

# OpenFOAM : Origin and Applications

An Introduction to OpenFOAM and HPC – Loughborough University

G. Tabor

CEMPS, University of Exeter

8th June 2015

# Presentation: Overview

1 OpenFOAM – Background

2 Resources

3 Examples – User

4 Examples – Modelling

5 Advanced Programming

6 Conclusions

# Presentation: Overview

1 OpenFOAM – Background

2 Resources

3 Examples – User

4 Examples – Modelling

5 Advanced Programming

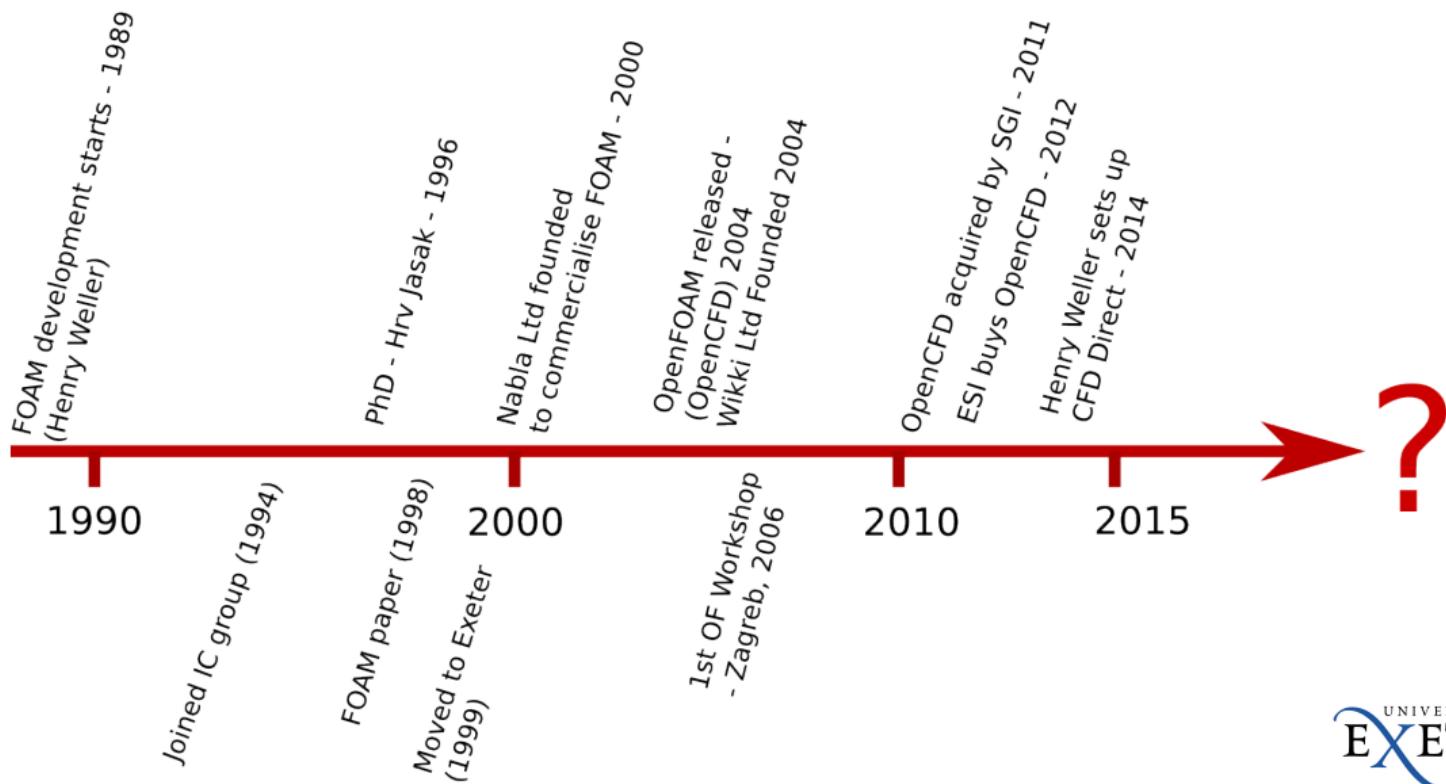
6 Conclusions

# What is OpenFOAM?

OpenFOAM is an Open Source CCM (predominantly CFD) code :

- Written in C++
- Based on FVM on arbitrary unstructured (polyhedral cell) meshes
- Originally developed by Henry Weller and others at IC (1990 – 2000) as FOAM (**Field Operation And Manipulation**)
- Released 2004 under Gnu GPL by OpenCFD Ltd
- Now one of several independent versions and developments (-dev, pyFoam)
- Extensive user community
- Academic and commercial usage.

## Timeline



# OpenFOAM Advantages

- ① Complete transparency and code availability (vs. “grey box” approach for commercial codes)
- ② Information exchange with other practitioners at a fundamental level
- ③ CFD use not rationed by license fees.

OpenFOAM comes with extensive pre-written solvers – can still be used as a “black box” CFD tool.

**Open Source** nature of OpenFOAM allows easy alteration of solvers and code.

**Open Source** software development encourages code sharing and information exchange. OpenFOAM’s structure and use of C++ is ideal for this and provides a common platform for CFD research.

