GA Optimisation of CFD Models and Meshes

Open Source CFD International Conference

Dr Gavin Tabor, Bjoern Fabritius

CEMPS, University of Exeter

CFD Research – Exeter Group

Remit; to advance methodology and application of CFD – particularly

- Turbulence (+ other physical modelling) + optimisation
- Biomedical (blood, lymph, IBM)
- Industrial – SUDS, packed beds, tidal turbines

Group consists of 3+2 PhD Students, 2 PDRA, 1 PGRA

Facilities : 64 core 256GB cluster; workroom with 4 high performance workstations. Also access to ALM, micro-CT facilities

Substantial investment in OpenFOAM; also Fluent, Pointwise.

GA Optimisation of Turbulence Models

PhD project (Bjoern Fabritius) to investigate optimisation of turbulence models.

Rationale :

- Turbulence models complex, typically include several parameters (standard $k - \epsilon$ contains 5)
- Parameters often taken to be universal constants – are they?
- Attempts made to provide justification for values; more often just parameter-fit to data.
- Fine tuning accepted for certain canonical flows (eg. circular impinging jet)

Application to turbulence modelling

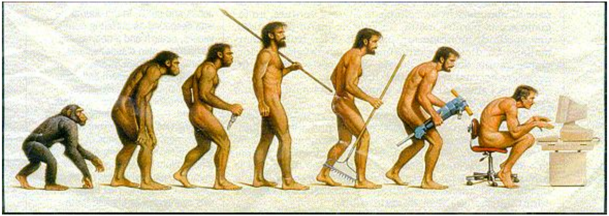
Aim to explore parameter space for particular turbulence models + demonstrate optimisation process for complex physical model.
Questions to answer;

- Are the standard parameters optimal for particular canonical flow problems (eg. BFS)?
- What are the tradeoffs between parameters for different flow problems (multi-objective optimisation)?
- Could we create a complete new model from scratch?

Also apply to meshing issues.

Outline of talk

1. Basics of GA and integration with OF
2. Optimisation of turbulence models
3. Optimisation of meshing with `snappyHexMesh`
4. Future directions



UNIVERSITY OF EXETER

What are GA's?

Genetic Algorithms – attempt to use Natural Selection techniques to "evolve" an optimal solution to a complex problem.

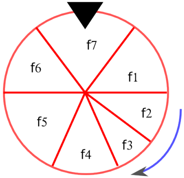
Methodology :

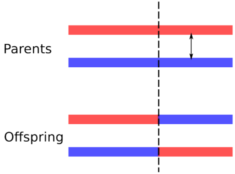
1. Develop coding for parameters of project (a genome)
2. Create a population of individuals with varying genomes
3. Evaluate fitness of individuals (run CFD code)
4. Eliminate "un-fit" individuals from gene pool
5. Create new population from retained individuals by
    - Exchanging genetic info (sex)
    - Random mutation
6. Repeat until convergence

Details

Various options possible (*artificial* process – not restricted)

1. Selection
   - roulette-wheel selection
   - tournament selection
   - best fitness selection
2. Cross-over (probability $P(C)$)
   - single-point crossover
   - multi-point crossover
   - uniform crossover
3. Mutation (probability $P(M)$)





Coding

- Use `Python` and `MPI` as technological basis
- pyFoam framework :
  - CloneCase – generate new individual
  - ParsedParameterFile – change model coefficients, write `RASProperties dictionary`
  - BasicRunner – execute solver/sample results
- Parallel evaluation of fitness function possible
- Used the 'strategy' design pattern → selection, fitness function, crossover methods interchangeable
- Toolkit extended to multi-objective optimisation

Coding (cont)

Master node

identify all available nodes
create *population*
while not terminal cond.
    send *individual* to free node
    receive *fitness*
until all *individuals* are evaluated
population.**evolve**

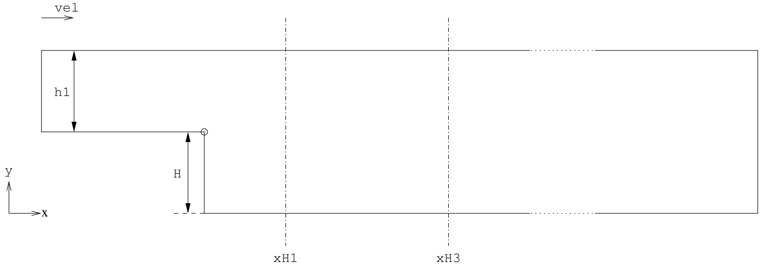
Slave nodes

receive *individual*
run solver / sample
compute *fitness*
send *fitness* to master

