Numerical acceleration of aero-engine cavity flow solutions and flutter predictions

Edouard Minoux
Statement of Originality

The work presented in the thesis is, to the best of the candidate’s knowledge and belief, original and the candidate’s own work, except as acknowledged in the text. The material has not been submitted, either in whole or in part, for a degree or comparable award of Imperial College or any other university or institution.

Edouard Minoux
December 2017
The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.
Abstract

The research in this thesis concerns aeroelastic flutter computations in turbomachinery labyrinth seals. The presence of annular cavities adjacent to the seal translates numerically into a highly ill-conditioned system of nonlinear equations. As a result, current prediction methods based on the coupling of the nonlinear time-accurate Navier-Stokes equations with structural Finite Element calculations are prohibitively expensive for industrial routine design. To alleviate this issue, the thesis emphasises on numerical acceleration techniques which can be regrouped into two related parts:

The first introduces an uncoupled time-linearized harmonic method for flutter in annular cavity domains. By assuming a periodicity in the flow unsteadiness, the perturbations induced by the seal motion are linearized about the steady-state background flow. The novelty consists in taking advantage of the cyclic symmetry of the vibrating seal nodal diameter and assuming the unsteady flow to be space-periodic in the circumferential direction. In the frequency-domain, the harmonic problem is reduced to two dimensions, therefore allowing the background solution to be axisymmetric, greatly reducing the overall computational cost.

The linear equations defining the unsteady flow are strongly dependant on the solution of the steady-state problem, and so, the unsteady solution accuracy is related to that of the background flow. Unfortunately, obtaining converged steady-state solutions for the compressible Navier-Stokes equations in cavities is a nontrivial task, as the nonlinear set of equations arising from their discretization are exceptionally stiff. This leads to the second aspect of this research which focuses on iterative procedures able to accelerate the convergence rates of the nonlinear steady-state solution in cavities. To that end, a Newton-GMRES method is implemented: the linear problem arising at every Newton artificial time-step is solved using GMRES, a linear multigrid routine for unstructured grids is used to precondition the system. With slight modification, it will also be seen that the linear part of the solver can be recycled for the unsteady problem and that its performance can further be increased by the use of the so-called GMRES with deflated restart method.

The performance of the implemented iterative methods is assessed for both nonlinear and linearized problems. In some cases, the stiffness of the linearized problem causes the solution procedure to stagnate, this is relieved by the use of deflated GMRES and/or increasing the number of multigrid cycles in the preconditioning iterations. The capability of the linear harmonic method coupled with multigrid-preconditioned GMRES for large parametric studies is demonstrated by obtaining aeroelastic damping criteria across a range of cavity/seal configurations, flow conditions and vibrational mode shapes. This allows the determination of trends in the different mechanisms causing aeroelastic instability in cavity-seal domains.
Acknowledgements

I would like to express my gratitude to Dr. Luca di Mare for introducing me to the field of numerical methods and fluid dynamics in my master’s thesis and for then taking me on as a Ph.D student. Over the last 5 years, his continuous stream of knowledge has never ceased to impress me, I consider myself extremely lucky to have had such a passionate and available supervisor.

I must also thank Professor Jeff Green for his contributions on the more engineering aspects of the project and for providing the mode shapes which have been used in this thesis.

I am grateful for the financial support Rolls-Royce and EPSRC have provided me to carry out this work.

I would also like to thank my friends and colleagues from the VUTC especially Mr. Ioan Hadade for getting me started with parallel computing, Dr. Feng Wang for our discussions on CFD and time-linearized methods, and our group administrator Mr. Peter Higgs.

Throughout the last four years my friends have been a constant source of support and motivation, I would like to thank in particular Mr. Alexandre Kitching, Mr. Teddy Szemerg O’Connor and Dr. Maria Esperanza Barrera Medrano.

Of course, I would also like to thank my father Charles and sister Kate who are so important to me. Finally, a special thought goes to my grandfather Dr. Jean Minoux, for being one of my main inspirations.
This thesis is dedicated to my parents,
for their love and kindness.
Contents

Abstract 4

1 Introduction 21
  1.1 Labyrinth Seals in Gas Turbines ........................ 21
  1.2 Flutter in Turbomachinery .............................. 22
  1.3 Numerical Methods ...................................... 24
    1.3.1 Numerical Methods in Aeroelastic Problems .......... 25
    1.3.2 Iterative Methods for Nonlinear Steady-State Problems 27
    1.3.3 Time-Linearized Solution Procedures ................. 32
  1.4 Thesis Objectives ...................................... 34
  1.5 Organisation .......................................... 35

2 Time-Linearized Harmonic Method about Axisymmetric Background Flows 36
  2.1 Introduction ........................................... 36
  2.2 Governing Equations for Deforming Control Volumes ........ 37
  2.3 Harmonic Assumption .................................... 40
  2.4 Linearized Governing Equations .......................... 42
  2.5 Circumferential Contribution Approach .................... 42
  2.6 Summary .............................................. 43
6 Results II: Linearized Flow

6.1 Introduction ................................................................. 101
6.2 Validation ........................................................................ 103
6.3 Performance of Iterative Methods ...................................... 107
6.4 Flutter Results ............................................................... 115

7 Conclusion ........................................................................ 132

7.1 Summary of Thesis Achievements ....................................... 132
7.2 Potential Applications ..................................................... 134
7.3 Future Work ...................................................................... 135