# Types of use

Three possible “levels” of use of the code :

- User
- use prewritten apps
  - “black box” use of code
  - CFD use *free at the point of use*

# Types of use

Three possible “levels” of use of the code :

## User

- use prewritten apps
- “black box” use of code
- CFD use *free at the point of use*

## Modeller

- Use existing classes to implement new physical models
- ”MatLab for CFD”

# Types of use

Three possible “levels” of use of the code :

User     

- use prewritten apps

- “black box” use of code

- CFD use *free at the point of use*

Modeller     

- Use existing classes to implement new physical models
- ”MatLab for CFD”

Guru     

- Redevelop/extend classes

- Implement (and release) substantial new developments

# Presentation: Overview

1 OpenFOAM – Background

2 Resources

3 Examples – User

4 Examples – Modelling

5 Advanced Programming

6 Conclusions

## Learning resources

OpenFOAM distribution comes with user guide and tutorials

*doxygen* code – run to create html-based class library list

However this is probably inadequate – numerous other resources available  
(free/commercial)

Several commercial companies running training courses – OpenCFD, Wikki, ICON, ...

University courses at UG/PG levels – e.g. ECMM148 Advanced CFD

Book “The OpenFOAM Technology Primer” – Tomislav Marić, Jens Höpken, Kyle Mooney – [Sourceflux](#)

## Online resources

Prof Hakan Nilsson (Chalmers University) runs an MSc course on CFD with OpenFOAM – students undertake individual modelling activities, write reports which are then downloadable.

See e.g. [http://www.tfd.chalmers.se/~hani/kurser/OS\\_CFD\\_2010/](http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2010/)

Several people have put OpenFOAM tutorials on YouTube – e.g. Dr Jozsef Nagy (Johannes Kepler University Linz)

Community resource – [OpenFOAM Extend project](#)

[PyFoam](#), written by Bernhard Gschaider (ICE Strömungsforschung) – Python wrapper for OpenFOAM for code control (etc)

OpenFOAM Foundation – [CFD Direct](#)

# Community activities

*OpenFOAM Workshop* – series of community-led workshops

- First workshop – Zagreb, Croatia (2006)
- Typically 4-day events, including presentations, posters, *free training*, networking sessions
- Latest – 10th OpenFOAM Workshop: U.Michigan 29 June - 2nd July 2015
- <http://www.ofw10.org/>

Regional user groups – based on German *stammtisch* model

- UKRI User Group (established 2014) – meetings CFMS Bristol, U.Leeds
- Next meeting – STFC Daresbury – 2nd-4th November 2015
- <https://eventbooking.stfc.ac.uk/news-events/openfoam-user-meeting>

# Presentation: Overview

1 OpenFOAM – Background

2 Resources

3 Examples – User

4 Examples – Modelling

5 Advanced Programming

6 Conclusions

# OpenFOAM – the User

Provides a fully functioning CFD (CCM) code with cutting edge capabilities.

Individual codes (apps) for specific tasks – “Ronseal philosophy”

Standard layout for files – *case* directory containing timesteps, constant, system subdirectories

Mesh generation with native apps or external tools (e.g. Pointwise) – postprocessing with paraView

Apps covering; compressible/incompressible flow, turbulence (RANS/LES), mesh motion (all types), non Newtonian fluids, multiphase and free surface flow, stress analysis and FSI, reaction, combustion ...

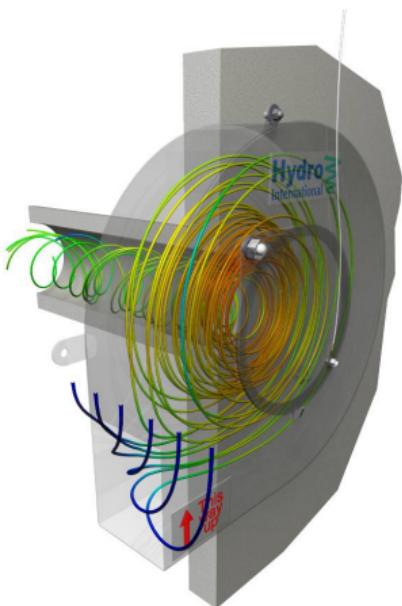
# Sustainable Urban Drainage Systems

Longstanding collaboration with Hydro International – UK leading company in SUDS

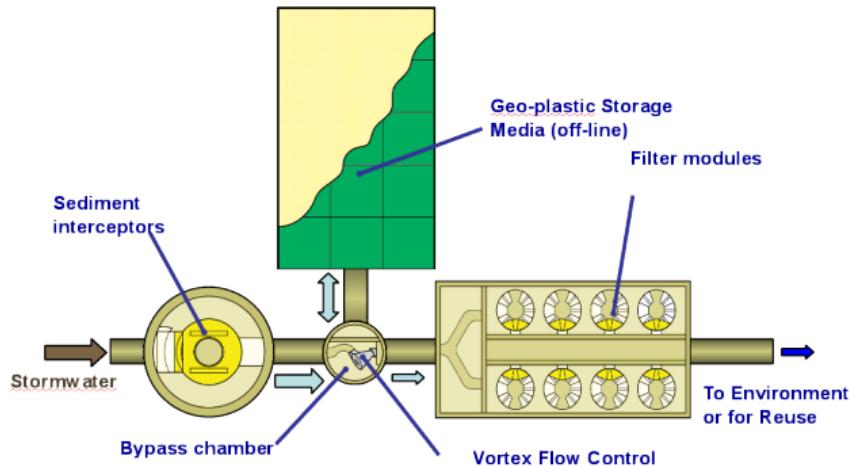
Vortex Flow Control; device used in SUDS to control stormwater surge through drainage system.

