
- Combines aspects of traditional finite volume (FVM) and finite element methods (FEM).

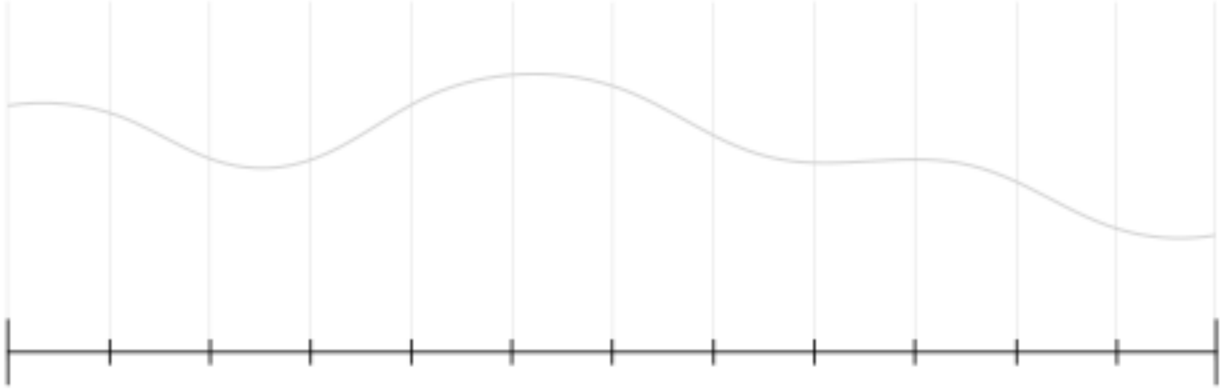
High-Order Methods

- Consider a smooth function



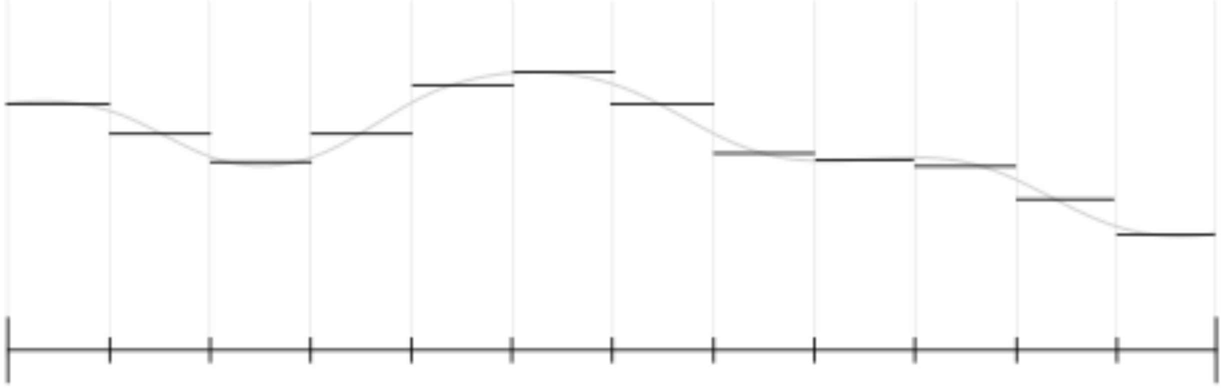
High-Order Methods

- In FVM we divide the domain into **cells**…



