

METHODOLOGIES FOR THE NUMERICAL SIMULATION OF FLUID FLOW IN INTERNAL COMBUSTION ENGINES

by
Ezequiel José López

A dissertation submitted to the Postgraduate Department of the
FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS
for partial fulfillment of the requirements
for the degree of
DOCTOR IN ENGINEERING
Field of Computational Mechanics

UNIVERSIDAD NACIONAL DEL LITORAL

2009

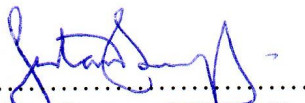


UNIVERSIDAD NACIONAL DEL LITORAL
Facultad de Ingeniería y Ciencias Hídricas

Santa Fe, 4 de mayo de 2009.

Como miembros del Jurado Evaluador de la Tesis de Doctorado titulada "*Metodologías para la simulación numérica del flujo de fluidos en motores de combustión interna*", desarrollada por el Ing. Ezequiel José López, certificamos que hemos evaluado la Tesis y recomendamos que sea aceptada como parte de los requisitos para la obtención del título de Doctor en Ingeniería – Mención Mecánica Computacional.

La aprobación final de esta disertación estará condicionada a la presentación de dos copias encuadernadas de la versión final de la Tesis ante el Comité Académico del Doctorado en Ingeniería.

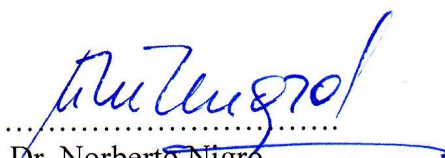

.....
Dr. Gustavo Buscaglia

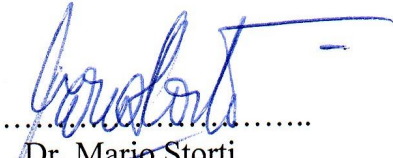

.....
Dr. Néstor Calvo


.....
Dr. Mario Díaz Terrado

Santa Fe, 4 de mayo de 2009

Certifico haber leído esta Tesis preparada bajo mi dirección y recomiendo que sea aceptada como parte de los requisitos para la obtención del título de Doctor en Ingeniería – Mención Mecánica Computacional.


.....
Dr. Norberto Nigro
Director de Tesis


.....
Dr. Mario Storti
Codirector de Tesis

Universidad Nacional del Litoral
Facultad de Ingeniería y
Ciencias Hídricas

Secretaría de Posgrado

Ciudad Universitaria
C.C. 217
Ruta Nacional N° 168 - Km. 472,4
(3000) Santa Fe
Tel: (54) (0342) 4575 229
Fax: (54) (0342) 4575 224
E-mail: posgrado@fich.unl.edu.ar

A mis amores, Sofía y Brenda

A la memoria de Jorge 'Toto' Toth

Author's Legal Declaration

This dissertation have been submitted to the Postgraduate Department of the *Facultad de Ingeniería y Ciencias Hídricas* for partial fulfillment of the requirements for the degree of Doctor in Engineering - Field of Computational Mechanics of the *Universidad Nacional del Litoral*. A copy of this document will be available at the University Library and it will be subjected to the Library's legal normative.

Some parts of the work presented in this thesis have been (or are going to be) published in the following journals: *International Journal for Numerical Methods in Engineering*, *International Journal for Numerical Methods in Fluids* and *Latin American Applied Research*.

Any comment about the ideas and topics discussed and developed through this document will be highly appreciated.

Ezequiel José López

Contents

Nomenclature	xiii
Introduction	xviii
Introducción	xxi
1 Governing equations and numerical approximation	1
1.1 Governing equations	1
1.1.1 Turbulence modeling	3
1.1.2 Boundary conditions	4
1.1.3 Arbitrary Lagrangian Eulerian description of governing equations	6
1.2 Numerical implementation	7
1.2.1 Finite element formulation	7
1.2.2 Time discretization	8
1.2.3 Dynamic boundary conditions using Lagrange multipliers	9
2 Mesh dynamics	11
2.1 Mesh quality	12
2.2 The mesh dynamics strategy	13
2.2.1 Functional design	13
2.2.2 Differential predictor	19
2.2.3 Avoiding the relaxation of the initial mesh	21
2.2.4 Results	23
2.3 Simultaneous mesh untangling and smoothing	29
2.3.1 Functional regularization	30
2.3.2 Solution strategy	31
2.3.3 Results	33
2.4 Conclusion	41
3 Resolution of compressible flows in the low Mach number limit	46
3.1 Problem definition and eigenvalues analysis	48
3.1.1 Preconditioning strategies	51
3.2 Numerical implementation	53
3.2.1 Finite element formulation	53
3.2.2 Dynamic boundary conditions	56
3.3 Results	57
3.3.1 Flow in a lid driven cavity	57
3.3.2 Flow in a channel with a moving indentation	57

3.3.3	In-cylinder flow in an opposed-piston engine	65
4	Thermodynamic and gas-dynamic based IC engine models	70
4.1	Mathematical models	71
4.1.1	Pipe model	71
4.1.2	Cylinder model	72
4.1.3	Valve model	75
4.1.4	Pipe junctions model	76
4.2	Numerical implementation	77
4.3	Results	79
4.3.1	Four-stroke spark-ignition engine test	79
4.3.2	Two-stroke spark-ignition engine test	80
4.3.3	Four-stroke diesel engine test	83
5	Coupling of 1D/multi-D domains for compressible flows	85
5.1	Coupling domains through absorbing boundary conditions	87
5.1.1	Algorithm Robin/Robin	88
5.1.2	Robin/Dirichlet scheme for right-going characteristics	88
5.1.3	Dirichlet/Robin scheme for left-going characteristics	89
5.1.4	Characteristics-based split for general systems	90
5.1.5	Resolution by penalization	91
5.2	Coupling for implicit schemes ‘monolithically’ solved	92
5.2.1	Coupling of 1D/multi-D domains	92
5.3	Results	93
5.3.1	1D/1D coupling	93
5.3.2	2D/1D coupling	93
5.3.3	3D/1D coupling	95
6	Numerical simulation of the MRCVC engine	104
6.1	Operation and geometry of MRCVC	104
6.2	Engine performance characteristics using 0D/1D models	107
6.2.1	Port design 1	108
6.2.2	Port design 2	109
6.2.3	Comparison of port designs	112
6.3	Numerical simulation of fluid flow in the MRCVC engine	112
6.3.1	Computational mesh dynamic problem	112
6.3.2	Computational fluid dynamic problem	114
7	Conclusions and future work	123
A	A finite element method for incompressible flows	125
A.1	Navier-Stokes equations for incompressible flows	125
A.1.1	Boundary conditions	125
A.2	Finite element formulation	126

B	Mesh generation using a CMD technique	128
B.1	Generation of orthogonal meshes	128
B.1.1	Transformation for the reentrant corner at right angle	128
B.1.2	Transformation of a square into a rhombus	130
B.1.3	Ellipse	132
C	Description of MRCVC engine geometry	135
C.1	Volume and wall area of the chamber	135
C.1.1	Interval 1	135
C.1.2	Interval 2	136
C.1.3	Interval 3	137
C.1.4	Interval 4	137
C.1.5	Interval 5	138
D	Resumen extendido en castellano	139
D.1	Ecuaciones de gobierno	139
D.1.1	Formulación mediante elementos finitos	140
D.2	Dinámica de la malla	140
D.2.1	Estrategia de <i>untangling-smoothing</i> simultáneos	141
D.3	Resolución de flujos compresibles a bajo número de Mach	142
D.3.1	Formulación del problema y análisis de autovalores	142
D.3.2	Estrategias de preconditionamiento	143
D.4	Acoplamiento de dominios para flujo compresible	143
D.4.1	Acoplamiento mediante condiciones de contorno absorbentes	144
D.4.2	Acoplamiento para esquemas implícitos resueltos ‘molíticamente’	145
D.5	Simulación numérica del motor rotativo MRCVC	145
D.6	Conclusiones	147

List of Figures

1.1	Two-stroke engine scheme.	9
1.2	Change in the type of boundary condition due to the mesh movement.	10
2.1	Invariance under dilatation.	14
2.2	Convexity preservation.	15
2.3	Relaxation of meshes.	22
2.4	Compensation for initial deformation in reference mesh.	23
2.5	Mesh deformation with surface refinement.	23
2.6	Domain of step 2D test.	24
2.7	Mesh deformations for step 2D test: 50 % (left), 90 % (center) and 99 % (right).	24
2.8	Step 2D minimum and mean element quality as a function of mesh deformation.	25
2.9	Problem definition for step 3D test.	25
2.10	Mesh deformations for step 3D test: 50 % (left), 80 % (center) and 87 % (right).	26
2.11	Step 3D minimum and mean element quality as a function of mesh deformation.	26
2.12	Problem definition for square within another square test.	27
2.13	Mesh deformations for square within another square test: 50 % (left), 90 % (center) and 99 % (right).	27
2.14	Minimum and mean element quality as a function of mesh deformation for square within another square test.	28
2.15	Initial mesh for the translation, rotation and bending 2D test.	28
2.16	Final deformed meshes for translation (top), rotation (center) and bending (bottom) tests.	29
2.17	Function $h(V)$	31
2.18	Mesh deformation of 50 % for step 2D test, initial (left) and final (right) meshes.	34
2.19	Mesh deformation of 90 % for step 2D test, initial (left) and final (right) meshes.	34
2.20	Mesh deformation of 99 % for step 2D test, initial (left) and final (right) meshes.	34
2.21	Mesh quality as a function of iterations for step 2D test.	35
2.22	Computational time in terms of the relative domain deformation for step 2D test.	35

2.23	Mesh deformation of 50 % for step 3D test, initial (left) and final (right) meshes.	36
2.24	Mesh deformation of 80 % for step 3D test, initial (left) and final (right) meshes.	36
2.25	Mesh deformation of 87 % for step 3D test, initial (left) and final (right) meshes.	37
2.26	Mesh quality as a function of iterations for step 3D test.	37
2.27	Computational time in terms of the relative domain deformation for step 3D test.	38
2.28	Initial tangled mesh (left) and the resulting final mesh (right) for the scaled cube test.	38
2.29	Mesh quality as a function of iterations for the scaled cube test.	39
2.30	Axisymmetrical flowmeter.	40
2.31	Mesh close-up for the axisymmetrical flowmeter.	40
2.32	Geometry of the combustion chamber of the diesel engine.	41
2.33	Mesh quality as a function of crank angle for the diesel engine.	42
2.34	Minimum mesh dihedral angle as a function of crank angle for the diesel engine.	42
2.35	Mesh quality field at 0 crank angle degree (TDC) for the diesel engine.	43
2.36	Mesh quality field over the transversal planes shown in figure 2.35.	43
2.37	Mesh quality field at 108.5 crank angle degree (maximum intake valve lift) for the diesel engine.	44
2.38	Mesh quality field over the transversal planes shown in figure 2.37.	44
2.39	Mesh quality field at 600 crank angle degree (maximum exhaust valve lift) for the diesel engine.	45
2.40	Mesh quality field over the transversal planes shown in figure 2.39.	45
3.1	Condition number as a function of Courant number CFL_c with $M = 1 \times 10^{-3}$ and $M_\epsilon = 1 \times 10^{-6}$ for the inviscid case.	52
3.2	Condition number as a function of Courant number CFL_c for several Reynolds numbers with the reference Mach number for the inviscid case and $M = 1 \times 10^{-3}$	53
3.3	Condition number as a function of Reynolds number with $CFL_c = 1 \times 10^4$, 1×10^2 and $M = 1 \times 10^{-3}$	53
3.4	Condition number as a function of Reynolds number for several CFL_c numbers, with $M = 1 \times 10^{-3}$	54
3.5	Density field ($[\text{kg}/\text{m}^3]$) for flow in a lid driven cavity.	58
3.6	Magnitude of the velocity field ($[\text{m}/\text{s}]$) for flow in a lid driven cavity.	58
3.7	Pressure perturbation field ($[\text{Pa}]$) for flow in a lid driven cavity.	58
3.8	Comparison of u_1 velocity component at vertical centerline of the cavity with numerical solution by Ghia <i>et al.</i> [29].	59
3.9	Comparison of u_2 velocity component at horizontal centerline of the cavity with numerical solution by Ghia <i>et al.</i> [29].	59
3.10	Geometry of the channel (not to scale): $b = 1$ cm, $l_1 = 9.85$ cm, $l_2 = 18.0$ cm.	59

3.11	Density field ($[\text{kg}/\text{m}^3]$) for the flow in a channel with a moving indentation computed by using the UP strategy. From top to bottom, times $t^* = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$ and 1 .	60
3.12	Magnitude of the velocity field ($[\text{m}/\text{s}]$) for the flow in a channel with a moving indentation computed by using the UP strategy. From top to bottom, times $t^* = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$ and 1 .	61
3.13	Pressure perturbation field ($[\text{Pa}]$) for the flow in a channel with a moving indentation computed by using the UP strategy. From top to bottom, times $t^* = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$ and 1 .	62
3.14	Comparison of predicted and experimentally observed positions of first three vortices center.	62
3.15	Density field ($[\text{kg}/\text{m}^3]$) for the flow in a channel with a moving indentation computed by using the NP strategy. From top to bottom, times $t^* = 0.2, 0.4, 0.6, 0.8$ and 1 .	63
3.16	Magnitude of the velocity field ($[\text{m}/\text{s}]$) for the flow in a channel with a moving indentation computed by using the NP strategy. From top to bottom, times $t^* = 0.2, 0.4, 0.6, 0.8$ and 1 .	63
3.17	Pressure perturbation field ($[\text{Pa}]$) for the flow in a channel with a moving indentation computed by using the NP strategy. From top to bottom, times $t^* = 0.2, 0.4, 0.6, 0.8$ and 1 .	64
3.18	Pressure perturbation field ($[\text{Pa}]$) for the flow in a channel with a moving indentation computed by using the NP strategy at $t^* = 0.5$, where the scale was modified.	64
3.19	Pressure perturbation field ($[\text{Pa}]$) for the flow in a channel with a moving indentation computed by using the NP strategy at $t^* = 0.5$ represented as 3D elevation and where the scale was modified.	65
3.20	Geometry of the simplified 2D model for the opposed-piston engine case (pistons at EDC).	66
3.21	Density ($[\text{kg}/\text{m}^3]$, left) and pressure ($[\text{Pa}]$, right) fields at EDC (0°).	67
3.22	Density ($[\text{kg}/\text{m}^3]$, left) and pressure ($[\text{Pa}]$, right) fields at IDC (180°).	67
3.23	Density ($[\text{kg}/\text{m}^3]$, left) and pressure ($[\text{Pa}]$, right) fields at 270° .	68
3.24	Cylinder mean density (left) and pressure (right) through a cycle for the opposed-piston engine.	68
3.25	Magnitude of flow velocity field ($[\text{m}/\text{s}]$) at several instants during a cycle.	69
4.1	Indicated power as a function of rpm.	80
4.2	Torque as a function of rpm.	80
4.3	Average mass flow rate of air at intake port as a function of rpm.	81
4.4	Computational model of QUB 400 engine.	81
4.5	Variation of pressure during a cycle.	82
4.6	Mass flow rates through transfer and exhaust ports.	82
5.1	Sketch of 1D domain splitting.	87
5.2	Gas discharge from a reservoir to the atmosphere.	94
5.3	Time evolution of solution at coupling sections.	94
5.4	Exhaust manifold geometry.	95
5.5	Geometry of the junction 3-to-1.	96

5.6	Computational models to solve a branch of the exhaust manifold.	96
5.7	Pressure field over the junction 3-to-1 skin at $t^* = 0$	97
5.8	Pressure field over the junction 3-to-1 skin at $t^* = 0.4$	97
5.9	Pressure field over the junction 3-to-1 skin at $t^* = 0.8$	98
5.10	Streamlines at $t^* = 0$ for the junction 3-to-1.	98
5.11	Streamlines at $t^* = 0.4$ for the junction 3-to-1.	99
5.12	Streamlines at $t^* = 0.8$ for the junction 3-to-1.	99
5.13	Time evolution of density at coupling sections through a cycle.	100
5.14	Time evolution of axial velocity at coupling sections through a cycle.	101
5.15	Time evolution of pressure at coupling sections through a cycle.	101
5.16	Relative differences in the hypothesis of the pipe junction 0D model by Corberan.	102
6.1	Cutaway drawing of four-vanes MRCVC engine.	105
6.2	Basic geometry of a three-vane MRCVC and definition of its main geometric parameters.	106
6.3	Changes of flow domain definition during a ‘stroke’ for a 3-vanes MRCVC engine.	107
6.4	Model of port design 1.	108
6.5	Gas-passage minimum area for intake and exhaust ports (design 1).	109
6.6	State in chamber, in intake port, in exhaust port, and mass flow rate through ports during a cycle at 4000 rpm for port design 1.	110
6.7	Model of port design 2.	110
6.8	Gas-passage area for intake and exhaust ports (design 2).	111
6.9	State in chamber, in intake port, in exhaust port, and mass flow rate through ports during a cycle at 4000 rpm for port design 2.	111
6.10	Performance characteristics for a MRCVC with two port designs.	112
6.11	Mesh quality as a function of the rotation angle.	113
6.12	Element quality field at the initial and final angles of interval 1.	114
6.13	Element quality field at the initial and final angles of interval 2.	114
6.14	Element quality field at the initial and final angles of interval 3.	115
6.15	Element quality field at the initial and final angles of interval 4.	115
6.16	Element quality field at the initial and final angles of interval 5.	115
6.17	Intake and exhaust port geometries for the two-dimensional model.	116
6.18	Magnitude of the flow velocity ([m/s]) in the chambers of the MRCVC with reference chamber at $\theta = 0^\circ$	117
6.19	Magnitude of the flow velocity ([m/s]) in the chambers of the MRCVC with the reference chamber at $\theta = 20^\circ$	117
6.20	Magnitude of the flow velocity ([m/s]) in the chambers of the MRCVC with the reference chamber at $\theta = 40^\circ$	118
6.21	Magnitude of the flow velocity ([m/s]) in the chambers of the MRCVC with the reference chamber at $\theta = 60^\circ$	118
6.22	Magnitude of the flow velocity ([m/s]) in the chambers of the MRCVC with the reference chamber at $\theta = 80^\circ$	119
6.23	Magnitude of the flow velocity ([m/s]) in the chambers of the MRCVC with the reference chamber at $\theta = 100^\circ$	119

6.24	Non-dimensional logarithmic pressure field in the chambers of the MRCVC with the reference chamber at $\theta = 0^\circ$.	120
6.25	Non-dimensional logarithmic pressure field in the chambers of the MRCVC with the reference chamber at $\theta = 20^\circ$.	120
6.26	Non-dimensional logarithmic pressure field in the chambers of the MRCVC with the reference chamber at $\theta = 40^\circ$.	121
6.27	Non-dimensional logarithmic pressure field in the chambers of the MRCVC with the reference chamber at $\theta = 60^\circ$.	121
6.28	Non-dimensional logarithmic pressure field in the chambers of the MRCVC with the reference chamber at $\theta = 80^\circ$.	122
6.29	Non-dimensional logarithmic pressure field in the chambers of the MRCVC with the reference chamber at $\theta = 100^\circ$.	122
B.1	Orthogonal mesh for the reentrant corner problem generated with the $z = \zeta^{3/2}$ mapping.	129
B.2	Comparison of nodes position along the x -axis.	130
B.3	Orthogonal mesh obtained with the proposed method for the reentrant corner. Quadrangle elements along with their diagonals are shown.	131
B.4	Maximum deviation angle at the nodes and deviation angles at the center of the elements for the reentrant corner.	131
B.5	Orthogonal mesh obtained with the proposed method for the transformation of a square into a rhombus. Quadrangle elements along with their diagonals are shown.	132
B.6	Deviation angle for the intersection of quadrangle diagonals for the mesh of 30×30 elements.	133
B.7	Deviation angle for the intersection of quadrangle diagonals for the mesh of 30×30 , 60×60 and 120×120 elements.	133
B.8	Orthogonal mesh obtained with the proposed CMD method for the region outside an ellipse of eccentricity $\epsilon = 0.932$ ($b/a = 0.361$).	134
B.9	Orthogonality deviation (at the diagonal intersection) for the mesh around an ellipse.	134
C.1	Basic geometry of the MRCVC engine.	136
D.1	Número de condición en función de CFL_c para $M = 1 \times 10^{-3}$.	143
D.2	Módulo de la velocidad ([m/s]) en las cámaras del MRCVC.	146
D.3	Performance del motor rotativo MRCVC.	147

List of Tables

2.1	Total number of iterations to reach a relative domain deformation of 90 % for step 2D test.	36
2.2	Total number of iterations to reach a relative domain deformation of 70 % for step 3D test.	37
2.3	Total elapsed computational time for the axisymmetrical flowmeter test.	40
3.1	Comparison of results obtained by using the UP strategy and two solutions of the incompressible Navier-Stokes equations for the channel with a moving indentation test.	61
4.1	Main cylinder data of V10 engine.	79
4.2	Valve data of V10 engine.	79
4.3	Performance characteristics of QUB 400 engine.	82
4.4	Operational variables of KamAZ-7405 diesel engine.	84
4.5	Performance characteristics of KamAZ-7405 diesel engine.	84
6.1	Intake and exhaust manifolds data of MRCVC engine.	108
6.2	Data of meshes generated.	113

Nomenclature

Roman Letters

A	Surface area
\mathbf{A}	Advective jacobians in conservative variables basis
$\tilde{\mathbf{A}}_v$	Advective jacobians in viscous variables basis
c	Sound speed
c_p	Specific heat at constant pressure
c_v	Specific heat at constant volume
C_s	Smagorinsky constant
d	Distance to the wall
D	Diameter
e	Internal energy per unit mass
E	Total energy per unit mass
f	Friction coefficient
\mathbf{f}_e	External forces vector
F	Cross-section area
\mathbf{F}^a	Advective flux vector
\mathbf{F}^d	Diffusive flux vector
G	Specific friction force
h	Element size
	Specific enthalpy
h_f	Film heat transfer coefficient
H_c	Calorific heat content of the fuel
\mathbf{I}	Second order identity tensor
\mathbf{k}	Wave number
\mathbf{K}	Diffusive jacobians in conservative variables basis
$\tilde{\mathbf{K}}_v$	Diffusive jacobians in viscous variables basis
m	Mass
m_{at}	Mass of delivered air trapped

m_b	Mass of burnt gases
m_{inj}	Mass of fuel injected per cycle
m_t	Total mass trapped
n_d	Number of space dimensions
n_{dof}	Number of degrees of freedom
n_{el}	Number of elements
\mathbf{n}	Unit normal vector
N	Engine speed
p	Pressure
p_{crit}	Critic pressure
p_{max}	Peak cylinder pressure
\dot{q}	Rate of heat transfer per unit mass of fluid
\mathbf{q}	Heat flux vector
\dot{Q}_{ch}	Heat release due to combustion
\dot{Q}_{ht}	Heat transfer rate
\mathbf{Q}	Viscous variables vector
R	Gas constant
\tilde{R}	Universal gas constant
S_p	Mean piston speed
\mathbf{s}	Unit vector aligned with \mathbf{u}
\mathbf{S}	Source vector in conservative variables basis
\mathbf{S}_n	Matrix of eigenvectors
$\tilde{\mathbf{S}}_v$	Source vector in viscous variables basis
t	Time
t_d	Ignition delay time
t_f	Final time
t_{ig}	Start of combustion time
t_{inj}	Start of injection time
\mathbf{t}	Unit tangent vector
T	Temperature
\mathbf{T}	Viscous stress tensor
u_f	Friction velocity
\mathbf{u}	Velocity
\mathbf{U}	Conservative variables vector
V	Volume
V_{ref}	Reference volume
\mathbf{w}	Arbitrary mesh velocity
\mathbf{W}	Weight function

x_b	Mass fraction of burnt gases
\mathbf{x}	Spatial coordinates

Greek Letters

γ	Specific heat ratio
Γ	Boundary domain
$\mathbf{\Gamma}$	Preconditioning matrix in conservative variables basis
$\mathbf{\Gamma}_v$	Preconditioning matrix in viscous variables basis
δ	Coefficient of regularization of CMD functional
	Coefficient of the time derivative of pressure
δ_{ij}	Kronecker delta
Δ	Damping function
Δt	Time step
$\Delta\theta$	Duration of combustion
ϵ	Penalization parameter
ϵ	Strain rate tensor
η	Size skewness
η_S	Scavenging efficiency
θ	Crank shaft angle
θ_{ig}	Angle of start of combustion
λ	Second coefficient of viscosity
	Wave speed
λ^0, λ^\pm	Path line, Mach line
λ_i	i -th eigenvalue of the system of governing equations
$\mathbf{\Lambda}$	Eigenvalues matrix
κ	Thermal conductivity
μ	First coefficient of viscosity
μ_{eff}	Dynamic effective viscosity
μ_t	Turbulent dynamic viscosity
ν	Kinematic viscosity
ν_{eff}	Kinematic effective viscosity
ν_t	Turbulent kinematic viscosity
ξ_i	Master element coordinates
$\boldsymbol{\xi}$	Coordinates of the reference configuration
ρ	Density
$\mathbf{\Pi}^\pm$	Projection matrices onto right/left-going characteristic modes
σ	Fourier number $\mu\Delta t/\rho h^2$

σ_v	Scavenge ratio by volume
$\boldsymbol{\sigma}$	Stress tensor
τ	Pseudo-time
τ_w	Viscous shear stress at wall
$\boldsymbol{\tau}_{SGS}$	LES subgrid-scale stress tensor
ϕ	Fuel/air equivalence ratio
ψ	F_T/F_P ratio
ω	Angular speed
Ω	Domain

Abbreviations

ALE	Arbitrary Lagrangian Eulerian
BDC, ABDC, BBDC	Bottom dead center crank position, after BDC, before BDC
CAD	Crank angle degree
CFD	Computational fluid dynamics
CFL	Courant-Friedrichs-Levy CFL number $\max(\lambda_i)\Delta t/h$
CMD	Computational mesh dynamics
CN	Condition number Fuel cetane number
D	Dirichlet
DNS	Direct numerical simulation
DP	Differential predictor
EDC	External dead center crank position
EPC, EPO	Exhaust port closing, opening
EVC, EVO	Exhaust valve closing, opening
FEM	Finite element method
FSI	Fluid-structure interaction
IC	Internal combustion
IDC	Internal dead center crank position
IMEP	Indicate mean pressure
IPC, IPO	Intake port closing, opening
IVC, IVO	Intake valve closing, opening
LES	Large eddy simulation
M	Mach number $\ \mathbf{u}\ /c$
M_r	Reference Mach number
$M_r _{\text{inv}}$	Inviscid reference Mach number
M_ϵ	Cut-off Mach number

MRCVC	<i>Motor rotativo de combustión a volumen constante</i>
N	Neumann
NP	Non-preconditioned
NSI	Incompressible Navier-Stokes
Nu	Nusselt number $h_f L / \kappa$
Pr	Prandtl number $\mu c_p / \kappa$
PSPG	Pressure-Stabilizing/Petrov-Galerkin
Pr_t	Turbulent Prandtl number ν_t / α_t
R	Robin
RANS	Reynolds-average Navier-Stokes
Re	Reynolds number $\rho U L / \mu$
Re_h	Cell Reynolds number $\rho \ \mathbf{u}\ h / \mu$
RHS	Right hand side
S	Smoothing
SFC	Specific fuel consumption
SP	Steady preconditioning
St	Stanton number $Nu / Re Pr$
	Strouhal number L / UT
SUPG	Streamline Upwind/Petrov-Galerkin
TDC, ATDC, BTDC	Top dead center crank position, after TDC, before TDC
ULSAR	Use last state as reference
UP	Unsteady preconditioning
U-S	Untangling and smoothing

Introduction

The Funeral is about to begin, Sir!

Marduk

The increase in the capability of computing in conjunction with the development of new mathematical models and numerical methods, allow to deal with the resolution of complex problems of importance for both science and engineering. Among these, the CFD (Computational Fluid Dynamics) problems in moving domains, such as Fluid-Structure Interaction (FSI) problems, are a topic of particular interest for researchers because of the difficulty that they present and the large number of applications in which these kind of problems are present. One of such problems is the computation of in-cylinder flows in internal combustion (IC) engines.

The modeling of IC engines is a multidisciplinary subject that involves chemical thermodynamics, fluid mechanics, turbulence, heat transfer, combustion, and numerical methods. In this thesis, the focus is placed on some aspects of the computational resolution of the fluid dynamics problem. In particular, the topics addressed are the mesh dynamics problem, the resolution of flows at low Mach numbers, and the coupling of 1D/multi-D domains for compressible flows.

When an Arbitrary Lagrangian Eulerian (ALE) strategy is applied to solve problems with deformable domains, it is necessary to have a Computational Mesh Dynamics (CMD) technique to resolve the dynamics of the mesh. While the movement of the mesh is an artificial field in a FSI problem, its significance is relevant because it affects considerably the efficiency and accuracy of the computation. For in-cylinder flows in IC engines the movement of the boundary domain is known *a priori*. In these cases the domain has a very high relative deformation and even changes on its topology. This demands great robustness from the CMD strategy to avoid an excessive deterioration of the grid quality and to reduce the number of remeshing needed in the whole simulation.

The flow inside of an IC engine is characterized by a low Mach number, except in the early moments in which the exhaust valve (or port) is opened. The numerical methods for compressible flow based on the density fail when they are applied to flows with low Mach numbers, which is due to the bad conditioning of the system of equations. For this reason, it is necessary to apply a technique that allows the resolution of compressible flows in all the range of Mach numbers, especially in the low Mach limit.

Then, to perform a simulation in an IC engine is necessary to have a CFD code able to compute compressible turbulent flows with low (and also relatively high) Mach numbers

in deformable 3D domains [65]. Given the highly complex geometry of the engines and the physical processes that occur within them, it is at present only possible to solve one part of such machines with a 3D model. In this way, and because of its dynamic behavior, another difficulty that appears is related to the boundary conditions to impose to the model. Usually, these problems are addressed by the simulation of the rest of the engine through 0D/1D models, which is achieved in one hand, modeling the entire machine simultaneously (but the level of detail varies depending on the model) and, on the other hand, providing appropriate conditions to the 3D code. Applying the above approximation, the need to couple appropriately the solutions obtained in the computing domains arises, which can be calculated by different codes.

The large spread in length and time scales of in-cylinder flows in IC engines requires a high degree of refinement in the finite element mesh and, then requires very large computational resources. Thus, a parallel code is needed in order to achieve accurate results in that problems. In addition, due to explicit and semi-implicit schemes have demonstrated to be inefficient when they are applied to IC engines [32], a full implicit scheme might be used.

In chapter 1, the equations governing the fluid flow in IC engines, namely the Navier-Stokes equations for compressible flows, are presented. This system of equations is simplified by neglecting radiation and considering a single component, because of the transport of species and combustion are not the goal of this thesis. The fluid is considered to be an ideal gas. Since it is impossible with present processors to resolve all the time and length scales of the flows in IC engines, turbulence models are required. In IC engines, turbulence relaxation times are of the same order as cycle times [66], and long-time-averaging methods are not strictly applicable. The Large Eddy Simulation model is used in this thesis, which is briefly described in chapter 1. Also in this chapter, the variational formulation for the Finite Element Method (FEM) utilized and the numerical strategies to solve the boundary conditions are included.

Chapter 2 deals with the CMD problem. Some definitions regarding element and mesh quality metrics are presented. The proposal of a CMD strategy based on the minimization of the mesh distortion is formulated and enhanced with a linear predictor of the solution. The mesh topology is assumed to remain constant along the mesh deformation. Several test cases are solved including 2D and 3D CMD problems, which show the robustness of the technique. Being discontinuous the proposed functional, its regularization leads to a simultaneous mesh untangling and smoothing method. This method is useful as a CMD technique which does not impose any condition on the time step used in the simulation since, for a given time, the mesh depends on its topology and on the boundary position only. Some benchmark problems are solved using the simultaneous untangling and smoothing technique and, also, 2D and 3D geometries of IC engines chambers are assessed.

The numerical method employed to solve compressible flows at low Mach numbers is discussed in chapter 3. Due to the Mach numbers of the flow in IC engines could range from very low values (for instance, at the Top Dead Center (TDC) in a reciprocating engine) to transonic values (when the exhaust port is opened), the method of preconditioning of the equations together with the dual time stepping technique is used. The preconditioning matrix utilized was originally designed by Choi and Merkle [16] to solve steady compressible flows using the Finite Volume Method. An eigenvalue analysis of the preconditioned system of equations is performed to define some parameters in the

preconditioning matrix. The chapter contains also the Finite Element variational formulation used, where the stabilization coefficients are properly defined for the preconditioned system. Steady and unsteady nearly incompressible tests are solved and compared with solutions of the Navier-Stokes equations for incompressible flows. In addition, the results of the flow within an opposite-piston engine under cold conditions are presented.

In chapter 4, some thermodynamic and gas-dynamic based models used to describe IC engine operating characteristics are reviewed. These models were implemented in a computational code which is useful to compute the global performance of an IC engine and, furthermore, it could be applied as a generator of boundary conditions for multi-D models. Implicit time discretization of the differential equations involved in the models is an option available in the code. The chapter contains, also, a brief description of the code implementation and some results for different type of IC engines.

Chapter 5 presents two strategies of coupling 1D/multi-D domains for compressible flows. One of them is based on absorbing boundary conditions, and the other one is simply a constraint to the state at the coupling interface assuming that the problem is solved in a ‘monolithic’ way. Results of 1D/1D, 1D/2D and 1D/3D couplings are presented, which were solved as a ‘monolithic’ system.

The models and techniques presented in chapters 1 to 5 were applied to simulate a particular engine, the MRCVC [80] (*Motor Rotativo de Combustión a Volumen Constante*). The results are summarized in chapter 6.

The base code utilized in this thesis is PETSc-FEM [73]. PETSc-FEM is a general purpose, parallel, multi-physics FEM program for CFD applications based on PETSc [9]. PETSc-FEM comprises both a library that allows the user to develop FEM (or FEM-like, *i.e.* non-structured mesh oriented) programs, and a suite of application programs. The computational tools developed during the thesis were implemented and tested as new applications of the PETSc-FEM code.

Introducción

El incremento en la capacidad de cálculo en conjunto con el desarrollo de nuevos modelos matemáticos y métodos numéricos, permite afrontar la resolución de problemas complejos de importancia tanto científica como ingenieril. Entre éstos, los problemas CFD (por *Computational Fluid Dynamics*) en dominios móviles, tales como los problemas de interacción fluido-estructura (FSI, por *Fluid-Structure Interaction*), constituyen un tópico de especial importancia para los investigadores debido a la dificultad que presentan y al gran número de aplicaciones en las cuales puede hallarse esta clase de problemas. Uno de tales problemas consiste en la resolución del flujo de fluidos en el interior de motores de combustión interna (IC, por *Internal Combustion*).

La modelación de motores IC es una tarea multidisciplinaria que involucra termoquímica, fluidodinámica, turbulencia, transferencia de calor, combustión y métodos numéricos. Esta tesis se enfoca en algunos aspectos de la resolución computacional del problema fluidodinámico. En particular, los temas abordados consisten en el problema de la dinámica de la malla, la resolución de flujos a bajos números de Mach, y el acoplamiento de dominios 1D/multi-D para flujos compresibles.

Aplicando una estrategia tipo ALE (por *Arbitrary Lagrangian Eulerian*) para resolver esta clase de problemas con dominios deformables, es necesario contar con una técnica CMD (por *Computational Mesh Dynamics*) para resolver la dinámica de la malla. Si bien el movimiento de la malla es un campo artificial en un problema FSI, su importancia es relevante debido a que afecta considerablemente la eficiencia y precisión del cálculo. En la resolución de flujos dentro de motores IC, el movimiento de la frontera del dominio es conocido *a priori*. En estos casos, el dominio de flujo experimenta una elevada deformación relativa con cambios en su topología. Esto demanda de la estrategia CMD una gran robustez a fin de evitar el deterioro excesivo de la calidad de la grilla y reducir el número de remallados en la simulación.

El flujo dentro de un motor IC se caracteriza por presentar en general un número de Mach bajo, excepto en los primeros instantes en que se abre la válvula o lumbrera de escape. Los métodos numéricos para flujo compresible basados en la densidad pueden fallar cuando se los aplica a flujos con bajo número de Mach, lo cual se debe al mal condicionamiento del sistema de ecuaciones. Por esta razón, resulta necesario aplicar una técnica que permita la resolución de flujos compresibles en todo el rango de números de Mach, especialmente en el límite de bajo Mach.

Para realizar la simulación en un motor IC, es necesario entonces disponer de un código CFD capaz de computar flujos compresibles turbulentos a bajos (y también relativamente altos) números de Mach en dominios tridimensionales deformables [65]. Dada la alta complejidad geométrica de los motores y de los procesos físicos que ocurren dentro de ellos, sólo es posible resolver una parte de tales máquinas con un modelo 3D. De este

modo, y por tratarse de un problema dinámico, otra dificultad adicional la presentan las condiciones de borde a imponer a dicho modelo. Usualmente estos problemas son abordados simulando el resto del motor mediante un simulador de motores 0D/1D, con lo cual se logra, por un lado, modelar toda la máquina simultáneamente (aunque con un nivel de detalle variable según el modelo) y, por otro, proveer de condiciones de contorno apropiadas al código 3D. Aplicando la referida aproximación, surge la necesidad de acoplar adecuadamente las soluciones obtenidas en los distintos dominios computacionales, las cuales pueden incluso ser calculadas por distintos códigos.

El amplio espectro en escalas de longitud y tiempo de los flujos dentro de motores IC requiere un elevado refinamiento de la malla de elementos finitos y, por lo tanto, demanda una elevada cantidad de recursos computacionales. Luego, es necesario un código paralelo a fin de obtener resultados precisos en tales problemas. Además, debido a que los esquemas explícitos y semi-implícitos han demostrado ser ineficientes cuando se los aplica a motores IC [32], se propone el empleo de un esquema totalmente implícito.

En el capítulo 1 se presentan las ecuaciones que gobiernan el flujo de fluidos en un motor IC, precisando, las ecuaciones de Navier-Stokes para flujos compresibles. Este sistema de ecuaciones es simplificado despreciando la radiación y considerando un único componente, dado que el transporte de especies y la combustión no son el objetivo de estudio en esta tesis. Además, el fluido es considerado un gas ideal. Debido a que es imposible con los procesadores actuales resolver todas las escalas de tiempo y longitud de los flujos en motores IC, resulta necesario modelar la turbulencia. En motores IC, los tiempos de relajación turbulentos son del mismo orden que los tiempos del ciclo [66], y los métodos de promediado en el tiempo no son estrictamente aplicables. El modelo de Simulación de Grandes Torbellinos (*Large Eddy Simulation*) es el utilizado en la presente tesis, el cual se describe brevemente en el capítulo 1. La formulación variacional para el Método de Elementos Finitos empleada y las estrategias numéricas para resolver las condiciones de contorno se incluyen también en este capítulo.

El capítulo 2 trata acerca del problema de la dinámica de la malla. Se presentan algunas definiciones relativas a métricas de calidad elemental y de malla. Se formula una estrategia CMD basada en la minimización de la calidad de la malla y mejorada con un predictor lineal de la solución, donde se asume que la topología de la malla permanece constante a lo largo de toda la deformación de la misma. Varios casos *test* son resueltos, incluyendo problemas CMD en 2D y 3D, los cuales muestran la robustez de la técnica. Dado que el funcional propuesto es discontinuo, su regularización permite obtener un método de *untagling* y *smoothing* simultáneos. Este método resulta útil como una técnica CMD que no impone ninguna condición sobre el paso de tiempo utilizado en la simulación debido a que, para un instante dado, la malla depende sólo de su topología y de la posición de la frontera. Son resueltos algunos *benchmarks* y también geometrías 2D y 3D de cámaras de motores IC aplicando la referida estrategia de *untagling* y *smoothing* simultáneos.

El método utilizado para resolver flujos compresibles a bajos números de Mach se discute en el capítulo 3. En los motores IC el número de Mach puede variar en un amplio rango, desde valores muy bajos (en, por ejemplo, el Punto Muerto Superior en un motor alternativo) hasta valores transónicos (cuando la válvula de escape se abre). Por lo tanto, son aplicados el método de preconditionamiento de las ecuaciones junto con la técnica de doble tiempo. La matriz de preconditionamiento utilizada fue originalmente diseñada por Choi y Merkle [16] para la resolución de flujos compresibles estacionarios. Se realiza

un análisis de autovalores del sistema de ecuaciones preconditionado, a partir del cual surge la redefinición de algunos parámetros del preconditionador original. El capítulo contiene la formulación variacional empleada, donde los coeficientes de estabilización son apropiadamente definidos para el sistema preconditionado. Son resueltos algunos problemas incompresibles estacionarios y no estacionarios, comparándoselos con soluciones de las ecuaciones de Navier-Stokes para flujo incompresible. Además, se presentan los resultados del flujo dentro de un motor de pistones opuestos funcionando en frío.

En el capítulo 4 se reveen algunos modelos termodinámicos y gasdinámicos usados para describir las características de operación de motores IC. Estos modelos fueron implementados en un código computacional que resulta útil para calcular la *performance* global de un motor IC y, además, que podría ser aplicado como un generador de condiciones de contorno para modelos multidimensionales. Una de las opciones disponibles en el código es la posibilidad de aplicar una discretización temporal implícita de las ecuaciones diferenciales involucradas en los modelos. El capítulo contiene también una breve descripción del código y algunos resultados para diferentes tipos de motores IC.

El capítulo 5 presenta dos estrategias de acoplamiento de dominios 1D/multi-D para flujos compresibles. Una de estas estrategias se basa en la utilización de condiciones de contorno absorbentes, mientras que la otra consiste simplemente en la restricción del estado sobre la superficie de acople asumiendo que el problema es resuelto en forma ‘monolítica’. Se presentan resultados de acoplamientos 1D/1D, 1D/2D y 1D/3D resueltos en forma ‘monolítica’.

Los modelos y técnicas presentadas en los capítulos 1 a 5 se aplicaron a la simulación de un MRCVC [80] (Motor Rotativo de Combustión a Volumen Constante). Los resultados obtenidos se incluyen en el capítulo 6.

El código base utilizado en esta tesis es PETSc-FEM [73]. PETSc-FEM es un programa FEM de propósito general, paralelo y multifísica para aplicaciones CFD basado en PETSc [9]. PETSc-FEM se compone de una biblioteca que permite al usuario desarrollar programas FEM y de un conjunto de códigos para aplicaciones diversas. Las herramientas computacionales desarrolladas durante la tesis fueron implementadas y validadas como nuevas aplicaciones de este código.

Chapter 1

Governing equations and numerical approximation

*O'Nightspirit
I am one with thee
I am the eternal power
I am the Emperor
Emperor*

1.1 Governing equations

The flow field in an internal combustion engine model is governed by the instantaneous time-dependent three-dimensional conservation equations of mass, momentum and energy. These equations can be simplified by neglecting radiation [66] and, in the particular case of this thesis, a single component is considered.

Let $\Omega \subset \mathbb{R}^{n_d}$ the spatial domain and $(0, t_f)$ the temporal domain, where n_d is the number of space dimensions, and let Γ the boundary of Ω . The spatial and temporal coordinates are denoted by \mathbf{x} and t , respectively.

The Navier-Stokes equations governing the fluid flow, in conservation form, are

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0 \quad \text{on } \Omega \times (0, t_f) \\ \frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) + \nabla p - \nabla \cdot \mathbf{T} &= \rho \mathbf{f}_e \quad \text{on } \Omega \times (0, t_f) \\ \frac{\partial(\rho E)}{\partial t} + \nabla \cdot (\rho E \mathbf{u}) + \nabla \cdot (p \mathbf{u}) - \nabla \cdot (\mathbf{T} \mathbf{u}) + \nabla \cdot \mathbf{q} &= \rho \mathbf{f}_e \cdot \mathbf{u} \quad \text{on } \Omega \times (0, t_f) \end{aligned} \quad (1.1)$$

where ρ , \mathbf{u} , p , \mathbf{T} , E and \mathbf{q} are the density, the velocity, the pressure, the viscous stress tensor, the total energy per unit mass, and the heat flux vector, respectively, and \mathbf{f}_e are external forces. Generally, these forces are null in IC engine simulation. It is assumed

a perfect gas constitutive relation and a Newtonian fluid defined by the two viscosity coefficients λ and μ . Thus, the viscous stress tensor is defined as

$$\mathbf{T} = \mu((\nabla \mathbf{u}) + (\nabla \mathbf{u})^T) + \lambda(\nabla \cdot \mathbf{u})\mathbf{I} = 2\mu\boldsymbol{\epsilon}(\mathbf{u}) + \lambda(\nabla \cdot \mathbf{u})\mathbf{I} \quad (1.2)$$

\mathbf{I} being the second order identity tensor, $\boldsymbol{\epsilon}(\mathbf{u}) = \frac{1}{2}((\nabla \mathbf{u}) + (\nabla \mathbf{u})^T)$ is the strain rate tensor and superscript T denotes transpose.

In addition, it is considered the range of fluid behavior within local thermodynamic equilibrium for which the Stokes relation $3\lambda + 2\mu = 0$ is valid.

Pressure is related to the other variables via the equation of state. For ideal gases, this equation has the form

$$p = (\gamma - 1)\rho e \quad (1.3)$$

where γ is the ratio of specific heats, and e is the internal energy per unit mass which is related to the total energy per unit mass and kinetic energy as

$$e = E - \frac{1}{2}\|\mathbf{u}\|^2 \quad (1.4)$$

The heat flux vector is defined as

$$\mathbf{q} = -\kappa \nabla T \quad (1.5)$$

where κ is the heat conductivity and T is the temperature. In the particular case of ideal gases, the following relations hold

$$\begin{aligned} T &= \frac{e}{c_v} \\ c_v &= \frac{R}{\gamma - 1} \\ c_p &= \frac{\gamma R}{\gamma - 1} \end{aligned} \quad (1.6)$$

where c_v is the specific heat of the fluid at constant volume, c_p is the specific heat of the fluid at constant pressure, and R is the ideal gas constant. Prandtl number (Pr) relates the heat conductivity of the fluid to its viscosity according to the following relation

$$\kappa = \frac{\mu c_p}{Pr} \quad (1.7)$$

The governing equations (1.1) can be written in compact form as [33]

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_i^a}{\partial x_i} = \frac{\partial \mathbf{F}_i^d}{\partial x_i} + \mathbf{S} \quad \text{on } \Omega \times (0, t_f) \quad (1.8)$$

where $\mathbf{U} = [\rho, \rho \mathbf{u}, \rho E]^T$ is the vector of conservative variables, $\mathbf{S} = [0, \rho \mathbf{f}_e, \rho \mathbf{f}_e \cdot \mathbf{u}]^T$ is the source vector, \mathbf{F}^a and \mathbf{F}^d are the advective (or inviscid) and viscous flux vectors respectively, defined as

$$\mathbf{F}_i^a = \begin{bmatrix} \rho u_i \\ \rho u_1 u_i + \delta_{i1} p \\ \rho u_2 u_i + \delta_{i2} p \\ \rho u_3 u_i + \delta_{i3} p \\ (\rho E + p) u_i \end{bmatrix} \quad (1.9)$$

$$\mathbf{F}_i^d = \begin{bmatrix} 0 \\ T_{i1} \\ T_{i2} \\ T_{i3} \\ T_{ik}u_k - q_i \end{bmatrix} \quad (1.10)$$

Here, u_i and q_i are the components of the velocity and heat flux vectors, respectively, T_{ik} are the components of the viscous stress tensor, and δ_{ij} is the Kronecker delta. The flux vectors in equations (1.9) and (1.10) are expressed for the 3D case.

In the quasi-linear form, equation (1.8) is written as [33]

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_i \frac{\partial \mathbf{U}}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial x_j} \right) + \mathbf{S} \quad \text{on } \Omega \times (0, t_f) \quad (1.11)$$

where

$$\mathbf{A}_i = \frac{\partial \mathbf{F}_i^a}{\partial \mathbf{U}} \quad (1.12)$$

is the advective jacobian matrix, and \mathbf{K}_{ij} is the diffusivity matrix satisfying

$$\mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial x_j} = \mathbf{F}_i^d \quad (1.13)$$

1.1.1 Turbulence modeling

The flow field in an internal combustion engine is turbulent and comprises many time and length scales. The ratio between the time (length) scales of the Kolmogorov and energy-containing eddies is of the order of $Re_l^{-1/2}$ ($Re_l^{-3/4}$), where Re_l is the turbulent Reynolds number [74]. In an IC engine cylinder, it is expected the most energetic scale to be of the order of 1/6 of the largest eddy size (the cylinder bore if the piston is halfway down on the intake stroke, or the clearance height if the piston is near TDC [46]). Furthermore, the turbulent velocity when the piston is in the middle of the intake stroke is about 10 times the mean piston speed (S_p) and at TDC, in the absence of swirl, tumble and squish, is about a half of S_p [46]. Let assume typical values of $S_p = 5$ m/s, a cylinder bore of 80 mm, a geometric compression ratio of 8:1, inlet density and temperature of 1 kg/m³ and 300 K, respectively, and an inlet dynamic viscosity of 2×10^{-5} Pa.s. At the two states considered, halfway down on the intake stroke and at TDC, the turbulent Reynolds numbers are respectively $Re_l \approx 3.33 \times 10^4$ and $Re_l \approx 1.26 \times 10^3$ ¹. Then, it is practically impossible with available processors to obtain a numerical solution of equations (1.1) that accounts for all the turbulent time and length scales and, therefore, models need to be introduced. There are two main approaches to model turbulence, the so-called RANS (Reynolds-Average Navier-Stokes) approach, and the LES (Large Eddy Simulation) models. In RANS methods, a set of partial differential equations describing suitable averaged quantities are used everywhere in the flow. For periodic engine flows, ensemble or phase average is replaced instead typical time averaging [32, 66]. Since the flow during the engine cycle is compressed and expanded, mass-weighted averaging

¹The dynamic viscosity in a gas is proportional approximately to \sqrt{T} [87]. In addition, isentropic relations were considered for compression.

(called Favre averaging) is commonly applied in conjunction with ensemble averaging [32]. LES is an approach in which the large-scale three-dimensional time-dependent turbulence structure is calculated in a single realization of the flow. Thus, only the small-scale turbulence need to be modeled, which is more isotropic than the large-scale structure [89].

Although the RANS methods involve a greater number of equations than LES strategies, they are cheaper since they can work with coarser meshes and time steps. In addition, the LES methods could need an *a posteriori* statistical analysis in order to compute turbulent variables. Nevertheless, RANS methods have demonstrated their inability to produce solutions for the Navier-Stokes equations [47]. This is obviously a very serious shortcoming of any turbulence modeling procedure, and although it has been recognized for a long time by theorists, especially mathematicians, it has had little, if any, impact on engineering analyzes of turbulence. In contrast to RANS, it can be shown that LES procedures generally converge to DNS (Direct Numerical Simulation) as discretization step sizes are refined. This occur since models for LES subgrid-scale stress tensor ($\boldsymbol{\tau}_{SGS}$) are generally constructed such that $\boldsymbol{\tau}_{SGS} \rightarrow 0$ as the discretization step size tends to zero [68]. Hence, the use of a LES flow solver is desirable.

In this thesis, the simplest Smagorinsky model [89, 69] will be applied, which takes the Smagorinsky coefficient as constant, in contrast with the dynamic counterpart proposed by Germano [28]. This is one of the most popular choices into the LES family of turbulence models. In this eddy viscosity model, the turbulent dynamic viscosity is defined as

$$\mu_t = \rho(C_s h)^2 \Delta \sqrt{2\boldsymbol{\epsilon}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{u})} \quad (1.14)$$

where $C_s = 0.1 - 0.2$ is the Smagorinsky constant, Δ is a damping function to reduce the amount of turbulent viscosity in the vicinity of solid objects immersed in the fluid flow, and h is the grid size (a parameter that divides the size of vortices being resolved by the size of vortices being modeled). Finally, $\sqrt{\boldsymbol{\epsilon}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{u})}$ represents the trace of the strain rate tensor making the eddy viscosity a local parameter.

1.1.2 Boundary conditions

At solid walls, the ‘classical’ approach to impose boundary conditions to the compressible Navier-Stokes equations is the *no-slip* condition [33]. For the velocity, this condition is expressed by

$$\mathbf{u} = \mathbf{u}_{\text{wall}} \quad (1.15)$$

where \mathbf{u}_{wall} is the velocity of the wall in the considered reference system.

For the temperature, either the wall temperature (T_{wall}) is fixed

$$T = T_{\text{wall}} \quad (1.16)$$

or the heat flux is determined by the physical conditions, that is

$$-\kappa \frac{\partial T}{\partial n} = q_{\text{wall}} \quad (1.17)$$

where q_{wall} is the wall heat flux, and n refers to the normal direction to the wall. The last boundary condition at the wall could be obtained projecting the momentum equation on the normal direction

$$\frac{\partial p}{\partial n} = (\boldsymbol{\nabla} \cdot \mathbf{T})_n \quad (1.18)$$

For thin shear layers at high Reynolds numbers, this might be replaced by the boundary layer approximation [33]

$$\frac{\partial p}{\partial n} = 0 \quad (1.19)$$

which was used for numerical simulations of internal combustion engines [66] by several researchers.

For turbulent flows, no-slip boundary condition could be a mistake if the mesh is not refined enough at boundary layers. In addition, most of the turbulence models that have been used to calculate the flow field in reciprocating and rotary engines do not account for low-Reynolds number effects, preferential dissipation, and streamline curvature. Therefore, these models cannot be applied up to the solid walls, and the boundary conditions are applied close to the wall but not on it. These boundary conditions are analogous to those used in statistically stationary, incompressible, turbulent flows along flat plates in the absence of pressure gradients. The flows in reciprocating and rotary engines are neither steady nor incompressible, but involve recirculation zones and pressure gradients. Nevertheless, boundary conditions based on solutions to flat plates are frequently used.

For the velocity, the boundary conditions near a solid wall with $\mathbf{u}_{\text{wall}} = \mathbf{0}$ can be written as follows [66]

$$\begin{aligned} \mathbf{u} \cdot \mathbf{n} &= 0 \\ \mathbf{u} \cdot \mathbf{t} &= \begin{cases} \frac{u_f^2 d}{\nu} & \text{if } 0 \leq \frac{u_f d}{\nu} \leq 11.63 \\ u_f \left(2.5 \ln \frac{u_f d}{\nu} + 5.5 \right) & \text{if } \frac{u_f d}{\nu} > 11.63 \end{cases} \end{aligned} \quad (1.20)$$

where \mathbf{n} and \mathbf{t} are the unit vectors normal and tangential to the solid wall, respectively; $u_f = (\tau_w/\rho)^{1/2}$ is the friction velocity, with τ_w the shear stress at the wall; d is the distance from the solid wall to the point closest to the wall; and ν the gas kinematic viscosity.

The boundary condition for the temperature profile near the wall in a turbulent flow is frequently imposed by means of the Reynolds analogy between linear momentum and energy [67]. The Reynolds analogy is strictly applicable when the Reynolds fluxes of linear momentum and energy are equal, and the profiles of the mean velocity and enthalpy are similar, that is, the laminar and turbulent Prandtl numbers are close to 1. This boundary condition is expressed as [66]

$$T = \begin{cases} T_{\text{wall}} - \frac{Pr_t q_{\text{wall}} d}{\rho c_p \nu} & \text{if } 0 \leq \frac{u_f d}{\nu} \leq 11.63 \\ T_{\text{wall}} - \frac{Pr_t q_{\text{wall}}}{\rho c_p u_f} \left(2.5 \ln \frac{u_f d}{\nu} + 5.5 \right) & \text{if } \frac{u_f d}{\nu} > 11.63 \end{cases} \quad (1.21)$$

where $Pr_t = \frac{\nu_t}{\alpha_t}$ is the turbulent Prandtl number, $\nu_t = \mu_t/\rho$ and α_t being, respectively, the turbulent kinematic viscosity and the turbulent thermal diffusivity. The analogy between linear momentum and energy may not be applicable to reciprocating and rotary engine flows. Thus, equation (1.21) should be used carefully as it was noted by Ramos [66]

Accurate heat transfer results can be obtained only if the hydrodynamic and thermal boundary layers are resolved and the boundary conditions are applied

at the walls. These accurate calculations can be performed only if enough grid points are placed in the boundary layer to resolve its structure and if low-Reynolds-number effects are introduced in the turbulence model. [...] Accurate calculations of the heat transfer to solid walls are essential for determining the engine efficiency.

Conditions for inlet and outlet boundaries are applied following the approach proposed by Storti *et al.* [72]. Considering a point on an inlet/outlet boundary, it is possible to do a simplified 1D analysis in the normal direction to the local boundary. The projection matrices onto the right/left-going characteristics modes are defined as

$$\mathbf{\Pi}_n^\pm = \mathbf{S}_n \mathbf{\Pi}_{V_n}^\pm \mathbf{S}_n^{-1} \quad (1.22)$$

where \mathbf{S}_n is the matrix of eigenvectors diagonalizing the projected system, being $\mathbf{\Lambda}_n = \text{diag}[(\lambda_n)_j]$ their respective eigenvalues; and

$$\begin{aligned} (\mathbf{\Pi}_{V_n}^-)_{jk} &= \begin{cases} 1 & \text{if } j = k \text{ and } (\lambda_n)_j < 0 \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{\Pi}_{V_n}^- + \mathbf{\Pi}_{V_n}^+ &= \mathbf{I} \end{aligned} \quad (1.23)$$

Then, the boundary condition is applied as a constraint to the system of governing equations as follows

$$\mathbf{\Pi}_n^-(\hat{\mathbf{U}})(\mathbf{U} - \hat{\mathbf{U}}) = \mathbf{0} \quad (1.24)$$

where $\hat{\mathbf{U}}$ is defined depending on whether the boundary is either an inlet or an outlet. Note that in equation (1.24) the projection matrix, which is a non linear function of the fluid state, is evaluated at the state $\hat{\mathbf{U}}$. This is true if it is assumed that the flow is composed of small perturbations around the state $\hat{\mathbf{U}}$. However, as long as the fluid state departs from the $\hat{\mathbf{U}}$ value, the condition becomes less and less absorbing.

1.1.3 Arbitrary Lagrangian Eulerian description of governing equations

It is well known that there are two viewpoints mostly used in the description of the flow motion equations: one is called the Lagrangian approach, where the observer moves with the fluid velocity, and the other option is the Eulerian approach, in which the observer is fixed. The Arbitrary Lagrangian Eulerian (ALE) description is a generalization of these approaches, where two configurations of the system are considered: an instantaneous configuration $\Omega_t(\mathbf{x})$ and a reference configuration $\Omega_0(\boldsymbol{\xi})$. Then, a mapping function between $\Omega_t(\mathbf{x})$ and $\Omega_0(\boldsymbol{\xi})$ is defined as $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi}, t)$. The ALE strategy applied in this thesis for the resolution of moving domain problems was proposed by Donea *et al.* [22]. In the conservation form, the governing equations can be written as

$$\frac{\partial(J\mathbf{U})}{\partial t} + J \frac{\partial}{\partial x_i} (\mathbf{F}_i^a - \mathbf{F}_i^d - w_i \mathbf{U}) = J\mathbf{S} \quad \text{on } \Omega_t \times (0, t_f) \quad (1.25)$$

where $J = \det \left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \right)$, $\mathbf{w} = [w_1, w_2, w_3]^T = \left. \frac{d\mathbf{x}}{dt} \right|_{\boldsymbol{\xi}}$ and $\Omega_t = \Omega_t(\mathbf{x})$.

Using this ALE strategy, it can be shown (see, for example, the works by Donea *et al.* [22], Nomura [59], etc.) that the system of Navier-Stokes in its quasi-linear form can be written as

$$\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{A}_i - w_i \mathbf{I}) \frac{\partial \mathbf{U}}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial x_j} \right) + \mathbf{S} \quad \text{on } \Omega_t \times (0, t_f) \quad (1.26)$$

or

$$\frac{1}{J} \frac{\partial (J\mathbf{U})}{\partial t} + (\mathbf{A}_i - w_i \mathbf{I}) \frac{\partial \mathbf{U}}{\partial x_i} - \frac{\partial w_i}{\partial x_i} \mathbf{U} = \frac{\partial}{\partial x_i} \left(\mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial x_j} \right) + \mathbf{S} \quad \text{on } \Omega_t \times (0, t_f) \quad (1.27)$$

1.2 Numerical implementation

1.2.1 Finite element formulation

In this section, the variational formulation of the Navier-Stokes equations for compressible flows is presented. The Finite Element Method stabilized by means of the Streamline Upwind/Petrov-Galerkin (SUPG) strategy and with the addition of a shock capturing operator is used. Consider a finite element discretization of the domain Ω into n_{el} subdomains Ω^e , $e = 1, 2, \dots, n_{\text{el}}$. Based on this discretization, the finite element function spaces for the trial solutions and for the weighting functions, \mathcal{S}^h and \mathcal{V}^h respectively, can be defined (see equation (1.29)).

Then, the finite element formulation of the problem (1.11) using SUPG is written as follows [57]:

Find $\mathbf{U}^h \in \mathcal{S}^h$ such that $\forall \mathbf{W}^h \in \mathcal{V}^h$

$$\begin{aligned} & \int_{\Omega} \mathbf{W}^h \cdot \left(\frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} \right) d\Omega + \int_{\Omega} \frac{\partial \mathbf{W}^h}{\partial x_i} \cdot \mathbf{K}_{ij}^h \frac{\partial \mathbf{U}^h}{\partial x_j} d\Omega \\ & + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \boldsymbol{\tau} (\mathbf{A}_k^h)^T \frac{\partial \mathbf{W}^h}{\partial x_k} \cdot \left[\frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} - \frac{\partial}{\partial x_i} \left(\mathbf{K}_{ij}^h \frac{\partial \mathbf{U}^h}{\partial x_j} \right) - \mathbf{S} \right] d\Omega^e \\ & + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \delta_{\text{sc}} \frac{\partial \mathbf{W}^h}{\partial x_i} \cdot \frac{\partial \mathbf{U}^h}{\partial x_i} d\Omega^e = \int_{\Omega} \mathbf{W}^h \cdot \mathbf{S} d\Omega + \int_{\Gamma^h} \mathbf{W}^h \cdot \mathbf{f} d\Gamma \end{aligned} \quad (1.28)$$

where

$$\begin{aligned} \mathcal{S}^h &= \{ \mathbf{U}^h | \mathbf{U}^h \in [\mathbf{H}^{1h}(\Omega)]^{n_{\text{dof}}}, \mathbf{U}^h|_{\Omega^e} \in [P^1(\Omega^e)]^{n_{\text{dof}}}, \mathbf{U}^h = \mathbf{g} \quad \text{on } \Gamma^g \} \\ \mathcal{V}^h &= \{ \mathbf{W}^h | \mathbf{W}^h \in [\mathbf{H}^{1h}(\Omega)]^{n_{\text{dof}}}, \mathbf{W}^h|_{\Omega^e} \in [P^1(\Omega^e)]^{n_{\text{dof}}}, \mathbf{W}^h = \mathbf{0} \quad \text{on } \Gamma^g \} \end{aligned} \quad (1.29)$$

$\mathbf{H}^{1h}(\Omega)$ being the finite dimensional Sobolev functional space over Ω , and with \mathbf{f} and \mathbf{g} representing the natural and Dirichlet boundary conditions vectors, respectively. Γ^g and Γ^h are the portion of the boundary with Dirichlet and Neumann conditions, respectively.

The first summation of element level integrals in equation (1.28) are added to the variational formulation to stabilize the computations against numerical instabilities. In the advection-dominated range, these terms prevent the node-to-node oscillations of the flow variables, where $\boldsymbol{\tau}$ is known as the intrinsic time tensor. The second summation of element level integrals in equation (1.28) are the shock capturing terms that stabilize

the computations in the presence of sharp gradients, δ_{sc} being the coefficient of shock capturing. Matrix $\boldsymbol{\tau}$ is defined by Aliabadi *et al.* [1] in the following way

$$\boldsymbol{\tau} = \max[\mathbf{0}, \boldsymbol{\tau}_a - \boldsymbol{\tau}_d - \boldsymbol{\tau}_\delta] \quad (1.30)$$

where

$$\begin{aligned} \boldsymbol{\tau}_a &= \frac{h}{2(c + \|\mathbf{u}\|)} \mathbf{I} \\ \boldsymbol{\tau}_d &= \frac{\sum_{j=1}^{n_d} \beta_j^2 \text{diag}(\mathbf{K}_{jj})}{(c + \|\mathbf{u}\|)^2} \mathbf{I} \\ \boldsymbol{\tau}_\delta &= \frac{\delta_{\text{sc}}}{(c + \|\mathbf{u}\|)^2} \mathbf{I} \end{aligned} \quad (1.31)$$

Here, $c = \sqrt{\gamma RT}$ is the sonic speed, h is the element size computed as the element length in the direction of the streamline

$$h = \frac{2}{\sum_{a=1}^{n_{\text{en}}} \|\mathbf{s} \cdot \nabla N_a\|} \quad (1.32)$$

N_a being the trial function associated with the node a , n_{en} the number of nodes in the element, \mathbf{s} a unit normalized velocity vector, and $\beta = \nabla \|\mathbf{U}\|^2 / \|\nabla \|\mathbf{U}\|^2\|$.

Regarding the shock capturing term, an isotropic operator proposed by Tezduyar and Senga [75] is presented here. Let $\mathbf{j} = \nabla \rho^h / \|\nabla \rho^h\|$ a unit vector oriented with the density gradient and the characteristic length $h_{JGN} = 2(\sum_{a=1}^{n_{\text{en}}} \|\mathbf{j} \cdot \nabla N_a\|)^{-1}$. The isotropic shock capturing factor included in equation (1.28) is then defined as

$$\delta_{\text{sc}} = \frac{h_{JGN}}{2} u_{\text{char}} \left(\frac{\|\nabla \rho^h\| h_{JGN}}{\rho_{\text{ref}}} \right)^{\beta^*} \quad (1.33)$$

where $u_{\text{char}} = \|\mathbf{u}\| + c$ is the characteristic velocity defined as the addition of the flow velocity magnitude and the sonic speed, ρ_{ref} is the density interpolated at gaussian point, and β^* is a parameter that could be taken as 1 or 2 according to the sharpness of the discontinuity to be captured [75].

Comparing (1.11) with (1.26), in the ALE formulation of Navier-Stokes equations only the advective jacobian are modified. Thus, the finite element formulation stabilized with SUPG of equation (1.26) is obtained by replacing \mathbf{A}_i by $\mathbf{A}_i - w_i \mathbf{I}$ in (1.28), and in the definition of stabilization coefficients (1.31) $\|\mathbf{u}\|$ by $\|\mathbf{u} - \mathbf{w}\|$. In the case of moving domains, care should be taken with integrals containing time derivatives due to the integration domain is a function of t .

1.2.2 Time discretization

Derivatives with respect to time are discretized using the trapezoidal difference scheme, expressed as

$$\frac{\partial \mathbf{U}}{\partial t} \approx \frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\vartheta \Delta t} \quad (1.34)$$

where ϑ is the implicitness parameter, Δt is the time step, and the superscripts n and $n + 1$ indicate the level time t and $t + \Delta t$, respectively.

Furthermore an implicit scheme is used, in such a way that all variables in equation (1.28) are evaluated at the $n + \vartheta$ level time, *i.e.* with the state vector $\mathbf{U}^\vartheta = \vartheta \mathbf{U}^{n+1} + (1 - \vartheta) \mathbf{U}^n$.

1.2.3 Dynamic boundary conditions using Lagrange multipliers

Boundary conditions at inlet and outlet (1.24) are imposed via Lagrange multipliers, as proposed by Storti *et al.* [72]. Let i a node lying on the inlet (or outlet) boundary. Then, the equations for this node are modified in the following way

$$\begin{aligned}\Pi_n^-(\hat{\mathbf{U}})(\mathbf{U}_i - \hat{\mathbf{U}}) + \Pi_n^+(\hat{\mathbf{U}})\mathbf{U}_{\text{lm}} &= \mathbf{0} \\ \mathbf{R}_i + \Pi_n^-(\hat{\mathbf{U}})\mathbf{U}_{\text{lm}} &= \mathbf{0}\end{aligned}\tag{1.35}$$

where \mathbf{U}_{lm} is the vector of Lagrange multipliers and \mathbf{R}_i is the FEM residue for node i . At inlet regions $\hat{\mathbf{U}} = \mathbf{U}_{\text{ref}}$, with \mathbf{U}_{ref} a reference state. At outlet regions, Storti *et al.* [72] propose to take $\hat{\mathbf{U}}$ as the state of the fluid in the previous time step if the external conditions are unknown. They named this strategy ULSAR (Use Last State As Reference) and show that Riemann invariants are preserved in the limit $\Delta t \rightarrow 0$ and $h \rightarrow 0$, if such invariants exist.

Some internal combustion engines utilize ports for the gas-exchange process, such as two-stroke and rotative (Wankel [7], MRCVC [80], etc.) engines. Generally, the ports are placed on fixed walls of the engine (the cylinder or the housing) and, thus, have a relative motion respect to the flow domain. For example, figure 1.1 shows a scheme of a two-stroke engine with intake and exhaust ports located on the cylinder wall. In this case, an observer placed on the centroid of the flow domain sees the ports moving away in the bottom direction.

