OpenFVM-Flow

Three-Dimensional Unstructured Finite Volume Non-Isothermal Incompressible Two-phase Flow Solver

version 1.1

Reference Manual

OpenFVM team
September 7, 2008

GNU Free Documentation License

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The **"Document"**, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as **"you"**. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A **"Modified Version"** of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A **"Secondary Section"** is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The **"Invariant Sections"** are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The **"Cover Texts"** are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A **"Transparent"** copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called **"Opaque"**.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The **"Title Page"** means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section **"Entitled XYZ"** means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as **"Acknowledgements"**, **"Dedications"**, **"Endorsements"**, or **"History"**.) To **"Preserve the Title"** of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the

Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

Contents

1 Installation **17**

1.1 External Tools and Libraries . . . 17

2 Numerical Overview **18**

2.1 Finite Volume Method . . . 18

2.2 Interface Tracking Methods . . . 18

2.2.1 Volume Fractions . . . 19

2.3 Space Derivative Approximations . . . 19

2.3.1 Upwind Differencing Scheme . . . 19

2.3.2 Central Differencing Scheme . . . 19

2.4 Time Marching Methods . . . 20

2.4.1 Explicit Methods . . . 20

2.4.2 Implicit Methods . . . 20

2.5 Segregated Solver . . . 20

2.6 Velocity-Pressure Coupling . . . 21

2.6.1 SIMPLE . . . 21

2.6.2 SIMPLER . . . 21

2.6.3 SIMPLEC . . . 21

2.6.4 PISO . . . 21

2.7 Collocated Arrangement . . . 22

2.8 Incompressible Flow . . . 22

2.9 General Transport Equation . . . 24

2.10 Momentum . . . 25

2.11 Continuity . . . 30

2.12 Temperature . . . 33

2.13 Interface Capturing . . . 35

2.14 Non-Orthogonality Correction . . . 37

2.15 Solution of Linear Equation Systems . . . 38

2.16 Solution Algorithm . . . 39

3 Pre-Processing **42**

3.1 Geometry . . . 42

3.2 Mesh Type . . . 42

3.3 Mesh Reordering . . . 42

3.4 Parallel Processing . . . 44

3.5 Domain Decomposition 44
3.6 Units 44
3.7 Boundary Conditions 45
3.8 Material 46
3.9 Parameters 47
3.10 Results 48

4 Tutorial **49**
4.1 Non-Isothermal Lid-Driven Cavity Flow 49

List of Figures

1 Three-dimensional control volume 24
2 Control volume of arbitrary topology 25
3 Boundary 38
4 Numerical algorithm flowchart 41
5 Mesh type 43
6 Matrix structure before and after applying the RCM algorithm 43
7 Partition and ghost cells 45
8 Boundary conditions file format 45
9 Boundary conditions example 47
10 Material file format 47
11 Parameter file format 48
12 Geometry of the lid-driven tutorial flow 49
13 Convergence of the non-isothermal lid-driven cavity flow 55
14 Velocity vectors of the lid-driven cavity flow 56

List of Tables

1 Tools and libraries 17
2 Comparison of several iterative solvers 39
3 Units 44
4 Operators and functions 46
5 Boundary types 46

Nomenclature

Roman Symbols

A	Face area	m^2
C	Specific heat	J/(kgK)
d	Distance between adjacent cells	m
F	Velocity face flux	m/s
g	Gravity	$\mathrm{m/s}^2$
k	Thermal conductivity	J/(Kms)
n	Face normal	
p	Pressure	$\mathrm{kg/(ms}^2)$
S	Source term	
T	Temperature	K
t	Time	s
U	Velocity vector	m/s
u	Component x of velocity vector U	m/s
V	Cell volume	m^3
v	Component y of velocity vector U	m/s
w	Component z of velocity vector U	m/s
D	Rate of strain tensor	
I	Unit tensor	
T	Total stress tensor	$\mathrm{kg/(ms}^2)$

Greek Symbols

β	CICSAM weighting interpolation factor	
Γ	Diffusivity	
γ	Volume fraction	
λ	Interpolation factor	

λ	Relaxation factor	
μ	Dynamic viscosity	kg/(ms)
ϕ	Flow quantity	
ψ	Material property	
ρ	Density	$\mathrm{kg/m^3}$
τ	Viscous stress tensor	$\mathrm{kg/(ms^2)}$

Subscripts

ϕ_A	Cell-averaged value of ϕ at the acceptor cell A
ϕ_B	Value of ϕ at the boundary
ϕ_D	Cell-averaged value of ϕ at the donor cell D
ϕ_j	Face-averaged value of ϕ at face j
ϕ_N	Cell-averaged value of ϕ at the neighbor cell N
ϕ_P	Cell-averaged value of ϕ at cell P

Superscripts

ϕ^t	Value of ϕ at time t
Q^x	Component x of vector Q
Q^y	Component y of vector Q
Q^z	Component z of vector Q

Dimensionless Numbers

Co	Courant number

Abbreviations

CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CDS	Central Differencing Scheme
CFD	Computational Fluid Dynamics
CG	Conjugate Gradient

CICSAM	Compressive Interface Capturing Scheme for Arbitrary Meshes
CT	Convective Term
CV	Control Volume
DT	Diffusive Term
FVM	Finite Volume Method
GUI	Graphical User Interface
ILU	Incomplete Lower-Upper
PISO	Pressure Implicit with Splitting of Operators
RCM	Reverse Cuthill-McKee
SIMPLE	Semi-Implicit Pressure Linked Equations
SIMPLEC	Semi-Implicit Pressure Linked Equations Consistent
SIMPLER	Semi-Implicit Pressure Linked Equations Revised
SOR	Successive Over Relaxation
ST	Source Term
UDS	Upwind Differencing Scheme
UT	Unsteady Term

1 Installation

1.1 External Tools and Libraries

A list of tools and libraries used in OpenFVM is presented in Table 1.

Table 1: Tools and libraries

Name	Link
Gmsh	`http://www.geuz.org/gmsh`
Gnuplot	`http://www.gnuplot.info`
RCM	`http://www.math.temple.edu/~daffi/software/rcm`
LASPack	`http://www.mgnet.org/mgnet/Codes/laspack/html/laspack.html`
Metis	`http://glaros.dtc.umn.edu/gkhome/metis/metis/overview`
PETSc	`http://www-unix.mcs.anl.gov/petsc/petsc-as/index.html`

Install these external tools and libraries. After, edit the makefile(s) to set the apropriate paths and type:

```
make all
```

If successful, a new executable will be created in the examples directory.

2 Numerical Overview

The objective of this section is to describe the mathematical models used to develop a three-dimensional finite volume unstructured CFD code capable of describing non-isothermal two-phase fluid flow. The method described here is suitable to be used with unstructured meshes, which is the most flexible type of grid since it can fit any boundary shape. These type of meshes are used extensively with finite element applications but their use with finite volume methods has increased in recent years. The mesh may be hybrid in which the elements or control volumes may have arbitrary shapes, while structured meshes are usually restricted to quadrilateral or hexahedral elements. Unstructured meshing usually requires less user intervention for complex geometries, specially when using integrated CAD/CAE applications . The geometry can also be modelled and exported to standard formats. Although these meshes are very flexible, they present the inconvenience of an irregular data structure which will be reflected in the structure of the matrices. The solution of sparse matrices usually requires more computational effort. However, these disadvantages are becoming less important as computational speed increases and sparse matrix solvers become more efficient. Another disadvantage compared with structured grids is the additional memory required to store the connectivity of the mesh.

2.1 Finite Volume Method

The Finite Volume Method (FVM) is quite popular in Computational Fluid Dynamics (CFD) due to mainly two reasons: they ensure that the discretisation is conservative and does not require a coordinate transformation when applied to irregular meshes. As a result, it can be easily adapted to unstructured meshes consisting of arbitrary polyhedra. Using the finite volume formulation, the integral form of the conservation laws are satisfied to some degree of approximation for all control volumes (CV) of the computational domain.

2.2 Interface Tracking Methods

Computation of free surface and fluid interfaces can be classified mainly into two groups: surface methods and volume methods. In the first type of methods the interface is tracked explicitly with marker points or by moving it with the mesh. Therefore, the position of the interface is known throughout the calculation and remains sharp as it is convected across the mesh. In volume methods, cells are marked with massless particles or an indicator fraction. This means that the exact position of the interface is not known explicitly and special techniques need to be applied to capture a well defined interface.

2.2.1 Volume Fractions

In this method a scalar function between zero and one is used to distinguish between the two different fluids. A cell value of zero or one indicates the presence of only one fluid in that cell. If the value is zero, only fluid 0 exists and if the value of is one, only fluid 1 is present. A volume fraction value between these two limits indicates two fluids exist in a given cell and therefore an interface is present. The value itself gives an indication of the relative proportions of each fluid occupying the control volume. In this method, only a scalar convective equation needs to be solved to propagate the indicator function throughout the domain. This means that it is easily implemented with finite volume methods and is efficient in terms of memory and computation time. However, this is also one of the drawbacks of this method, since it is difficult to obtain a bounded volume fraction between physical limits of zero and one and maintain a sharp interface. To assure a well defined interface, several techniques have been employed within the volume fraction framework such as: line techniques, donor-acceptor formulations and higher order differencing schemes.

2.3 Space Derivative Approximations

Solution of the conservation laws gives cell-averaged values of the dependent variable ϕ. However, the approximation to the integrals requires values at locations other than computational nodes. In order evaluate the convective and diffusive fluxes, the value ϕ and its gradient normal to the cell face at one or more locations of the Control Volume (CV) surface are needed. Therefore, they have to be calculated from the cell center values using a method of interpolation. Each interpolation method introduces a degree of error in the formulation.

2.3.1 Upwind Differencing Scheme

The Upwind Differencing Scheme (UDS) approximates the value at the face center using the value at the center of the upstream cell. This scheme satisfies the boundedness criteria, i.e. it does not give oscillatory solutions. However, this is achieved by introducing numerical diffusion. This scheme is first order accurate.

2.3.2 Central Differencing Scheme

The central differencing scheme (CDS) approximates the value at face center using linear interpolation between the value at the center of the two adjacent cells. This scheme is second order accurate. The assumption of a linear profile between adjacent cells also offers the simplest approximation of the gradient which is needed for the evaluation of the diffusive flux.