Bibliography .......................................................................... 136

A Flutter prediction Comparison with Abbott’s Criteria ............ 145
# List of Figures

1.1 Labyrinth seal schematics: a) Stepped labyrinth seal and surrounding HP and LP cavities. b) Stepped labyrinth seal (close-up view). c) Straight-through labyrinth seal. ............................... 23

1.2 Flow Mach number distribution at typical operating conditions for a stepped labyrinth seal: a) Surrounding cavity view. b) Exit clearance view. ....... 28

2.1 Illustration of labyrinth seal with flexural travelling wave number ND = 3: a) (z,r) plane. b) (r,θ) plane, cross sectional view................ 37

2.2 Illustration of flow domain reduction to 2D axisymmetric computational grids using the harmonic assumption - (r,θ) plane. a) Original 3D computational grid. b) 3D harmonic grid (unperturbed vertices are shown for comparison. c) 3D harmonic grid - Enlarged view. d) 2D axisymmetric grid with linearized circumferential flux contributions acting as a diagonal term. 44

3.1 Subsonic flow over a NACA0012 airfoil: a) Computational grid - 12288 quadrilateral elements. b) Flow Mach number distribution. ............... 53

3.2 NACA0012 - Newton-GMRES $l_2$ residual norm convergence comparison between numerically and analytically obtained Jacobian-vectors products. 54

3.3 NACA0012 - Newton-GMRES $l_2$ residual norm convergence comparison between numerically and analytically obtained Jacobian-vectors products. a) Nonlinear convergence history. b) Equivalent RHS evaluations ....... 55

3.4 Illustration of the unstructured restriction process of fine grid quantities (left) to a coarse grid element (right) ..................... 58

3.5 Illustration of the unstructured prolongation process of coarse grid quantities (right) to a fine grid element (left) ......................... 59

3.6 METIS domain decomposition on 20 processors of a straight-through labyrinth with surrounding cavities. ......................... 61
3.7 METIS domain decomposition on 20 processors of a straight-through labyrinth with surrounding cavities. - Seal inner-volume view. .......................... 61

4.1 Control volume. ................................................................. 68

4.2 NACA0012 - Second-order Newton-GMRES $l_2$ residual norm convergence comparison for between van Albada and Venkatakrishnan flux limiters. .... 70

4.3 Straight-through seal: original computational grid - 163634 quadrilateral elements. ................................................................. 73

4.4 Straight-through seal in Figure 4.3: perturbed computational grid. .... 73

4.5 Straight-through seal - Inlet fin tip. a) Original computational grid b) Perturbed computational grid. ................................................. 74

4.6 Straight-through seal - Outlet fin tip. a) Original computational grid b) Perturbed computational grid. ................................................. 74

5.1 Computational domains - Cases A, B and C. ............................... 78

5.2 Multigrid levels for case A05 - Level 0: 163634 elements ............... 79

5.3 Multigrid levels for case A05 - Level 1: 43880 elements ............... 79

5.4 Multigrid levels for case A05 - Level 2: 13623 elements ............... 79

5.5 Multigrid levels (enlarged views) for case A05 .......................... 80

5.6 Computational grid for case B05 - Level 0: 266228 elements ........... 81

5.7 Computational grid for case C05 - Level 0: 178111 elements ........... 81

5.8 Start-up phase - Nonlinear convergence history for case A05 with $p_r = 3.5$ . 84

5.9 Flow function at Inlet/Outlet boundaries and relative error for case A05 with $p_r = 3.5$ ................................................................. 84

5.10 Nonlinear convergence history against outer iterations: Newton-GMRES and baseline solver - Case A & B for lowest ($p_r = 1.1$) and highest ($p_r = 3.5$) pressure ratios. ................................................................. 87

5.11 Nonlinear convergence history against CPU-time (hours): Newton-GMRES and baseline solver - Case A & B for lowest ($p_r = 1.1$) and highest ($p_r = 3.5$) pressure ratios. ................................................................. 88

5.12 Nonlinear convergence history: Newton-GMRES and baseline solver - Case C with $p_r = 3.2$. ................................................................. 91
5.13 Newton-GMRES Nonlinear convergence history: Multigrid cases levels $n_{\text{lev}} = 1, 2, 3$ with $n_{\text{cy}} = 1, 5$ - Case C with $p_r = 3.2$.

5.14 First 150 approximate eigenvalues of $M^{-1}A$ and $A$ (no preconditioning) obtained from converged solution - Multigrid case levels $n_{\text{lev}} = 1, 2, 3$ with $n_{\text{cy}} = 1, 5$ - Case C with $p_r = 3.2$.

5.15 GMRES Linear convergence history - Multigrid cases levels $n_{\text{lev}} = 1, 2, 3$ with $n_{\text{cy}} = 1, 5$ - Case C with $p_r = 3.2$.