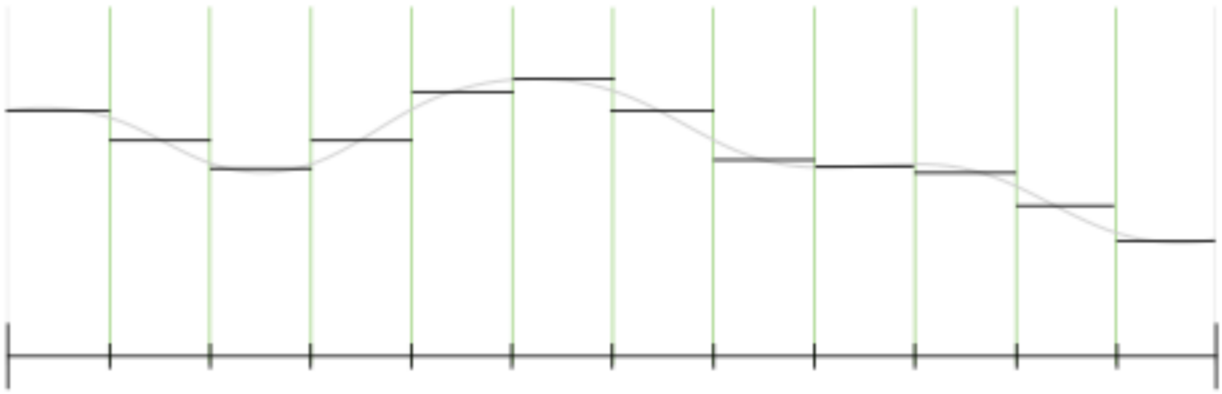
High-Order Methods

- …and in each cell store the **average** of the function.



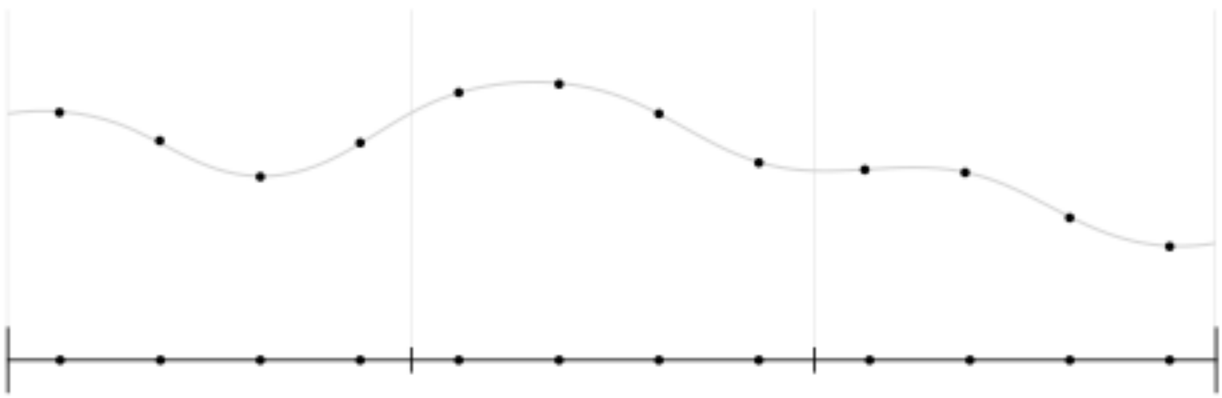
High-Order Methods

- Cells are coupled via Riemann solves at the interfaces.