2.4 Time Marching Methods

2.4.1 Explicit Methods

The implicit Euler method is the simplest time marching method in which all fluxes and sources are evaluated using the previous time step values. In the equation for a control volume, the only unknown is the value at the center of the cell, while neighbor values are all evaluated at earlier time levels. It is first order accurate in time. When there is no diffusion this method is unstable (unconditionally unstable) and when there is no convection it is stable provided that the time step is sufficiently small (conditionally stable).

2.4.2 Implicit Methods

Implicit methods require the solution of a system of equations since the fluxes and neighbor values are evaluated in terms of the unknown variable values at the new time step. Therefore implicit methods require more memory since the coefficient matrix needs to be stored. The use of implicit Euler method allows large time steps to be taken, however, problems may arise when CDS is used on coarse grids and the time step may also be limited due to subsequent interface tracking methods. This method is first order accurate in time and is specially useful for solving steady flow problems. For steady state problems the final solution does not depend on the transition between the initial guess and the final stage. To reach steady state a pseudo-time marching or under-relaxation scheme may be employed. The difference between them is that the use of a under-relaxation factor is equivalent to applying a different time step to each control volume.

The Crank.Nicolson method is sometimes refered as the trapezoid rule method or trapezoidal method. This method is second order accurate and it is particularly useful when time accuracy is of importance. Although this scheme is more accurate than implicit Euler, it requires almost the same computational effort. This method is quite popular since it is unconditionally stable, but oscillatory solutions are possible when using large time steps.

2.5 Segregated Solver

The segregated method solves equations of momentum, continuity and temperature are solved sequentially, while the coupled approach solves them simoultaneously, requiring usually more computational memory. The coupled solver is usually used when the velocity and pressure are strongly coupled such as in transonic and supersonic flows. The fully coupled approach solves the equations based on the density, being pressure calculated using an equation of state.

2.6 Velocity-Pressure Coupling

Considering segregated methods, several algorithms have been developed to couple velocity and pressure. However, algorithms such as SIMPLE, SIMPLER, SIMPLEC and PISO have a simliar behaviour. These are predictor-corrector procedures. The momentum equations are solved for a guessed pressure field normally using the value of the previous time step. Solving the set of algebraic equations, a new velocity field is determined which does not satisfy the continuity. During the calculation of the momentum, a velocity field without the pressure contribution is determined and considered during the calculation of pressure. Considering the continuity equation for incompressible flow, sum of the fluxes in and out of the control volume should be zero. These fluxes are related with a face pressure gradient using the interpolation purposed by Rhie and Chow. The continuity equation is transformed into a Poisson's equation for pressure. The solution of the resultant elliptic equation yields a new pressure field, which is then used as the initial guess of the subsequent iteration in time.

2.6.1 SIMPLE

The algorithm was first developed by Patankar and Spalding (1972). The main disadvantage of this method is that it requires several iterations to converge and, therefore, is slower than other methods that were derived from it. Also it is more suitable for steady state calculations.

2.6.2 SIMPLER

SIMPLER of Patanakar (1980) is a revised version of the SIMPLE method. This method uses the continuity equation to derive a discretized equation for pressure.

2.6.3 SIMPLEC

This algorithm was developed by Van Doormal and Raithby (1984) and is similiar to SIMPLE, except the momentum equations are manipulated so that the velocity correction equations omit terms that are less significant.

2.6.4 PISO

PISO is a non-iterative technique to solve implicitly discretized flow equations [9]. This is a segregated aproach to solve the Navier-Stokes equations in which pressure and velocity are solved sequentially. Its main advantages are its avoidance of iterations during each time step and can be applied to steady state simulations without any modification. In order to reach steady state, the method adances in time until steady state is reached. Relatively to SIMPLE-like algorithms it is more complex.

Using sufficiently small time steps it yields accurate results and can be used to simulate both incompressible and compressible flow.

2.7 Collocated Arrangement

The dependant variables are computed at the center of the CVs. This type of arrangement is easier to implement on unstructured meshes and provides minimized storage requirements. However, compared with the staggered arrangement, it presents difficulties with pressure-velocity coupling and the appearance of pressure oscillations.

2.8 Incompressible Flow

The flow is mathematically described by the conservation laws, namely the conservation of mass and conservation of momentum. The following equation states that mass is neither created nor destroyed and an increase of mass is only possible if the fluid is compressible:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U) = 0, \tag{1}$$

where ρ is the density, t stands for time and U is the velocity vector. By providing the appropriate expression for the divergence operator, this form can be transformed into a form specific to a given coordinate system. In this work, the equations are descritised considering the Cartesian coordinate system. For incompressible flows, the continuity equation reduces to:

$$\nabla \cdot U = 0. \tag{2}$$

In the case of the conservation of momentum, the conserved property is the velocity:

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot (\rho U U) - \nabla \cdot \mathrm{T} = \rho g, \tag{3}$$

where g is the gravity. The total stress tensor T is defined as:

$$\mathrm{T} = -p\mathrm{I} + \tau, \tag{4}$$

being p the pressure and I the unit tensor. For a Newtonian fluid, the viscous stress tensor τ is given by:

$$\tau = 2\mu \mathrm{D} - \frac{2}{3}(\mu \nabla \cdot U) I, \tag{5}$$

where μ is the dynamic viscosity. For non-Newtonian fluids, the relation between the viscous stress tensor and the velocity is defined by a set of partial differential equations. Due to the incompressibility of the flow (Equation 2), the second term of Equation 5 is zero, so the viscous stress tensor becomes:

$$\tau = 2\mu \mathrm{D}. \tag{6}$$

The rate of strain tensor D is calculated using the following expression:

$$\mathrm{D} = \frac{1}{2}(\nabla U + (\nabla U)^T). \tag{7}$$

Considering the above simplifications, the momentum equation (Equation 3) can be expressed as:

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot (\rho U U) - \nabla \cdot \tau = -\nabla p + \rho g. \tag{8}$$

The viscous stress term can be reformulated [4]:

$$\begin{aligned} \nabla \cdot \tau &= \nabla \cdot (\mu(\nabla U + (\nabla U)^T)) \\ &= \nabla \cdot (\mu \nabla U + \nabla \cdot (\mu(\nabla U)^T)) \\ &= \nabla \cdot (\mu \nabla U + \nabla U \cdot \nabla \mu + \mu \nabla(\nabla \cdot U)). \end{aligned} \tag{9}$$

Since the flow is incompressible (Equation 2):

$$\nabla \cdot \tau = \nabla \cdot (\mu \nabla U) + \nabla U \cdot \nabla \mu \tag{10}$$

and substituting Equation 10 into Equation 8 gives:

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot (\rho U U) - \nabla \cdot (\mu \nabla U) = \nabla U \cdot \nabla \mu - \nabla p + \rho g. \tag{11}$$

For viscoelastic flow, the UCM consitutive equation may be used to estimate the viscous stress tensor:

$$\tau + \lambda \left[\frac{\partial \tau}{\partial t} + \tau(\nabla \cdot U) - \tau(\nabla U - (\nabla U)^T) \right] = 2\mu D = \mu(\nabla U + (\nabla U)^T) \tag{12}$$

where λ is the relaxation factor.

2.9 General Transport Equation

The above equations can be rearranged to a general transport equation with appropriate choice of ϕ, Γ and S_ϕ. The flow quantity ϕ can be a scalar, vector or tensor field:

$$\underbrace{\frac{\partial \rho\phi}{\partial t}}_{\text{unsteady term}} + \underbrace{\nabla \cdot (\rho\phi U)}_{\text{convective term}} - \underbrace{\nabla \cdot (\Gamma \nabla \phi)}_{\text{diffusive term}} = \underbrace{S_\phi}_{\text{source term}} , \tag{13}$$

where Γ is the diffusivity and S_ϕ is the source term. Applying the finite volume method, it is possible to integrate Equation 13 over a three-dimensional control volume presented in Figure 1:

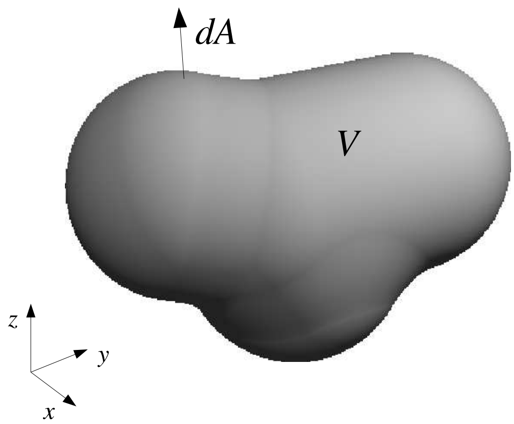


Figure 1: Three-dimensional control volume

$$\int \frac{\partial \rho\phi}{\partial t} dV + \int \nabla \cdot (\rho\phi U) dV - \int \nabla \cdot (\Gamma \nabla \phi) dV = \int S_\phi dV, \tag{14}$$

where V is the volume. Using the Gauss divergence theorem, the volume integrals are transformed into surface integrals [8]:

$$\int \nabla \cdot U dV = \int n \cdot U dA, \tag{15}$$

where A is the area. The integral form of the general transport equation is given by [1]:

$$\frac{\partial}{\partial t} \int \rho\phi dV + \int \rho\phi U \cdot n dA - \int \Gamma \nabla \phi \cdot n dA = \int S_\phi dV. \tag{16}$$

The solution domain is divided into a finite number of non-overlapping control volumes. Each CV has a center node P and one or more neighbors with center nodes N_f where f is the face index. It is considered that the volume does not vary with time. The colocated arrangement was chosen for this formulation. The average values are calculated at cell centers and interpolated to the faces using a linear interpolation scheme. A schematic overview of the control volume is presented in Figure 2.

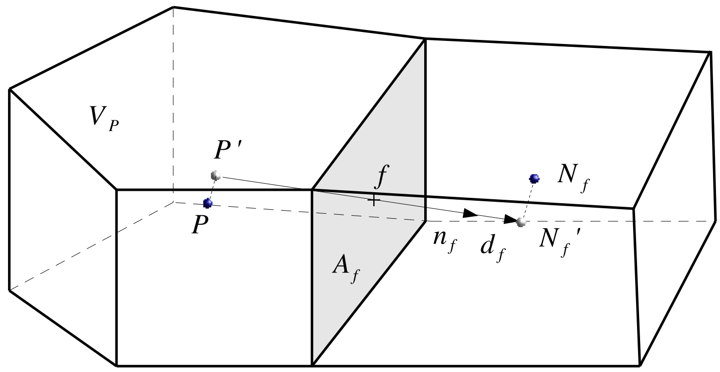


Figure 2: Control volume of arbitrary topology

In order to calculate the surface and volume integrals, the midpoint approximation rule is adopted [2]:

$$\left(\frac{\partial \rho\phi}{\partial t}\right)_P V_P + \sum_{f=1}^{n} \rho_f \phi_f u_f A_f - \sum_{f=1}^{n} \Gamma_f \nabla\phi_f A_f = (S_\phi)_P V_P, \tag{17}$$

where V_P is the volume of control volume P, A_f is the area vector of face f, n is the number of faces, Γ_f is the diffusivity evaluated at the face f and u_f is the velocity flux at face f.