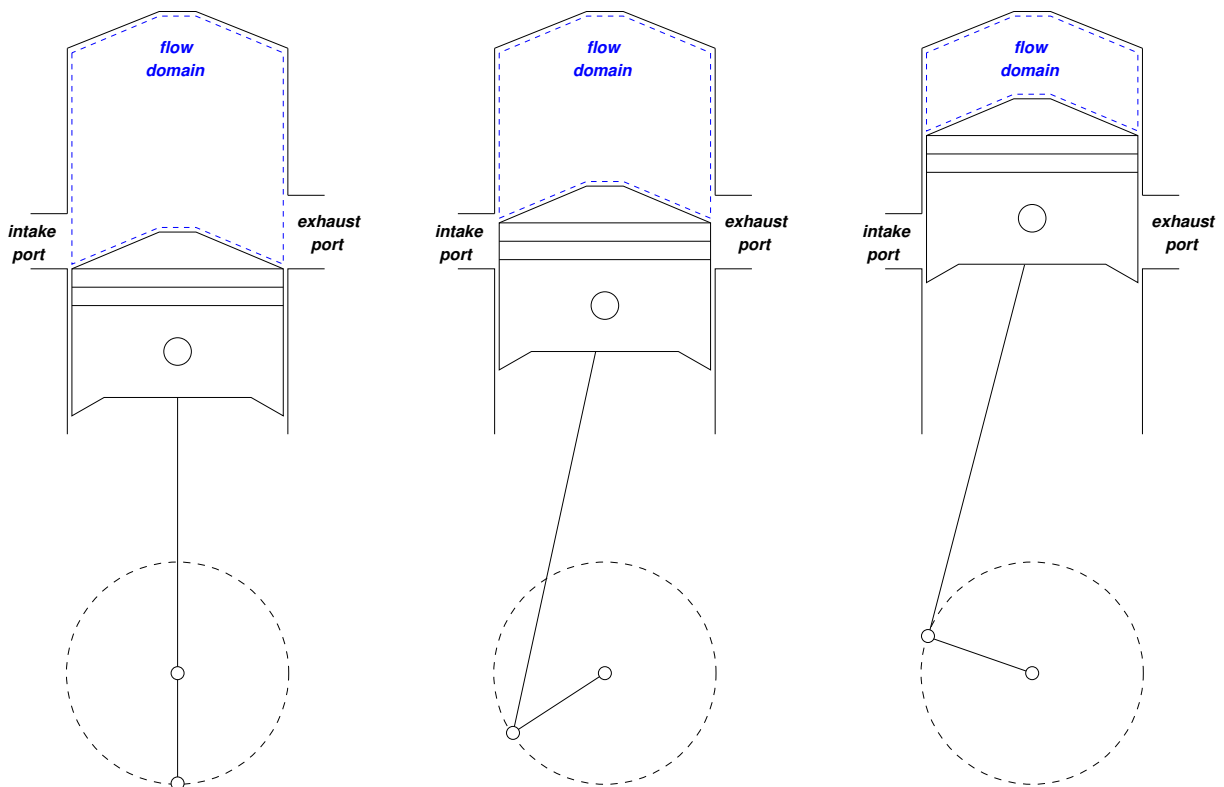


Figure 1.1: Two-stroke engine scheme.

A port could be modeled as a ‘hole’ in relative motion with respect to the boundary domain. This hole changes its passage area as the boundary moves, from open position to

the closed one, and vice-versa. Due to the nodal displacement produced by the deformation of the flow domain, mesh nodes lying on a boundary with a hole could change their position between the wall and the port (an inlet/outlet for the flow problem). Therefore, the boundary condition applied on each of such nodes must be changed appropriately in order to account for the node position. This is sketched in figure 1.2, where the mesh boundary nodes are filled according to the type of boundary condition. At time t , nodes 1 to 5 have absorbing boundary conditions since they lie on the passage area of the ports. At time $t + \Delta t$ and due to the mesh movement, nodes 1 to 4 are located on the cylinder walls and their boundary condition must change to the wall type. The strategy used consists in switching from an absorbing boundary condition (equation (1.35)) when the node is placed on the port region to a wall boundary condition when the node moves on the solid wall. The wall boundary condition is applied by means of constraints using

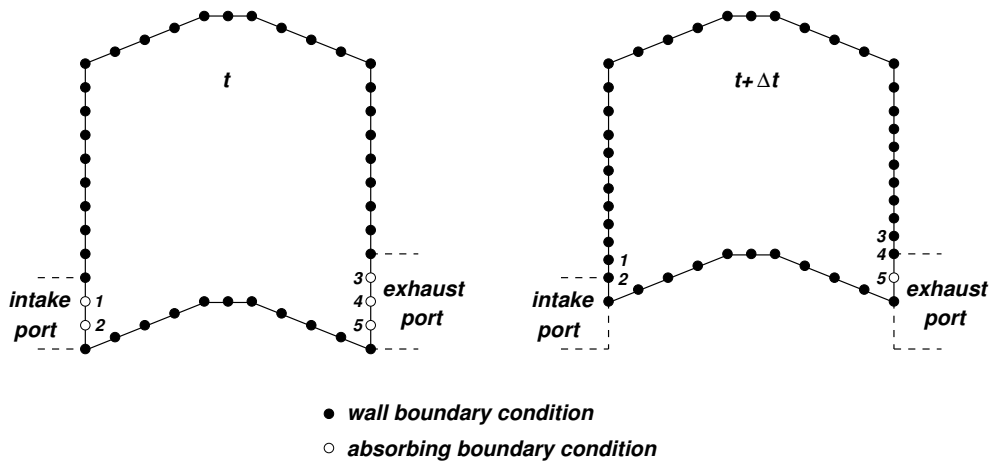


Figure 1.2: Change in the type of boundary condition due to the mesh movement.

Lagrange multipliers in order to keep the total number of degrees of freedom constant. For instance, in a 3D problem using a no-slip boundary condition and considering as null the velocity of the solid wall, the system of equations to solve for the node i is written as

$$\begin{aligned} \mathbf{M}\mathbf{U}_i + (\mathbf{I} - \mathbf{M})\mathbf{U}_{\text{lm}} &= \mathbf{0} \\ \mathbf{R}_i + \mathbf{M}\mathbf{U}_{\text{lm}} &= \mathbf{0} \end{aligned} \tag{1.36}$$

where $\mathbf{M} = \text{diag}[0, 1, 1, 1, 0]$.

Chapter 2

Mesh dynamics

*At the mass grave of religions man's triumphant age begins
the heartblood of these tyrants will wash us clean of sin
we'll stand among the victors through this colossal war
the sight of moving mountains is what we are here for*
Alghazanth

Several scientific and industrial applications of Computational Mechanics involve domains with moving boundaries. Examples of this kind of problems include free surfaces, two-fluid interfaces, fluid-object interaction, fluid-structure interaction, and moving mechanical components.

In computation of fluid flow problems with moving boundaries and interfaces, either an interface-tracking or an interface-capturing technique could be used, depending on the complexity of the interface as well as other aspects of the problem. An interface-tracking technique requires meshes that ‘track’ the interfaces, then, they need to be updated as the flow evolves. Besides, in an interface-capturing technique the computations are based on fixed spatial domains, where an interface function sets the location of the interface. This function needs to be computed in order to ‘capture’ the interface within the finite element mesh covering the area where the interface is located [76, 77].

In fluid-structure interaction problems, one of the most popular interface-tracking techniques is the ALE formulation [37, 11, 22], as described in chapter 1. In such applications, a body-conforming mesh has to be regenerated at each time step, or the existing grid has to be allowed to deform in order to follow the computational domain geometries. The former option is rather cumbersome and computationally expensive, especially for 3D problems, and could introduce an additional degradation of the numerical solution due to the projection of solutions from a mesh to another one. Furthermore, when implicit schemes are applied in an environment of parallel computing, the matrix profile must be calculated at each remeshing stage. More precisely, a total or a partial change in the topology of the mesh involves changes in the matrix profile. Thus, this additional computational cost introduced by the remeshing could become very important if the frequency of remeshing stages increases. The second option introduces the concept of a moving and deforming grid known as ‘dynamic’ mesh. In this case, the motion of the grid could cause

the deterioration of the mesh quality and, in some situations, generate an invalid mesh where any of the grid elements is inverted. It is well known that poor quality elements have strong influence on stability, convergence and accuracy of the numerical method used. In Computational Mechanics, the strategies developed to solve the mesh motion are grouped in a special topic named Computational Mesh Dynamics (CMD).

In this thesis, the moving mesh approach is preferable upon the remeshing approach due to the reasons mentioned above. Although a FSI problem may have instants at which a new mesh must be introduced, the goal is to develop a CMD strategy that permits to reduce the total number of remeshing stages. For relatively small domain deformations, there are many techniques which can solve the dynamics of the mesh. However, when the boundary displacements are relatively high most of these methods could fail to give a valid mesh. The domain deformation in internal combustion engines is very high, with topological changes and contact between different boundaries. Therefore, the CMD strategy utilized to solve the mesh dynamics in these problems should be as robust as possible. An advantage in the particular case of internal combustion engines, is the fact that the movement of the boundaries is known *a priori* and has a periodic behavior. Thus, the complete sequence of meshes could be generated before the resolution of the CFD problem.

2.1 Mesh quality

Following the paper by Knupp [39], an element quality metric is defined as

Definition. An element *quality metric* is a scalar function of node positions that measures some geometric property of the element.

If a 3D element has J nodes with coordinates $x_j \in \mathbb{R}^3$, $j = 0, 1, \dots, J - 1$, then a mesh quality metric is denoted by $\hat{f} : \mathbb{R}^{3J} \rightarrow \mathbb{R}$.

Some examples of element quality metrics are

- Volume (V).
- Aspect ratio, defined as the ratio between the radius of the sphere circumscribed to the element and the radius of the sphere inscribed in the element.
- Minimal dihedral angle.
- Size skewness (η), defined as

$$\eta = \frac{V - V_{\text{ref}}}{V_{\text{ref}}}$$

where V_{ref} is the volume of the equilateral element with the same (radius of the) sphere circumscribed as the actual element.

Definition. For a given element quality metric \hat{f} , the *mesh quality* (\hat{f}_{mesh}) is the minimum of \hat{f} over all the elements in the mesh, *i.e.* $\hat{f}_{\text{mesh}} = \min_e \hat{f}_e$.

2.2 The mesh dynamics strategy

The Computational Mesh Dynamics strategy developed could be classified as a mesh smoothing method. The strategy is based on an optimization problem where the functional is defined in terms of some appropriate element quality indicator.

The mesh topology (element connectivity) is assumed to remain constant, and the nodal coordinates are updated at each time step minimizing a given functional. In their most general form, the proposed functional is written as

$$F = F(\mathbf{x}) = F(\{x_j^\alpha\}) \quad (2.1)$$

where $\mathbf{x} = \{x_j^\alpha\}$ represents the whole set of nodal coordinates, and x_j^α is the α coordinate of the node j . The widely used pseudo-elastic CMD strategies can be expressed in that form, being F the functional of elastic energy.

At the time step n , the problem to solve is written as¹

$$\begin{aligned} & \min_{\mathbf{x}^n} F(\mathbf{x}^n) \\ & \text{s. to } \mathbf{x}^n \in \Gamma_{t^n} \Leftrightarrow \mathbf{x}^{n-1} \in \Gamma_{t^{n-1}} \end{aligned} \quad (2.2)$$

where Γ_t is the domain boundary. As it is observed, the nodes on the domain boundary are free to slide in the tangent direction. This possibility represents a non-linear restriction for the optimization problem (2.2) in the general case. Note that the constraint in equation (2.2) reduces in 1 the number of degrees of freedom for each boundary node. Thus, nodes lying on the intersection of two boundaries have two degrees of freedom less, and so on. In the particular case of vertexes, for instance, the number of constraints is equal to the number of spatial dimensions (and the number of degrees of freedom) and, hence, the node is attached to the vertex. A most simple type of boundary condition consists in fixing the nodal displacement in a predetermined value, but it is more restrictive from the point of view of the optimization problem.

2.2.1 Functional design

Being the dynamics of the mesh an artificial field in a FSI problem, there is enough freedom to design the functional in order to obtain meshes having good quality. Some design conditions for F are

- F should be computed from element contributions (as in an usual finite element assembly process).
- The minimum of F should give the best mesh quality.
- F should be well behaved enough in order to solve the minimization problem with Newton-like methods. In general, it will require F to have continuous first derivatives.
- F should be convex in order to guarantee uniqueness of the minimum and positivity of the stiffness matrices (the Hessian matrices of the functional).

¹The notation of equation (2.2) stands for ‘minimize $F(\mathbf{x}^n)$ subject to constraints $\mathbf{x}^n \in \Gamma_{t^n} \Leftrightarrow \mathbf{x}^{n-1} \in \Gamma_{t^{n-1}}$ ’.

The first item requires that the functional may be computed as

$$F = g(F_1(\{x_j^\alpha\}_1), F_2(\{x_j^\alpha\}_2), \dots) \quad (2.3)$$

where $F_e(\{x_j^\alpha\}_e)$ is the functional for element e , which is a function of the coordinates of its nodes and $g(\cdot)$ is some associative function that preserves convexity, for instance the sum or the maximum

$$\begin{aligned} g(F_1, F_2, \dots) &= F_1 + F_2 + \dots \\ g(F_1, F_2, \dots) &= \max(F_1, F_2, \dots) \end{aligned} \quad (2.4)$$

The second requirement is somewhat in conflict with the third and fourth ones. In the ideal case, F_e would be some indicator of the element badness (or distortion), and $g(\cdot)$ would be the maximum of its arguments, so the minimization of the mesh functional is equivalent to search for the mesh whose badness (*i.e.* the badness of the worst element) is minimum. However, using the maximum for $g(\cdot)$ it leads to functionals with non-differentiable first derivatives. Therefore functionals of the form

$$F = \sum_e |F_e(\{x_j^\alpha\}_e)|^p, \quad (2.5)$$

will be consider, preserving regularity while for $p \rightarrow \infty$ the maximum (L_∞) criterion is recovered.

Regarding for the design of the element functional F_e itself, at first sight it should be a function of its deformation only, in order to be invariant under dilatation, translation or rotation. However, the corresponding functional would be non-convex. Consider, for instance, a 1D problem covering the interval $[0, 1]$ with two linear elements, as in figure 2.1. There are three nodes, for which the position of nodes 1 and 3 are fixed by the boundary

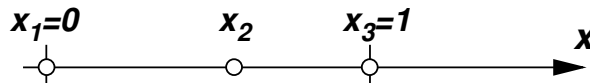


Figure 2.1: Invariance under dilatation.

conditions and the only unknown variable is the position of node 2. But if the functional is invariant under dilatation, then the functional for all 1D elements would be the same, and the position of the node 2 would be undetermined. In order to regularize the problem a term depending on the volume should be added to the functional.

The convexity of the functional is perhaps the most difficult restriction to accomplish. Consider for instance the case of two triangles $T = ABC$ and $T' = A'B'C'$ in figure 2.2. Both of them are well oriented (counterclockwise) and they are not too far from an equilateral one, therefore they should have a relatively low functional value (badness). Consider now the linear path that connects both of them, *i.e.* the family of triangles $T(\alpha)$ that are formed by linear interpolation of the coordinates of T and T' . For instance, for the A vertex

$$\mathbf{x}_A(\alpha) = (1 - \alpha)\mathbf{x}_A + \alpha\mathbf{x}_{A'} \quad (2.6)$$

The triangles for $\alpha = 0, 0.25, 0.5, 0.75$ and 1 are shown in figure 2.2. If the functional is convex, it should satisfy the inequality

$$F(T(\alpha)) \leq (1 - \alpha)F(T) + \alpha F(T') \quad (2.7)$$

in such a way the badness of the interpolated triangles for $0 < \alpha < 1$ should be lower than the extreme ones, *i.e.* they should be ‘nicer’. But as could be seen in the figure, for $\alpha = 0.5$ the triangles collapse in a line, so it not seems to be an appropriate criterion of badness that would be convex. Then the convexity requirement on the element functional is dropped.

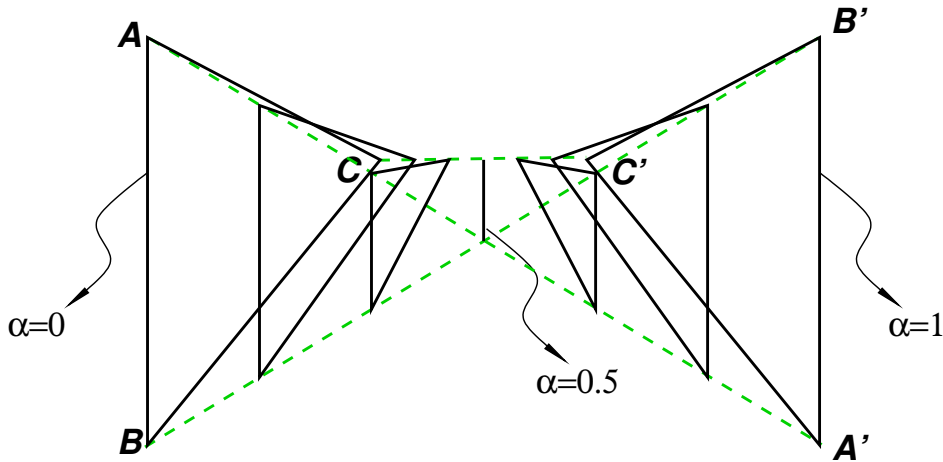


Figure 2.2: Convexity preservation.

Considering the items discussed above, the following expression for the element functional is proposed

$$F_e = C_v \left(\frac{V_e}{V_{\text{ref}}^e} - 1 \right)^m + C_q q_e^n \quad (2.8)$$

where V_e is the element volume, V_{ref}^e is the target element volume, q_e is any element-wise quality indicator, C_q and C_v are weight coefficients for shape and size terms in the functional respectively. The exponent m must be even and $n \in \mathbb{Z}^-$ in order to set the optimization problem as a minimization one. Then, the element distortion is defined as the reciprocal of the element quality. Furthermore, when negative values are adopted for the exponent n , the functional behaves like a geometric mean over the elements and, thus, gives more weight to the most distorted elements. Normally, C_v should be kept as small as possible but preserving the well-posedness of the problem. Exponents m and n allow to use different discrete norms to measure the element distortion and the element size change, *e.g.* $n = -2$ means Euclidean norm, and $n \rightarrow -\infty$ means maximum norm. This last case is equivalent to maximize the worst element quality [92].

Depending on how the system is solved, there are two possibilities: local methods and global methods. The global methods update the nodal position simultaneously for the whole set of nodes, while the local algorithms apply their methodologies over each subset of nodes until the whole set of nodes is updated, *i.e.*, the free nodes are relocated one by one iteratively, keeping the remainder fixed until the convergence is reached. Local methods are to global ones as explicit schemes are to implicit schemes for the resolution of differential equations with preponderant diffusive character. Furthermore, there is no guarantee that a solution of a global strategy may always be reached for a local method. In this thesis, the optimization problem is solved with a global scheme in order to avoid the drawbacks that local methods present.

The coordinate x_j^α is written as its initial grid coordinate plus a deformation \tilde{u}_j^α . When Newton-like methods are used, it is necessary to compute the gradient of the functional and, in some cases, its Hessian matrix too. The derivative of the functional given by equation (2.8) respect to \tilde{u}_j^α could be written as

$$R_j^\alpha = \frac{\partial F}{\partial \tilde{u}_j^\alpha} = \frac{mC_v}{V_{\text{ref}}} \left(\frac{V}{V_{\text{ref}}} - 1 \right)^{m-1} \frac{\partial V}{\partial \tilde{u}_j^\alpha} + nC_q q^{n-1} \frac{\partial q}{\partial \tilde{u}_j^\alpha} \quad (2.9)$$

The Hessian matrix is given by

$$\begin{aligned} K_{j^k}^{\alpha\beta} = \frac{\partial}{\partial \tilde{u}_k^\beta} \left(\frac{\partial F}{\partial \tilde{u}_j^\alpha} \right) &= \frac{mC_v}{V_{\text{ref}}} \left(\frac{V}{V_{\text{ref}}} - 1 \right)^{m-1} \left[\frac{m-1}{V - V_{\text{ref}}} \frac{\partial V}{\partial \tilde{u}_k^\beta} \frac{\partial V}{\partial \tilde{u}_j^\alpha} + \frac{\partial}{\partial \tilde{u}_k^\beta} \left(\frac{\partial V}{\partial \tilde{u}_j^\alpha} \right) \right] + \\ &+ nC_q q^{n-1} \left[\frac{n-1}{q} \frac{\partial q}{\partial \tilde{u}_k^\beta} \frac{\partial q}{\partial \tilde{u}_j^\alpha} + \frac{\partial}{\partial \tilde{u}_k^\beta} \left(\frac{\partial q}{\partial \tilde{u}_j^\alpha} \right) \right] \end{aligned} \quad (2.10)$$

The way in which the functional was written allows its application for any type of element if the quality indicator is properly defined. Here, it is proposed to use the following geometric quality indicator based on the subdivision of the element in simplexes (triangles in 2D and tetrahedra in 3D)

$$q = C \left[\sum_{i=1}^N (q_{S,i})^n \right]^{1/n} \quad (2.11)$$

where C is a normalization constant such that $0 < q \leq 1$, N is the total number of simplexes in the all possible subdivisions of the element in simplicial ones, and $q_{S,i}$ is computed for the simplex element i in the subdivision and it is given by

$$q_S = \frac{V}{S_e} \quad (2.12)$$

with $S_e = \sum_j l_j^{m_d}$, being l_j and V the length of the j -edge and the volume of the simplex, respectively. As can be shown, with an appropriate normalization constant q_S is an algebraic quality metric for simplicial elements. Thus, the quality indicator for non-simplicial elements is based on those defined for simplicial ones. Due to this fact, without loss of generality, only the simplex element case is considered in the analysis that follows.

According to equation (2.9), for the computation of the functional gradient it is necessary to have the expression of the derivative of the quality indicator with respect to the deformation. Therefore, the derivative of the expression (2.11) with respect to \tilde{u}_j^α (for $N = 1$) is

$$\frac{\partial q}{\partial \tilde{u}_j^\alpha} = \frac{C}{S_e^2} \left(S_e \frac{\partial V}{\partial \tilde{u}_j^\alpha} - V \frac{\partial S_e}{\partial \tilde{u}_j^\alpha} \right) \quad (2.13)$$

Computing the derivative of the last equation with respect to \tilde{u}_k^β , the following expression is obtained

$$\frac{\partial}{\partial \tilde{u}_k^\beta} \left(\frac{\partial q}{\partial \tilde{u}_j^\alpha} \right) = -\frac{2}{S_e} \frac{\partial S_e}{\partial \tilde{u}_k^\beta} \frac{\partial q}{\partial \tilde{u}_j^\alpha} + \frac{C}{S_e^2} \left[S_e \frac{\partial}{\partial \tilde{u}_k^\beta} \left(\frac{\partial V}{\partial \tilde{u}_j^\alpha} \right) + \frac{\partial S_e}{\partial \tilde{u}_k^\beta} \frac{\partial V}{\partial \tilde{u}_j^\alpha} - \frac{\partial V}{\partial \tilde{u}_k^\beta} \frac{\partial S_e}{\partial \tilde{u}_j^\alpha} - V \frac{\partial}{\partial \tilde{u}_k^\beta} \left(\frac{\partial S_e}{\partial \tilde{u}_j^\alpha} \right) \right] \quad (2.14)$$

Due to V and S_e are functions of the element nodal coordinates, it is algebraically convenient to adopt some local coordinates, for instance using the nodal distance to a given element node taken as the origin. Adopting the node with label 1 in the element as the reference node for every element in the mesh, the local coordinates can be defined as

$$\tilde{x}_j^\alpha = x_{j+1}^\alpha - x_1^\alpha \quad (2.15)$$

with $j = 1, \dots, n_{\text{en}} - 1$ and $\alpha = 1, \dots, n_{\text{d}}$, where n_{en} is the number of nodes in the element. This coordinate change allows to simplify the algebraic expressions for the computation of the element volume and the edges length, with no increment in complexity for the derivatives expressions. For example, for the volume, using the chain rule leads to

$$\frac{\partial V}{\partial \tilde{u}_j^\alpha} = \frac{\partial V}{\partial \tilde{x}_i^\gamma} \frac{\partial \tilde{x}_i^\gamma}{\partial \tilde{u}_j^\alpha} \quad (2.16)$$

From the definition given by equation (2.15), it is found that the tensor

$$\frac{\partial \tilde{x}_i^\gamma}{\partial \tilde{u}_j^\alpha} = (\delta_{(i+1)j} - \delta_{1j}) \delta_{\alpha\gamma} \quad (2.17)$$

is constant. Applying the chain rule in the equation (2.16), the second derivatives of the element volume may be computed as

$$\frac{\partial}{\partial \tilde{u}_k^\beta} \left(\frac{\partial V}{\partial \tilde{u}_j^\alpha} \right) = \frac{\partial}{\partial \tilde{x}_l^\mu} \left(\frac{\partial V}{\partial \tilde{x}_i^\gamma} \right) \frac{\partial \tilde{x}_l^\mu}{\partial \tilde{u}_k^\beta} \frac{\partial \tilde{x}_i^\gamma}{\partial \tilde{u}_j^\alpha} \quad (2.18)$$

In order to obtain expressions for the derivatives of the elemental volume respect to the local coordinates, the element type must be specified. As mentioned above, triangles and tetrahedra will be considered.

For triangles, the ‘volume’ (*i.e.* the area) could be computed as

$$V = \frac{1}{2} \begin{vmatrix} x_1^1 & x_1^2 & 1 \\ x_2^1 & x_2^2 & 1 \\ x_3^1 & x_3^2 & 1 \end{vmatrix}$$

In local coordinates, the last expression is written as

$$V = \frac{1}{2} \begin{vmatrix} 0 & 0 & 1 \\ x_2^1 - x_1^1 & x_2^2 - x_1^2 & 1 \\ x_3^1 - x_1^1 & x_3^2 - x_1^2 & 1 \end{vmatrix} = \frac{1}{2} \begin{vmatrix} \tilde{x}_1^1 & \tilde{x}_1^2 \\ \tilde{x}_2^1 & \tilde{x}_2^2 \end{vmatrix}$$

Therefore, the element volume expression is simplified to the following

$$V = \frac{1}{2}(\tilde{x}_1^1 \tilde{x}_2^2 - \tilde{x}_1^2 \tilde{x}_2^1) \quad (2.19)$$

The derivative of equation (2.19) with respect to \tilde{x}_i^γ could be written as

$$\frac{\partial V}{\partial \tilde{x}_i^\gamma} = \frac{1}{2} [\delta_{\gamma 1}(\tilde{x}_2^2 \delta_{i1} - \tilde{x}_1^2 \delta_{i2}) + \delta_{\gamma 2}(\tilde{x}_1^1 \delta_{i2} - \tilde{x}_2^1 \delta_{i1})] \quad (2.20)$$

Then, the derivative of the last equation with respect to \tilde{x}_l^μ is expressed as

$$\frac{\partial}{\partial \tilde{x}_l^\mu} \left(\frac{\partial V}{\partial \tilde{x}_i^\gamma} \right) = \frac{1}{2} (\delta_{l2} \delta_{i1} - \delta_{l1} \delta_{i2}) (\delta_{\gamma 1} \delta_{\mu 2} - \delta_{\gamma 2} \delta_{\mu 1}) \quad (2.21)$$

The denominator of equation (2.12) for the 2D-case in local coordinates is written as

$$S_e = |\tilde{\mathbf{x}}_1|^2 + |\tilde{\mathbf{x}}_2|^2 + |\tilde{\mathbf{x}}_2 - \tilde{\mathbf{x}}_1|^2 \quad (2.22)$$

where

$$|\tilde{\mathbf{x}}_j|^2 = \sum_{\alpha=1}^{n_d} (\tilde{x}_j^\alpha)^2 \quad (2.23)$$

The derivatives of the equation (2.22) are given by the following expressions

$$\frac{\partial S_e}{\partial \tilde{x}_i^\gamma} = 2 [\tilde{x}_i^\gamma + (-1)^i (\tilde{x}_2^\gamma - \tilde{x}_1^\gamma)] \quad (2.24)$$

$$\frac{\partial}{\partial \tilde{x}_l^\mu} \left(\frac{\partial S_e}{\partial \tilde{x}_i^\gamma} \right) = 2 \delta_{\mu \gamma} [\delta_{il} + (-1)^i (\delta_{2l} - \delta_{1l})] \quad (2.25)$$

The volume of a tetrahedral element could be computed as

$$V = \frac{1}{6} \begin{vmatrix} x_1^1 & x_1^2 & x_1^3 & 1 \\ x_2^1 & x_2^2 & x_2^3 & 1 \\ x_3^1 & x_3^2 & x_3^3 & 1 \\ x_4^1 & x_4^2 & x_4^3 & 1 \end{vmatrix}$$

This expression is reduced to a third order determinant when local coordinates are introduced

$$V = \frac{1}{6} \begin{vmatrix} 0 & 0 & 0 & 1 \\ x_2^1 - x_1^1 & x_2^2 - x_1^2 & x_2^3 - x_1^3 & 1 \\ x_3^1 - x_1^1 & x_3^2 - x_1^2 & x_3^3 - x_1^3 & 1 \\ x_4^1 - x_1^1 & x_4^2 - x_1^2 & x_4^3 - x_1^3 & 1 \end{vmatrix} = \frac{1}{6} \begin{vmatrix} \tilde{x}_1^1 & \tilde{x}_1^2 & \tilde{x}_1^3 \\ \tilde{x}_2^1 & \tilde{x}_2^2 & \tilde{x}_2^3 \\ \tilde{x}_3^1 & \tilde{x}_3^2 & \tilde{x}_3^3 \end{vmatrix}$$

The last determinant may be computed using the third order Levi-Civita tensor (ϵ_{pqr}) as

$$V = \frac{1}{6} \epsilon_{pqr} \tilde{x}_1^p \tilde{x}_2^q \tilde{x}_3^r \quad (2.26)$$

The summation over the power of length edges in the denominator of the element quality indicator for a tetrahedron in local coordinates may be expressed as

$$S_e = |\tilde{\mathbf{x}}_1|^3 + |\tilde{\mathbf{x}}_2|^3 + |\tilde{\mathbf{x}}_3|^3 + |\tilde{\mathbf{x}}_2 - \tilde{\mathbf{x}}_1|^3 + |\tilde{\mathbf{x}}_3 - \tilde{\mathbf{x}}_2|^3 + |\tilde{\mathbf{x}}_1 - \tilde{\mathbf{x}}_3|^3 \quad (2.27)$$

The derivatives of the equations (2.26) and (2.27) are given by the following expressions

$$\frac{\partial V}{\partial \tilde{x}_i^\gamma} = \frac{1}{6} \epsilon_{pqr} (\tilde{x}_2^q \tilde{x}_3^r \delta_{1i} \delta_{p\gamma} + \tilde{x}_1^p \tilde{x}_3^r \delta_{2i} \delta_{q\gamma} + \tilde{x}_1^p \tilde{x}_2^q \delta_{3i} \delta_{r\gamma}) \quad (2.28)$$

$$\begin{aligned} \frac{\partial}{\partial \tilde{x}_l^\mu} \left(\frac{\partial V}{\partial \tilde{x}_i^\gamma} \right) &= \frac{1}{6} \epsilon_{pqr} [\delta_{1l} \delta_{p\mu} (\tilde{x}_3^r \delta_{2i} \delta_{q\gamma} + \tilde{x}_2^q \delta_{3i} \delta_{r\gamma}) + \delta_{2l} \delta_{q\mu} (\tilde{x}_3^r \delta_{1i} \delta_{q\gamma} + \tilde{x}_1^p \delta_{3i} \delta_{r\gamma}) + \\ &\quad \delta_{3l} \delta_{r\mu} (\tilde{x}_2^q \delta_{1i} \delta_{p\gamma} + \tilde{x}_1^p \delta_{2i} \delta_{q\gamma})] \end{aligned} \quad (2.29)$$

$$\frac{\partial S_e}{\partial \tilde{x}_i^\gamma} = 3 [|\tilde{\mathbf{x}}_i| \tilde{x}_i^\gamma - |\tilde{\mathbf{x}}_{i+1} - \tilde{\mathbf{x}}_i| (\tilde{x}_{i+1}^\gamma - \tilde{x}_i^\gamma) + |\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_{i-1}| (\tilde{x}_i^\gamma - \tilde{x}_{i-1}^\gamma)] \quad (2.30)$$

$$\begin{aligned} \frac{\partial}{\partial \tilde{x}_l^\mu} \left(\frac{\partial S_e}{\partial \tilde{x}_i^\gamma} \right) &= 3 \left\{ [|\tilde{\mathbf{x}}_i|^{-1} \tilde{x}_i^\mu \tilde{x}_i^\gamma + |\tilde{\mathbf{x}}_i| \delta_{\mu\gamma}] \delta_{il} - \right. \\ &\quad [|\tilde{\mathbf{x}}_{i+1} - \tilde{\mathbf{x}}_i|^{-1} (\tilde{x}_{i+1}^\mu - \tilde{x}_i^\mu) (\tilde{x}_{i+1}^\gamma - \tilde{x}_i^\gamma) + \\ &\quad |\tilde{\mathbf{x}}_{i+1} - \tilde{\mathbf{x}}_i| \delta_{\mu\gamma}] (\delta_{i+1,l} - \delta_{il}) + \\ &\quad [|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_{i-1}|^{-1} (\tilde{x}_i^\mu - \tilde{x}_{i-1}^\mu) (\tilde{x}_i^\gamma - \tilde{x}_{i-1}^\gamma) + \\ &\quad \left. |\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_{i-1}| \delta_{\mu\gamma}] (\delta_{il} - \delta_{i-1,l}) \right\} \end{aligned} \quad (2.31)$$

The element functional (2.8) is continuous if $q_e \neq 0$ for all mesh elements, but $F_e(\mathbf{x})$ tends to infinity when q_e tends to zero. With the quality metric (2.12), this last situation could happen when there is at least one element in the mesh with $V_e \rightarrow 0$, due to S_e is bounded below if the element do not shrinks to a single point. Therefore, the application of this technique is restricted only to valid meshes, since infinite barriers arise when the element volume tends to zero, making it impossible to recover a valid mesh starting from an invalid one.

2.2.2 Differential predictor

The optimization strategy presented in section §2.2 means that at each time step the unknown node positions are obtained by solving the minimization problem (2.2). The mesh coordinates vector (\mathbf{x}) is composed by nodes at the boundary (\mathbf{x}_b) and the internal nodes (\mathbf{x}_{int})

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_b \\ \mathbf{x}_{\text{int}} \end{bmatrix} \quad (2.32)$$

A Newton-like strategy is used to solve this optimization problem and, for the sake of clarity, it is assumed that boundary nodes have their displacement prescribed. At each time step, the minimization problem consists in finding the vector \mathbf{x} that minimizes the

functional $F(\mathbf{x})$. Due to the fact that some components of \mathbf{x} (those in \mathbf{x}_b) are fixed by the boundary conditions, then

$$\mathbf{x}_{\text{int}}^n = \underset{\hat{\mathbf{x}}_{\text{int}}}{\operatorname{argmin}} F \left(\begin{bmatrix} \mathbf{x}_b^n \\ \hat{\mathbf{x}}_{\text{int}} \end{bmatrix} \right) \quad (2.33)$$

The recurrence formula from the Newton-Raphson strategy is

$$\mathbf{x}_{\text{int}}^{n,k+1} = \mathbf{x}_{\text{int}}^{n,k} - (\mathbf{K}^k)^{-1} \mathbf{R}^k \quad (2.34)$$

where

$$\begin{aligned} \mathbf{R} &= \frac{\partial F}{\partial \mathbf{x}_{\text{int}}} \\ \mathbf{K} &= \frac{\partial \mathbf{R}}{\partial \mathbf{x}_{\text{int}}} \end{aligned} \quad (2.35)$$

This generates a sequence $\mathbf{x}_{\text{int}}^{n,k}$ that, if it converges, will give the solution for the optimization problem

$$\lim_{k \rightarrow \infty} \mathbf{x}_{\text{int}}^{n,k} = \mathbf{x}_{\text{int}}^n \quad (2.36)$$

The simplest choice for the initial value $\mathbf{x}_{\text{int}}^{n,0}$ is taking the unknown vector at the previous time step, *i.e.*

$$\mathbf{x}_{\text{int}}^{n,0} = \mathbf{x}_{\text{int}}^{n-1,\infty} \quad (2.37)$$

However, this has the drawback that, if the elements near the moving boundary are small, then the initial combination $[\mathbf{x}_{\text{int}}^{n-1,\infty}, \mathbf{x}_b^n]$ may lead to invalid elements, even for small time steps. In fact, the time step is limited by the element size at the wall, and the limit time step of the moving mesh problem decreases with mesh refinement.

To avoid this, a linear predictor for the initial mesh is performed. If the solution $\mathbf{x}_{\text{int}}(t)$ is considered for each t in the range $t^{n-1} \leq t \leq t^n$, then

$$\mathbf{R}(\mathbf{x}_{\text{int}}(t), \mathbf{x}_b(t)) = 0 \quad (2.38)$$

Taking derivatives with respect to time and making an evaluation at $t = t^{n-1}$

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}_{\text{int}}} \right)_{t^{n-1}} \dot{\mathbf{x}}_{\text{int}}(t^{n-1}) + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}_b} \right)_{t^{n-1}} \dot{\mathbf{x}}_b(t^{n-1}) = 0 \quad (2.39)$$

then the Newton-Raphson sequence can be initialized with the extrapolation

$$\begin{aligned} \mathbf{x}_{\text{int}}^{n,0} &= \mathbf{x}_{\text{int}}^{n-1,\infty} + \Delta t \dot{\mathbf{x}}_{\text{int}}(t^{n-1}) \\ &= \mathbf{x}_{\text{int}}^{n-1,\infty} - \Delta t \left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}_{\text{int}}} \right)_{t^{n-1}}^{-1} \left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}_b} \right)_{t^{n-1}} \dot{\mathbf{x}}_b(t^{n-1}) \end{aligned} \quad (2.40)$$

where the dot designates time differentiation. For instance, consider a 1D problem with a homogeneous mesh of N linear elements in the interval $[0, 1]$. The right boundary is fixed and the left boundary moves to the right with velocity 1. With the standard initialization strategy, the limit time step is initially $\Delta t = h = 1/N$, since a larger time step will cause the left boundary to pass over the position of the first internal node (initially at $x = h$). Besides, with the differential predictor, the limit time step is $\Delta t = 1$, since, in fact, the differential predictor gives the optimal solution, and the subsequent Newton-Raphson iteration is not needed. It has been verified through numerical experiments that with the differential predictor the limiting time step Δt is independent of the mesh refinement.

2.2.3 Avoiding the relaxation of the initial mesh

Pseudo-elastic CMD strategies have the property that they do not move the initial mesh unless the domain boundary is deformed. This may be justified by means of elastic energy minimization arguments.

This property is not shared by the proposed functional because sometimes the initial mesh introduced by the user is not the optimal mesh with respect to this functional. Consider, for instance, the structured mesh $M1$ shown in figure 2.3. The mesh is composed by 200 triangular elements. Even if the mesh has a good quality, the optimization strategy tends to bring each element to a regular (equilateral) shape, then after a relaxation stage the mesh $M3$ is obtained. In this case, during the relaxation process the nodes on sides AB , CD are fixed, whereas those on BC , AD are left to slide on the horizontal direction. As a consequence of the optimization problem, the elements near vertexes A and C tend to shrink, whereas those near B and D tend to grow. This effect is caused by the particular way in which the squares have been split up into triangles. Note how the elements tend to reach the equilateral shape in the relaxed mesh. After the mesh has relaxed, subsequent displacements of the boundary nodes produce displacement of the internal nodes, as described before.

This initial ‘relaxation’ stage may or may not be desirable. If the initial mesh has bad quality, then this stage should produce a better new mesh. However, if the initial mesh has some *ad-hoc* refinement, then it is possible that the relaxation stage will revert this refinement. Consider for instance the mesh $M2$ in figure 2.3, which has a refinement towards the side AB in such a way that the horizontal spacing near CD is 3.5 times larger than the same at AB . As a result of the relaxation process, the relaxed mesh $M3$ is reached. The resulting relaxed mesh depends only on the topology of the mesh and on the constraints on the boundary nodes, but not on the initial position of the internal nodes. In fact, both meshes $M2$ and $M1$ (with and without refinement, respectively) produce the same final mesh $M3$ after relaxation.

The functional can be easily modified in order to keep the initial refinement. First, note that for simplicial elements there is a unique linear transformation $(\mathbf{x}_0, \mathbf{T})$ that transforms the coordinates $\{\mathbf{x}_{\text{reg},j}\}$ of the regular element (*i.e.* the equilateral triangle in 2D, the regular tetrahedron in 3D) to the actual element coordinates $\{\mathbf{x}_j\}$

$$\mathbf{x}_j = \mathbf{x}_0 + \mathbf{T} \mathbf{x}_{\text{reg},j} \quad (2.41)$$

It is easy to see that the functional can be expressed as a function of the transformation matrix \mathbf{T}

$$F_e = g(\mathbf{T}) \quad (2.42)$$

This fact can be seen because the functional can be computed by taking the nodal coordinates of the regular element, applying the transformation and finally computing the edge lengths, volume, and the functional. All these computations are encapsulated in the function $g(\cdot)$. Of course, the functional does not depend on the translation \mathbf{x}_0 . In fact, it only depends on the metric of the transformation $\mathbf{T}^T \mathbf{T}$, because it is independent of rotations. However, for the analysis that follows, it is needed to accept that it only depends on the transformation matrix, as reflected in (2.42). By construction, g has a minimum when $\mathbf{T} = c \mathbf{O}$, with $c \in \mathbb{R}$ a scaling factor and \mathbf{O} an orthogonal matrix, since in this case the actual element is similar to the regular one.

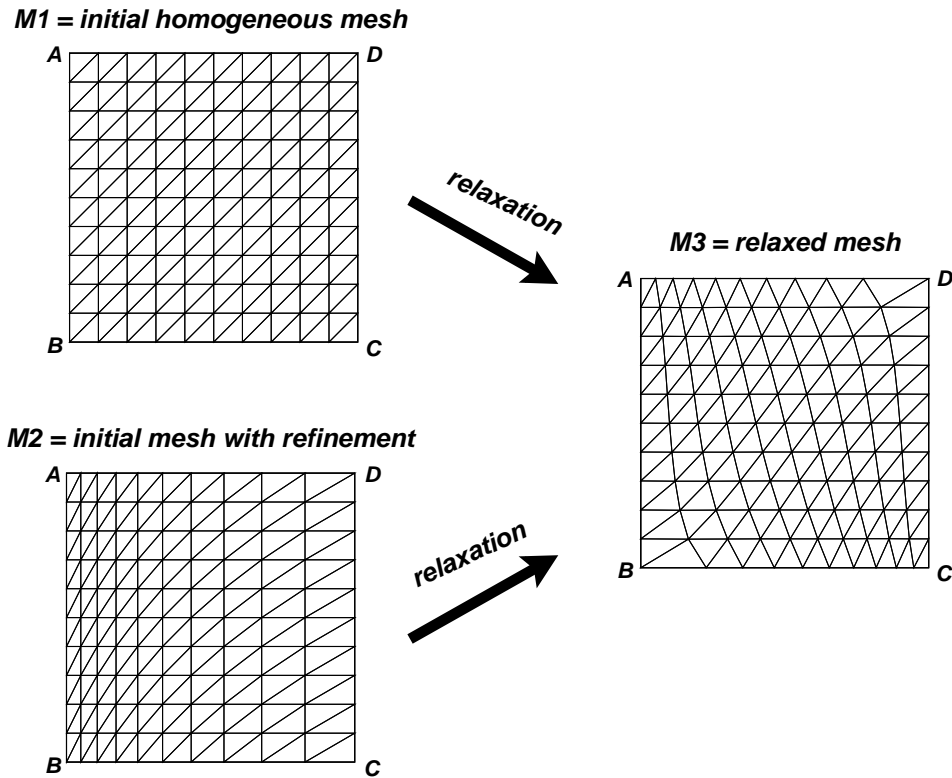


Figure 2.3: Relaxation of meshes.

The purpose is to modify the functional in order to the optimal element shape for F_e is not longer the regular element shape but it could be the shape of some reference element with coordinates $\{\mathbf{x}_{\text{ref},j}\}$ (see figure 2.4). It is easy to see that this can be done by considering the transformation from the reference element to the actual element as

$$F_e = g(\mathbf{T} \mathbf{T}'^{-1}) \quad (2.43)$$

where \mathbf{T}' transforms the regular element to the reference element. For instance, as mentioned above, a minimum is reached when $\mathbf{T} \mathbf{T}'^{-1} = c \mathbf{O}$, *i.e.* when the current element is similar in shape to the reference one.

Note that this modification can be simply introduced by computing the transformations \mathbf{T} , \mathbf{T}' and then computing the functional with the coordinates $\mathbf{x}'_j = \mathbf{T} \mathbf{T}'^{-1} \mathbf{x}_{\text{ref},j}$. An example can be seen in figure 2.5. The original mesh on the right has a refinement ratio of 1:10 near the AB side. Then, it is deformed on the side AB with a ramp of amplitude 0.2 (resulting in the mesh shown on the left of the figure). Note that if no initial relaxation is produced, the final mesh still has the refinement towards the AB side.

Computations of the analytical jacobians are also straightforward. The jacobians with respect to \mathbf{x}'_j are computed in the standard way, and then they are composed with the jacobian

$$\frac{\partial \mathbf{x}'_j}{\partial \mathbf{x}_j} = \mathbf{T} \mathbf{T}'^{-1} \mathbf{T}^{-1} \quad (2.44)$$

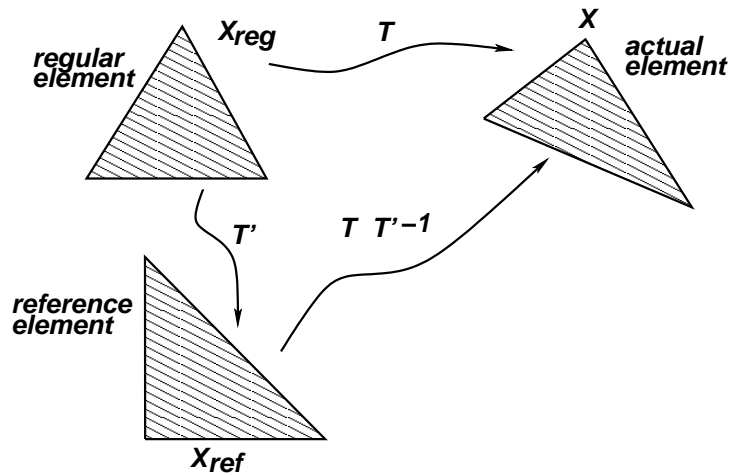


Figure 2.4: Compensation for initial deformation in reference mesh.

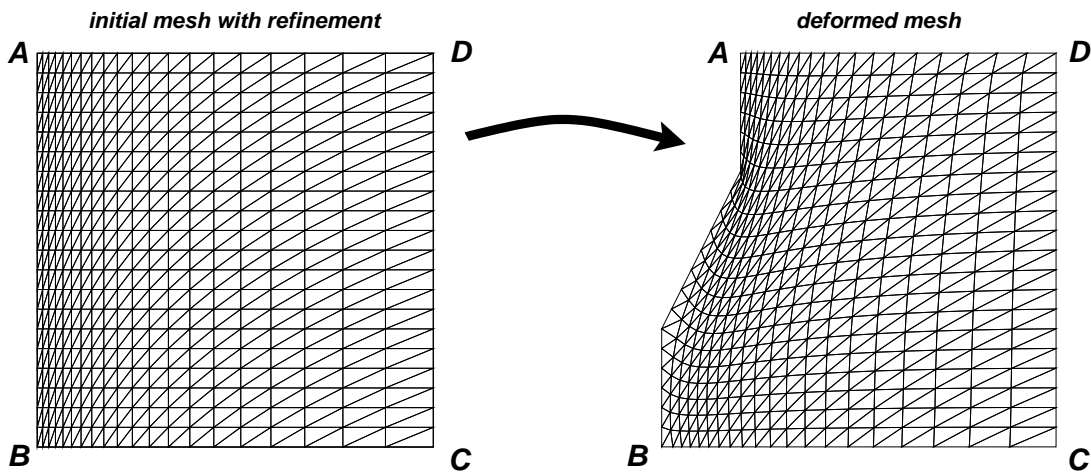


Figure 2.5: Mesh deformation with surface refinement.

2.2.4 Results

The proposed CMD strategy was applied to several test problems, in 2D and 3D. A test is considered successful if it achieves a valid mesh at every time step. Values for $C_q = 1$, $C_v = 0$ and $n = -1$ were adopted in the tests solved. The boundary was moved in such a way to produce an initial valid mesh for the nonlinear solver at each time step. The imposed law of movement for a boundary moving node i was of the form $\phi_i(t) = a_i \phi(t)$, where a_i is the spatial amplitude and

$$\phi(t) = \tanh\left(\frac{t}{\tilde{\tau}}\right) \quad (2.45)$$

is the temporal part of the dependency, $\tilde{\tau}$ being a coefficient that represents the duration of the change from the starting value to the end value and $t \in (0, 1)$.

Regarding for the nonlinear system resolution, the computational cost depends on the percentage of the domain deformation. Using a tolerance of 1×10^{-5} in the residue, for small deformation domain it is enough with one or two Newton iterations, while for harder problems, between five and seven Newton iterations were employed to guarantee the residual convergence.

In the next sections the definition of each case together with their results are presented.

Step 2D

Figure 2.6 sketches the domain and the deformation sequence for this problem. The mesh used has 200 triangular elements and 121 nodes, and the time step adopted was $\Delta t = 0.005$. On the nodes of the boundary domain the displacements have been imposed, being null for those nodes on fixed boundaries.

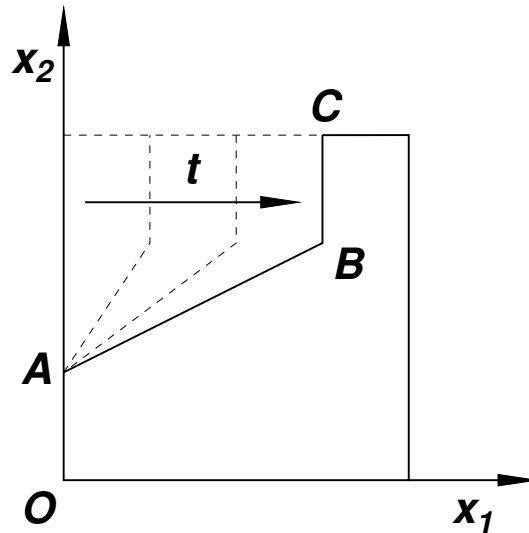


Figure 2.6: Domain of step 2D test.

In figure 2.7 several meshes obtained during the deformation sequence are shown. In figure 2.8 the minimum and mean values of the element quality indicator q as a function of the mesh deformation is plotted. As can be noted, the application of the strategy to this problem makes it possible to reach deformations larger than 99 %. Of course, the meshes obtained with the highest percentages of relative deformation may not be useful for a computation. Such deformations are reached in order to show the robustness of the method.

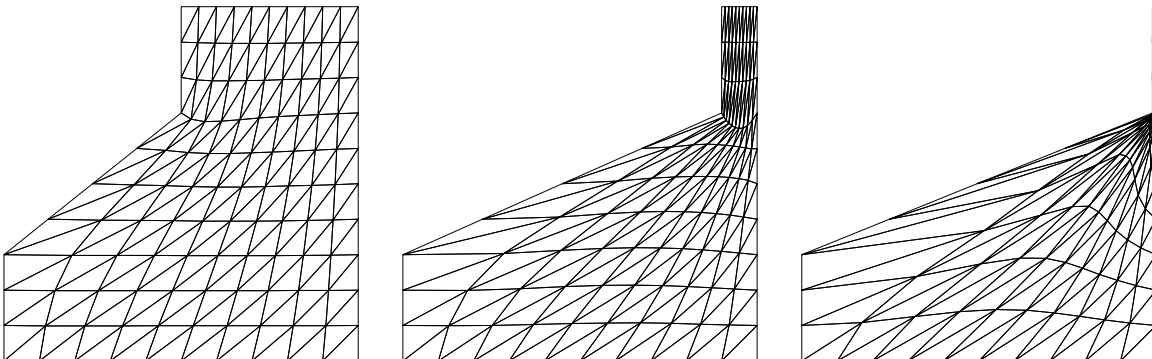


Figure 2.7: Mesh deformations for step 2D test: 50 % (left), 90 % (center) and 99 % (right).

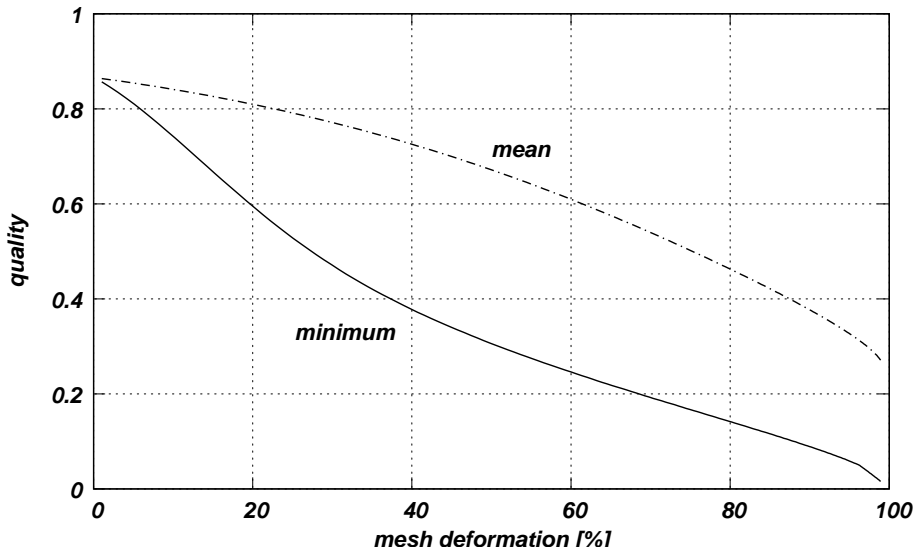


Figure 2.8: Step 2D minimum and mean element quality as a function of mesh deformation.

Step 3D

This test is the 3D extension of the test presented in the last section. Figure 2.9 shows the domain for different deformations. The top face of the cube is moved in the vertical direction and, during the deformation, this face is transformed into two planes of different heights joined by a truncated cone with upper radius $r_1 = 0.5l$ and lower radius $r_2 = 0.9l$, where l is the length side of the cube.

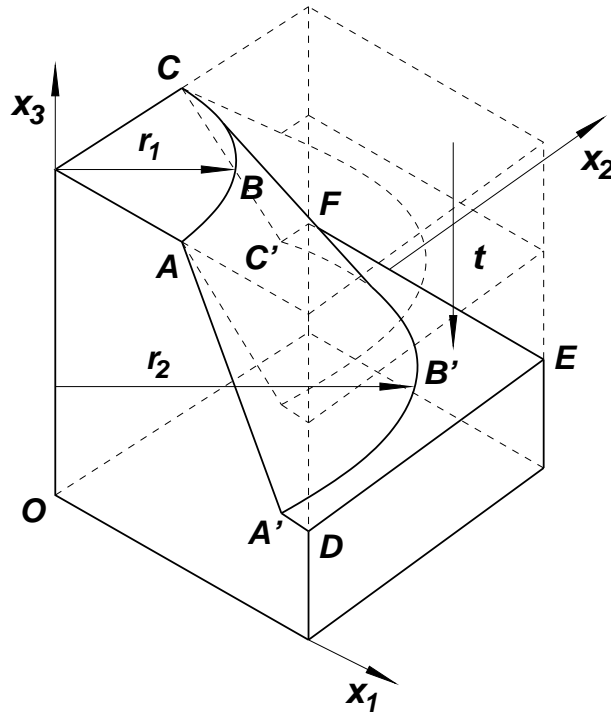


Figure 2.9: Problem definition for step 3D test.

The mesh used has 1080 elements and 343 nodes. As boundary conditions, imposed displacements were used. For nodes on moving boundaries, only the component of displacement in the direction x_3 is nonzero. The time step is $\Delta t = 0.005$ and the time law of movement is given by equation (2.45) with $\tilde{\tau} = 0.4$. Figure 2.10 shows the surface mesh at three different instants allowing to see how the moving mesh is deformed. The mean and minimum values of element quality indicator q as a function of mesh deformation is shown in figure 2.11.

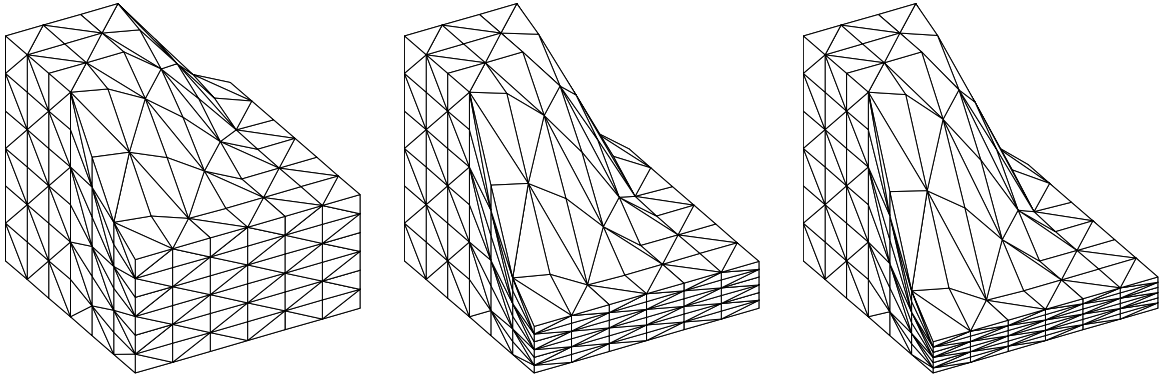


Figure 2.10: Mesh deformations for step 3D test: 50 % (left), 80 % (center) and 87 % (right).

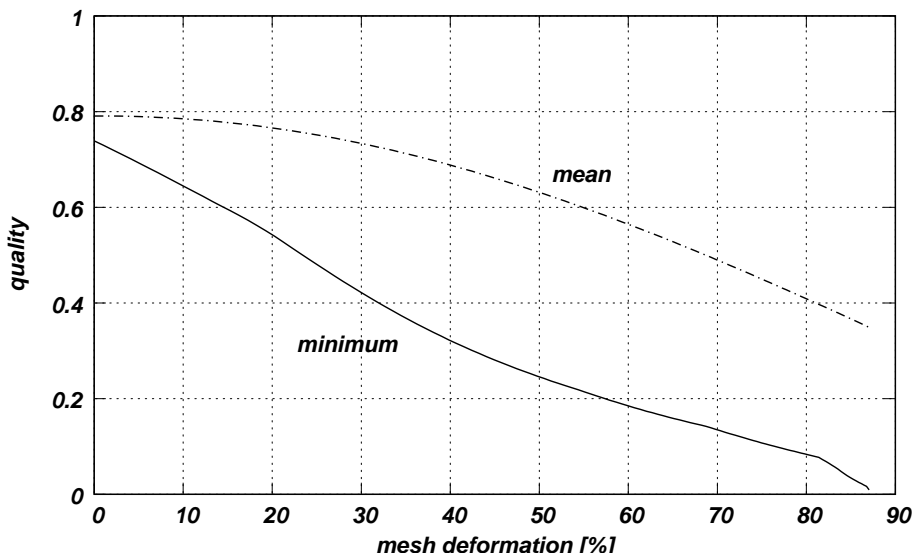


Figure 2.11: Step 3D minimum and mean element quality as a function of mesh deformation.

Moving a square inside another square

This test consists in a unit square inside another square with three units as length side. Initially, both squares are centered, as can be observed in figure 2.12. The internal square

is displaced in the vertical direction towards the top side of the external square. This test presents some features similar to the problem of a reciprocating piston or the closing of a poppet valve in an IC engine. The mesh has 710 triangular elements and 415 nodes, the time step adopted was $\Delta t = 0.005$, and $\tilde{\tau} = 0.5$ in equation (2.45).

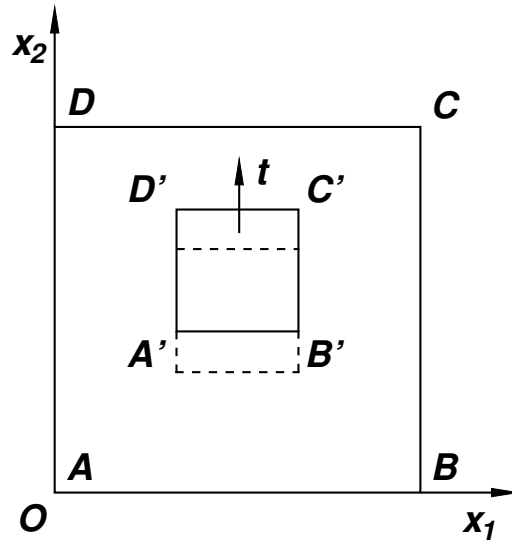


Figure 2.12: Problem definition for square within another square test.

Figure 2.13 shows deformed meshes for different times. As can be observed, it is possible to attain deformations larger than 99 % for this test case. In figure 2.14 the mean value and minimum value of element quality indicator q as a function of mesh deformation are shown.

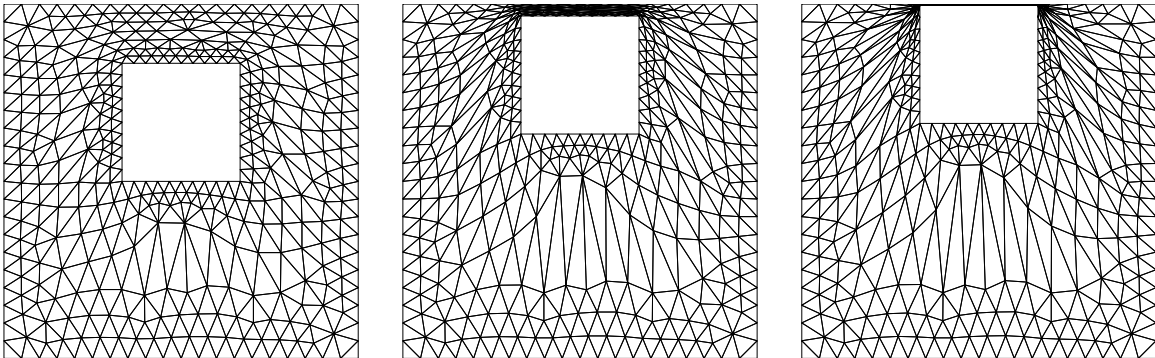


Figure 2.13: Mesh deformations for square within another square test: 50 % (left), 90 % (center) and 99 % (right).

Translation, rotation and bending 2D test

This test was proposed by Stein *et al.* [71] and it is concerned with the deformation of a two dimensional unstructured triangular mesh containing a very thin embedded structure, in this case a zero thickness plate was considered. The mesh domain is defined as $|x_1| \leq 1.0$ and $|x_2| \leq 1.0$, and the structure is placed in the region defined as $x_2 = 0.0$ and $|x_1| \leq 0.5$.

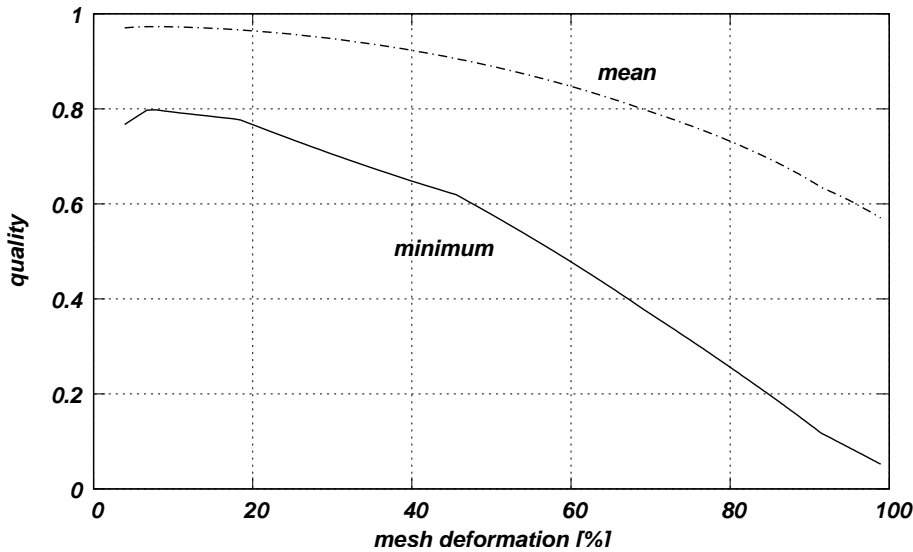


Figure 2.14: Minimum and mean element quality as a function of mesh deformation for square within another square test.

Around the structure, the mesh has three structured layers of elements of width $l_2 = 0.01$ with 50 elements along the span-wise of the structure ($l_1 = 0.02$). Figure 2.15 shows the original mesh which has 4.8K elements and 2.4K nodes. This kind of refinement is typical of viscous flow problems where the drag and lift deserve special attention.

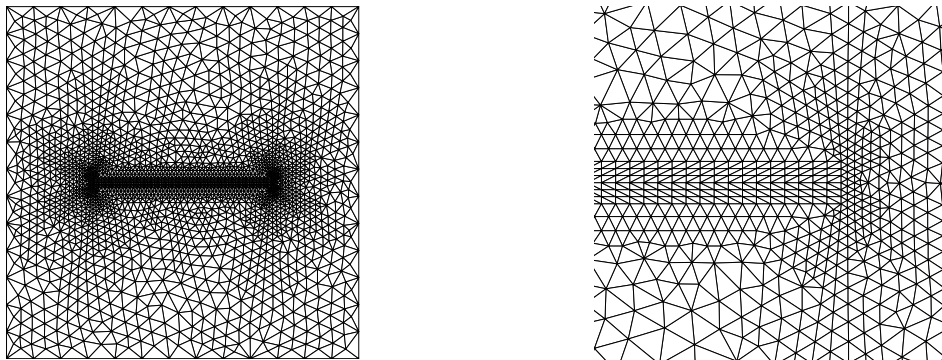


Figure 2.15: Initial mesh for the translation, rotation and bending 2D test.

The test involves three different types of structure movements: a rigid translation along direction x_2 , a rigid 2D rotation around the origin, and a prescribed bending. The maximum rigid translation in the vertical direction was $\Delta x_2 = 0.5$, for the rotation was $\Delta \theta = \pi/4$ and for bending the structure finally adopted a half circle shape ($\theta = \pi$). For every case the maximum deformation was adopted after 100 time steps. The nodes inside the structured layers were moved rigidly with the structure while the nodes on the domain boundaries are free to slide in the tangential direction.

Figure 2.16 shows the meshes for the time step of maximum deformation for each case. A detail of the region in the vicinity of the structure is also included.

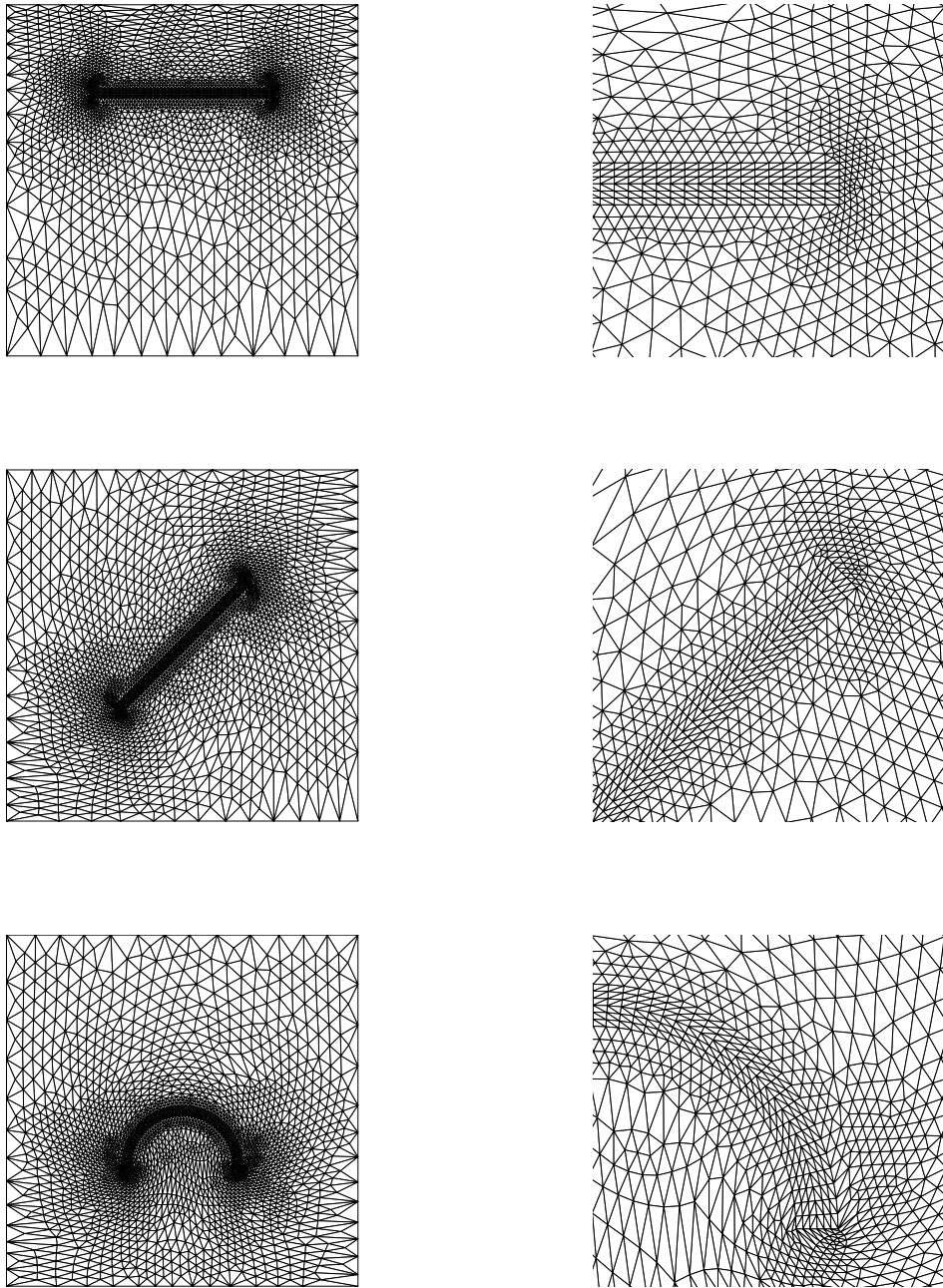


Figure 2.16: Final deformed meshes for translation (top), rotation (center) and bending (bottom) tests.

2.3 Simultaneous mesh untangling and smoothing

The CMD strategy presented requires valid meshes at the begin of each time step, as discussed in section §2.2. This limitation affects the maximum time step that can be used for the CMD problem, since the movement of the domain boundary could cause the tangling of the mesh (see section §2.2.2). In FSI problems this limitation is sometimes by-passed decreasing the time step size. However the computational cost suffers large increments, specially if some clustering of nodes is used near the moving boundary to

capture fluid dynamic details like boundary layers. When the mesh turns invalid, an untangling methodology should be used to recover a valid mesh. Untangling methods are commonly based on the element volume [38] and, in general, smoothing and untangling procedures are treated separately. Thinking of FSI problems, the CMD method should have the capability of solving the mesh motion even though inverted elements were found, guaranteeing a smooth mesh at each time step. Therefore, a simultaneous procedure of smoothing and untangling is preferable [25, 20, 23, 52]. It is in this sense that the CMD strategy, enhanced with simultaneous untangling and smoothing, is useful. The technique provides a way to recover a valid mesh, if it exists, despite starting from an invalid one.