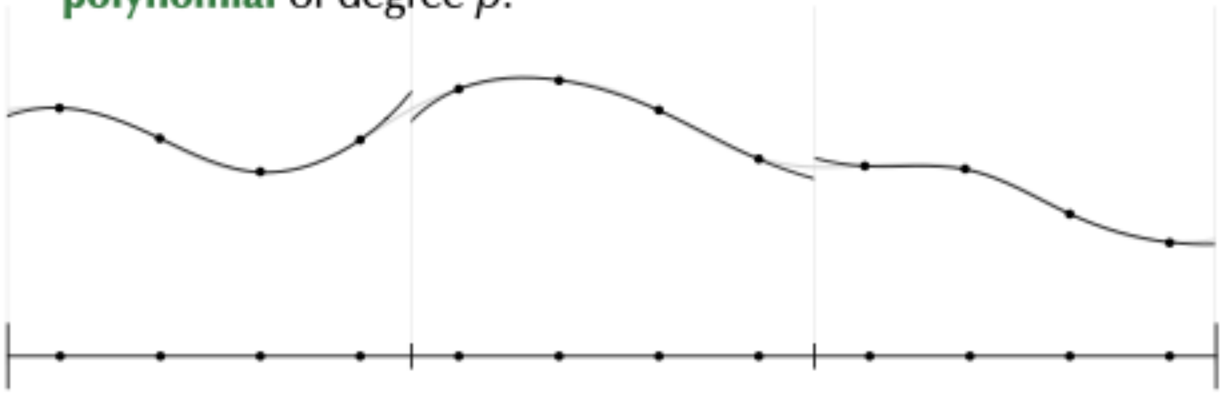
High-Order Methods

- In FR we divide the domain into **elements**…



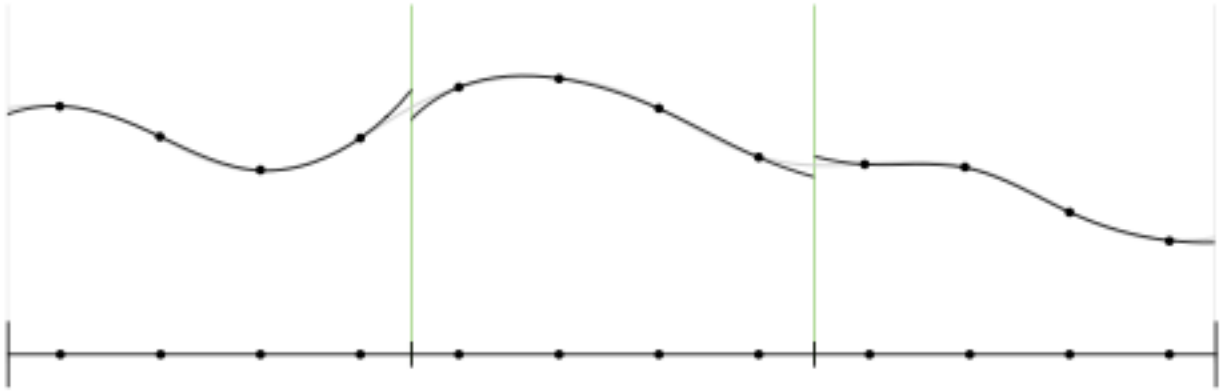
High-Order Methods

- …and in each element store a **discontinuous interpolating polynomial** of degree *p*.



High-Order Methods

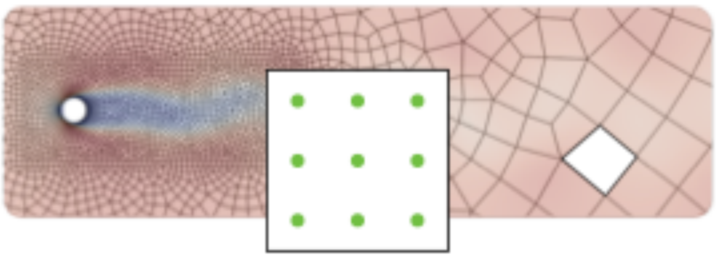
- As before elements are coupled via Riemann solves.



High-Order Methods

- Greater **resolving power** per degree of freedom (DOF)…
  - …and thus **fewer overall DOFs** for same accuracy.
- Tight **coupling between DOFs** inside of an element…
  - …reduces indirection and **saves memory bandwidth**.

High-Order Methods



- Direct extension into 2D and 3D.

Polynomial Multigrid

- Within the context of **implicit time-stepping** it is necessary to solve a **non-linear system of equations** of the form:

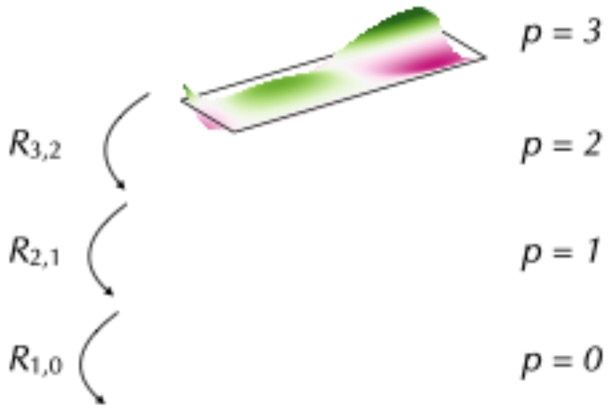
$$\mathbf{R}(\mathbf{u}) = 0,$$

where $\mathbf{u}$ is our solution and $\mathbf{R}$ a function.