2.10 Momentum

Considering $\phi = U$, $\Gamma = \mu$ and $S_\phi = -\nabla p + \rho g + (\nabla U) \cdot \nabla\mu$, the generic transport equation is transformed into the momentum equation:

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot (\rho U U) - \nabla \cdot (\mu \nabla U) = (\nabla U) \cdot \nabla\mu - \nabla p + \rho g. \tag{18}$$

The velocity vector at cell center U is formed by three components with values u, v, w in x, y, z direction respectively. Therefore, replacing the dependent variable with u, v, w and the source term with the contribution of pressure, gravity and surface forces gives three momentum equations.

Momentum in x direction:

$$\left(\frac{\Delta\rho u}{\Delta t}\right)_P + \frac{1}{V_P}\sum_{f=1}^{n}\rho_f u_f u_f A_f - \frac{1}{V_P}\sum_{f=1}^{n}\mu_f \nabla u_f A_f = (\nabla u)_P \cdot \nabla\mu_P - (\nabla p^x)_P + \rho_P g^x. \tag{19}$$

Momentum in y direction:

$$\left(\frac{\Delta\rho v}{\Delta t}\right)_P + \frac{1}{V_P}\sum_{f=1}^{n}\rho_f v_f u_f A_f - \frac{1}{V_P}\sum_{f=1}^{n}\mu_f \nabla v_f A_f = (\nabla v_P) \cdot \nabla\mu_P - (\nabla p^y)_P + \rho_P g^y. \tag{20}$$

Momentum in z direction:

$$\left(\frac{\Delta\rho w}{\Delta t}\right)_P + \frac{1}{V_P}\sum_{f=1}^{n}\rho_f w_f u_f A_f - \frac{1}{V_P}\sum_{f=1}^{n}\mu_f \nabla w_f A_f = (\nabla w)_P \cdot \nabla\mu_P - (\nabla p^z)_P + \rho_P g^z. \tag{21}$$

The gradient of the pressure at the center of the control volume is calculated using Gauss integration [3]:

$$(\nabla p^x)_P = \frac{1}{V_P}\sum_{f=1}^{n} p_f A_f n_f^x, \tag{22}$$

$$(\nabla p^y)_P = \frac{1}{V_P}\sum_{f=1}^{n} p_f A_f n_f^y, \tag{23}$$

and

$$(\nabla p^z)_P = \frac{1}{V_P}\sum_{f=1}^{n} p_f A_f n_f^z, \tag{24}$$

where n_f^x is x component of the face normal n , p_f is the pressure at face center, which is evaluated using linear interpolation. In the same manner, the gradient of the velocity is given by:

$$(\nabla u)_P = \frac{1}{V_P}\sum_{f=1}^{n} u_f A_f, \tag{25}$$

$$(\nabla v)_P = \frac{1}{V_P}\sum_{f=1}^{n} v_f A_f, \tag{26}$$

and

$$(\nabla w)_P = \frac{1}{V_P} \sum_{f=1}^{n} w_f A_f. \tag{27}$$

The central differencing scheme is used to evaluate a scalar at cell face center, which is obtained by linear interpolation:

$$\phi_f = \phi_{N_f} \lambda + \phi_P (1 - \lambda). \tag{28}$$

This scheme is second order accurate in space. The interpolation factor λ is given by:

$$\lambda = \frac{\partial_{Pf}}{\partial_{Pf} + \partial_{Nf}} \tag{29}$$

The first-order upwind differencing scheme can be used for the convective term to increase the stability of the calculation but it achieves this by introducing numerical diffusion. The value at the face center is obtained according to flux direction:

$$u_f < 0 \Rightarrow \phi_f = \phi_{N_f}, \tag{30}$$

$$u_f \geq 0 \Rightarrow \phi_f = \phi_P. \tag{31}$$

The face normal gradient of the pressure and velocity in each direction is calculated using the following expressions [5]:

$$\nabla p_f = \frac{p_{N_f} - p_P}{|d_f|}, \tag{32}$$

$$\nabla u_f = \frac{u_{N_f} - u_P}{|d_f|}, \tag{33}$$

$$\nabla v_f = \frac{v_{N_f} - v_P}{|d_f|}, \tag{34}$$

$$\nabla w_f = \frac{w_{N_f} - w_P}{|d_f|}, \tag{35}$$

where d_f is the distance from center of control volume P to center of control volume N. This approximation is second order accurate when the vector d_f is orthogonal to the face plane.

From here on, only the momentum for x direction is described. Central differencing scheme is used to evaluate values at cell faces. The temporal discretization is first order accurate in time using implicit Euler. Each term of the momentum equation will be treated independently in order to simplify the discretisation:

$$\mathrm{UT} + \mathrm{CT} - \mathrm{DT} = \mathrm{ST}. \tag{36}$$

Unsteady term (UT):

$$\mathrm{UT} = \frac{\rho_P (u_P^t - u_P^{t-1})}{\Delta t}, \tag{37}$$

$$\mathrm{UT} = \frac{\rho_P u_P^t - \rho_P u_P^{t-1}}{\Delta t}. \tag{38}$$

Convective term (CT):

$$\mathrm{CT} = \frac{1}{V_P} \sum_{f=1}^{n} \rho_f u_f u_f A_f, \tag{39}$$

$$\mathrm{CT} = \frac{1}{V_P} \sum_{f=1}^{n} \rho_f u_f A_f \left[u_{N_f} \lambda + u_P (1 - \lambda) \right], \tag{40}$$

$$\mathrm{CT} = \frac{1}{V_P} \sum_{f=1}^{n} (1 - \lambda) \rho_f u_f A_f u_P + \frac{1}{V_P} \sum_{f=1}^{n} \left(\lambda \rho_f u_f A_f \right) u_{N_f}. \tag{41}$$

Diffusive term (DT):

$$\mathrm{DT} = \frac{1}{V_P} \sum_{f=1}^{n} \mu_f \nabla u_f A_f, \tag{42}$$

$$\mathrm{DT} = \frac{1}{V_P} \sum_{f=1}^{n} \mu_f \frac{u_{N_f} - u_P}{|d_f|} A_f, \tag{43}$$

$$\mathrm{DT} = \frac{1}{V_P} \sum_{f=1}^{n} \mu_f \frac{u_{N_f} - u_P}{|d_f|} A_f. \tag{44}$$

Source term (ST):

$$\mathrm{ST} = -(\nabla P^x)_P - \rho_P g^x + (\nabla u)_P \cdot \nabla \mu_P, \tag{45}$$

$$\mathrm{ST} = -\frac{1}{V_P} \sum_{f=1}^{n} p_f A_f n_f^x - \rho_P g^x + (\nabla u)_P \cdot \nabla \mu_P. \tag{46}$$

After rearranging all the terms, the final algebraic equation which links the value of the velocity at the each cell center with the neighboring velocities can be written in matrix form:

$$a_P^m u_P^t + \sum_{f=1}^{n} a_{N_f}^m u_{N_f}^t = b_{u_P}^m, \tag{47}$$

where a_P^m is the central coefficient of the momentum matrix, $a_{N_f}^m$ are the off-diagonal coefficients and $b_{u_P}^m$ is the vector source term.

$$a_P^m = \frac{\rho_P}{\Delta t} + \frac{1}{V_P}\sum_{f=1}^{n}\left((1-\lambda)\rho_f u_f A_f + \mu_f \frac{A_f}{|d_f|}\right), \tag{48}$$

$$a_{N_f}^m = \frac{1}{V_P}\left(\lambda \rho_f u_f A_f - \mu_f \frac{A_f}{|d_f|}\right), \tag{49}$$

$$b_{u_P}^m = \frac{u_P^{t-1}\rho_P}{\Delta t} - \frac{1}{V_P}\sum_{f=1}^{n} P_f A_f n_f^x - \rho g^x + (\nabla u^{t-1})_P \cdot \nabla \mu_P^{t-1}. \tag{50}$$

The momentum equations are solved for a guessed pressure field normally using the value of the previous time step. Solving the set of algebraic equations, a new velocity field U^* is determined which does not satisfy the continuity equation. These predicted velocities are used to assemble vectors H_{u_P}, H_{v_P} and H_{w_P} which are used in the continuity equation. Therefore the continuity equation is transformed into a pressure equation. The solution of the continuity equation yields a new pressure field p^*. If the mesh in non-orthogonal several iterations may be necessary to obtain a more accurate pressure field. The new pressure field gives a conservative set of volumetric fluxes and is used to correct the velocity field. The discretised momentum equation isolating pressure contribution is presented for all three velocity components.