Another approach commonly used to solve the mesh dynamics in FSI problems is to move the mesh as long as it is possible, and remesh when it is tangled or too distorted. It is possible to go back to the previous time step, or a few time steps before the mesh is tangled, to generate a (fully or partially) new mesh, project the solution from the old mesh to the new one, and keep on moving the new mesh. The projection introduces an error over the solution and it should be properly designed in order to conserve some physical quantities. Then, it is remarkable to note that one of the main disadvantages of remeshing lies on its inherent non-compliance of some conservation laws for the solution of FSI problems. Furthermore, an excessive number of remeshings increases the computational cost for implicit schemes in an environment of parallel computing due to the updating of the matrix profile. In some sense, simultaneous mesh untangling and smoothing is a procedure that allows to reduce the number of remeshing stages used during a simulation. With this feature, the error introduced by the projection needed for remeshing and the extra cost of updating the matrix profile may be decreased.

2.3.1 Functional regularization

In order to circumvent the drawbacks produced by the singularities of the proposed functional when the element volume tends to zero, it was modified the element quality indicator following an idea presented by Escobar *et al.* [23]. The modification consists in replacing the volume in equation (2.12) by the function

$$h(V) = \frac{1}{2}(V + \sqrt{V^2 + 4\delta^2}) \quad (2.46)$$

This is a strictly increasing function of the volume and it is also a positive function for all V (see figure 2.17). The regularization parameter δ represents the value of the function for a null volume.

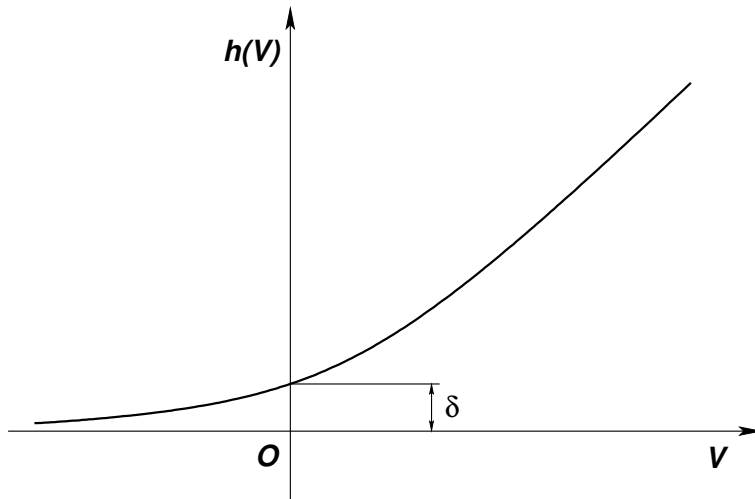
The modified functional is written as

$$F_e^*(\mathbf{x}) = C_v \left(\frac{V_e}{V_{\text{ref}}^e} - 1 \right)^m + C_q q_e^{*n} \quad (2.47)$$

where, for simplicial elements,

$$q^* = \frac{Ch(V)}{S_e} \quad (2.48)$$

Due to $V > 0 \forall V$, the regularized functional is continuous in the whole space of nodal coordinates.

Figure 2.17: Function $h(V)$.

The dependence of $h(V)$ with the parameter δ is such that

$$\lim_{\delta \rightarrow 0} h(V) = \begin{cases} V & \text{if } V \geq 0 \\ 0 & \text{if } V < 0 \end{cases}$$

Therefore, when the parameter δ tends to zero, the modified functional tends to the original one (for $V \geq 0$), and also, the modified optimal solution tends to the original one. Particularly, in the region of valid meshes, as $\delta \rightarrow 0$, the function $F^*(\mathbf{x})$ converges pointwise to $F(\mathbf{x})$. Besides, by considering that $\forall V > 0$,

$$\lim_{\delta \rightarrow 0} h'(V) = 1 \quad (2.49)$$

and

$$\lim_{\delta \rightarrow 0} h^{(r)}(V) = 0, \quad \text{for } r \geq 2 \quad (2.50)$$

it is easy to show that the derivatives of the modified functional verify the same property of convergence.

The purpose is to find a solution close enough to the optimal solution of the original functional, assuming that a valid mesh exists for the given topology and boundary position. Thus, it should be defined a decreasing sequence $\{\delta^k\}$ such that $\delta^k \rightarrow 0$ as $k \rightarrow \infty$. Then, with the elements of such a sequence applied for regularizing the functional, a simultaneous mesh untangling and smoothing strategy is obtained. This strategy could be used as a CMD technique with the property of not conditioning the time step in FSI problems.

2.3.2 Solution strategy

According to numerical examples, the lower the parameter δ , the slower the convergence rate of the optimization algorithm, without guarantee of final convergence. Moreover, if δ is not small enough the ‘optimal’ mesh finally obtained may be invalid, being even worse for high relative domain deformations. Therefore, two main problems arise

- finding the decreasing rate of the sequence $\{\delta^k\}$ to ensure the final convergence to a valid mesh, if it exists.
- finding a suitable initial value for the sequence (δ^0) .

To determine an equation that allows the decreasing of δ , this parameter is assumed as a new global variable only for theoretical purposes, *i.e.* considering $F^* = F^*(\mathbf{x}, \delta)$. If the problem is posed in terms of the variables (\mathbf{x}, δ) and using a Newton-like solver, the problem is written as follows

$$\begin{bmatrix} \frac{\partial^2 F^*}{\partial \mathbf{x}^2} & \frac{\partial^2 F^*}{\partial \mathbf{x} \partial \delta} \\ \frac{\partial^2 F^*}{\partial \delta \partial \mathbf{x}} & \frac{\partial^2 F^*}{\partial \delta^2} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \delta \end{bmatrix} = - \begin{bmatrix} \frac{\partial F^*}{\partial \mathbf{x}} \\ \frac{\partial F^*}{\partial \delta} \end{bmatrix}$$

Writing the last equation in the following way

$$\begin{aligned} \frac{\partial^2 F^*}{\partial \mathbf{x}^2} \Delta \mathbf{x} + \frac{\partial^2 F^*}{\partial \mathbf{x} \partial \delta} \Delta \delta &= - \frac{\partial F^*}{\partial \mathbf{x}} \\ \frac{\partial^2 F^*}{\partial \delta \partial \mathbf{x}} \Delta \mathbf{x} + \frac{\partial^2 F^*}{\partial \delta^2} \Delta \delta &= - \frac{\partial F^*}{\partial \delta} \end{aligned} \quad (2.51)$$

it is observed that this system may be solved in an uncoupled way if the parameter δ is kept fixed for the first equation ($\Delta \delta = 0$). This is equivalent to solve the system (2.51) using the block Gauss-Seidel method. Thus, the variable increments $\Delta \mathbf{x}$ and $\Delta \delta$ are obtained

$$\begin{aligned} \Delta \mathbf{x} &= - \left(\frac{\partial^2 F^*}{\partial \mathbf{x}^2} \right)^{-1} \frac{\partial F^*}{\partial \mathbf{x}} \\ \Delta \delta &= - \frac{\left(\frac{\partial F^*}{\partial \delta} + \frac{\partial^2 F^*}{\partial \delta \partial \mathbf{x}} \Delta \mathbf{x} \right)}{\frac{\partial^2 F^*}{\partial \delta^2}} \end{aligned} \quad (2.52)$$

The expression for $\Delta \delta$ in (2.52) is adopted as the maximum value to reduce δ . Therefore, the updated δ in the iteration k is defined in the following way

$$\delta^k = \max(\delta^{k-1} - \tilde{\alpha} |\Delta \delta^{k-1}|, \tilde{\beta} \delta^{k-1}) \quad (2.53)$$

where $\tilde{\alpha}$ and $\tilde{\beta}$ are constants lower than one.

In addition, it was found that the off-diagonal terms in the element-wise matrix have a strong influence on the convergence of this optimization method. In the untangling stage, it is advisable to relax these off-diagonal terms to make the matrix more diagonal-dominant. However, in the smoothing stage these terms should be restored to take advantage of the convergence rate of full Newton schemes. Here, the relaxation parameter for these off-diagonal terms ($\tilde{\gamma} \leq 1$) may be constant or a function of the iterations. For example, for a 2D case the element-wise matrix is modified as

$$\mathbf{K}^e = \begin{bmatrix} \left(\frac{\partial^2 F^*}{\partial \mathbf{x}_1^2} \right)_e & \tilde{\gamma} \left(\frac{\partial^2 F^*}{\partial \mathbf{x}_1 \partial \mathbf{x}_2} \right)_e \\ \tilde{\gamma} \left(\frac{\partial^2 F^*}{\partial \mathbf{x}_2 \partial \mathbf{x}_1} \right)_e & \left(\frac{\partial^2 F^*}{\partial \mathbf{x}_2^2} \right)_e \end{bmatrix}$$

The problem is solved by the Newton-Raphson method with Armijo inexact line search [61]. At each iteration δ is diminished only if the line search strategy gives a unit step as result.

If the mesh is initially invalid, the initial value δ^0 is chosen according to the following criterion based on the minimum volume ($V_{\min} = \min_e V_e$). Due to the fact that $h(V)$ is a strictly increasing function, then

$$h_{\min} = h(V_{\min}) = \frac{1}{2} \left(V_{\min} + \sqrt{V_{\min}^2 + 4\delta^2} \right) \quad (2.54)$$

Defining $h_{\min}^* = h_{\min}/\delta$ as a user-defined parameter and getting δ from the last equation, the following criterion to initialize δ arises

$$\delta^0 = \begin{cases} \frac{h_{\min}^* V_{\min}}{h_{\min}^{*2} - 1} + \epsilon_\delta & \text{if } V_{\min} \leq \epsilon_\delta \\ 0 & \text{if } V_{\min} > \epsilon_\delta \end{cases}$$

where $\epsilon_\delta > 0$ is the minimum value given to the initial value of δ such that $\delta^0 > 0$ when $V_{\min} = 0$.

2.3.3 Results

In this section, the numerical results for some test examples are presented. These examples are CMD problems in 2D and 3D with different deformations of the boundary. In the whole set of test cases, the following convergence criteria had been applied

- Valid mesh.
- For the iteration k , $\frac{|q_{\min}^k - q_{\min}^{k-1}|}{q_{\min}^k} < \epsilon_q$, being $q_{\min} = \min_e q_e$ and $\epsilon_q > 0$ a prefixed tolerance.

The relaxation coefficient for the Hessian matrix was chosen as $\tilde{\gamma} = 0.5$ for the untangling stage, and $\tilde{\gamma} = 1$ for the smoothing stage. In the whole set of numerical examples, the following set of parameter values were used: $C_v = 0$, $C_q = 1$, $n = -1$, $\tilde{\alpha} = 1$, $\tilde{\beta} = 0.1$, $\epsilon_q = 0.01$, $h_{\min}^* = 0.75$ and $\epsilon_\delta = 1 \times 10^{-6}$. In these tests, the reference element used was the regular element.

Step 2D

This test was defined in section §2.2.4, and here is solved for a relative domain deformation of 50 %, 90 % and 99 % using the simultaneous untangling and smoothing strategy. The total mesh deformation is carried out in one time step in order to show the robustness of the proposed algorithm. For each case, the initial tangled mesh and the resulting valid grid are presented in figures 2.18 to 2.20. In addition, the evolution of the mesh quality with iterations is included in figure 2.21.

Figure 2.22 shows the comparison between the smoothing (S) and the simultaneous untangling and smoothing (U-S) strategies. It is observed that the elapsed time in the computation with the U-S technique is approximately independent of the deformation. It

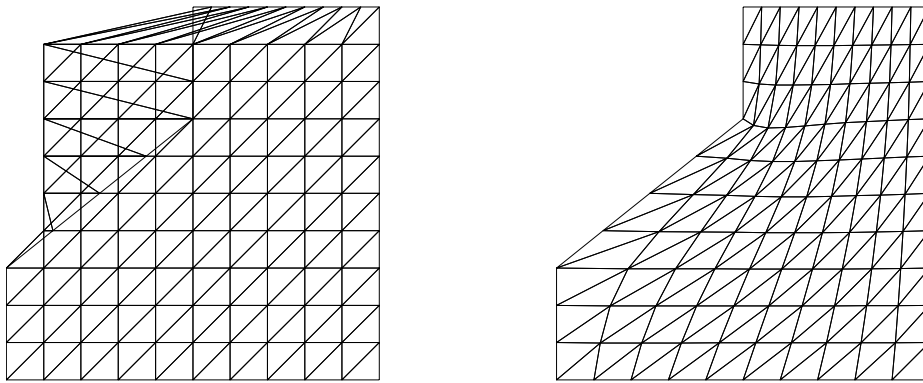


Figure 2.18: Mesh deformation of 50 % for step 2D test, initial (left) and final (right) meshes.

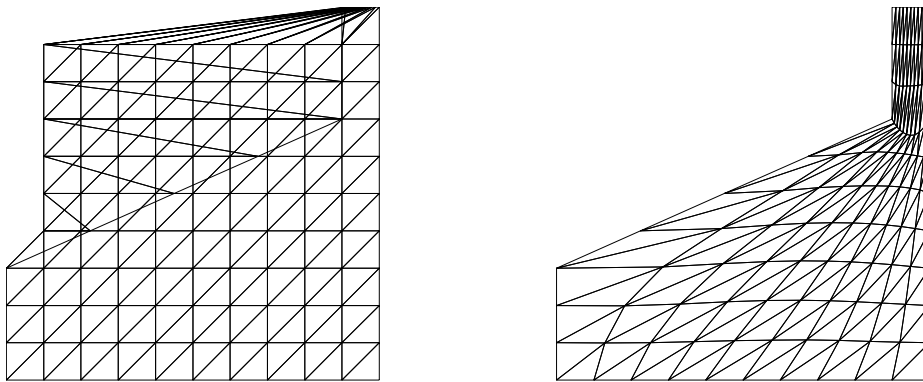


Figure 2.19: Mesh deformation of 90 % for step 2D test, initial (left) and final (right) meshes.

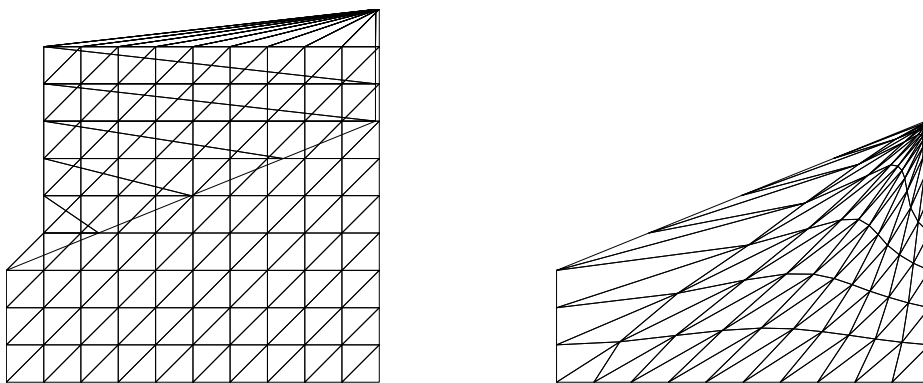


Figure 2.20: Mesh deformation of 99 % for step 2D test, initial (left) and final (right) meshes.

does depend on the number of time steps used, and tends towards the elapsed time of the S method as the number of time steps are increased.

In order to verify the utility of the differential predictor (DP) explained in section §2.2.2, the test was solved for a relative domain deformation of 90 % varying the

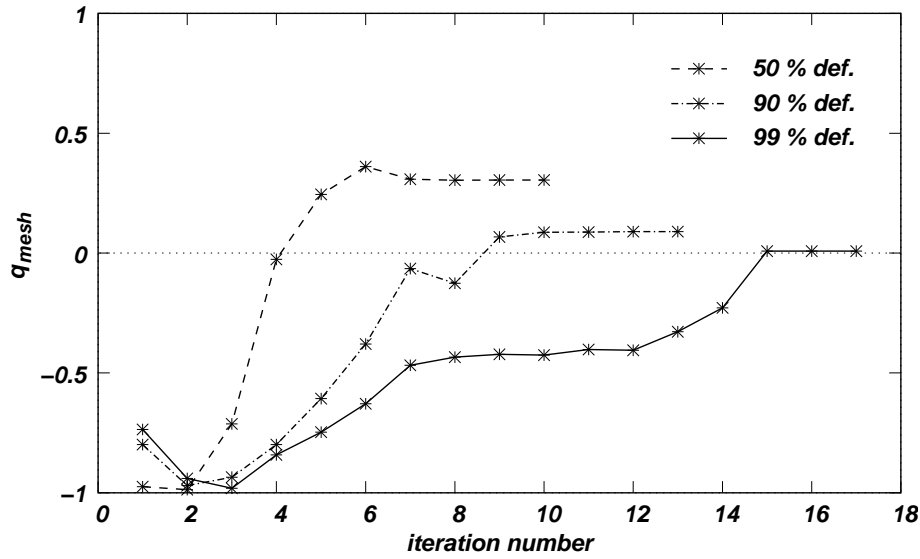


Figure 2.21: Mesh quality as a function of iterations for step 2D test.

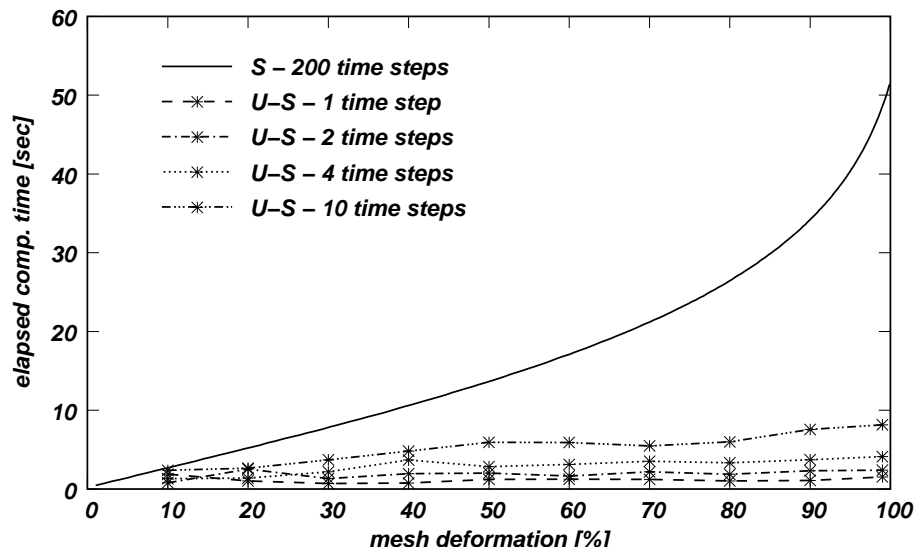


Figure 2.22: Computational time in terms of the relative domain deformation for step 2D test.

amount of time steps used. In table 2.1 the total number of iterations in each case (with and without DP) is presented. As it is observed, the use of the DP makes it possible to decrease in a noticeable way the number of iterations (and therefore the cost), mainly when the time step used diminishes.

Step 3D

As presented in section §2.2.4, this test is the 3D extension of the step 2D test. The test is solved for 50 %, 80 % and 87 % of relative domain deformation using one time step for each case. Figures 2.23 to 2.25 show the initial and final meshes obtained.

number of time steps used	iterations with DP	iterations without DP
1	15	13
2	19	27
4	34	47
8	39	69

Table 2.1: Total number of iterations to reach a relative domain deformation of 90 % for step 2D test.

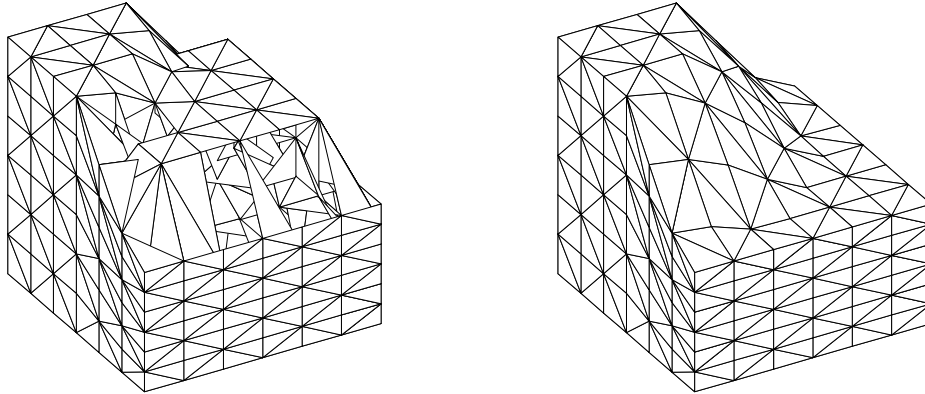


Figure 2.23: Mesh deformation of 50 % for step 3D test, initial (left) and final (right) meshes.

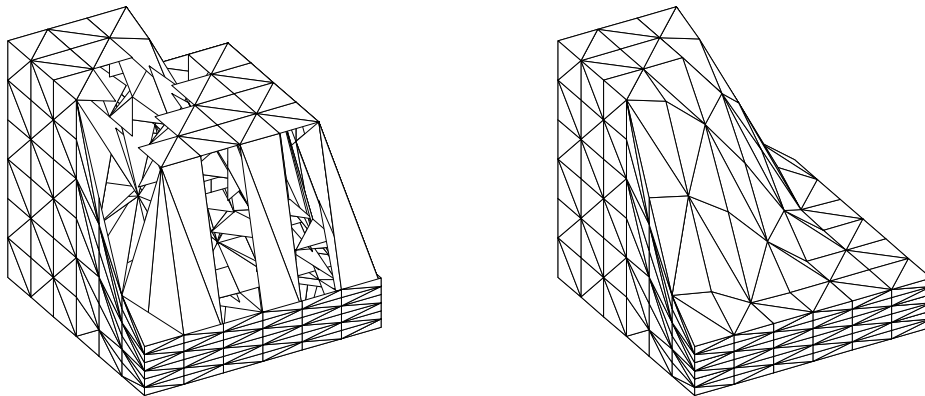


Figure 2.24: Mesh deformation of 80 % for step 3D test, initial (left) and final (right) meshes.

Figure 2.26 shows the mesh quality as a function of iterations. For 87 % of mesh deformation, the number of iterations is high since the boundary position is close to the limit in which a valid mesh exists for the conditions of the test. A comparison of computational cost with mesh deformation between S and U-S strategies was done, and the results achieved are shown in figure 2.27.

Again, as in previous tests, the advantage of applying the DP can be observed in table 2.2. In this case, the problem was solved for a relative domain deformation of 70 %

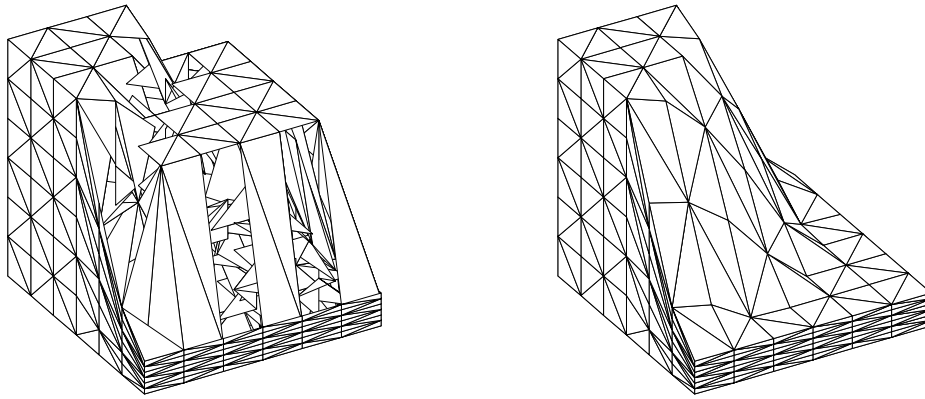


Figure 2.25: Mesh deformation of 87 % for step 3D test, initial (left) and final (right) meshes.

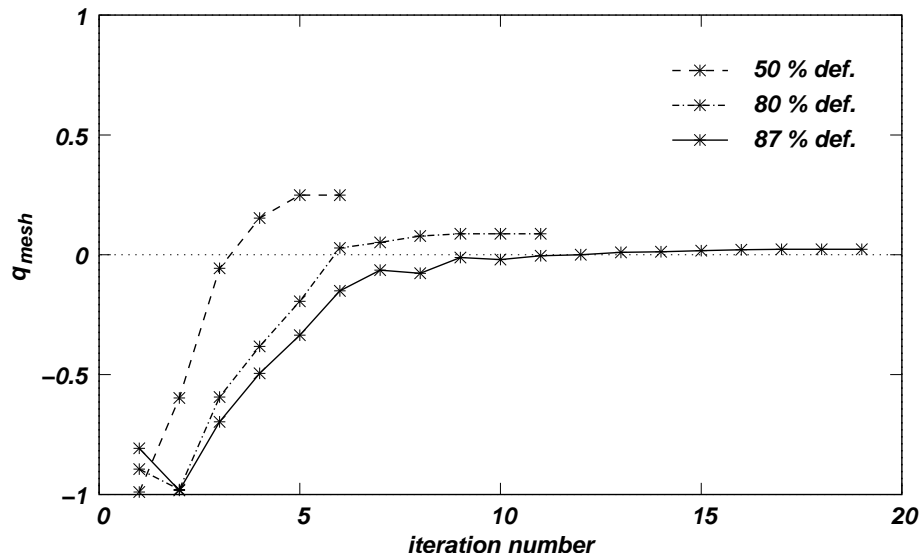


Figure 2.26: Mesh quality as a function of iterations for step 3D test.

varying the size of the time step.

number of time steps used	iterations with DP	iterations without DP
1	8	7
2	9	12
4	13	24
8	20	48

Table 2.2: Total number of iterations to reach a relative domain deformation of 70 % for step 3D test.

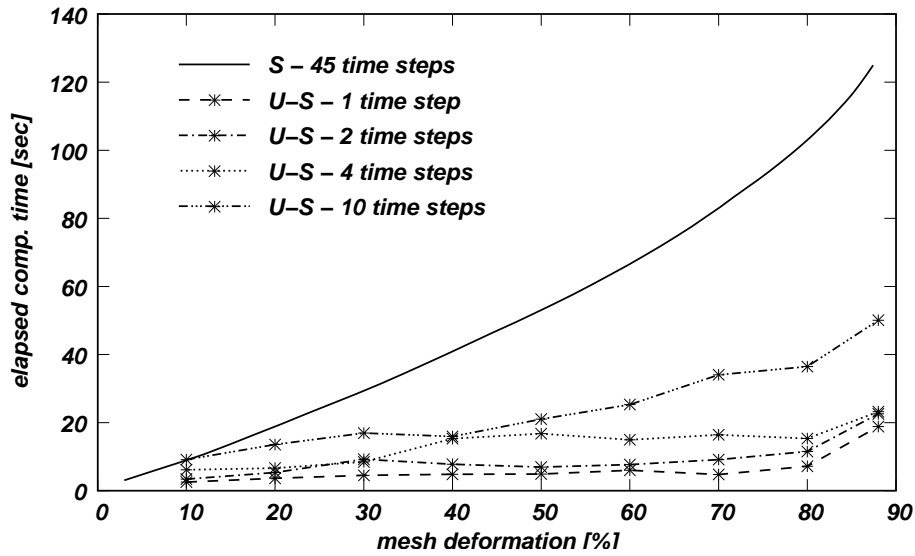


Figure 2.27: Computational time in terms of the relative domain deformation for step 3D test.

Scaled cube

This test was proposed by Montenegro *et al.* [52] and it consists in a unit cube with 625 tetrahedrons and 216 nodes. The invalid initial mesh was obtained by transforming the cube into another cube of 10 units of side length, changing the coordinates of some nodes in the following way: the internal nodes were kept fixed, those nodes lying on the edges were relocated on the edges of the new cube and those nodes lying on the faces of the original cube were projected on the new faces, respectively (see figure 2.28). Figure 2.28 shows the final mesh obtained.

In figure 2.29 the minimum quality as a function of the number of iterations is presented.

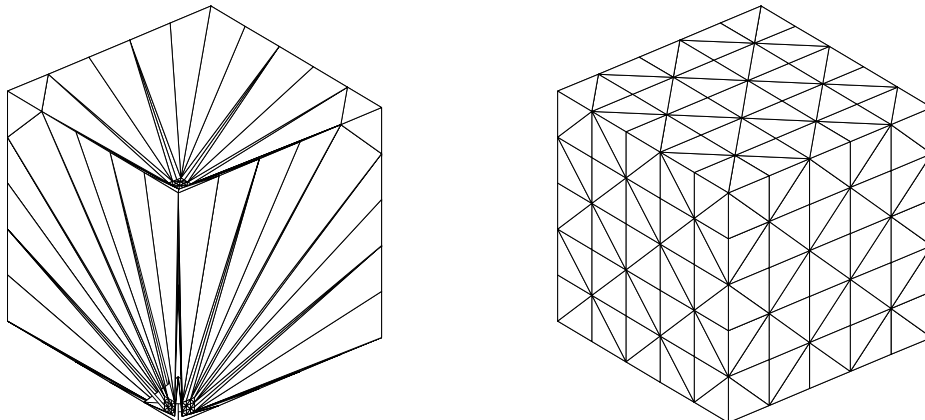


Figure 2.28: Initial tangled mesh (left) and the resulting final mesh (right) for the scaled cube test.

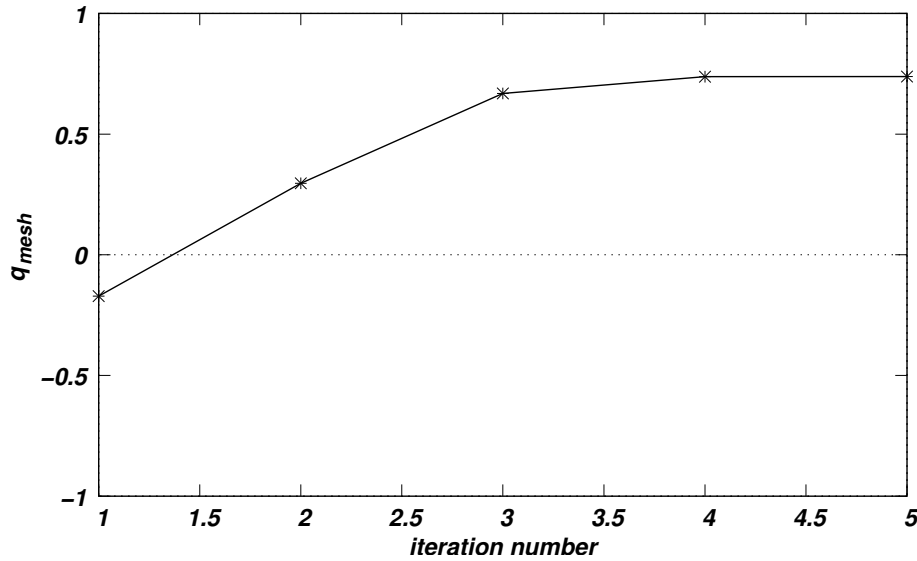


Figure 2.29: Mesh quality as a function of iterations for the scaled cube test.

Axisymmetrical flowmeter

Finally, the methodology is applied to IC engine geometries. One of the most important device to measure the mass flow rate of different cylinder heads is the flowmeter bench. This example consists in solving the mesh dynamics of an axisymmetrical flowmeter, whose geometry is shown in figure 2.30. The head of the valve moves as a rigid solid and the portion of the boundary representing its stem stretches and shortens depending on the displacement of the valve. The other boundary that stretches and shortens is the axis of symmetry. Initially, the valve lift is 5 mm, and the valve is moved with a linear law in the x_2 -direction until reaching the minimum valve lift of 0.5 mm.

The mesh has 12K triangular elements and 6.4K nodes, with $h \simeq 0.2$ mm in the region between the valve and its seat. Nodes lying on the vertical walls of the cylinder and those lying on the valve stem are left to slide in the x_2 -direction, while the remaining boundary nodes have their displacements prescribed. In figure 2.31(a) it is shown a close-up of the mesh utilized. Figure 2.31(b) presents the mesh from which the CMD strategy starts, and where invalid elements were filled in green. The final valid mesh is shown in figure 2.31(c). In this case, the total domain deformation was applied in one time step.

In table 2.3 it is summarized the elapsed computational time for obtain a valid mesh for the minimum valve lift varying the number of time steps. The problem was solved with and without the differential predictor for comparison purposes. The coefficients used for the U-S technique were chosen in order to minimize the total number of iterations when one time step is utilized. As it is observed in the table, when the differential predictor is applied the elapsed computational time remains approximately unchanged.

Diesel engine with three valves

The last case presented in this chapter is the mesh dynamics for the 3D geometry of a diesel engine. The engine has three valves and bowl-in piston. The purpose is to solve the movement of a unique mesh along the whole cycle in order to show how the CMD

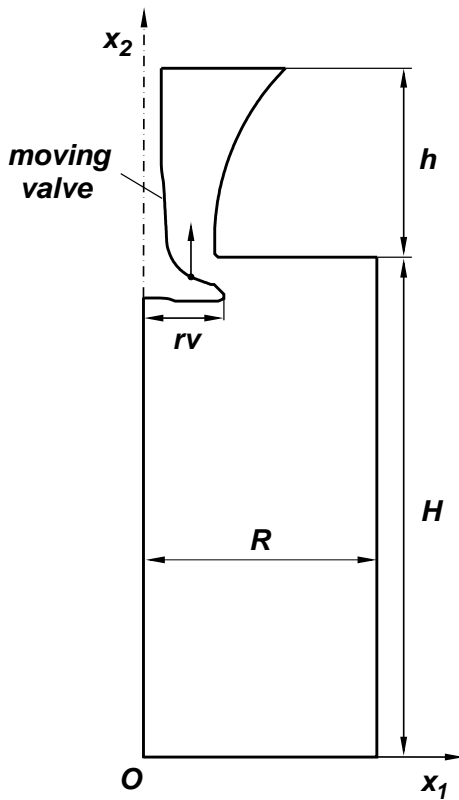


Figure 2.30:
Axisymmetrical flowmeter.

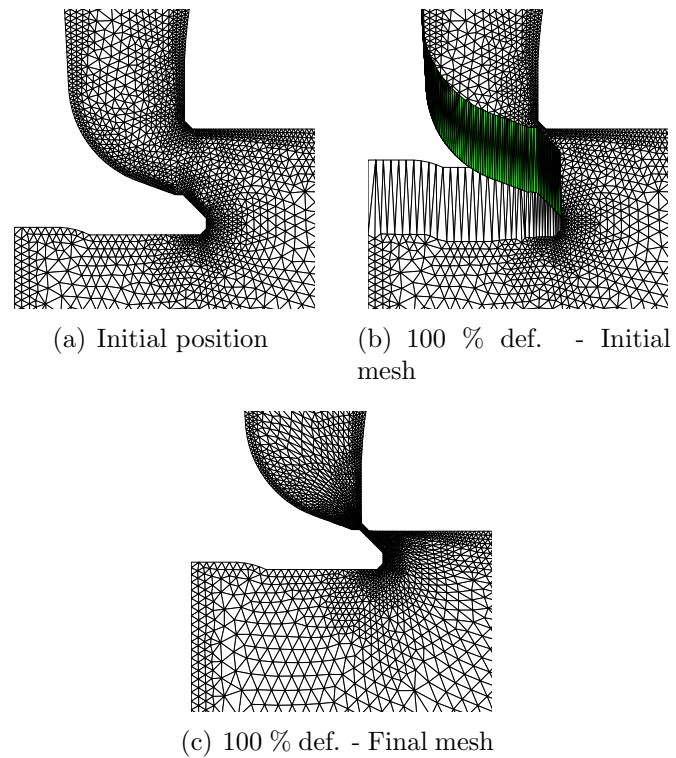


Figure 2.31: Mesh close-up for the axisymmetrical flowmeter.

number of time steps	elapsed time without DP	elapsed time with DP
1	39.486 s.	77.721 s.
2	113.897 s.	79.217 s.
5	121.616 s.	82.272 s.
10	204.734 s.	78.549 s.

Table 2.3: Total elapsed computational time for the axisymmetrical flowmeter test.

strategy works with a real IC engine geometry. Hence, the valve closure is approximated with a small enough minimum valve lift greater than zero.

Some views of the chamber geometry are included in figure 2.32. The cylinder bore is 93.0 mm, the stroke is 103.0 mm, and the geometric compression ratio is 17.8:1. The maximum intake valve lift is 9.0 mm and the maximum exhaust valve lift is 11.0 mm. The valve timing values relative to TDC (when the intake stroke starts) are the following

- Intake Valves Opening (IVO): 5°
- Intake Valves Closing (IVC): 212°
- Exhaust Valve Opening (EVO): 485°
- Exhaust Valve Closing (EVC): 715°

The (topology of the) mesh was generated with the valves placed at a half of the maximum lift from their seats and the crown piston at a half of the stroke from the head.

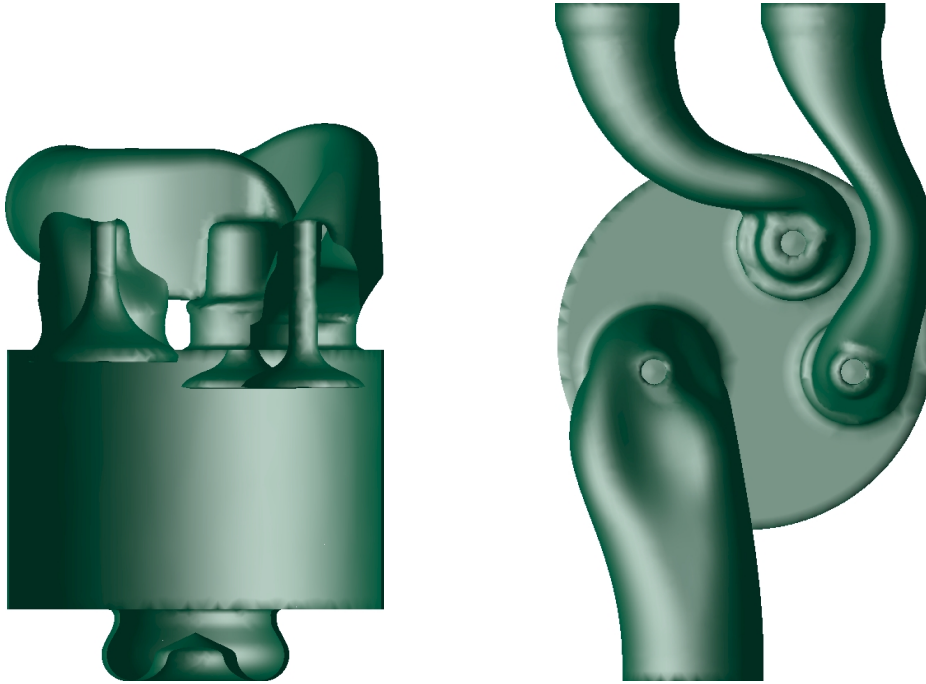


Figure 2.32: Geometry of the combustion chamber of the diesel engine.

The mesh is coarse in the cylinder, but refined in the region of valves, ducts and piston bowl. The mesh has 925700 tetrahedrons and 157630 nodes. The mesh dynamics was solved with the simultaneous mesh untangling and smoothing technique and using the differential predictor presented in section §2.2.2. The time step used corresponds to 1 crank angle degree. Due to the initial mesh was generated with a configuration of the boundary that not corresponds to any instant in the cycle, firstly the valves and piston are moved to the reference crank angle. In this case, such angle was adopted as the TDC when the intake stroke starts.

The mesh quality as a function of the crank angle is plotted in figure 2.33, where the element quality metric applied was q (see equation (2.11)). Figure 2.34 shows the minimum mesh dihedral angle as a function of the crank angle. As could be observed, the mesh quality is approximately constant along the whole cycle. In order to compare, the initial mesh has a quality $q = 0.0211$ and a minimum dihedral angle of 1.4812° .

Figures 2.35, 2.37 and 2.39 show the distribution of the element quality in the domain for three crank angles. These angles correspond to the TDC with the three valves closed (0°), the maximum intake valve lift (108.5°), and maximum exhaust valve lift (600°). The elements with worse quality in the mesh are located between a valve and its seat when the valve approaches the closed position.

2.4 Conclusion

A CMD strategy, which is based on the mesh distortion minimization, was proposed and tested. The element distortion is computed as the reciprocal of an appropriate element quality indicator. In this thesis, a geometric element quality metric is used. The method has proven to be very robust, allowing for valid mesh deformation levels close to the

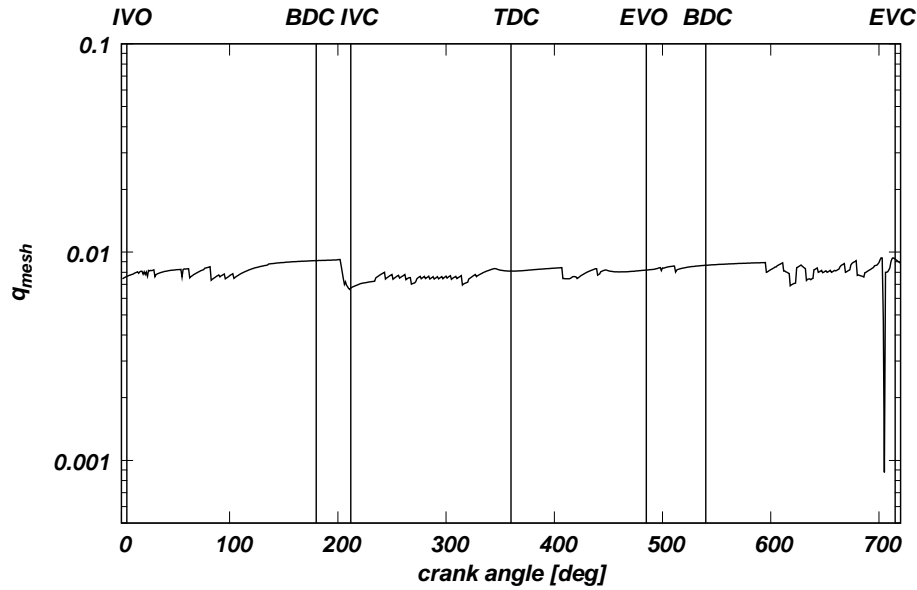


Figure 2.33: Mesh quality as a function of crank angle for the diesel engine.

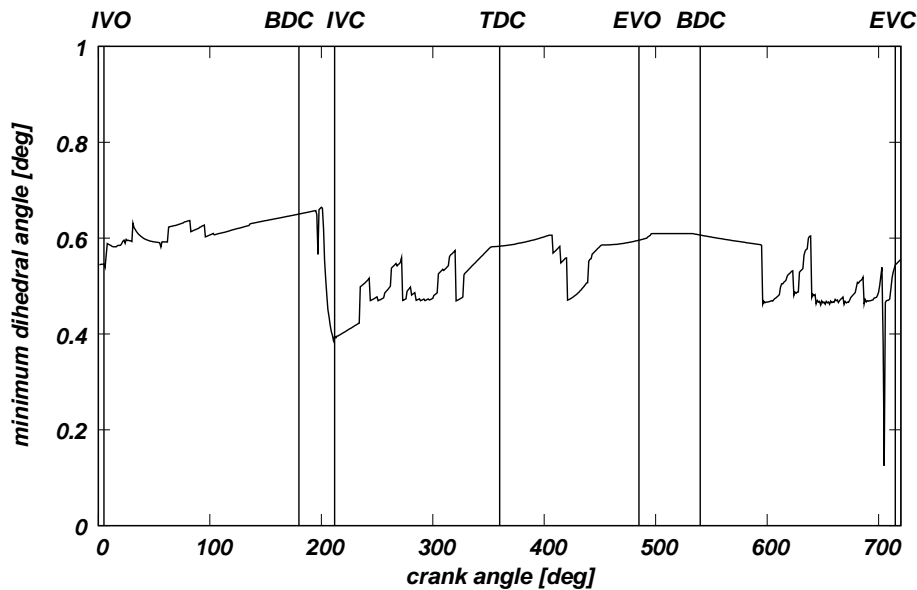


Figure 2.34: Minimum mesh dihedral angle as a function of crank angle for the diesel engine.

maximum extent possible. Several test cases in 2D and 3D were successfully solved, including combustion chamber geometries of IC engines.

The regularization of the element functional leads to a simultaneous mesh untangling and smoothing technique. Basically, the method needs the definition of a decreasing sequence for the regularization parameter (δ). Here, a definition for such a sequence was given and tested in various CMD problems. While this proposal have worked in a good shape for the cases solved, the optimal sequence for computing δ remains an open problem. The number of iterations, and thus the computational cost, can be decreased

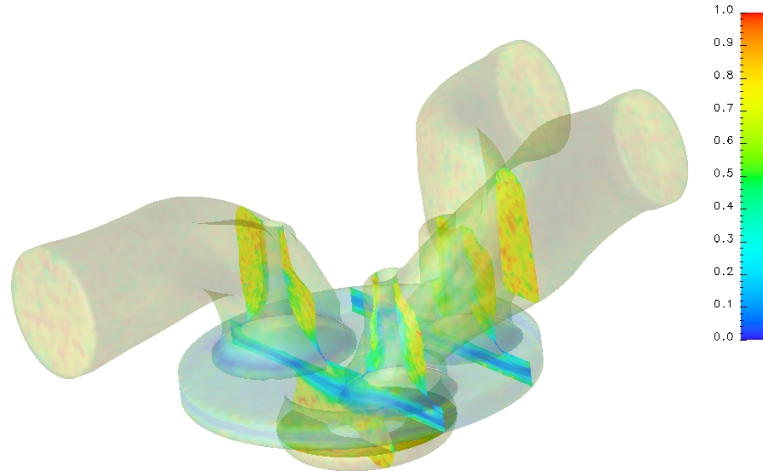


Figure 2.35: Mesh quality field at 0 crank angle degree (TDC) for the diesel engine.

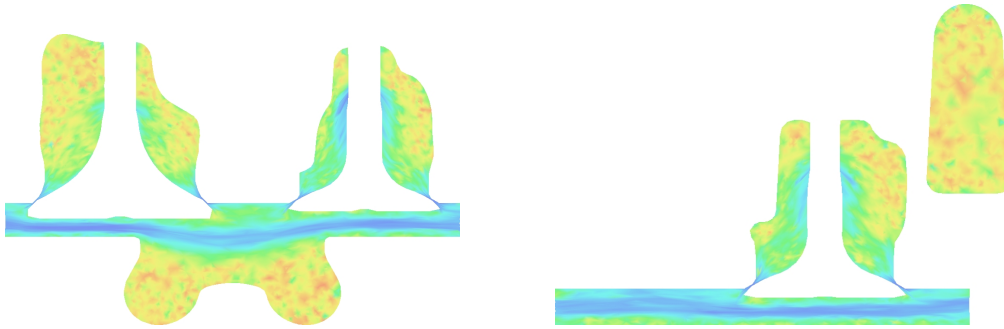


Figure 2.36: Mesh quality field over the transversal planes shown in figure 2.35.

by the use of the proposed solution predictor.

On the other hand, the proposed method can be useful for mesh generation. In this case, the topology is generated in an auxiliary domain in which the mesh may be generated in a structured way. Then, the boundary nodes in that mesh are relocated on the real boundary. This sharp movement of the boundary nodes is similar to the situation faced in mesh dynamics. In appendix B, the generation of conformal meshes is presented.

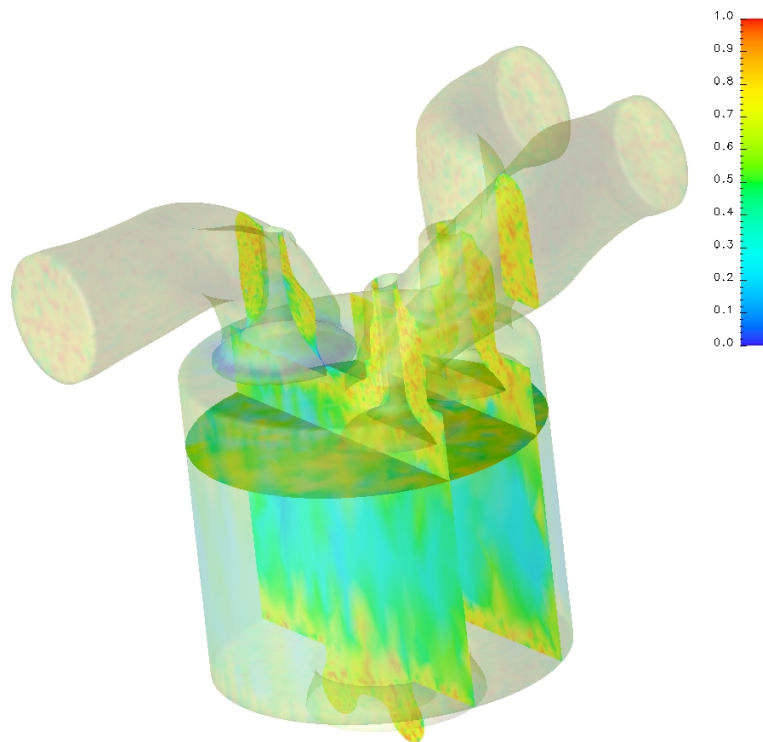


Figure 2.37: Mesh quality field at 108.5 crank angle degree (maximum intake valve lift) for the diesel engine.

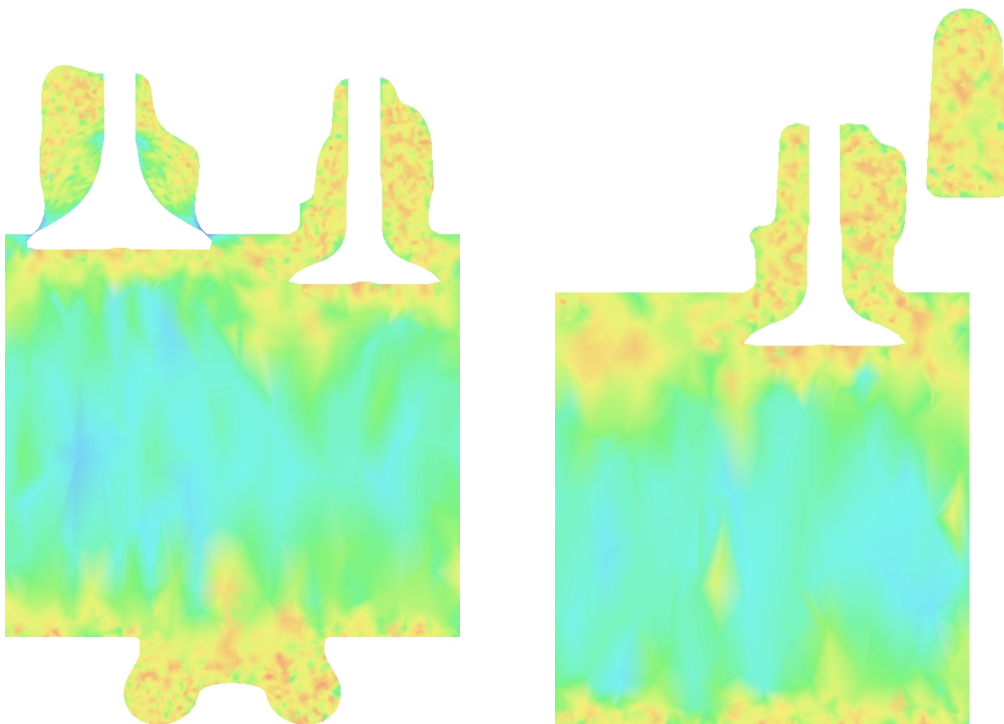


Figure 2.38: Mesh quality field over the transversal planes shown in figure 2.37.

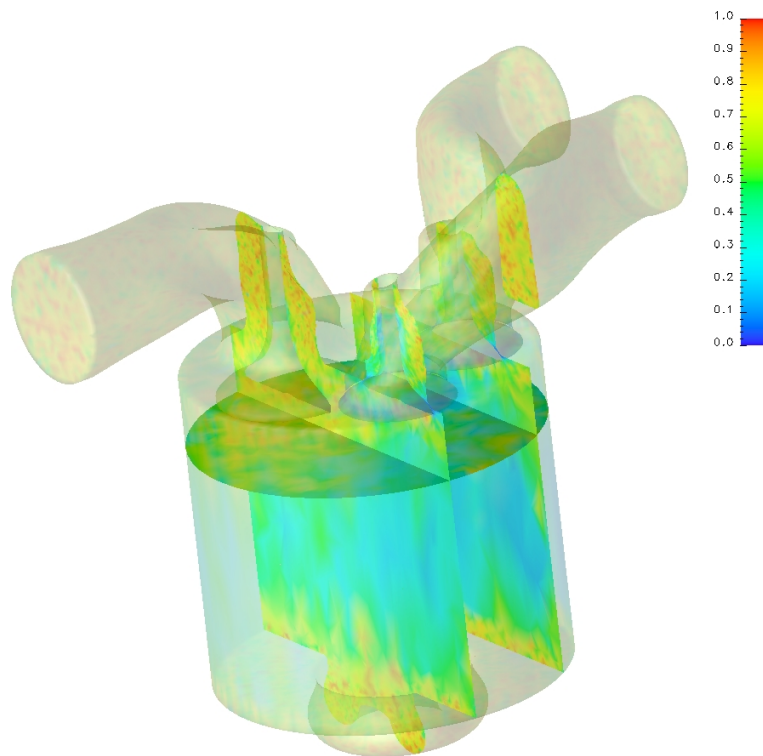


Figure 2.39: Mesh quality field at 600 crank angle degree (maximum exhaust valve lift) for the diesel engine.

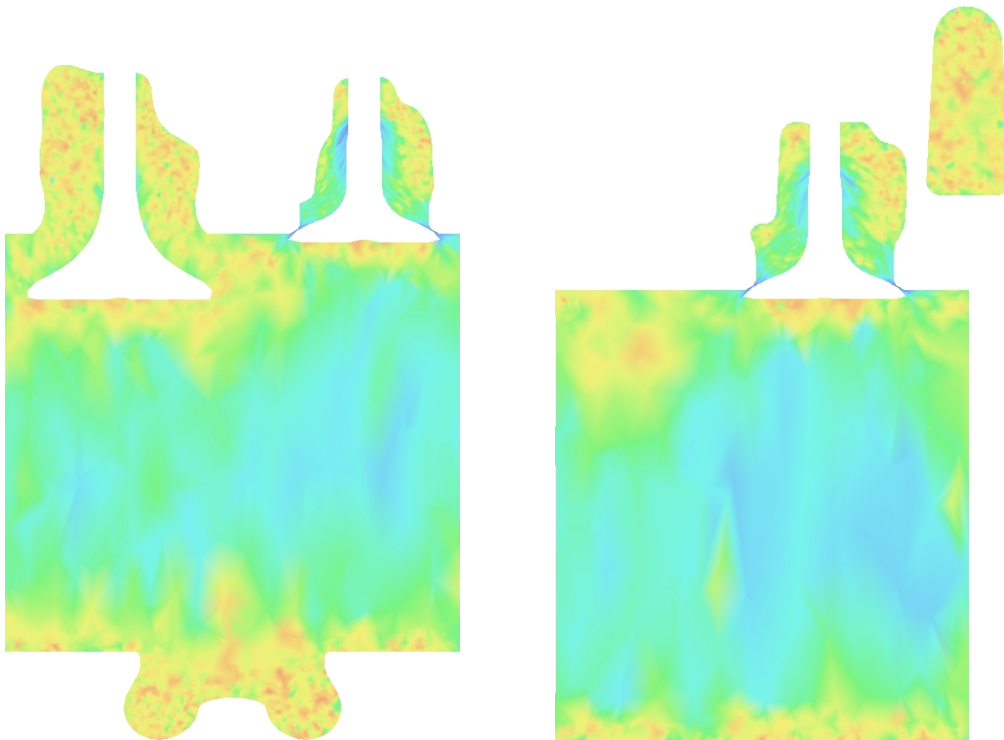


Figure 2.40: Mesh quality field over the transversal planes shown in figure 2.39.

Chapter 3

Resolution of compressible flows in the low Mach number limit

*My quest reverse the time... within the dark mind
Strength in it I find... within the dark mind
Immortal*

Low Mach number (M) flows represent a limit situation in the solution of compressible flows. When the Mach number approaches to zero, the strategies based on density to solve the flow equations suffer severe deficiencies, both in efficiency and accuracy. Turkel *et al.* [82] and Guillard and Viozat [30] have identified that, in the low Mach number limit, the discretized solution of the compressible fluid flow equations may fail to provide an accurate approximation to the incompressible flow equations. In the subsonic regime, when the magnitude of the flow velocity is small in comparison with the acoustic wave speed, dominance of convective terms within the time-dependent equation system renders the system stiff and the solvers could converge slowly [16]. Time-marching procedures may suffer severe stability and accuracy restrictions and become inefficient for low Mach number flow regimes. Here, for explicit schemes, the time step must satisfy the Courant-Friedrichs-Levy (CFL) conditions, where numerical stability considerations lead to small time steps. On the other hand implicit methods suffer from stiffness due to large disparity in the eigenvalues of the system, the condition number of the system of equations being $\mathcal{O}(1/M)$ in the low Mach number limit [16]. There are two main approaches to circumvent this drawback: firstly, the modification of compressible solvers (density-based) downward to low Mach numbers [16, 83, 85, 43, 91]; secondly, extending incompressible solvers (pressure-based) towards the compressible regime [33]. Also there are unified formulations, as the method proposed by Mittal and Tezduyar [50].

The in-cylinder flow in an internal combustion engine is characterized by low Mach numbers, except in the exhaust blowdown phase [32]. Thus, this application limits the strategies for low Mach number flows to those, under a unified formulation, work correctly for (at least) all subsonic Mach numbers.

For density-based methods, two distinct techniques have been proposed to capture solution convergence for low-Mach number regimes: preconditioning and asymptotic meth-

ods. Both techniques achieve rescaling of the system condition number. The asymptotic method introduces a perturbed form of the equations discarding specific terms, so that the physical acoustic waves are replaced by pseudo-acoustic modes. The magnitude of the propagation speeds of these pseudo-acoustic modes are similar to the fluid velocity [16]. Although perturbation procedures are highly robust and applicable for both viscous and inviscid flows, the nature of the perturbation limits their usage, particularly with respect to mixed compressible-incompressible flows.

Preconditioning schemes consist in premultiplying time derivatives by a suitable preconditioning matrix. This scales the eigenvalues of the system to similar orders of magnitude and removes the disparity in wave-speeds, leading to a well-conditioned system [82]. The modified equations have only steady-state solutions in common with the original system (hence, are devoid of true transients). For the application of these methods to unsteady problems the ‘dual-time-stepping’ technique has emerged, where the physical time derivative terms are treated as source and/or reactive terms. During each physical time step, the system of pseudo-temporal equations is advanced in artificial time to reach a pseudo-steady-state [83, 85, 43]. Several local preconditioning matrices have been designed for steady state problems, with very good results [16, 56]. However, recent works have shown that these preconditioners may not be appropriate for unsteady flows [85].

In internal combustion engines applications, one procedure widely used is a semi-implicit method named *acoustic subcycling* [32]. In this method, all terms in the governing equations that are not associated with sound waves are explicitly advanced with a larger time step Δt similar to the used with implicit methods. The terms associated with acoustic waves (the compression terms in the continuity and energy equations and the pressure gradient in the linear momentum equation) are explicitly advanced using a smaller time step δt that satisfies the CFL stability criterion, and of which the main time step is an integral multiple. While this method works well in many internal combustion engines applications where the Mach number is not too low, it is unsuitable for very low Mach number flows since the number of subcycles ($\Delta t/\delta t$) tends to infinity as the Mach number tends to zero. *Pressure gradient scaling* can be used to extend the method to lower Mach numbers [32]. The Mach number is artificially increased to a larger value (but still small in an absolute sense) by multiplying the pressure gradient in the linear momentum equation by a time-dependent scaling factor $1/\alpha^2(t)$, where $\alpha(t) > 1$. This reduces the effective sound speed by the factor α . Coupling pressure gradient scaling with acoustic subcycling reduces the number of subcycles by α [5].

In this thesis, it is proposed to use the method of preconditioning due to its ability to work in a wide range of Mach and Reynolds numbers [58]. This method is applied in conjunction with the ‘dual-time-stepping’ technique. The preconditioning matrix applied is the proposed by Choi and Merkle [16] to solve steady compressible flows and which has been adapted by Nigro *et al.* [55, 56] to the finite element method. Via an eigenvalue analysis of the system of equations, some parameters involved in this matrix are redefined in order to solve transient problems.

3.1 Problem definition and eigenvalues analysis

The eigenvalue analysis of the Navier-Stokes equations for compressible flows is done by using viscous variables, defined as

$$\mathbf{Q} = [p, \mathbf{u}, T]^T \quad (3.1)$$

In the low Mach number limit, it is convenient to make the analysis using the viscous variables instead the conservative ones because, at a fixed time, the density tends to be constant in the space domain.

The preconditioning of equations consists in premultiply the time derivatives by a properly defined matrix. The purpose is to modify the eigenvalues of the system of equations in order to decrease the condition number. Due to this fact, it is only applicable to steady state simulations. In order to apply the preconditioning strategy in unsteady problems, the dual time technique has emerged [42]. In this technique, two times must be considered: the *physical time* (t) and the *pseudo-time* (τ). The solution is obtained by means of preconditioned fully implicit pseudo-transient iterations adding a pseudo-time derivative to equation (1.11). At each physical time step, the system is solved until a pseudo-steady state is reached when $\tau \rightarrow \infty$. If $\mathbf{\Gamma}$ denotes the preconditioning matrix, the system of equations modified by the dual time strategy is written as [83]

$$\mathbf{\Gamma} \frac{\partial \mathbf{U}}{\partial \tau} + \frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_i \frac{\partial \mathbf{U}}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial x_j} \right) + \mathbf{S} \quad (3.2)$$

When the system approaches to the pseudo-steady state, the term $\frac{\partial \mathbf{U}}{\partial \tau}$ tends to zero. Thus, the solution of equation (3.2) at fixed time t tends to the corresponding solution of equation (1.11).

In the viscous variables basis, equation (3.2) is expressed as

$$\mathbf{\Gamma} \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial \tau} + \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial t} + \mathbf{A}_i \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial x_j} \right) + \mathbf{S} \quad (3.3)$$

Let

$$\mathbf{\Gamma}_v = \mathbf{\Gamma} \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} \quad (3.4)$$

the preconditioning matrix in the viscous variables basis. Hence, after premultiplying equation (3.3) by the inverse matrix $\mathbf{\Gamma}_v^{-1}$, the following expression is obtained

$$\frac{\partial \mathbf{Q}}{\partial \tau} + \mathbf{\Gamma}_v^{-1} \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial t} + \mathbf{\Gamma}_v^{-1} \mathbf{A}_i \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial x_i} = \mathbf{\Gamma}_v^{-1} \frac{\partial}{\partial x_i} \left(\mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial x_j} \right) + \mathbf{\Gamma}_v^{-1} \mathbf{S} \quad (3.5)$$

The analysis is done by using the preconditioning matrix proposed by Choi and Merkle [16] for the resolution of steady state problems at the low Mach limit. The pre-

conditioning matrix takes the form

$$\mathbf{\Gamma}_v = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{\beta M_r^2}{u_1} & \rho & 0 & 0 & 0 \\ \frac{\beta M_r^2}{u_2} & 0 & \rho & 0 & 0 \\ \frac{\beta M_r^2}{u_3} & 0 & 0 & \rho & 0 \\ \frac{\rho e + p}{\rho \beta M_r^2} - \delta & \rho u_1 & \rho u_2 & \rho u_3 & \frac{\gamma \rho R}{\gamma - 1} \end{bmatrix} \quad (3.6)$$

where M_r is a reference Mach number, δ is a constant that plays the role of a coefficient of the time derivative of pressure, and $\beta = zc^2$, being $z = \max(z_{\text{inv}}, z_{\text{vis}})$,

$$\begin{aligned} z_{\text{inv}} &= 1 \\ z_{\text{vis}} &= \frac{\alpha_{\text{vis}}(\alpha_{\text{vis}} - 1)}{M_r^2[\alpha_{\text{vis}} - 1 + c^2/(\mathbf{u} \cdot \mathbf{s})^2]} \\ \alpha_{\text{vis}} &= \frac{C\tilde{F}L}{\tilde{\sigma}Re_h} \end{aligned}$$

In the last equation, $C\tilde{F}L$ and $\tilde{\sigma}$ are, respectively, the CFL and the Fourier numbers for the preconditioned system based on pseudo-time discretization parameters, Re_h is the cell Reynolds number based on the characteristic element length h , and \mathbf{s} is the unit vector aligned with the flow velocity. The reference Mach number M_r replaces the Mach number because the term $1/M^2$ in the preconditioning matrix becomes singular in regions where the Mach number tends to zero (*e.g.* stagnation points). In the next section a discussion about the election of M_r is presented (see equation (3.19) for the definition given by Choi and Merkle [16]).

The preconditioned equations are nearly identical to the equations obtained when the method of artificial compressibility is applied, with the addition of the energy conservation equation [16]. Considering null the source vector and a steady flow in the physical time, the individual equations are

$$\begin{aligned} \frac{1}{\beta M_r^2} \frac{\partial p}{\partial \tau} + \nabla \cdot (\rho \mathbf{u}) &= 0 \\ \rho \left(\frac{\partial \mathbf{u}}{\partial \tau} + \mathbf{u} \cdot \nabla \mathbf{u} \right) + \nabla p &= \nabla \cdot \mathbf{T} \\ \rho c_p \left(\frac{\partial T}{\partial \tau} + \mathbf{u} \cdot \nabla T \right) &= \delta \frac{\partial p}{\partial \tau} + \mathbf{u} \cdot \nabla p + \nabla \cdot (\kappa \nabla T) + \nabla \cdot (\mathbf{T} \mathbf{u}) \end{aligned} \quad (3.7)$$

In order to study the eigenvalues of the system of equations, a dispersion analysis on equation (3.5) is done. Let

$$\begin{aligned} \tilde{\mathbf{A}}_{v,i} &= \mathbf{\Gamma}_v^{-1} \mathbf{A}_i \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} \\ \tilde{\mathbf{K}}_{v,ij} &= \mathbf{\Gamma}_v^{-1} \mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} \\ \tilde{\mathbf{S}}_v &= \mathbf{\Gamma}_v^{-1} \mathbf{S} \end{aligned} \quad (3.8)$$

The discretization of physical time derivative is done using backward differences

$$\frac{\partial \mathbf{Q}}{\partial t} \approx \frac{c_t \mathbf{Q}^{n+1} - E(\mathbf{Q}^n, \mathbf{Q}^{n-1}, \dots)}{\Delta t} \quad (3.9)$$

where c_t is a constant that depends on the temporal order of the scheme. Then, the system of equations (3.5) can be written as

$$\frac{\partial \mathbf{Q}}{\partial \tau} + \frac{c_t}{\Delta t} \Gamma_v^{-1} \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} \mathbf{Q} + \tilde{\mathbf{A}}_{v,i} \frac{\partial \mathbf{Q}}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\tilde{\mathbf{K}}_{v,ij} \frac{\partial \mathbf{Q}}{\partial x_j} \right) + \tilde{\mathbf{S}}_v + \frac{1}{\Delta t} \Gamma_v^{-1} \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} E(\mathbf{Q}^n, \mathbf{Q}^{n-1}, \dots) \quad (3.10)$$

In order to simplify the notation, the index $n+1$ for the variables evaluated at the current time is dropped from equation (3.10).

If the source term is assumed to have no effect on the dispersion equation and neglecting the diffusive terms (Euler equations), the following equation is reached

$$\frac{\partial \mathbf{Q}}{\partial \tau} + \frac{c_t}{\Delta t} \Gamma_v^{-1} \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} \mathbf{Q} + \tilde{\mathbf{A}}_{v,i} \frac{\partial \mathbf{Q}}{\partial x_i} = \mathbf{0} \quad (3.11)$$

By introducing a Fourier mode

$$\mathbf{Q} = \mathbf{Q}_0 \exp [i(\mathbf{k}^T \mathbf{x} - \omega \tau)] \quad (3.12)$$

into the equation (3.11), the following equation of dispersion for ω is obtained

$$\left(-i\omega \mathbf{I} + \frac{c_t}{\Delta t} \Gamma_v^{-1} \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} + ik_i \tilde{\mathbf{A}}_{v,i} \right) \mathbf{Q} = \mathbf{0} \quad (3.13)$$

Due to the finite number of mesh nodes, the wavelengths are limited by the grid spacing. To take this filtering into account it is set $\|\mathbf{k}\| = \phi/h$, where h is a measure of the grid spacing and $\phi \in [0, \pi]$. Let $\lambda = \omega/\|\mathbf{k}\|$ the wave speed and $u_k = \mathbf{u}^T \mathbf{k}/\|\mathbf{k}\|$. Writing equation (3.13) as a system of equations $\mathbf{G}\mathbf{Q} = \mathbf{0}$, and looking for solutions for $\mathbf{Q} \neq \mathbf{0}$, achieves \mathbf{G} as

$$\mathbf{G} = -i \frac{c_t}{\|\mathbf{k}\| \Delta t} \Gamma_v^{-1} \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} + \frac{k_i}{\|\mathbf{k}\|} \tilde{\mathbf{A}}_{v,i} - \lambda \mathbf{I} \quad (3.14)$$

after dividing by $i\|\mathbf{k}\|$. The equation $\mathbf{G}\mathbf{Q} = \mathbf{0}$ have a non-trivial solution if $\det \mathbf{G} = 0$. This condition is equivalent to compute the eigenvalues of the matrix

$$\hat{\mathbf{G}} = -i \frac{c_t}{\|\mathbf{k}\| \Delta t} \Gamma_v^{-1} \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} + \frac{k_i}{\|\mathbf{k}\|} \tilde{\mathbf{A}}_{v,i} \quad (3.15)$$

Let the CFL numbers

$$\begin{aligned} CFL_u &= \frac{u_k \Delta t}{h} \\ CFL_c &= \frac{c \Delta t}{h} \end{aligned} \quad (3.16)$$

Then, the eigenvalues of $\hat{\mathbf{G}}$ are

$$\begin{aligned}\lambda(\hat{\mathbf{G}})_{1,2,3} &= u_k(1 - ic_t CFL_u^{-1}) \\ \lambda(\hat{\mathbf{G}})_{4,5} &= \frac{u_k}{2}(1 - ic_t CFL_u^{-1})T_{\pm}\end{aligned}\quad (3.17)$$

where

$$T_{\pm} = (1 + M_r^2 \chi) \pm \sqrt{(1 - M_r^2 \chi)^2 - 4M_r^2 \left[\frac{1}{(iM + c_t CFL_c^{-1})^2} + 1 - \chi \right]}\quad (3.18)$$

In equation (3.18), $\chi = \gamma - (\gamma - 1)\delta$ and the definition of M_r allows to choose among different preconditioning matrices. The eigenvalues of the preconditioned system using the preconditioning matrix designed for steady state solutions can be obtained by means of the definition of M_r proposed by Choi and Merkle [16]

$$M_r = \begin{cases} M_\epsilon & \text{if } M < M_\epsilon \\ M & \text{if } M_\epsilon \leq M < 1 \\ 1 & \text{if } M \geq 1 \end{cases}\quad (3.19)$$

or, equivalently, $M_r = \min(1, \max(M, M_\epsilon))$, where M_ϵ is a cut-off of the Mach number in a neighborhood of stagnation points.