5.16 Nonlinear convergence history - Cases A & B.

5.17 Cases A & B: flow function against pressure ratios. - Comparison between CFD and analytical method.

5.18 Static pressure distribution for different pressure ratios on seal rotor - cases A

5.19 Static pressure distribution - case B05 - $p_r = 3.5$

5.20 Computed streamlines - Case C05 - $p_r = 3.2$.

5.21 Computed streamlines - Case B05 - $p_r = 3.5$.

5.22 Computed streamlines - Case B05 inner seal volume - $p_r = 3.5$.

5.23 Computed streamlines - Case A05 - $p_r = 3.5$.

5.24 Computed streamlines - Case A05 inner seal volume - $p_r = 3.5$.

5.25 Flow Mach number distribution - Case A05 inner seal volume.

6.1 Rotor deformation for the first five nodal diameter numbers: a) Case A. b) Case B.

6.2 Linear residual norm of case C with $p_r = 3.2$ for first 5 forward travelling wave number against: a) Krylov vector evaluation (linear iterations). b) CPU-time.

6.3 Logarithmic decrement history against linear residual norm - Case C with $p_r = 3.2$.

6.4 Logarithmic decrement for rearward and forward travelling wave numbers - case C with $p_r = 3.2$: a) Time-linearized results. b) Comparison with time-accurate method.

6.5 Influence of the nonlinear background flow on the time-linearized iterative solution procedure with and without deflated restart - case A05 with $p_r = 1.1$ and 3.5 for ND = 1: a) Linear residual norm against Krylov vector evaluation. b) Linear residual norm against CPU-time. c) Logarithmic decrement history against CPU-time. d) First 150 approximate eigenvalues of non-preconditioned complex operator $A$. 
6.6 Linear residual norm against Krylov vector evaluation using GMRES(150)/GMRES-DR(150,30) with $n_{cy} = 3, 10$ for a selection of stagnating cases. 111

6.7 Linear residual norm against CPU-time using GMRES(150)/GMRES-DR(150,30) with $n_{cy} = 3, 10$ for a selection of stagnating cases. 112

6.8 First 150 approximate eigenvalues of preconditioned complex operator $\mathcal{M}^{-1} \hat{A}$ for $n_{cy} = 3, 10$. - Cases A03 and B03. 113

6.9 First 150 approximate eigenvalues of preconditioned complex operator $\mathcal{M}^{-1} \hat{A}$ for $n_{cy} = 10$ - Case A03, $p_r = 1.1$, ND = 2. a) 30 first dominant $\lambda(\mathcal{M}^{-1} \hat{A})$ used in GMRES-DR(150,30). b) Enlarged view. 114

6.10 Pressure eigenvector magnitude associated with dominant $\lambda(\mathcal{M}^{-1} \hat{A})$ with $n_{cy} = 10$ - Case A03, $p_r = 1.1$, ND = 2. 114

6.11 Logarithmic decrements at different steady-state pressure ratios and nodal diameter numbers. - Cases A & B 116

6.12 Logarithmic decrement against MRTP for different nodal diameters - cases A 117

6.13 Logarithmic decrement against MRTP for different nodal diameters - cases B 118

6.14 Logarithmic decrement against background flow pressure ratios $p_r = 1.1-3.5$ for selected cases. 119

6.15 Work contributions of surrounding fluid flow on different seal locations - Selection of cases showing different aeroelastic mechanisms. 121

6.16 Case B03 - $p_r = 1.7, 2.4$ stable & $p_r = 3.2$ unstable, ND = -2. a,c) Unsteady pressure amplitude. b,d) Unsteady pressure phase. 122

6.17 Case B03 - $p_r = 1.7, 2.4, 3.2$ unstable, ND = -5. a,c) Unsteady pressure amplitude. b,d) Unsteady pressure phase. 123

6.18 Logarithmic decrement against background flow pressure ratios $p_r = 1.1-3.5$ - Influence of seal clearance and pressure ratio. 124

6.19 Work contributions of surrounding fluid flow on different seal locations - Case B03 & B05 ND = +5. 126

6.20 Work contributions of surrounding fluid flow on different seal locations - Case A05 ND = 1, 2. 126

6.21 Case A05 - $p_r = 3.5$, ND = +1 stable & ND = +2 unstable. a,c) Unsteady pressure amplitude. b,d) Unsteady pressure phase. 127

6.22 Work contributions of surrounding fluid flow on different seal locations - Case B03 & B05 ND = +3. 128
6.23 Case B03 - $p_r = 1.7, 2.4, 3.2$ stable, ND = +3. a,c) Unsteady pressure amplitude. b,d) Unsteady pressure phase. .............................................. 129

6.24 Case B05 - $p_r = 1.7$ stable & $p_r = 2.4, 3.2$ unstable, ND = +3. a,c) Unsteady pressure amplitude. b,d) Unsteady pressure phase. .............. 130
List of Tables

5.1 Computational test cases ........................................ 76
5.2 Characteristics of computational grids for cases A, B and C .......... 76

6.1 Natural frequencies (Hz) for the first five nodal diameters modes - Cases A, B and C .............................................. 102

A.1 Cavity acoustic frequencies against labyrinth seal natural frequencies - Comparison with Abbott’s criteria - A03 for $p_r = 3.5$. ........................................ 146
A.2 Cavity acoustic frequencies against labyrinth seal natural frequencies - Comparison with Abbott’s criteria - A05 for $p_r = 3.5$. ........................................ 147
A.3 Cavity acoustic frequencies against labyrinth seal natural frequencies - Comparison with Abbott’s criteria - B03 for $p_r = 3.5$. ........................................ 148
A.4 Cavity acoustic frequencies against labyrinth seal natural frequencies - Comparison with Abbott’s criteria - B05 for $p_r = 3.5$. ........................................ 149
List of Algorithms