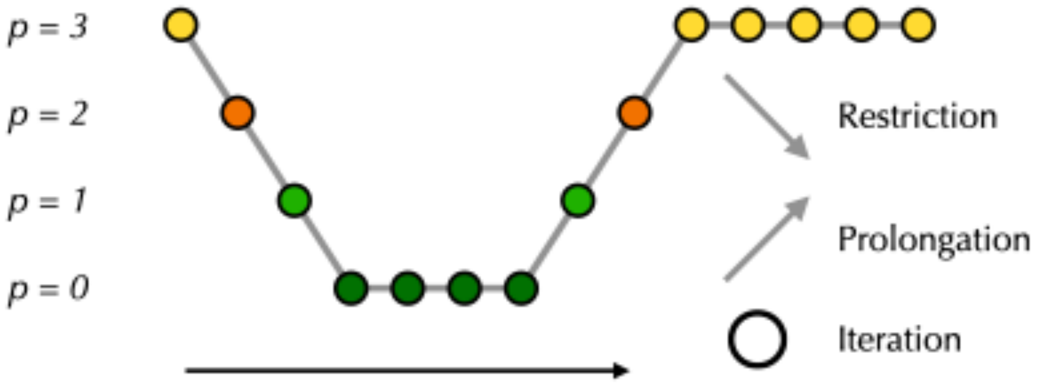
Polynomial Multigrid

- For real-world problems this system **must be solved iteratively**.
- We are therefore extremely interested in techniques for **accelerating convergence** with a powerful approach being **polynomial multigrid**.

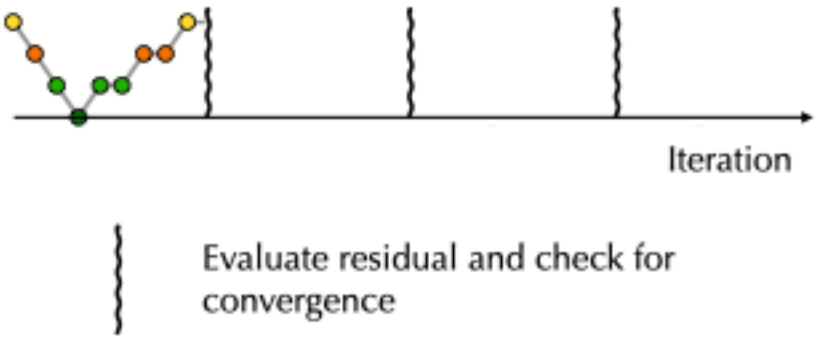
Polynomial Multigrid



Polynomial Multigrid

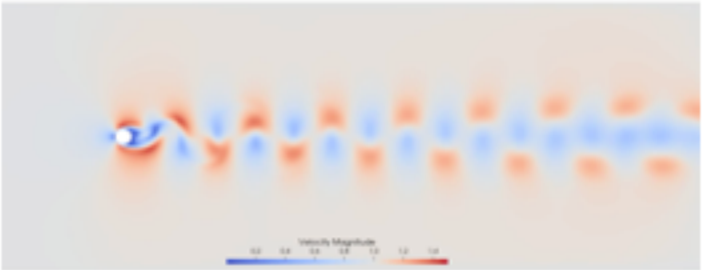


Polynomial Multigrid



Polynomial Multigrid

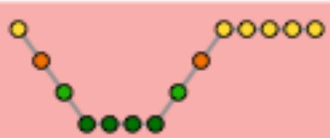
- Cycle configuration can have a **big impact on runtime performance**.
- Example: Incompressible 2D cylinder at *Re* = 200.



Polynomial Multigrid

8.3 ± 1.1 seconds per time unit

7.3 ± 0.6 seconds per time unit



13.6 ± 3.0 seconds per time unit

7.2 ± 1.6 seconds per time unit

Polynomial Multigrid

- Idea: use an optimiser to **learn an ideal cycle on the fly**.
- Problems:

  (i) Cycles are **arbitrary length**.