When the ALE strategy is applied (see section §1.1.3), the eigenvalues for the preconditioned system of Euler equations using the dual-time formulation are written as

$$\begin{aligned}\lambda(\hat{\mathbf{G}}_{\text{ALE}})_{1,2,3} &= u_k(1 - ic_t CFL_u^{-1}) - w_k \\ \lambda(\hat{\mathbf{G}}_{\text{ALE}})_{4,5} &= \frac{1}{2}[u_k(1 - ic_t CFL_u^{-1}) - w_k]T_{\pm}^{\text{ALE}}\end{aligned}\quad (3.20)$$

being

$$T_{\pm}^{\text{ALE}} = (1 + M_r^2 \chi) \pm \sqrt{(1 - M_r^2 \chi)^2 - 4M_r^2 \left[\frac{1}{(i\tilde{M} + c_t CFL_c^{-1})^2} + 1 - \chi \right]},\quad (3.21)$$

$\tilde{M} = (u_k - w_k)/c$, and $w_k = \mathbf{w}^T \mathbf{k} / \|\mathbf{k}\|$.

For viscous flows, it is not longer easy to compute the eigenvalues of the system. Therefore, many authors use approximations that depend on the Reynolds number. In the following section, a numerical analysis of the viscous flow case is presented.

3.1.1 Preconditioning strategies

The condition number of the system is defined as

$$CN = \frac{\max(|\lambda_i|)}{\min(|\lambda_i|)} = \frac{\max(1, |T_+|, |T_-|)}{\min(1, |T_+|, |T_-|)}\quad (3.22)$$

Figure 3.1 shows the condition number of the system of Euler equations as a function of the Courant number CFL_c , where $c_t = 1$, $\delta = 1$ ($\chi = 1$), $M_\epsilon = 1 \times 10^{-6}$ and $M = 1 \times 10^{-3}$. The preconditioning matrix for steady state solutions (SP, Steady Preconditioner) corresponds

to the definition of M_r given by equation (3.19). For unsteady problems, Vigneron *et al.* [85] suggest

$$M_r = \min(1, \max(\sqrt{M^2 + CFL_c^{-2}}, M_\epsilon)) \quad (3.23)$$

This definition is named UP (Unsteady Preconditioner) on figure 3.1. In addition, the condition number of the non-preconditioned system (NP) is included, which, in the case $\delta = 1$, is obtained when $M_r = \chi^{-1/2}$ in equation (3.18).

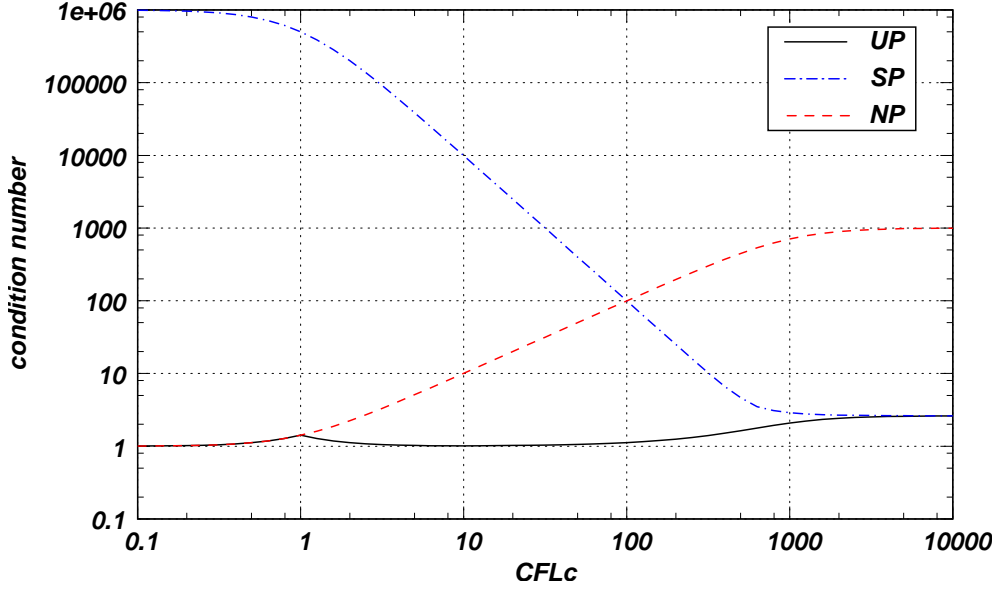


Figure 3.1: Condition number as a function of Courant number CFL_c with $M = 1 \times 10^{-3}$ and $M_\epsilon = 1 \times 10^{-6}$ for the inviscid case.

As it is shown in the figure, when the ‘unsteady’ preconditioning is applied the condition number of the system is $\mathcal{O}(1)$ for all CFL_c numbers in the inviscid case. For viscous flows, the eigenvalues of the system were computed numerically adopting firstly $z = z_{\text{inv}}$ for the definition of β in equation (3.6). Figure 3.2 shows the condition number of the system for several Reynolds numbers using the inviscid reference Mach number ($M_r|_{\text{inv}}$) defined in equation (3.23). For Reynolds numbers lower than 1 the condition number of the system increase significantly, specially for high CFL_c numbers. Merkle [49] proposed to use the following approximation

$$M_r = \begin{cases} M_r|_{\text{inv}} & \text{if } Re > 1 \\ M/Re & \text{if } Re < 1 \end{cases} \quad (3.24)$$

In figure 3.3 the condition number of the system as a function of Reynolds number for $CFL_c = 1 \times 10^4, 1 \times 10^2$ is plotted. Two pairs of curves are shown in the figure, one of them corresponds to the reference Mach number for inviscid flows and the other one was obtained by using the approximation (3.24). The approximation could not be good enough for low CFL_c numbers, as it is shown in figure 3.3. In this thesis, no correction is used for the viscous flow case since the original definition of the preconditioning matrix includes some control of the time step in the viscous regions via the β parameter. From the viewpoint of the conditioning of the system, the β parameter allows to keep the condition number $\mathcal{O}(1)$ for all Reynolds and CFL_c numbers, as it is shown in figure 3.4 for $M = 1 \times 10^{-3}$.

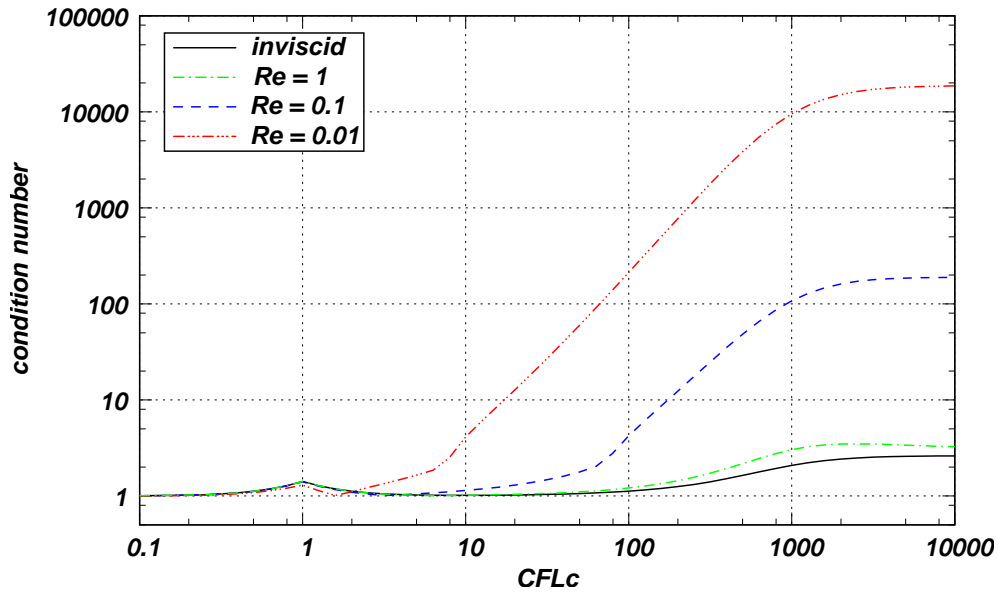


Figure 3.2: Condition number as a function of Courant number CFL_c for several Reynolds numbers with the reference Mach number for the inviscid case and $M = 1 \times 10^{-3}$.

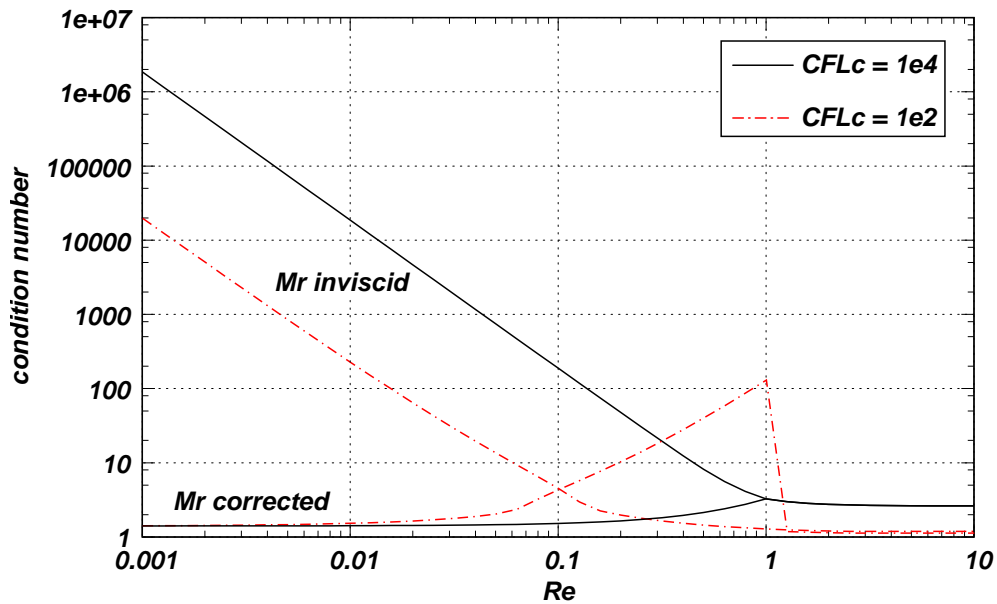


Figure 3.3: Condition number as a function of Reynolds number with $CFL_c = 1 \times 10^4$, 1×10^2 and $M = 1 \times 10^{-3}$.

3.2 Numerical implementation

3.2.1 Finite element formulation

In order to simplify the notation, the mesh velocity is considered null since when the ALE technique is used, only the advective jacobians are modified in the system of equations (see equation (1.26)). The variational formulation of the problem is written as follows:

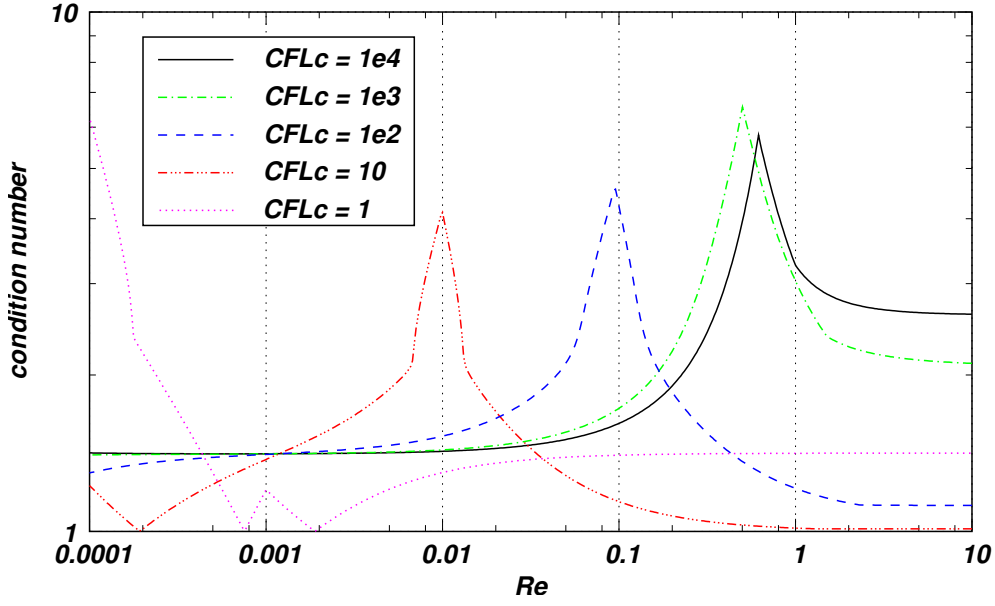


Figure 3.4: Condition number as a function of Reynolds number for several CFL_c numbers, with $M = 1 \times 10^{-3}$.

Find $\mathbf{U}^h \in \mathcal{S}^h$ such that $\forall \mathbf{W}^h \in \mathcal{V}^h$

$$\begin{aligned}
& \int_{\Omega} \mathbf{W}^h \cdot \left(\Gamma^h \frac{\partial \mathbf{U}^h}{\partial \tau} + \frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} \right) d\Omega + \int_{\Omega} \frac{\partial \mathbf{W}^h}{\partial x_i} \cdot \mathbf{K}_{ij}^h \frac{\partial \mathbf{U}^h}{\partial x_j} d\Omega \\
& + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \boldsymbol{\tau}' (\mathbf{A}_k^h)^T \frac{\partial \mathbf{W}^h}{\partial x_k} \cdot \left[\Gamma^h \frac{\partial \mathbf{U}^h}{\partial \tau} + \frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} - \frac{\partial}{\partial x_i} \left(\mathbf{K}_{ij}^h \frac{\partial \mathbf{U}^h}{\partial x_j} \right) - \mathbf{S} \right] d\Omega^e \\
& + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \delta'_{sc} \frac{\partial \mathbf{W}^h}{\partial x_i} \cdot \frac{\partial \mathbf{U}^h}{\partial x_i} d\Omega^e = \int_{\Omega} \mathbf{W}^h \cdot \mathbf{S} d\Omega + \int_{\Gamma^h} \mathbf{W}^h \cdot \mathbf{f} d\Gamma
\end{aligned} \tag{3.25}$$

where spaces \mathcal{S}^h and \mathcal{V}^h are defined by equation (1.29). The primes in $\boldsymbol{\tau}'$ and δ'_{sc} indicate that a redefinition of these stabilization parameters must be done due to the preconditioning.

The derivative with respect to τ is discretized using the backward Euler difference scheme

$$\frac{\partial \mathbf{U}}{\partial \tau} \approx \frac{\mathbf{U}^{n+1,m+1} - \mathbf{U}^{n+1,m}}{\Delta \tau} \tag{3.26}$$

Notice the indexes used to indicate each time level: $n + 1$ is the current physical time step and $m + 1$ is the current pseudo-time step. In addition, an implicit formulation is proposed in both, t and τ .

The definition of the matrix of intrinsic time scale ($\boldsymbol{\tau}'$) is very important in order to stabilize the numerical scheme correctly. For this formulation, it is proposed to apply the SUPG strategy, *i.e.* to stabilize the numerical scheme considering the advective part of the system only. From equation (3.5) and using the definitions given in (3.8), the system

of Navier-Stokes equations expressed in the viscous variables basis is written as

$$\frac{\partial \mathbf{Q}}{\partial \tau} + \mathbf{\Gamma}_v^{-1} \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial t} + \tilde{\mathbf{A}}_{v,i} \frac{\partial \mathbf{Q}}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\tilde{\mathbf{K}}_{v,ij} \frac{\partial \mathbf{Q}}{\partial x_j} \right) + \tilde{\mathbf{S}}_v \quad (3.27)$$

The numerical diffusivity introduced by the SUPG method in the inviscid case is [36]

$$\tilde{\mathbf{K}}_v^{\text{num}} = \tilde{\mathbf{A}}_v \tilde{\boldsymbol{\tau}}_v \tilde{\mathbf{A}}_v \quad (3.28)$$

where $\tilde{\boldsymbol{\tau}}_v$ is the matrix of intrinsic time scale in the viscous variables basis. In the conservative variables basis this matrix is expressed as

$$\boldsymbol{\tau}' = \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} \tilde{\boldsymbol{\tau}}_v \frac{\partial \mathbf{Q}}{\partial \mathbf{U}} \mathbf{\Gamma}^{-1} = \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} \tilde{\boldsymbol{\tau}}_v \mathbf{\Gamma}_v^{-1} \quad (3.29)$$

There are several approaches to compute the matrix $\tilde{\boldsymbol{\tau}}_v$. One of them is the definition given by Hughes and Mallet [36] adapted to the preconditioned system

$$\tilde{\boldsymbol{\tau}}_v = \|\tilde{\mathbf{B}}_v\|^{-1} \quad (3.30)$$

being

$$\tilde{\mathbf{B}}_v = \frac{\partial \xi_i}{\partial x_j} \tilde{\mathbf{A}}_{v,j} \quad (3.31)$$

the preconditioned advective jacobians transformed to the master element, in which ξ_i represents the master element coordinates.

Another option is the proposed by Le Beau *et al.* [40]. For the preconditioned system, this proposal is expressed as

$$\tilde{\boldsymbol{\tau}}_v = \frac{h}{2 \max |\lambda(\tilde{\mathbf{A}}_v)|} \mathbf{I} \quad (3.32)$$

In this thesis, the definition given by equation (3.32) is used to stabilize the numerical scheme plus a correction due to viscous effects (see, for instance, the work by Mittal and Tezduyar [50]). The eigenvalues of the advective jacobian matrix $\tilde{\mathbf{A}}_v$ are

$$\begin{aligned} \lambda(\tilde{\mathbf{A}}_v)_{1,2,3} &= u \\ \lambda(\tilde{\mathbf{A}}_v)_{4,5} &= \frac{u(1 + \beta M_r^2 \chi / c^2 \pm \tilde{c})}{2} \end{aligned} \quad (3.33)$$

where $\tilde{c}^2 = u^2 \left(1 + \frac{\beta M_r^2 \chi}{c^2} \right)^2 + 4\beta M_r^2 \left(1 - \frac{u^2}{c^2} \right)$.

Considering the non preconditioned system, for the wave with propagation speed $\lambda(\mathbf{A})_j$ the numerical diffusion introduced by the stabilization strategy given by equation (3.32) is proportional to

$$\frac{h \lambda^2(\mathbf{A})_j}{2 \max_i |\lambda(\mathbf{A})_i|}$$

Thus, at the low Mach number limit, there are sub-stabilized modes due to the disparity in the wavespeeds. This sub-stabilization could leads to spurious numerical oscillations in the solution.

Regarding for the shock capturing operator, it is used a modification of the ‘standard’ definition given by (1.33). The characteristic velocity is computed as $u_{\text{char}} = \|\mathbf{u}\| + \tilde{c}$, and the operator is affected by a tuning coefficient (\tilde{a})

$$\delta'_{\text{sc}} = \tilde{a} \frac{h_{JGN}}{2} (\|\mathbf{u}\| + \tilde{c}) \left(\frac{\|\nabla \rho^h\| h_{JGN}}{\rho_{\text{ref}}} \right)^{\beta^*} \quad (3.34)$$

3.2.2 Dynamic boundary conditions

As with the stabilization in the Finite Element Method, the dynamic boundary conditions must be reformulated for the preconditioned system of equations. The idea here is to follow the proposal by Storti *et al.* [72] but applied to the equation (3.10) expressed in viscous variables basis. In multidimensional problems a simplified 1D analysis in the normal direction to the boundary is done by considering the projection of the advective jacobians onto this direction, as follows

$$\tilde{\mathbf{A}}_{v,n} = \tilde{\mathbf{A}}_{v,i} n_i \quad (3.35)$$

where n_i are the components of the unit vector normal to the local boundary. After diagonalization of the projected jacobian

$$\tilde{\mathbf{A}}_{v,n} = \tilde{\mathbf{M}}_{v,n} \tilde{\mathbf{\Lambda}}_{v,n} \tilde{\mathbf{M}}_{v,n}^{-1} \quad (3.36)$$

with $\tilde{\mathbf{\Lambda}}_{v,n} = \text{diag}[\lambda(\tilde{\mathbf{A}}_{v,n})]$, the projection matrices onto the right/left-going characteristics modes in the diagonal basis (\mathbf{V} variables) are obtained by

$$\begin{aligned} (\mathbf{\Pi}_{Vn}^-)_{ij} &= \begin{cases} 1 & \text{if } i = j \text{ and } \lambda_i(\tilde{\mathbf{A}}_{v,n}) < 0 \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{\Pi}_{Vn}^- + \mathbf{\Pi}_{Vn}^+ &= \mathbf{I} \end{aligned} \quad (3.37)$$

The projection matrices in the viscous variables basis are computed changing the basis

$$\mathbf{\Pi}_{Qn}^\pm = \tilde{\mathbf{M}}_{v,n} \mathbf{\Pi}_{Vn}^\pm \tilde{\mathbf{M}}_{v,n}^{-1} \quad (3.38)$$

Finally, coming back to the \mathbf{U} basis, the projection matrices expressed in this basis are

$$\mathbf{\Pi}_{Un}^\pm = \frac{\partial \mathbf{U}}{\partial \mathbf{Q}} \mathbf{\Pi}_{Qn}^\pm \frac{\partial \mathbf{Q}}{\partial \mathbf{U}} \quad (3.39)$$

As explained in section §1.2.3, the dynamic boundary conditions are applied using Lagrange multipliers. Therefore, for a node i on the boundary, the system of equations to be solved is

$$\begin{aligned} \mathbf{\Pi}_{Un}^-(\hat{\mathbf{U}})(\mathbf{U}_i - \hat{\mathbf{U}}) + \mathbf{\Pi}_{Un}^+(\hat{\mathbf{U}})\mathbf{U}_{\text{lm}} &= \mathbf{0} \\ \mathbf{R}_i(\mathbf{U}) + \mathbf{\Gamma} \mathbf{\Pi}_{Un}^-(\hat{\mathbf{U}})\mathbf{U}_{\text{lm}} &= \mathbf{0} \end{aligned} \quad (3.40)$$

The state $\hat{\mathbf{U}}$ is adopted according to the direction of the normal flow respect to the boundary, as in section §1.2.3.

3.3 Results

The preconditioning strategy presented in this chapter to solve compressible flows at the low Mach number limit was applied to several problems. Among the solved problems, there are steady and unsteady incompressible flows with moving domains. The purpose of these problems was to compare the preconditioned-system solution and the solution obtained by using a standard incompressible Navier-Stokes code. Furthermore, a test inherently compressible, which is the simulation of the in-cylinder flow in an opposed-piston IC engine under cold conditions, was solved.

The pseudo-time step is increased during the pseudo-transient following the rule

$$\Delta\tau^{m+1} = \Delta\tau^0 \frac{\|\mathbf{R}^{n,0}\|}{\|\mathbf{R}^{n,m}\|} \quad (3.41)$$

where $\mathbf{R}^{n,m}$ is the global residue and $\Delta\tau^0$ is an initial pseudo-time step defined by the user.

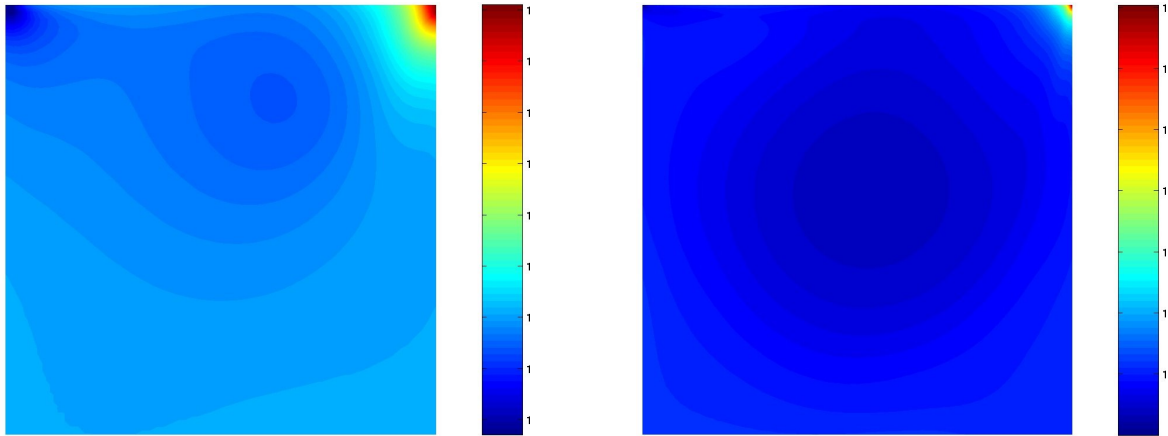
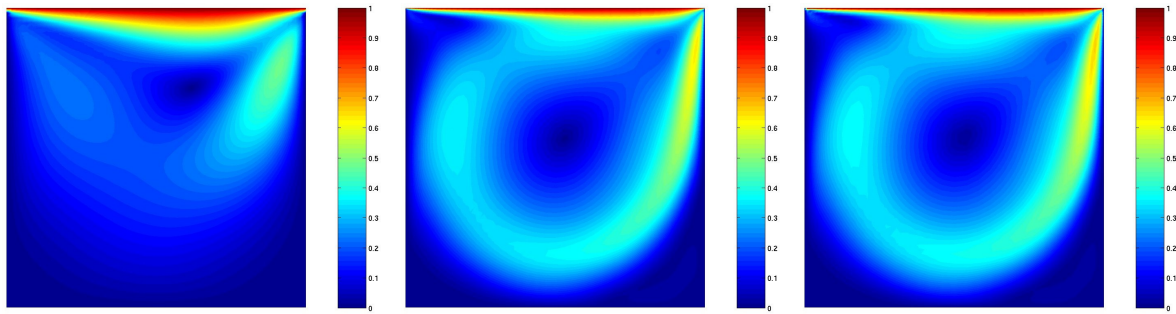
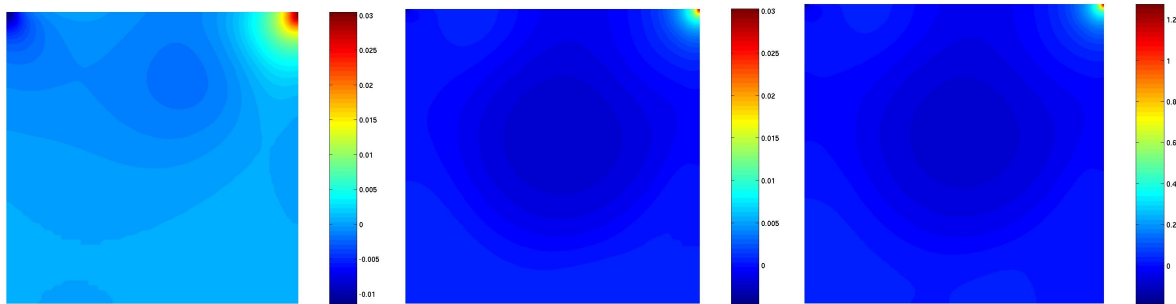
3.3.1 Flow in a lid driven cavity

This test has served as a benchmark for the Navier-Stokes equations for incompressible flow for decades. The problem consists in a fluid into a square cavity whose top wall moves with uniform velocity. Two Reynolds numbers cases 100 and 1000 are considered and a 60×60 quadrangles mesh is employed. A uniform grid is used for $\text{Re} = 100$, and a stretched grid is used for $\text{Re} = 1000$, with a ratio of 1:10 between elements near the wall and elements in the central region of the domain. The Mach number of the moving lid is 4.5×10^{-4} . Wall boundary conditions are no slip and constant temperature. Initially, the fluid is at rest and its pressure and temperature are constants. The test was solved with the preconditioning strategy presented in this chapter and also with a standard incompressible Navier-Stokes (NSI) code. The incompressible Navier-Stokes equations were solved using a stabilized finite element SUPG-PSPG [15] [78] (Pressure-Stabilizing/Petrov-Galerkin) method (see appendix A). Figures 3.5 to 3.7 show the density, the magnitude of the velocity, and the pressure perturbation, respectively. The pressure perturbation is computed as $p - \bar{p}$, with $\bar{p} = 1 \times 10^5$ Pa. Notice that the solution obtained is smooth and with no numerical oscillations.

In order to verify the accuracy of the presented method, the velocity profiles at vertical and horizontal centerlines of the cavity ($x_1 = 0.5$ and $x_2 = 0.5$, respectively) are compared with two numerical solutions of the incompressible Navier-Stokes equations, one of them by Ghia *et al.* [29] and the other one by using the FEM formulation presented in appendix A (implemented in the PETSc-FEM code). The u_1 velocity is compared at the vertical centerline of the cavity, and the u_2 velocity is compared at the horizontal centerline of the cavity. Figures 3.8 and 3.9 show the results, where good agreement can be observed.

3.3.2 Flow in a channel with a moving indentation

This test case consists in a flow through a 2D channel with a moving indentation, which has been studied experimentally by Pedley and Stephanoff [62], and numerically by Ralph and Pedley [64] and by Demirdžić and Perić [21]. Figure 3.10 shows a scheme of the channel

(a) $Re = 100$ - UP(b) $Re = 1000$ - UPFigure 3.5: Density field ($[\text{kg}/\text{m}^3]$) for flow in a lid driven cavity.(a) $Re = 100$ - UP(b) $Re = 1000$ - UP(c) $Re = 1000$ - NSIFigure 3.6: Magnitude of the velocity field ($[\text{m}/\text{s}]$) for flow in a lid driven cavity.(a) $Re = 100$ - UP(b) $Re = 1000$ - UP(c) $Re = 1000$ - NSIFigure 3.7: Pressure perturbation field ($[\text{Pa}]$) for flow in a lid driven cavity.

geometry. The shape of the indentation was taken from Pedley and Stephanoff [62], whose specified the following analytic function which approximately fit the real shape used in the experiment

$$x_2(x_1) = \begin{cases} H & \text{for } 0 < x_1 < c_1 \\ 0.5H\{1 - \tanh[a(x_1 - c_2)]\} & \text{for } c_1 < x_1 < c_3 \\ 0 & \text{for } x_1 > c_3 \end{cases}$$

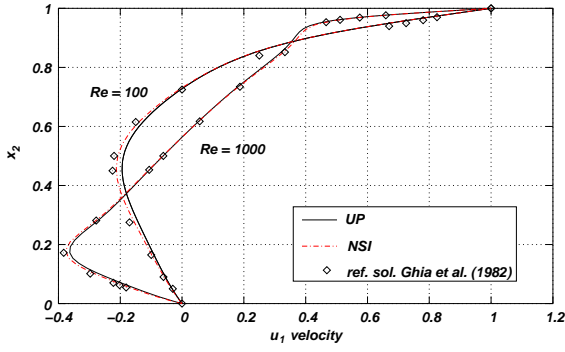


Figure 3.8: Comparison of u_1 velocity component at vertical centerline of the cavity with numerical solution by Ghia *et al.* [29].

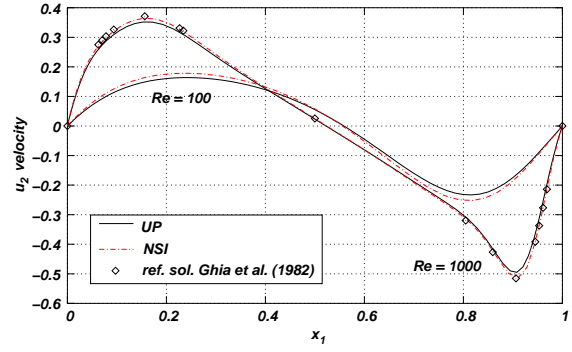


Figure 3.9: Comparison of u_2 velocity component at horizontal centerline of the cavity with numerical solution by Ghia *et al.* [29].

where $a = 4.14$, $c_1 = 4b$, $c_3 = 6.5b$, $c_2 = 0.5(c_1 + c_3)$, and

$$H = 0.5H_{\max} [1 - \cos(2\pi t^*)] \quad (3.42)$$

being $t^* = \frac{t - t_0}{\Upsilon}$.

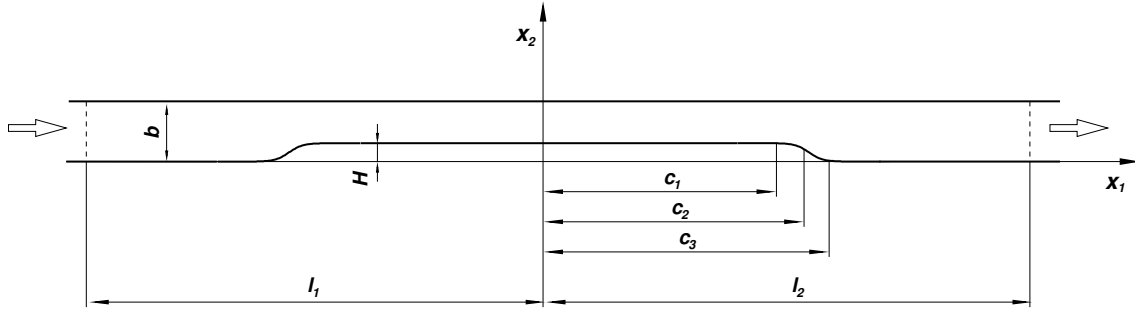


Figure 3.10: Geometry of the channel (not to scale): $b = 1$ cm, $l_1 = 9.85$ cm, $l_2 = 18.0$ cm.

Here b is the channel height, Υ is the oscillation period and $H_{\max} = 0.38b$ specifies the maximum blockage of the channel cross-section at $t^* = 0.5$. The geometry is symmetric around $x_1 = 0$. The Strouhal number based on the channel height, the bulk velocity $U = 2/3u_{1,\max}$ and the oscillation period,

$$St = \frac{b}{U\Upsilon}, \quad (3.43)$$

is 0.037. The Reynolds number based on the same reference quantities is 507. At the initial time $t = t_0$ the flow is assumed to be fully developed (Poiseuille flow). The maximum velocity is $u_{1,\max} = 1.5$ m/s. The velocity profile at the inlet cross-section is taken to remain constant throughout the cycle. Also a unit density is imposed at the inlet cross-section. At the other channel end, dynamic boundary conditions with the ULSAR strategy were imposed, as presented in section §3.2.2. Walls are assumed isothermic and

no slip boundary condition is imposed on them. Mesh dynamics was solved applying the method described in chapter 2. The mesh used has 12.4K triangular elements and 6.8K nodes. The physical time step adopted in the simulation was $\Delta t = \Upsilon/200$, and the initial pseudo-time step $\Delta\tau^0 = 1$ (see equation (3.41)).

Figures 3.11, 3.12 and 3.13 show the density, the magnitude of the velocity, and the pressure perturbation (with $\bar{p} = 1 \times 10^5$ Pa) at times $t^* = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$ and 1. These results were obtained by using the preconditioning strategy presented above.

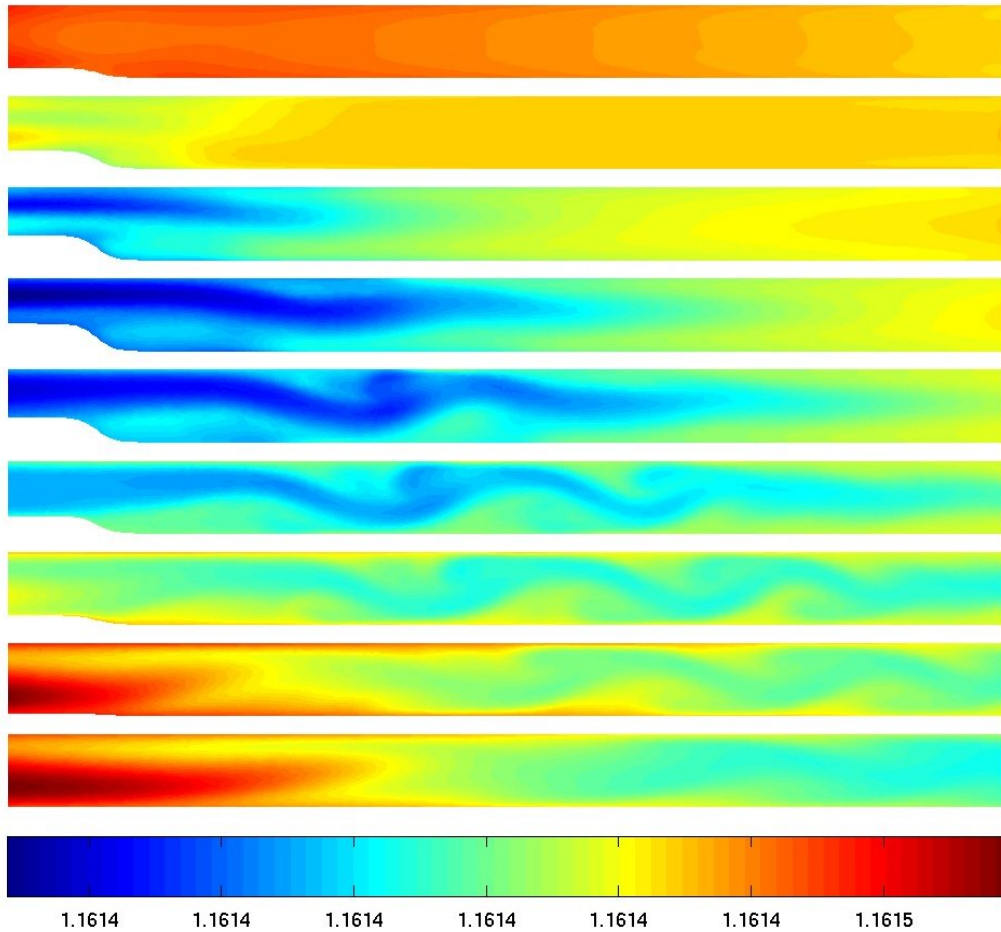


Figure 3.11: Density field ($[\text{kg}/\text{m}^3]$) for the flow in a channel with a moving indentation computed by using the UP strategy. From top to bottom, times $t^* = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$ and 1.

Table 3.1 compares the solution obtained by means of the UP strategy and two solutions of the Navier-Stokes equations for incompressible flow, one obtained using the PETSc-FEM code and the other extracted from the bibliography. The comparison is done taking the maximum velocity differences and the time in which appear the three first vortices.

Figure 3.14 shows the positions of the first three eddies center as a function of time. In that figure, the experimental data reported by Pedley and Stephanoff [62], and the numerical solutions of the UP technique and the Navier-Stokes equations for incompressible

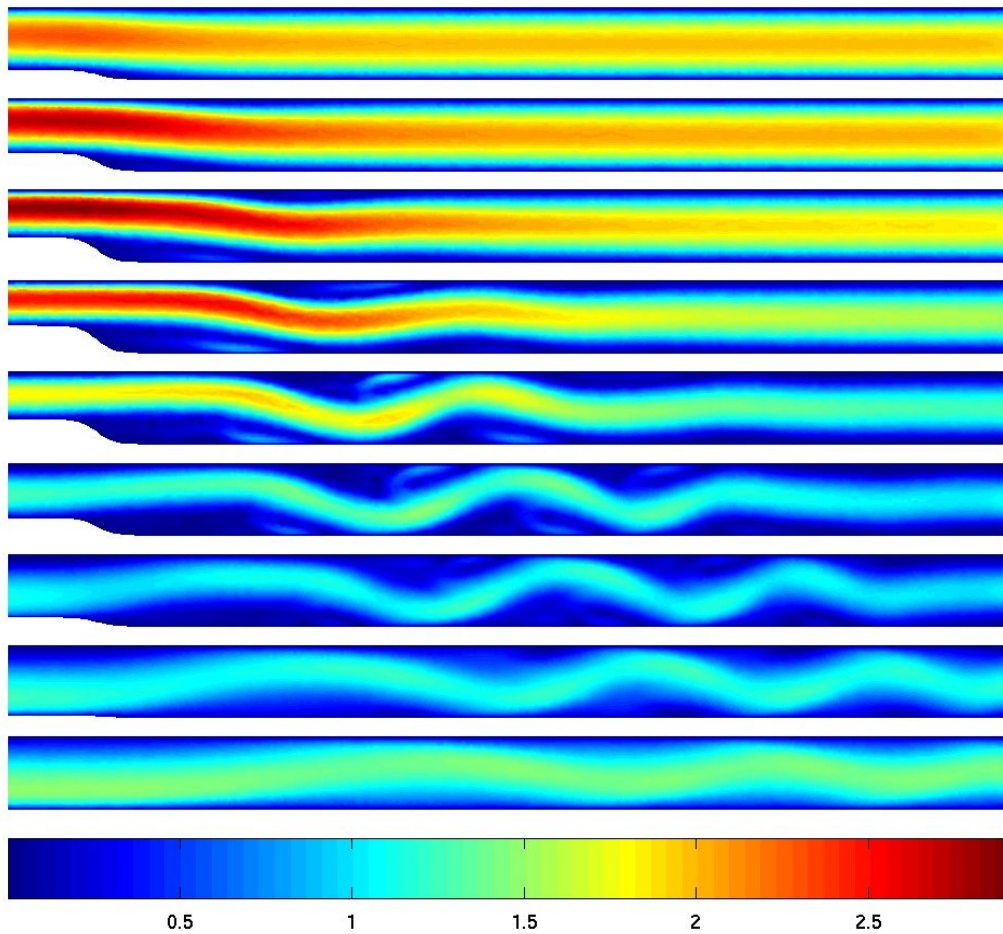


Figure 3.12: Magnitude of the velocity field ([m/s]) for the flow in a channel with a moving indentation computed by using the UP strategy. From top to bottom, times $t^* = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$ and 1 .

	UP	López <i>et al.</i> [45]	Demirdžić and Perić [21]
Max. velocity [m/s]	2.916 at $t^* = 0.37$	2.931 at $t^* = 0.38$	2.645 at $t^* = 0.4$
1 st vortex	$t^* = 0.22$	$t^* = 0.23$	$t^* = 0.2-0.25$
2 nd vortex	$t^* = 0.345$	$t^* = 0.35$	$t^* = 0.35-0.4$
3 rd vortex	$t^* = 0.425$	$t^* = 0.42$	$t^* = 0.45$

Table 3.1: Comparison of results obtained by using the UP strategy and two solutions of the incompressible Navier-Stokes equations for the channel with a moving indentation test.

flow [45] are included. According to Pedley and Stephanoff [62], the abscissa is defined as

$$x_1^* = \frac{(x_1 - c_1)(10St)^{1/3}}{b} \quad (3.44)$$

Also this test was solved using the non-preconditioned system of equations, with the same conditions as presented above. In figures 3.15, 3.16 and 3.17 the density field, the velocity magnitude field and the pressure perturbation field are shown at $t^* = 0.2, 0.4, 0.6, 0.8$ and 1 respectively. The non-preconditioned solution does not represent the behavior

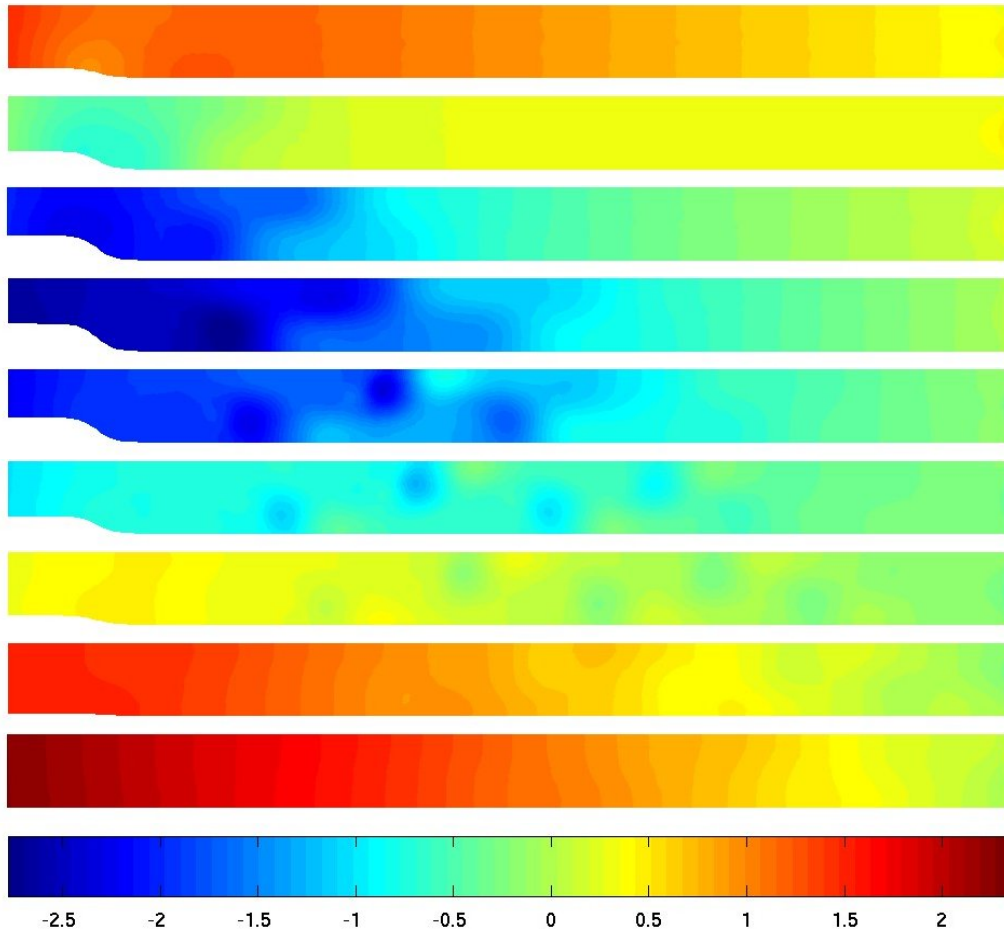


Figure 3.13: Pressure perturbation field ([Pa]) for the flow in a channel with a moving indentation computed by using the UP strategy. From top to bottom, times $t^* = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$ and 1 .

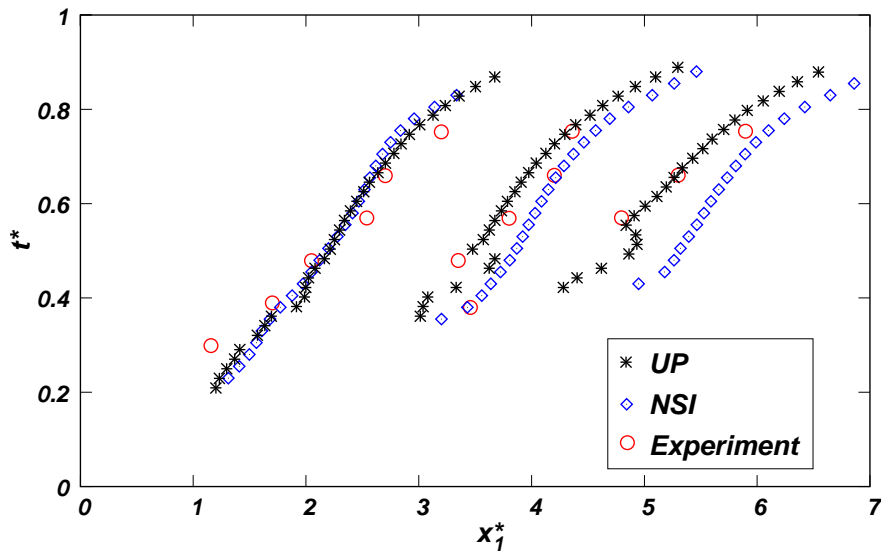


Figure 3.14: Comparison of predicted and experimentally observed positions of first three vortices center.

of the flow since it does not produce the different vortices experimentally observed. The pressure field has numerical oscillations, which can not be observed in figure 3.17 due to the scale used. Thus, the pressure perturbation field at $t^* = 0.5$ is shown in figure 3.18 with an appropriate color scale. Figure 3.19 presents the same field represented as a 3D plot.

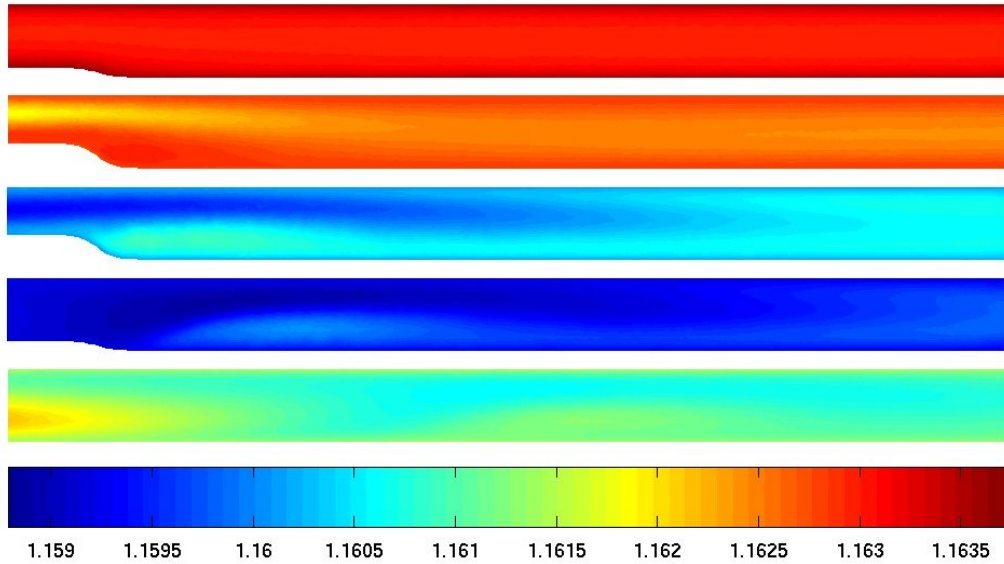


Figure 3.15: Density field ($[\text{kg}/\text{m}^3]$) for the flow in a channel with a moving indentation computed by using the NP strategy. From top to bottom, times $t^* = 0.2, 0.4, 0.6, 0.8$ and 1 .

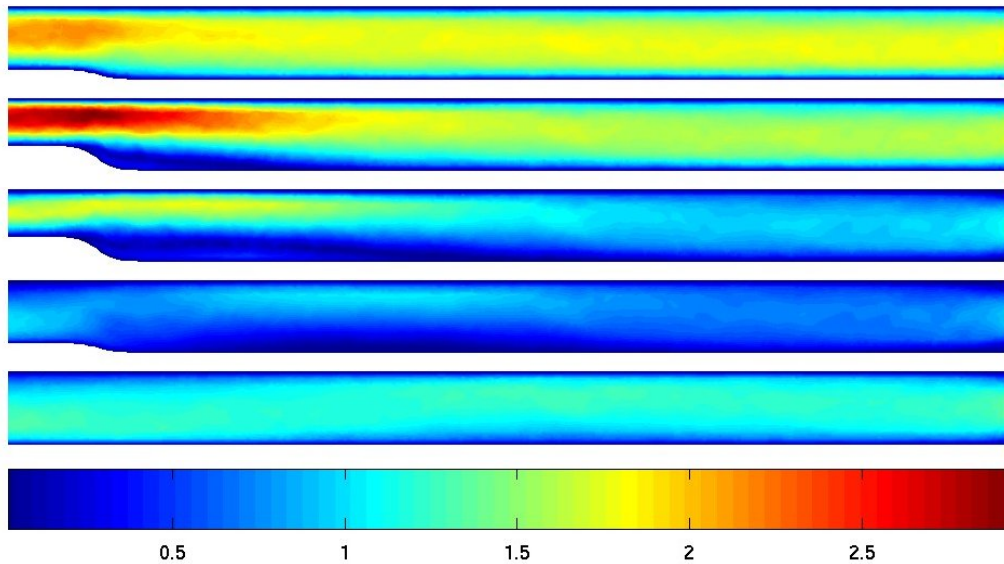


Figure 3.16: Magnitude of the velocity field ($[\text{m}/\text{s}]$) for the flow in a channel with a moving indentation computed by using the NP strategy. From top to bottom, times $t^* = 0.2, 0.4, 0.6, 0.8$ and 1 .

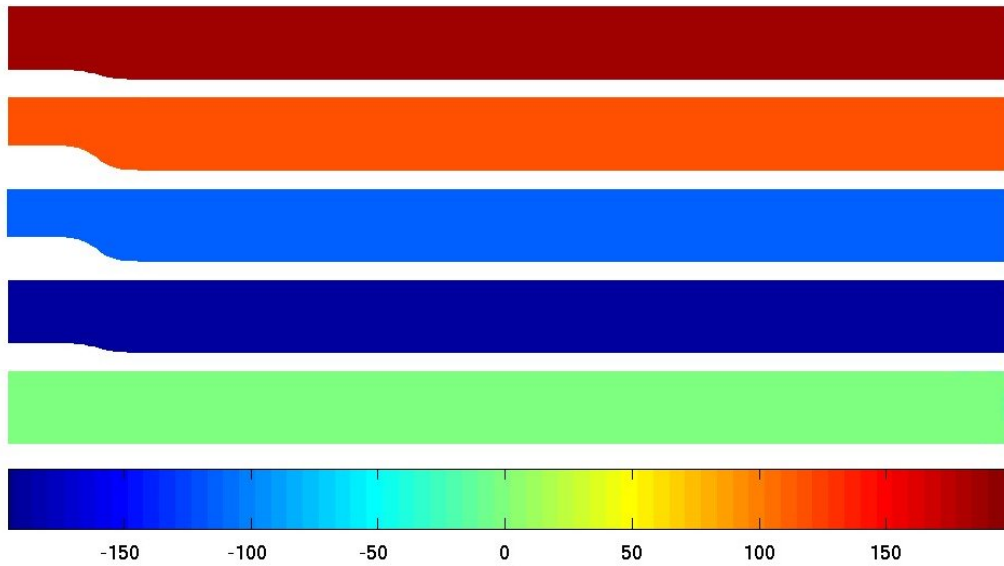


Figure 3.17: Pressure perturbation field ([Pa]) for the flow in a channel with a moving indentation computed by using the NP strategy. From top to bottom, times $t^* = 0.2, 0.4, 0.6, 0.8$ and 1.

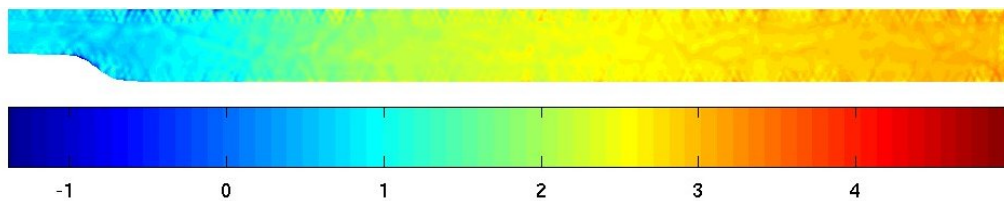


Figure 3.18: Pressure perturbation field ([Pa]) for the flow in a channel with a moving indentation computed by using the NP strategy at $t^* = 0.5$, where the scale was modified.

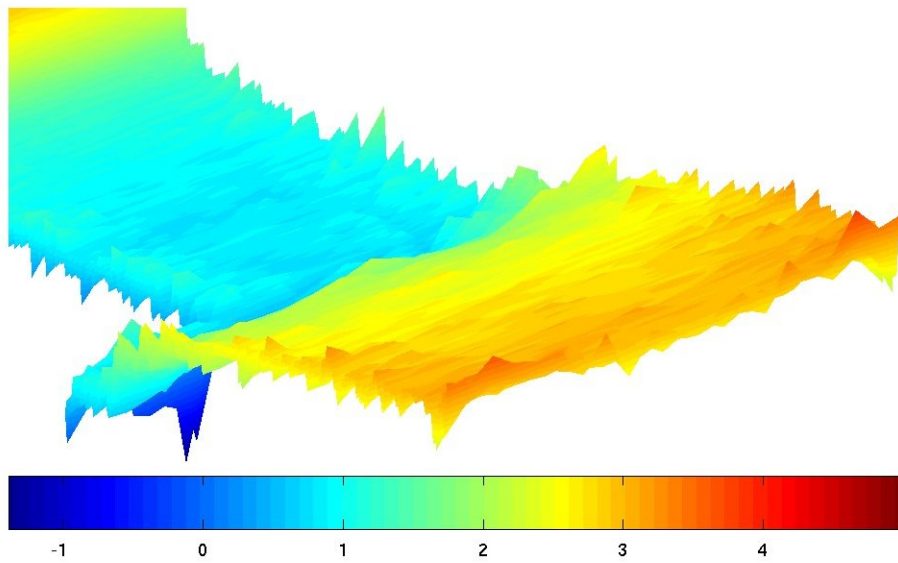


Figure 3.19: Pressure perturbation field ([Pa]) for the flow in a channel with a moving indentation computed by using the NP strategy at $t^* = 0.5$ represented as 3D elevation and where the scale was modified.

3.3.3 In-cylinder flow in an opposed-piston engine

The last case presented in this chapter is the resolution of the fluid flow inside the cylinder of an opposed-piston engine under cold conditions, *i.e.* without combustion. This test was selected in order to apply the preconditioning strategy to an inherently compressible case similar to what it is found in real engine geometries and also due to its simplicity. The engine geometry was taken from the KIVA-3 [4] tutorial. The cylinder bore is 100 mm, the stroke of each piston is 85 mm, and the geometric compression ratio is 9.5:1. The cylinder has 8 exhaust ports equally distributed in the circumferential direction and 12 intake ports uniformly separated also. Assuming the reference angle as the EDC (External Dead Center), the timing of the ports are the following

- Intake Port Opening (IPO) = 295.13°
- Intake Port Closing (IPC) = 64.87°
- Exhaust Port Opening (EPO) = 280.2°
- Exhaust Port Closing (EPC) = 79.8°

In order to simplify the problem, the flow domain is reduced to a two-dimensional one by means of the intersection between the 3D cylinder and a plane containing the axis of the cylinder. This plane must be oriented in such a way to ‘cut’ two intake and exhaust ports simultaneously. The resultant geometry is shown in figure 3.20 for pistons located at EDC.

The mesh was generated with the pistons at EDC (ports totally opened) and has 76K triangular elements (structured) and 38K nodes. The mean element size is $h = 0.5$ mm. Due to the simplicity of the geometry and the boundary movement, the mesh dynamics is

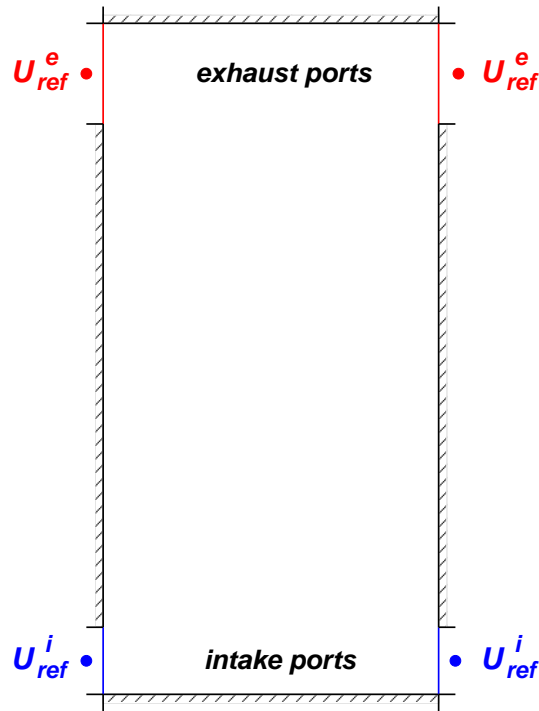


Figure 3.20: Geometry of the simplified 2D model for the opposed-piston engine case (pistons at EDC).

solved by using an algebraic law following a linear distribution with respect to the position of pistons at IDC (Internal Dead Center).

No-slip condition is imposed at solid walls. In addition, solid walls are assumed insulated. Mixed absorbing/wall boundary conditions are used to model the ports, as explained in section §1.2.3. For absorbing boundary conditions, the reference state used for intake ports is $\mathbf{U}_{\text{ref}}^i = [1.2195 \text{ kg/m}^3, 0 \text{ m/s}, 0 \text{ m/s}, 105 \text{ kPa}]^T$, and for the exhaust ports is $\mathbf{U}_{\text{ref}}^e = [0.662 \text{ kg/m}^3, 0 \text{ m/s}, 0 \text{ m/s}, 95 \text{ kPa}]^T$. The engine speed is 3000 rpm. The time step used in the simulation corresponds to 0.5 crank angle degree (CAD) and, for the conditions of the flow in the test, a $CFL_u \mathcal{O}(1)$ is obtained along the whole simulation. The stationary cyclic state is reached with approximately four cycles. The following results correspond to the last cycle simulated.

For some instants in the cycle, the density and pressure fields into the chamber are depicted in the following figures. The purpose is to show that the unsteady preconditioning strategy presented in this chapter produces smooth solutions (without numerical oscillations) when it is applied to computations of in-cylinder flows problems. Figures 3.21 and 3.22 show the distribution of density and pressure into the cylinder at EDC and IDC, respectively. In figure 3.23 these fields are plotted at an intermediate position of the pistons, corresponding to 270 CAD.

In figure 3.24, the mean density and pressure in the cylinder during a whole cycle are presented. This brief overview of the results obtained is completed with some pictures of the magnitude of the flow velocity at several instants in the cycle. Figure 3.25 shows these results.

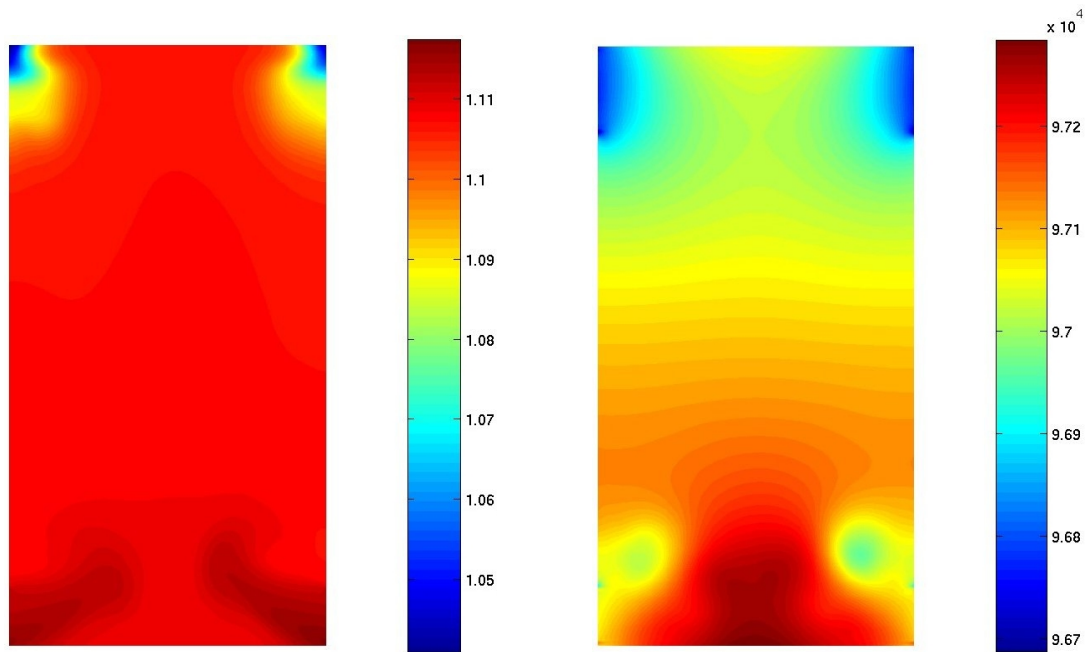


Figure 3.21: Density ($[\text{kg/m}^3]$, left) and pressure ($[\text{Pa}]$, right) fields at EDC (0°).

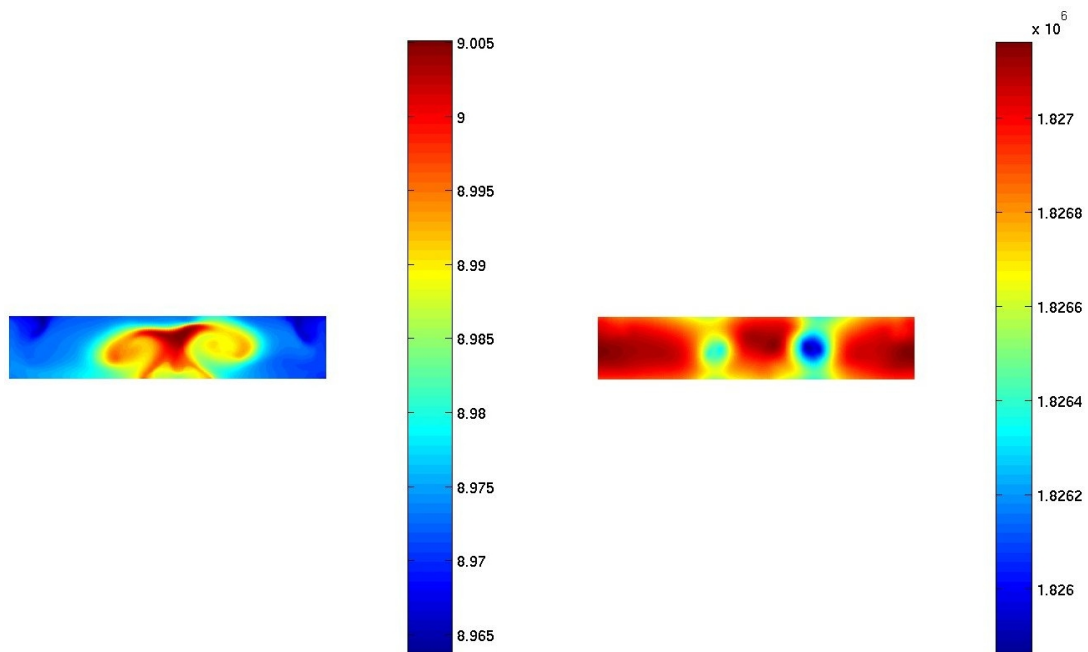


Figure 3.22: Density ($[\text{kg/m}^3]$, left) and pressure ($[\text{Pa}]$, right) fields at IDC (180°).

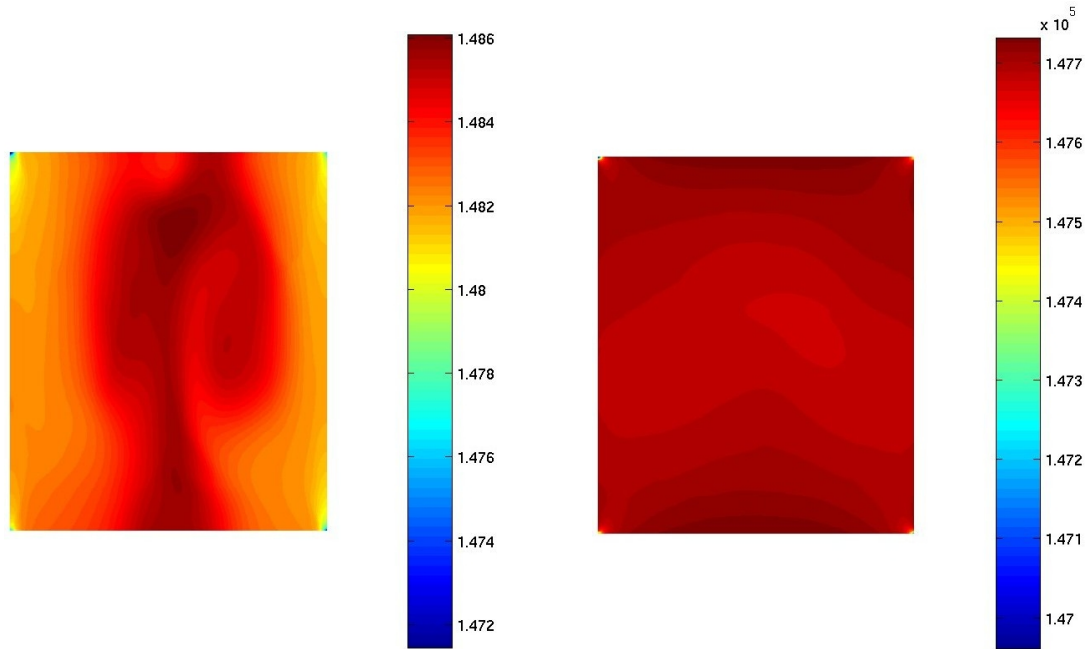


Figure 3.23: Density ($[\text{kg}/\text{m}^3]$, left) and pressure ($[\text{Pa}]$, right) fields at 270° .

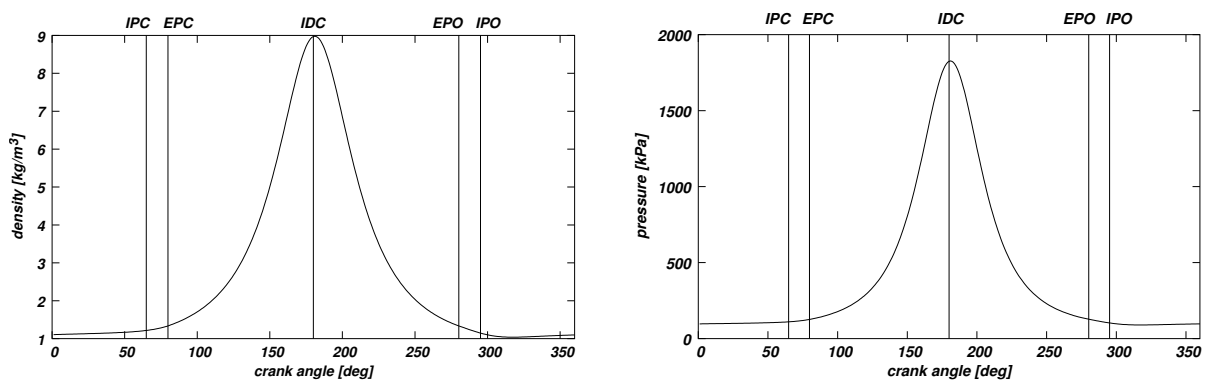


Figure 3.24: Cylinder mean density (left) and pressure (right) through a cycle for the opposed-piston engine.

