OpenFOAM®
in wastewater applications:
1 - Introduction

nelson.marques@bluecape.com.pt

13-14th June 2015
Progress

**Introduction**
- OpenFOAM characterization
- blueCFD installation

**Getting started**
- Simulation process overview

**Meshing**
- Meshers in OpenFOAM

**Hands-On: UV disinfection**
- Simple 2-lamps system

**Hands-On: Clarifier**
- Solids settling problem

**Simulation**
- How to setup and run a case

**Hands-On: Splitter box**
- Monitoring flow repartition

**Advanced topics**
Contents

1. What is OpenFOAM®?
2. Short history of OpenFOAM
3. Ecosystem around OpenFOAM
4. Operating Systems and Standards
5. What is blueCFD®-Core?
6. Installing blueCFD-Core
7. Overview of installed packages
8. Overview of installation directory
9. Getting started with the interface
What is OpenFOAM®? (1/3)

- OpenFOAM® is essentially an open-source software package that is primarily meant to be used as toolbox for applying the principles, methods and modelling strategies conceived in the field of Computational Fluid Dynamics.
- The acronym FOAM stands for "Field Operation and Manipulation".
- It is maintained and delivered by the OpenFOAM Foundation: www.openfoam.org
- OPENFOAM and OpenCFD are registered trademarks of OpenCFD Ltd (ESI Group): www.openfoam.com
- OpenFOAM as an open-source software package, is licensed under the GNU General Public License v3 (GPLv3): www.gnu.org/licenses/gpl.html
• Users are free to use OpenFOAM software, which can be freely used and modified by each user in any field (personal, academic or commercial), without any licensing fees, as long as GPLv3 license terms are respected.

• The modifications to the source code only have to be made available to whom the binary packages are provided.

• In many simulation scenarios, OpenFOAM is ready to be used after installing.

• Nonetheless, not all modelling strategies are available out-of-the-box and the user may have to code a new modelling strategy, or deploy one already made available by the community that uses OpenFOAM.
What is OpenFOAM®? (3/3)

A major evolution!

• The original FOAM software was created by Henry Weller in 1989.
• Development of FOAM was done in an academic environment until 2000, including collaborative development.
• FOAM was commercialized as a CFD source code toolbox between 2000 and 2004 by the company Nabla Ltd.
• After the closure of Nabla Ltd in 2004, FOAM was modified, improved and released as open-source by OpenCFD on the 10th of December 2004, with the new name "OpenFOAM".
• The trade marks OPENFOAM and OPENCFD were registered ~2 years later, to help deter any abuse.
• OpenCFD was bought by SGI in 2011 and the OpenFOAM Foundation was created at the same time.
• The Foundation was created to ensure the source code remains open-source and the copyright is respected, independently of the trade mark.
• OpenCFD was later bought by ESI in 2012.
• In 2014, Henry Weller left OpenCFD/ESI and remains as director of the Foundation.
• 2015: Development in OpenFOAM continues to evolve, done by the those at the Foundation.
Although we have mostly mentioned Henry Weller as the original author, there have been a lot of contributions from several people and companies that have worked directly with him throughout FOAM/OpenFOAM's life span.

Contributions are welcome and guidelines are outlined here: www.openfoam.org/dev.php

References:
- http://cfd.direct/openfoam/
• The community that uses the technology mostly uses these forums:
  • [www.cfd-online.com/Forums/openfoam/](http://www.cfd-online.com/Forums/openfoam/)
  • [www.extend-project.de](http://www.extend-project.de)
• The unofficial wiki, driven by the community: [openfoamwiki.net](http://openfoamwiki.net)
• The main public open-source forks of OpenFOAM:
  • foam-extend ([foam-extend.org](http://foam-extend.org)) is a community driven fork of OpenFOAM, mostly developed by Wikki Ltd: [wikki.co.uk](http://wikki.co.uk)
  • Caelus-CML is another fork of OpenFOAM done by Applied CCM: [www.caelus-cml.com](http://www.caelus-cml.com)
Ecosystem around OpenFOAM® (2/3)

• There are several variants of OpenFOAM, where most were created for adding support into the source code for working in other Operating Systems (Windows and Mac OS X).