- ① Industrial PhD project – Dr Dan Jarman
- ② Subsequent UG group project

Valuable demonstration of value of Open Source CFD in industrial setting; cost per run *considerably* lower than for commercial use.



# Vortex Flow Controls



# VFC – Desired physical behaviour

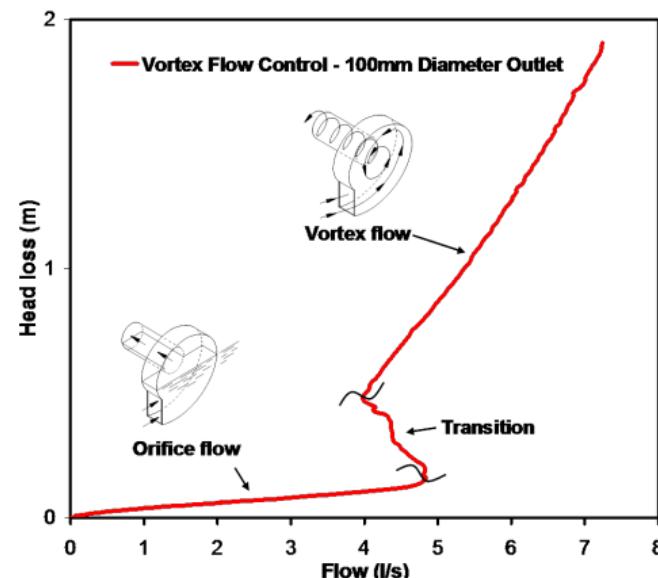
Required bi-stable system :

- Low flow rate – low head loss
- High flow rate – high head loss

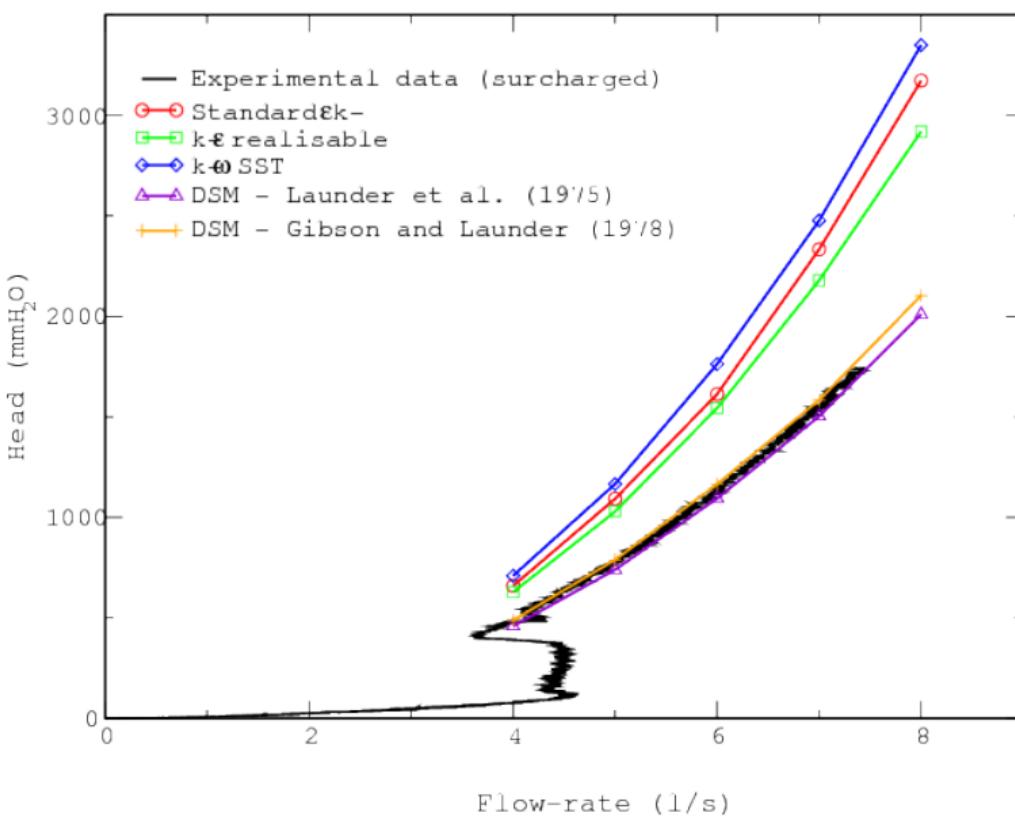
VFC device designed to create vortex at high flow.

Need to be able to predict (and design) curve for given situation.

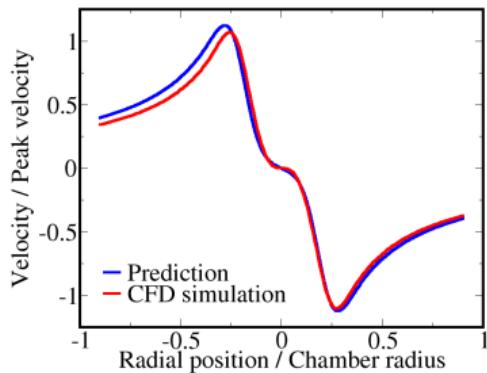
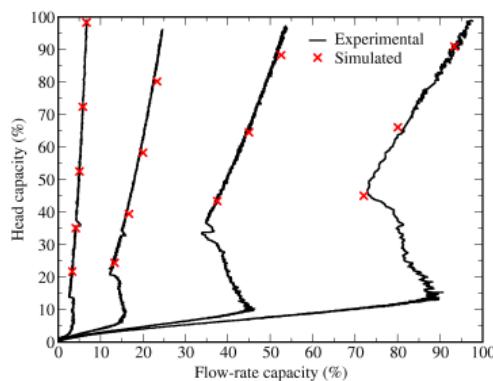
Examine surcharged and free-surface vortexing behaviour with CFD. Turbulence modelling known challenge