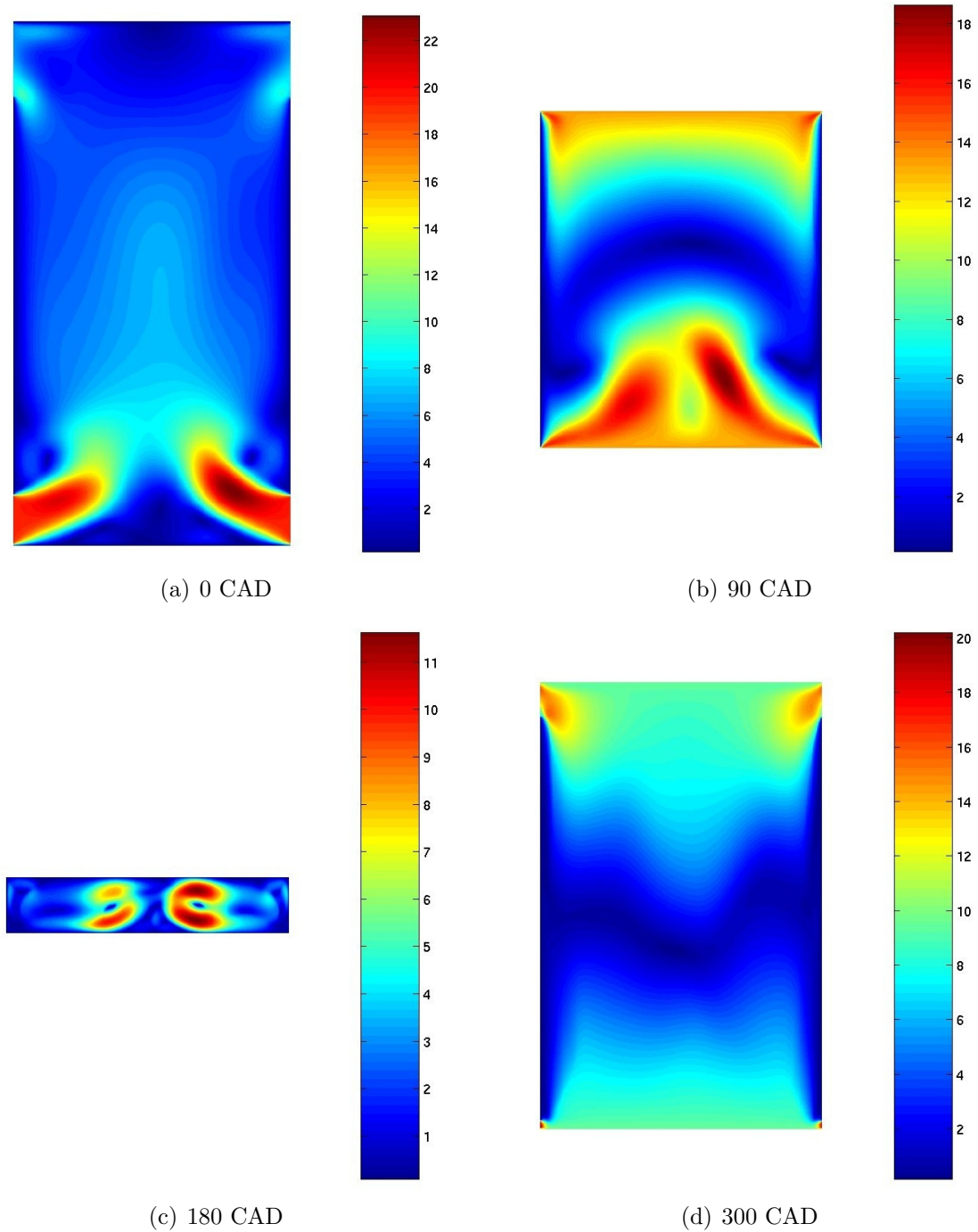


Figure 3.25: Magnitude of flow velocity field ([m/s]) at several instants during a cycle.

Chapter 4

Thermodynamic and gas-dynamic based IC engine models

*Unleash the ultimate all-devastating
wave of deac-dimensional ironflames
Release the over-poxer of de-dreation
Emflame cosmos, space time, from infinite to infinite
Endtime, endspace,
infernial eternal, eternity frozen
Supreme total annihilation divine to dimension Zero
Otargos*

This chapter reviews some models and their primary use to describe IC engine operating characteristics. For the processes that govern engine performance, two basic types of models have been developed. These can be categorized as *thermodynamic* or *fluid dynamic* in nature, depending on whether the equations which give the model its predominant structure are based on energy conservation or on the analysis of the fluid motion. Other labels given to thermodynamic energy-conservation-based models are: zero-dimensional (since in the absence of any flow modeling, geometric features of the fluid motion cannot be predicted), phenomenological (since additional detail beyond the energy conservation equations is added for each phenomenon in turn), and quasi-dimensional (where specific geometric features, *e.g.* the spark-ignition engine flame geometry or the diesel fuel spray shapes, are added to the basic thermodynamic approach). The fluid-dynamic model widely used is the represented by the one-dimensional gas-dynamic equations. This approach is applied to intake and exhaust manifolds, for which the assumption of plane wave motion could be satisfactory. Thermodynamic models are used for some components of the engine such as cylinders, valves, pipe junctions, air-boxes, etc. The 1D approach becomes unsatisfactory for a detailed modeling of unsteady flows in these components since complex wave motion occurs into them. Such devices are considered as boundary regions in the 1D simulation.

The development of a computational code able to predict IC engine performance and emissions began at CIMEC (*Centro Internacional de Métodos Computacionales en In-*

geniería) with the implementation of a single-cylinder four-stroke spark-ignition engine simulator [54]. Then, with the target focused on the prediction of more real situations, models for pipe junctions in multi-cylinder configurations were added. In order to check the suitability and reliability of this computational tool in industrial applications, that code was written in the compiled language Fortran 90/95 and several test cases were solved validating the results with measurements [2]. This engine simulator uses explicit schemes for time integration, which is standard for that kind of computational tool.

Since the goal is to use the 0D/1D code as a generator of dynamic boundary conditions for CFD-3D models, the availability of a version of the 0D/1D engine simulator with implicit time integration gives greater generality to the tool. With this purpose a new code was written using an object-oriented programming language. The numerical code developed is able to simulate spark-ignition and compression-ignition, two-stroke and four-stroke, multi-cylinder and multi-valve engines, naturally aspirated or turbo-charged. Also, different geometries of the combustion chamber are available.

4.1 Mathematical models

4.1.1 Pipe model

One-dimensional unsteady flow equations are used for modeling pipes and manifolds. In order to include effects like variable cross-section, viscous friction, and wall heat transfer, some source terms are added to the inviscid gas dynamic model represented by the system of Euler equations. The resultant system of equations can be written as [12]

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} &= -\frac{1}{F} \frac{dF}{dx} \rho u \\ \frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p)}{\partial x} &= -\frac{1}{F} \frac{dF}{dx} \rho u^2 - \rho G \\ \frac{\partial(\rho E)}{\partial t} + \frac{\partial[(\rho E + p)u]}{\partial x} &= -\frac{1}{F} \frac{dF}{dx} u(\rho E + p) + \dot{q} \pi \frac{D}{F} \end{aligned} \quad (4.1)$$

where F is the pipe cross-section area;

$$G = f \frac{u|u|}{2} \pi \frac{D}{F}$$

is the specific friction force, with the friction coefficient given by $f = 8\tau_w/\rho u^2$, τ_w being the viscous shear stress at the pipe wall and D the equivalent diameter of the pipe; and \dot{q} is the heat transfer per unit mass of fluid per unit time.

The friction coefficient is frequently assumed to be that of a fully developed, steady-state, turbulent flow [34]

$$f = 0.316 Re^{-1/4} \quad (4.2)$$

where the Reynolds number is defined as

$$Re = \frac{\rho |u| D}{\mu} \quad (4.3)$$

The film heat transfer coefficient can be calculated by using the Reynolds-Colburn analogy [34] as

$$StPr^{2/3} = \frac{f}{8} \quad (4.4)$$

where St is the Stanton number ($St = Nu/RePr$) and Nu is the Nusselt number. Substitution of equation (4.2) into equation (4.4) yields

$$Nu = \frac{h_f D}{\kappa} = 0.0395 Re^{3/4} Pr^{1/3} \quad (4.5)$$

h_f being the film heat transfer coefficient. For constant-temperature walls (T_{wall}), the heat transfer rate per unit area can be expressed as

$$\dot{q} = h_f(T_{\text{wall}} - T) = 0.0395 \frac{\kappa}{D} Re^{3/4} Pr^{1/3} (T_{\text{wall}} - T) \quad (4.6)$$

4.1.2 Cylinder model

The cylinder model used is a single-zone model in which the charge is assumed to be a homogeneous mixture of ideal gases at all times. The equations of the model are the conservation of mass and the first law of thermodynamics

$$\begin{aligned} \frac{dm}{dt} &= \sum_j \dot{m}_j \\ \frac{d}{dt}(me) &= -p\dot{V} + \dot{Q}_{\text{ch}} - \dot{Q}_{\text{ht}} + \sum_j h_j \dot{m}_j \end{aligned} \quad (4.7)$$

where m is the mass contained in the cylinder; \dot{m}_j is the instantaneous mass flow rate through the j -th inlet/outlet (intake and exhaust valves, fuel addition, leakages, etc.); e is the specific internal energy of the mixture; V is the cylinder volume; \dot{Q}_{ch} represents the heat release due to combustion; \dot{Q}_{ht} is the heat transfer rate; and $h_j \dot{m}_j$ represents the enthalpy fluxes through the j -th inlet/outlet.

The model is closed specifying the geometry of the combustion chamber, the heat release rate, the heat transfer rate through the cylinder walls, and the mass flow rate of air and fuel. The sub-models used here will be presented in the following sections.

Geometry of the combustion chamber

The geometric data necessary for the cylinder model are the total surface area of the cylinder walls (A), the volume of the cylinder and its time derivative. In the code developed, it is possible to simulate conventional reciprocating engines, opposed-piston engines, and the MRCVC (*Motor Rotativo de Combustión a Volumen Constante* [80]). For all these cases, analytical formulae were implemented to compute the geometric variables of the combustion chamber.

Heat transfer model

The instantaneous heat transfer rate that appears in equation (4.7) is calculated applying Nusselt-Reynolds-Prandtl number correlations as, for example, the developed by Woschni [90] or Annand [6]. All of them allows to compute a film transfer coefficient h_f with expressions like the following

$$Nu = \frac{h_f L}{\kappa} = C Re^\alpha Pr^\beta \quad (4.8)$$

where L is a characteristic dimension; and C , α and β are constants.

Then, the heat transfer rate to the walls is

$$\dot{Q}_{ht} = Ah_f(T - T_{wall}) \quad (4.9)$$

where T is the temperature of the gas into the cylinder, and T_{wall} is the cylinder wall temperature.

Heat release model

In order to model combustion, several approaches and mathematical models are used. These approaches have the goal to describe the actual heat release via combustion as exactly as possible by means of the so-called *substitute heat release rates* [48].

For spark-ignition engines, the mass fraction of burnt gases (x_b) is computed by using a Wiebe function [32]

$$x_b = \frac{m_b}{m} = 1 - \exp \left[-c \left(\frac{\theta - \theta_{ig}}{\Delta\theta} \right)^{s+1} \right] \quad (4.10)$$

In equation (4.10) m_b represents the mass of burnt gases; θ is the crank shaft angle; $\Delta\theta$ is the duration of combustion; θ_{ig} denotes the angle at which burning starts; c and s are parameters, where s is designated as the shape parameter and c accounts for combustion efficiency. The heat release rate can be computed as

$$\dot{Q}_{ch} = H_c \dot{x}_b m_f \quad (4.11)$$

H_c being the calorific heat content of the fuel and m_f the total mass of fuel trapped into the cylinder.

For compression-ignition engines, two substitute heat release rates were implemented: the model proposed by Watson *et al.* [86] and the double Wiebe function [51, 53]. In the model by Watson *et al.* the mass burning rate of fuel is expressed as

$$\frac{dx_b}{dt} = \frac{\omega}{\Delta\theta} [\beta c_1 c_2 \tau^{c_1-1} (1 - \tau^{c_1})^{c_2-1} + (1 - \beta) c'_1 c'_2 \tau^{c'_1-1} \exp(-c'_1 \tau^{c'_2})] \quad (4.12)$$

where $\tau = (\theta - \theta_{ig})/\Delta\theta$, and ω is the angular speed of the shaft. The coefficients proposed in the original model are

- $c_1 = 2 + 1.25 \times 10^{-8} (N t_d)^{2.4}$, where N is the engine speed, in rpm; and t_d is the ignition delay, in ms.

- $c_2 = 5000$.
- $c'_1 = 14.2\phi_{\text{ig}}^{-0.644}$, ϕ_{ig} being the equivalence ratio at the time of ignition.
- $c'_2 = 0.79(c'_1)^{0.25}$.
- $\beta = 1 - 0.926\phi_{\text{ig}}^{0.37}t_d^{-0.26}$.

The double Wiebe function is an extension of the model used for spark-ignition engines in order to describe the premixed and diffusive combustion periods observed in diesel engines [66]. The mass fraction of burnt gases can be written as

$$x_b = 1 - x_p \exp \left[-c \left(\frac{\theta - \theta_{\text{ig}}}{\Delta\theta_p} \right)^{s_p+1} \right] - x_{\text{di}} \exp \left[-c \left(\frac{\theta - \theta_{\text{ig}}}{\Delta\theta_{\text{di}}} \right)^{s_{\text{di}}+1} \right] \quad (4.13)$$

where x_p is the mass fraction of fuel burnt in the premixed combustion period, x_{di} is the mass fraction of fuel burnt in the diffusive combustion period, $\Delta\theta_p$ and $\Delta\theta_{\text{di}}$ are, respectively, the duration of premixed and diffusive combustion.

In diesel engines, the ignition delay time can be calculated as the difference between the time at which combustion starts (t_{ig}) and the time at which injection starts (t_{inj}). The time t_{ig} can be obtained from the following expression [8]:

$$\int_{t_{\text{inj}}}^{t_{\text{ig}}} \frac{dt}{t_d} = 1 \quad (4.14)$$

which accounts for the pressure and temperature variations resulting from compression. The ignition delay time as a function of T and p was correlated for a variety of fuels with the expression

$$t_d = Cp^{-n} \exp(T_a/T),$$

being C , n and T_a constants [8]. Also, the empirical formula developed by Hardenberg and Hase for predicting the ignition delay time in direct-injection diesel engines was implemented in the code. This formula is given by the following expression [31]

$$t_d[CA] = (0.36 + 0.22\bar{S}_p) \exp \left[E_A \left(\frac{1}{\tilde{R}T} - \frac{1}{17190} \right) \left(\frac{21.2}{p - 12.4} \right)^{0.63} \right] \quad (4.15)$$

where the temperature of the charge is given in kelvins and the pressure in bars (taken as TDC conditions), \bar{S}_p is the mean piston speed (meters per second), \tilde{R} is the universal gas constant, and E_A (joules per mole) is the apparent activation energy given by

$$E_A = \frac{618840}{CN + 25} \quad (4.16)$$

CN being the fuel cetane number.

Scavenge model

The scavenge process in two-stroke engines is modeled via a semi-empirical model proposed by Blair [14]. The mass of delivered air trapped into the cylinder at exhaust closure (m_{at}) is computed as $m_{at} = \eta_S m_t$, where m_t is the total mass retained after the exhaust port closure, and the scavenging efficiency (η_S) can be computed as

$$\eta_S = 1 - \exp(b\sigma_v + d) \quad (4.17)$$

b and d being constants experimentally determined, which depend on the type of scavenge. σ_v is the scavenge ratio by volume, defined as the ratio between the volume of air supplied during the scavenge period and the cylinder volume.

4.1.3 Valve model

To calculate the flow rates through the intake and exhaust valves, an analogy with flow through convergent nozzles proposed by Benson [12] is used. The model assumes the passage area through the valve as the nozzle throat (whose state is represented by the subscript T in the equations), and the nozzle connecting the cylinder (subscript C in the equations) and the end of the pipe (subscript P in the equations). Depending on the direction of the flow velocity with respect to the pipe end, the problem could be an *inlet* (from cylinder to pipe) or an *outlet* (from pipe to cylinder). In addition, the flow at the throat could be sonic or subsonic. The equations of the model are presented below.

- Subsonic inlet:

$$\begin{aligned} \left(\frac{dp}{dt}\right)_P \pm \rho_{PCP} \left(\frac{du}{dt}\right)_P &= (\text{RHS}_2^\pm)_P \quad \text{along } \lambda^\pm \\ \rho_T u_T \psi &= \rho_P u_P \\ c_C^2 &= c_P^2 + \delta u_P^2 \\ \frac{p_C}{p_T} &= \left(\frac{\rho_C}{\rho_T}\right)^\gamma \\ c_C^2 &= c_T^2 + \delta u_T^2 \\ p_T &= p_P \end{aligned} \quad (4.18)$$

where $\psi = F_T/F_P$, $\delta = (\gamma - 1)/2$, and

$$\text{RHS}_2^\pm = \rho(\gamma - 1) \left(\frac{\dot{q}}{\rho} \pi \frac{D}{F} + uG \right) \mp \rho cG - \frac{\rho u c^2}{F} \frac{dF}{dx} \quad (4.19)$$

In system (4.18), the first equation accounts for the compatibility along the incoming Mach line λ^\pm ; the second equation is the mass conservation between T and P ; the third and fifth equations represent the energy conservation between C and P , and C and T , respectively; the fourth equation represents an isentropic evolution between the cylinder and the nozzle throat; and the last equation is the condition on the pressure at the nozzle exit.

- Subsonic outlet:

$$\begin{aligned}
\left(\frac{dp}{dt}\right)_P \pm \rho_P c_P \left(\frac{du}{dt}\right)_P &= (\text{RHS}_2^\pm)_P \quad \text{along } \lambda^\pm \\
\rho_T u_T \psi &= \rho_P u_P \\
\left(\frac{Dp}{Dt}\right)_P - c_P^2 \left(\frac{D\rho}{Dt}\right)_P &= (\text{RHS}_1)_P \quad \text{along } \lambda^0 \\
\frac{p_P}{p_T} &= \left(\frac{\rho_P}{\rho_T}\right)^\gamma \\
c_P^2 + \delta u_P^2 &= c_T^2 + \delta u_T^2 \\
p_T &= p_C
\end{aligned} \tag{4.20}$$

with

$$\text{RHS}_1 = (\gamma - 1) \left(\dot{q} \pi \frac{D}{F} + \rho u G \right) \tag{4.21}$$

and D/Dt denoting the material derivative. From first equation to the last one in the system (4.20), they represent, respectively, the compatibility along the incoming Mach line, the mass conservation between T and P , the compatibility along the incoming path line λ^0 , the isentropic evolution between P and T , the energy conservation between P and T , and the condition on the pressure at the nozzle exit.

- Sonic inlet: for this case, the system of equations is the same as equation (4.18) with the last equation replaced by the condition $c_T = u_T$.
- Sonic outlet: for this case, the system of equations is the same as equation (4.20) with the last equation replaced by $c_T = u_T$.

4.1.4 Pipe junctions model

The pipe junction model used was proposed by Corberan [18]. If the junction is composed by r incoming pipes and s outgoing pipes, the model is expressed as

- Mass conservation

$$\sum_{j=1}^N \dot{m}_j = 0, \quad \text{with} \quad \dot{m}_j = \rho_j F_j u_j n_j$$

where $N = r + s$ is the total number of pipes at the junction, F_j is the cross-section area of the j -th pipe and n_j its exterior normal.

- Energy conservation

$$\sum_{j=1}^N \dot{h}_j = 0, \quad \text{with} \quad \dot{h}_j = \frac{\dot{m}_j}{\gamma - 1} (c_j^2 + \delta u_j^2)$$

- Compatibility equation along incoming Mach lines λ_j^\pm

$$\left(\frac{dp}{dt}\right)_j \pm \rho_j c_j \left(\frac{du}{dt}\right)_j = (\text{RHS}_2^\pm)_j, \quad j = 1, \dots, N$$

- Compatibility equation along incoming path lines λ_j^0

$$\left(\frac{Dp}{Dt}\right)_j - c_j^2 \left(\frac{D\rho}{Dt}\right)_j = (\text{RHS}_1)_j, \quad j \in r$$

- Equality of pressure at all branches in the junction

$$p_i = p_j, \quad \forall i \neq j$$

- Equality of enthalpy at all outgoing branches in the junction

$$c_i^2 + \delta u_i^2 = c_j^2 + \delta u_j^2, \quad \forall i, j \in s, \quad i \neq j$$

4.2 Numerical implementation

The use of an implicit scheme for the discretization of equations that model the several devices do not represent a problem for its practical implementation, with the exception of the models of valve and pipe junction. As presented above, the systems of equations that model valves and pipe junctions change with the flow regime and the direction of its velocity.

In the valve model, to determine whether the problem is an inlet or an outlet, the cylinder pressure and the stagnation pressure at the end of the pipe (p_{0P}) are compared. This stagnation pressure is given by [88]

$$p_{0P} = p_P \left[1 + \delta \left(\frac{u_P}{c_P} \right)^2 \right]^{\gamma/(\gamma-1)} \quad (4.22)$$

The valve is considered open when the passage area is larger than a prefixed tolerance $\epsilon_A > 0$. Then, if $p_C > p_{0P}$ the inlet flow equations are solved, otherwise it is considered an outgoing flow. Besides, to take into account the transition between subsonic and sonic regime flow the following convex combination is used

$$\mathbf{E} = (1 - \chi)\mathbf{E}_{\text{sonic}} + \chi\mathbf{E}_{\text{subsonic}} \quad (4.23)$$

where $\mathbf{E}_{\text{subsonic}}$ and $\mathbf{E}_{\text{sonic}}$ represent the systems of equations that model the subsonic and sonic cases, respectively; and

$$\chi = \frac{1}{2} \left[1 + \tanh \left(\frac{c_T - u_T}{\alpha c_T} \right) \right] \quad (4.24)$$

α being a coefficient that adjusts the transition of χ between 0 and 1. In the valve model, the compatibility equations along the characteristic curves are used to complete the system of equations. These must be solved according to an explicit scheme, implying that the states at the pipe end, the valves, and the cylinder, at the time t do not depend on the state at interior points in the pipe at time $t + \Delta t$. Thus, the system of equations for the valve is decoupled from the remaining equations. However, the resolution is done in a coupled way with the goal of being able to implement other valve models of (full) implicit type in the code. When the valve is either opening or closing, it is important

to determine the direction of the flow due to the change in the system of equations that model each situation, which directly influences the global convergence. To predict the flow direction it is assumed an isentropic flow through a convergent nozzle between the cylinder and the corresponding end of the pipe. The flow is established between the states \mathbf{U}_0 and \mathbf{U}_e , which are assumed constant and are identified with either the state in the cylinder (\mathbf{U}_C) or the state at the end of the pipe (\mathbf{U}_P), depending on the relationship between the pressure in the cylinder and the stagnation pressure p_{0P} . If $p_C \geq p_{0P}$, then it is adopted $\mathbf{U}_0 = \mathbf{U}_C$ and $\mathbf{U}_e = \mathbf{U}_P$, being the direction of flow from the cylinder to the pipe. When $p_C < p_{0P}$, then $\mathbf{U}_0 = \mathbf{U}_P$ and $\mathbf{U}_e = \mathbf{U}_C$, yielding to a flow from the pipe to the cylinder. The critical pressure

$$p_{\text{crit}} = p_0 \left(\frac{2}{\gamma + 1} \right)^{\gamma/(\gamma-1)} \quad (4.25)$$

determines if the nozzle throat is choked or not. If $p_e > p_{\text{crit}}$, the predicted state at valve (\mathbf{U}_T), *i.e.* the nozzle throat, is taken as

$$\begin{aligned} p_T &= p_e \\ \rho_T &= \rho_0 \left(\frac{p_T}{p_0} \right)^{1/\gamma} \\ u_T &= \left\{ \frac{\gamma p_T}{\delta \rho_T} \left[\left(\frac{p_0}{p_T} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right] \right\}^{1/2} \end{aligned} \quad (4.26)$$

If $p_e \leq p_{\text{crit}}$, then it is assumed

$$\begin{aligned} p_T &= p_{\text{crit}} \\ \rho_T &= \rho_0 \left(\frac{2}{\gamma + 1} \right)^{1/(\gamma-1)} \\ u_T &= \left(\frac{2\gamma p_0}{\gamma + 1 \rho_0} \right)^{1/2} \end{aligned} \quad (4.27)$$

To solve the system of equations (4.1), the stabilized Finite Element Method with the SUPG technique is used. Time derivatives were discretized applying a trapezoidal finite difference scheme.

The code was implemented in the scripting language `Python` [84]. `Python` is a dynamic object-oriented programming language that can be used for many kinds of software development. It offers strong support for integration with other languages and tools, and comes with extensive standard libraries. Object-oriented programming allows to develop the code in an organized manner, and the possibility of integration with other languages make it suitable for solving the coupling between the engine simulator and CFD codes. Another feature that `Python` offers is the possibility of writing higher-level parts of large-scale scientific applications and driving simulations in parallel architectures like clusters of PCs [19].

Regarding the implementation, a class was defined for each device, with methods that perform the computation of sub-models and the contributions to both global residue and global jacobian matrix. For instance, the class `Cylinder` contains the methods

combustion, geometry, heat_transfer, etc. All of these classes derive from the classes that define an specific set of equations for each device. For example, the class `Flow` involve the equations of mass, energy and linear momentum conservation. This allows to add new features to the code as, for example, transport of chemical species with very few changes. Currently, only the class `Flow` is implemented in the code. The constructor of each class defines attributes for the object, some of them are required data and the remaining ones are optional (defined through default values).

The global nonlinear system of equations is solved via the Newton-Raphson method. The linear system arising at each nonlinear iteration is solved with functions contained into the package NumPy [60].

The code also allows to define the parameters of calibration and the operational variables as functions of the engine speed, the cycles, and the time.

4.3 Results

In the next sections, the results obtained from the application of the code to the simulation of some internal combustion engines are presented. The cases were selected to show the spectrum of IC engines that can be simulated.

4.3.1 Four-stroke spark-ignition engine test

The first example of application is a 8.4 liters V10 four-stroke spark-ignition engine. Tables 4.1 and 4.2 contain the main data of the engine.

Cylinder	
Bore	103.0 mm
Stroke	100.6 mm
Connecting rod length	158.55 mm
Compression ratio	9.6:1

Table 4.1: Main cylinder data of V10 engine.

	Intake valve	Exhaust valve
Diameter	52.8 mm	40.89 mm
Max. lift	14.4 mm	13.59 mm
VO	41° BTDC	129° BBDC
VC	123° ABDC	41° ATDC

Table 4.2: Valve data of V10 engine.

The intake manifold pipes have a diameter of 60 mm and a length of 525 mm. The diameter of the exhaust pipes is 50 mm and their length is 500 mm. The wall temperature of these pipes is 298 K and 373 K for the intake and exhaust manifolds, respectively.

The equivalence ratio, the coefficient of heat transfer through the cylinder walls, and the crank angle where the combustion starts, are functions of the engine speed. These functions were obtained from the experimental data available. The combustion is modeled

by using a Wiebe function, and the wall temperature of the cylinder is 573 K. The engine was tested at speeds ranging from 1600 rpm to 6000 rpm.

Figures 4.1 to 4.3 show the computed indicated power, the torque and the average mass flow rate of air versus the real curves obtained experimentally.

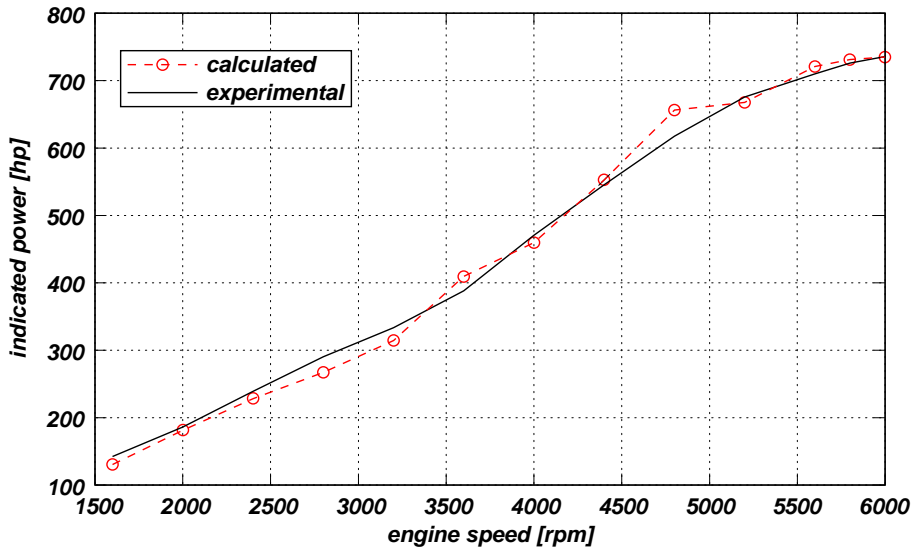


Figure 4.1: Indicated power as a function of rpm.

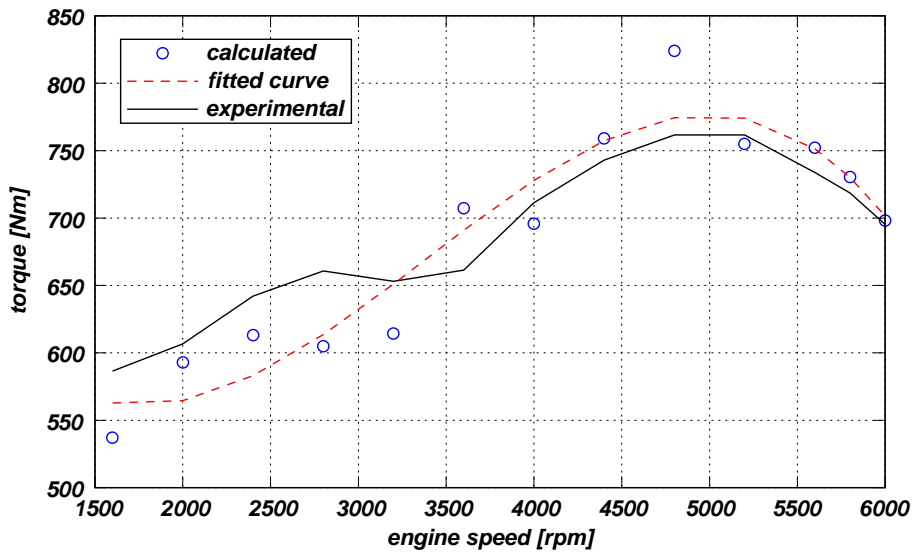


Figure 4.2: Torque as a function of rpm.

4.3.2 Two-stroke spark-ignition engine test

This test case was taken from the literature [14] and consists in a two-stroke spark-ignition single-cylinder research engine denominated QUB 400. The engine speed is 3000 rpm at full throttle. To model the crankcase compression it is used a cylinder without combustion, then the computational model is composed by two cylinders as could be observed in

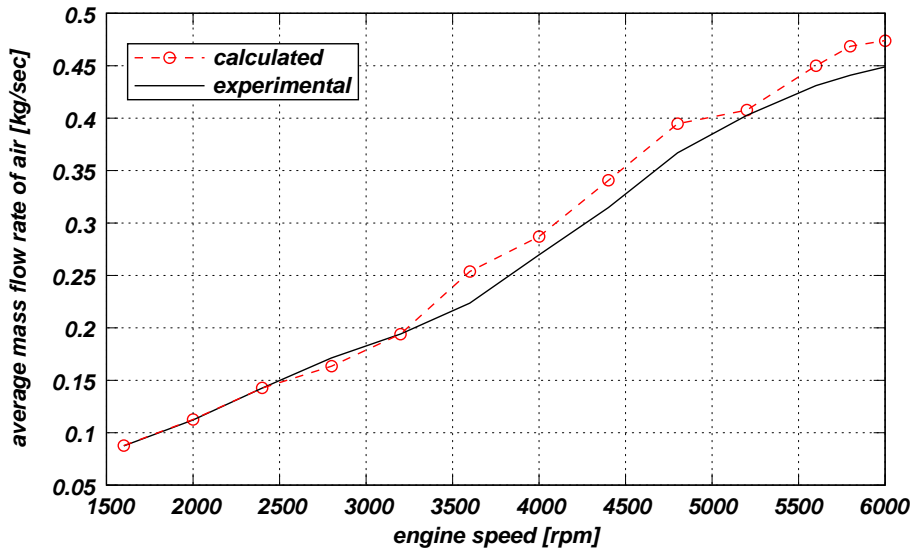


Figure 4.3: Average mass flow rate of air at intake port as a function of rpm.

figure 4.4. The engine has six transfer ducts between the cylinder and the crankcase, which were modeled as a unique pipe with the same total cross-sectional area (see figure 4.4). Ports were assumed as poppet valves with appropriate discharge coefficients, and where the valve lift was defined in such a way to make the passage area computed by the code the same as the instantaneous passage area of the real port. The complete set of data can be found in the literature by Blair [14].

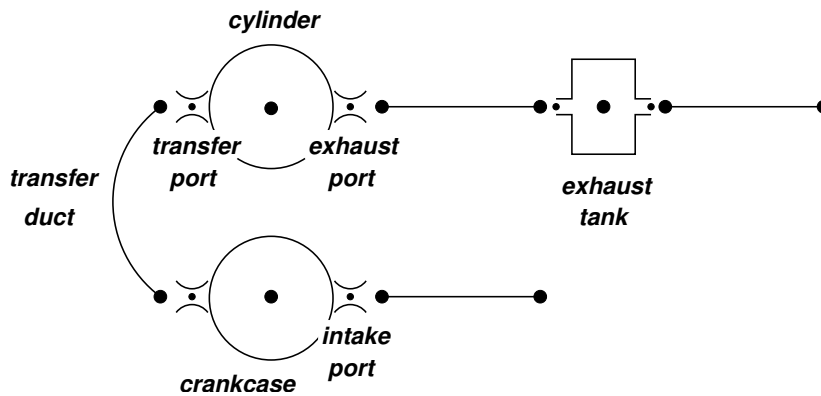


Figure 4.4: Computational model of QUB 400 engine.

Figure 4.5 shows the pressure in the cylinder, in the crankcase, in the transfer duct, and in the intake and exhaust ports as a function of the crank angle during a cycle. Mass flow rate through transfer and exhaust ports are shown in figure 4.6, where positives values represent incoming flow to cylinder.

Table 4.3 shows some performance characteristics: power, indicated mean pressure (IMEP), scavenge efficiency (η_S), peak cylinder pressure (p_{\max}) and the crank angle at which it occurs. Results reported by Blair [14] and experimental data are included for comparison.

These results were in a very good agreement with those presented by Blair [14] coming from his numerical simulations and also with his experiments.

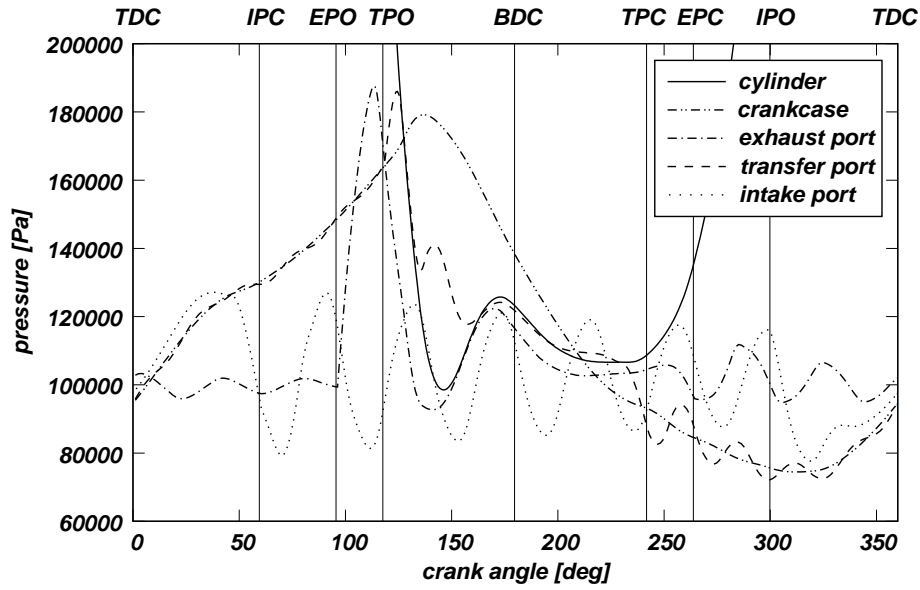


Figure 4.5: Variation of pressure during a cycle.

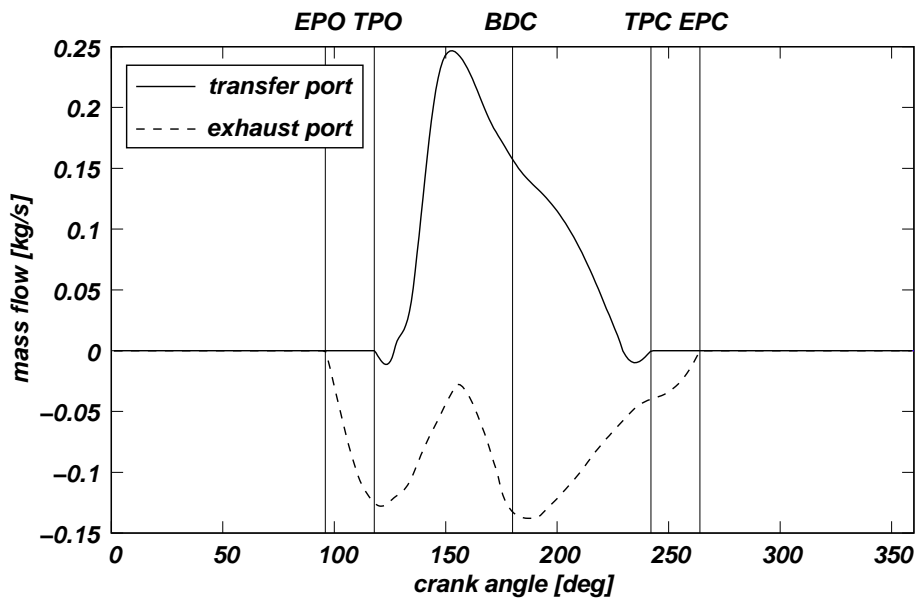


Figure 4.6: Mass flow rates through transfer and exhaust ports.

	Calculated	Blair [14]	Exp.
Power [kW]	12.37	12.37	12.40
IMEP [bar]	6.83	6.81	6.80
η_s	0.77	0.84	-
p_{\max} [bar]	37.0	36.2	36.9
Angle [deg]	15.9	16.2	-

Table 4.3: Performance characteristics of QUB 400 engine.

4.3.3 Four-stroke diesel engine test

The next test case was selected to show the application of the code to a real diesel engine. It was the KamAZ-7405 diesel engine applied to heavy duty vehicles and taken from the web-page <http://www.diesel-rk.bmstu.ru/>. The main engine data are the following:

- Number of cylinders: 8.
- Cylinder data:
 - Bore: 120 mm.
 - Stroke: 120 mm.
 - Connecting rod length: 225 mm.
 - Compression ratio: 16:1.
 - Injection starting angle: 14° BTDC.
 - Combustion starting angle: 9° BTDC.
 - Duration of combustion: 69°.
 - Wall temperature: 459 K.
- Intake pipe data:
 - Diameter: 44 mm - 52 mm.
 - Length: 150 mm.
 - Wall temperature: 292 K.
- Exhaust pipe data:
 - Diameter: 38 mm.
 - Length: 400 mm.
 - Wall temperature: 743 K.
- Intake valve data:
 - Diameter: 40 mm.
 - Maximum lift: 8.845 mm.
 - IVO: 15° BTDC.
 - IVC: 50° ABDC.
- Exhaust valve data:
 - Diameter: 40 mm.
 - Maximum lift: 8.803 mm.
 - EVO: 65° BBDC.
 - EVC: 15° ATDC.

Case	N [rpm]	p_i [bar]	T_i [K]	p_e [bar]	m_{inj} g/cycle
1	2200	1.97	390	1.51	0.0778
2	1400	1.52	354	1.36	0.0838
3	1000	1.28	338	1.20	0.0818

Table 4.4: Operational variables of KamAZ-7405 diesel engine.

Table 4.4 contains the operational data of the three cases solved. In that table, p_i and T_i are the intake manifold mean pressure and temperature, respectively; p_e is the exhaust manifold mean pressure; and m_{inj} is the mass of fuel injected per cycle.

The combustion was modeled using two Wiebe functions, and the ignition delay time was calculated with the correlation proposed by Hardenberg and Hase (equation (4.15)).

The calibration of the code was done at 2200 rpm, defining the parameters for the two Wiebe functions, the coefficients of the heat transfer model, and the duration of the combustion. The results are presented in table 4.5 together with the relevant experimental results, where SFC is the specific fuel consumption and G_a is the mean mass flow rate of air through the intake system. In general there is a good agreement with experimental data, the differences being within the typical error margins for the type of code employed.

		Power [kW]	IMEP [bar]	SFC [g/kW h]	G_a [kg/s]
Case 1	Calc.	193.4	11.69	212.4	0.326
	Exp.	193.6	11.70	212.4	0.346
Case 2	Calc.	139.5	12.08	201.8	0.174
	Exp.	138.8	12.03	202.8	0.182
Case 3	Calc.	91.6	10.95	214.4	0.107
	Exp.	92.5	11.05	212.2	0.112

Table 4.5: Performance characteristics of KamAZ-7405 diesel engine.

Chapter 5

Coupling of 1D/multi-D domains for compressible flows

Mother north - united we stand (together we walk)
Phantom north - I'll be there when you hunt them down
Satyricon

Generally, when CFD-3D models are used to simulate the fluid flow in IC engines, due to computational resources availability reasons only a few components of the engine are studied at each time step. Usually, the boundary conditions for these 3D models are dynamic and, hence, are not easy to impose. A typical approach is to use 0D/1D codes as boundary condition generators for the 3D problem.

When dimensionally heterogeneous models are applied to solve a given problem, the need to perform the coupling between sub-domain arises. For IC engine simulation, the coupling between multi-D and 1D domains is the most useful. Thus, the methods presented in this chapter are focused on such coupling type.

The Domain Decomposition theory provides the framework to develop a coupling domain algorithm. Several research work was done in this sense, specially considering elliptic operators and the advection-diffusion-reaction equation [13, 63, 26, 17, 81, 27, 3]. Given a boundary value problem defined on a domain Ω , a partition of that domain in n_s sub-domains ($\Omega_i, i = 1, \dots, n_s$) is built. These sub-domains could be disjoint or overlapping. The original boundary value problem is reformulated in a split form on the sub-domains, and the sub-domain solutions satisfy suitable matching conditions at sub-domain interfaces. These boundary conditions are named transmission conditions [63], and could involve appropriate combinations of the following types

- Dirichlet type: condition on the unknown of the problem.
- Neumann type: condition on the first derivative of the unknown.
- Robin type: linear combination of Dirichlet and Neumann conditions.

For second order differential equations problems, some combination of the above cited boundary conditions could fail to give the original solution on the corresponding sub-domain [35]. In particular, Dirichlet/Dirichlet (D/D) coupling as well as the Neumann/Neumann (N/N) coupling are not possible. For advection-dominated advection-diffusion equations it is important that the boundary conditions accounts for the flow direction. Some techniques, as the adaptive strategies, introduce iterative methods splitting the above interface conditions in a way which is adapted to the local flow direction [17, 81]. Other methods, such as the proposed by Alonso *et al.* [3], do not care about the local direction of the advective field, but only need that the boundary value problems defined on the sub-domains are associated with a suitable coercive bilinear form.

As mentioned above, the purpose is to link dimensionally heterogeneous models for numerical simulation of IC engines. Thus, the focus is placed on the governing equations of compressible fluid flows, namely, the Navier-Stokes and Euler equations. For these equations, the different boundary condition types are interpreted here as follows

- Dirichlet: condition on the vector state (\mathbf{U}) or the advective flux (\mathbf{F}^a). This is true due to \mathbf{F}^a is a function of the vector state only, and not of their derivatives. Moreover, in the case of a perfect gas the advective flux is a homogeneous function of degree 1 of the conservative variable vector [10, 70], this is

$$\mathbf{F}^a(\lambda\mathbf{U}) = \lambda\mathbf{F}^a(\mathbf{U}) \quad \text{for any } \lambda \in \mathbb{R}$$

and taking derivatives with respect to λ and setting $\lambda = 1$, it is obtained the relation [33]

$$\mathbf{F}^a(\mathbf{U}) = \frac{\partial \mathbf{F}^a}{\partial \mathbf{U}} \mathbf{U} = \mathbf{A}\mathbf{U}$$

- Neumann: condition on the diffusive flux (\mathbf{F}^d), since it contains first order derivatives (see equation (1.10)).
- Robin: some linear combination of the other two types, for instance, a condition on the total flux ($\mathbf{F} = \mathbf{F}^a + \mathbf{F}^d$).

For viscous flows, the system of equations contains second order derivatives and, hence, conditions on the state and its first derivatives must to be imposed on the coupling boundary [33].

The discussion is started with the analysis of a 1D domain case divided into two intervals. Let Ω a bounded, open interval of \mathbb{R} discretized by a grid with N elements and $N + 1$ nodes numbered from 1 to $N + 1$. After standard discretization, for instance with the SUPG stabilized Finite Element Method, a system of equations of the following form is obtained

$$P \begin{cases} \mathbf{E}_1(\mathbf{U}_1, \mathbf{U}_2) = \mathbf{0} \\ \mathbf{E}_2(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3) = \mathbf{0} \\ \vdots \\ \mathbf{E}_{N+1}(\mathbf{U}_N, \mathbf{U}_{N+1}) = \mathbf{0} \end{cases} \quad (5.1)$$

It has been assumed that the equation at node i involves only the nodal states at nodes $i - 1$, i and $i + 1$, as is true for first order Finite Difference Method and FEM discretization

methods. Also a steady system of equations is assumed. The system (5.1) represents $(N + 1) \times n_{\text{dof}}$ equations in the $(N + 1) \times n_{\text{dof}}$ unknowns $\{\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_{N+1}\}$. It is assumed that this non-linear system of equations has a unique solution. Equations at the boundary nodes may include some mixture of Dirichlet or Robin and Neumann boundary conditions.

This system of equations is split up at a certain internal node i , so that the domain $\Omega = [x_{1,1}, x_{1,N+1}]$ is split in the sub-domains $\Omega_1 = [x_{1,1}, x_{1,i}]$ and $\Omega_2 = [x_{1,i}, x_{1,N+1}]$ (see figure 5.1). Now, node i splits in $i1$ and $i2$ for the left and right sub-domains, respectively. Appropriate boundary conditions at $i1$ ($i2$) for Ω_1 (Ω_2) must be provided. These conditions must ensure that each sub-domain problem can be solved independently, and that the problem is well posed. An iterative approach to solve the problem must guarantee that the solutions at each boundary converge to the solution of the coupled system, *i.e.*

$$\begin{aligned} \mathbf{U}_{i1}^k, \mathbf{U}_{i2}^k &\rightarrow \mathbf{U}_i \\ \mathbf{U}_j^k &\rightarrow \mathbf{U}_j \quad \forall j, j \neq i \end{aligned} \quad (5.2)$$

for $k \rightarrow \infty$, k being the iteration number.

In addition, it is assumed that the equation at node i can be separated in its right and left contributions

$$\mathbf{E}_i(\mathbf{U}_{i-1}, \mathbf{U}_i, \mathbf{U}_{i+1}) = \mathbf{E}_{i1}(\mathbf{U}_{i-1}, \mathbf{U}_i) + \mathbf{E}_{i2}(\mathbf{U}_i, \mathbf{U}_{i+1}) = \mathbf{0} \quad (5.3)$$

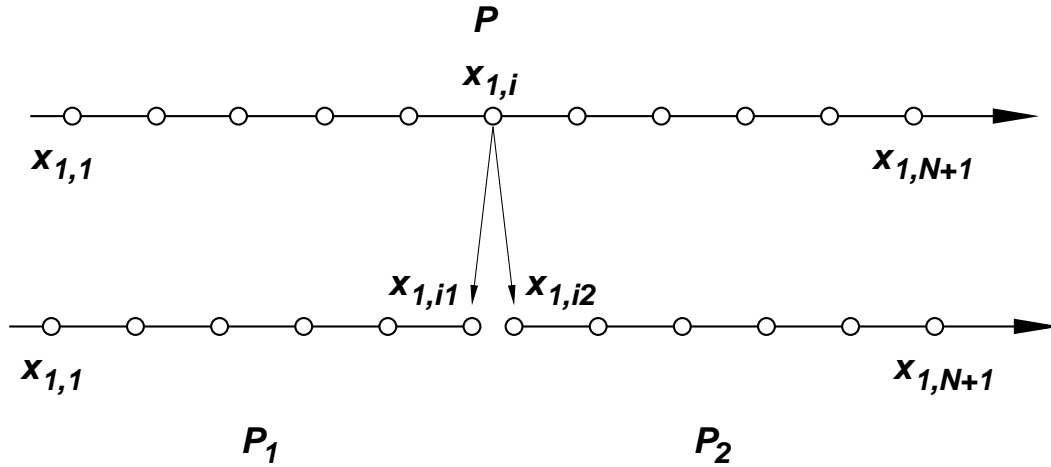


Figure 5.1: Sketch of 1D domain splitting.

In this thesis, non-overlapping sub-domain coupling methods will be considered.

5.1 Coupling domains through absorbing boundary conditions

The coupling interface is an inlet/outlet for the left and right sub-domains. Thus, an approach to solve the coupling of domains could be by using absorbing boundary conditions at the coupling interface. The technique presented in section §1.1.2 to solve dynamic boundary conditions will be considered. In order to formulate the coupling strategy based

on absorbing boundary conditions, some options to solve the problem are analyzed. One possibility is at each step to solve the left system using Dirichlet boundary conditions at $i1$, and compute the (total) flux at this boundary. Then, this flux is passed to the right domain as a Robin boundary condition and the solution at the right domain is computed. This, in turn, produces a value at the boundary \mathbf{U}_{i2} which is passed as a new Dirichlet boundary condition at the left boundary, and so on. Others combinations of Dirichlet, Neumann and Robin boundary conditions are possible.

5.1.1 Algorithm Robin/Robin

For instance, the iterative algorithm using a Robin/Robin (R/R) approach is written as

$$\begin{aligned}
 P_1 \left\{ \begin{array}{l} \mathbf{E}_1(\mathbf{U}_1^{k+1}, \mathbf{U}_2^{k+1}) = \mathbf{0} \\ \mathbf{E}_2(\mathbf{U}_1^{k+1}, \mathbf{U}_2^{k+1}, \mathbf{U}_3^{k+1}) = \mathbf{0} \\ \vdots \\ \mathbf{E}_{i1}(\mathbf{U}_{i-1}^{k+1}, \mathbf{U}_{i1}^{k+1}) + \mathbf{R}^k = \mathbf{0} \end{array} \right. \\
 P_2 \left\{ \begin{array}{l} \mathbf{E}_{i2}(\mathbf{U}_{i2}^{k+1}, \mathbf{U}_{i+1}^{k+1}) - \mathbf{R}^k = \mathbf{0} \\ \mathbf{E}_{i+1}(\mathbf{U}_{i2}^{k+1}, \mathbf{U}_{i+1}^{k+1}, \mathbf{U}_{i+2}^{k+1}) = \mathbf{0} \\ \vdots \\ \mathbf{E}_{N+1}(\mathbf{U}_N^{k+1}, \mathbf{U}_{N+1}^{k+1}) = \mathbf{0} \end{array} \right. \quad (5.4) \\
 \mathbf{R}^{k+1} = \mathbf{R}^k + \tilde{\omega}(\mathbf{U}_{i1}^{k+1} - \mathbf{U}_{i2}^{k+1})
 \end{aligned}$$

Note that

- Terms \mathbf{R}^k are ‘reactions’, *i.e.* terms that represent the fluxes that come from the opposite domain or from a Dirichlet boundary condition.
- Both sub-domain problems P_1 and P_2 can be solved independently at each step. They only need \mathbf{R}^k , which is evaluated in the previous step.
- The new value \mathbf{R}^{k+1} is computed as the sum of \mathbf{R}^k and an increment proportional to the imbalance of \mathbf{U} at the boundary with a relaxation factor ($\tilde{\omega}$).

It can be shown that if such a scheme converges, then it converges to the solution of the original problem. Effectively, letting $k \rightarrow \infty$, the last equation in (5.4) reduces to $\mathbf{U}_{i1} = \mathbf{U}_{i2}$. Replacing this in the $i1$ and $i2$ equations, and adding them, the reaction \mathbf{R} is eliminated and the original equation (5.3) is recovered.

A key problem is to determine the best combinations of boundary conditions that make the fixed point algorithm to converge as fast as possible. In the context of the solution of hyperbolic systems of equations, some combinations of boundary conditions can be much better than others depending on the direction of waves propagation.

5.1.2 Robin/Dirichlet scheme for right-going characteristics

If all characteristics go right (*i.e.* the sense of propagation is positive), then it is natural to use Dirichlet boundary conditions on the right domain, and Robin on the left. The

algorithm R/D is as follows

$$\begin{aligned}
P_1 & \begin{cases} \mathbf{E}_1(\mathbf{U}_1^{k+1}, \mathbf{U}_2^{k+1}) = \mathbf{0} \\ \mathbf{E}_2(\mathbf{U}_1^{k+1}, \mathbf{U}_2^{k+1}, \mathbf{U}_3^{k+1}) = \mathbf{0} \\ \vdots \\ \mathbf{E}_{i1}(\mathbf{U}_{i-1}^{k+1}, \mathbf{U}_{i1}^{k+1}) + \mathbf{R}^k = \mathbf{0} \end{cases} \\
P_2 & \begin{cases} \mathbf{U}_{i2}^{k+1} = \mathbf{U}_{i1}^k \\ \mathbf{E}_{i+1}(\mathbf{U}_{i2}^{k+1}, \mathbf{U}_{i+1}^{k+1}, \mathbf{U}_{i+2}^{k+1}) = \mathbf{0} \\ \vdots \\ \mathbf{E}_{N+1}(\mathbf{U}_N^{k+1}, \mathbf{U}_{N+1}^{k+1}) = \mathbf{0} \end{cases} \\
\mathbf{R}^{k+1} & = \mathbf{E}_{i2}(\mathbf{U}_{i2}^{k+1}, \mathbf{U}_{i+1}^{k+1})
\end{aligned} \tag{5.5}$$

The counting of equations and unknowns for the P_1 problem is the same as for a standard problem with Robin boundary conditions at the $i1$ -th node. There are $i \times n_{\text{dof}}$ equations in the $i \times n_{\text{dof}}$ unknowns $\{\mathbf{U}_1, \dots, \mathbf{U}_{i1}\}$ (\mathbf{R}^k is known at the $k+1$ stage, since it comes from the previous stage k). On the other hand, the counting for P_2 is similar to a problem with Dirichlet boundary conditions at $i2$. It represents $(N-i+1) \times n_{\text{dof}}$ equations on the $\{\mathbf{U}_{i+1}, \dots, \mathbf{U}_{N+1}\}$ unknowns (\mathbf{U}_{i2}^{k+1} is given by the Dirichlet boundary condition).

5.1.3 Dirichlet/Robin scheme for left-going characteristics

If all eigenvalues are left-going, then the appropriate scheme would be

$$\begin{aligned}
P_1 & \begin{cases} \mathbf{E}_1(\mathbf{U}_1^{k+1}, \mathbf{U}_2^{k+1}) = \mathbf{0} \\ \mathbf{E}_2(\mathbf{U}_1^{k+1}, \mathbf{U}_2^{k+1}, \mathbf{U}_3^{k+1}) = \mathbf{0} \\ \vdots \\ \mathbf{U}_{i1}^{k+1} = \mathbf{U}_{i2}^k \end{cases} \\
P_2 & \begin{cases} \mathbf{E}_{i2}(\mathbf{U}_{i2}^{k+1}, \mathbf{U}_{i+1}^{k+1}) - \mathbf{R}^k = \mathbf{0} \\ \mathbf{E}_{i+1}(\mathbf{U}_{i2}^{k+1}, \mathbf{U}_{i+1}^{k+1}, \mathbf{U}_{i+2}^{k+1}) = \mathbf{0} \\ \vdots \\ \mathbf{E}_{N+1}(\mathbf{U}_N^{k+1}, \mathbf{U}_{N+1}^{k+1}) = \mathbf{0} \end{cases} \\
\mathbf{R}^{k+1} & = -\mathbf{E}_{i1}(\mathbf{U}_{i-1}^{k+1}, \mathbf{U}_{i1}^{k+1})
\end{aligned} \tag{5.6}$$

5.1.4 Characteristics-based split for general systems

For a general problem with positive and negative eigenvalues, problems at each side of the coupling interface could be written in the following way

$$\begin{cases}
 P_1 \left\{ \begin{array}{l}
 \mathbf{E}_1(\mathbf{U}_1^{k+1}, \mathbf{U}_2^{k+1}) = \mathbf{0} \\
 \mathbf{E}_2(\mathbf{U}_1^{k+1}, \mathbf{U}_2^{k+1}, \mathbf{U}_3^{k+1}) = \mathbf{0} \\
 \vdots \\
 \mathbf{E}_{i1}(\mathbf{U}_{i-1}^{k+1}, \mathbf{U}_{i1}^{k+1}) + \mathbf{\Pi}_{U1}^+ \mathbf{R}^k + \mathbf{\Pi}_{U1}^- \mathbf{R}^{k+1} = \mathbf{0} \\
 \mathbf{\Pi}_{U1}^-(\mathbf{U}_{i1}^{k+1} - \mathbf{U}_{i2}^k) = \mathbf{0} \\
 \mathbf{E}_{i2}(\mathbf{U}_{i2}^{k+1}, \mathbf{U}_{i+1}^{k+1}) - \mathbf{\Pi}_{U2}^+ \mathbf{R}^k - \mathbf{\Pi}_{U2}^- \mathbf{R}^{k+1} = \mathbf{0} \\
 \mathbf{\Pi}_{U2}^-(\mathbf{U}_{i1}^k - \mathbf{U}_{i2}^{k+1}) = \mathbf{0} \\
 \mathbf{E}_{i+1}(\mathbf{U}_{i2}^{k+1}, \mathbf{U}_{i+1}^{k+1}, \mathbf{U}_{i+2}^{k+1}) = \mathbf{0} \\
 \vdots \\
 \mathbf{E}_{N+1}(\mathbf{U}_N^{k+1}, \mathbf{U}_{N+1}^{k+1}) = \mathbf{0}
 \end{array} \right. \\
 P_2 \left\{ \begin{array}{l}
 \mathbf{E}_{i2}(\mathbf{U}_{i2}^{k+1}, \mathbf{U}_{i+1}^{k+1}) - \mathbf{\Pi}_{U2}^+ \mathbf{R}^k - \mathbf{\Pi}_{U2}^- \mathbf{R}^{k+1} = \mathbf{0} \\
 \mathbf{\Pi}_{U2}^-(\mathbf{U}_{i1}^k - \mathbf{U}_{i2}^{k+1}) = \mathbf{0} \\
 \mathbf{E}_{i+1}(\mathbf{U}_{i2}^{k+1}, \mathbf{U}_{i+1}^{k+1}, \mathbf{U}_{i+2}^{k+1}) = \mathbf{0} \\
 \vdots \\
 \mathbf{E}_{N+1}(\mathbf{U}_N^{k+1}, \mathbf{U}_{N+1}^{k+1}) = \mathbf{0}
 \end{array} \right.
 \end{cases} \quad (5.7)$$

where $\mathbf{\Pi}_{Uj}^\pm = \mathbf{\Pi}_{Uj}^\pm(\mathbf{U}_{ij})$, $j = 1, 2$ are the projection matrices onto the right/left-going characteristic modes [72] computed with the outward unit normal (n_j) at the coupling interface for the sub-domain j . Due to $n_1 = -n_2$, if both projection matrices are computed at the same state, then $\mathbf{\Pi}_{U1}^\pm = \mathbf{\Pi}_{U2}^\mp = \mathbf{\Pi}_U^\pm$. Note that after each iteration P_1 provides the left going part of the new reactions ($\mathbf{\Pi}_U^- \mathbf{R}^{k+1}$), whereas P_2 provides the right going part. Altogether, the whole reaction vector \mathbf{R}^{k+1} is obtained.

Counting of equations and unknowns is a little more complicated in this case. There are $in_{\text{dof}} + n_-$ equations and unknowns for P_1 , where n_- is the number of left-going characteristics. The n_- equations come from the last row of equations, since $\mathbf{\Pi}_U^-$ has rank n_- . The n_- additional unknowns come from the term $\mathbf{\Pi}_U^- \mathbf{R}^{k+1}$, since the term $\mathbf{\Pi}_U^+ \mathbf{R}^k$ is known. For the problem P_2 , the counting of equations and unknowns is analogue.

Again, if the system converges it can be shown that it converges to the solution of the coupled problem. Effectively, if the scheme converges the following system is obtained

$$\begin{cases}
 P_1 \left\{ \begin{array}{l}
 \mathbf{E}_1(\mathbf{U}_1, \mathbf{U}_2) = \mathbf{0} \\
 \vdots \\
 \mathbf{E}_{i1}(\mathbf{U}_{i-1}, \mathbf{U}_{i1}) + \mathbf{\Pi}_{U1}^+ \mathbf{R} + \mathbf{\Pi}_{U1}^- \mathbf{R} = \mathbf{0} \\
 \mathbf{\Pi}_{U1}^-(\mathbf{U}_{i1} - \mathbf{U}_{i2}) = \mathbf{0} \\
 \mathbf{E}_{i2}(\mathbf{U}_{i2}, \mathbf{U}_{i+1}) - \mathbf{\Pi}_{U2}^+ \mathbf{R} - \mathbf{\Pi}_{U2}^- \mathbf{R} = \mathbf{0} \\
 \mathbf{\Pi}_{U2}^-(\mathbf{U}_{i1} - \mathbf{U}_{i2}) = \mathbf{0} \\
 \mathbf{E}_{i+1}(\mathbf{U}_{i2}, \mathbf{U}_{i+1}, \mathbf{U}_{i+2}) = \mathbf{0} \\
 \vdots \\
 \mathbf{E}_{N+1}(\mathbf{U}_N, \mathbf{U}_{N+1}) = \mathbf{0}
 \end{array} \right. \\
 P_2 \left\{ \begin{array}{l}
 \mathbf{E}_{i2}(\mathbf{U}_{i2}, \mathbf{U}_{i+1}) - \mathbf{\Pi}_{U2}^+ \mathbf{R} - \mathbf{\Pi}_{U2}^- \mathbf{R} = \mathbf{0} \\
 \mathbf{\Pi}_{U2}^-(\mathbf{U}_{i1} - \mathbf{U}_{i2}) = \mathbf{0} \\
 \mathbf{E}_{i+1}(\mathbf{U}_{i2}, \mathbf{U}_{i+1}, \mathbf{U}_{i+2}) = \mathbf{0} \\
 \vdots \\
 \mathbf{E}_{N+1}(\mathbf{U}_N, \mathbf{U}_{N+1}) = \mathbf{0}
 \end{array} \right.
 \end{cases} \quad (5.8)$$

Adding restrictions,

$$\mathbf{\Pi}_{U1}^-(\mathbf{U}_{i1} - \mathbf{U}_{i2}) + \mathbf{\Pi}_{U2}^-(\mathbf{U}_{i1} - \mathbf{U}_{i2}) = (\mathbf{\Pi}_{U1}^- + \mathbf{\Pi}_{U2}^-)(\mathbf{U}_{i1} - \mathbf{U}_{i2}) = \mathbf{0} \quad (5.9)$$

implies $\mathbf{U}_{i1} = \mathbf{U}_{i2}$ and $\mathbf{\Pi}_{U1}^{\pm} = \mathbf{\Pi}_{U2}^{\mp} = \mathbf{\Pi}_U^{\pm}$, as noted above. Again, adding the $i1$ -th and $i2$ -th equations, the original i -th equation (5.3) is recovered. The advantage of this scheme is that for each sub-domain the system of equations obtained is equivalent to a problem with absorbing boundary conditions.

5.1.5 Resolution by penalization

In this section, the algorithm to solve by penalization the scheme proposed in equation (5.7) is presented. Assuming $\mathbf{\Pi}_{U1}^{\pm} = \mathbf{\Pi}_{U2}^{\mp} = \mathbf{\Pi}_U^{\pm}$, both restrictions can be added

$$\mathbf{\Pi}_U^-(\mathbf{U}_{i1}^{k+1} - \mathbf{U}_{i2}^k) + \mathbf{\Pi}_U^+(\mathbf{U}_{i1}^k - \mathbf{U}_{i2}^{k+1}) = \mathbf{0} \quad (5.10)$$

since each part can be recovered by multiplying at right by $\mathbf{\Pi}_U^-$ or $\mathbf{\Pi}_U^+$. For instance if it is multiplied by $\mathbf{\Pi}_U^+$, the equation

$$\mathbf{\Pi}_U^+(\mathbf{U}_{i1}^k - \mathbf{U}_{i2}^{k+1}) = \mathbf{0} \quad (5.11)$$

is recovered, because $\mathbf{\Pi}_U^+\mathbf{\Pi}_U^+ = \mathbf{\Pi}_U^+$ and $\mathbf{\Pi}_U^+\mathbf{\Pi}_U^- = \mathbf{0}$

Now, in order to ‘regularize’ the problem, a small term can be added to equation (5.10)

$$\mathbf{\Pi}_U^-(\mathbf{U}_{i1}^{k+1} - \mathbf{U}_{i2}^k) + \mathbf{\Pi}_U^+(\mathbf{U}_{i1}^k - \mathbf{U}_{i2}^{k+1}) - \epsilon(\mathbf{R}^{k+1} - \mathbf{R}^k) = \mathbf{0} \quad (5.12)$$

in such a way \mathbf{R}^{k+1} can be eliminated and a penalized version is obtained.

$$\mathbf{R}^{k+1} = \mathbf{R}^k + \frac{1}{\epsilon} [\mathbf{\Pi}_U^-(\mathbf{U}_{i1}^{k+1} - \mathbf{U}_{i2}^k) + \mathbf{\Pi}_U^+(\mathbf{U}_{i1}^k - \mathbf{U}_{i2}^{k+1})] \quad (5.13)$$

Replacing in the $i1$ -th equation

$$\begin{aligned} \mathbf{E}_{i1}(\mathbf{U}_{i-1}^{k+1}, \mathbf{U}_{i1}^{k+1}) + \mathbf{\Pi}_U^+\mathbf{R}^k + \mathbf{\Pi}_U^-\mathbf{R}^{k+1} &= \mathbf{0} \\ \mathbf{E}_{i1}(\mathbf{U}_{i-1}^{k+1}, \mathbf{U}_{i1}^{k+1}) + \mathbf{\Pi}_U^+\mathbf{R}^k + \mathbf{\Pi}_U^-\mathbf{R}^k + \frac{1}{\epsilon}\mathbf{\Pi}_U^-(\mathbf{U}_{i1}^{k+1} - \mathbf{U}_{i2}^k) &= \mathbf{0} \\ \mathbf{E}_{i1}(\mathbf{U}_{i-1}^{k+1}, \mathbf{U}_{i1}^{k+1}) + \mathbf{R}^k + \frac{1}{\epsilon}\mathbf{\Pi}_U^-(\mathbf{U}_{i1}^{k+1} - \mathbf{U}_{i2}^k) &= \mathbf{0} \end{aligned} \quad (5.14)$$

In a similar way, the equation for the $i2$ -th node can be written as

$$\mathbf{E}_{i2}(\mathbf{U}_{i2}^{k+1}, \mathbf{U}_{i+1}^{k+1}) - \mathbf{R}^k - \frac{1}{\epsilon}\mathbf{\Pi}_U^+(\mathbf{U}_{i1}^k - \mathbf{U}_{i2}^{k+1}) = \mathbf{0} \quad (5.15)$$

Some simple examples were solved with this coupling strategy based on absorbing boundary conditions. The cases successfully solved consist in splitted 1D domains where a perturbation is created in the state as initial condition of the flow. The results obtained in such cases are good when they are compared with solutions in a unique domain, but the convergence rate is generally low. The strategy fails when it is applied to solve the first test of Sod. As it is known, this test presents discontinuities in their solution [33] which causes the failure of the coupling strategy. The inability of the technique to solve flows with discontinuities is an important drawback for its use in IC engines simulation since, for instance, contact discontinuities are present in exhaust pipes [32].

Due to the fact that it is not the goal here to develop a general coupling technique able to work with different codes, other useful strategy for the purpose of this thesis is presented in the following section. Nevertheless, the coupling strategy based on absorbing boundary conditions deserves further development which is proposed as future work.

5.2 Coupling for implicit schemes ‘monolithically’ solved

The coupling strategy presented in the previous section could be useful when the governing equations on each domain are solved by different codes. Using an implicit scheme for time integration and assuming that the resolution could be performed as a ‘monolithic’ system, the coupling strategy reduces to a constraint between the states at the interface nodes. This strategy could be useful when the codes that perform the computation on each sub-domain are not ‘black boxes’, but the contributions to the global residue (and perhaps the global jacobian matrix) are available.

For the problem (5.1), if the equality of states at the coupling node is imposed, then it results in the linear constraint $\mathbf{U}_{i1} = \mathbf{U}_{i2}$. With the notation of the previous section and using Lagrange multipliers (\mathbf{U}_{lm}) to impose the constraint, the system of equations is expressed as

$$\left\{ \begin{array}{l} \mathbf{E}_1(\mathbf{U}_1, \mathbf{U}_2) = \mathbf{0} \\ \mathbf{E}_2(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3) = \mathbf{0} \\ \vdots \\ \mathbf{E}_{i1}(\mathbf{U}_{i-1}, \mathbf{U}_{i1}) + \mathbf{U}_{\text{lm}} = \mathbf{0} \\ \mathbf{U}_{i1} - \mathbf{U}_{i2} = \mathbf{0} \\ \mathbf{E}_{i2}(\mathbf{U}_{i2}, \mathbf{U}_{i+1}) - \mathbf{U}_{\text{lm}} = \mathbf{0} \\ \vdots \\ \mathbf{E}_{N+1}(\mathbf{U}_N, \mathbf{U}_{N+1}) = \mathbf{0} \end{array} \right. \quad (5.16)$$

where the constraint $\mathbf{U}_{i1} - \mathbf{U}_{i2} = \mathbf{0}$ imposes the continuity of the solution and forces the continuity of fluxes through the coupling interface. By elimination and using (5.3), it is easy to see that the problem (5.16) reduces to (5.1).

5.2.1 Coupling of 1D/multi-D domains

If the purpose is to link a multi-D domain with a 1D domain, the constraints are not straightforward as in the 1D/1D coupling. In this case, the conditions at the coupling interface for the multi-D domain are defective [24, 41] and a simplification must be done at the coupling interface.

One possibility is to impose each node at coupling surface of the multi-D domain to have the same state as the corresponding node in the 1D domain. Let M the number of nodes lying on the coupling interface of the multi-D domain, and let j the node identifying the end of the 1D domain. Then, the Mn_{dof} constraints are

$$\mathbf{U}_j = \mathbf{U}_i \quad i = 1, \dots, M \quad (5.17)$$

This kind of coupling could be useful with a uniform flow (‘piston’-like) through the multi-D domain boundary, limiting the shape of variables profile in the coupling interface. For instance, this limitation does not allow to apply a no-slip boundary condition at walls, even whether the flow is parallel to the wall.