• Complete list of forks and variants: 
  openfoamwiki.net/index.php/Forks_and_Variants

• List of available forks/variants for Windows: 
  http://openfoamwiki.net/index.php/Windows

• List of available forks/variants for Mac OS X: 
  openfoamwiki.net/index.php/Installation/Mac_OS
Ecosystem around OpenFOAM® (3/3)

• Major contributions done by the community as toolboxes:
  • PyFoam is a Python based scripting toolkit, which enhances the abilities for using OpenFOAM from the command line: openfoamwiki.net/index.php/Contrib/PyFoam
  • swak4Foam is a toolkit designed for users that don't know C++, making it easier to use simple mathematical code in utilities, boundary conditions and post-processing tools: openfoamwiki.net/index.php/Contrib/swak4Foam
  • All known community contributions:
    • openfoamwiki.net/index.php/Contrib
    • openfoamwiki.net/index.php/Extend-bazaar
• Before 1980, one of the most common operating system (OS) was Unix, of which there were several variants, most incompatible with each other.
• In 1981 MS-DOS was released, which was completely incompatible with Unix systems, but was easier to use.
• The first Mac OS was released in 1984, an alternative to all other operating systems.
• Microsoft Windows 1.0 was released in 1985.
• In 1988 was published the first POSIX standard, in an effort to standardize compatibility between operating systems, at least for those akin to Unix.
• Linux was first released in 1991. Later on it was named GNU/Linux.
• Mac OS X was released in 2001, which implements most of the POSIX standard.
• The main detail that matters for OpenFOAM: an open-source CFD toolbox should rely on open-source technology and open standards.
• The detail that matters to a lot of users:
  • Can I use it on Windows or Mac OS X?
• What matters for making OpenFOAM work on most closed source OS':
  • How to adapt the POSIX standard that is followed in OpenFOAM, to the systems we need it working on.
• The result a few unofficial variants of OpenFOAM:
  • For Windows, where POSIX is not supported, which requires a considerable effort in adapting the source code, depending on the approach.
  • For Mac OS X, which requires some effort in adapting the source code, since Mac OS X adopts most of the POSIX standard.
• Among these efforts, blueCFD was created in 2009, to improve upon existing work of porting OpenFOAM for Windows.
• In November 2013, blueCFD was rebranded to blueCFD®-Core, as our product line expanded.
What is blueCFD®-Core? (1/3)

Features:
• A service package available upon request, of a high quality build of OpenFOAM® for up-to-date Windows 7, 8 and 8.1 64-bit, fully compilable on Windows.
• Complete functionality with the original scripts of OpenFOAM on Windows, by relying on MSys.
• All features in OpenFOAM 2.3 that require compiling, will build as intended in blueCFD-Core 2.3-1.
• Customized solvers and libraries can also be compiled directly with OpenFOAM 2.3 on Windows.
• Third-Party software is also provided, including: ParaView, Discretizer, enGrid, Blender, Gnuplot, GDB, Notepad2 and Python.
Features (cont...):
• A Portable functionality, that allows copying the installed blueCFD-Core into an USB drive and ready to be used in other Windows machines.
• A single User Guide that addresses all major features of blueCFD-Core.
• Provide the full source code of OpenFOAM, including the modifications done for making it work on Windows.

References:
• http://bluecfd.com/Core
• http://bluecfd.com/Core#Features
Objectives:

• Bring OpenFOAM technology to Windows, enabling all features available in GNU/Linux Distributions.
• Preserving full compatibility and functionality with the original source code, with the minimal impact to the source code.
• Quality assurance tests, in order to ensure and document which features are working in accordance with the official Linux distribution.
Installing blueCFD-Core (1/17)

In the provided DVD/USB should be the following file and folder:

![File and folder screenshot]

To start the installer, double click on the file “blueCFD-Core-2.3.1-win64-setup.exe”

Just to let you know, inside the folder “blueCFD-Core-downloads” is an optional package that provides the code documentation.
Installing blueCFD-Core (2/17)