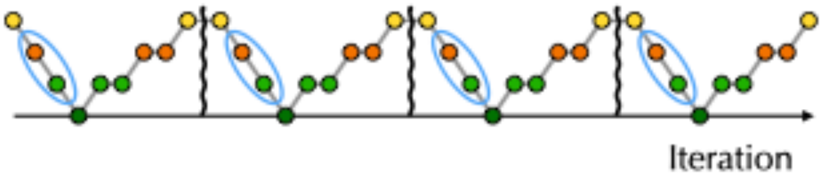
  (ii) Iterations are discrete leading to an **integer-programming problem**.

Parameterising Cycles

- Address the variable-length issue by **restricting ourselves to deep-V cycles with four parameters**.
- Such cycles have been employed in almost all real-world applications of polynomial multigrid for FR.

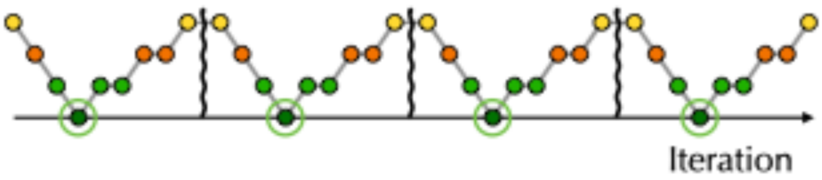
Parameterising Cycles

[**1**, 1, 2, 1]



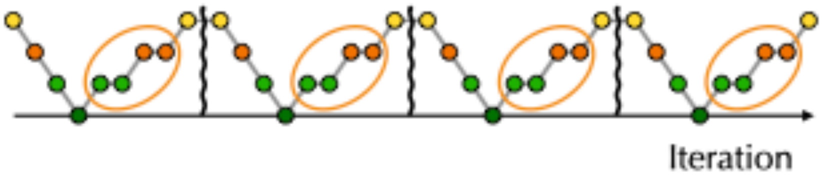
Parameterising Cycles

[1, **1**, 2, 1]



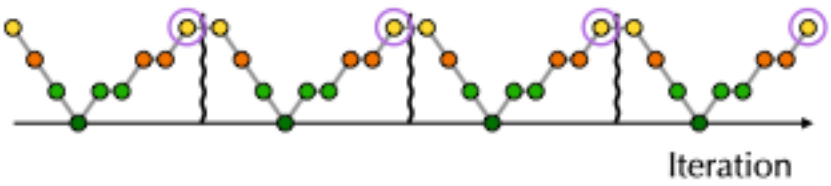
Parameterising Cycles

[1, 1, **2**, 1]



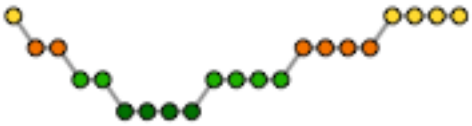
Parameterising Cycles

[1, 1, 2, **1**]

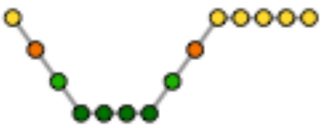


Parameterising Cycles

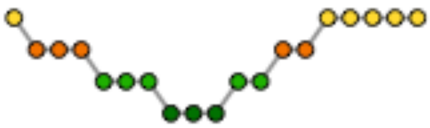
[1, 1, 1, 1]



[2, 4, 4, 4]



[1, 4, 1, 5]



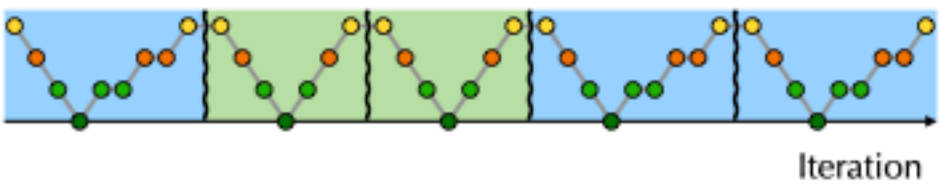
[3, 3, 2, 5]

Parameterising Cycles

- We enable cycles to have **fractional components** through **stochastic rounding**.
- This is a strategy which was pioneered in the **machine learning community** for enhancing the accuracy of reduced precision data types.