1. Arnoldi orthogonalization procedure .................................................. 48
2. Right-preconditioned Arnoldi orthogonalization procedure ..................... 56
3. Linear Multigrid V-cycle pseudo-code for approximately solving $Mw = v_j$ 57
4. Pseudo-code of Morgan’s GMRES-DR$(m,k)$ procedure .......................... 65
# Nomenclature

## Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>Two-Dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>Three-Dimensional</td>
</tr>
<tr>
<td>AMG</td>
<td>Algebraic multigrid</td>
</tr>
<tr>
<td>Bi-CGSTAB</td>
<td>Bi-Conjugate Gradient STABlized</td>
</tr>
<tr>
<td>CFD</td>
<td>Computational Fluid Dynamics</td>
</tr>
<tr>
<td>CFL</td>
<td>Courant-Friedrich-Lewy number</td>
</tr>
<tr>
<td>FAS</td>
<td>Full Approximation Scheme</td>
</tr>
<tr>
<td>FE</td>
<td>Finite Element</td>
</tr>
<tr>
<td>GMG</td>
<td>Geometric multigrid</td>
</tr>
<tr>
<td>GMRES</td>
<td>Generalized Minimal RESidual</td>
</tr>
<tr>
<td>GMRES(m)</td>
<td>Restarted GMRES</td>
</tr>
<tr>
<td>GMRES-DR(m,k)</td>
<td>GMRES with deflated restart</td>
</tr>
<tr>
<td>HP</td>
<td>High-Pressure</td>
</tr>
<tr>
<td>ILU</td>
<td>Incomplete LU factorization</td>
</tr>
<tr>
<td>JFNK</td>
<td>Jacobian-Free Newton-Krylov</td>
</tr>
<tr>
<td>LHS</td>
<td>Left-Hand-Side</td>
</tr>
<tr>
<td>LP</td>
<td>Low-Pressure</td>
</tr>
<tr>
<td>MUSCL</td>
<td>Monotonic Upstream-Centered Scheme for Conservation Laws</td>
</tr>
<tr>
<td>ND</td>
<td>Nodal Diameter number</td>
</tr>
<tr>
<td>NK</td>
<td>Newton-Krylov</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>NKMG</td>
<td>Multigrid Preconditioned Newton-Krylov method</td>
</tr>
<tr>
<td>NS</td>
<td>Navier-Stokes</td>
</tr>
<tr>
<td>RANS</td>
<td>Reynolds-Averaged Navier-Stokes</td>
</tr>
<tr>
<td>RHS</td>
<td>Right-Hand-Side</td>
</tr>
<tr>
<td>SER</td>
<td>Switched-Evolution Relaxation scheme</td>
</tr>
</tbody>
</table>

**Calligraphic Symbols**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{K}_m$</td>
<td>Krylov subspace</td>
</tr>
<tr>
<td>$\mathcal{M}$</td>
<td>Preconditioning operator</td>
</tr>
<tr>
<td>$\mathcal{P}$</td>
<td>Multigrid prolongation operator</td>
</tr>
<tr>
<td>$\mathcal{R}$</td>
<td>Multigrid restriction operator</td>
</tr>
<tr>
<td>$S$</td>
<td>Multigrid smoothing operator</td>
</tr>
<tr>
<td>$\Re, \Im$</td>
<td>Real and imaginary operators</td>
</tr>
</tbody>
</table>

**Greek Symbols**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta$</td>
<td>Initial linear residual Euclidian norm (GMRES)</td>
</tr>
<tr>
<td>$\delta u$</td>
<td>Newton correction</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>Time-step</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Logarithmic decrement</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>Perturbation factor for finite difference Jacobian</td>
</tr>
<tr>
<td>$\epsilon_{\text{tol}}$</td>
<td>Termination criterion for time-linearized equations</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Forcing term</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Specific heat ratio</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>Thermal conductivity coefficient</td>
</tr>
<tr>
<td>$\mu, \mu_t, \mu_l$</td>
<td>Total, molecular and turbulent viscosity</td>
</tr>
<tr>
<td>$\Omega$</td>
<td>Rotational frame angular velocity</td>
</tr>
<tr>
<td>$\Phi$</td>
<td>Flux limiter</td>
</tr>
<tr>
<td>$\Psi$</td>
<td>Flux accumulation in the control volume surface normal direction</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Density</td>
</tr>
</tbody>
</table>
\( \tau \)  
Viscous stress terms

\( v^{(l)} \)  
Number of smoothing iterations at a grid level \( l \)

\( \zeta \)  
SER tuning factor

**Roman Symbols**

\( A \)  
Pseudo-transient Jacobian

\( \hat{A} \)  
Coefficient matrix of the time-linearized equations or *complex operator*

\( A_c \)  
Seal clearance area

\( a \)  
Speed of sound

\( e_m \)  
Identity matrix \( m^{th} \) column

\( e \)  
Specific energy

\( (F, G, H) \)  
Fluxes in the cylindrical coordinate system

\( F \)  
Flow function

\( \bar{H}_m \)  
\((m + 1) \times m\) Hessenberg matrix (GMRES)

\( h \)  
Specific enthalpy

\( I \)  
Identity matrix

\( i \)  
\( \sqrt{-1} \)

\( J \)  
Residual function Jacobian

\( l_2 \)  
Euclidean norm

\( M \)  
Mach number

\( m \)  
Krylov basis size

\( N \)  
\( N_{eq} \times N_{nodes} \)