In the x direction, component u of the velocity vector:

$$a_P^m u_P^t = -\sum_{f=1}^{n} a_{N_f}^m u_N^t + S_{u_P} - (\nabla p^x)_P, \tag{51}$$

$$b_{u_P}^m = S_{u_P}^m - (\nabla p^x)_P, \tag{52}$$

$$u_P^t = \frac{H_{u_P}}{a_P^m} - \frac{1}{a_P^m}(\nabla p^x)_P, \tag{53}$$

$$H_{u_P} = -\sum_{f=1}^{n} a_{N_f}^m u_{N_f}^t + S_{u_P}. \tag{54}$$

In the y direction, component v of the velocity vector:

$$a_P^m v_P^t = -\sum_{f=1}^{n} a_{N_f}^m v_N^t + S_{v_P} - (\nabla p^y)_P, \tag{55}$$

$$b_{v_P}^m = S_{v_P}^m - (\nabla p^y)_P, \tag{56}$$

$$v_P^t = \frac{H_{v_P}}{a_P^m} - \frac{1}{a_P^m}(\nabla p^y)_P, \tag{57}$$

$$H_{v_P} = -\sum_{f=1}^{n} a_{N_f}^m v_{N_f}^t + S_{v_P}. \tag{58}$$

In the z direction, component w of the velocity vector:

$$a_P^m w_P^t = -\sum_{f=1}^{n} a_{N_f}^m w_N^t + S_{u_P} - (\nabla p^z)_P, \tag{59}$$

$$b_{w_P}^m = S_{w_P}^m - (\nabla p^z)_P, \tag{60}$$

$$w_P^t = \frac{H_{w_P}}{a_P^m} - \frac{1}{a_P^m}(\nabla p^z)_P, \tag{61}$$

$$H_{w_P} = -\sum_{f=1}^{n} a_{N_f}^m w_{N_f}^t + S_{w_P}. \tag{62}$$

2.11 Continuity

Considering $\phi = 1$, $\Gamma = 0$ and $S_\phi = 0$, the generic transport equation is transformed into the continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U) = 0, \tag{63}$$

$$\frac{\partial \rho}{\partial t} + U \cdot \nabla \rho + \rho \cdot \nabla U = 0. \tag{64}$$

Considering incompressible flow:

$$\sum_{f=1}^{n} u_f A_f = 0. \tag{65}$$

The values of H are interpolated to the face using linear interpolation:

$$H_{u_f} = H_{u_N}\lambda + H_{u_P}(1-\lambda), \tag{66}$$

$$H_{v_f} = H_{v_N}\lambda + H_{v_P}(1-\lambda), \tag{67}$$

$$H_{w_f} = H_{w_N}\lambda + H_{w_P}(1-\lambda), \tag{68}$$

$$a_f^m = a_{N_f}^m\lambda + a_P^m(1-\lambda). \tag{69}$$

The velocity face flux $\overline{u_f}$ is obtained from the interpolated values of H (Rhie-Chow interpolation). This is the velocity from the momentum equation without the effect of pressure:

$$\overline{u}_f = \frac{H_{u_f}}{a_f^m}, \overline{v}_f = \frac{H_{v_f}}{a_f^m}, \overline{w}_f = \frac{H_{w_f}}{a_f^m}, \tag{70}$$

$$\overline{u_f} = \overline{u}_f n_f^x + \overline{v}_f n_f^y + \overline{w}_f n_f^z. \tag{71}$$

The face flux velocity is corrected using the interpolated velocity obtained by the solution of the momentum equation.

$$u_f = \overline{u_f} - \frac{1}{a_f^m}(\nabla p)_f. \tag{72}$$

Substituting Equation 72 into Equation 65 gives:

$$\sum_{f=1}^{n} \frac{1}{a_f^m}(\nabla p)_f A_f = \sum_{f=1}^{n} \overline{u_f} A_f, \tag{73}$$

$$\sum_{f=1}^{n} \frac{1}{a_f^m}\frac{p_{N_f} - p_P}{|d_f|} A_f = \sum_{f=1}^{n} \overline{u_f} A_f. \tag{74}$$

The equations are rearranged into a matrix form which will is solved to determine a new pressure field which respects continuity:

$$a_P^c p_P + \sum_{f=1}^{n} a_{N_f}^c p_{N_f} = b_P^c, \tag{75}$$

$$a_P^c = -\sum_{f=1}^{n} \frac{A_f}{a_f^m |d_f|}, \tag{76}$$

$$a^c_{N_f} = \frac{A_f}{a^m_f |d_f|}, \tag{77}$$

$$b^c_P = \sum_{f=1}^{n} \overline{u_f} A_f. \tag{78}$$

The face flux velocity is then explicitly corrected using new pressure field:

$$u_f = \overline{u_f} - \frac{1}{a^m_f} \frac{p_{N_f} - p_P}{|d_f|}. \tag{79}$$

Boundary conditions

This section presents the treatment of the above equations near each boundary type. Three boundary types are considered: inlet, outlet and wall. The inlet and wall boundary types have fixed velocity and zero pressure gradient. Therefore the velocity face flux is calculated from the velocity at the boundary. In this case, no-slip conditions are assumed:

$$\overline{u_f} = u_B n^x_f + v_B n^y_f + w_B n^z_f, \tag{80}$$

where u_B is the x component of the velocity at the boundary. For the pressure gradient to be zero, the pressure at the boundary should be equal to the pressure of the adjacent cell:

$$p_B = p_P. \tag{81}$$

At the outlet the value of the velocity is extrapolated from the flow. In this case, the pressure at the boundary is specified while the velocity gradient is set to zero:

$$u_f = \overline{u_f} - \frac{1}{a^m_f} \frac{p_{N_f} - p_P}{|d_f|}. \tag{82}$$

The diffusive term (DT) becomes:

$$\frac{1}{V_P} \mu_f \nabla u_f \cdot A_f = \underbrace{\frac{1}{V_P} \frac{\mu_f A_f}{|d_n|} u_B}_{1} - \underbrace{\frac{1}{V_P} \frac{\mu_f A_f}{|d_n|}}_{2} u_P. \tag{83}$$

Since term 1 is known, it is added to the source term, while term 2 is added to the central coefficient.

The convective term (CT) can be written as:

$$\frac{1}{V_P} \rho_f u_f A_f u_B. \tag{84}$$

Since this term is known it is added to the source term (the face flux was calculated during the previous iteration).

2.12 Temperature

Considering $\phi = CT$, $\Gamma = k$ and $S_\phi = \Phi$, the generic transport equation is transformed into the temperature transport equation. The only temperature source considered is viscous thermal dissipation:

$$\frac{\partial \rho CT}{\partial t} + \nabla \cdot (\rho CTU) - \nabla \cdot (k\nabla CT) = S_{CT}, \tag{85}$$

being T the temperature, k the thermal conductivity and C the specific heat. Considering constant density and specific heat:

$$\rho C\frac{\partial T}{\partial t} + \rho C\nabla \cdot (TU) - \nabla \cdot (k\nabla CT) = p\nabla \cdot U + \Phi. \tag{86}$$

Using this mathematical transformation:

$$\nabla \cdot (TU) = U \cdot \nabla T + T\nabla \cdot U, \tag{87}$$

and considering the flow is incompressible:

$$\nabla \cdot (\phi U) = U \cdot \nabla \phi, \tag{88}$$

Equation 86 becomes:

$$\rho C\frac{\partial T}{\partial t} + \rho CT\nabla \cdot U + \rho CU \cdot \nabla T - \nabla \cdot (k\nabla T) = p\nabla \cdot U + \Phi. \tag{89}$$

Rearranging and adding source terms gives [7]:

$$\rho_P C_p \left[\frac{\partial T}{\partial t} + U \cdot \nabla T\right] - \nabla \cdot (k\nabla T) = \Phi \tag{90}$$

Unsteady term (UT):

$$\mathrm{UT} + \mathrm{CT} - \mathrm{DT} = \mathrm{ST}, \tag{91}$$

$$\mathrm{UT} = \rho_P C_p \frac{T_P^t - T_P^{t-1}}{\Delta t}. \tag{92}$$

Convective term (CT):

$$\mathrm{CT} = u_P \frac{1}{V_P} \rho_P C_P \sum_{f=1}^{n} T_f^t A_f n_f^x + v_P \frac{1}{V_P} \rho_P C_P \sum_{f=1}^{n} T_f^t A_f n_f^y + w_P \frac{1}{V_P} \rho_P C_P \sum_{f=1}^{n} T_f^t A_f n_f^z, \tag{93}$$

$$\begin{aligned} \mathrm{CT} = u_P \frac{1}{V_P} \rho_P C_P \sum_{f=1}^{n} [T_{N_f}^t \lambda + T_P^t (1-\lambda)] A_f n_f^x \\ + v_P \frac{1}{V_P} \rho_P C_P \sum_{f=1}^{n} [T_{N_f}^t \lambda + T_P^t (1-\lambda)] A_f n_f^y \\ + w_P \frac{1}{V_P} \rho_P C_P \sum_{f=1}^{n} [T_{N_f}^t \lambda + T_P^t (1-\lambda)] A_f n_f^z \end{aligned} \quad . \tag{94}$$

Diffusive term (DT):

$$\mathrm{DT} = \frac{1}{V_P} \sum_{f=1}^{n} k_f \nabla T_f A_f, \tag{95}$$

$$\mathrm{DT} = \frac{1}{V_P} \sum_{f=1}^{n} k_f \frac{T_{N_f}^t - T_P^t}{|d_f|} A_f. \tag{96}$$

Source term (ST):

$$\mathrm{ST} = \Phi. \tag{97}$$

Adding all terms gives:

$$a_P^e T_P^t + \sum_{f=1}^{n} a_{N_f}^e T_{N_f}^t = b_{T_P}^e, \tag{98}$$

$$\begin{aligned} a_P^e = \frac{\rho_P C_P}{\Delta t} + u_P \frac{1}{V_P} \rho_P C_P \sum_{f=1}^{n} (1-\lambda) A_f n_f^x + v_P \frac{1}{V_P} \rho_P C_P \sum_{f=1}^{n} (1-\lambda) A_f n_f^y \\ + w_P \frac{1}{V_P} \rho_P C_P \sum_{f=1}^{n} (1-\lambda) A_f n_f^z + k \frac{A_f}{|d_f|} \end{aligned} \quad , \tag{99}$$

$$\begin{aligned} a_{N_f}^e = u_P \frac{1}{V_P} \rho_P C_P \lambda A_f n_f^x + v_P \frac{1}{V_P} \rho_P C_P \lambda A_f n_f^y \\ + w_P \frac{1}{V_P} \rho_P C_P \lambda A_f n_f^z - k_f \frac{A_f}{|d_f|} \end{aligned} \quad , \tag{100}$$

$$b_{T_P}^e = \frac{\rho_P C_P}{\Delta t} T_P^{t-1} + \Phi. \tag{101}$$

The viscous dissipation is calculated as:

$$\Phi = \mu \left\{ \Phi_1 + \Phi_2 + \Phi_3 \right\}, \tag{102}$$

$$\Phi_1 = 2[(\nabla u^x)^2 + (\nabla v^y)^2 + (\nabla w^z)^2], \tag{103}$$

$$\Phi_2 = (\nabla v^x + \nabla u^y)^2 + (\nabla w^y + \nabla v^z)^2 + (\nabla u^y + \nabla w^x)^2, \tag{104}$$

$$\Phi_3 = -\frac{2}{3}(\nabla u^x + \nabla v^y + \nabla w^z)^2. \tag{105}$$

Boundary conditions

If the wall is adiabatic no heat transfer occurs between the boundary and the adjacent cell. In other cases, the temperature of the boundary should be considered in the diffusive and convective terms. The diffusive term (DT) becomes:

$$\frac{1}{V_P} k_f \nabla T_f \cdot A_f = \underbrace{\frac{1}{V_P} \frac{k_f A_f}{|d_n|} T_B}_{1} - \underbrace{\frac{1}{V_P} \frac{k_f A_f}{|d_n|}}_{2} T_P. \tag{106}$$

Since term 1 is known, it is added to the source term, while term 2 is added to the central coefficient. The convective term (CT) becomes:

$$\mathrm{CT} = u_P \frac{1}{V_P} \rho_P C_P T_B^t A_f n_f^x + v_P \frac{1}{V_P} \rho_P C_P T_B^t A_f n_f^y + w_P \frac{1}{V_P} \rho_P C_P T_B^t A_f n_f^z. \tag{107}$$

Since this term is known it is added to the source term.