Parameterising Cycles

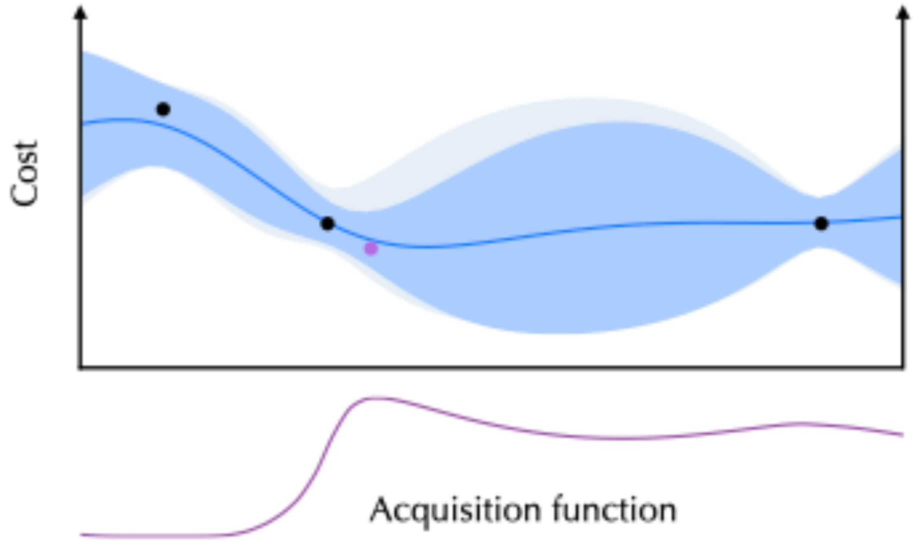
[1, 1, **1.6**, 1] = [1, 1, **1**, 1] with probability 0.4
[1, 1, **2**, 1] with probability 0.6



Bayesian Optimisation

- A powerful **gradient-free algorithm** for non-linear programming is **Bayesian optimisation**.
- It is particularly well suited to problems with **expensive objective functions**.

Bayesian Optimisation



Bayesian Optimisation

- There are several different types of acquisition functions with differing properties and computational costs.
- In this work we start with **Knowledge Gradient** [1] then switch to **Expected Improvement** [2].

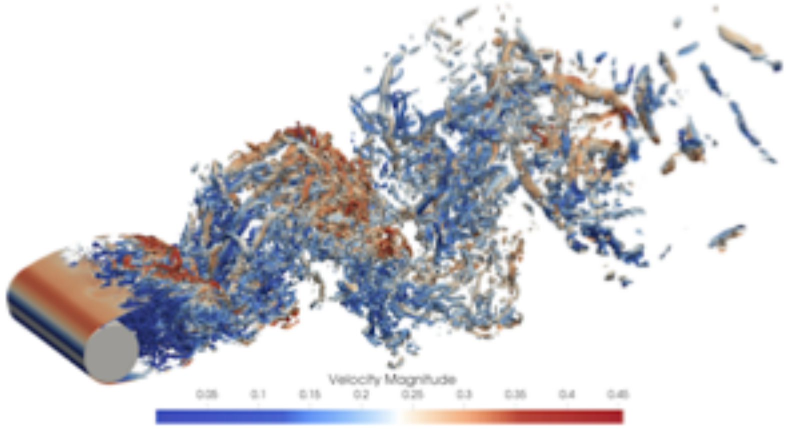
[1] J. Wu and P. Frazier. The parallel knowledge gradient method for batch Bayesian Optimization, 2016.
[2] P. Frazier. A Tutorial on Bayesian Optimization, 2018.