\( N_{eq} \)  
Number of flow equations per grid element

\( N_{lev} \)  
Number of grid levels (multigrid)

\( N_{nodes} \)  
Number of grid elements

\( n_{cy} \)  
number Multigrid preconditioning V-cycles

\( p_b \)  
Static pressure at the seal rotor boundary

\( p_i \)  
Inlet static pressure
Pressure ratio
Nonlinear residual function
Linear residual
Rotational frame source term
Inlet total temperature
Vector of conservative variables
Velocity vector
Complex flow perturbation
Control volume surface normal velocity
Velocity components in the cylindrical coordinate system
Matrix of m Krylov vectors
Rotor wall velocities
Aerodynamic work
Aerodynamic work at the rotor boundary
Excitation frequency associated with the k nodal diameter mode shape
Acoustic frequency
Chapter 1

Introduction

1.1 Labyrinth Seals in Gas Turbines

Gas turbine engines produce work through a thermodynamic cycle involving a combustion process. In the power industry, mechanical energy is converted to electricity via an alternator. In the case of aero-engines, the focus of this thesis, part of the exhaust gases, via the turbine, is used to drive the compressor and fan assemblies, increasing the air density prior to the combustion process. The remainder, through a mechanical process is expanded and converted into propulsive thrust. The turbine and compressor parts are composed of successive static and rotating blade-row stages which impose on the working fluid the thermodynamic property variations (e.g. pressure, temperature) needed to operate the engine.

It is crucial to maintain the desired pressure in each part of the engine, and so the need for a way to seal the working fluid becomes apparent. To that end, labyrinth seals are placed in the primary gas path (compressor and turbine) along the shaft, over rotor blade tips and between stages. Modern high-bypass engines can incorporate over 50 labyrinth seals. They are essential components in minimizing gas recirculation between stages or preventing hot gas from entering compressor/turbine bearing cavities. Labyrinth seals also influence the mechanical design of the engine as their location determines the thrust load imposed on the bearings.

To accommodate for the rotating nature of turbomachinery parts, labyrinth seals are composed of a non-contacting rotating (rotor) and stationary (stator) part, the concession being that the sealing is not complete: a leakage flow will occur across the seal. To limit this behaviour, the rotor is built with a series of radial knife-edge teeth (fins) resulting in a series of cavities in between. The leakage flow is successively contracted and expanded: at every fin tip the flow is throttled, the pressure difference drives the flow to form an annular
jet, the kinetic energy of which is then dissipated into the inner-fin cavities by turbulence [2]. This tortuous path increases the friction imposed on the flow and greatly reduces leakage when compared to an annular slit. Schematics of different seal configurations are presented in Figure 1.1. In the case of the secondary air system, the leakage flow across the seal is put to use by metering air to different parts of the engine. They are placed at various locations, such as the compressor discharge, with the aim of providing cooling air to components or to pressurize bearing chambers. But in general, the cumulative effect of seal leakage (in the primary flow path) has an important impact on the engine thrust, and especially, its overall efficiency [1].

Despite this, labyrinth seals have several attractive features: they are easy to manufacture, have good system adaptability with minor impact on rotor dynamics, they are not restricted by any pressure limitation and can operate at high temperature [13]. Labyrinth seal integrity is essential, as mechanical failures result in hot gas ingestion which may degrade material properties. Also, inadequate sealing of bearing chambers allows oil to enter hot parts of the engine and potentially ignite, releasing debris which could damage other critical parts of the engine.

1.2 Flutter in Turbomachinery

Aeroelasticity is defined as the study of inertial, elastic and aerodynamic force interactions acting on a body exposed to an airstream, and the impact of these interactions on the body design [28]. The subject is of particular concern in axial-flow gas turbines, especially for turbomachinery blades which has led to extensive research effort.

A classical example used to introduce the subject is blade static aeroelasticity: the deflection that a turbine or compressor blade undergoes while in operation, due to pressure load and centrifugal forces, needs to be taken into account at design stage, as the aerodynamic performance of a blade depends on its geometry. Another aeroelastic phenomenon, which causes the blades to vibrate, is Forced Response. These vibrations are induced by aerodynamic forcing initiated from a flow non-uniformity. These flow non-uniformities can be caused by the interaction with upstream (wake-rotor) or downstream (potential-rotor) blade rows, structural support with different speeds, and sometimes non-uniform flow conditions coming from the combustion exit. Flutter also causes blade vibrations, but in this case the aerodynamic forces which cause the structure to vibrate are generated by the vibrations themselves. That is, any momentary perturbation creating a vibration, instead of being inhibited by the mechanical damping of the body, is sustained through the surrounding flow unsteady forces generated by the vibration itself. Such vibration can lead to blade failure by a high-cycle fatigue mechanism, and so it is imperative that aeroelastic phenomena be considered during design.

Although not as well documented, labyrinth seals are also prone to flutter. In the '60s, Alford [5] reported cracks on the stator part of a labyrinth seal located at the compressor discharge. At the same location, Lewis et al. [54] discuss a seal which had met the
Figure 1.1: Labyrinth seal schematics: a) Stepped Labyrinth seal and surrounding HP and LP cavities. b) Stepped labyrinth seal (enlarged view). c) Straight-through labyrinth seal. Different labyrinth seal configurations exist, however in aero-engines, stepped and straight-through seal geometries prevail.
requirements during testing but displayed evidence of fatigue failure after several operating hours. Cracks were located in the fins radial direction, some of which had propagated to the base of the component before growing in the axial direction.