# Head loss



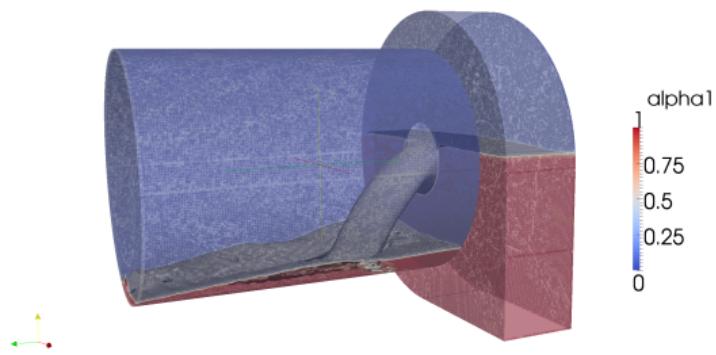
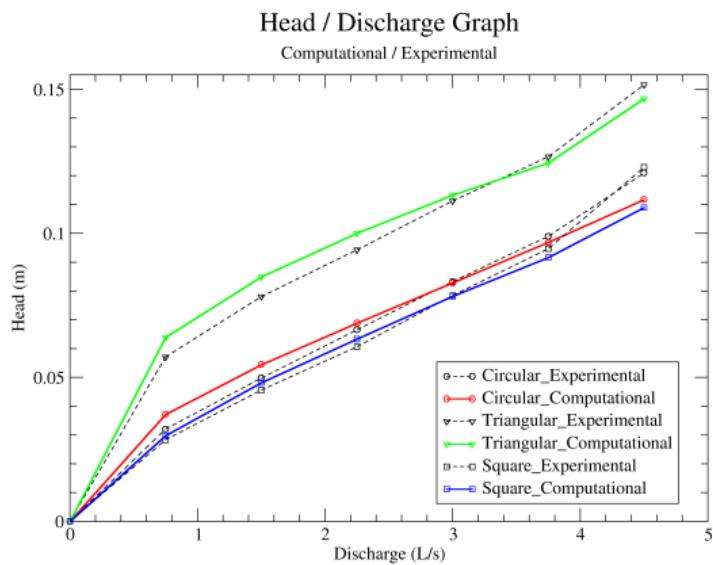
# Analysis



- CFD compares well with experiment, theory
  - weir flow for low flow rates, Sullivan vortex theory high flow rates
- Repeat calculation for different hydraulic capacity, swirl parameters
- Greater understanding of flow – input into design process (new designs for VFC)
- Very significant increase in CFD capability through adoption of open source software

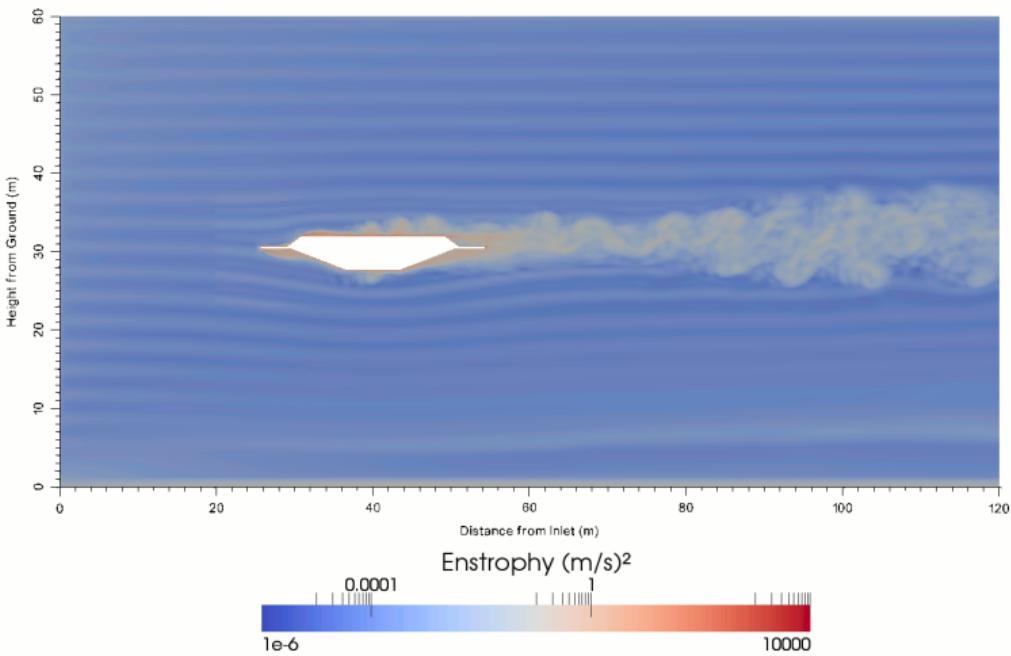
# Lower flow rates

Lower flow rates examined by UG project group (final years); 3 different shaped orifices; VOF free surface flow compared with experiment and theory.