Once the installer starts, it will show the following window:

Click on the “Next” button.
Installing blueCFD-Core (3/17)

The next window provides the license information and the request for agreement:

![License Agreement window](image)

After accepting the agreement, click on the “Next” button.
Installing blueCFD-Core (4/17)

In the next window, it asks where blueCFD-Core should be installed:

![Setup - blueCFD-Core](image)

Notes in the next slide...
Installing blueCFD-Core (5/17)

Notes on “Select Destination Location”:

- The standard location should work for most people, although keep in mind that the installer will activate the ability to write files within specific user sub-folders inside this folder.
- Alternatively, you can install in “C:\blueCFD-Core-2.3” or in a similar drive letter.
- Or if you prefer, you can install this only for your own personal area, by closing the installer and running it manually from the command line, like this:

  blueCFD-Core-2.3-1-win64-setup.exe /SINGLEUSER=1
Installing blueCFD-Core (6/17)

Once the location is chosen, click on the “Next” button and it will ask what type of installation to perform:

More details on the next slide...
The types of installation are essentially:

- “Typical installation” – To install everything, except for the code documentation.

- “Full installation” – To install everything, including the code documentation. The advantage here is that this makes it easier to find more details about certain features that OpenFOAM has got, such as Boundary Conditions and Function Objects, which are not documented in the User Guide.

- "Custom installation" – (next slide...)
"Custom installation" – Where you can choose which applications and features you want to install, as exemplified here:
Installing blueCFD-Core (9/17)

Once the choices have been made, click on the “Next” button, which will allow choosing the Start Menu group where the blueCFD-Core shortcuts should be placed:
Installing blueCFD-Core (10/17)

After choosing the group name, click on the “Next” button, which lead to the window with the following options:

Details in the next slide...
Notes regarding “Select Additional Tasks” (1/2):

• The desktop icon is useful specially on Windows 8 and 8.1, due to the non-existence of a Start Menu. Without this icon, it could get very complicated to use blueCFD-Core on those versions of Windows.

• The option to "Add Notepad2 to the right-click on any file in Windows Explorer" is useful for editing the OpenFOAM case files.
Notes regarding “Select Additional Tasks” (2/2):
• The option to "enable write permissions" is necessary and advisable when the user currently installing blueCFD-Core is able to perform administrative installations.
  • This is needed when installing in the default installation folder: “C:\Program Files”
  • If not enabled in such a situation, namely to give the ability to write in the main user folders “ofuser-2.3”, “msys\home\ofuser” and “msys\etc”, will disrupt the conventional installation process.
Once the choices have been made, click on the “Next” button. The final window before the installation begins is shown:

Click on the “Next” button to proceed.
Installing blueCFD-Core (14/17)

While it is installing blueCFD-Core, it should show the progress bar, as exemplified here:

The progress bar will go forward with the advancement of the installation process.
Installing blueCFD-Core (15/17)

When it reaches the end of the files to be installed, it will run the external installers and unpack the Code Documentation package, if selected. This will reset the progress bar for this second progress stage:
Note:
One of the possible steps in this second progress stage is to install the Microsoft 2010 Runtime, which will interactively ask you to follow its own installation steps. The steps should be fairly simple:
1. Accept the license.
2. Click on the OK button to install.
3. Wait a little while.

The control will then return to the blueCFD-Core installer.
Installing blueCFD-Core (17/17)

Once the installation is complete, it will show the following window:

Once you click in the “Finish” button, blueCFD-Core should be installed with the chosen features!
The Start Menu, as shown on the right, is the conventional way to access installed applications in Windows, at least until Windows 8 appeared. Click on "All Programs" and look for blueCFD-Core-2.3, which is shown below:
On Windows 8, blueCFD-Core provides a shortcut in your Windows Desktop (optional during installation) →

Double-click on it, which show a new Windows Explorer window a list similar to the following image:
Overview of installed packages (3/9)

There-in the following items can be found:

• **Browse blueCFD-Core folder** – This shortcut leads to the folder where blueCFD-Core was installed.