When compared to the lifetime of an engine, these cracks occur prematurely. Besides maintenance considerations and the risk of failure, seal flutter is an issue as it has repercussions on the engine’s performance: the seal stator is typically covered with an abradable material or a thin material honeycomb to account for eventual rotor deflections, flutter increases the occurrence of rotor-stator rubs which in turn increase the effective clearance as the abradable material is gouged out. For a large diameter seal, this change in clearance can increase the leakage flow loss up to $\approx 0.5\%$ of the engine mass flow [3].

From these observations, a series of experimental tests followed, in some cases corroborated with bulk flow or analytical models, to determine the parameters involved in the aeromechanical stability of the seal. Alford [5] showed that the side of the seal support affected the stability behaviour and concluded that low-pressure side supported seals are aeromechanically stable. Abbot [3] furthered the study and showed that in addition to the support side, the ratio between the natural frequency of the seal and the acoustic frequency in the inter-fin cavities must be taken into account: a high-pressure side supported seal is stable if the mechanical to acoustic frequency ratio is less than 1 and inversely for a low-pressure side supported seal. A fix consists in increasing the mechanical damping of the seal either by friction damping [5] or by using sleeve dampers [54]. Using an analytical model, Ehrich [32] showed that tighter seal clearance favored instability, this was observed experimentally in [54]. Lewis et al. also determined that a lower pressure drop across the seal increased stability. Unfortunately, these stability trends cannot always be applied as they either go against the primary purpose of the seal (increasing the clearance increases leakage), or are dependant on the engine requirements (the pressure conditions are fixed). An extensive literature survey on the experimental procedures and analytical methods used to determine the parameters influencing seal flutter can be found in [73]. Yet in spite of a better understanding of these instability mechanisms, seal flutter remains an issue: in 2009, several labyrinth seals at the outlet of a space shuttle main engine turbine showed evidence of high-cycle fatigue [79].

### 1.3 Numerical Methods

The experimental studies carried out in the 60s set the foundation for work on seal instability and its influencing parameters. Today, engine manufacturers regularly carry out experimental flutter tests, they are however, subject to limitations as the condensed/rotating nature of the flow path excludes the use of many measurement devices. During rig testing, the seal inner volume is pressurized at values simulating engine conditions. Strain gauges are placed on the stator and rotor support, pressure probes are placed in the surrounding cavity and in some cases in the stator honeycomb cells to measure the pressure perturbations in the seal inner cavities [3]. The occurrence of flutter can then be detected by the strain gauges, and by analyzing their phase relationship, the nodal diameter number of
the travelling flexural wave can be determined. However, the information as to the level of instability (damping criteria) tends to be incomplete, and far from being able to provide a fully detailed aeroelastic response (complete vibrational pattern, flow description). Besides experimental difficulties, the financial costs associated with experimental testing and the increasingly shorter time-frames accorded for development, have led numerical methods to become an essential component in the design of aeroelastic performance.

Analytical methods are quick and useful for preliminary design. However, most of these methods require empirical data as corrective factors, and for this reason do not perform well for off-design conditions. They provide no or little information on the flow distribution, cannot handle different seal geometries, nor are they able to compare instability levels between different seals. Hirano et al. [44] showed that not taking into account the reverse flow in the upstream cavity resulted in an underprediction of the rotordynamic forces, demonstrating the need for the complete geometric information of the seal and cavities when modelling their aeroelastic behavior.

Computational Fluid Dynamics (CFD) methods have been used with success to calculate leakage flows in various seal designs [90, 76], and the effects of its interaction with the main engine flow [38, 7]. In a more recent example, Tyacke et al. [89] compared several Large Eddy Simulation (LES) methods on different labyrinth seal configurations.

1.3.1 Numerical Methods in Aeroelastic Problems

While CFD has been used to investigate the rotordynamic forces acting on a labyrinth seal [44, 77, 63], the literature concerning seal flutter computations is relatively scarce. Sayma et al. [84] applied an aeroelastic code until then used for turbomachinery blades to an air riding seal: the 3D nonlinear time-accurate Reynolds-Averaged Navier-Stokes (RANS) were used to obtain the unsteady flow field at every time-step. Using the unsteady solution, the aerodynamic forces at the given time-step acting on the seal were calculated. These forces could then in turn be used as boundary conditions for a Finite Element (FE) calculation to compute the updated seal position and velocity. By assuming the flow computational grid to act as a spring system, the vertices are displaced to accommodate for the updated wall boundaries. The unsteady flow field is then calculated for the following time-step on the updated grid. This is done recursively until the solution reaches a periodic state. di Mare et al. [30] applied a similar methodology to a labyrinth seal and its surrounding cavities. To reduce the computational work, the 2D computational grid was extruded in the circumferential direction over a 10° sector instead of the whole annulus. Periodic boundary conditions were prescribed at the circumferential domain extremities for the steady-state calculations. Phase-lagged boundary conditions were applied for the flutter computation for solution variables and nodal displacements. Flutter computations were carried out for backward and forward travelling waves with nodal diameters 1 to 5, the calculations matched the experimental observations. For the first time, aerodynamic work contributions per seal cavities were presented. Using the same code, Phibel [73] investigated the influence of different seal configurations (by varying seal clearance,
pivot location and seal pitch) and flow conditions (by varying pressure drop and swirl) on stability.