# DES of Humber Bridge Section

Student project 2014/2015 :



# Presentation: Overview

1 OpenFOAM – Background

2 Resources

3 Examples – User

4 Examples – Modelling

5 Advanced Programming

6 Conclusions

# OpenFOAM – the Modeller

Strictly, OpenFOAM is **not** a CFD code – it is a C++ library of classes for writing CFD codes.

OpenFOAM uses the full range of the C++ language – inheritance, polymorphism, templating, operator overloading etc – where appropriate :

- Class mechanism – define new “types” for CFD
- Operator Overloading – provides standard mathematical syntax
- Inheritance, polymorphism etc – encodes relationships between conceptual entities in code
- Interface vs implementation : segregation of effort.

Effective result is a high level “language” for encoding CFD.

At the highest level, provides a framework for significant development projects

# Example : Burgers equation

1d Burgers equation :

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}$$

3d version (conservative form) :

$$\frac{\partial \underline{u}}{\partial t} + \frac{1}{2} \nabla \cdot (\underline{u} \underline{u}) = \nu \nabla^2 \underline{u}$$

Implemented in OpenFOAM as :

```
fvVectorMatrix UEqn
(
    fvm::ddt(U)
    + 0.5*fvm::div(phi, U)
    - fvm::laplacian(nu, U)
);

UEqn.solve();
```

## Enclose in loop and iterate :

```
while (runTime.loop())
{
    Info << "Time = " << runTime.timeName() << nl << endl;

    #include "CourantNo.H"

    fvVectorMatrix UEqn
    (
        fvm::ddt(U)
        + 0.5*fvm::div(phi, U)
        - fvm::laplacian(nu, U)
    );

    UEqn.solve();

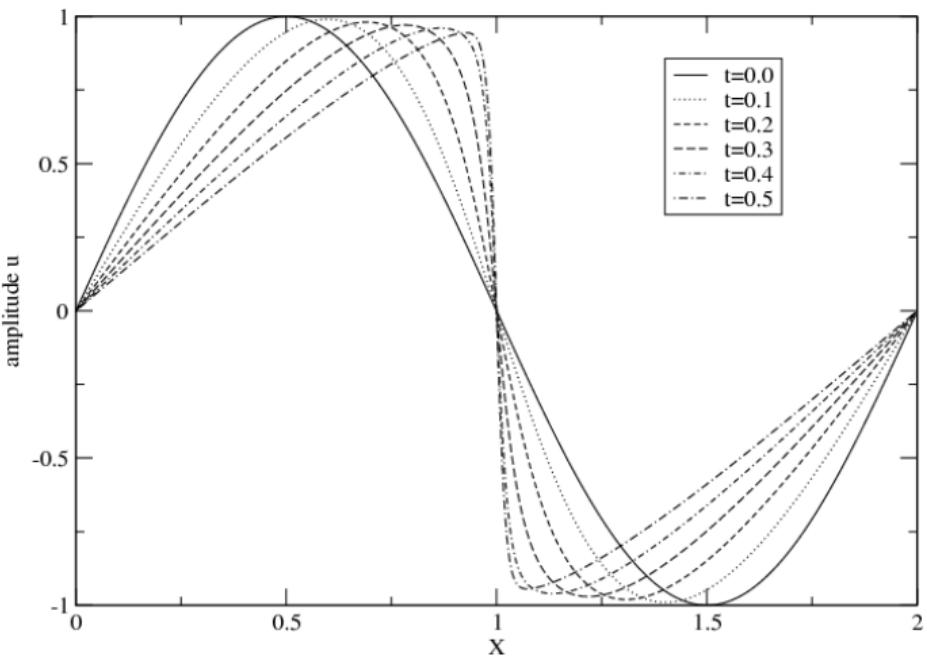
    U.correctBoundaryConditions();

    phi = (fvc::interpolate(U) & mesh.Sf());

    runTime.write();

    Info << "ExecutionTime = " << runTime.elapsedCpuTime() << " us"
        << " ClockTime = " << runTime.elapsedClockTime() << " us"
        << nl << endl;
}
```

## Test case : 1-d sine wave



# Tidal Turbines

Novel design for tidal turbine based on cycloidal turbine involving complex rotating airfoil blades

- Blades act in drag mode on one side; rotate ( $0.5\Omega$ ) to develop lift on other side
- Unit operates as cross-flow turbine
- Energy extracted through volume – high efficiency (measured efficiency of  $\sim 50\%$ )
- High blockage factor; suitable for near-surface (eg. estuarine) sites.
- Likely deployment in very large arrays



# Turbine modelling – simplified

Full simulation very expensive to run – need something cheaper

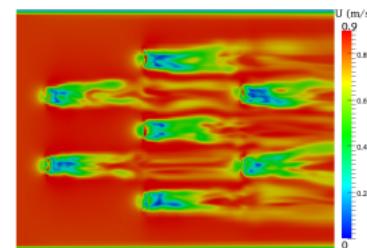
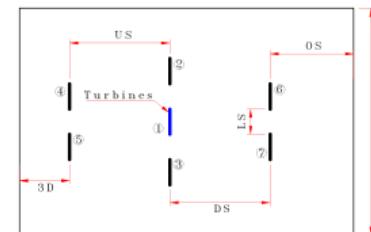
Immersed body force method :

- Blades represented by stationary body forces

$$\bar{\mathbf{F}} = \bar{\mathbf{F}}_D + \bar{\mathbf{F}}_L$$

- Plus 'vortex ring' body forces
- Compromise between accuracy and efficiency
- Capable of representing large scale vortexes – 3d LES

Able to compute power, wake recovery for different turbine loadings within a farm.