• **Install your blueCFD-Core in a Portable drive or folder** – This is a convenient application for copying the whole blueCFD-Core installation to a portable drive.

• **MSys terminal** – This is the primary command based interface with OpenFOAM, similar to how it works on Linux.

• **Windows Command Line** – This is the alternative command based interface with OpenFOAM, which relies only on the features that are common to MS-DOS.

“Powershell” scripts are not available in blueCFD-Core, because it would require rewriting all scripts that are available in OpenFOAM.
Within the **Documentation** folder:

- **blueCFD-Core User Guide** – Main document.
- **OpenFOAM User Guide** – One of the most important documents for learning how to use OpenFOAM.
- **OpenFOAM Programmers Guide** – Complementing document to the OpenFOAM User Guide, also developed by OpenCFD, but not fully maintained.
- **Local OpenFOAM 2.3 Code Documentation** – Only available when the Code Documentation is installed.
Within the **GUI** folder, all open-source software (1/5):

- **Notepad2** – Simple, yet very advanced, text editor. It is provided with blueCFD-Core, because it acts as a good replacement to the default Windows Notepad application, specially due to the limitations in Notepad to properly open OpenFOAM files.

- **ParaView** – Standard post-processing application used for processing the results generated with OpenFOAM.
Within the **GUI** folder (2/5):

- **Blender** – A well known 3D modelling application, although mostly oriented for the creation of computer animated movies. It's provided with blueCFD-Core also due to Blender's powerful add-on mechanism, for which it's also included:
  - enGrid's import and export scripts, for bridging geometry manipulation for enGrid (next slide).
  - SwiftBlock and SwiftSnap, which are interfaces for making it easier to configure OpenFOAM’s main meshers **blockMesh** and **snappyHexMesh**.

- [http://openfoamwiki.net/index.php/Contrib/SwiftBlock](http://openfoamwiki.net/index.php/Contrib/SwiftBlock)
- [http://openfoamwiki.net/index.php/Contrib/SwiftSnap](http://openfoamwiki.net/index.php/Contrib/SwiftSnap)
Within the **GUI** folder (3/5):

- **enGrid** — A mesh generation software with CFD applications in mind. It first imports a surface mesh, can generate/improve said surface mesh and only then will it generate the volume mesh. It uses Netgen for generating tetrahedral meshes, but can then perform an advanced export to a polyhedral mesh.

- **Discretizer** and **Discretizer Setup** — These two are part of the same software application, where:
  - **Discretizer** for manually drawing meshes.
  - **Discretizer Setup** is a pre-processor, i.e. helps prepare cases.
Overview of installed packages (8/9)

Within the **GUI** folder (4/5):

- **Gnuplot** – Advanced application for plotting data. It comes in two flavors:
  - **GUI**, which provides a mouse assisted approach to creating graphs.
  - **Shell**, which provides a strict command line approach to creating graphs.
- **Python Shell** – A command line interface for using the programming language Python. It's currently provided as a complement to ParaView's Python capabilities, but it can also be used with PyFoam, SciPy, etc...
Within the **GUI** folder (5/5):

- **Python IDLE** – A graphical user interface for Python, making it easier to write and test Python source code.
- **CMake (cmake-gui)** – A graphical user interface for building software that rely in CMake, which is a cross-platform build system. This is mostly useful for building some of the communities contributions for OpenFOAM.

Within the **Web** folder are all links to online websites which provide information for the software provided with blueCFD-Core. This is where most of the remaining documentation can be found.
Overview of installation directory (1/2)

In the main installation folder, the most important are:

• **DOS_Mode.bat** – Batch file that launches the Windows Command Line for using blueCFD-Core.

• **docs** – Has all of the documents for blueCFD-Core.

• **msys** – “Minimal System” which is similar to a terminal interface in a Linux Distribution.

• **ofuser-2.3** – Where your personal simulations cases and source code should be placed.

• **OpenFOAM-2.3** – Location for OpenFOAM’s source code, binaries, tutorials and documentation.

• **ThirdParty-2.3** – Where all third-party software is installed, e.g.: enGrid, Discretizer, ParaView...