Another option is to equalize the mean value of the state on the whole coupling surface and the corresponding state in the 1D domain, *i.e.*

$$\int_S \mathbf{U}_{MD} dS = \int_S \mathbf{U}_{1D} dS = \mathbf{U}_{1D} S \quad (5.18)$$

where S is the coupling surface. The constraint (5.18) could cause the code failure since the problem is not restricted enough, without guarantee of a unique solution. For example, it is possible to obtain a profile of density with non physical negative values on the coupling surface and whose mean value is a feasible value.

5.3 Results

Several test cases were solved using the coupling strategy based on a monolithic system with linear constraints, as presented above. The following sections show some of these results for 1D/1D, 2D/1D and 3D/1D couplings. In the whole set of test cases, the ‘standard’ Navier-Stokes equations for compressible flow are solved, *i.e.* the equations (1.8) with the variational formulation given by (1.28).

5.3.1 1D/1D coupling

In this particular case, strategies (5.17) and (5.18) are equivalent. Using two or more sub-domains, the solution obtained is equal to the solution computed from a unique domain.

5.3.2 2D/1D coupling

The 2D/1D coupling test proposed consists in a reservoir connected to the atmosphere by a pipe with length $L = 7$ m. Pressure and temperature of gas into the reservoir are constant, with values of $p_0 = 1.2 \times 10^5$ Pa and $T_0 = 278.75$ K, respectively. The atmospheric pressure is $p_a = 1 \times 10^5$ Pa. Initially, the gas inside the pipe is at rest, and at the same pressure and density as the gas into the reservoir. At $t = 0$, the pipe end connected to the atmosphere is suddenly open. The pipe is modeled using three sub-domains, 1D at the pipe ends and 2D in its middle region as shown in figure 5.2 (1D-2D-1D). Also, in order to compare the solutions, the problem is solved using three 1D domains (1D-1D-1D). The duct is adiabatic and without friction, and the fluid is inviscid.

The 2D domain is discretized by using an unstructured mesh containing 592 triangular elements. For 1D domains uniform meshes are used, with element size $h = 0.05$ m. The time step in the simulations is $\Delta t = 1 \times 10^{-4}$ s, which gives a CFL number $\mathcal{O}(1)$. The left coupling section is located at $x_1 = 2$ m, and the right coupling section at $x_1 = 5$ m.

Solving the problem with the constraints given by equation (5.17) for the density, the pressure and the mean axial velocity, the test is successful and the whole time interval can be solved. Figures 5.3(a) to 5.3(c) show the results achieved at coupling sections. As expected, there are no differences between the cases 1D-2D-1D and 1D-1D-1D.

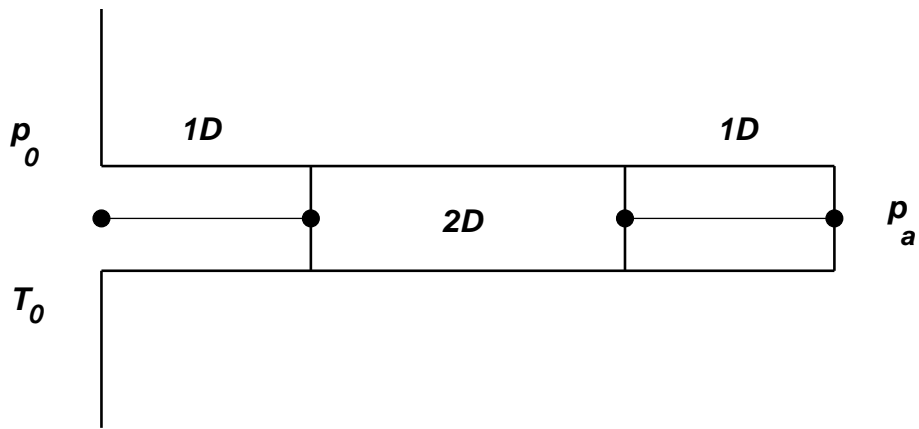
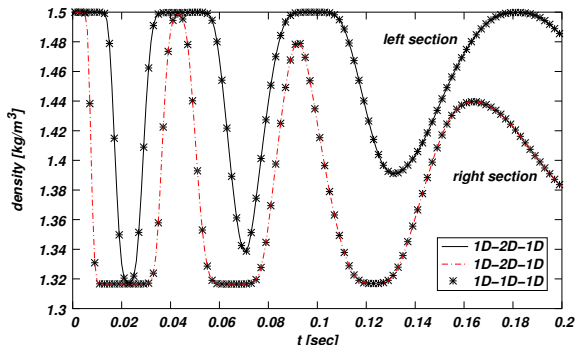
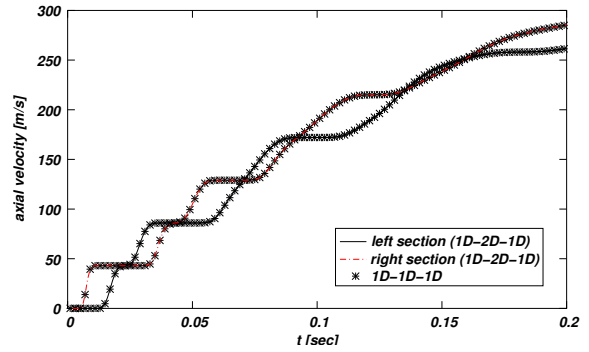


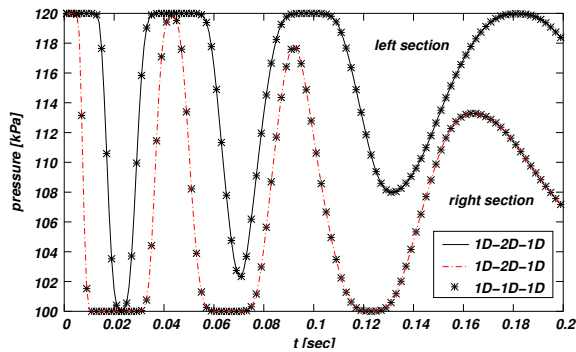
Figure 5.2: Gas discharge from a reservoir to the atmosphere.



(a) Density



(b) Axial velocity



(c) Pressure

Figure 5.3: Time evolution of solution at coupling sections.

5.3.3 3D/1D coupling

This case consists in the exhaust manifold of a six-cylinder spark-ignition four-stroke engine¹. The purpose is to solve a junction of the manifold with a CFD-3D code and to solve the rest of the engine with 0D/1D models. The exhaust manifold is composed by three junctions, as it is shown in figure 5.4. Two of these junctions connect three header pipes with one of the two intermediate pipes (junctions 3-to-1). These pipes converge to the third junction that connects them with the exhaust tailpipe (junction 2-to-1). The diameters and lengths of the pipes in the manifold are the following

- Header pipes: $D = 51.7$ mm, $L = 570$ mm.
- Intermediate pipes: $D = 61$ mm, $L = 650$ mm.
- Tailpipe: $D = 70 - 85$ mm, $L = 600$ mm.

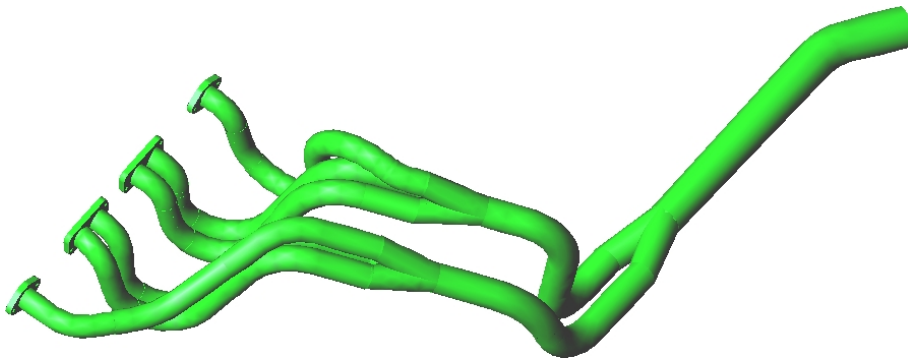


Figure 5.4: Exhaust manifold geometry.

One of the junctions 3-to-1 is modeled as a 3D domain, and coupled to 1D domains representing the pipes connected with it. The geometric model is shown in figure 5.5, where a (relatively short) stretch of the pipes were added to the 3D model of the junction in order to avoid the failure of the 1D approximation at the coupling section. With the aim of simplifying the resolution, only the branch of the exhaust manifold containing the junction 3-to-1 is solved. In figure 5.6 a sketch of the computational model of the branch is presented. This figure shows two models, one of them is composed by four 1D pipes and the 3D junction, while in the other one the junction is represented as a 0D component. In this last case, the model by Corberan (see section §4.1.4) is applied to solve the pipe junction. Thus, a comparison of the solution in the coupling interfaces between both computational models is done and a verification of the hypothesis in the junction model by Corberan is made also.

The boundary conditions for these models are obtained simulating the whole engine by using the 0D/1D code until a stationary (cyclic) state is reached. The engine speed is 8000 rpm and, therefore, the period of the cycle is $\Upsilon = 0.015$ s. For the 3D junction, insulated walls are considered and slip condition is imposed on them. At coupling interfaces, the density and pressure over the whole section are equalized to the corresponding value of the 1D domains. In addition, the component of the velocity in the normal direction to

¹I want to thank to Juan Pablo Alianak for provide me the computational 3D geometric model of the six-cylinder exhaust manifold.

the surface is equalized to the axial velocity at end pipes connected to the 3D domain. The tangential components of velocity on inlet/outlet sections are constrained to be null. The mesh of the 3D model has 278K tetrahedra and 67K nodes, and the 1D domains were discretized by means of uniform meshes with element size of $h = 5$ mm. The time step of the simulation is $\Delta t = 5 \times 10^{-5}$ s. Several periods were simulated using the monolithic coupling strategy until to reach (approximately) a stationary state in the solution. For the 1D-1D/0D-1D model, the discretizations in time and space are the same as in the 1D-3D-1D model.



Figure 5.5: Geometry of the junction 3-to-1.

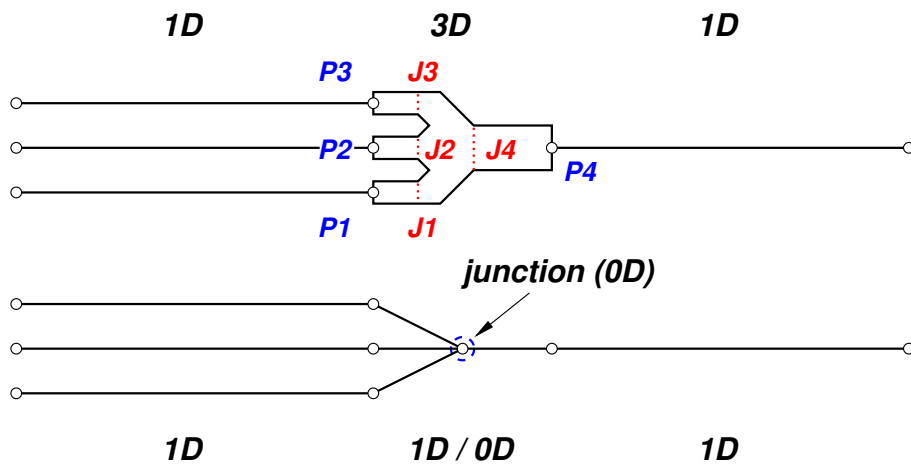


Figure 5.6: Computational models to solve a branch of the exhaust manifold.

Some instantaneous distributions of the pressure over the junction skin and streamlines are shown in figures 5.7 to 5.12, where

$$t^* = \frac{t - t_0}{\Upsilon} \quad (5.19)$$

is the non-dimensional time, being t_0 the start time of the cycle.

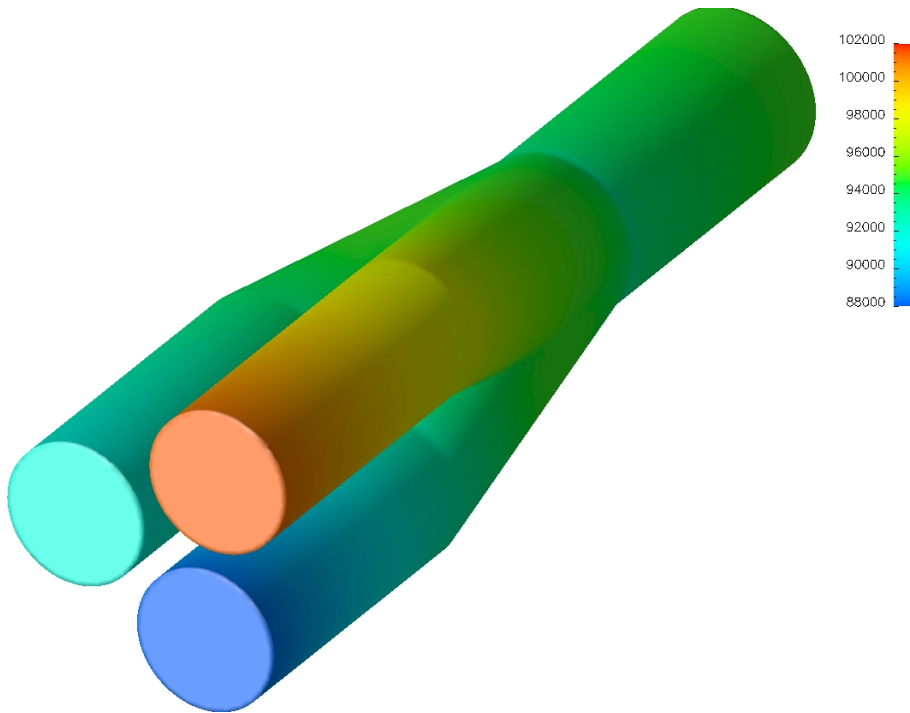


Figure 5.7: Pressure field over the junction 3-to-1 skin at $t^* = 0$.

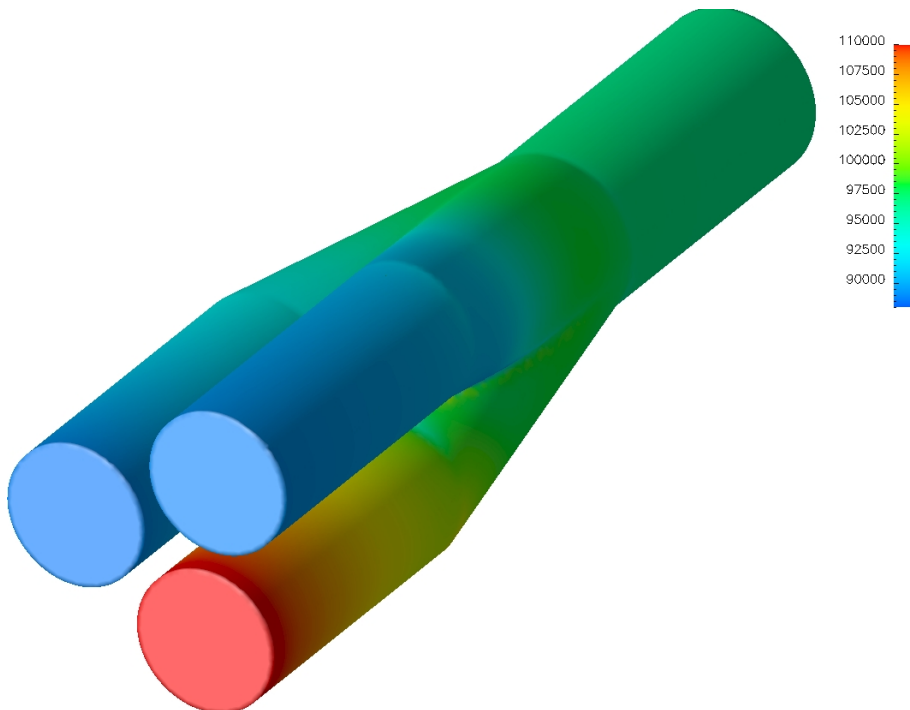


Figure 5.8: Pressure field over the junction 3-to-1 skin at $t^* = 0.4$.

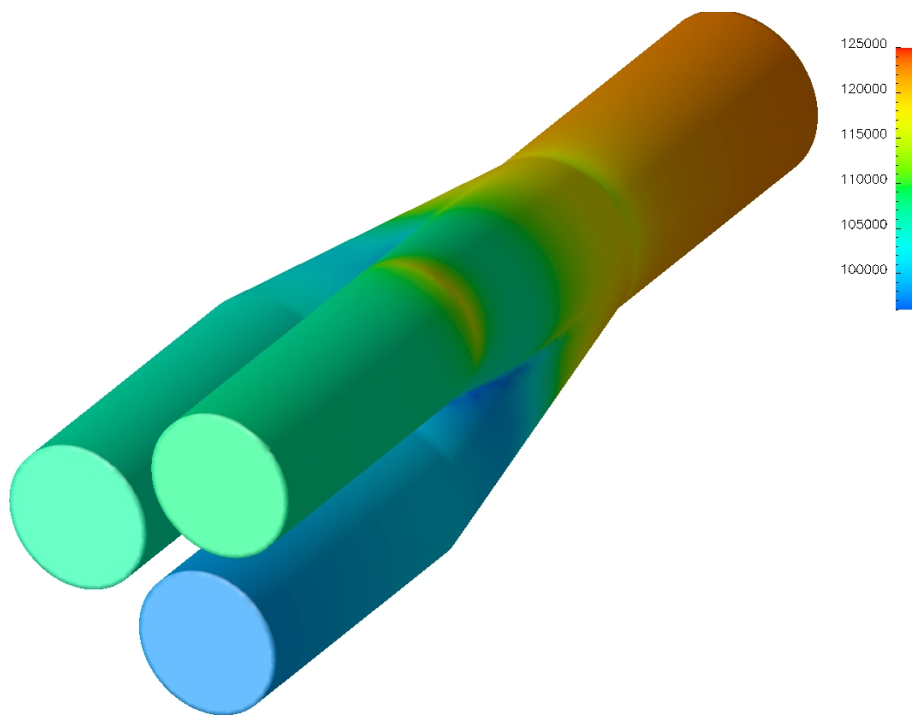


Figure 5.9: Pressure field over the junction 3-to-1 skin at $t^* = 0.8$.

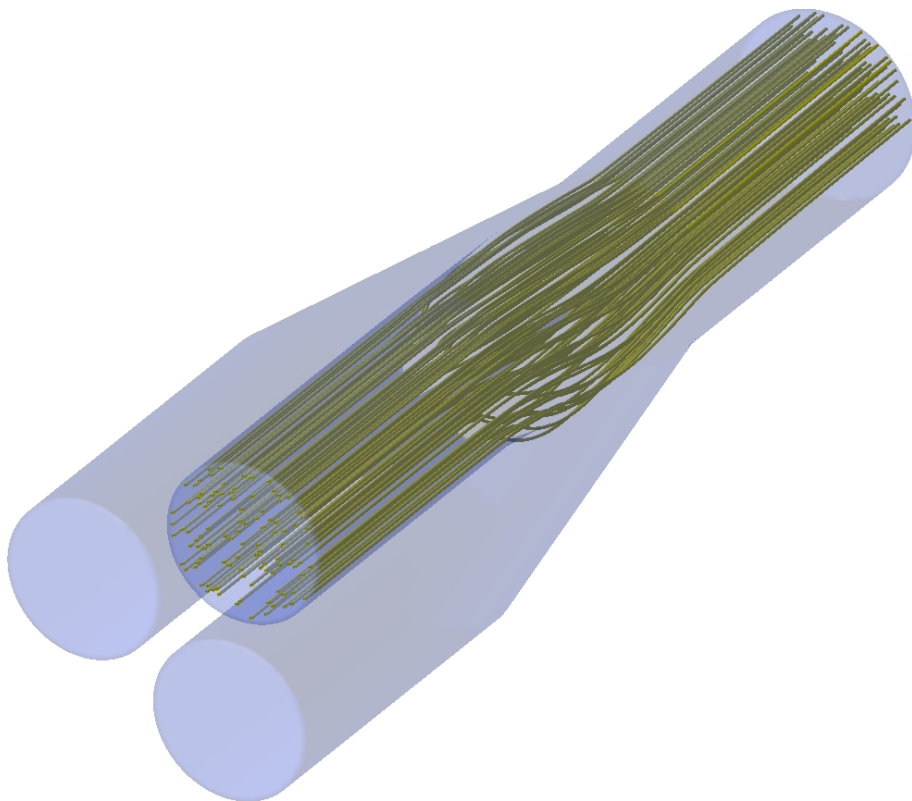


Figure 5.10: Streamlines at $t^* = 0$ for the junction 3-to-1.

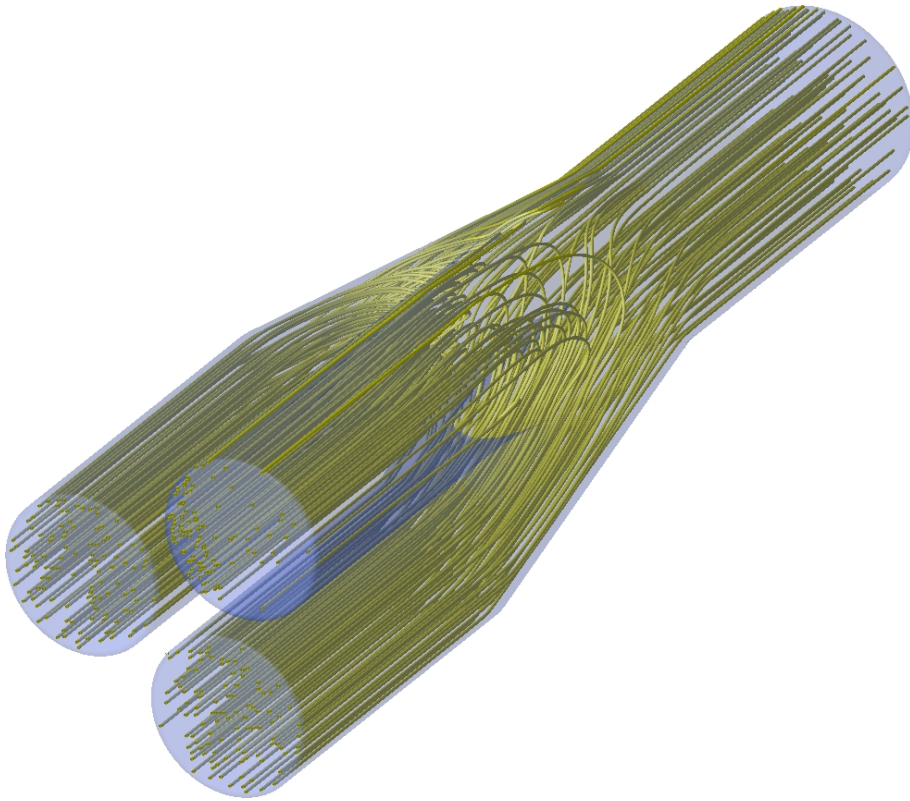


Figure 5.11: Streamlines at $t^* = 0.4$ for the junction 3-to-1.

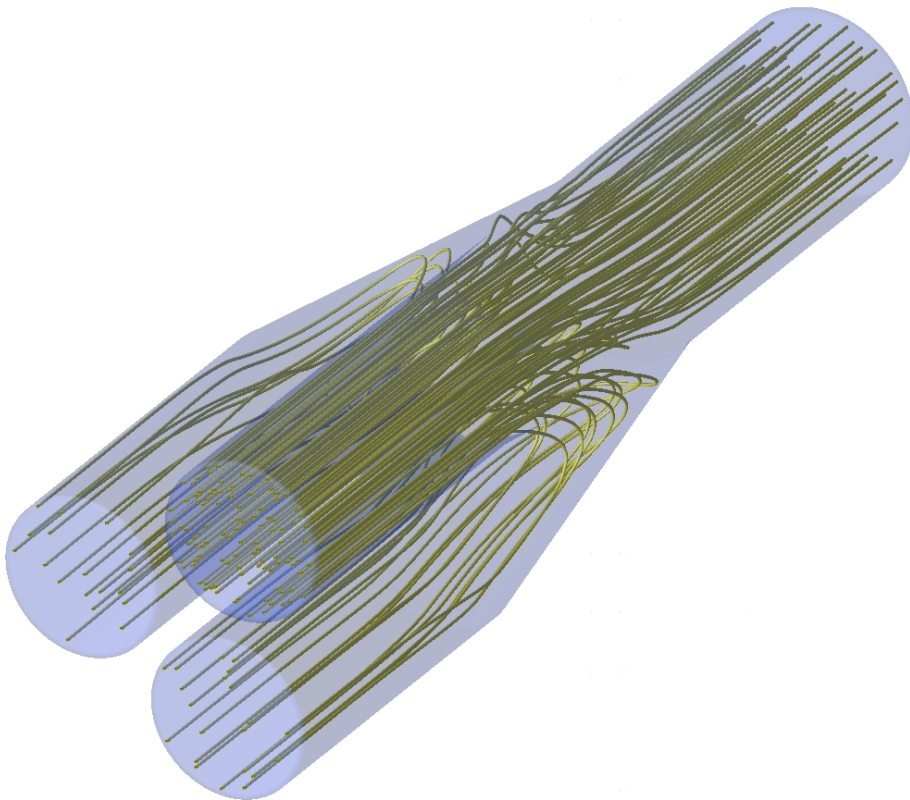


Figure 5.12: Streamlines at $t^* = 0.8$ for the junction 3-to-1.

Solutions from 1D-3D-1D and 1D-1D/0D-1D models are compared at coupling 1D/3D interfaces, named from P1 to P4 as it is indicated in figure 5.6. These comparisons are presented in figures 5.13 to 5.15. In figure 5.14 the velocity is referred as axial velocity which, for the 3D model, should be interpreted as the component of the velocity vector in the normal direction to the given surface.

In the coupling sections of the header pipes, the waves in both models tend to have similar behavior, especially when the pulse of the velocity wave reaches the interface. The solution of the 1D-1D/0D-1D model is more oscillatory than the solution of the other model, which is because the pipe junction 0D model has no inertia. In general, the amplitude values are near each other on both solutions, being more attenuated for the 1D-3D-1D solution. The largest differences are found when the flow is established from the junction to the pipe. The influence of both the finite volume of the pipe junction and its geometric shape is more clearly evident in the solutions obtained in coupling section P4. In this interface, the pressure peaks on the 1D-3D-1D solution are higher than on the 1D-1D/0D-1D model.

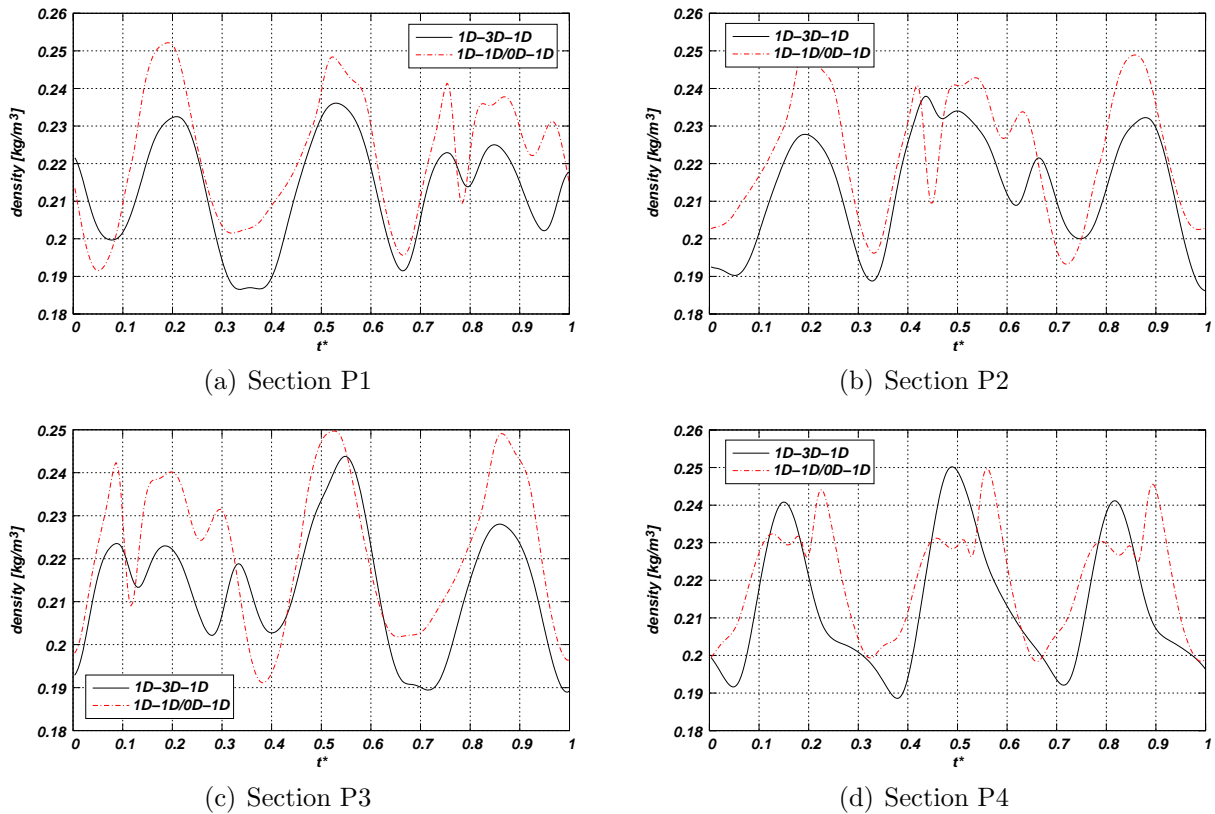


Figure 5.13: Time evolution of density at coupling sections through a cycle.

The CFD 3D could be useful to validate or to improve a 0D model. For the junction 3-to-1 solved, the hypothesis of the pipe junction 0D model by Corberan will be checked. This model is composed by conservation equations, incoming path/Mach lines equations, the equality of the pressure in all pipes end, and the equality of the enthalpy in all outgoing branches in the junction, as presented in section §4.1.4. With the exception of the incoming path/Mach lines equations, the remaining equations can be verified.

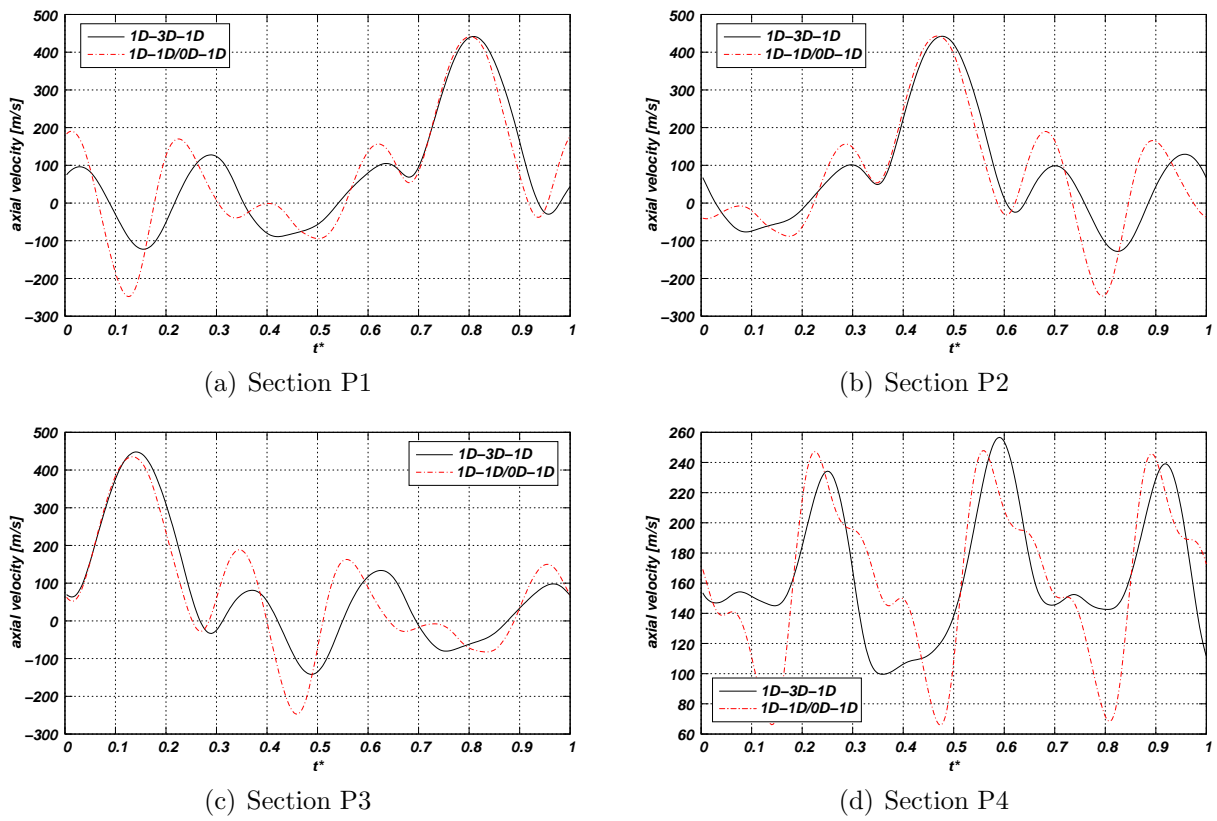


Figure 5.14: Time evolution of axial velocity at coupling sections through a cycle.

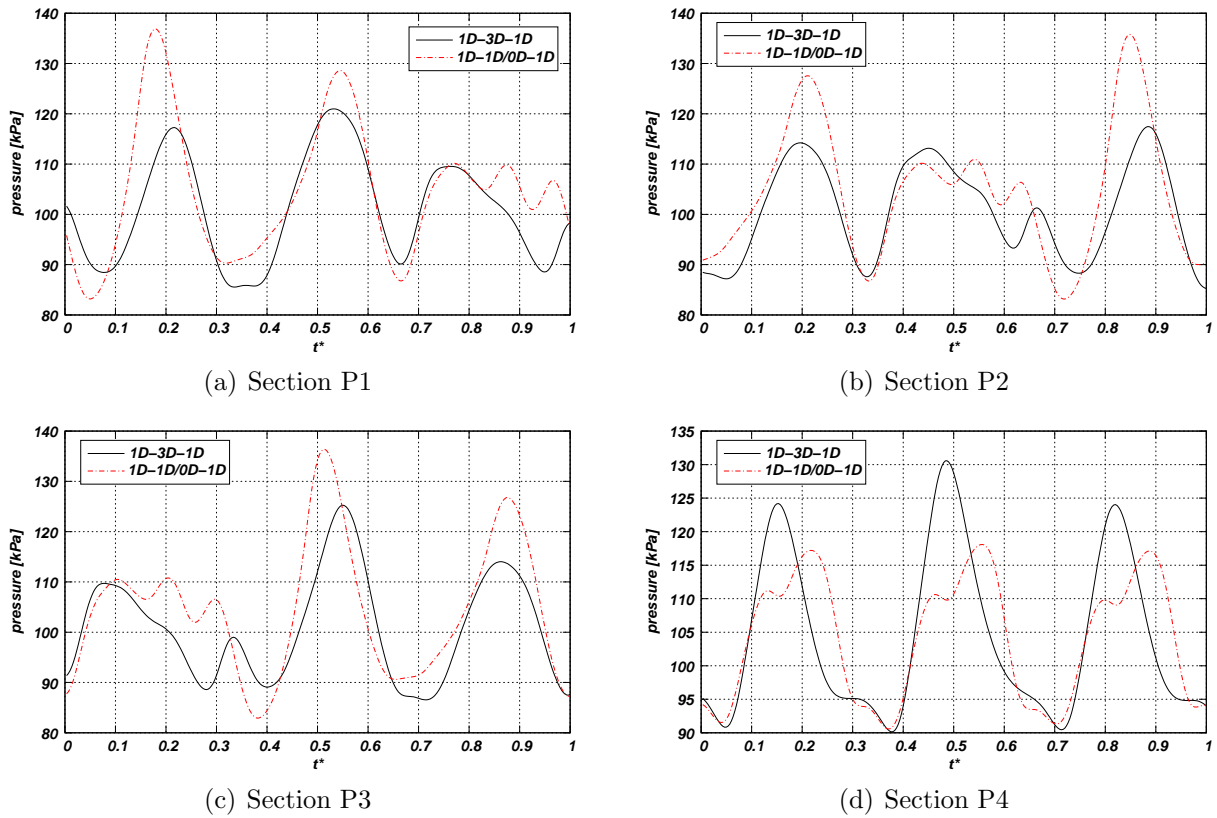


Figure 5.15: Time evolution of pressure at coupling sections through a cycle.

The mass conservation equation is evaluated by computing the following quantity as a function of time

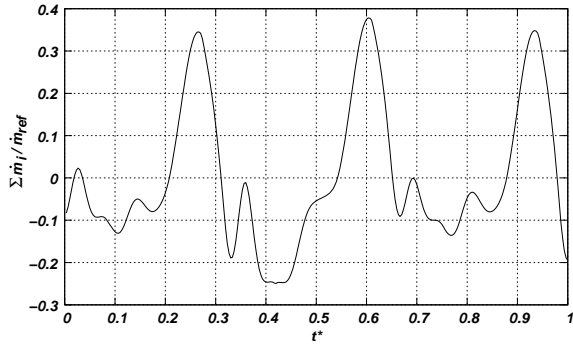
$$\frac{\sum_{i=1}^4 \dot{m}_i}{\dot{m}_{\text{ref}}} \quad (5.20)$$

where $\dot{m}_{\text{ref}} = \sum_{i=1}^4 |\dot{m}_i|$. For the last cycle simulated, the curve for expression (5.20) is plotted in figure 5.16(a). Important deviations from the zero value (as assumed in the 0D model) could be observed in the figure.

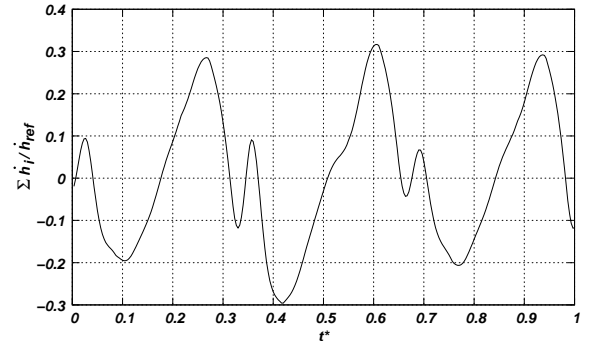
In a similar manner, for the energy conservation it is calculated the following expression

$$\frac{\sum_{i=1}^4 \dot{h}_i}{\dot{h}_{\text{ref}}} \quad (5.21)$$

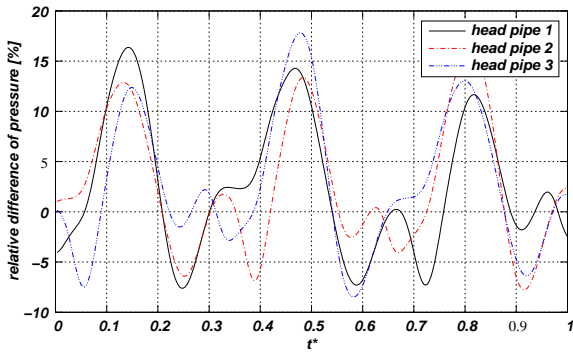
with $\dot{h}_{\text{ref}} = \sum_{i=1}^4 |\dot{h}_i|$. The results obtained are shown in figure 5.16(b) where, again, the difference between the 0D assumption and the 3D solution is relatively high.



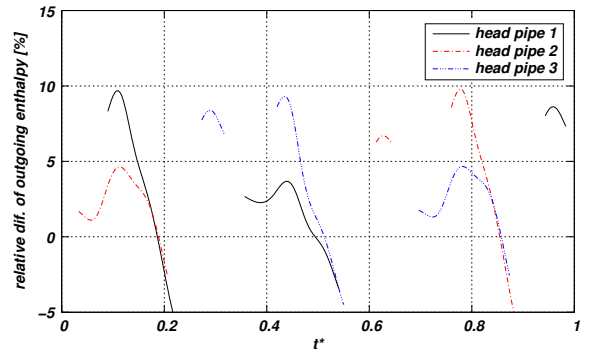
(a) Equilibrium of relative mass fluxes at junction



(b) Equilibrium of relative enthalpy fluxes at junction



(c) Pressure difference with respect to pressure at J4



(d) Outgoing enthalpy difference at junction with respect to enthalpy at J4

Figure 5.16: Relative differences in the hypothesis of the pipe junction 0D model by Corberan.

In figure 5.16(c) the relative difference of pressure is presented, which is computed as

$$\frac{p_4 - p_i}{p_4}, \quad i = 1, 2, 3 \quad (5.22)$$

where p_4 is the mean pressure over the surface corresponding to the interface between the junction and the intermediate pipe (J4, see figure 5.6), and p_i the mean pressure over

the respective surfaces for header pipes (named Ji , $i = 1, 2, 3$ in figure 5.6). With p_4 as reference pressure the maximum error is approximately 18 % and, as could be inferred from the figure, it is expected errors of the same order whether other pressures were adopted as reference.

The final hypothesis in the model by Corberan is the equality of the enthalpy in all outgoing pipes. From the viewpoint of the intermediate pipe the interface J4 is an inlet for all times, then, the enthalpy over such a section (h_4) is adopted as reference. The relative difference of outgoing enthalpy is computed as

$$\frac{h_4 - h_i}{h_4}, \quad i = 1, 2, 3 \quad (5.23)$$

h_i being the mean enthalpy over the section Ji . Figure 5.16(d) shows this relative difference, where the maximum error is about 10 %. As could be noted, the curves are discontinuous due to the change in the sense of flow through the cycle.

Summarizing, for the example solved the major deviations in the assumptions of the Corberan model were found in the conservation of mass and energy. This could be because the volume of the junction is neglected in this 0D model. The hypothesis of equality of pressure in all branches and the equality of the enthalpy in all outgoing pipes present a moderate relative error.

Chapter 6

Numerical simulation of the MRCVC engine

- *Because you were inspired by the devil?*

- *Yes... that's it.*

I was inspired by the devil.

I'm inspired by the devil!

Dialog from the film 'The Name Of The Rose'

In this chapter, the computational tools developed in previous chapters are applied to simulate the Constant-Volume Combustion Rotative Engine (MRCVC, for *Motor Rotativo de Combustión a Volumen Constante* [80]).

The main feature of the MRCVC engine is that the combustion could be performed at constant volume (effectively). To be more precise, when the chamber reaches its minimum volume and during a finite angular interval, the chamber changes its shape but not its volume. Furthermore, the combustion chamber has a ratio surface area/volume similar to those found in reciprocating engines [79]. Thus, a net increment on the engine thermodynamic efficiency could be reached when it is compared with both, the rotative Wankel engine and the classical engine with reciprocating pistons. The MRCVC engine has a perfect static and dynamic balance of its moving components, and hence, allowing to achieve high smoothness and low engine vibration. Also, the contact between apex seals and walls is harmonic, which should permits to reduce wear and noise.

6.1 Operation and geometry of MRCVC

The MRCVC was invented and patented by Jorge Toth [80], and it is under development at the Applied Mechanical Department of National University of Comahue (Neuquén, Argentina). In order to test the kinematic of the mechanism designed, at present, only a compressor with the geometry of the MRCVC was built [79]. This engine is composed by a rotor and two or more vanes inside a cylindrical housing. Figure 6.1 illustrates the geometry of a MRCVC with four vanes. The central region of the housing and the

vanes have oval shape, with apex seals to avoid gas leakage. The rotor is a ring with cylindrical hollows allowing the relative rotation movement of the vanes. Each vane must keep parallel its centerline with respect to the other vanes while their centers revolve around the output shaft. In this form, the vanes have translational movement only. This kinematic constraint is accomplished by means of a rim, which also links the engine shaft with the rotor and vanes. Breathing could be through ports in the side housings and/or through lateral ports in the center housing. However, the gas-exchange system is undefined yet and it is under development.

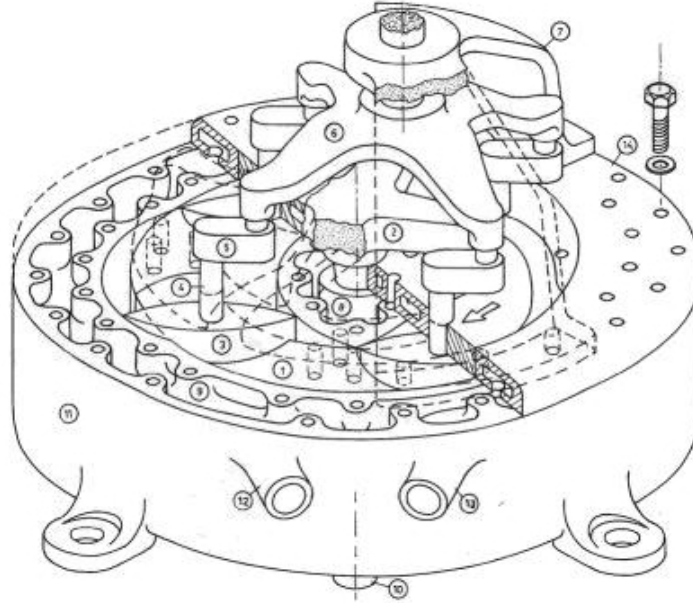


Figure 6.1: Cutaway drawing of four-vanes MRCVC engine.

Considering as null the radii of the apex seals, the geometry of MRCVC engine is completely defined by specifying the number of vanes (n), the radius of the trajectory center of the vanes (R), the half length of vane centerline (r), and the height of the chamber (h) [79]. Figure 6.2 shows a sketch of the top view of the MRCVC indicating the main geometric parameters. This geometric simplification is applied in this thesis to model the flow domain. Although a MRCVC with the simplest geometry is technologically unfeasible, the simplification turns simpler the analysis but retaining the main characteristics of the machine.

The rotation angle of engine shaft (θ) is measured clockwise relative to the trailing vane position when the chamber has its maximum volume at the start of compression ‘stroke’. Due to the symmetry of the MRCVC geometry, it is enough to analyze the angular interval

$$0 \leq \theta \leq \frac{n+2}{2n} \pi \quad (6.1)$$

which corresponds to a variation of the chamber volume from its maximum value to the minimum one. The geometry chamber for the remain part of the cycle is obtained by flipping the domain around the geometric symmetry axis. In order to get analytical formulae for describing the chamber geometry, the interval (6.1) is split in five sub-intervals due to topological changes and boundary redefinition. These intervals are defined as

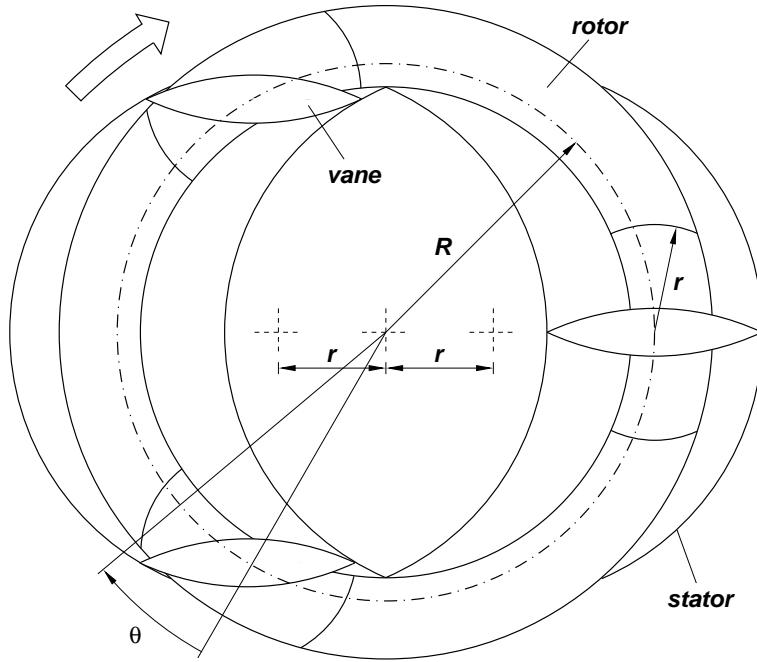


Figure 6.2: Basic geometry of a three-vane MRCVC and definition of its main geometric parameters.

- Interval 1:

$$0 \leq \theta < \frac{n-2}{2n}\pi - \phi$$

where $\phi = \arcsin\left(\frac{r}{R}\right)$.

- Interval 2:

$$\frac{n-2}{2n}\pi - \phi \leq \theta < \frac{n-2}{2n}\pi + \phi$$

- Interval 3:

$$\frac{n-2}{2n}\pi + \phi \leq \theta < \frac{n+2}{2n}\pi + \phi - 2\psi$$

where $\psi = \arctan\left(\frac{2r}{\sqrt{R^2 - r^2}}\right)$.

- Interval 4:

$$\frac{n+2}{2n}\pi + \phi - 2\psi \leq \theta < \frac{n+2}{2n}\pi - \phi$$

- Interval 5:

$$\frac{n+2}{2n}\pi - \phi \leq \theta \leq \frac{n+2}{2n}\pi$$

Figure 6.3 illustrates the position and shape of the flow domain for the angles at lower and higher limits of each sub-interval covering the interval (6.1).

Both, the surface area of the chamber walls (A) and the chamber volume (V) as a function of the shaft angle are necessary for 0D/1D simulations. On the other hand, the boundaries of the flow domain for the multi-D case were computed as a function of θ also.

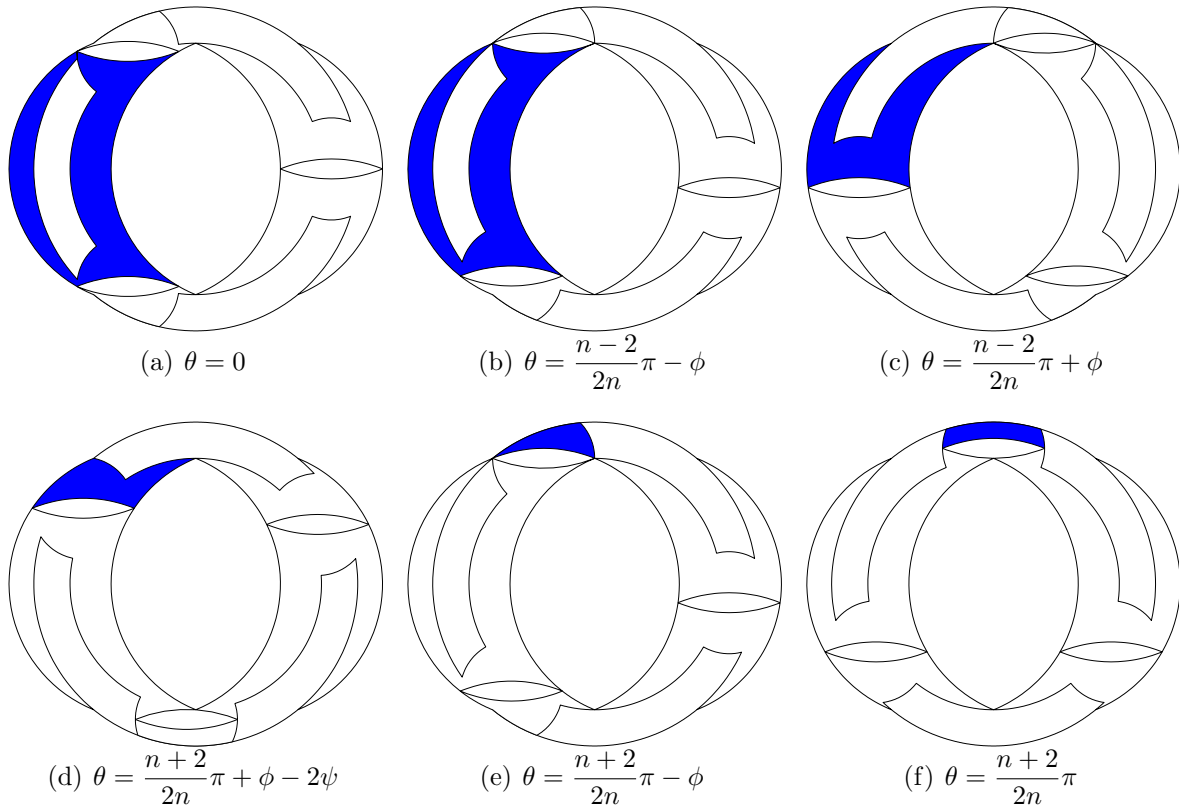


Figure 6.3: Changes of flow domain definition during a ‘stroke’ for a 3-vanes MRCVC engine.

The complete set of equations describing the cited geometric variables are presented in appendix C.

As mentioned above, the crank shaft rotates $(n+2)\pi/2n$ radians when the chamber evolves from its maximum volume to the minimum volume. Considering as a ‘stroke’ the evolution of the chamber volume between extrema values, then, the cycle duration is

$$\Delta\theta = 2\pi \left(1 + \frac{2}{n}\right) \quad (6.2)$$

in order to complete the four-stroke cycle. Note that the cycle duration depends on the number of vanes and, thus, contrasts with classical reciprocating and rotative engines, which works with two-stroke ($\Delta\theta = 2\pi$) or four-stroke ($\Delta\theta = 4\pi$) operating cycles.

There are $n+2$ operating chambers along a cycle, which produce a power ‘stroke’ per cycle. Therefore, n power pulses occur for each output shaft revolution.

6.2 Engine performance characteristics using 0D/1D models

In this section, the results obtained from the simulation of a MRCVC using a 0D/1D code are presented. Although there are not experimental results for this engine, the simulation has two purposes: firstly, to obtain a first estimate of the performance characteristics; and secondly, to test two type of port geometry.

The engine simulated has $n = 3$ vanes, with $R = 116.1$ mm and $r = h = 44.1$ mm. The maximum chamber volume is 500 cm^3 and the geometric compression ratio is 9:1. As discussed in the last section, the cycle duration for the geometric configuration used is 600° , with five operating chambers per cycle. The MRCVC considered is spark-ignited and naturally aspirated.

In order to simplify the computational model, the engine is simulated as five independent cylinders. Thus, ports could be modeled as poppet valves with appropriate passage area and discharge coefficients. But, with this approach, the wave interference between two contiguous chambers can not be modeled. The chamber wall temperature is adopted as 450 K and the combustion is modeled via a Wiebe function, being 267° the initial angle of combustion and 60° its duration. Table 6.1 contains the intake and exhaust manifolds data. The model is tested from 2000 rpm up to 6000 rpm. For this analysis, the reference

	Intake manifold	Exhaust manifold
Length	500 mm	400 mm
Diameter	50 mm	50 mm
Wall temperature	298 K	743 K

Table 6.1: Intake and exhaust manifolds data of MRCVC engine.

shaft angle (0°) is taken as the angle of the center vane labeled with 1 in figure 6.4.

6.2.1 Port design 1

In the first port design tested, the ports are placed in the central housing as sketched in figure 6.4. In this case, the intake and exhaust ports are equal. The minimum passage area for each port as a function of the shaft angle is plotted in figure 6.5. Ports timing are the following

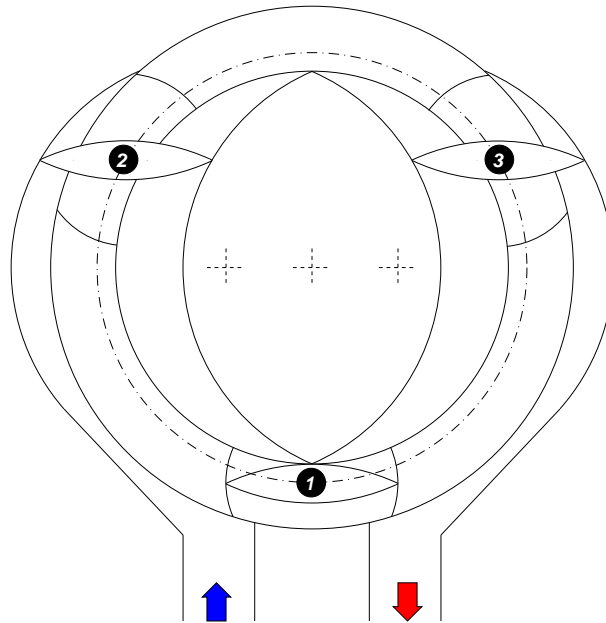


Figure 6.4: Model of port design 1.

- IPO: 595°.
- IPC: 170°.
- EPO: 430°.
- EPC: 5°.

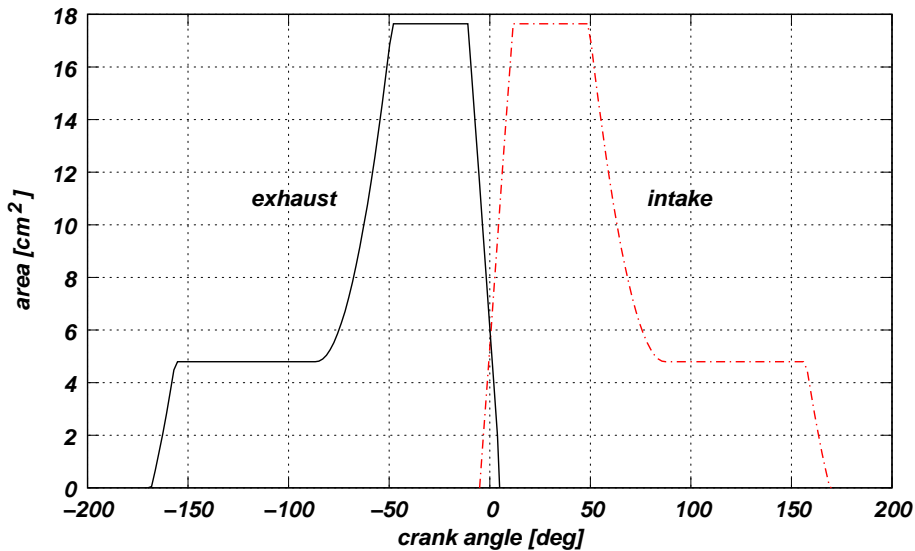


Figure 6.5: Gas-passage minimum area for intake and exhaust ports (design 1).

Figure 6.6 shows some results during a cycle at 4000 rpm, after reaching the stationary state. The pressure in a chamber, and in the intake and exhaust ports is plotted in sub-figures 6.6(a) and 6.6(b). The variation of mass air, fuel and burnt gases into the chamber is shown in figure 6.6(c). As could be noted, this port design allows good scavenge of the chamber. The mass flow rate through ports is presented in figure 6.6(d), where positive values represent inflow for the intake port and outflow for the exhaust port.

Performance curves are presented in section §6.2.3.

6.2.2 Port design 2

In this design, the ports are placed in the side housing. Figure 6.7 shows a sketch of the geometry. The gas-passage area for intake and exhaust ports as a function of the shaft angle is plotted in figure 6.8.

For the case tested, ports timing are

- IPO: 20°.
- IPC: 170°.
- EPO: 430°.
- EPC: 590°.

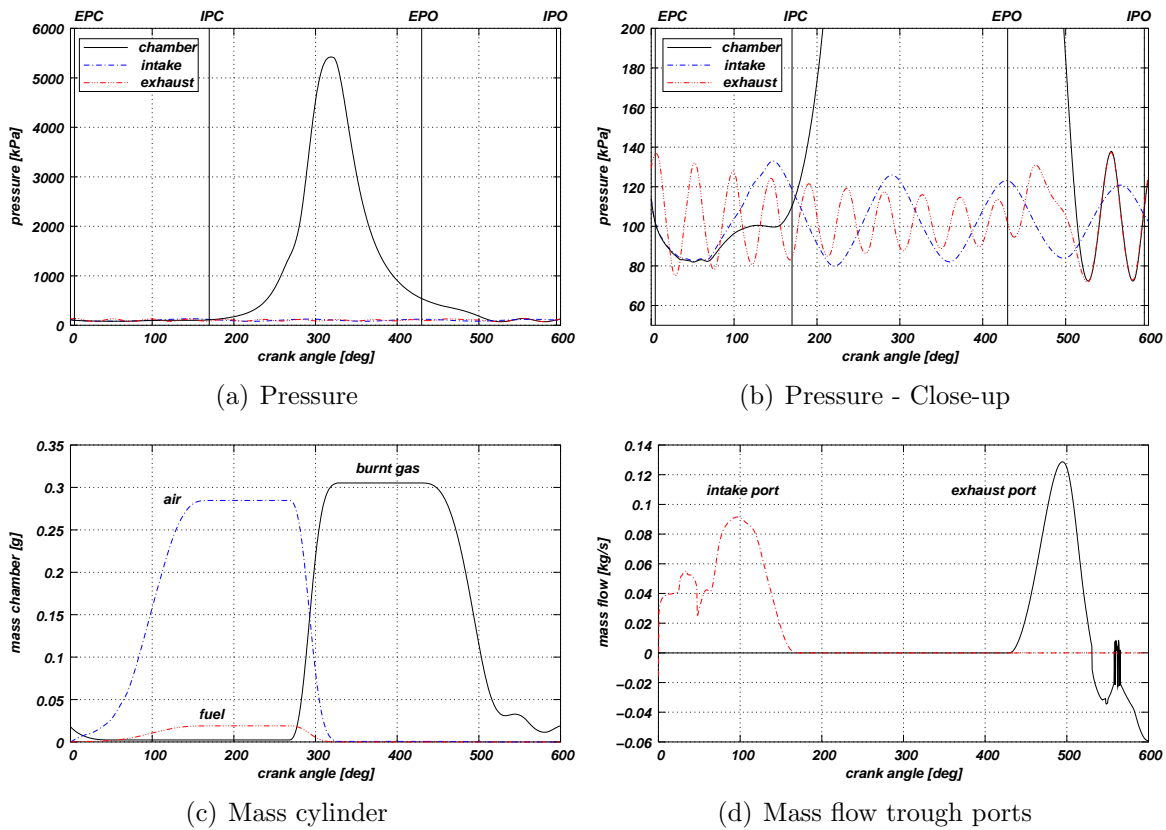


Figure 6.6: State in chamber, in intake port, in exhaust port, and mass flow rate through ports during a cycle at 4000 rpm for port design 1.

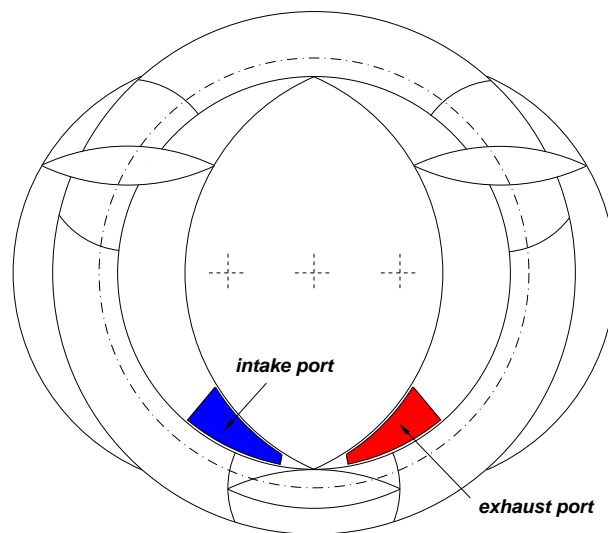


Figure 6.7: Model of port design 2.

Note that there is no possibility to overlap the final part of the exhaust process with the beginning of intake process. This fact could difficult the scavenging of the chamber. In addition, the gas passage area is limited by ports timing.

Figures 6.9(a) to 6.9(d) show the pressure in chamber and in intake and exhaust ports during a cycle, a close-up of these pressure curves, the mass into the chamber during a

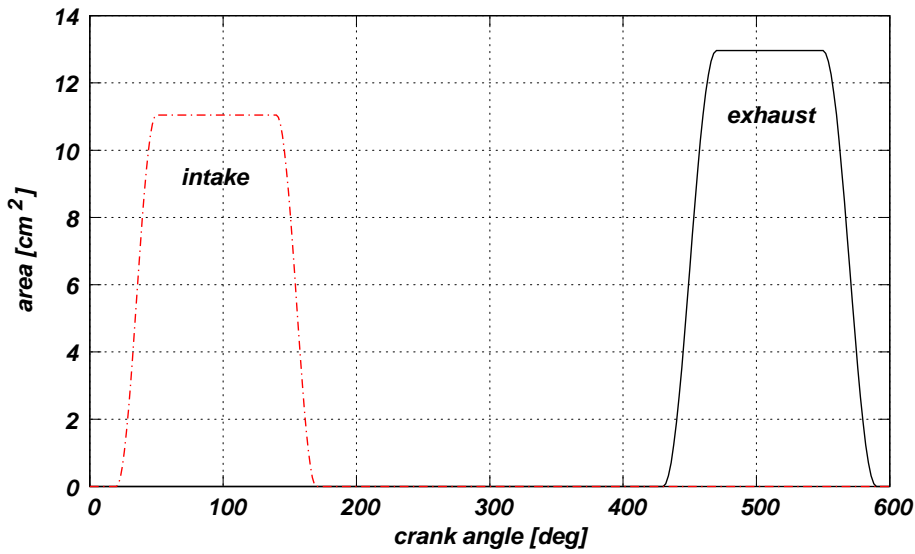


Figure 6.8: Gas-passage area for intake and exhaust ports (design 2).

cycle, and the mass flow rate through intake and exhaust ports respectively. These figures correspond to an engine velocity of 4000 rpm. In figure 6.9(c), a relatively high amount of residual gases remain into the chamber at EPO, which decreases the volumetric efficiency.

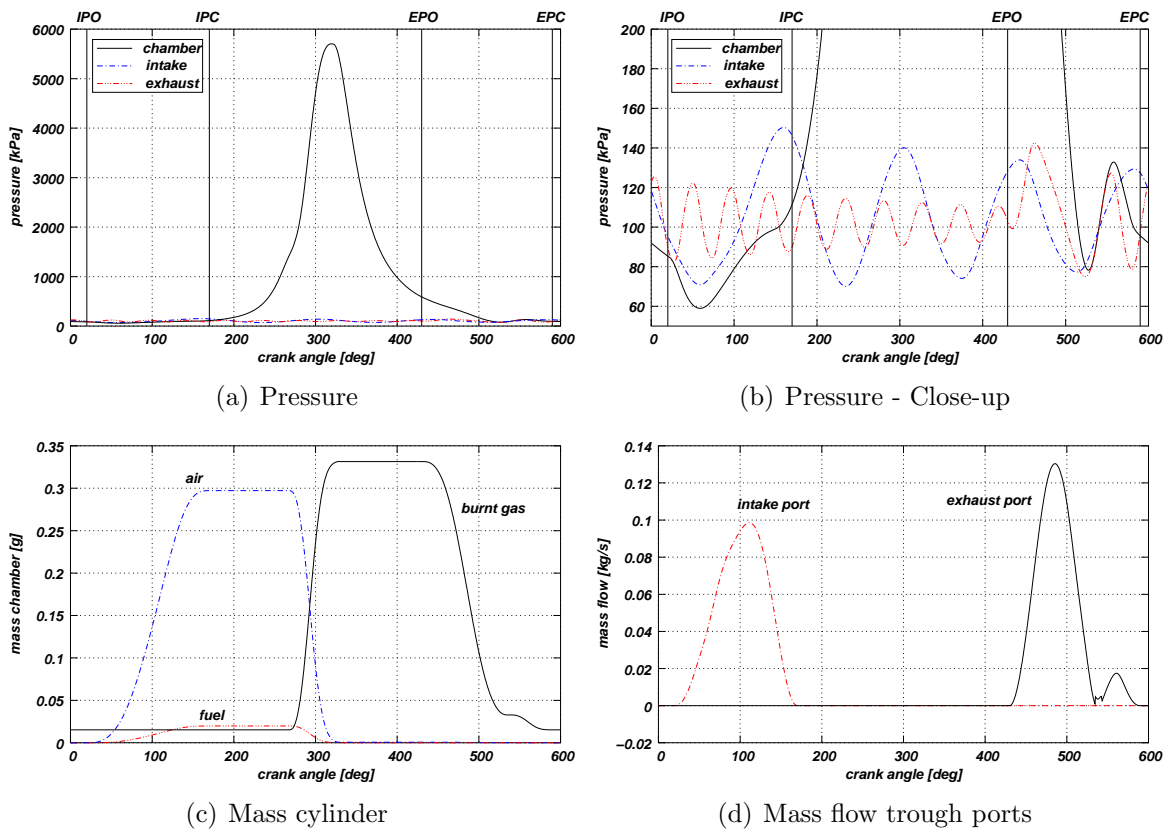


Figure 6.9: State in chamber, in intake port, in exhaust port, and mass flow rate through ports during a cycle at 4000 rpm for port design 2.

6.2.3 Comparison of port designs

In this section, the port designs presented above are compared via the performance characteristics. Indicated power and torque are plotted as a function of the engine velocity in figure 6.10. As could be noted, there are no significant difference between the two port designs. The reason of this may be that the negative effects introduced in the performance by the deficiencies of each design are approximately equal, tending to equalize their own outputs. Then, it is concluded that the best performance could be achieved using simultaneously the two designs in a proper way.

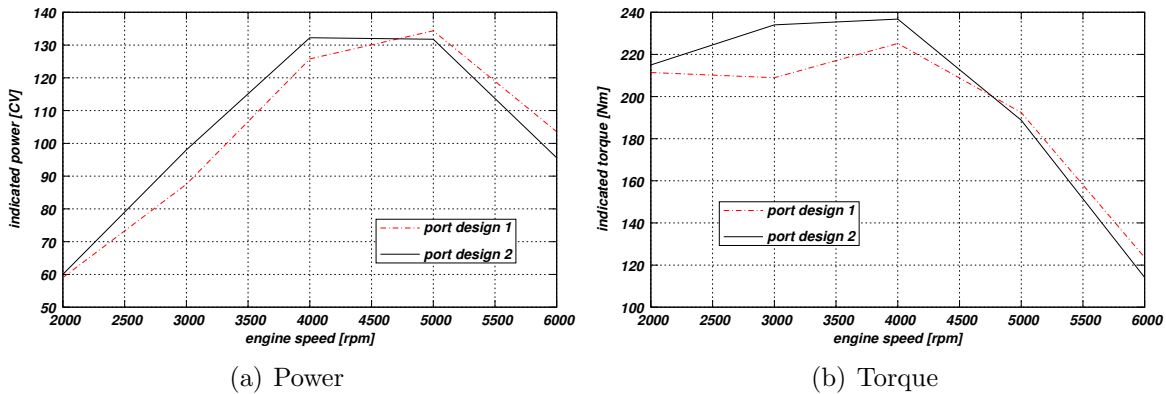


Figure 6.10: Performance characteristics for a MRCVC with two port designs.

6.3 Numerical simulation of fluid flow in the MRCVC engine

The fluid flow problem is solved using a two-dimensional approximation of the three-dimensional chamber. Due to the cylindrical shape of the chamber in the output shaft axis direction, this approximation could give a good representation of the flow in the middle plane of the chamber. The geometric data of the MRCVC considered are the same of the engine used in the simulation whose results were presented in section §6.2.