Bayesian Optimisation

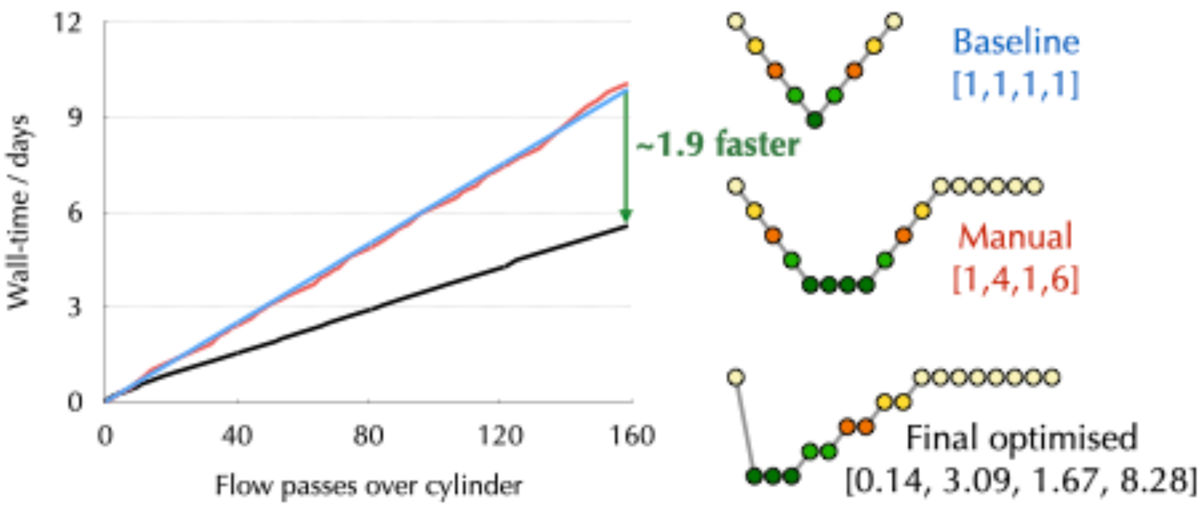
- To make the approach practical it is **necessary for the solver to be able to rewind itself** in case of a bad cycle.
- Additionally, code is required to **automatically adjust the domain** of the optimisation problem.

Results

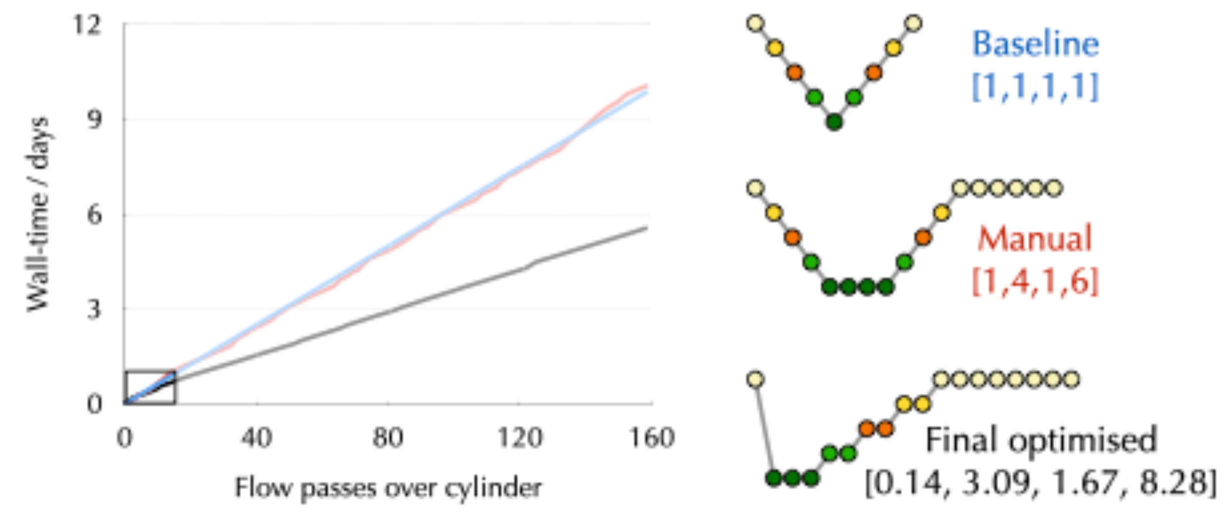
- Cylinder at *Re* = 3900 on 10 NVIDIA H100 GPUs.