2.13 Interface Capturing

The VOF method can be used to track the interface of two immiscible fluids. Considering $\rho = \gamma$, $\phi = 1$, $\Gamma = 0$ and $S_\phi = 0$, the generic transport equation is transformed into the volume-of-fluid equation:

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot (\gamma U) = 0, \tag{108}$$

where γ is the volume fraction. The Crank-Nicolson differencing scheme is used for temporal discretization. There are several interface capturing schemes, however most of them are limited to structured cell arrangements. The scheme develop by Ubbink (1997) named CICSAM (Compressive Interface Capturing Scheme for Arbitrary Meshes) was employed in this work. CICSAM is a high resolution scheme based on the Normalized Variable Diagram (NVD) [6]:

$$\left(\frac{V_P}{\Delta t}+\sum_{f=1}^{n}\frac{(1-\beta_f)}{2}u_f A_f\right)\gamma_P^t+\sum_{f=1}^{n}\frac{\beta_f}{2}u_f A_f\gamma_N^t=\left(\frac{V_P}{\Delta t}-\sum_{f=1}^{n}\frac{(1-\beta_f)}{2}u_f A_f\right)\gamma_P^{t-1} \\ -\sum_{f=1}^{n}\beta_f u_f A_f\gamma_N^{t-1} \quad . \tag{109}$$

In order to calculate β_f , which is the CICSAM face weighting interpolation factor, a predictor and corrector method is used. These steps are described in full detail in [6]. The matrix form of the indicator function is given the following expressions:

$$a_P^i\gamma_P^t+\sum_{f=1}^{n}a_{N_f}^i\gamma_N^t=b_P^i, \tag{110}$$

$$a_P^i=\frac{V_P}{\Delta t}+\sum_{f=1}^{n}\frac{(1-\beta_f)}{2}\rho_f u_f A_f, \tag{111}$$

$$a_{N_f}^i=\frac{\beta_f}{2}\rho_f u_f A_f, \tag{112}$$

$$b_P^i=\left(\frac{V_P}{\Delta t}-\sum_{f=1}^{n}\frac{(1-\beta_f)}{2}\rho_f u_f A_f\right)\gamma_P^{t-1}-\sum_{f=1}^{n}\frac{\beta_f}{2}\rho_f u_f A_f\gamma_N^{t-1}. \tag{113}$$

The CICSAM scheme requires the calculation of the Courant number Co . It is calculated for each cell using the following expression:

$$Co=\sum_{f=1}^{n}\max\left\{\frac{u_f A_f\Delta t}{V_P},0\right\}. \tag{114}$$

Material properties

The average material properties of each cell is updated using the volume fraction distribution:

$$\psi_P=\gamma_P\psi_0+(1-\gamma_P)\psi_1, \tag{115}$$

where ψ_0 is a material property such as density, viscosity, or thermal conductivity of fluid 0 while ψ_1 is a material property of fluid 1.

Boundary conditions

At the outlet the gradient of the volume fraction is set to zero:

$$\gamma_B = \gamma_P. \tag{116}$$

2.14 Non-Orthogonality Correction

If the mesh is non-orthogonal it is necessary to correct the gradient of dependent variables at the cell face center. The correction method described by Ferziger and Perić (2002) was employed in this work [1]. The gradient at cell face center is calculated using auxiliary nodes P' and N'_f, which are the projection of P and N, respectively, on to the straight line with same direction as the normal of the face:

$$\nabla p_f = \frac{p_{N'_f} - p_{P'}}{|d_f|}. \tag{117}$$

The locations of auxiliary nodes P' and N'_f are calculated as (see Figure 2):

$$\begin{aligned} r_{P'} &= r_f - [(r_f - r_P) \cdot n_f]n_f, \\ r_{N'_f} &= r_f - [(r_f - r_{N_f}) \cdot n_f]n_f, \end{aligned} \tag{118}$$

where r is the position vector of each node. Vector d_f is given by:

$$d_f = (r_{N'_f} - r_{P'}). \tag{119}$$

Pressure at auxiliary nodes P' and N'_f, can be evaluated using the pressure gradient at cell center:

$$\begin{aligned} p_{P'} &= p_P + \nabla p_P \cdot (r_{P'} - r_P), \\ p_{N'_f} &= p_N + \nabla p_{N_f} \cdot (r_{N'_f} - r_{N_f}). \end{aligned} \tag{120}$$

At the boundary, d_f is calculated as a vector from node P', which is the projection of node P to the center of the boundary face (see Figure 4):

$$r_{P'} = r_f - [(r_f - r_P) \cdot n_f]n_f, \tag{121}$$

$$d_f = (r_f - r_{P'}). \tag{122}$$

This correction affects the approximation of diffusive fluxes in Equation 73. In this case, a second term is added to calculate the gradient at cell face center:

$$(\nabla p)_f = \frac{p_{N'_f} - p_{P'}}{|d_f|}, \tag{123}$$

$$(\nabla p)_f = \frac{p_{N_f} - p_P}{|d_f|} + \frac{\nabla p_{N'_f} \cdot (r_{N'_f} - r_{N_f}) - \nabla p_P \cdot (r_{P'_f} - r_{P_f})}{|d_f|}. \tag{124}$$

The gradient at the center of the boundary face is calculated as:

$$(\nabla p)_f = \frac{p_B - p_{P'}}{|d_f|}. \tag{125}$$

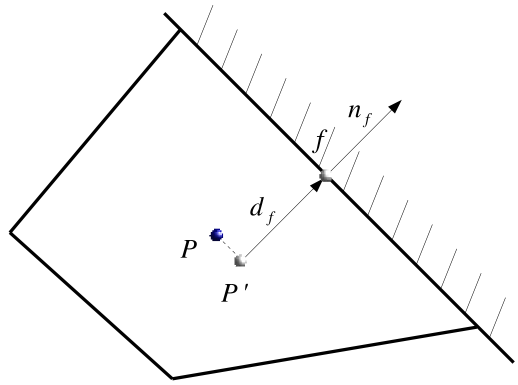


Figure 3: Boundary

2.15 Solution of Linear Equation Systems

Any systems of equations can be solved by Gauss elimination or LU decomposition. However, these methods are quite slow when applied the solution of large sparse matrices. Usually, it is not necessary to solve the system of equations so accurately since the error of discretisation is much larger than the arithmetic computational errors.

Using implicit temporal discretisation, the final set of equations can be written is the following form:

$$M\phi = b, \tag{126}$$

where M is a matrix and b is the source vector. Since the mesh in unstructured, matrix M is sparse thus suitable iterative methods are necessary for the solution of the equations. In this study, the LASPack library was used. It has several iterative solvers such as Jacobi, successive over relaxation (SOR), Chebyshev, and conjugate gradient methods (CG) which can handle non-symmetric sparse systems: CGN, GMRES, BiCG, QMR, CGS, and BiCGStab [14]. A benchmarking test case with 400 control volumes was used to compare the speed of each solver without preconditioning. The matrices were sparse with an average of 5 non-zero elements in each row and a total dimension of 400×400. Computation times for each iterative solver is presented in Table 2.

Table 2: Comparison of several iterative solvers

Solver	Time (s)
`CGN`	44.60
`SSOR`	36.91
`GMRES`	23.93
`QMR`	23.82
`CGS`	20.85
`BiCG`	20.27
`BiCGSTAB`	19.63

By default, BiCGStab method is used for solving momentum, continuity and temperature transport equations and GMRES is used for solving interface capturing equations, both preconditioned with an incomplete LU decomposition (ILU) preconditioner.

2.16 Solution Algorithm

The algorithm starts by setting the initial conditions thus defining the initial state of the flow. In this study, the equations of motion for both fluids are solved using the segregated approach, where the different equations are solved sequentially by iterating over them. The PISO algorithm is used for velocity-pressure coupling. The velocity field is calculated using an estimation of the pressure field. Then the pressure field is calculated to respect continuity and the velocities are corrected accordingly. The temperature field is calculated solving the temperature transport equation. The fraction of both fluids in each cell is calculated using the CICSAM which is highly compressive scheme and guarantees boundedness of the solution. The algorithm performs the following steps:

1. Set initial conditions
2. Calculate volume fraction field γ (Equation 110)
3. Calculate material properties of each cell (Equation 115)

4. Set an initial guess for the pressure field (p) and flux (F) (use values from previous iteration)
5. Calculate velocity field (u, v, w) (Equations 51, 55, 59)
6. Calculate pressure field (p) (Equation 75)
7. Correct the flux (F) to satisfy the continuity equation
8. Correct velocity field (u, v, w) (Equation 60)
9. Calculate temperature field T (Equation 98)
10. Advance in time
11. Finish or go to step 2