In the following sections results of the CMD and CFD problems are presented and discussed.

6.3.1 Computational mesh dynamic problem

As noted in chapter 2, the numerical simulation of in-cylinder flows in internal combustion engines has the advantage that the CMD problem could be solved *a priori*, *i.e.* before to solve the fluid problem. The displacement of solid walls is imposed and their movement is not affected by the flow field. In addition, the periodicity of the movement allows to compute the mesh only for one cycle. In the particular case of the MRCVC engine, the mesh could be generated for a ‘stroke’ and, then, to flip it around vertical and horizontal symmetry axis.

As it is shown in figure 6.3, the flow domain changes either their topology or their boundary definition four times during a ‘stroke’. Then, at least five remeshing stages

must be introduced per ‘stroke’. More remeshing stages could be necessary in order to keep a good mesh quality.

The meshes generated have triangular elements with $h = 0.2$ mm on the boundaries and $h = 0.5$ mm in the interior region of the domain. Two sequences of meshes were generated in order to compare the mesh quality obtained. In the first sequence, a mesh were generated for each region (sub-interval) at the angle representing the middle of the interval where the region is valid. In the second sequence, the intervals 2, 3 and 4 were subdivided into two sub-intervals with the same length. In both cases, the mesh movement was computed using prescribed displacements of the boundary nodes. Table 6.2 contains the number of elements and nodes in each mesh generated for both sequences.

	First sequence		Second sequence	
	Elements	Nodes	Elements	Nodes
Interval 1	111980	58156	111980	58156
Interval 2	94496	48711	103215	53567
Interval 3	47054	24358	80880	41986
Interval 4	17047	9056	58014	30174
Interval 5	13083	6888	36645	19145
			21662	11422
			14205	7575
			13083	6888

Table 6.2: Data of meshes generated.

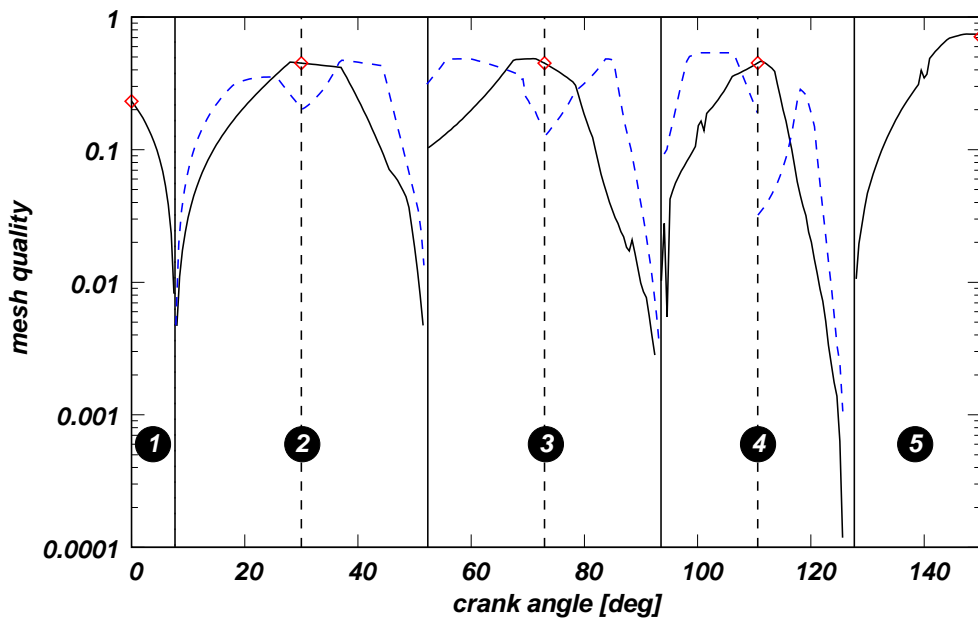


Figure 6.11: Mesh quality as a function of the rotation angle.

Figure 6.11 shows (in logarithmic scale) the mesh quality as a function of the rotation angle of the output shaft. The element quality is measured using the expression (2.11). In the figure, the black solid line corresponds to the first sequence of meshes and the blue dashed line to the second one. As expected, the mesh quality deteriorates sharply

at the angles that limit two consecutive sub-intervals. This occurs since some circular arcs belonging to the domain boundary appear and/or disappear from an interval to the following one. Then, the area of elements attached to these boundary arcs tends to zero and, thus, the mesh quality goes to zero also.

In figures 6.12 to 6.16 the distribution of the element quality q over the whole mesh at several crank angles is presented. Each figure contains the element quality field at the initial and final angles of the intervals in which the ‘stroke’ is sub-divided. These distributions correspond to the second sequence of meshes generated.

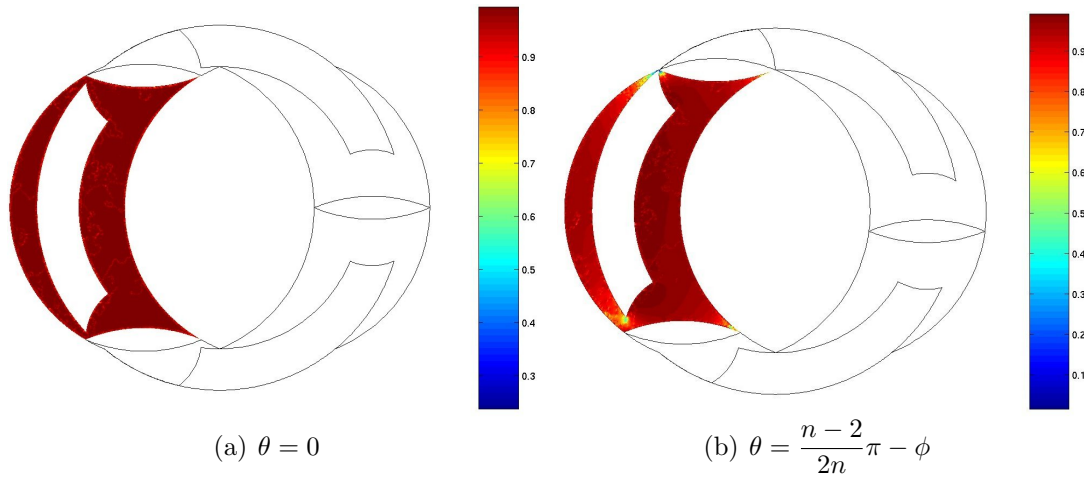


Figure 6.12: Element quality field at the initial and final angles of interval 1.

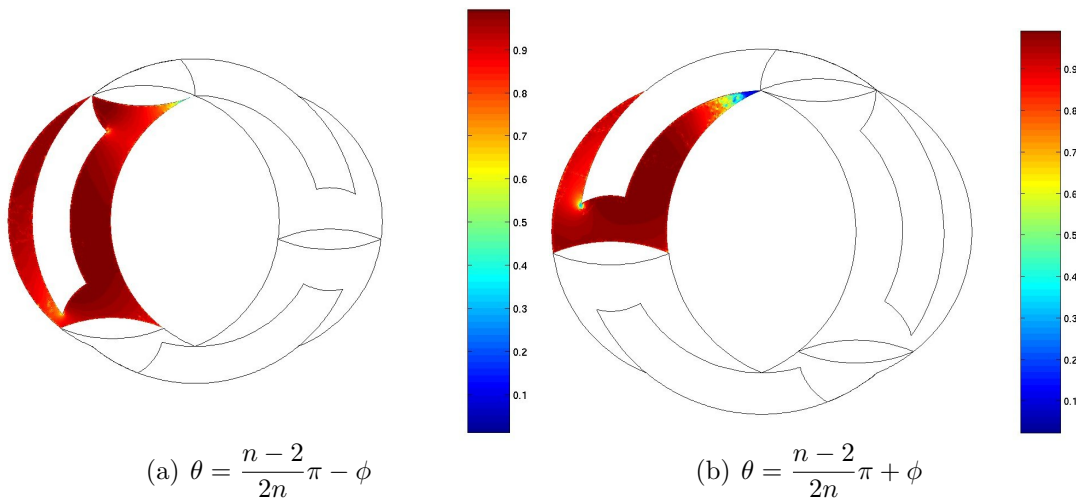


Figure 6.13: Element quality field at the initial and final angles of interval 2.

6.3.2 Computational fluid dynamic problem

In order to avoid the geometric difficulties associated with the port design 1 presented in section §6.2 and since the port design 2 can not be simulated in a 2D model, the intake and exhaust ports were placed in the internal region of the stator as sketched in figure 6.17.

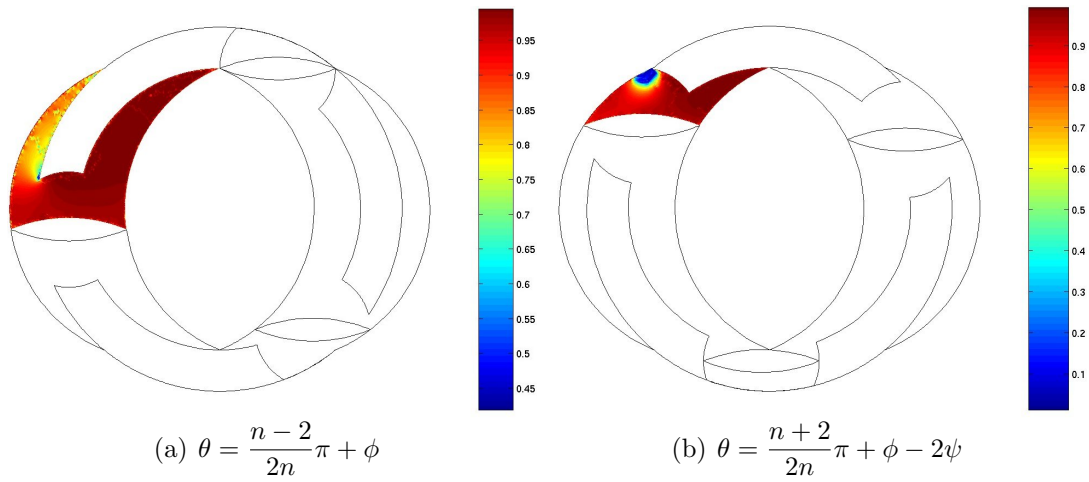


Figure 6.14: Element quality field at the initial and final angles of interval 3.

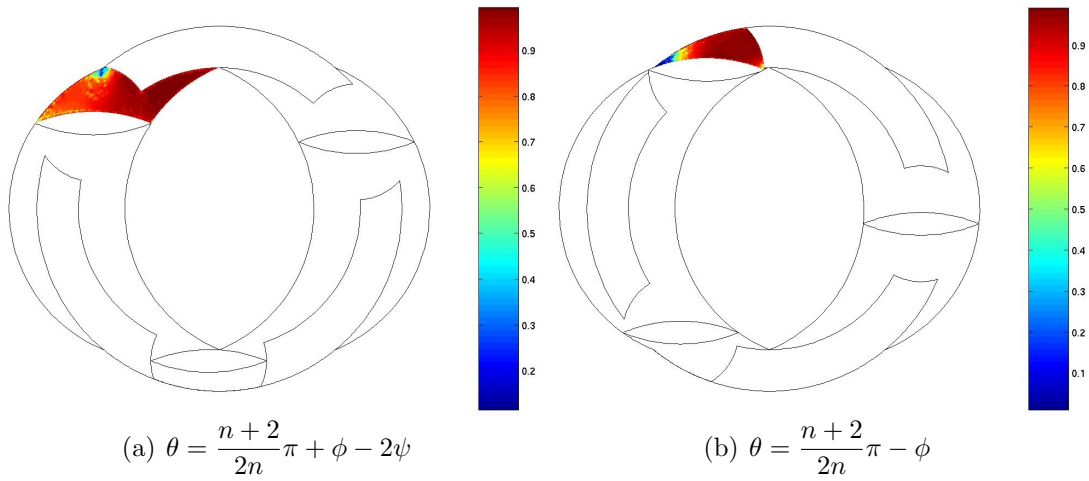


Figure 6.15: Element quality field at the initial and final angles of interval 4.

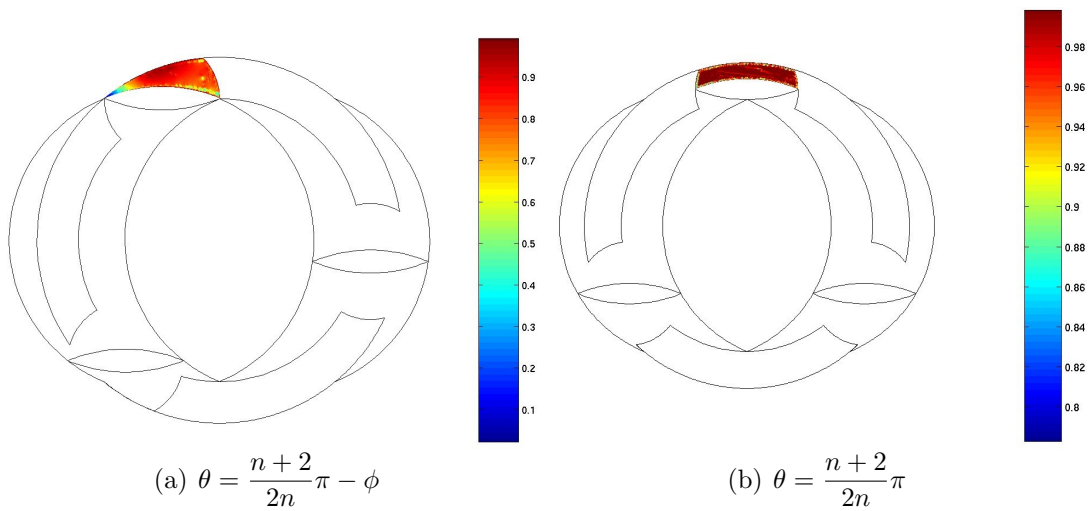


Figure 6.16: Element quality field at the initial and final angles of interval 5.

The size and position of ports are determined from the port design 2 used in section §6.2. Cold conditions (without combustion) are assumed for the computation.

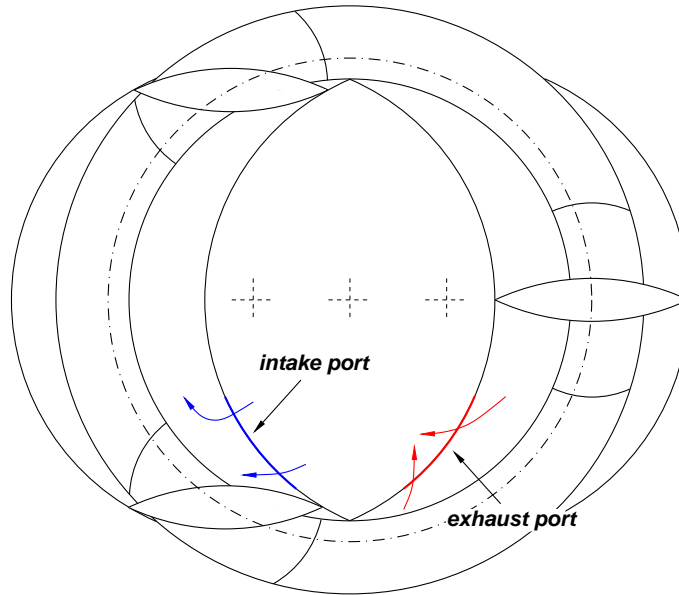


Figure 6.17: Intake and exhaust port geometries for the two-dimensional model.

The local preconditioning strategy presented in chapter 3 is applied to solve the problem. No-slip boundary condition is imposed on the solid walls, which are assumed to be insulated. Intake and exhaust ports are modeled using the combination of wall and absorbing boundary conditions presented in section §1.2.3 with restrictions imposed via Lagrange multipliers. Reference states for absorbing boundary conditions are assumed constant in time. For the intake port, this reference state is $\mathbf{U}_{\text{ref}}^i = [1.3937 \text{ kg/m}^3, 0 \text{ m/s}, 0 \text{ m/s}, 120 \text{ kPa}]^T$, and for the exhaust port is $\mathbf{U}_{\text{ref}}^e = [0.5575 \text{ kg/m}^3, 0 \text{ m/s}, 0 \text{ m/s}, 80 \text{ kPa}]^T$. Initially, the fluid is at rest with constant pressure (100 kPa) and temperature (300 K). The gas has constant properties along the whole cycle with the following values: $R = 287 \text{ J/kg K}$, $\mu = 1 \times 10^{-5} \text{ Pa}\cdot\text{s}$, $\kappa = 0.01415 \text{ W/m K}$, and $\gamma = 1.4$. The rotation speed of the engine shaft is 3000 rpm and the time step is calculated with a target CFL_u number of 10. The minimum $\Delta\theta$ permitted is 0.25 CAD and the maximum is 3 CAD.

At each remeshing stage, the solution is projected from the ‘old’ mesh to the new one by using the interpolation functions of the FEM method. This projection is non-conservative, nevertheless it was used because that subject is out from the scope of this thesis and will be improved in the future.

The problem is solved until the solution is approximately stationary (cyclic), which is reached in three cycles. The results obtained from the last cycle simulated are presented in the following figures. Figures 6.18 to 6.23 show the magnitude of the flow velocity. Although only a chamber was simulated (the ‘reference’ chamber), the remaining four chambers are represented in the figures by repetition via a rotation transformation of an integral multiple of $2\pi/n$.

Due to the large difference of the pressure along the whole cycle, the pressure field is presented in non-dimensional form and with a logarithmic scale. Figures 6.24 to 6.29 present the quantity $\log(p/p_{\text{ref}})$, where $p_{\text{ref}} = 100 \text{ kPa}$ was adopted.

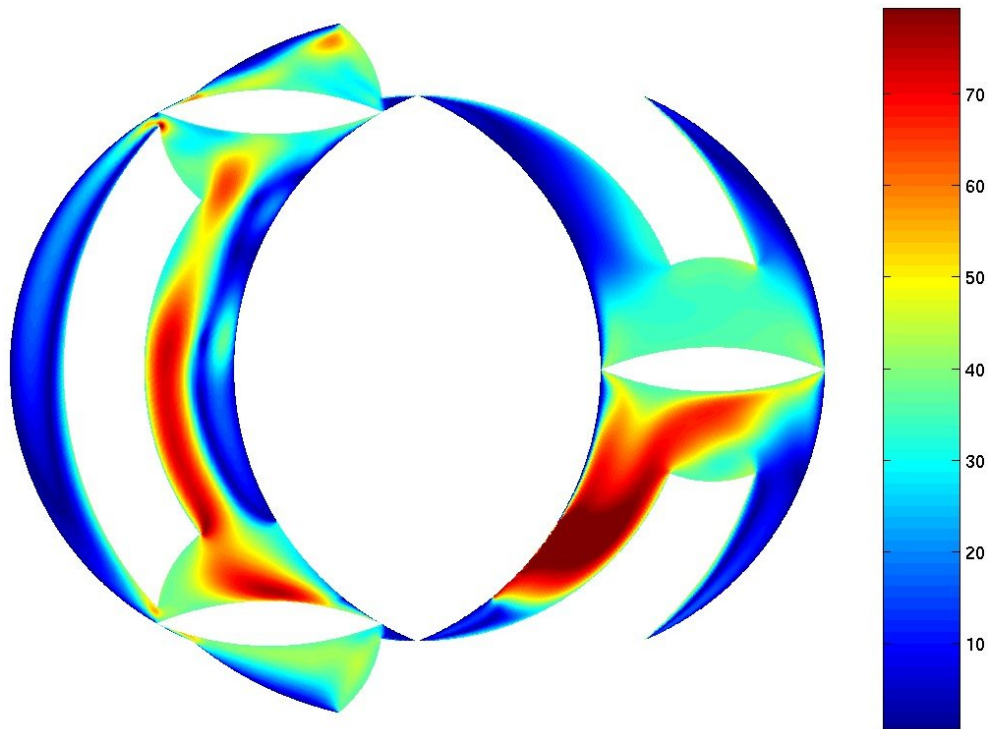


Figure 6.18: Magnitude of the flow velocity ([m/s]) in the chambers of the MRCVC with reference chamber at $\theta = 0^\circ$.

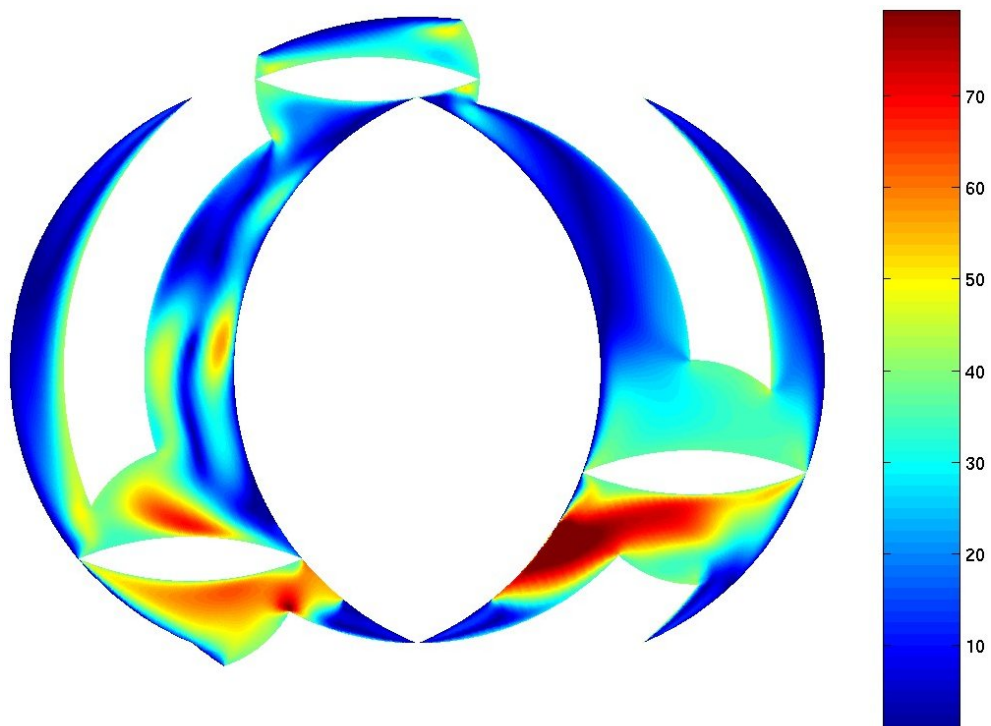


Figure 6.19: Magnitude of the flow velocity ([m/s]) in the chambers of the MRCVC with the reference chamber at $\theta = 20^\circ$.

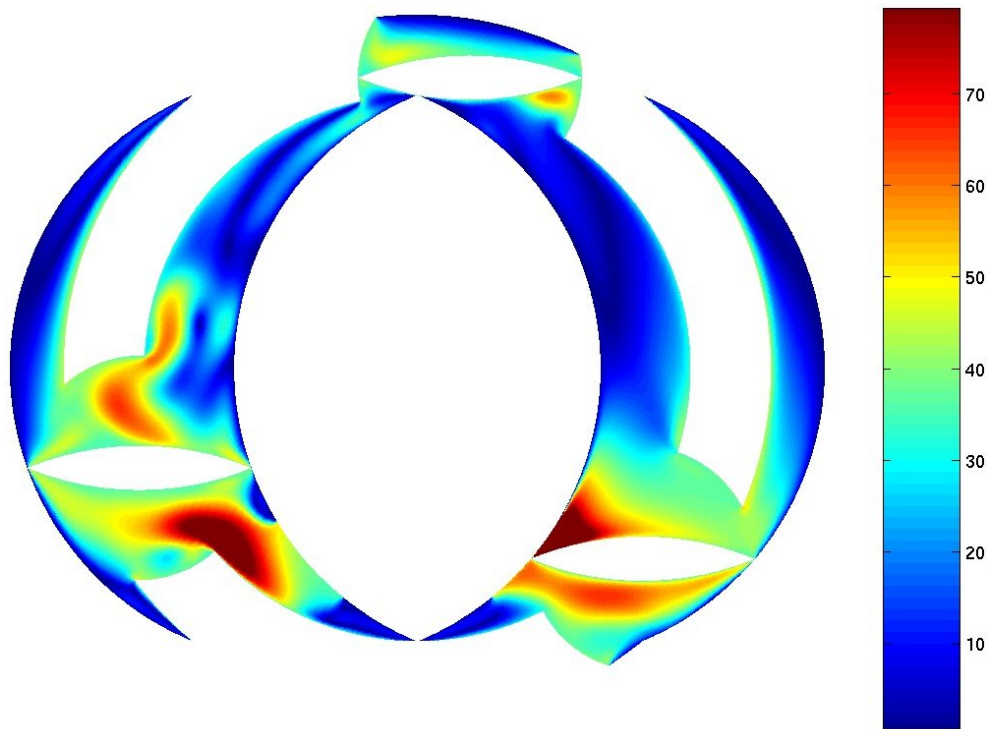


Figure 6.20: Magnitude of the flow velocity ([m/s]) in the chambers of the MRCVC with the reference chamber at $\theta = 40^\circ$.

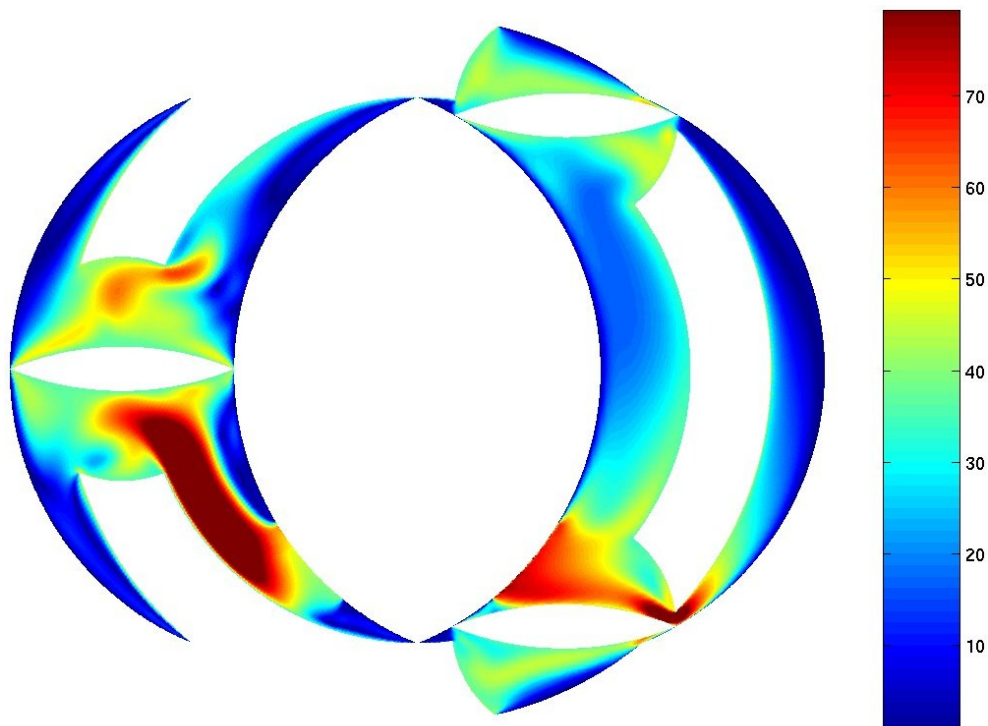


Figure 6.21: Magnitude of the flow velocity ([m/s]) in the chambers of the MRCVC with the reference chamber at $\theta = 60^\circ$.

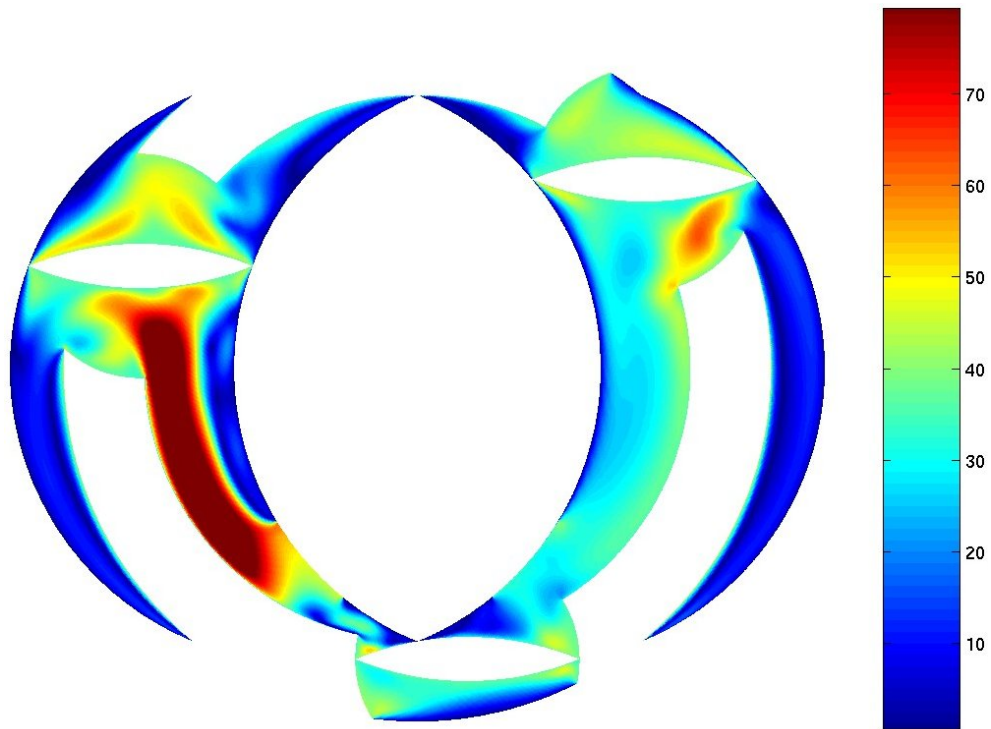


Figure 6.22: Magnitude of the flow velocity ([m/s]) in the chambers of the MRCVC with the reference chamber at $\theta = 80^\circ$.

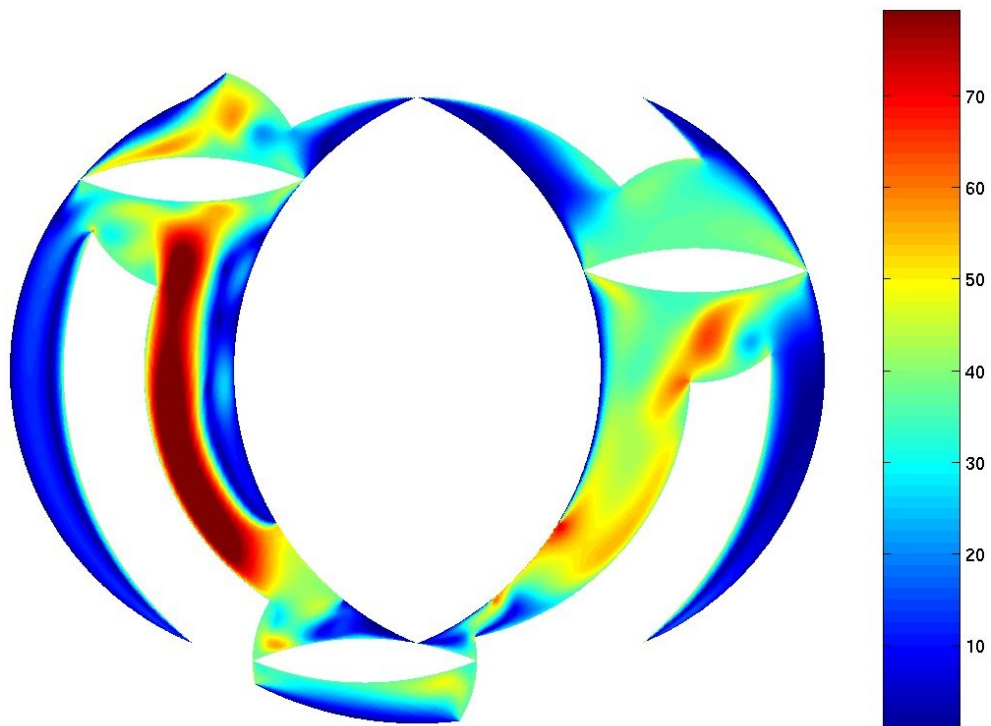


Figure 6.23: Magnitude of the flow velocity ([m/s]) in the chambers of the MRCVC with the reference chamber at $\theta = 100^\circ$.

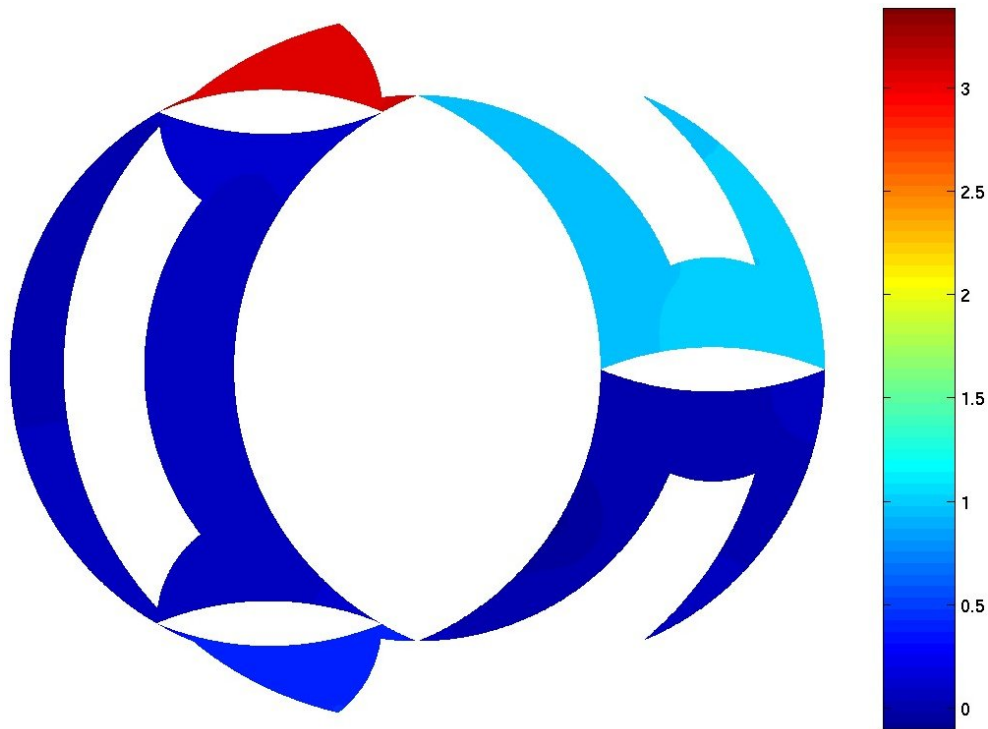


Figure 6.24: Non-dimensional logarithmic pressure field in the chambers of the MRCVC with the reference chamber at $\theta = 0^\circ$.

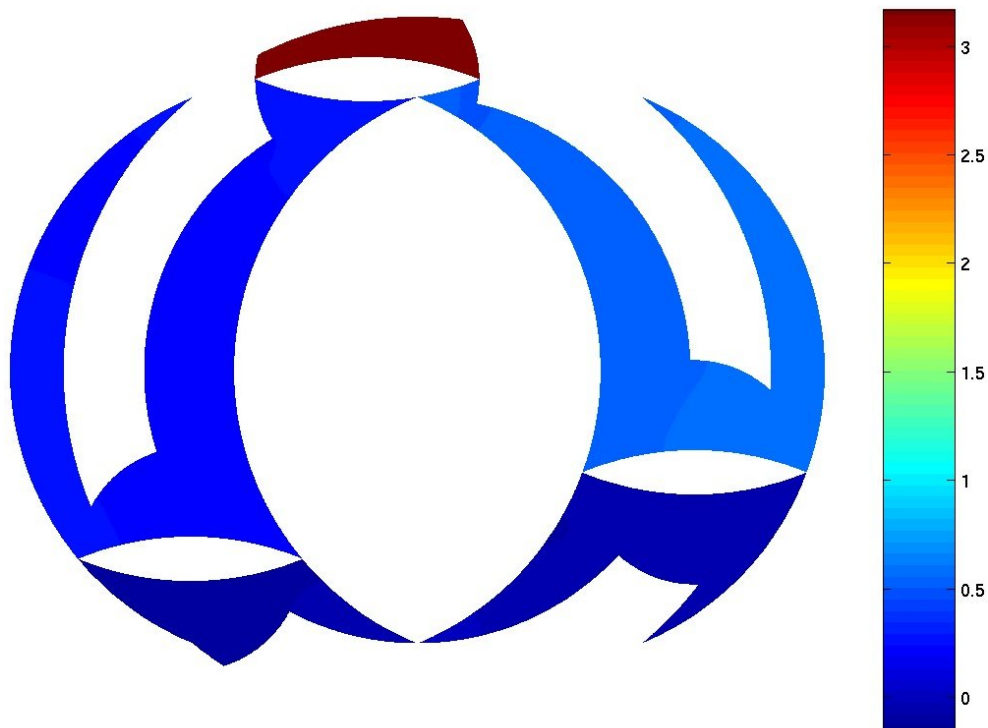


Figure 6.25: Non-dimensional logarithmic pressure field in the chambers of the MRCVC with the reference chamber at $\theta = 20^\circ$.

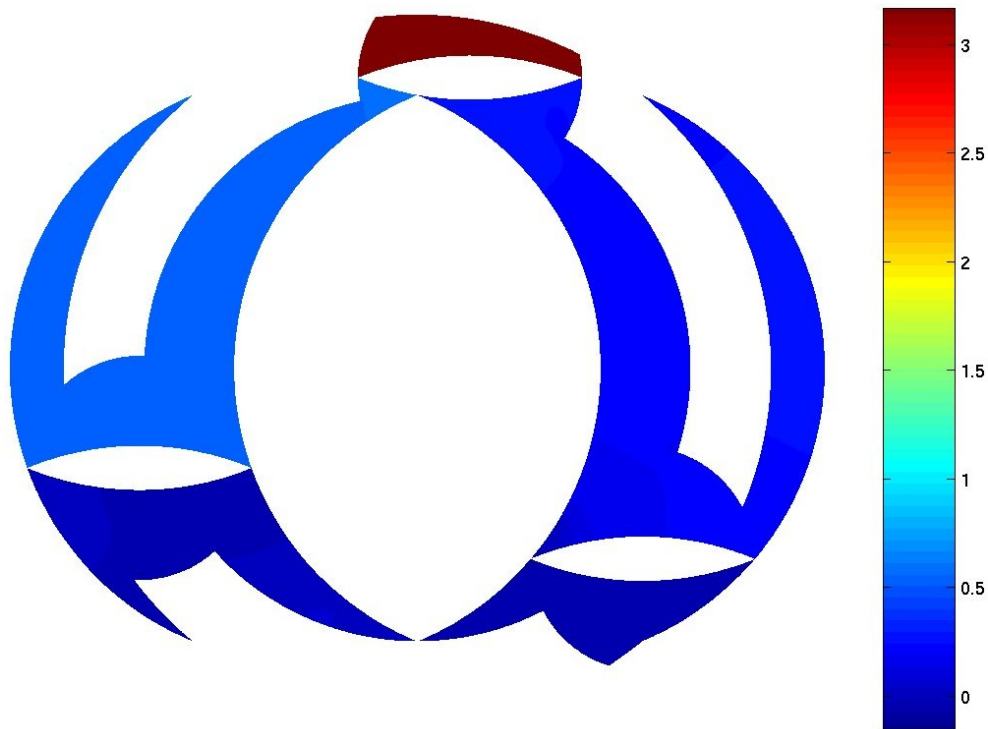


Figure 6.26: Non-dimensional logarithmic pressure field in the chambers of the MRCVC with the reference chamber at $\theta = 40^\circ$.

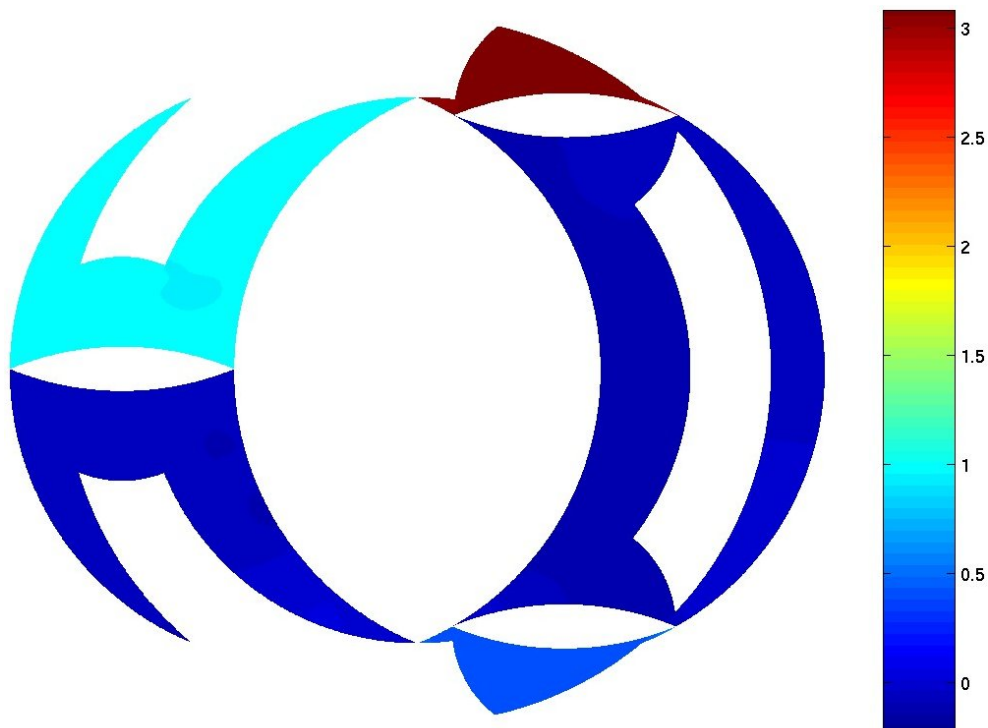


Figure 6.27: Non-dimensional logarithmic pressure field in the chambers of the MRCVC with the reference chamber at $\theta = 60^\circ$.

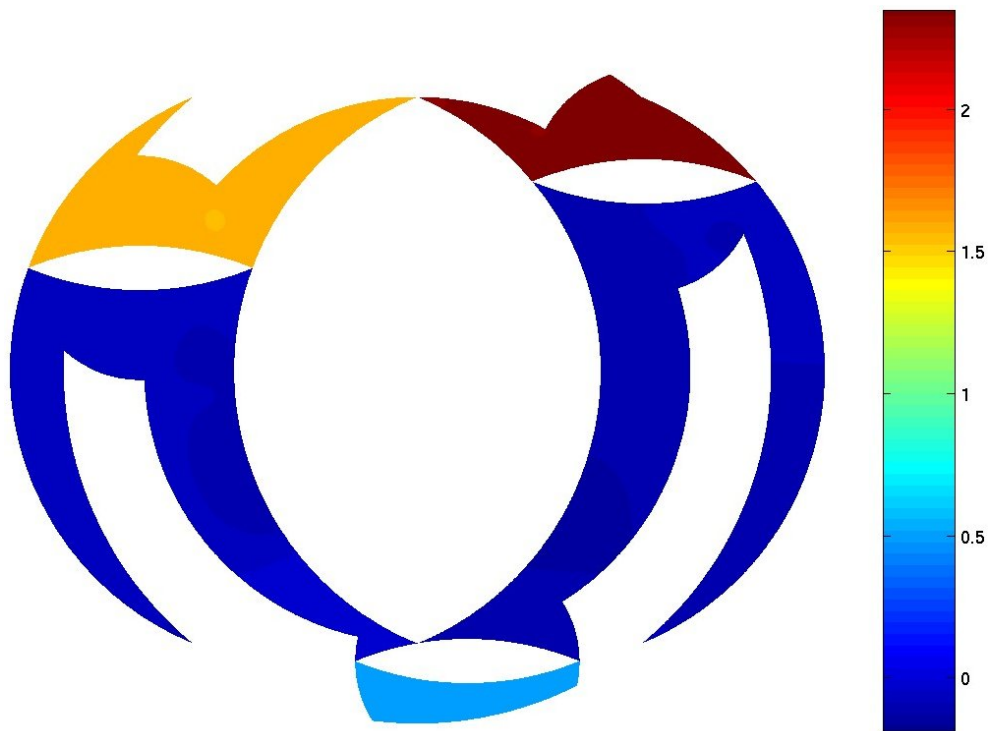


Figure 6.28: Non-dimensional logarithmic pressure field in the chambers of the MRCVC with the reference chamber at $\theta = 80^\circ$.

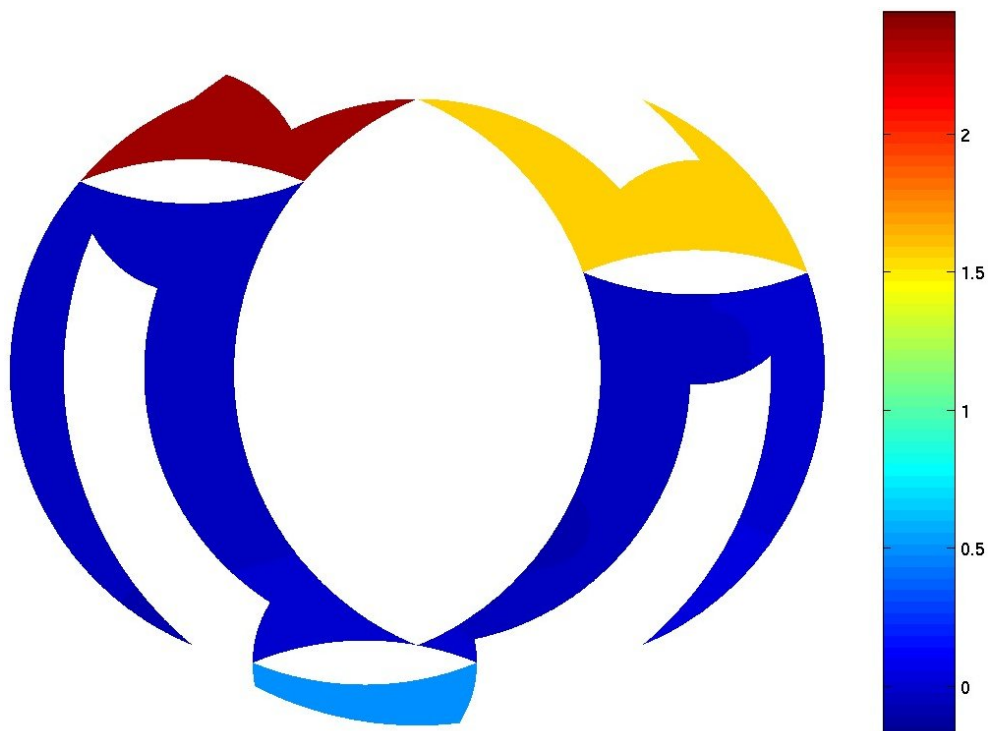


Figure 6.29: Non-dimensional logarithmic pressure field in the chambers of the MRCVC with the reference chamber at $\theta = 100^\circ$.

Chapter 7

Conclusions and future work

*Time of departure
From the depths of despair
I seek no paradise, though the end draws near.
It is an endless overture of my own reconstructions
I seek no paradise, just desire the salvation*
Limbonic Art

The main goal of this thesis was the proposition, description and testing of some computational tools to solve in-chamber flows in IC engines. This type of problems involve several aspects to be solved, from which were addressed the mesh dynamics problem, the resolution of compressible flows at the low-Mach number limit, and the coupling between models with different level of approximation, particularly 1D/multi-D coupling for compressible flow.

Regarding the mesh dynamic problem, an optimization-based simultaneous mesh untangling and smoothing strategy was proposed. The functional adopted here, based on a mesh quality indicator, showed to be very robust to follow severe boundary deformations including in IC engine problems. Generally, in FSI problems the time step size is restricted by one of the two physical problems, the structural dynamic problem or the fluid dynamic one. Being the mesh dynamic an auxiliary problem, it is expected that it will not be more restrictive than any of the other two. However, in several applications the refinement imposes the reduction of the time step size due to the mesh dynamics in order to avoid the element inversion. The enhancement of the CMD with simultaneous untangling and smoothing circumvents this drawback. A global solver is very attractive to make this procedure less user-dependent. In addition, this proposed technique was successfully applied to conformal mesh generation in 2D domains (see appendix B).

One of the most popular strategies to solve flow problems at the low-Mach number limit, the preconditioning of the governing equations, was adopted in the thesis. Since the Mach number of in-cylinder flows in IC engines could range from very low values to transonic values, the method of preconditioning is the most appropriate. This technique was applied in conjunction with the dual time stepping strategy in order to solve transient problems. The preconditioning matrix used was originally proposed by Choi

and Merkle [16], and designed for steady compressible flows. Some ‘free’ parameters of that matrix were redefined for transient flows analyzing the eigenvalues of the system. A stabilized finite element formulation was derived from the original work formulated in terms of a finite volume method. This formulation was tuned having into account the modified wave propagation introduced by the preconditioned mass matrix for unsteady problems. The resultant method was compared with solutions of Navier-Stokes equations for incompressible flows with very good results. Furthermore, it was shown that, under the same conditions (mesh element size, time step, etc.), the preconditioning strategy gives better results than the non-preconditioned system. In addition, the strategy was tested in IC engine problems under cold conditions.

A 1D/0D code for computational simulation of IC engines was developed. The mathematical models and the numerical methods used were described briefly. Some solution strategies were reformulated to use an implicit scheme of integration in time. The code was written in the language `Python` in order to take advantage of the object-oriented programming and the possibility of integration with other languages.

Besides, two simple approaches were studied to solve the 1D/multi-D domain coupling for compressible flows. One of such approaches consists in the use of absorbing boundary conditions at the coupling interface. In the proposed method, the constraints were applied via penalization. The technique is attractive from the theoretical point of view, but its rate of convergence and robustness must be improved in order to be applied to practical problems. The other coupling method proposed is based on constraints of the state at the coupling interface. This method could be useful when an implicit time scheme is applied and the problem is solved in a ‘monolithic’ way. In the 1D/1D case, the non-splitting system is recovered. The method was successfully applied to solve 2D/1D and 3D/1D coupling problems. The junction 3-to-1 of the exhaust manifold of a six-cylinder IC engine was simulated using the 3D/1D coupling strategy. The solution was compared with the results obtained from the 1D/0D code and a verification of the hypothesis in the pipe junction 0D model proposed by Corberan was also carried out.

Furthermore, simulations of the rotative IC engine MRCVC were performed. Performance characteristic curves were estimated for two different port designs applying 1D/0D models. Also, a two dimensional CFD simulation in cold conditions was performed.

A great amount of work remains to be done on the improvement of the computational tools for the simulation of IC engines. For instance, the effective coupling of the 1D/0D code and the CFD-3D code is proposed as a future work.

The thermodynamic models of valve and pipe junction use the compatibility equations along characteristic lines in order to close the system of equations. The integration along these curves is done explicitly in the code developed. Thus, a full implicit model for such devices should be proposed. On the other hand, CFD multi-D computations could be used for verification and/or improvement of the hypothesis involved in the 0D models applied in the description of IC engine operating characteristics.

In addition, the computational tools developed in this thesis should be massively applied to the simulation of in-cylinder flows in 3D geometries of IC engine, in particular the MRCVC. These simulations should incorporate the modeling of the combustion process, spray dynamics, mixture formation, more accurate boundary conditions (*e.g.* a law of the wall for the temperature to take into account turbulent effects), etc.

Appendix A

A finite element method for incompressible flows

A.1 Navier-Stokes equations for incompressible flows

Let $\Omega_t \subset \mathbb{R}^{n_d}$ the spatial domain and $(0, t_f)$ the temporal domain, and let Γ_t denote the boundary of Ω_t .

The Navier-Stokes equations governing an incompressible flow using an ALE description are

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \quad \text{on } \Omega_t \times (0, t_f) \\ \rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} - \mathbf{w}) \cdot \nabla \mathbf{u} \right] - \nabla \cdot \boldsymbol{\sigma} &= \mathbf{0} \quad \text{on } \Omega_t \times (0, t_f) \end{aligned} \quad (\text{A.1})$$

with $\boldsymbol{\sigma}$ the stress tensor, given by

$$\begin{aligned} \boldsymbol{\sigma} &= -p\mathbf{I} + 2\mu_{\text{eff}}\boldsymbol{\epsilon}(\mathbf{u}) \\ \boldsymbol{\epsilon}(\mathbf{u}) &= \frac{1}{2}((\nabla \mathbf{u}) + (\nabla \mathbf{u})^T) \end{aligned} \quad (\text{A.2})$$

where $\mu_{\text{eff}} = \rho\nu_{\text{eff}}$ is the effective dynamic viscosity proportional to the effective kinematic viscosity (ν_{eff}) defined below and $\boldsymbol{\epsilon}$ is the strain rate tensor. The effective kinematic viscosity is computed as the sum of the molecular (ν) and the turbulent (ν_t) kinematic viscosities: $\nu_{\text{eff}} = \nu + \nu_t$. In order to compute the turbulent kinematic viscosity, the eddy viscosity model of Smagorinsky [69] for the LES strategy is used

$$\nu_t = (C_s h)^2 \Delta \sqrt{2\boldsymbol{\epsilon}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{u})} \quad (\text{A.3})$$

A.1.1 Boundary conditions

The continuum formulation is completed by the initial and boundary conditions. Relative to the boundary conditions the whole boundary may be split in parts, imposing on each part a Dirichlet condition, a Neumann condition or a near wall boundary condition. For

the velocity field these boundary conditions are

$$\begin{aligned} \mathbf{u} &= \mathbf{g} && \text{on } \Gamma_t^g \\ \mathbf{n} \cdot \boldsymbol{\sigma} &= \mathbf{h} && \text{on } \Gamma_t^h \\ \mathbf{n} \cdot \boldsymbol{\sigma} &= \mathbf{h}_{\text{wall}}(u_*(\mathbf{u})) && \text{on } \Gamma_t^{\text{wall}} \end{aligned} \quad (\text{A.4})$$

where

$$\begin{aligned} \Gamma_t &= \Gamma_t^g \cup \Gamma_t^h \cup \Gamma_t^{\text{wall}} \\ \Gamma_t^g \cap \Gamma_t^h \cap \Gamma_t^{\text{wall}} &= \emptyset \end{aligned} \quad (\text{A.5})$$

If $\Gamma_t^h = \emptyset$ and $\Gamma_t^{\text{wall}} = \emptyset$, the pressure should be fixed at a reference value in at least one node in order to remove the corresponding rigid mode in the numerical computation.

A.2 Finite element formulation

The functional spaces for the weight and interpolation functions are defined as follows

$$\begin{aligned} \mathcal{S}_{\mathbf{u}}^h &= \{\mathbf{u}^h | \mathbf{u}^h \in (H^{1h})^{n_d}, \mathbf{u}^h = \mathbf{g}^h \text{ on } \Gamma_t^g\} \\ \mathcal{V}_{\mathbf{u}}^h &= \{\mathbf{W}^h | \mathbf{W}^h \in (H^{1h})^{n_d}, \mathbf{W}^h = \mathbf{0} \text{ on } \Gamma_t^g\} \\ \mathcal{S}_p^h &= \{q^h | q^h \in H^{1h}\} \\ \mathcal{V}_p^h &= \{p^h | p^h \in H^{1h}\} \end{aligned} \quad (\text{A.6})$$

The SUPG-PSPG formulation of equations (A.1) is written as [78]
Find $\mathbf{u}^h \in \mathcal{S}_{\mathbf{u}}^h$ and $p^h \in \mathcal{S}_p^h$ such that $\forall \mathbf{W}^h \in \mathcal{V}_{\mathbf{u}}^h, \forall q^h \in \mathcal{V}_p^h$

$$\begin{aligned} &\int_{\Omega_t} \mathbf{W}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + (\mathbf{u}^h - \mathbf{w}^h) \cdot \nabla \mathbf{u}^h \right) + \int_{\Omega_t} \boldsymbol{\epsilon}(\mathbf{W}^h) : \boldsymbol{\sigma}^h d\Omega + \\ &+ \sum_{e=1}^{n_{\text{el}}} \int_{\Omega_t} \boldsymbol{\delta}^h \cdot \left[\rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + (\mathbf{u}^h - \mathbf{w}^h) \cdot \nabla \mathbf{u}^h \right) - \nabla \cdot \boldsymbol{\sigma}^h \right] d\Omega + \\ &+ \sum_{e=1}^{n_{\text{el}}} \int_{\Omega_t} \boldsymbol{\epsilon}^h \cdot \left[\rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + (\mathbf{u}^h - \mathbf{w}^h) \cdot \nabla \mathbf{u}^h \right) - \nabla \cdot \boldsymbol{\sigma}^h \right] d\Omega + \\ &+ \sum_{e=1}^{n_{\text{el}}} \int_{\Omega_t} \nu_{\text{LSIC}} \nabla \cdot \mathbf{W}^h \rho \nabla \cdot \mathbf{u}^h d\Omega + \int_{\Omega_t} q^h \nabla \cdot \mathbf{u}^h d\Omega = \int_{\Gamma_t^h} \mathbf{W}^h \cdot \mathbf{h}^h d\Gamma \end{aligned} \quad (\text{A.7})$$

The stabilization parameters are defined as

$$\begin{aligned} \boldsymbol{\delta}^h &= \tau_{\text{SUPG}} ((\mathbf{u}^h - \mathbf{w}^h) \cdot \nabla) \mathbf{N}^h \\ \boldsymbol{\epsilon}^h &= \tau_{\text{PSPG}} \frac{1}{\rho} \nabla q^h \\ \tau_{\text{SUPG}} &= \frac{h}{2 \|\mathbf{u}^h - \mathbf{w}^h\|} z(Re_u) \\ \tau_{\text{PSPG}} &= \tau_{\text{SUPG}} \\ \nu_{\text{LSIC}} &= \frac{h}{2} \|\mathbf{u}^h - \mathbf{w}^h\| z(Re_u) \end{aligned} \quad (\text{A.8})$$

where Re_u is the Reynolds number based on the element parameters, *i.e.*

$$Re_u = \frac{\|\mathbf{u}^h - \mathbf{w}^h\| h}{2\nu} \quad (\text{A.9})$$

The element length h is computed by using equation (1.32).

$$z(Re) = \begin{cases} Re/3 & 0 \leq Re < 3, \\ 1 & 3 \leq Re \end{cases} \quad (\text{A.10})$$

Appendix B

Mesh generation using a CMD technique

The CMD method proposed in chapter 2 can be useful for mesh generation. In this case, the topology could be generated in an auxiliary domain where the mesh may be structured. Then, the boundary nodes in that mesh are relocated in the real boundary. This sharp movement of the boundary nodes is similar to the situation faced in mesh dynamics. Therefore, using the mesh untangling and smoothing technique presented, a valid mesh is generated. The generation of orthogonal meshes of quadrangles is perhaps the most interesting application from the practical point of view. In the next sections several examples of orthogonal mesh generation are presented and compared their results with analytical mappings.

B.1 Generation of orthogonal meshes

The minimum internal angle of elements is often used as a criterion for the quality of meshes. In this sense orthogonal meshes are optimal, since all the internal angles are right ones or almost. Methods like conformal mapping let to generate orthogonal meshes around moderately simple geometries. However, a general practical method for the generation of orthogonal meshes is still a subject of research.

As the CMD technique proposed in chapter 2 generates meshes through minimization of a functional related with the distortion of the elements, it may be possible that under certain conditions the method returns optimal meshes that are obtained through other analytical methods like conformal mapping, for instance.

The following list summarizes the interest in the generation of orthogonal meshes

- Orthogonal meshes are optimal according to the minimal internal angle criterion.
- Some numerical methods are simplified (and possibly faster) in orthogonal meshes.
- Some numerical methods require orthogonal meshes.

B.1.1 Transformation for the reentrant corner at right angle

Conformal mapping is a technique based on representing the 2D space as a complex plane. As transformations between complex planes that are derived from analytical functions preserve angles, orthogonal meshes can be obtained through mapping the actual domain

to a rectangle, generating an orthogonal (Cartesian) mesh in this rectangle, and mapping back the node positions to the actual domain. The mesh created is almost orthogonal, *i.e.* the angles at the intersections converge to $\pi/2$ under mesh refinement.

One of such a mesh is shown in figure B.1. It is obtained by mapping the complex plane $\zeta = \xi + i\eta$, onto $z = x + iy$ with the transformation $z = \zeta^{3/2}$. This transformation is often used to generate orthogonal meshes for the reentrant corner flow problem with a right angle. The mesh shown in the figure was obtained by applying the transformation to a homogeneous square mesh of 30×30 elements in the unit square in $0 \leq \xi, \eta \leq 1$.

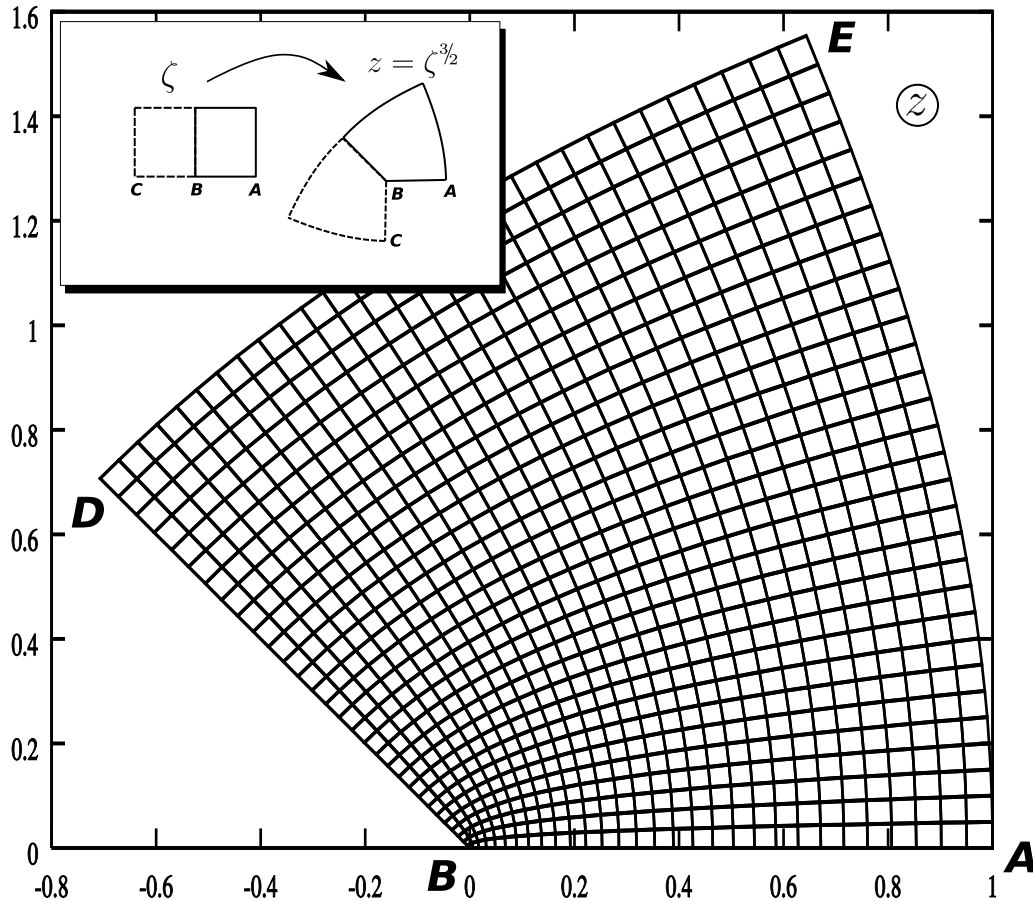


Figure B.1: Orthogonal mesh for the reentrant corner problem generated with the $z = \zeta^{3/2}$ mapping.

This mesh can be obtained with the proposed method by deforming the square mesh while imposing displacements to nodes at the sides DB and BA , and leaving free the nodes on sides AE and ED . However it is crucial to let the nodes on the DB and BA sides to slide freely along the wall, imposing a linear restriction on the displacements. If both displacements are imposed at this walls, then the mesh spacing along them will remain constant and the mesh will not be orthogonal. This is hard to achieve for curved boundaries, where the nodes must be leaved to slide freely representing a nonlinear restriction on the displacements. Furthermore, the nodes must be able to slide across the corners, as if they had a large but finite curvature. This restriction makes the use of this technique rather impractical for generating conformal meshes. However some examples are shown here only for the sake of demonstrating the feature.

In this particular case, the problem is ill-posed because for a given set of node positions $\{x_j\}_{j=1}^n$ a scalar multiple $\{\alpha x_j\}_{j=1}^n$ has the same functional value. In order to solve this problem, the functional is regularized by adding a term proportional to the volume of the element ($C_v \neq 0$ in definition (2.8)).

The mesh obtained with the proposed method is shown in figure B.3. Each quadrangular element has been split in four triangles (see equation (2.11)), so that each quadrangle appears plotted along with its diagonals. In order to assess quantitatively the error between the analytical mapping and the numerical one, the position of the nodes in the $x = \text{Re}\{z\}$ axis (*i.e.* the relation $x = x(\xi)$) are superimposed in figure B.2. There is no appreciable error in the numerical result.

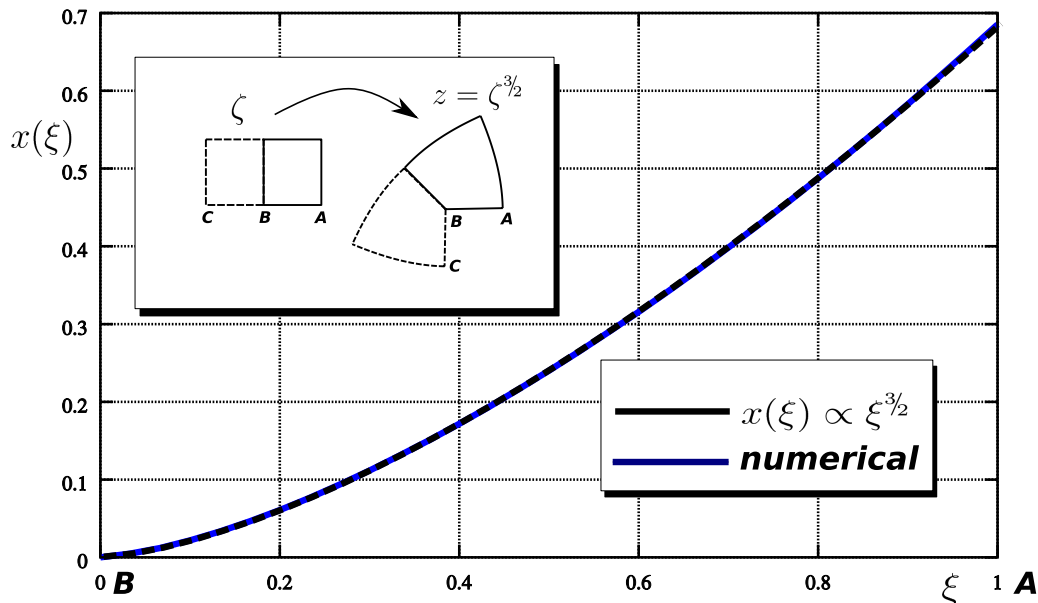


Figure B.2: Comparison of nodes position along the x -axis.

In order to check quantitatively the error in the orthogonality of the mesh, the angles at the corners of the quadrangular elements are computed and the maximum deviation from a right angle for all the elements converging to a node is computed. This is plotted for all the nodes of the mesh versus the radial distance to the origin B in figure B.4. Of course, there is a large error of 45° for the node at the origin B that is fixed and can not be improved. The orthogonality deviation quickly diminishes with the distance to the origin, and for $r = 0.1$ it is lower than 2.5° . Another criterion is the deviation of the angle between the diagonals at the point of intersection with respect to 90° . This is shown in the same figure and is much lower than the deviation at the nodes. This is always smaller than 1.5° and for $r > 0.1$ it is lower than 0.1° .

B.1.2 Transformation of a square into a rhombus

Another example where a analytical solution can be achieved is the transformation of a square into a rhombus (see figure B.5). In this case the nodes are left to slide freely along each side. Note that, by symmetry, the number and distribution of nodes will be the same at each side, so that the nodes will not slide across corners from one side to another. Any other quadrangle would have the nodes required to slide across the corners, from one

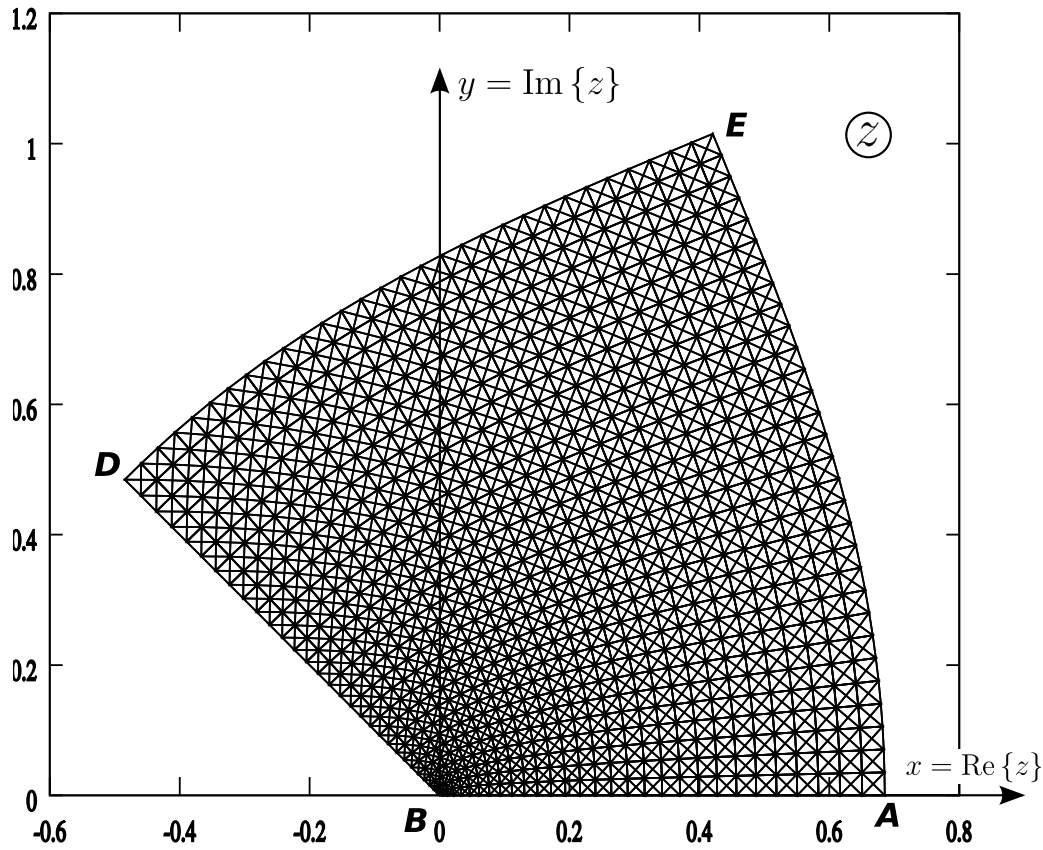


Figure B.3: Orthogonal mesh obtained with the proposed method for the reentrant corner. Quadrangle elements along with their diagonals are shown.