# Farm Modelling

Ultimate aim to optimise farm of 10's or 100's of devices, optimise based on position, loading factor etc. Targets; power output, cost

Most suitable technique – *Genetic Algorithm*. Capable of exploring complex N-d parameter space and reliably identifying optimum (Pareto front).

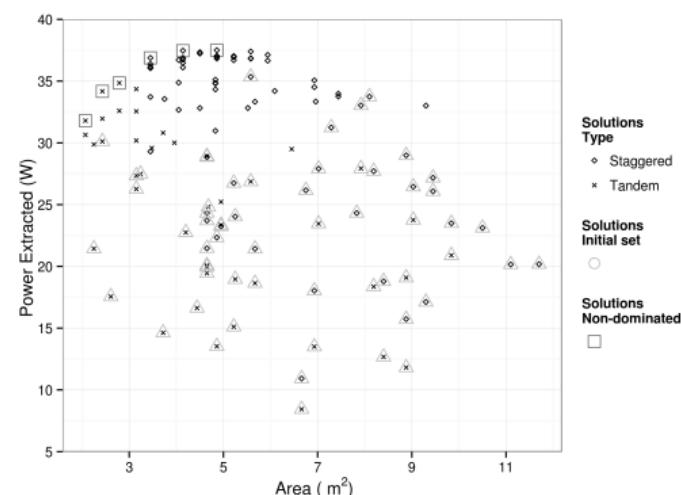
Case	Details	Spec	Time
GGI	160k cells, 30 revolutions	16 cores	5 days
IBF	1 turbine, 148k cells	12 cores	17 hrs
IBF	7 turbines, 1M cells	12 cores	44 hrs

Current task to develop *surrogate model* – run 10's of simulations and use *Kriging* to mine results and create correlation.

# Optimisation

Preliminary results :

- 3 row farms – 2 alignments (staggered/tandem)
- 6 parameters (592704 layouts in total)
- Surrogate model based on initial sample of 30 solutions (per alignment)
  - create using Latin Hypercube sample
- Optimisation – GA evaluation using surrogate – evaluate new solutions using farm model



# Presentation: Overview

1 OpenFOAM – Background

2 Resources

3 Examples – User

4 Examples – Modelling

5 Advanced Programming

6 Conclusions

# Surface water runoff – FRMRC-II project

Simulation of surface water runoff from roadway to gully pot.

- Critical point of coupling between 2d (surface flow) and 1d (pipe network) models
- Experimental data (Sheffield) to validate CFD models
- Vary all aspects of geometry + flow conditions
- Develop empirical models for energy loss



# CFD modelling

Problem must deal with both shallow depth effects (over the road surface) and deep flow (in the gully pot). Initial simulations performed using VOF in 3d (`interFoam`)

Problem; mesh resolution for thin surface flow makes mesh requirements prohibitive over road surface

Thus developed Finite Area approach (surface flow) coupled to VOF (gully pot).

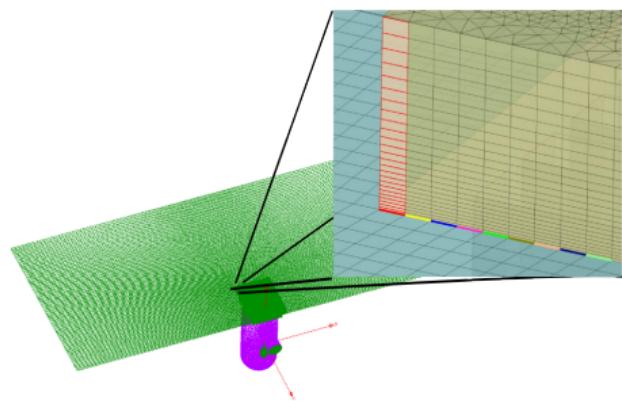
VOF solver using PISO,  $k - \omega$  turbulence closure

Finite area – derive shallow water equations given prescribed velocity profile (boundary layer approximation applicable)

# 2d/3d interface

3d block embedded within 2d domain – new boundary condition derived to couple regions together

- Need to switch velocity and pressure between fixed value and zero gradient depending on direction of flow
- Transfer 2d $\rightarrow$ 3d – depth function becomes  $\alpha$ ; prescribed velocity profile
- Transfer 3d $\rightarrow$ 2d –  $\alpha$  converts to depth, average velocity