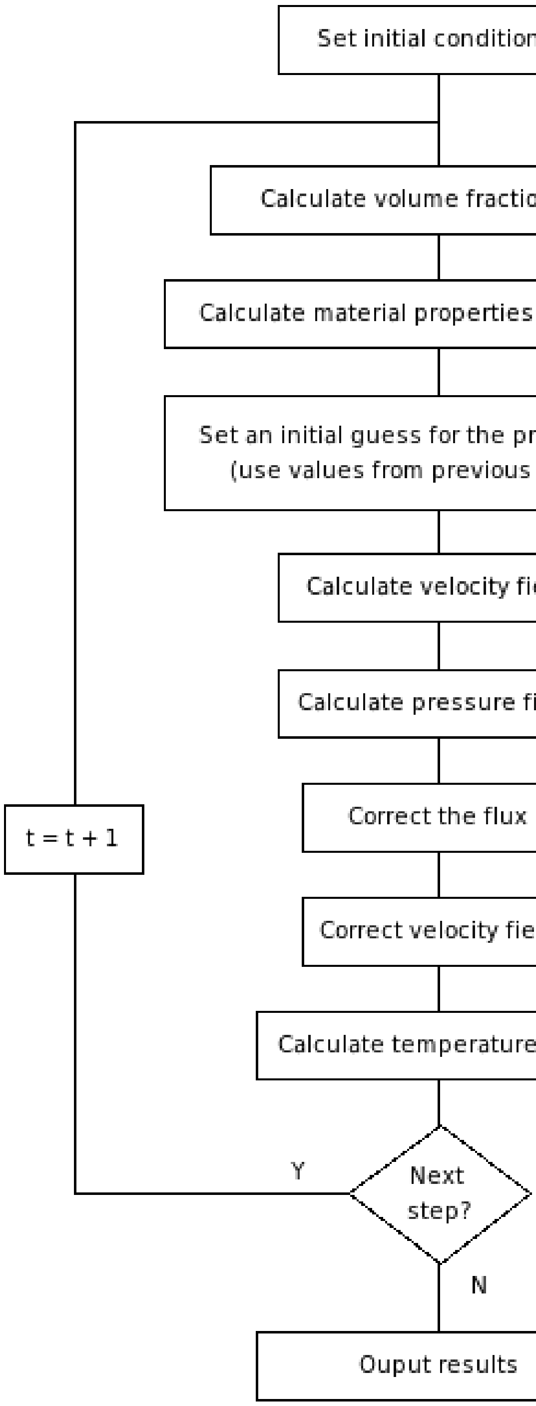


Figure 4: Numerical algorithm flowchart

3 Pre-Processing

OpenFVM requires input data which is stored in four different files regarding mesh (*.msh), boundary conditions (*.bcd), material (*.mtl) and parameters (*.par). The data in each file is grouped using codes. Codes between [10000, 19999] are reserved for different boundary condition types, between [20000, 29999] for material properties and between [30000, 39999] for parameters.

3.1 Geometry

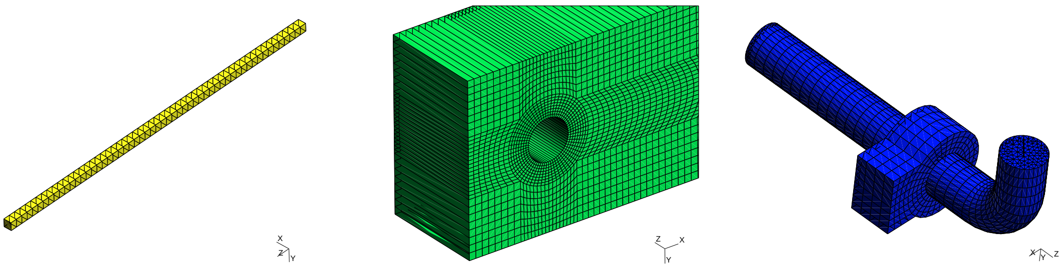
The geometry should be defined using Gmsh, which is a three-dimensional finite element mesh generator with pre- and post-processing capability developed by Christophe Geuzaine and Jean-François Remacle. All geometrical, mesh, solver and post-processing instructions are prescribed either interactively using the graphical user interface (GUI) or in ASCII data files using the Gmsh scripting language. This makes it possible to automate procedures, using loops, conditionals and external systems calls. The Gmsh mesh file format is described in [13]. This CFD code can handle three-dimensional shapes such as tetrahedra, hexahedra and prisms. Boundary conditions are applied using a two-dimensional surface mesh composed of triangles and/or quadrangles. These boundary conditions are applied using the physical surface entity, while initial conditions are applied using the physical volume entity. The connectivity of the mesh is determined and stored before initiating the simulation, using an octree partioning algorithm. The face-based connectivity data structure assumes that each face has none or only one pair. If the face has a pair, it is an internal face and if it does not, it is a boundary face.

3.2 Mesh Type

Although the geometry is always three-dimensional, it is possible to simulate 1D, 2D and 3D cases. In a one-dimensional cases, the mesh is divided only along one axis, in two-dimensional cases, the mesh is divided along two axis and in three-dimensional cases, the mesh is divided in all three dimensions. The mesh can be hybrid in which the mesh contains several types of elements or control volumes.

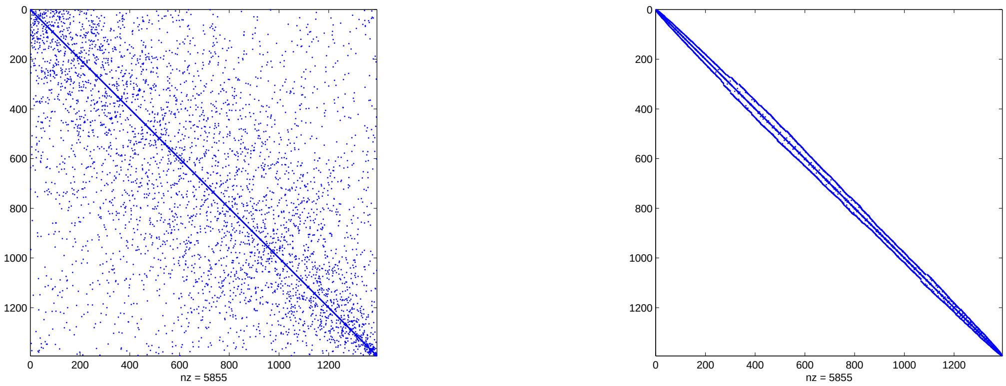
3.3 Mesh Reordering

The Reverse Cuthill-McKee (RCM) algorithm implemented by David Fritzsche is used to reorder the mesh and reduce the bandwidth of the resulting sparse matrices. Figure 6 shows the structure of the sparse matrices before and after the application of the RCM algorithm. It can be observed that the RCM algorithm produces a reordering of the mesh which leads to matrices with a more narrow bandwidth.



1D 2D 3D

Figure 5: Mesh type



Before After

Figure 6: Matrix structure before and after applying the RCM algorithm

3.4 Parallel Processing

Parallelization of the code is done within the PETSc framework. PETSc is a suite of data structures and routines suitable for the development of large-scale scientific applications on parallel and serial computers. OpenFVM uses a Single Program Multiple Data (SPMD) message passing model, i.e. each process runs the same program and performs computations on its own subset of data. Each process has links its neighbouring processes to exchange data. Dependant variables are stored in global vectors with ghost cell padding. Communication is performed when necessary to update information of the ghost cells.

3.5 Domain Decomposition

The mesh can be divided into several regions for parallel processing. The domain decompositon is obtained using Metis. Two partiotioning schemes are implemented: the recursive-bisection scheme and the k-way scheme. The first scheme is used when the number of regions is less or equal to 8, while the second is used for all other cases. After the domain is divided, cells are renumbered and stored in seperate mesh files. Each processor computes one region of the domain. Therefore, the number of processes should be equal to the number of regions. The values of dependent variables are stored in parallel vectors with ghost padding on each processor and values at ghost cells are updated when necessary. The implicit solver produces several parallel global matrices, which are solved in parallel using the library PETSc. Figure 7 shows one region of the domain with global indices. Each process uses local indices to minimize memory storage. Relationship between global and local indices is garanteed by PETSc data structures.

3.6 Units

Table 3: Units

Quantity	Unit (SI)
Length (L)	m
Velocity (U)	m/s
Pressure (p)	kg/(m · s)
Density (ρ)	kg/m^3
Viscosity (μ)	kg/(m · s)
Specific Heat (C)	m^2/(s^2 · K)
Thermal Conductivity (k)	(m · kg)/(s · K)

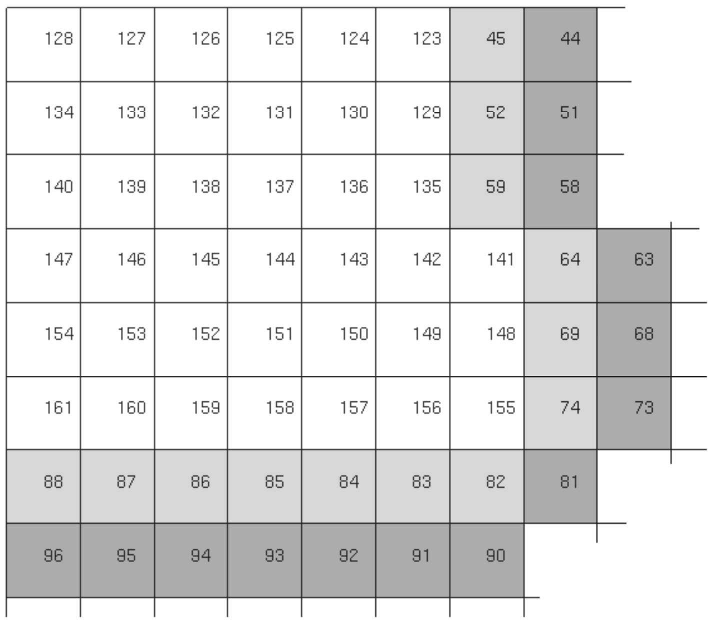


Figure 7: Partition and ghost cells

3.7 Boundary Conditions

The format of the boundary conditions file (*.bcd) is described in Figure 8. The value of *nb-bcd-types* is the number of boundary types defined in the file, `$CODE` is an integer that defines the boundary type, *nb-entries* is the number of entries of a given boundary type and *desc* is a short description of the boundary type. The value of *reg-phys* is the number of the physical entity defined in Gmsh. The rest of the line defines the appropriate expressions for the velocity vector (u, v, w), pressure (p), temperature (T) and indicator function (α) at the boundary. These expressions may be defined as a constant or a function of three variables x, y and z. The internal expression parser interprets a list of operators and functions defined in Table 4.

`$Title OpenFVM`
`$File Boundary conditions file`
`$Parameter 1` *nb-bcd-types* `Description of $Code`
`$Code` *nb-entries desc*
reg-phys fu fv fw fp fT fs
`...`
`$EndFile`

Figure 8: Boundary conditions file format

Table 4: Operators and functions

Expression	Type	Expression	Type
`+`	Addition	`sqrt`	Square root
`-`	Subtraction	`sin`	Sine
`*`	Multiplication	`cos`	Cosine
`/`	Division	`atan`	Inverse tangent
`^`	Power	`log`	Natural logarithm
`abs`	Absolute value	`exp`	Exponential function

The codes for each boundary type (`$CODE`) are listed in Table 5. Entrance of a fluid into the domain can be specified using boundary type "Inlet", while the exit of the fluid is specified using boundary type "Outlet". Boundary type "Cyclic" allows the application of boundary conditions for one-dimensional domains. A surface assigned with boundary type "Adiabatic Wall" does not exchange heat, while boundary types "Wall", "Moving Wall" and "Surface" influence heat transfer. Initial conditions of each region of the mesh can be defined using boundary type "Volume".