When referring to the classification terminology used by Marshal et al. in their review paper on aeroelastic methods [57], the fluid-structure seal cases discussed previously fall within the *integrated methods* class. That is, the interaction between the unsteady flow and the seal is modelled, and in the present cases, this is done by means of a boundary condition exchange between Finite Element program and time-accurate unsteady flow calculation. Such methods were originally applied to turbomachinery blades, so that nonlinearities in the flow response or in the structural deformations could be modelled. For example, Ayers et al. [8] showed that the shockwave position and strength had an important impact on occurrence of blade flutter, or to model the structural nonlinearities due to the presence of friction dampers at the blade root. However, the computational cost associated with such analyses, even today, precludes the coupled nonlinear time-accurate approach from being used in preliminary design stages.

At the other end of the spectrum, *classical methods* do not attempt to model the fluid-structure interaction: flow and structural problems are modelled independently of one another. The decoupling can be justified by considering the mass ratio between the considered body and its surrounding fluid, in the case of a high mass ratio, the aerodynamic perturbations caused by the vibration have negligible impact on the frequency or mode shape of the structure’s free vibration behaviour. Naturally, this is dependant on the application at hand, but in turbomachinery it is commonly acknowledged that for blade flutter the aerodynamic loads are a lot smaller than the blade inertial forces and stiffness [57] and hence are not able to cause structural modal coupling, i.e. the structural and aeroelastic mode can be considered to be identical. In an uncoupled aeroelastic method for flutter, a Finite Element calculation produces the frequencies and mode shapes of the structure vibrational pattern, these mode shapes are then incorporated in the flow computation. The aerodynamic work produced by the unsteady flow on the structure is then calculated: in the case of positive aerodynamic work, energy is fed to the structure, implying aeroelastic instability.

An alternative to solving the flow equations in a time-accurate manner, is to use the time-linearized harmonic method which is widely used in the turbomachinery field. The underlying idea consists of assuming a periodicity in the flow unsteadiness and treating it as a harmonic perturbation about the nonlinear steady-state background flow. The time-linearized method for the Euler equations was first introduced by Ni et al. [69], who used it to obtain unsteady perturbations for subsonic and transonic flow across a flat plate cascade. Since then, the time-linearized Euler equations have been applied to different turbomachinery unsteady flow problems [41, 45, 39], amongst them, Hall et al. [40] demonstrated that the linearity assumption holds to relatively large amounts of unsteadiness. A notable contribution was made by Lindquist et al. [55] who introduced the shock capturing scheme for the method, this was an important improvement to existing methods, as the shock motion influences turbine/compressor blade lift coefficients and the occurrence of fan flutter. Holmes et al. extended the methods to the viscous Navier-Stokes equations [46]. Concurrently, Clark [26] in his doctoral thesis applied the time-linearized
NS to solve flutter in airfoil cascades and forced response in turbine vane/Blades. Other turbomachinery cases using the time-linearized Navier-Stokes can be found in [34, 85, 21].

From a computational perspective, linearized solvers are efficient for flutter prediction, as the nonlinear unsteady problem is replaced by successively solving a steady-state and a linear problem. This can be done because 1) the frequency of the flow unsteadiness is known beforehand as it is prescribed by the output of the FE program, 2) the flow perturbations are small (because the blade vibration amplitudes are small) and are assumed to be within the linearity bounds. As a result, the computational cost is greatly reduced: nonlinear time-accurate computations can be an overwhelming task taking weeks of intensive parallel computing to pass the transients and observe a periodicity in the solution, whereas the non-time accuracy property inherent the steady-state problem permits the use of convergence acceleration techniques (more in section 1.3.2). Ni et al. [69] showed that the linear problem could be solved in a similar manner as the steady-state problem by introducing a pseudo time-step in the 1st order unsteady equations.

Another important feature of time-linearized methods is their ability to reduce the size of the computational domain. For a blade row, this is done by means of Lane’s assumption [53] which consists in treating all the blades in the annulus to vibrate with the same frequency but with different phase shift between one another. This phase shift is known as the Inter-Blade Phase Angle (IBPA). By assuming the unsteady flow to be periodic in the annulus circumferential direction, the domain can be reduced to a single blade passage. The IBPA can then be used to reconstruct the solution for all passages.

In summary, time-accurate integrated methods can capture the full nonlinear features of the flow and aeroelastic response, the computational constraints associated with such methods makes their usage often reserved to intricate problems. Classical methods where the flow is solved using the time-linearized frequency method allow accurate flutter predictions and flow behaviour at computational costs permitting parametric studies. Hence, the two approaches can be regarded as being complimentary.