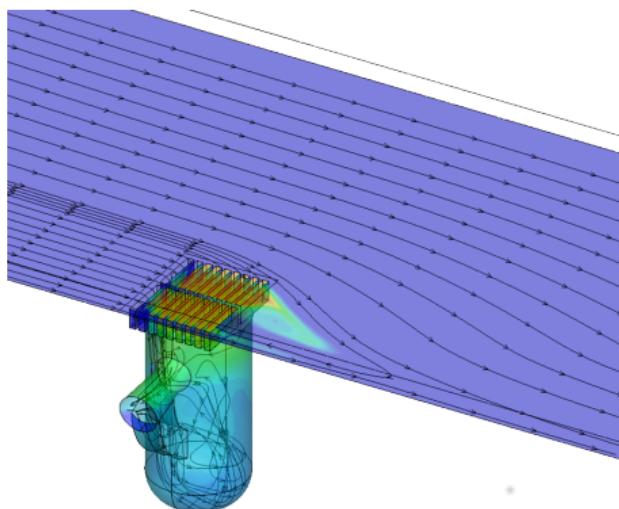
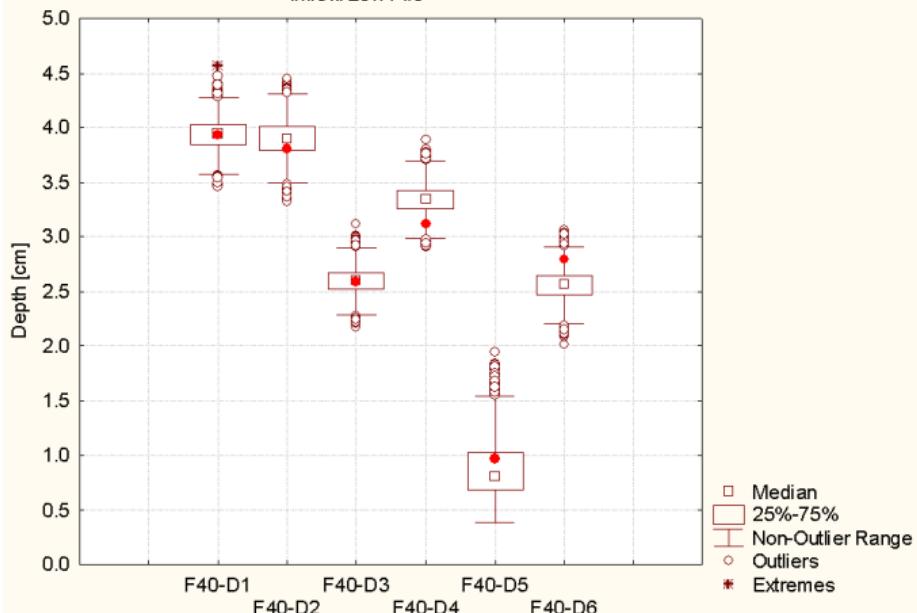


# Results

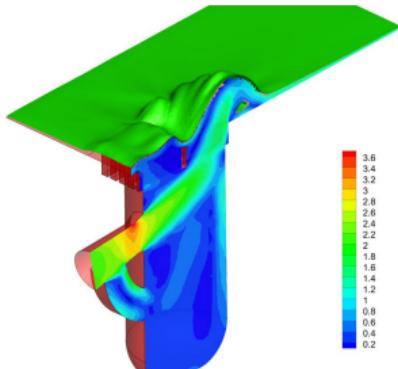
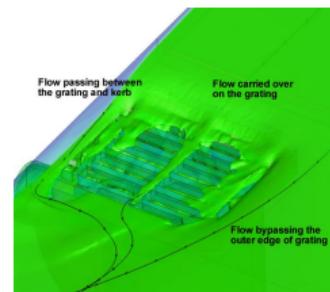
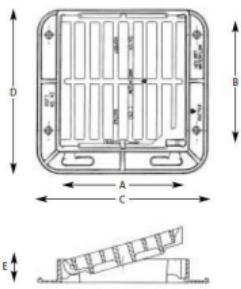
Results validated against experiment for several flow rates (and surcharging)

Comparison of observed and simulated depth

Inflow: 26.17 l/s



# Analysis



CFD models can be used to probe details of flow and derive valuable relationships (eg. Flow-Head curves)

- Three different gratings used – geometry resolved
- Details of interception and bypass flow analysed
- Derived empirical relations for surface/subsurface coupling
- Analysed factors affecting grating efficiency

# 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)
- Attempts made to provide justification for values; more often just parameter-fit to data.
- Parameters often taken to be universal constants – are they?
- Fine tuning accepted for certain canonical flows (eg. circular impinging jet)

Aim of project was to use modern optimisation techniques to explore parameter space and look for optimal values

## Application to turbulence modelling

GA's can explore parameter space and evolve optimal solutions to complex multi-parameter problems.

Created optimisation toolkit in PyFoam with OpenFOAM running CFD simulations.  
GA algorithm written in Python accessing hacked OF apps to evaluate objective function(s)

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

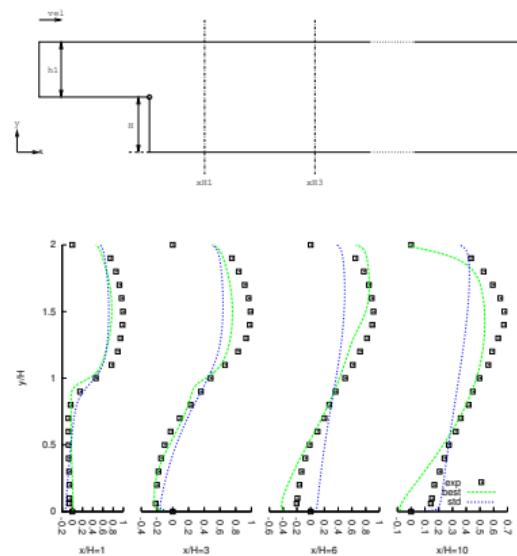
- Are the standard parameters optimal for particular canonical flow problems?
- What are the tradeoffs between parameters for different flow problems (multi-objective optimisation)?