Table 5: Boundary types

`$CODE`	Type	`$CODE`	Type
10000	Empty	10160	Slip
10020	Cyclic	10170	Wall
10050	Open	10180	Adiabatic wall
10100	Inlet	10190	Moving wall
10110	Pressure inlet	10200	Surfac
10150	Outlet	10250	Volume

An example of a boundary conditions file is shown in Figure 9. Boundary and initial conditions can be assigned to surfaces and volumes. This has to be considered when modeling the geometry in gmsh. In this example, the x component of the inlet velocity is defined by a mathematical expression, in this case, as a function of y. Physical surface 61 is defined as "Inlet" and 62 as "Outlet". Physical entity 60 which can be a group of surfaces and is defined with boundary type "Wall", i.e. no-slip velocity condition.

3.8 Material

The format of the material file (*.mtl) is described in Figure 10. The value of *nb-mtl-prop* is the number of properties defined in the file, `$CODE` is an integer that

```
$Title OpenFVM
$File Boundary conditions file
$Boundary 1 4 Description of $Code
10100 1 Inlet - Physical surfaces (ID u v w p T s)
     61 2/3*(y+0.4)^2 0.0 0.0 0.0 23.0 0.0
10150 1 Outlet - Physical surfaces (ID u v w p T s)
     62 0.0 0.0 0.0 0.0 23.0 0.0
10200 1 Wall - Physical surfaces (ID u v w p T s)
     60 0.0 0.0 0.0 0.0 23.0 0.0
10250 1 Volume - Physical volumes (ID u v w p T s)
     63 0.0 0.0 0.0 0.0 23.0 0.0
$ENDF
```

Figure 9: Boundary conditions example

defines the material property, *nb-entries* is the number of entries of a given material property, *desc* is a string with a short description of the materail property and *value* is the value the material property. This file contains the required material properties of both fluids such as density, viscosity, thermal conductivity, specific heat and surface tension.

```
$Title OpenFVM
$File Material file
$Parameter 1 nb-mtl-prop Description of $Code
$Code nb-entries desc
value
...
$EndFile
```

Figure 10: Material file format

3.9 Parameters

The parameter file (*.par) contains simulation data and options. This file defines tolerances, maximum number of iterations, output options, time advancement criteria, time controls and others. Its format is described in Figure 11. The value of *nb-par* is the number of parameters defined in the file, `$CODE` is an integer that defines the parameter type, *nb-entries* is the number of entries of a given parameter, *desc* is a short description of the parameter and *value-1* are the values the the parameter.

```
$Title OpenFVM
$File Parameter file
$Parameter 1 nb-par Description of $Code
$Code nb-entries desc
value-1 value-2 ... value-n
...
$EndFile
```

Figure 11: Parameter file format

3.10 Results

The results are stored in Gmsh post-processing format which is described in [13]. The code can output node-averaged, face/cell-centered plots of all the dependent variables on faces and cells, velocity vector plots and vorticity , in both ASCII or binary format. Gmsh is also used for probing data using plug-ins and scripting language. The data can be exported and used by other applications for further analysis, such as Gnuplot.

4 Tutorial

This chapter describes the process of setup, simulation and post-processing for a test case in order to introduce the user to the basic procedures of running OpenFVM. More test cases can be accessed in the examples directory.

4.1 Non-Isothermal Lid-Driven Cavity Flow

This tutorial shows how to prepare, run and post-process a case involving the non-isothermal, incompressible flow in a closed two-dimensional square domain. The geometry is shown in Figure 12. The flow is assumed as laminar and will be solved on a uniform mesh.

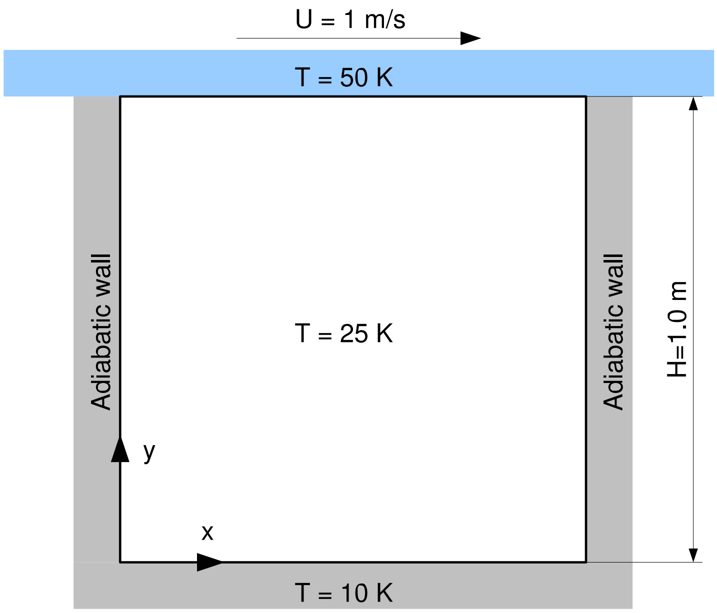


Figure 12: Geometry of the lid-driven tutorial flow

First create a new directory inside the examples directory case called tutorial. Start Gmsh and click `Elementary` > `Add` > `New` > `Point` to create points. Create points (0,0,0), (1,0,0), (1,1,0) and (0,1,0). Select module `Geometry` and click on `Edit`. If Gmsh is configured corectly an editor will open with the following lines:

```
Point(1) = {0, 0, 0, 0.1};
```

```
Point(2) = {1, 0, 0, 0.1};
Point(3) = {1, 1, 0, 0.1};
Point(4) = {0, 1, 0, 0.1};
```

Alter this file in the editor to look like the following listing. Add three lines with variables for dimensions in x, y and z directions (`dx`, `dy`, `dz`), number of divisions in each direction (`nx`, `ny`, `nz`) and the cell dimensions (`cx`, `cy`, `cz`):

```
dx = 1.0;
dy = 1.0;
dz = 5.0;

nx = 30;
ny = 30;
nz = 1;

cx = dx/nx;
cy = dy/ny;
cz = (cx + cy)/2;

Point(1) = {0.0,0.0,0.0,1.0};
Point(2) = {dx,0.0,0.0,1.0};
Point(3) = {dx,dy,0.0,1.0};
Point(4) = {0.0,dy,0.0,1.0};
```

Save the file and click on Gmsh `Reload` to reload the new data. Go to menu `Tools` > `Options...` and click on `Geometry` in the list. Go to the `Visibility` tab and check the `Point numbers` check box. Click `Apply` and then `Cancel` to exit the `Options` dialog box. Now click `Elementary` > `Add` > `New` > `Straight line` to create lines. Create four lines from point 1 to point 2, from point 2 to point 3, from point 3 to point 4 and from point 4 to point 1. Click `Elementary` > `Add` > `New` > `Plane surface` to create a surface. Click on any line to create a loop and then press key `E` to end the command. A new surface will be created. Go to menu `Tools` > `Options...` and click on `Geometry` in the list. Go to `Visibility` tab and check the "Surface numbers" check box. Click `Apply` to view the number of the new surface and then "Cancel" to exit the `Options` dialog box. Select module `Geometry` and click on `Edit`. The file should look like this:

```
dx = 1.0;
dy = 1.0;
dz = 5.0;

nx = 20;
```

```
ny = 20;
nz = 1;

cx = dx/nx;
cy = dy/ny;
cz = (cx + cy)/2;

Point(1) = {0.0,0.0,0.0,1.0};
Point(2) = {dx,0.0,0.0,1.0};
Point(3) = {dx,dy,0.0,1.0};
Point(4) = {0.0,dy,0.0,1.0};

Line(1) = {1,2};
Line(2) = {2,3};
Line(3) = {3,4};
Line(4) = {4,1};
Line Loop(5) = {1,2,3,4};
Plane Surface(6) = {5};
```

Add these lines to the end of the file and save:

```
out[] = Extrude {0,0,dz} {
Surface{6};
Layers { 1 };
Recombine;
};
```

In Gmsh select module `Geometry` and click on `Reload`. Now you should see the three-dimensional extrusion of the surface. In order to define the mesh add the following lines:

```
Transfinite Line {4,10,2,8} = nx + 1 Using Progression 1.0;
Transfinite Line {3,9,1,11} = ny + 1 Using Progression 1.0;
Transfinite Line {14,18,13,22} = nz + 1 Using Progression 1.0;

Transfinite Surface {6} = {3,2,1,4};
Transfinite Surface {27} = {5,14,2,3};
Transfinite Surface {15} = {5,3,4,6};
Transfinite Surface {28} = {6,10,14,5};
Transfinite Surface {23} = {14,2,1,10};
Transfinite Surface {19} = {6,10,1,4};

Recombine Surface {27,23,6,19,15,28};
```

and to define the top, bottom, left, right, front and back physical surfaces and the physical volume add the following:

```
// Top surface
Physical Surface(33) = {23};
// Bottom surface
Physical Surface(34) = {15};
// Left surface
Physical Surface(35) = {27};
// Right surface
Physical Surface(36) = {19};
// Front surface
Physical Surface(37) = {28};
// Back surface
Physical Surface(38) = {6};
// Box volume
Physical Volume (39) = {out[1]};
```

Save the file as `tutorial.geo`. In Gmsh click menu `File` > `Open...` and open `tutorial.geo`. Go to module `Mesh`, click `3D` and click `Save` to save the mesh. This creates file `tutorial.msh`. Now that the mesh has been created it is necessary to define other input data such as boundary conditions. Open a text editor and copy the following lines to an empty file:

```
$Title OpenFVM
$File Boundary conditions file
$Boundary 1 4 Description of $Code
10000 2 Empty - Physical surfaces (ID u v w p T s)
       37 0.0 0.0 0.0 0.0 0.0 0.0
       38 0.0 0.0 0.0 0.0 0.0 0.0
10170 2 Wall - Physical surfaces (ID u v w p T s)
       33 1.0 0.0 0.0 0.0 50.0 0.0
       34 0.0 0.0 0.0 0.0 10.0 0.0
10180 2 Adiabatic wall - Physical surfaces  (ID u v w p T s)
       35 0.0 0.0 0.0 0.0 0.0 0.0
       36 0.0 0.0 0.0 0.0 0.0 0.0
10250 1 Volume - Physical volumes (ID u v w p T s)
       39 0.0 0.0 0.0 0.0 25.0 0.0
$EndFile
```