• **shortcuts** – All shortcuts for a portable installation.
Important sub-folders:
• **msys\home\ofuser** – Where the Msys shell environment will start and where most personal files are stored.
• **ofuser-2.3\run** – Where your personal simulations cases should be placed.
• **OpenFOAM-2.3:**
  • **doc** – Location for OpenFOAM’s documentation.
  • **tutorials** – Location for OpenFOAM’s tutorial case folders.
Getting started with the interface (1/11)

The main interface available in blueCFD-Core is essentially the same that is available in OpenFOAM: the command line interface.

Interface on Linux

Interface on Windows
Commands for file management (1/4):

- `ls`  
  list directory contents

- `ls -l`  
  same as above, but in a single column

- `ls -al`  
  formatted listing with hidden files

- `ll`  
  formatted listing, same as `ls -l`

- `cd dirname`  
  go to directory `dirname`

- `cd`  
  go to user home

- `cd ..`  
  go back one directory

- `pwd`  
  show current directory path

- `mkdir dirname`  
  create directory `dirname`
Commands for file management (2/4):

```
rm filename          delete file  filename
rm -r dirname        delete directory  dirname
rm -f filename       force delete file  filename  (CAUTION)
rm -rf dirname       force delete directory  dirname  (CAUTION)
```

```
cp filename1 filename2
   copy file  filename1  to file  filename2

cp -r dirname1 dirname2
   copy directory  dirname1  to directory  dirname2
```
Commands for file management (3/4):

mv filename1 filename2
    rename or move file filename1 to file filename2

ln -s filename linkname
    create symbolic link linkname to file filename

touch filename
    create file filename or change times of file filename

less filename
    interactively output the contents of file filename
Commands for file management (4/4):

```
less filename  output the contents of file filename  
    "q" for ending the interactive mode

head filename  output the first 10 lines of file filename

tail filename  output the last 10 lines of file filename

tail -f filename  same as above, but updates continuously
```
Commands for system information:

<command>  --help
          if available, shows the available help for  <command>

date    show the current date and time
whoami  display the user name you are logged in as
uname -a show operating system kernel information
df      show disk usage
du      show directory space usage
Commands for process management:

- **ps**: display your currently active processes
- **kill pid**: kill process with identification `pid` (CAUTION)
- **jobs**: lists stopped or background jobs;
- **bg**: resume a background job
- **fg**: brings the most recent job to foreground
- **fg n**: brings job `n` to the foreground
Commands for searching files and content:

- `grep pattern filename`
  search for pattern in filename
- `grep -r pattern dirname`
  search recursively for pattern in dirname
- `command | grep pattern`
  search for pattern in the output of command
- `find dir -name pattern`
  search for pattern in a directory hierarchy
- `which command`
  locate a command
- `where command`
  show possible locations of command
Getting started with the interface (9/11)

Text editors:

`vi` *file*  
*use text editor* `vi` *to edit* *file*

`nano` *file*  
*use text editor* `notepad2` *to edit* *file*  
(because `nano` isn’t available for MSys and is easier to use than `vi`)

There are a lot of other text editors worth mentioning, which aren’t installed in blueCFD-Core, e.g.:

- `Notepad++`  
  `notepad-plus-plus.org`
Command Line Navigation:

- Ctrl+a  moves cursor to beginning of line
- Ctrl+e  moves cursor to end of line
- Ctrl+→ moves cursor to beginning of next word in the line
- Ctrl+← moves cursor to beginning of previous word in the line
- Ctrl+k  deletes words until end of line from current cursor position
- Ctrl+u  deletes words until the start of line from current cursor position
- Ctrl+y  *pastes* the words that were deleted with Ctrl+k/u
- Alt+backspace  deletes previous word in line from current cursor position
Getting started with the interface (11/11)

Additional information about shells, commands and procedures on Linux can be obtained through:
- The Linux Documentation Project: www.tldp.org
- Linux Command website: linuxcommand.org

In the Linux Documentation Project website, we can also see a general introduction on Linux:

Thank you for your time.

Next:
2 - Getting Started