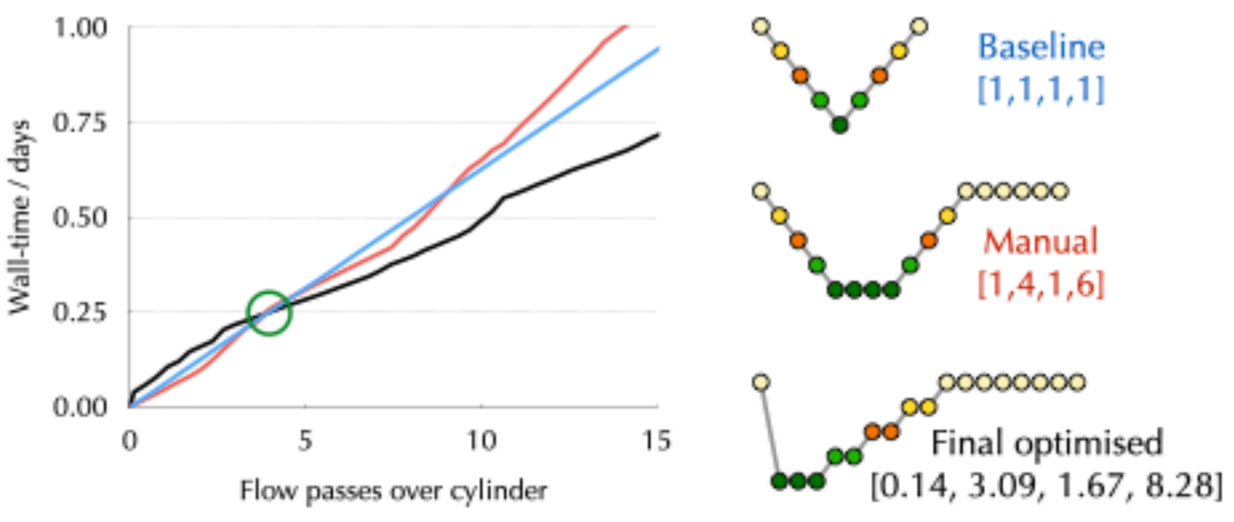
Results



Results



Results

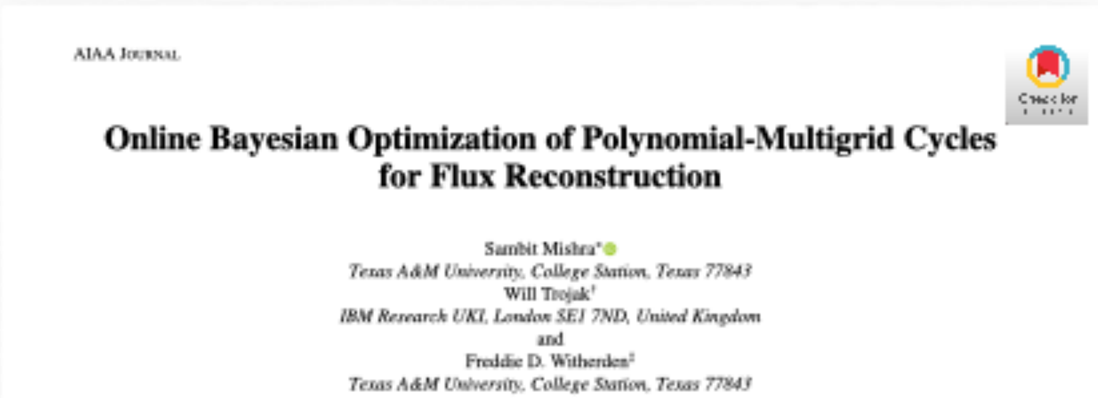


Conclusions

- Have shown how **Bayesian optimisation** can be used to learn polynomial multigrid cycles in real-time.
- Shown how this can lead to a **twofold improvement in time-to-solution** for unsteady turbulent flow problems.

Conclusions

- Further details: S. Mishra, W. Trojak, and F. D. Witherden. AIAA Journal, 2024.



AIAA Journal

Online Bayesian Optimization of Polynomial-Multigrid Cycles for Flux Reconstruction

Sambit Mishra*
Texas A&M University, College Station, Texas 77843
Will Trojak†
IBM Research UKI, London SE1 7ND, United Kingdom
and
Freddie D. Witherden‡
Texas A&M University, College Station, Texas 77843

Acknowledgements

- **Air Force Office of Scientific Research** for support under grant FA9550-23-1-0232.
- **National Science Foundation** for support under awards 2112356 (ACES) and 1925764 (SWEETER).

Backup Slides