Save this file as `tutorial.bcd` in the same directory as above. In the same manner, copy these lines to a file called `tutorial.mtl`:

```
$Title OpenFVM
$File Material file
$Material 1 12 Description of $Code
20012 1 Compressibility of fluid 0
0.0
20015 1 Density of fluid 0
1.0
20021 1 Viscosity of fuild 0
1.0E-2
20022 1 Specific heat of fuild 0
1.0
20023 1 Thermal conductivity of fluid 0
0.5
21012 1 Compressibilty of fluid 1
0.0
21015 1 Density of fluid 1
1.0
21021 1 Viscosity of fluid 1
1.0E-2
21022 1 Specific heat of fuild 1
1.0
21023 1 Thermal conductivity of fluid 1
0.5
26050 1 Constant surface tension (fluid 0 / fluid 1)
0.0
26060 1 Thermal conductivity (boundary)
0.5
$EndFile
```

Create one more file copying these lines to a file called `tutorial.par`:

```
$Title OpenFVM
$File Parameter file
$Parameter 1 26 Description of $Code
30005 1 Convection interpolation scheme
1 1 1 1 0 1
30020 1 Binary output
1
30040 1 Calculate variable (u v w p T s)
1 1 0 1 0 0
30100 1 Steady state
1
30105 1 Convergence for steady state solutions
```

```
1E-6 1E-6 1E-6 1E-6 1E-6 1E-6
30200 1 Adjust time interval
0
30201 1 Maxmimum Courant number
0.9
30400 1 Number of saves
1
30450 1 Write face scalars (u v w p T s)
0 0 0 0 0 0
30455 1 Write face vectors (uvw)
0
30460 1 Write element scalars (u v w p T s)
0 0 0 0 0 0
30465 1 Write element vectors (uvw)
1
30470 1 Write vorticity (x y z)
0 0 0
30475 1 Write stream function (xy)
1
30485 1 Probe (u v w p T s)
1 0 0 0 0 0
30550 1 Maximum number of non-othorgonal corrections
0
30600 1 Convergence criterion (matrix solution)
1E-12 1E-12 1E-12 1E-12 1E-12 1E-12
30601 1 Maximum number of iterations (matrix solution)
5000 5000 5000 5000 5000 5000
30650 1 Matrix solver (u v w p T s)
3 3 3 4 3 3
30651 1 Matrix preconditioner (0-Null, 1-Jacobi, 2-SOR, 3-ILU)
3 3 3 3 3 3
30800 1 Interface scheme factor - CICSAM
1.0
30900 1 Maximum number of CICSAM corrections
2
32000 1 Start time
0.0
32001 1 End time
100.0
32002 1 Time interval
0.01
34000 1 Gravity vector
0.0 0.0 0.0
```

```
$EndFile
```

Go to the tutorial directory and type:

```
../OpenFVM tutorial d 1
```

Using this command a new mesh file is created: `tutorial.000.msh`. In this file, elements are renumbered from 0 to n - 1. In the same directory type:

```
../OpenFVM tutorial f 1
```

In order to track the residuals copy the following lines and save the file as `tutorial.plt`:

```
set title 'Convergence'
set xlabel 'Iteration'
set ylabel 'Residual'
set logscale y
plot 'tutorial.res' using 1:2 title 'u' with lines, \
     'tutorial.res' using 1:3 title 'v' with lines, \
     'tutorial.res' using 1:5 title 'p' with lines
pause -1
```

The residual history as a function of iteration number is shown in Figure 13:

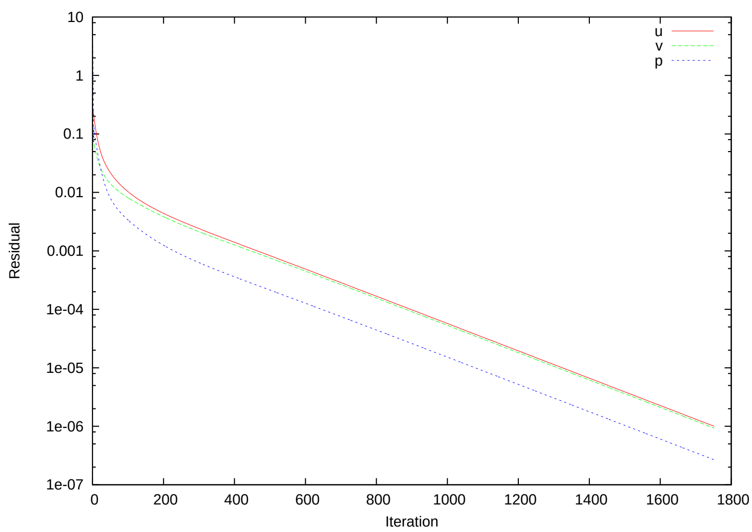


Figure 13: Convergence of the non-isothermal lid-driven cavity flow

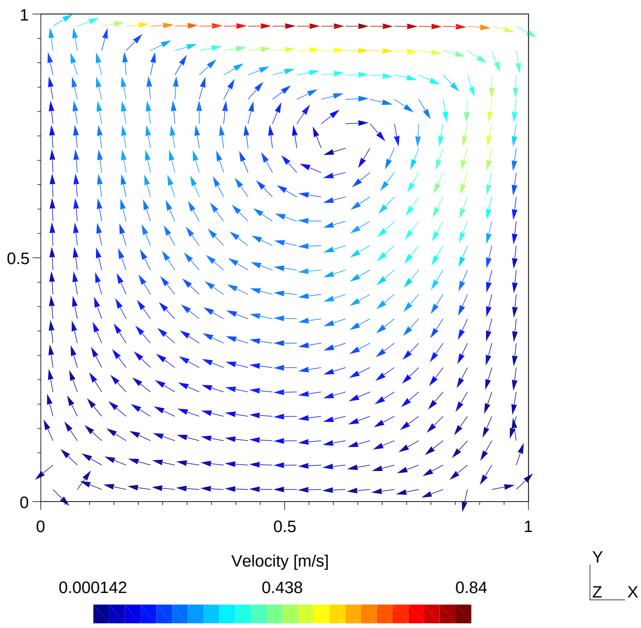


Figure 14: Velocity vectors of the lid-driven cavity flow

The results for the vector plots are shown in Figure 14. To visualize the results open the file `tutorial.pos` in Gmsh.

The Gmsh script can be used to automatically porbe the u-velocity along the center line and save it to a file. Copy the following lines to a file named `tutorial.scr`.

```
// Include "tutorial.geo";

Include "tutorial.000.prb";

nbviews = PostProcessing.NbViews;

View[0].Name = "U-Velocity [m/s]";
View[0].Visible = 0;

Plugin(CutGrid).X0 = 0.5;
Plugin(CutGrid).Y0 = 0.0;
Plugin(CutGrid).Z0 = 2.5;
Plugin(CutGrid).X1 = 1.0;
Plugin(CutGrid).Y1 = 1.0;
Plugin(CutGrid).Z1 = 2.5;
Plugin(CutGrid).X2 = 0.5;
```

```
Plugin(CutGrid).Y2 = 1.0;
Plugin(CutGrid).Z2 = 2.5;
Plugin(CutGrid).nPointsU = 1;
Plugin(CutGrid).nPointsV = 250;
Plugin(CutGrid).ConnectPoints = 0;
Plugin(CutGrid).iView = -1;

Plugin(CutGrid).Run;

View[1].Axes = 2;
View[1].Type = 2;
View[1].IntervalsType = 2;
View[1].Name = "U-Velocity [m/s] - Centerline";
View[1].AxesLabelX = "U/U0";
View[1].AxesLabelY = "y/y0";

PostProcessing.Format = 4;

// Save results to graph for using gnuplot
Save View[1] "tutorial.txt";

//System "gnuplot tutorial.gph";
```

Create a file called `tutorial.gph` to create a plot:

```
#set terminal postscript eps   # set output to eps file
#set output 'tutorial.eps'
set xtic auto                  # set xtics automatically
set ytic auto                  # set ytics automatically
set title 'Velocity profiles at the centerline \
           of the lid-driven cavity at Re = 100'
set xlabel 'U/U0'
set ylabel 'y/y0'
set xrange[-0.4:1.0]
set yrange[0.0:1.0]
set key 0.8,0.6
set pointsize 1
plot 'tutorial.txt' using 4:2 title 'tutorial' with lines 1
pause -1
```

Now open the `tutorial.scr` file in Gmsh and run the following command in the tutorial directory:

```
gnuplot tutorial.gph
```

at command prompt. This graph should show the velocity profile at the centerline of the lid-driven cavity.

References

[1] Ferziger J. and Perić M., *Computational methods for fluid dynamics*, Springer, 2002.

[2] Chang, R. and Yang W., *Numerical simulation of mold filling in injection molding using a three-dimensional finite volume approach*, International journal for numerical methods in fluids, 2001.

[3] Demirdzic, I., Muzaferija, S. and Peric, M., *Computation of turbulent flows in complex geometries*, Applied industrial fluid dynamics, 2001.

[4] Rusche, H., *Computational fluid dynamics of dispersed two-phase flows at high phase fractions*, phD thesis, Imperial college, 2002.

[5] Muzaferija S. and Gosman D., *Finite-volume CFD procedure and Adaptive Error Control Strategy for grids of arbitrary topology*, Journal of computational physics, 1997.

[6] Ubbink, O., *Numerical prediction of two fluid systems with sharp interfaces*, phD thesis, Imperial college of science, technology and medicine, 1997.

[7] Yang, W.; Peng, A.; Liu, L.; Hsu, D.; Chang, R., *Integrated numerical simulation of injection molding using true 3D approach*, Antec, 2004.

[8] Versteeg, H.K. and Malalasekera, W., *An introduction to computational fluid dynamics: the finite volume method*, Longman Group Ltd, 1998.

[9] Issa, R.I., *Solution of the implicitly discretised fluid flow equations by operator-splitting*, Journal of Computational Physics, 62, 40-65, 1986.

[10] Ghia, U.; Ghia, K. and Shin, C.T., *High-Re solutions for incompressible flow using the Navier-Stokes Equations and a Multigrid Method*, Journal of computational physics, 48, 387-411, 1982.

[11] Lomax, H.; Pulliam, T.H. and Zingg, D.W., *Fundamentals of computational fluid dynamics*, Springer-Verlag, 2001.

[12] Kamerich, E., *A guide to Maple*, Springer-Verlag, 1999.

[13] Geuzaine, C. and Remacle, J-F., *Gmsh reference manual*, 2005.

[14] Skalicky, T., *LASPack reference manual*, 1995.