Run parameters can be controlled via `gaDict`
– model coefficients and limits can be specified

Typical results

Specific flow problems; backward facing step (Pitz-Daily case)



Model equations

Transport equations for k and ε

$$\frac{\partial(\rho k)}{\partial t} + \operatorname{div}(\rho k u) = \operatorname{div}\left[\frac{\mu_t}{\sigma_k}\operatorname{grad} k\right] + 2\mu_t S_{ij} \cdot S_{ij} - \rho\varepsilon$$

$$\frac{\partial(\rho \varepsilon)}{\partial t} + \operatorname{div}(\rho \varepsilon u) = \operatorname{div}\left[\frac{\mu_t}{\sigma_\varepsilon}\operatorname{grad} \varepsilon\right] + C_1 \frac{\varepsilon}{k} 2\mu_t S_{ij} \cdot S_{ij} - C_2 \rho \frac{\varepsilon^2}{k}$$

$$\mu_t = \rho C_\mu k^2 / \varepsilon$$

C_1, C_2 most significant parameters

Pitz-Daily Optimisation: $k - \epsilon$

- 50 individuals, 30 generations
- tournament selection, single point crossover
- k-ε model, Re=64,000
- simpleFoam on 10 cores, runtime approx. 2.5h

	C_1	C_2
Standard	1.44	1.92
Best Indiv.	1.91	1.86
Δ/%	32.6	-3.1

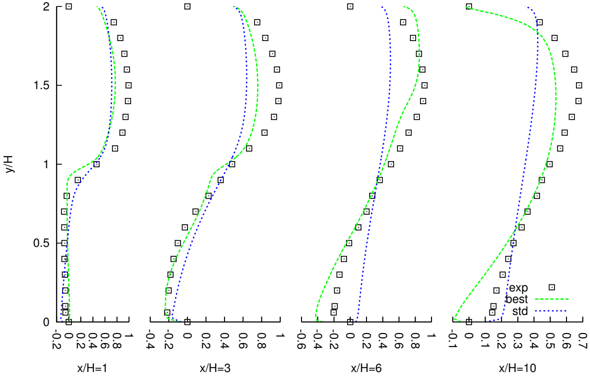
UNIVERSITY OF EXETER

Pitz-Daily Optimisation: $k - \omega$

- GA setup as above
- k-ω model, Re=64,000
- simpleFoam on 10 cores, runtime approx. 3.0h

	γ_1	γ_2	β_1	β_2	$\beta^\star$
Standard	0.553	0.440	0.075	0.083	0.09
Best Indiv.	0.606	0.510	0.053	0.076	0.095
$\Delta/\%$	9.5	15.9	-30.0	-8.4	5.5

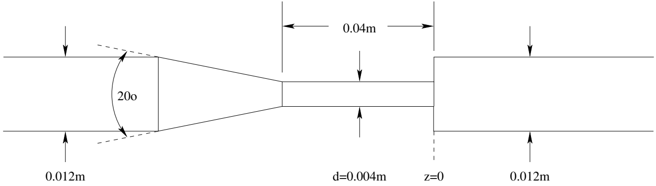
Pitz-Daily Optimisation: Velocity Profiles



Re=64,000 ; $k - \varepsilon$ model

FDA test case

Conical concentrator and sudden expansion

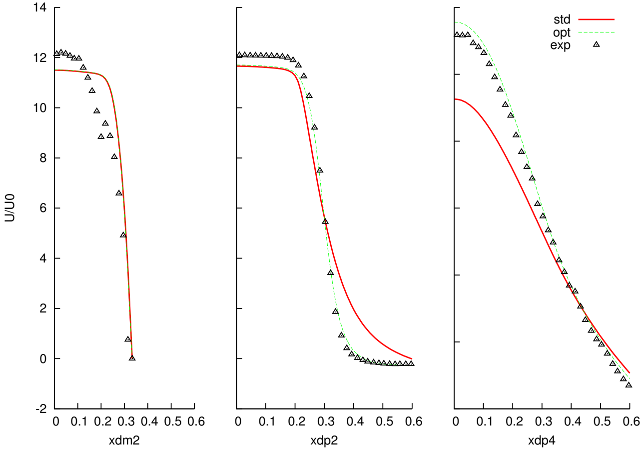


FDA test case – Spalart-Allmaras

- 50 individuals, 30 generations
- tournament selection, single point crossover, elitist
- Spalart-Allmaras model, Re=5,000 (throat)
- simpleFoam on 10 cores ⇒ runtime approx. 36h