Same process applied to optimise mesh generation using snappyHexMesh – objective function evaluated from rewritten OF apps

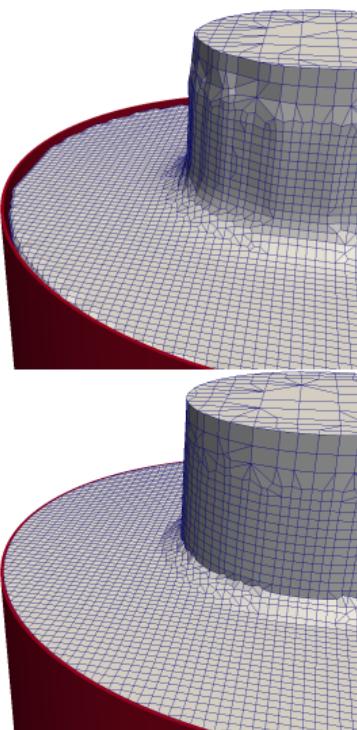
# Pitz-Daily Optimisation: $k - \epsilon$

- 50 individuals, 30 generations
- tournament selection, single point crossover
- $k-\epsilon$  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
$\Delta/\%$	<b>32.6</b>	<b>-3.1</b>



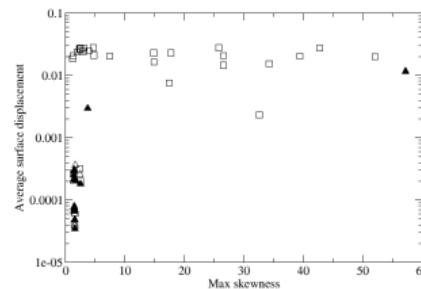
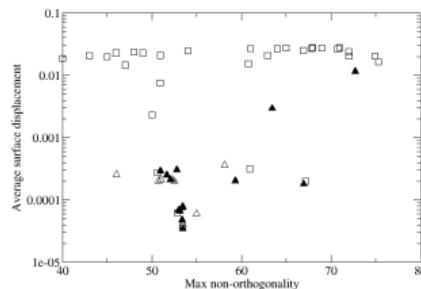
# Meshing



Meshing critical step in CFD process – demanding and time-consuming

Tools such as `snappyHexMesh` robust – but setting parameters is tricky

Solution : treat as multi-parameter, multi-objective optimisation process



# Presentation: Overview

1 OpenFOAM – Background

2 Resources

3 Examples – User

4 Examples – Modelling

5 Advanced Programming

6 Conclusions

# Conclusions

OpenFOAM is fully-functioned CFD code : use as “black box” or extend as needed :

**Academic use** Open source environment allows complete flexibility to understand and modify code; framework for (collaborative) projects and code sharing

**Industrially** Cutting edge CFD code with no restrictions on usage – particularly important with HPC

# Acknowledgements

**Exeter:** Prof David Butler, Prof Slobodan Djordjevic

**Students:** Dr Istvan Galambos, Shenan Grossberg, Ben Jankauskas

**Portugal:** Dr Jorge Leandro, Pedro Lopes

**Hydro International:** Dr Dan Jarman

# 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

Part of CEMPS – Centre for Water Systems – also significant research in materials, climate change (UK Met Office), Machine Learning, Astrophysics, Tidal (and other) renewable energy ...

# Research group

Current composition: 4 PhD students; 1 visiting student; 2 PDRA :

- Shenan Grossberg – Adjoint Optimisation applied to vortex separators
- Ed Shorthouse – Development of surrogate models for built environment
- Miriam Garcia – Flood risk for estuarine tidal turbines
- Ben Jankauskas – Hydrodynamic vortex separators as reaction vessels
- Pedro Lopes (U.Coimbra, Portugal) – Air entrainment at stepped spillways.
- Dr Muluaalem Gebreslassie – Modelling and optimisation of tidal turbine arrays
- Dr Recep Karaman – phase change in AM HX

Resources; computer lab (5 Linux workstations) – 64 core parallel machine ‘Callisto’ – access to 1920 core University machine ‘Zen’ – MRI, micro-CT scanner, 3d printing facilities (CALM)

## References and publications

"CFD of Vortex Flow Controls at high flow rates", D. Jarman, D.Butler and G. Tabor, *ICE.Proc: Engineering and Computational Mechanics* **168#EM1** pp. 17 – 34 (2014)

"Computational investigation of vortex flow controls at low flow rates", G.Queguineur, D. Jarman, E.Paterson, G.Tabor, *ICE.Proc: Engineering and Computational Mechanics* **166 # 4** pp. 211–221 (2013)

"Experimental and numerical investigation of interactions between above and below ground drainage systems", S. Djordjevic, A.J.Saul, G. Tabor, J.R.Blanksby, I.Galambos, I. Sabtu, G.Sailor, *Water Science and Technology* **67#3** pp.535-542 (2013)

"Vortex flow control performance in response to single and time-series rainfall events", D. Jarman, P. LeCornu, G. Tabor, D.Butler, *Computing and Control in the Water Industry (CCWI)*, University of Exeter, 5-7th September (2011).

"Application of Open Source CFD in Urban Water Management", G.Tabor, D.Jarman, R.Andoh, D.Butler, I.Galambos, S.Djordjevic, *ASCE Conf. Proc.* doi:10.1061/41173(414)153. *World Environmental and Water Resources Congress (EWRI)*, Palm Springs, California, May 22-26th (2011).

"The use of CFD coupled with physical testing to develop a new range of vortex flow controls with attributes approaching the ideal flow control device", D.S.Jarman, G.Tabor, D.Butler, R.Y.G.Andoh, *ASCE Conf. Proc.* DOI:10.1061/41173(414)156 *World Environmental and Water Resources Congress (EWRI)*, Palm Springs, California, May 22-26th (2011).