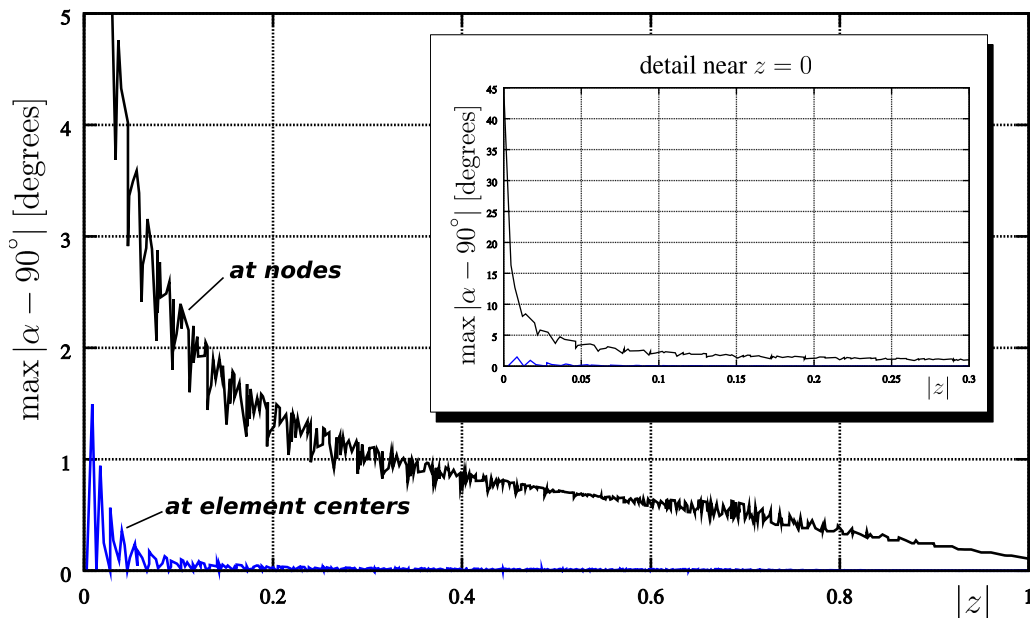


Figure B.4: Maximum deviation angle at the nodes and deviation angles at the center of the elements for the reentrant corner.

side to another and then requires special treatment for imposing this kind of boundary condition, which is no longer a linear restriction.

In figure B.5 the numerically obtained transformation is shown for a mesh of homogeneously spaced 30×30 quadrangular elements, and a tilt angle of 54° . Note the growth of the elements near the corners A and C , where the angle is reduced from the right angle, and the contraction of the elements at corners B and D . Again, the deviation of the angle between the diagonals respect to 90 degree has been computed and it is shown in figure B.6. The deviation angles have been sorted and plotted against the element index. It can be seen that, at a tilt angle of 54° there are 855 elements (95%) with a deviation angle under 0.35° . Furthermore, the deviation angles for finer meshes of 60×60 and 120×120 quadrangles are computed in order to check the convergence of the mesh to an orthogonal, conformal one. In figure B.7 the distribution of deviation angles for the three meshes is shown, in logarithmic plot. It is observed that while the reduction in deviation angles is roughly by a factor of 5 from the 30×30 to the 60×60 meshes, the reduction from the 60×60 to the 120×120 is by a factor of more than 100.

The conformal mapping from the square to the rhombus can be found analytically through means of two *Schwartz-Christoffel transformation* (SCT). The SCT allows the computation of the transformation between a half plane and an arbitrary region enclosed by a polyhedron. However, some free parameters of the SCT (the position of the transformed vertexes in the half plane boundary) must be computed numerically. This requires solving nonlinear equations whose residual functions involve singular integrals, and then is not straightforward. In this case two SCT's can be computed so that the first maps a half plane into the square and the second maps the half plane into the rhombus. Then, the transformation from the square into the rhombus is obtained composing the second with the inverse of the first.

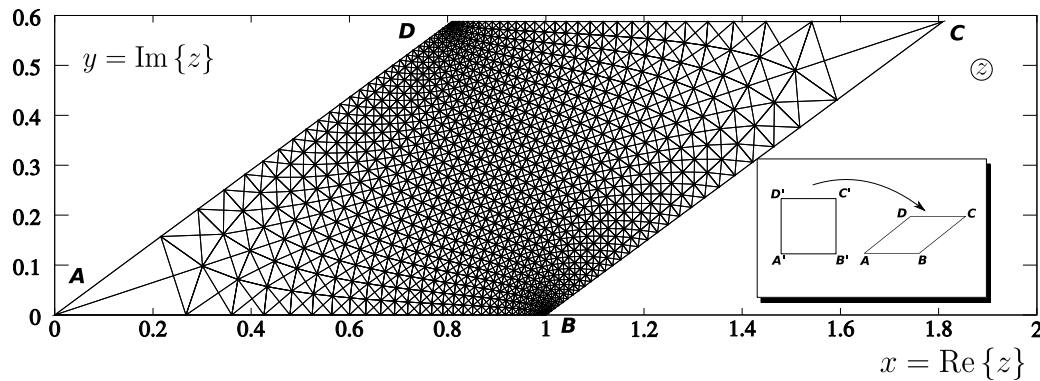


Figure B.5: Orthogonal mesh obtained with the proposed method for the transformation of a square into a rhombus. Quadrangle elements along with their diagonals are shown.

B.1.3 Ellipse

A half plane can be transformed in the exterior of an ellipse by composing the transformation $v = e^w$ that maps the half plane $\text{Re}\{w\} \geq 0$ in the exterior of a circle $|v| > 1$ with a Joukowski transformation

$$z = (1 + a^2)^{-1} \left(w + \frac{a^2}{w} \right) \quad (\text{B.1})$$

The value of a is adjusted with the eccentricity of the ellipse. In figure B.8 a mesh of 50×50 elements around an ellipse of eccentricity $\epsilon = 0.932$ (this corresponds to a minor

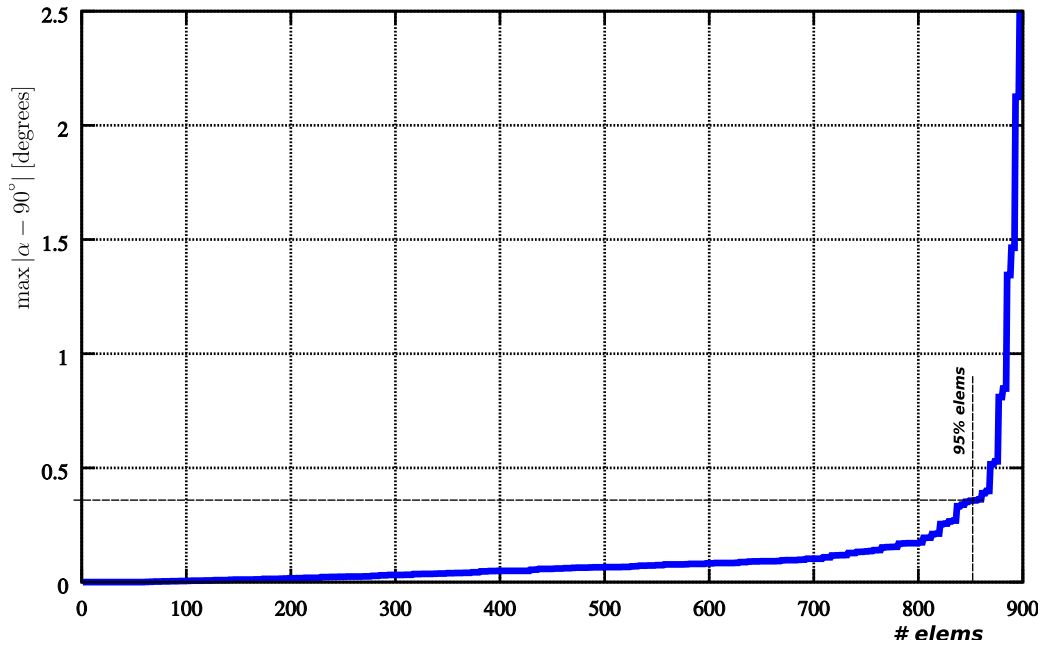


Figure B.6: Deviation angle for the intersection of quadrangle diagonals for the mesh of 30×30 elements.

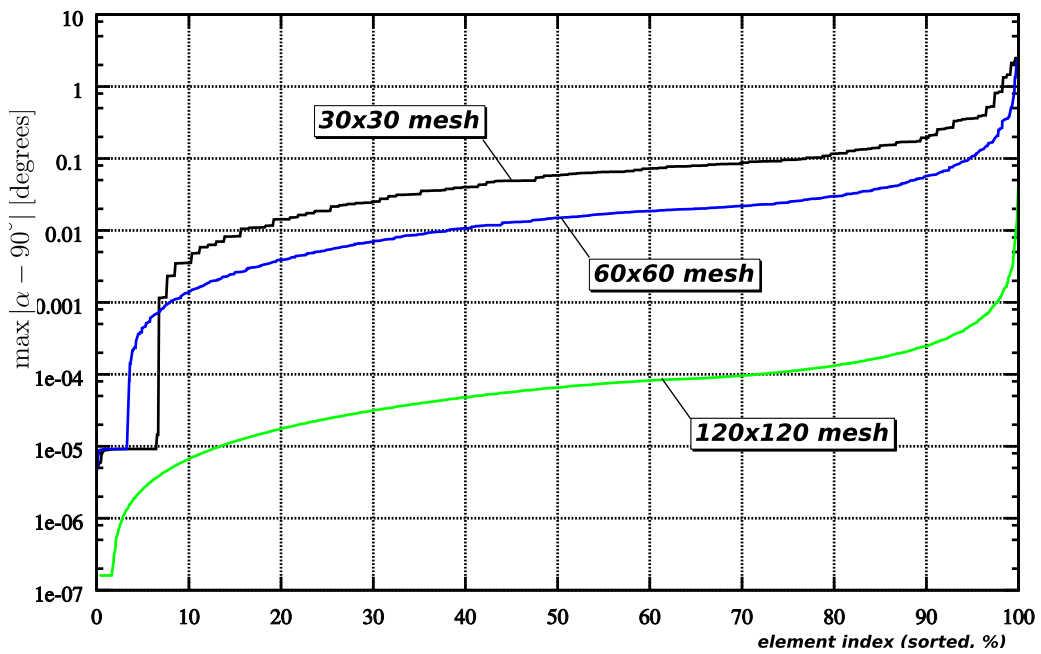


Figure B.7: Deviation angle for the intersection of quadrangle diagonals for the mesh of 30×30 , 60×60 and 120×120 elements.

to major axis ratio of $b/a = 0.361$, eccentricity is defined $\epsilon = \sqrt{1 - b^2/a^2}$). The nodes at the BC and AD sides are left to slide freely along the horizontal and vertical directions respectively, and the nodes at the outer boundary BD are left free. At the ellipse skin AB the nodes are allowed to slide freely along the curved boundary. Due to the curvature of the boundary, this is as a nonlinear restriction on the displacements of those nodes. These restrictions are imposed using penalization. The solution was obtained through continuation in the ellipse eccentricity and on the penalization parameter.

In figure B.9 it could be seen the element distribution of deviation from orthogonality. The maximum deviation is 0.002° and 95% of the elements have deviation under 8×10^{-4} degrees.

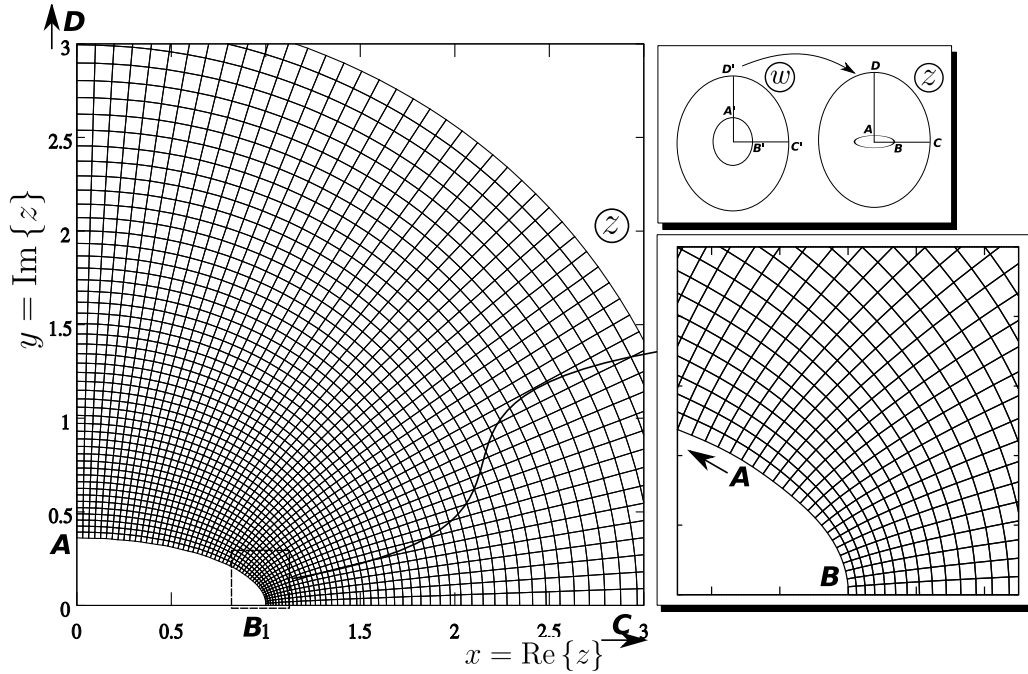


Figure B.8: Orthogonal mesh obtained with the proposed CMD method for the region outside an ellipse of eccentricity $\epsilon = 0.932$ ($b/a = 0.361$).

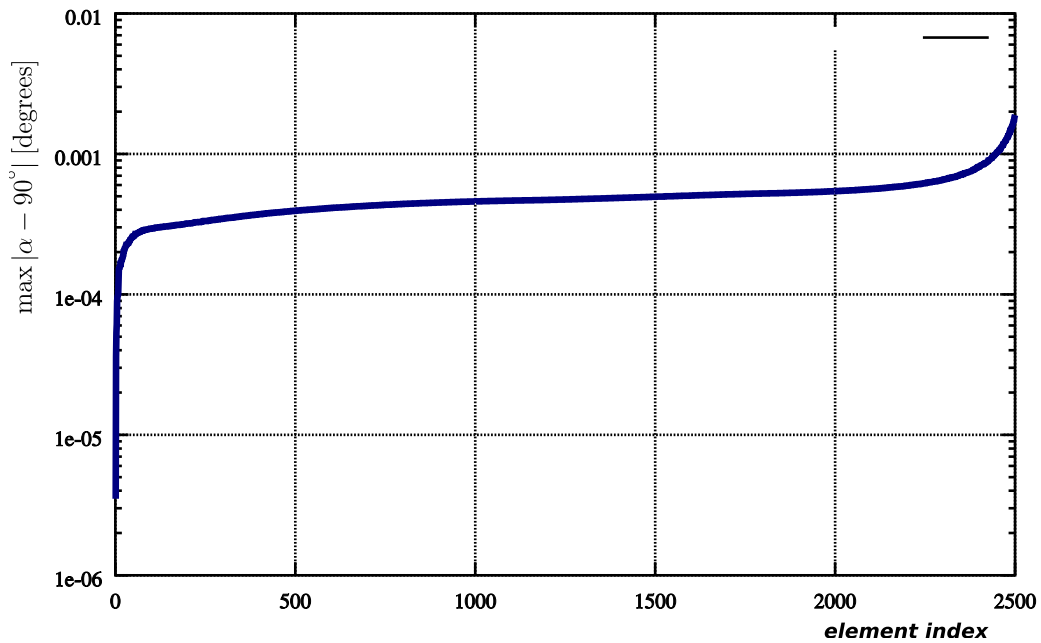


Figure B.9: Orthogonality deviation (at the diagonal intersection) for the mesh around an ellipse.

Appendix C

Description of MRCVC engine geometry

C.1 Volume and wall area of the chamber

The area calculated (A) corresponds to the total surface area of the side walls of the chamber, without including the upper and lower lids. Each of these can be calculated as the volume of the chamber (V) divided by the chamber height (h).

In the formulas developed in this appendix, the following radii and angles (specified in figure C.1) are used repeatedly

$$\begin{aligned} R_i &= \sqrt{R^2 - r^2} \\ R_e &= \sqrt{R^2 + 3r^2} \\ \phi &= \arcsin\left(\frac{r}{R}\right) \\ \delta &= \frac{\pi}{n} - \phi \\ \beta &= \frac{\pi}{n} + \phi - \arctan\left(\frac{2r}{R_i}\right) \end{aligned} \tag{C.1}$$

C.1.1 Interval 1

This interval extends from the initial position of reference to the position in which the outside apex of the leading vane touches the vertex of the rotor and, simultaneously, the interior apex of the vane touches the top apex of the internal region of the housing. The interval angular validity is

$$0 \leq \theta < \frac{n-2}{2n}\pi - \phi \tag{C.2}$$

The area and volume are calculated according to the following equations

$$A = 2h \left[2 \left(\phi R + \phi r + \frac{\pi}{n} R \right) + \delta R_i + \beta R_e \right] \tag{C.3}$$

$$V = h \left[4Rr \sin\left(\frac{\pi}{n}\right) \cos \theta + R_i (2r - 3\phi R_i) - \frac{4\pi r^2}{n} + R_e^2 \arccos\left(\frac{R^2 + r^2}{RR_e}\right) \right] \tag{C.4}$$

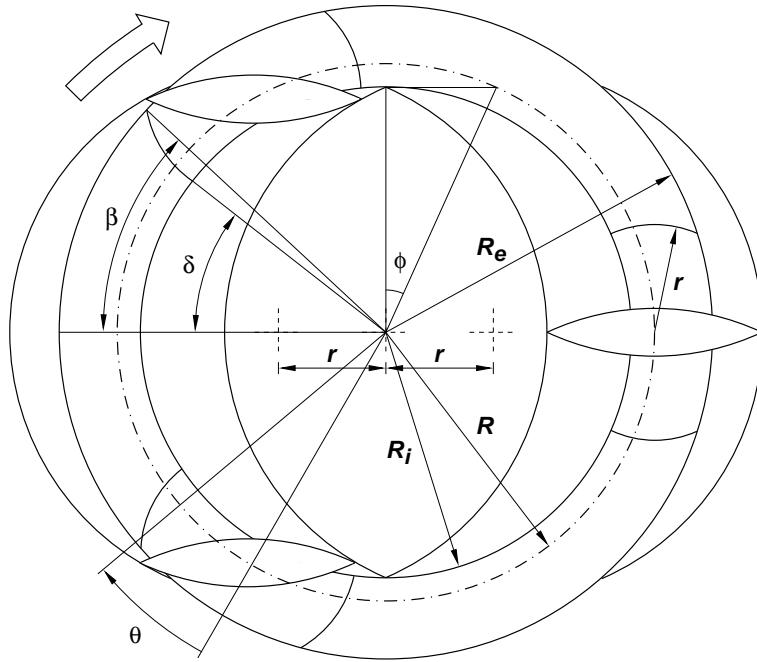


Figure C.1: Basic geometry of the MRCVC engine.

C.1.2 Interval 2

The second interval extends from the position of contact of the rotor with the apex of the leading vane, up when this vane is located in the middle of the region in which the volume of chamber remains constant. The angle of rotation of the output shaft is in the range

$$\frac{n-2}{2n}\pi - \phi \leq \theta < \frac{n-2}{2n}\pi \quad (\text{C.5})$$

with the area and volume of the chamber given by the expressions

$$\begin{aligned} A = h \left[\left(\frac{n-2}{2n}\pi + 3\phi - \theta \right) (R+r) + 2 \left(\frac{n+2}{2n}\pi - \phi - \theta \right) R \right. \\ \left. + 2\delta R_i - \left(\phi + \theta - 2\beta - \frac{n-2}{2n}\pi \right) R_e \right] \end{aligned} \quad (\text{C.6})$$

$$\begin{aligned} V = h \left\{ 2r \left[R_i + R \sin \left(\frac{\pi}{n} - \theta \right) \right] + \frac{1}{2} R_e^2 \arctan \left(\frac{2r}{R_i} \right) - 2r^2 \left(\frac{n+2}{2n}\pi - \theta \right) \right. \\ + \frac{1}{2} \left[R_e^2 \arccos \left(\frac{R^2 + r^2}{RR_e} \right) - 3\phi R_i^2 \right] + \sqrt{b(b-l_1)(b-l_2)(b-l_3)} \\ + \frac{1}{2} \left[r^2(\gamma_2 - \sin \gamma_2) - R^2(\gamma_1 - \sin \gamma_1) - R_i^2(\gamma_3 - \sin \gamma_3) \right] \\ \left. - R_i^2 \left[\epsilon_i - \frac{\sin(2\epsilon_i)}{2} \right] - R^2 \left[\epsilon_r - \frac{\sin(2\epsilon_r)}{2} \right] \right\} \end{aligned} \quad (\text{C.7})$$

where $b = (l_1 + l_2 + l_3)/2$, $l_1 = 2R \sin(\gamma_1/2)$, $l_2 = 2r \sin(\gamma_2/2)$, $l_3 = 2R_i \sin(\gamma_3/2)$, $\gamma_1 = (n-2)\pi/2n + \phi - \theta - 2\epsilon_r$, $\gamma_2 = (n-2)\pi/2n + \phi - \theta$, $\gamma_3 = (n-2)\pi/2n + \phi - \theta - 2\epsilon_i$, being

$$\begin{aligned}\epsilon_i &= \arcsin \left[\frac{\sqrt{4R^2a^2 - (a^2 + r^2)^2}}{2R_i a} \right] \\ \epsilon_r &= \arcsin \left[\frac{\sqrt{4R^2a^2 - (a^2 + r^2)^2}}{2Ra} \right] \\ a &= \sqrt{2R \left[R + R_i \cos \left(\frac{n-2}{2n} \pi - \theta \right) \right] - r^2}\end{aligned}$$

C.1.3 Interval 3

The upper limit of validity of the interval 3 corresponds to the position in which the leading vane leaves the contact with the rotor, starting again the contact with the housing. This interval is given by

$$\frac{n-2}{2n} \pi \leq \theta < \frac{n-2}{2n} \pi + \phi \quad (\text{C.8})$$

The following are the expressions for the calculation of the area and volume of the chamber

$$\begin{aligned}A &= h \left[\left(\frac{n-2}{2n} \pi + 3\phi - \theta \right) (R+r) + 2 \left(\frac{n+2}{2n} \pi - \phi - \theta \right) R + \right. \\ &\quad \left. 2\delta R_i - \left(\phi + \theta - 2\beta - \frac{n-2}{2n} \pi \right) R_e \right] \quad (\text{C.9})\end{aligned}$$

$$\begin{aligned}V &= h \left\{ 2r \left[R_i + R \sin \left(\frac{\pi}{n} - \theta \right) \right] + \frac{1}{2} R_e^2 \arctan \left(\frac{2r}{R_i} \right) - 2r^2 \left(\frac{n+2}{2n} \pi - \theta \right) + \right. \\ &\quad \left. \frac{1}{2} \left[R_e^2 \arccos \left(\frac{R^2 + r^2}{RR_e} \right) - 3\phi R_i^2 \right] + \sqrt{c(c-l_1)(c-l_2)(c-l_3)} - \right. \\ &\quad \left. R_i^2 (\gamma - \sin \gamma) \right\} \quad (\text{C.10})\end{aligned}$$

where $c = (l_1 + l_2 + l_3)/2$, $l_1 = 2R \sin(\gamma/2)$, $l_2 = 2r \sin(\gamma/2)$, $l_3 = 2R_i \sin(\gamma/2)$, $\gamma = (n-2)\pi/2n + \phi - \theta$.

C.1.4 Interval 4

This interval extends to

$$\frac{n-2}{2n} \pi + \phi \leq \theta < \frac{n+2}{2n} \pi - \phi \quad (\text{C.11})$$

establishing its upper limit for the position in which the apex of the vane encounters the internal part of the housing, and where the volume of the chamber reaches its minimum. The expressions for the area and volume for this region are given according to

$$A = h \left[2\phi(R+r) + \left(\frac{n+2}{2n}\pi - \phi - \theta \right) (2R+R_i) - \left(\phi + \theta - 2\beta - \frac{n-2}{2n}\pi \right) R_e \right] \quad (\text{C.12})$$

$$V = h \left\{ 2r \left[R_i + R \sin \left(\frac{\pi}{n} - \theta \right) \right] + \frac{1}{2} R_e^2 \arctan \left(\frac{2r}{R_i} \right) - 2r^2 \left(\frac{n+2}{2n}\pi - \theta \right) + \frac{1}{2} \left[R_e^2 \arccos \left(\frac{R^2 + r^2}{RR_e} \right) - 3\phi R_i^2 \right] \right\} \quad (\text{C.13})$$

C.1.5 Interval 5

This last interval covers a half of the period in which the volume of the chamber remains constant. The angle of rotation is in

$$\frac{n+2}{2n}\pi - \phi \leq \theta < \frac{n+2}{2n}\pi \quad (\text{C.14})$$

Both the volume and area remain constant and can be calculated using the following expressions

$$A = h \left[2\phi(R+r) + \left(\beta - \frac{\pi}{n} \right) R_e \right] \quad (\text{C.15})$$

$$V = h \left\{ 2r \left[R_i - R \sin \left(\frac{\pi}{2} - \phi \right) \right] + \frac{1}{2} R_e^2 \arctan \left(\frac{2r}{R_i} \right) - 2r^2 \phi + \frac{1}{2} \left[R_e^2 \arccos \left(\frac{R^2 + r^2}{RR_e} \right) - 3\phi R_i^2 \right] \right\} \quad (\text{C.16})$$

Appendix D

Resumen extendido en castellano

Título: Metodologías para la simulación numérica del flujo de fluidos en motores de combustión interna

En esta tesis se propone realizar el desarrollo teórico y la implementación computacional de un conjunto de nuevas herramientas o metodologías que permitan la resolución de problemas de la Mecánica de Fluidos Computacional en dominios con fronteras móviles. El desarrollo se plantea principalmente enfocado hacia la realización de flujometrías dinámicas virtuales en motores de Combustión Interna, en particular el novedoso Motor Rotativo de Combustión a Volumen Constante [80] (MRCVC).

Entre las diversas características que posee el flujo dentro de las cámaras de un motor de combustión interna, se listan a continuación las de mayor importancia [65]:

1. Flujo turbulento, viscoso y compresible.
2. Dominios 3D móviles y geometrías complejas.
3. Bajo número de Mach en gran parte del ciclo.
4. Flujo reactivo.
5. Inyección de combustible.
6. Las condiciones de contorno del problema son dinámicas y desconocidas *a priori*.

Los objetivos de esta tesis se centran en los ítems 2, 3 y 6 de la lista precedente.

D.1 Ecuaciones de gobierno

El modelo básico para un flujo viscoso compresible lo constituye el sistema de ecuaciones de Navier-Stokes. Aplicando una formulación ALE [22] (*Arbitrary Lagrangian Eulerian*) es posible resolver las citadas ecuaciones en dominios móviles. Se asume que el fluido es newtoniano y para el cual se verifican la hipótesis de Stokes y la ley de Fourier para el

flujo de calor. La ecuación de estado adoptada es la de un gas perfecto con constante R , relación de calores específicos $\gamma = c_p/c_v$, viscosidad dinámica μ y conductividad térmica κ . Bajo estas hipótesis y empleando la técnica ALE, el sistema de ecuaciones que gobierna el problema se escribe, en su forma cuasi-lineal, del siguiente modo [33]

$$\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{A}_i - w_i \mathbf{I}) \frac{\partial \mathbf{U}}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial x_j} \right) + \mathbf{S} \quad \text{en } \Omega_t \times (0, t_f) \quad (\text{D.1})$$

donde $\mathbf{U} = [\rho, \rho \mathbf{u}, \rho E]^T$ es el vector de variables conservativas, ρ es la densidad, \mathbf{u} el vector velocidad, E es la energía total por unidad de masa, $\mathbf{w} = [w_1, w_2, w_3]^T$ es la velocidad del sistema de referencia, \mathbf{A}_i y \mathbf{K}_{ij} son, respectivamente, las matrices jacobianas advectivas y difusivas [33], \mathbf{I} es el tensor identidad de segundo orden y $\mathbf{S} = [0, \rho \mathbf{f}_e, \rho \mathbf{f}_e \cdot \mathbf{u}]^T$ es el vector que contiene los términos fuente.

El modelado de la turbulencia se realiza mediante el modelo LES (*Large Eddy Simulation*) con la viscosidad turbulenta (μ_t) calculada mediante el modelo de Smagorinsky [69].

D.1.1 Formulación mediante elementos finitos

La discretización espacial del sistema de ecuaciones (D.1) se realiza aplicando el Método de Elementos Finitos estabilizado mediante la estrategia *Streamline Upwind/Petrov-Galerkin* (SUPG) y con la adición de un operador de *shock capturing*. Considerando nula la velocidad de la grilla (sistema de referencia), esta formulación se escribe como:

Hallar $\mathbf{U}^h \in \mathcal{S}^h$ tal que $\forall \mathbf{W}^h \in \mathcal{V}^h$

$$\begin{aligned} & \int_{\Omega} \mathbf{W}^h \cdot \left(\frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} \right) d\Omega + \int_{\Omega} \frac{\partial \mathbf{W}^h}{\partial x_i} \cdot \mathbf{K}_{ij}^h \frac{\partial \mathbf{U}^h}{\partial x_j} d\Omega \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \boldsymbol{\tau} (\mathbf{A}_k^h)^T \frac{\partial \mathbf{W}^h}{\partial x_k} \cdot \left[\frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} - \frac{\partial}{\partial x_i} \left(\mathbf{K}_{ij}^h \frac{\partial \mathbf{U}^h}{\partial x_j} \right) - \mathbf{S} \right] d\Omega^e \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \delta_{sc} \frac{\partial \mathbf{W}^h}{\partial x_i} \cdot \frac{\partial \mathbf{U}^h}{\partial x_i} d\Omega^e = \int_{\Omega} \mathbf{W}^h \cdot \mathbf{S} d\Omega + \int_{\Gamma} \mathbf{W}^h \cdot \mathbf{f} d\Omega \end{aligned} \quad (\text{D.2})$$

donde

$$\begin{aligned} \mathcal{S}^h &= \{ \mathbf{U}^h | \mathbf{U}^h \in [\mathbf{H}^{1h}(\Omega)]^{n_{dof}}, \mathbf{U}^h|_{\Omega^e} \in [P^1(\Omega^e)]^{n_{dof}}, \mathbf{U}^h = \mathbf{g} \text{ sobre } \Gamma_g \} \\ \mathcal{V}^h &= \{ \mathbf{W}^h | \mathbf{W}^h \in [\mathbf{H}^{1h}(\Omega)]^{n_{dof}}, \mathbf{W}^h|_{\Omega^e} \in [P^1(\Omega^e)]^{n_{dof}}, \mathbf{W}^h = \mathbf{0} \text{ sobre } \Gamma_g \} \end{aligned} \quad (\text{D.3})$$

con \mathbf{f} y \mathbf{g} representando los vectores de condiciones de contorno naturales y Dirichlet, respectivamente.

Los parámetros de estabilización son aquellos definidos por Aliabadi *et al.* [1]. La discretización en el tiempo se realiza empleando el esquema en diferencias trapezoidal.

D.2 Dinámica de la malla

Para resolver la dinámica de la malla fue propuesta una técnica de suavizado basada en un problema de optimización. En dicho problema el funcional representa la distorsión de

la malla y el mismo es resuelto en forma global. Tal funcional fue definido del siguiente modo [45]:

$$F(\mathbf{x}) = \sum_e F_e(\mathbf{x}) = \sum_e C_v \left(\frac{V_e}{V_{\text{ref}}} - 1 \right)^m + C_q q_e^n \quad (\text{D.4})$$

donde V_e es el volumen del elemento, V_{ref} es un volumen de referencia, q_e algún indicador de calidad elemental, y C_q y C_v son coeficientes que pesan la influencia del término de volumen y de calidad en el funcional, respectivamente. El exponente m debe ser par y $n \in \mathbb{Z}^-$ a fin de poner el problema de optimización como uno de minimización.

En esta tesis se propone utilizar el siguiente indicador geométrico de calidad

$$q = C \left[\sum_{i=1}^N (q_{S,i})^n \right]^{1/n} \quad (\text{D.5})$$

donde C es una constante de normalización tal que $0 < q \leq 1$, N es el número total de posibles subdivisiones del elemento en elementos símlices, $q_{S,i}$ se calcula para cada símlice de la subdivisión y esta dado por

$$q_S = \frac{V}{S_e} \quad (\text{D.6})$$

con $S_e = \sum_j l_j^{n_d}$, siendo l_j la longitud del lado j del símlice y n_d el número de dimensiones espaciales.

D.2.1 Estrategia de *untangling-smoothing* simultáneos

El funcional (D.4) es continuo siempre que $q_e \neq 0$ para todos los elementos de la malla, debido a que $n < 0$. $q_e \rightarrow 0$ cuando para algún símlice i de la subdivisión del elemento $V_i \rightarrow 0$, dado que $\sum_j l_{i,j}^{n_d}$ se encuentra acotado por debajo si el símlice no tiende a colapsar en un punto. Por lo tanto, la aplicación de la técnica se restringe a mallas válidas (sin elementos ‘invertidos’) debido a que se forman ‘barreras’ infinitas cuando el volumen de alguno de los elementos tiende a cero, impidiendo así la obtención de una malla válida a partir de una inválida. Con el objetivo de evitar los problemas asociados con las singularidades referidas, el funcional puede regularizarse reemplazando V en la ecuación (D.6) por la función

$$h(V) = \frac{1}{2}(V + \sqrt{V^2 + 4\delta^2}) \quad (\text{D.7})$$

Esta es una función del volumen positiva y estrictamente creciente. El parámetro δ representa el valor de la función cuando el volumen es nulo. Entonces, para volúmenes positivos, a medida que δ disminuye el funcional modificado se acerca cada vez más al funcional original, al igual que sus respectivos óptimos. En el límite cuando $\delta \rightarrow 0$, $F^*(\mathbf{x}) \rightarrow F(\mathbf{x})$ punto a punto. Así, definiendo una sucesión decreciente para δ se obtiene una estrategia de *untangling-smoothing* simultáneos de mallas. El k -ésimo elemento de la sucesión adoptada, correspondiente al paso k en la iteración de Newton-Raphson, se calcula aquí como [44]

$$\delta^k = \max(\delta^{k-1} - \tilde{\alpha}|\Delta\delta^k|, \tilde{\beta}\delta^{k-1}) \quad (\text{D.8})$$

con $\tilde{\alpha}$ y $\tilde{\beta}$ constantes menores que la unidad, y

$$\Delta\delta^k = - \left(\frac{\partial^2 F^*}{\partial \delta^2} \right)_{n,k-1}^{-1} \left[\left(\frac{\partial F^*}{\partial \delta} \right)_{n,k-1} + \left(\frac{\partial^2 F^*}{\partial \delta \partial \mathbf{x}} \right)_{n,k-1} (\mathbf{x}^{n,k} - \mathbf{x}^{n,k-1}) \right] \quad (\text{D.9})$$

D.3 Resolución de flujos compresibles a bajo número de Mach

Las velocidades características del flujo dentro de la cámara de un motor de combustión interna son del orden de la velocidad de los órganos móviles en contacto con el gas (pistón en un motor alternativo, rotor en un motor rotativo, etc.), excepto en los instantes de apertura/cierre de válvulas o lumbreras de admisión y escape. Siendo en general bajas las velocidades características, el número de Mach del flujo permanece en valores relativamente bajos, lo cual causa el mal condicionamiento de las ecuaciones y la posibilidad de falta de convergencia y precisión de los métodos numéricos aplicados. Luego, la resolución de flujos a bajos números de Mach suele acometerse utilizando alguna técnica que mejore el número de condición del sistema de ecuaciones que gobiernan el flujo, con el objetivo de obtener buenas soluciones numéricas.

D.3.1 Formulación del problema y análisis de autovalores

En esta tesis se propone utilizar la estrategia de preconditionamiento de las ecuaciones, la cual consiste en premultiplicar la derivada respecto al tiempo por una matriz apropiadamente definida. El objetivo es modificar los autovalores del sistema de tal forma que el número de condición se reduzca. La estrategia de preconditionamiento resulta útil cuando el problema a resolver es estacionario, debido a que los sistemas preconditionado y no preconditionado sólo comparten la misma solución estacionaria. Para aplicar el método de preconditionamiento a problemas no estacionarios se ha propuesto la técnica de ‘doble tiempo’ en la cual, para cada instante de tiempo (tiempo *físico*, t), se avanzan las ecuaciones en el *pseudo-tiempo* (τ) hasta alcanzar un estado pseudo-estacionario cuando $\tau \rightarrow \infty$. Introduciendo el término de la derivada pseudo-temporal afectada por un preconditionador $\mathbf{\Gamma}$, el sistema de ecuaciones para la estrategia de doble tiempo posee la forma [83]

$$\mathbf{\Gamma} \frac{\partial \mathbf{U}}{\partial \tau} + \frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_i \frac{\partial \mathbf{U}}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial x_j} \right) + \mathbf{S} \quad (\text{D.10})$$

Se analizará el preconditionador propuesto por Choi y Merkle [16] para la resolución de problemas estacionarios a bajo número de Mach, el cual fue aplicado en el contexto de elementos finitos por Nigro *et al.* [58, 55, 56].

Realizando un análisis de dispersión de la ecuación (D.10) se obtienen los autovalores del sistema los cuales, para el caso invíscido, poseen la forma

$$\begin{aligned} \lambda_{1,2,3} &= u_k (1 - ic_t CFL_u^{-1}) \\ \lambda_{4,5} &= \frac{u_k}{2} (1 - ic_t CFL_u^{-1}) T_{\pm} \end{aligned} \quad (\text{D.11})$$

donde

$$T_{\pm} = (1 + M_r^2 \chi) \pm \sqrt{(1 - M_r^2 \chi)^2 - 4M_r^2 \left[\frac{1}{(iM + c_t CFL_c^{-1})^2} + 1 - \chi \right]} \quad (\text{D.12})$$

D.3.2 Estrategias de preconditionamiento

La figura D.1 muestra el número de condición del sistema de ecuaciones de Euler en función del número de Courant CFL_c . El preconditionador para estado estacionario (SP, por *Steady Preconditioner*) corresponde a la definición del número de Mach de referencia (M_r) dada por Choi y Merkle [16]. Si el problema es no estacionario, Vigneron *et al.* [85] sugieren tomar

$$M_r = \min(1, \max(\sqrt{M^2 + CFL_c^{-2}}, M_\epsilon)) \quad (\text{D.13})$$

Esta definición es la denominada UP (*Unsteady Preconditioner*) en la figura D.1. Se incluye además el número de condición para cuando no se utiliza ningún preconditionador (NP), lo cual equivale a tomar $M_r = \chi^{-1/2}$ en la ecuación (D.12) cuando $\delta = 1$ en la definición de la matriz de preconditionamiento.

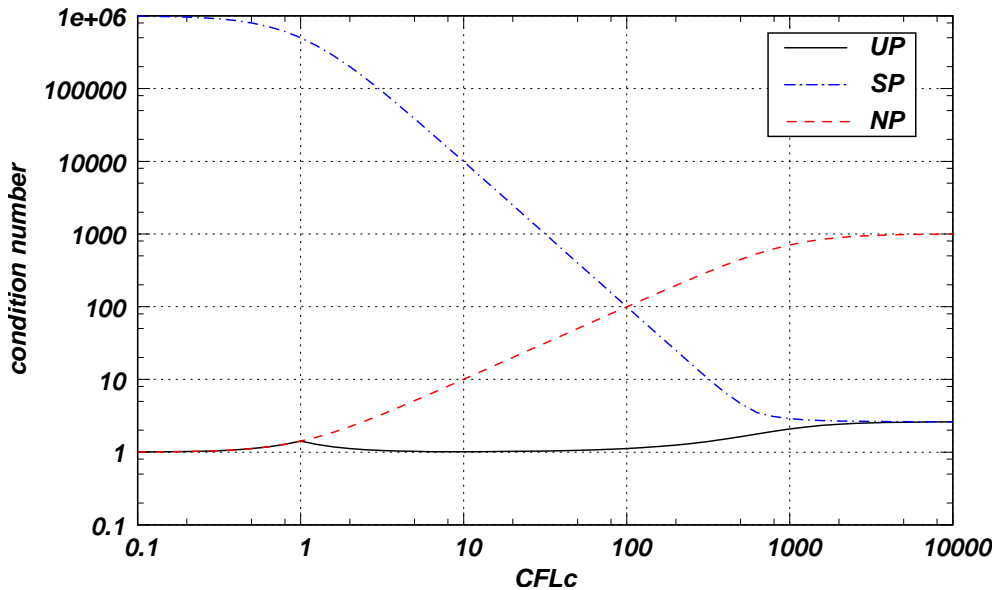


Figure D.1: Número de condición en función de CFL_c para $M = 1 \times 10^{-3}$.

Como se observa en la figura, adoptando el preconditionamiento ‘no estacionario’ se tiene un número de condición de orden 1 para todos los CFL_c .

D.4 Acoplamiento de dominios para flujo compresible

Dada la alta complejidad geométrica de los motores y de los procesos físicos que ocurren dentro de ellos, sólo es posible resolver una parte de tales máquinas con un modelo 3D. De este modo y por tratarse de un problema dinámico, una dificultad adicional la presentan

las condiciones de borde a imponer a dicho modelo. Usualmente estos problemas son abordados simulando el resto del motor mediante un simulador de motores 0D/1D, con lo cual se logra, por un lado, modelar toda la máquina simultáneamente (aunque con un nivel de detalle variable según el modelo) y, por otro, proveer de condiciones de contorno apropiadas al código 3D. Aplicando la referida aproximación, surge la necesidad de acoplar adecuadamente las soluciones obtenidas en los distintos dominios computacionales, las cuales pueden incluso ser calculadas por distintos códigos.

D.4.1 Acoplamiento mediante condiciones de contorno absorbentes

Un enfoque para resolver el acoplamiento entre los dominios puede ser emplear condiciones de contorno absorbentes en la interfaz.

Sea una grilla 1D con N elementos y $N + 1$ nodos numerados desde 1 a $N + 1$. Luego de una discretización estandar de las ecuaciones de flujo mediante algún método numérico (*e.g.* Elementos Finitos estabilizados mediante la técnica SUPG), se obtiene un sistema de ecuaciones no lineales de la forma

$$P \begin{cases} \mathbf{E}_1(\mathbf{U}_1, \mathbf{U}_2) = \mathbf{0} \\ \mathbf{E}_2(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3) = \mathbf{0} \\ \vdots \\ \mathbf{E}_{N+1}(\mathbf{U}_N, \mathbf{U}_{N+1}) = \mathbf{0} \end{cases} \quad (\text{D.14})$$

Se divide el sistema de ecuaciones en algún nodo interno i , de tal forma que el dominio $\Omega = [x_{1,1}, x_{1,N+1}]$ quede dividido en $\Omega_1 = [x_{1,1}, x_{1,i}]$ y $\Omega_2 = [x_{1,i}, x_{1,N+1}]$. De este modo, el nodo i posee ahora estados $i1$ e $i2$ en los subdominios izquierdo y derecho, respectivamente. Deben proveerse condiciones de contorno en el nodo $i1$ ($i2$) para Ω_1 (Ω_2) que aseguren que el problema en cada subdominio esté bien planteado y que pueda ser resuelto en forma independiente. Un esquema iterativo de resolución debe asegurar que la solución en cada subproblema converja a la solución del sistema acoplado, es decir,

$$\begin{aligned} \mathbf{U}_{i1}^k, \mathbf{U}_{i2}^k &\rightarrow \mathbf{U}_i \\ \mathbf{U}_j^k &\rightarrow \mathbf{U}_j \quad \forall j, j \neq i \end{aligned} \quad (\text{D.15})$$

para $k \rightarrow \infty$, con k denotando el número de iteración.

Se asume que la ecuación en el nodo i puede separarse en contribuciones de la izquierda y derecha

$$\mathbf{E}_i(\mathbf{U}_{i-1}, \mathbf{U}_i, \mathbf{U}_{i+1}) = \mathbf{E}_{i1}(\mathbf{U}_{i-1}, \mathbf{U}_i) + \mathbf{E}_{i2}(\mathbf{U}_i, \mathbf{U}_{i+1}) = \mathbf{0} \quad (\text{D.16})$$

Algoritmo basado en características

Para un sistema general, los problemas a ambos lados de la interfaz se escriben del siguiente modo:

$$\begin{cases}
 P_1 \left\{ \begin{array}{l}
 \mathbf{E}_1(\mathbf{U}_1^{k+1}, \mathbf{U}_2^{k+1}) = \mathbf{0} \\
 \mathbf{E}_2(\mathbf{U}_1^{k+1}, \mathbf{U}_2^{k+1}, \mathbf{U}_3^{k+1}) = \mathbf{0} \\
 \vdots \\
 \mathbf{E}_{i1}(\mathbf{U}_{i-1}^{k+1}, \mathbf{U}_{i1}^{k+1}) + \mathbf{\Pi}_{U1}^+ \mathbf{R}^k + \mathbf{\Pi}_{U1}^- \mathbf{R}^{k+1} = \mathbf{0} \\
 \mathbf{\Pi}_{U1}^-(\mathbf{U}_{i1}^{k+1} - \mathbf{U}_{i2}^k) = \mathbf{0} \\
 \mathbf{E}_{i2}(\mathbf{U}_{i2}^{k+1}, \mathbf{U}_{i+1}^{k+1}) - \mathbf{\Pi}_{U2}^+ \mathbf{R}^k - \mathbf{\Pi}_{U2}^- \mathbf{R}^{k+1} = \mathbf{0} \\
 \mathbf{\Pi}_{U2}^-(\mathbf{U}_{i1}^k - \mathbf{U}_{i2}^{k+1}) = \mathbf{0} \\
 \mathbf{E}_{i+1}(\mathbf{U}_{i2}^{k+1}, \mathbf{U}_{i+1}^{k+1}, \mathbf{U}_{i+2}^{k+1}) = \mathbf{0} \\
 \vdots \\
 \mathbf{E}_{N+1}(\mathbf{U}_N^{k+1}, \mathbf{U}_{N+1}^{k+1}) = \mathbf{0}
 \end{array} \right.
 \end{cases} \quad (\text{D.17})$$

donde $\mathbf{\Pi}_{Uj}^\pm = \mathbf{\Pi}_{Uj}^\pm(\mathbf{U}_{ij})$, $j = 1, 2$ son las matrices de proyección sobre las características que viajan a derecha/izquierda [72]. Luego de cada iteración, P_1 provee la parte de las reacciones que viaja a izquierda ($\mathbf{\Pi}_U^- \mathbf{R}^{k+1}$), mientras que P_2 proporciona la parte que viaja a derecha.

D.4.2 Acoplamiento para esquemas implícitos resueltos ‘molíticamente’

La estrategia de acoplamiento presentada en la sección anterior puede ser útil cuando las ecuaciones de gobierno en cada dominio son resueltas por códigos distintos. Empleando un esquema implícito en el tiempo y asumiendo que la resolución puede realizarse como un sistema ‘monolítico’, la estrategia de acoplamiento se reduce a una restricción entre los estados en los nodos de la interface.

D.5 Simulación numérica del motor rotativo MRCVC

Resultados de la simulación CFD bidimensional del MRCVC son presentados en la figura D.2, en la cual se muestra el módulo de la velocidad del flujo.

La figura D.3 muestra las curvas características estimadas empleando modelos 1D/0D para un MRCVC de tres aspas, 500 cm³ de cilindrada unitaria y relación geométrica de compresión de 9:1.

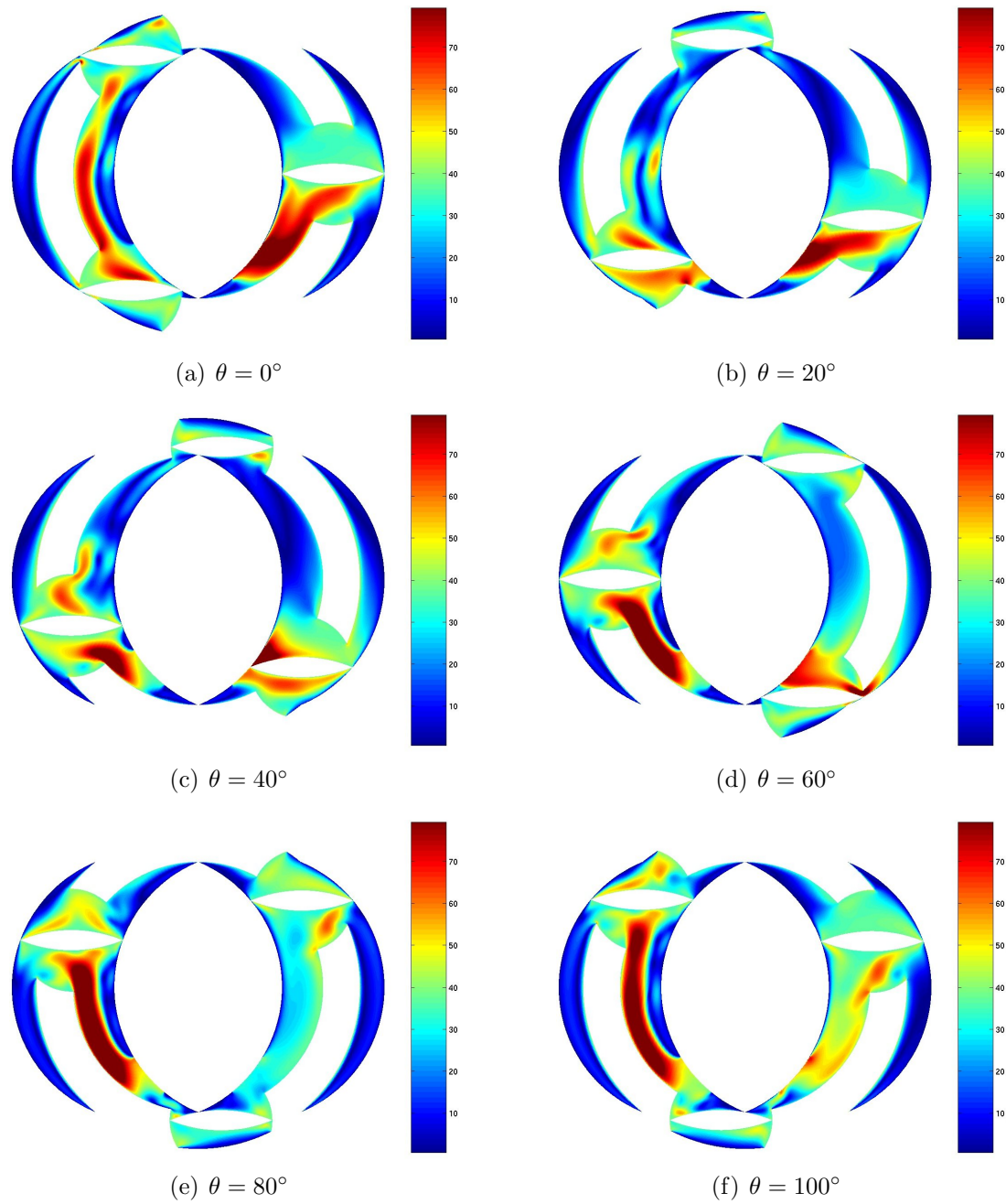


Figure D.2: Módulo de la velocidad ([m/s]) en las cámaras del MRCVC.

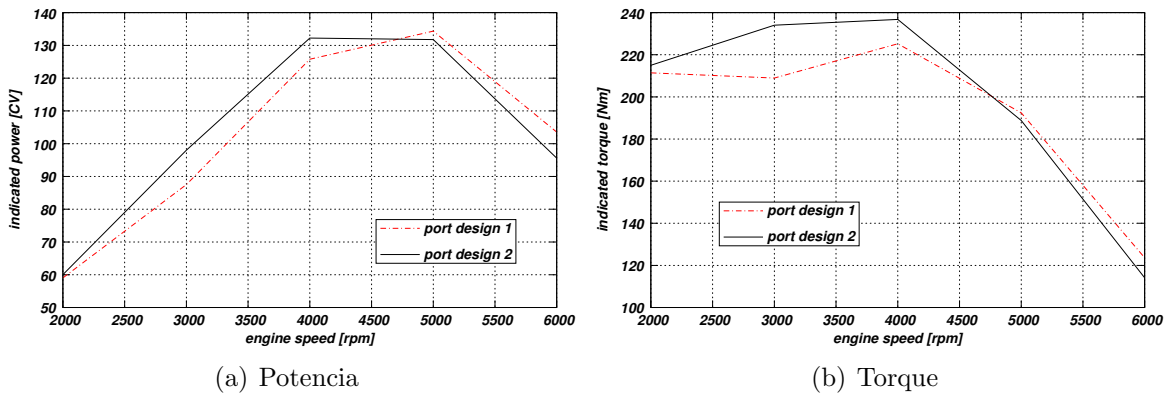


Figure D.3: Performance del motor rotativo MRCVC.

D.6 Conclusiones

El principal objetivo de esta tesis fue la propuesta, descripción y validación de algunas herramientas computacionales útiles para la resolución del flujo de fluidos en el interior de motores de combustión interna. Este tipo de problemas involucra varios aspectos a ser resueltos, de los cuales fueron abordados la dinámica de la malla, la resolución de flujos compresibles en límite de bajo número de Mach y el acoplamiento entre modelos dimensionalmente heterogéneos, en particular acoplamientos 1D/multi-D para flujos compresibles.

Las herramientas desarrolladas fueron aplicadas a la simulación del motor rotativo MRCVC. Se estimaron curvas características de este motor para dos tipos de diseño de puertos de admisión y escape. Además, fue realizada una simulación CFD bidimensional de las cámaras asumiendo condiciones de motor arrastrado.

Bibliography

- [1] S. Aliabadi, S. Ray, and T. Tezduyar. SUPG finite element computation of viscous compressible flows based on the conservation and entropy variables formulations. *Computational Mechanics*, 11:300–312, 1993.
- [2] J.P. Alianak and N.M. Nigro. Intake and exhaust system optimization of internal combustion engines. Technical Report RT-ID-012, Departamento de Ingeniería, Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, 2003.
- [3] A. Alonso, R.L. Trotta, and A. Valli. Coercive domain decomposition algorithms for advection-diffusion equations and systems. *Journal of Computational and Applied Mathematics*, 96:51–76, 1998.
- [4] A.A. Amsden. KIVA-3: A KIVA program with block-structured mesh for complex geometries. Technical report, Los Alamos, New Mexico, 1993.
- [5] A.A. Amsden, T.D. Butler, P.J. O'Rourke, and J.D. Ramshaw. KIVA-A comprehensive model for 2-D and 3-D engine simulations. *SAE paper*, 850554, 1985.
- [6] W. Annand. Heat transfer in the cylinders of reciprocating internal combustion engines. *Proc. Instn. Mech. Engrs.*, 177:973–980, 1963.
- [7] R.F. Ansdale. *The Wankel RC Engine Desing and Performance*. Iliffe Books, London, 1968.
- [8] D.N. Assanis. *A computer simulation of the turbocharged turbocompounded diesel engine system for studies of low heat rejection engine performance*. PhD thesis, Massachusetts Institute of Technology, 1985.
- [9] S. Balay, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, and H. Zhang. PETSc: Portable, extensible toolkit for scientific computation. <http://www.mcs.anl.gov/petsc/>, 2007.
- [10] R.M. Beam and R.F. Warming. An implicit finite-difference algorithm for hyperbolic systems in conservation-law form. *Journal of Computational Physics*, 22:87–110, 1976.
- [11] T. Belytschko, D.P. Flanagan, and J.M. Kennedy. Finite element methods with user-controlled meshes for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 33:669–688, 1982.

- [12] R. Benson. *The Thermodynamics and Gas Dynamics of Internal Combustion Engines*, volume I. Clarendon Press, Oxford, 1982.
- [13] P.E. Bjøstard and O.B. Widlund. Iterative methods for the solution of elliptic problems on regions partitioned into structures. *SIAM Journal on Numerical Analysis*, 23:1097–1120, 1986.
- [14] G.P. Blair. *The Basic Design of Two-Stroke Engines*. Society of Automotive Engineers, Inc., 1990.
- [15] A.N. Brooks and T.J.R. Hughes. Streamline upwind Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.
- [16] Y-H. Choi and C.L. Merkle. The application of preconditioning in viscous flows. *Journal of Computational Physics*, 105:207–223, 1993.
- [17] M.C. Ciccoli. Adaptive domain decomposition algorithms and finite volume/finite element approximation for advection-diffusion equations. *Journal of Scientific Computing*, 11(4):229–341, 1996.
- [18] J.M. Corberan. A new constant pressure model for N-branch junctions. *Proceedings of the Institution of Mechanical Engineers. Part D, Journal of automobile engineering*, 206:117–123, 1992.
- [19] L.D. Dalcín. *Techniques for high performance distributed computing in computational fluid mechanics*. PhD thesis, CIMEC-INTEC-UNL, June 2008.
- [20] M. Delanaye, Ch. Hirsch, and K. Kovalev. Untangling and optimization of unstructured hexahedral meshes. *Computational Mathematics and Mathematical Physics*, 43(6):807–814, 2003.
- [21] I. Demirdžić and M. Perić. Finite volume method for prediction of fluid flow in arbitrarily shaped domains with moving boundaries. *Journal of Algorithms*, 10:771–790, 1990.
- [22] J. Donea, S. Giuliani, and J.P. Halleux. An arbitrary, Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions. *SIAM Journal on Scientific Computing*, 33:689–700, 1982.
- [23] J.M. Escobar, E. Rodríguez, Montenegro R., G. Montero, and J.M. González-Yuste. Simultaneous untangling and smoothing of tetrahedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 192:2775–2787, 2003.
- [24] L. Formaggia, J.-F. Gerbeau, F. Nobile, and A. Quarteroni. Numerical treatment of defective boundary conditions for the Navier-Stokes equations. *SIAM Journal on Numerical Analysis*, 40(1):376–401, 2002.
- [25] L.A. Freitag and P. Plassmann. Local optimization-based simplicial mesh untangling and improvement. *International Journal for Numerical Methods in Engineering*, 49(1):109–125, 2000.

- [26] F. Gastaldi and L. Gastaldi. On a domain decomposition for the transport equation: Theory and finite element approximation. *IMA Journal of Numerical Analysis*, 14:111–135, 1993.
- [27] F. Gastaldi, L. Gastaldi, and A. Quarteroni. ADN and ARN domain decomposition methods for advection-diffusion equations. In Magne E. Espedal Petter E. Bjørstad and David E. Keyes, editors, *Ninth International Conference on Domain Decomposition Methods*, 1998.
- [28] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot. A dynamic subgrid-scale eddy viscosity model. In *Stanford Univ., Studying Turbulence Using Numerical Simulation Databases. 3: Proceedings of the 1990 Summer Program p 5-17 (SEE N92-30648 21-34)*, pages 5–17, 1990.
- [29] U. Ghia, K.N. Ghia, and C.T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 48:387–411, 1982.
- [30] H. Guillard and C. Viozat. On the behavior of upwind schemes in the low Mach number limit. *Computers & Fluids*, 28(1):63–86, 1999.
- [31] H.O. Hardenberg and F.W. Hase. An empirical formula for computing the pressure rise delay of a fuel from its cetane number and from the relevant parameters of direct-injection diesel engines. In *SAE International Congress and Exposition*, Detroit, Michigan, USA, 1979. SAE Paper N° 790493.
- [32] J.B. Heywood. *Internal Combustion Engine Fundamentals*. McGraw-Hill, Inc., 1988.
- [33] Ch. Hirsch. *Numerical Computation of Internal and External Flows. Volume 2: Computational Methods for Inviscid and Viscous Flows*. John Wiley & Sons, 1990.
- [34] J.P. Holman. *Heat Transfer*. McGraw-Hill, New York, 8th edition, 1997.
- [35] G. Houzeaux. *A Geometrical Domain Decomposition Method in Computational Fluid Dynamics*. PhD thesis, Escola Tècnica Superior d'Enginyers de Camins, 2002.
- [36] T. Hughes and M. Mallet. A new finite element method for CFD: III. The generalized streamline operator for multidimensional advection-diffusion systems. *Computer Methods in Applied Mechanics and Engineering*, 58:305–328, 1986.
- [37] T.J.R. Hughes, W.K. Liu, and T.K. Zimmermann. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29:239–349, 1981.
- [38] P. Knupp. Hexahedral mesh untangling and algebraic mesh quality metrics. *Proceedings of 9th International Meshing Roundtable*, pages 173–183, 2000.
- [39] P.M. Knupp. Algebraic mesh quality metrics. *SIAM Journal on Scientific Computing*, 23(1):192–218, 2001.

- [40] G. Le Beau, S. Ray, and T. Tezduyar. SUPG finite element computation of compressible flows with the entropy and conservation variables formulations. *Computer Methods in Applied Mechanics and Engineering*, 104:27–42, 1993.
- [41] J.S. Leiva and G.C. Buscaglia. Estrategias de acoplamiento entre códigos 0D/1D y códigos CFD 3D. *Mecánica Computacional, Volumen XXV*, pages 53–82, 2006.
- [42] C. Liu and Z. Liu. High order finite difference and multigrid methods for spatially evolving instability in a planar channel. *Journal of Computational Physics*, 106(1):92–100, 1993.
- [43] K. Liu and R.H. Pletcher. A fractional step method for solving the compressible Navier-Stokes equations. *Journal of Computational Physics*, 226:1930–1951, 2007.
- [44] E.J. López, N.M. Nigro, and M.A. Storti. Simultaneous untangling and smoothing of moving grids. *International Journal for Numerical Methods in Engineering*, 76(7):994–1019, 2008.
- [45] E.J. López, N.M. Nigro, M.A. Storti, and J.A. Toth. A minimal element distortion strategy for computational mesh dynamics. *International Journal for Numerical Methods in Engineering*, 69:1898–1929, 2007.
- [46] J.L. Lumley. *Engines: an introduction*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK, 1999.
- [47] J.M. McDonough. On intrinsic errors in turbulence models based on Reynolds-Averaged Navier-Stokes equations. *International Journal of Fluid Mechanics Research*, 22:27–55, 1995.
- [48] G.P. Merker, C. Schwarz, G. Stiesch, and F. Otto. *Simulating Combustion. Simulation of combustion and pollutant formation for engine-development*. Springer, Berlin, Germany, 2006.
- [49] C.L. Merkle. Computation of flows with arbitrary equations of state. *AIAA Journal*, 36:515–521, 1998.
- [50] S. Mittal and T.E. Tezduyar. A unified finite element formulation for compressible and incompressible flows using augmented conservation variables. *Computer Methods in Applied Mechanics and Engineering*, 161:229–243, 1998.
- [51] N. Miyamoto, T. Chikahisa, T. Murayama, and R. Sawyer. Description and analysis of diesel engine rate of combustion and performance using wiebe’s functions. In *SAE International Congress and Exposition*, Detroit, Michigan, USA, 1985. SAE Paper N° 850107.
- [52] R. Montenegro, J.M. Escobar, E. Rodríguez, G. Montero, and J.M. González-Yuste. Improved objective functions for tetrahedral mesh optimisation. *Lecture Notes in Computer Science*, 2657:568–578, 2003.

- [53] T. Murayama, N. Miyamoto, T. Yamada, J.-I. Kawashima, and K. Itow. A method to improve the solubility and combustion characteristics of alcohol-diesel fuel blends. In *SAE International Congress and Exposition*, Detroit, Michigan, USA, 1982. SAE Paper N° 821113.
- [54] N. Nigro, M. Storti, and L. Ambroggi. Modelización numérica de un motor de combustión interna monocilíndrico. *Revista Internacional de Métodos Numéricos para cálculo y diseño en Ingeniería*, 15:21–54, 1999.
- [55] N. Nigro, M. Storti, and S. Idelsohn. GMRES physics-based preconditioner for all Reynolds and Mach numbers: numerical examples. *International Journal for Numerical Methods in Fluids*, 25:1374–1371, 1997.
- [56] N. Nigro, M. Storti, S. Idelsohn, and T. Tezduyar. Physics based GMRES preconditioner for compressible and incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 154:203–228, 1998.
- [57] N.M. Nigro. *Simulación numérica de problemas de mecánica de fluidos por elementos finitos*. PhD thesis, Universidad Nacional de Córdoba, 1993.
- [58] N.M. Nigro. CFD by supercomputers. Post-doctoral thesis, Minnesota Supercomputer Institute, University of Minnesota, 1995.
- [59] T.H. Nomura. An arbitrary lagrangian-eulerian finite element method for interaction of fluid and a rigid body. *Computer Methods in Applied Mechanics and Engineering*, 95:115–138, 1992.
- [60] T.E. Oliphant. Guide to numpy. <http://numpy.scipy.org/>, 2006.
- [61] P.Y. Papalambros and D.J. Wilde. *Principles of Optimal Design. Modeling and Computation*. Cambridge University Press, 1988.
- [62] T.J. Pedley and K.D. Stephanoff. Flow along a channel with a time-dependent indentation in one wall: the generation of vorticity waves. *Journal of Fluid Mechanics*, 160:337–367, 1985.
- [63] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford Science Publications, 1999.
- [64] M.E. Ralph and T.J. Pedley. Flow in a channel with a moving indentation. *Journal of Fluid Mechanics*, 190:87–112, 1988.
- [65] D.E. Ramajo. *Simulación computacional de los procesos fluido-dinámicos en el interior de motores de combustión interna*. PhD thesis, CIMEC-INTEC-UNL, June 2008.
- [66] J.I. Ramos. *Internal combustion engine modeling*. Hemisphere Publishing Corporation, New York, 1989.
- [67] A.J. Reynolds. *Turbulent Flows in Engineering*. Wiley, New York, 1974.

- [68] P. Sagaut. *Large Eddy Simulation for Incompressible Flows, an Introduction*. Springer, Berlin, 2001.
- [69] J. Smagorinsky. General circulation experiments with the primitive equation: I the basic experiment. *Monthly Weather Review*, 91:216211, 1963.
- [70] J. Steger and R.F. Warming. Flux vector splitting of the inviscid gasdynamics equations with applications to finite-difference methods. *Journal of Computational Physics*, 40:263–293, 1981.
- [71] K. Stein, T.E. Tezduyar, and R. Benney. Automatic mesh update with the solid-extension mesh moving technique. *Computer Methods in Applied Mechanics and Engineering*, 193:2019–2032, 2004.
- [72] M.A. Storti, N.M. Nigro, R.R. Paz, and L.D. Dalcín. Dynamic boundary conditions in computational fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 197:1219–1232, 2008.
- [73] M.A. Storti, N.M. Nigro, R.R. Paz, L.D. Dalcín, and E.J. López. PETSc-FEM: A General Purpose, Parallel, Multi-Physics FEM Program. <http://venus.ceride.gov.ar/petscfem/>, 1999-2007.
- [74] H. Tennekes and J.L. Lumley. *A First Course in Turbulence*. MIT Press, Cambridge, Massachusetts, 1972.
- [75] T. Tezduyar and M. Senga. Determination of the shock-capturing parameters in SUPG formulation of compressible flows. In Tsinghua University Press & Springer-Verlag, editor, *Computational Mechanics WCCM IV*, Beijing, China, 2004.
- [76] T.E. Tezduyar. Computation of moving boundaries and interfaces and stabilization parameters. *International Journal for Numerical Methods in Engineering*, 43:555–575, 2003.
- [77] T.E. Tezduyar. *Finite Element Methods for Fluid Dynamics with Moving Boundaries and Interfaces. Volume 3: Fluids*. John Wiley & Sons, (eds. E. Stein, R. De Borst and T.J.R. Hughes), New York, 2004.
- [78] T.E. Tezduyar, S. Mittal, S.E. Ray, and R. Shih. Incompressible flow computations with stabilized bilinear and linear equal order interpolation velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering*, 95:221–242, 1992.
- [79] J. Toth, J. Di Nezio, C. Staniscia, and E. López. Ventajas mecánicas y termodinámicas de un nuevo motor rotativo. In *9° Congreso Chileno de Ingeniería Mecánica*, Valparaíso, Chile, October 2000.
- [80] J.A. Toth. *Motor Rotativo de Combustión a Volumen Constante (MRCVC)*, 2004. Patent Res. N° AR004806B1, Rec. N° P 19960105411.
- [81] R.L. Trotta. Multidomain finite elements for advection-diffusion equations. *Applied Numerical Mathematics*, 21:91–118, 1996.

- [82] E. Turkel, R. Radespiel, and N. Kroll. Assessment of preconditioning methods for multidimensional aerodynamics. *Computers & Fluids*, 26(6):613–634, 1997.
- [83] E. Turkel and V.N. Vasta. Local preconditioners for steady and unsteady flow applications. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39:515–535, 2005.
- [84] G. van Rossum. Python programming language. <http://www.python.org/>, 1990-2007.
- [85] D. Vigneron, G. Deliége, and Essers J-A. Low Mach number local preconditioning for unsteady viscous finite volumes simulations on 3D unstructured grids. *European Conference on Computational Fluid Dynamics*, 2006.
- [86] N. Watson, A.D. Pilley, and M. Marzouk. A combustion correlation for diesel engine simulation. In *SAE International Congress and Exposition*, Detroit, Michigan, USA, 1980. SAE Paper N° 800029.
- [87] F.M. White. *Viscous fluid flow*. McGraw-Hill, New York, 1974.
- [88] F.M. White. *Mecánica de fluidos*. McGraw-Hill, México, 1983.
- [89] D.C. Wilcox. *Turbulence Modeling for CFD*. D C W Industries, 2 edition, 2002.
- [90] G. Woschni. A universally applicable equation for the instantaneous heat transfer coefficient in the internal combustion engine. In *SAE International Congress and Exposition*, Detroit, Michigan, USA, 1967. SAE Paper N° 670931.
- [91] X. Xu, J.S. Lee, and R.H. Pletcher. A compressible finite volume formulation for large eddy simulation of turbulent pipe flows at low Mach number in cartesian coordinates. *Journal of Computational Physics*, 203:22–48, 2005.
- [92] P.D. Zavattieri, E.A. Dari, and G.C. Buscaglia. Optimizaton strategies in unstructured mesh generation. *International Journal for Numerical Methods in Engineering*, 39:2055–2071, 1998.