	C_{b1}	C_{v1}	s	κ
Standard	0.1355	7.1	0.666	0.41
Best Indiv.	0.172	9.187	0.447	0.274
$\Delta/\%$	26.9	38.3	-32.9	-33.2

FDA test case



snappyHexMesh

OpenFOAM provides automated mesh generator `snappyHexMesh`. Its behaviour can be prescribed by setting parameters to control for example :

- total number of mesh cells
- thickness of boundary layers
- cell quality w.r.t.
  - cell skewness
  - mesh orthogonality
  - cell volume and face area

Finding the best setting for these parameters to create a high quality mesh is a good candidate for genetic optimisation.

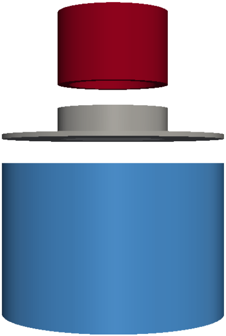
snappyHexMesh Functionality

`snappyHexMesh` algorithm takes these steps:

- create a base mesh that envelopes the target geometry completely
- load the geometry (STL format)
- castellate by identifying intersection between geometry and base mesh
- discard cells outside of target geometry
- snap boundary cells to target surfaces
- refine cells until requested quality is reached

snappyHexMesh Test Case

Bearing problem : 2 pipes + connector :



- small radii
- mesh inside the pipes
- round surfaces
- sharp corners

Optimisation Objectives

Aim of the optimisation was the minimisation of grid cells with a maximisation of cell quality. The result of this multi-objective problem is a trade-off between these two targets:

Quality Measure	target	bad individual	good individual
Skewness	low	3.759	1.372
Non-Orthogonality	low	86.2384	63.9662
Max aspect ratio	low	44.5033	37.1293
Grid size	low	42,574	60,972

(Higher fitness is highlighted.)

Results

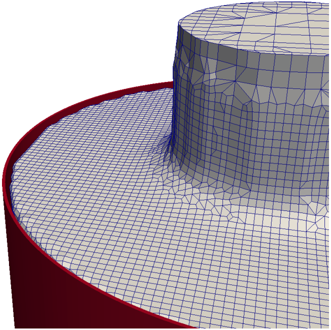


Figure: Bad Mesh Quality

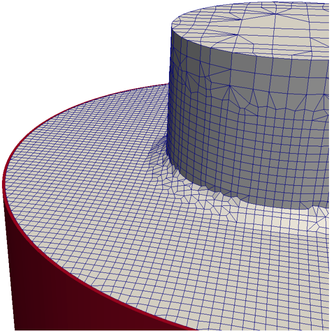


Figure: Optimised Mesh Quality

Conclusions

- We have developed a multi-objective GA framework for optimisation problems on top of OpenFOAM. OF provides excellent framework for these developments.
- Turbulence modelling can be considered a multi-parameter optimisation problem. Previously optimisation carried out ‘by hand’ – automated optimisation techniques preferable.
- Meshing can also be seen as multi-parameter optimisation process – have demonstrated potential here.

UNIVERSITY OF EXETER

Future Work

- Multi-objective optimisation for turbulence models – examine tradeoffs between different canonical flows.
- Application to other canonical flows (have already examined impinging jet case).
- Application of meshing techniques to more complex cases (Ahmed body, concept car).
- GA optimisation of flow cases, automated surrogate modelling (recent blood flow project).

Acknowledgements

GA Optimisation Prof Eric Paterson, Prof Hrvoje Jasak, Prof Richard Everson, Dr Bernhard Gschaider (PyFoam)

"Use of Evolutionary Algorithms in the Analysis and Optimisation of Turbulence Models", B.Fabritius, G.Tabor. *Seventh International Conference on Computational Fluid Dynamics* (ICCFD7) Hawaii (2012)

Submitted paper on GA optimisation of turbulence models to *Computers and Fluids*.

UNIVERSITY OF EXETER

Shameless plug

IACM/ECCOMAS Conference
20-25th July 2014,
Barcelona

www.wccm-eccm-ecfd2014.org

Minisymposium: CCM with OpenFOAM^{TM}

Please Contribute!
(Abstract submission 29th Nov)