### 1.3.2 Iterative Methods for Nonlinear Steady-State Problems

The linear harmonic method requires a nonlinear background flow about which the perturbation can be linearized. This is obtained by solving the steady-state compressible RANS equations. After discretization of the domain, a large system nonlinear algebraic equations is obtained. These systems are stiff because of the intricate physics of the problems at hand such as for example: turbulence, boundary layer/shockwave interaction, or in the current cases low-Mach number regions. Sayma et al. [84] highlighted convergence issues in his steady-state calculations of an air-riding seal and suggested the use of a preconditioning strategy to handle the low Mach number flow areas. Similar complications were encountered by Moore [63] in his study of labyrinth seal rotordynamics. Phibel et al. [74] noticed that these same low Mach number region hindered the convergence of the linear systems arising at every time-accurate step in his flutter calculations. Figure 1.2 shows the flow Mach number distribution of a stepped labyrinth seal for typical operating
conditions, the flow domain, and in particular the cavities, are dominated by a low-Mach number distribution. Transonic flow occurs at the fin tips and in the regions where the jets are discharged. Sonic conditions are reached at the exit fin clearance (figure 1.2b) for which the flow is choked. The disparity in the flow regime justifies the use of the compressible equations. Historically, the solution of nearly incompressible flow does not fare well with compressible solvers [94], as the time or length scale variations between the fluid velocity and the speed of sound are difficult to handle for the code. In the case of viscous low-speed flows, the diffusion terms present in the Navier-Stokes usually further exacerbate this behaviour [24].

Parallel computing advances have allowed fluid dynamicists to run larger computational grids, with evergrowing numerical schemes and physics models able to accurately capture flow features. However, solution procedures will be extremely inefficient without considering the interaction between these hardware advances with the mathematical structure of the problem and the algorithmic aspects of the numerical solver. This led to a collaboration between the CFD and applied-mathematics communities, in the aim of developing convergence acceleration methods, usually based on iterative procedures, for these discretized sets of equations [92, 59].

Amongst these, the Newton-GMRES (Generalized Minimal RESidual) method is widely used. It consists of a Newton method to artificially time march the nonlinear system to convergence by partially solving successive linear systems using Saad’s GMRES [83]. GMRES belongs to the Krylov subspace family, these are projection methods originally used to solve linear systems, however, they gained in popularity when being used in conjunction with Newton’s methods. The resulting Newton-Krylov (NK) methods have proven effective at solving highly ill-conditioned systems with severe nonlinearities such as low-Mach number combustion problems [50], they seem to be insensitive to the computational grid
properties in particular high element aspect-ratios [75], and are the method of choice when deep levels of convergence are needed. An informative review paper on NK methods was published by Knoll [48].

Besides GMRES, several iterative methods fall within the Krylov category, one of the first attributes is the method’s applicability to non-symmetric systems, such as the ones obtained from the discretization of the NS equations. Within this branch, a popular choice is the Bi-Conjugate Gradient STABlized (Bi-CGSTAB) method in which the Krylov subspace is built using the Lanczos algorithm. One of Bi-CGSTAB advantages is its efficient memory usage, however, the method lacks robustness when being used for the subiterations of a Newton method, the monotonic convergence of the linear residuals not being guaranteed. A technical discussion on Bi-CGSTAB and other Krylov methods can be found in [86], but in general the Arnoldi-based GMRES method is considered more robust than its counterparts [36, 4, 62]. Some features of GMRES make it an ideal linear solver for Newton’s method, amongst these: the minimization of the linear residual norm property which provides an “optimal” correction for the nonlinear solution update, the non-increase in linear residual norm confers robustness to the overall nonlinear method, the flexibility in different parameters which facilitates the use of globalization strategies.

For these reasons, Newton-GMRES has been used in various physics-based problems requiring the solution of nonlinear sets of equations, these include plasma physics, blade shape optimization, geophysical flows. In CFD, one of the first Newton-GMRES implementations is believed to have been carried out by Wigton et al. [96] to model the 2D Euler equations for transonic flow over airfoils. Since then, the algorithm has been used numerous times to compute steady-state compressible flow over NACA airfoils: it has been applied to solve the inviscid Euler [70, 9, 68] and viscous NS [75, 35, 36, 23] equations. Newton-GMRES is less common in the solution of steady-state turbomachinery flows, an example can be found in [12] where Bonfiglioni et al. use the method to calculate subsonic and transonic viscous flow through a turbine rotor cascade.

The convergence of the linear systems is heavily dependant on the eigenvalue spectrum of the coefficient matrix, in the case of the Newton method, this is the Jacobian obtained from the linearization of the nonlinear function defining the problem at a given time step. The Jacobians obtained from the NS equations are typically ill-conditioned, to improve the convergence behaviour it is common to transform these linear systems into better conditioned ones, with improved clustering of the Jacobian eigenvalues. This step is called preconditioning and is an essential component of any GMRES implementation. This is especially true in the nonlinear Newton context, where the computational time spent solving the linear systems accounts for most of the calculation’s overall CPU-time.

A preconditioner should be a better conditioned and computationally inexpensive approximation to the Jacobian inverse. A straight forward way is to use a lower order representation of the Jacobian matrix (i.e. the diagonal entries) and inversely apply it to the candidate system Right-Hand-Side (RHS). In a more elaborate fashion, an Incomplete LU (ILU(m)) factorization of the Jacobian can be performed, replacing the Jacobian in-

\[1\] U and L are upper and lower triangular matrices
version by successively solving two triangular systems. A full factorization would result in the exact inversion of the system, hence the term incomplete where the degree of approximation is controlled by the fill-in level $m$. In ILU(0), the factors $L$ and $U$ have identical non-zero entries to the Jacobian matrix, increasing $m$ increases the quality of the preconditioning, but also increases storage and computational costs (because of the application of the factors which are not longer sparse). This balance is typically case dependant, examples of ILU($m$) preconditioning can be found in [68, 35, 68]. Preconditioning can also be seen as the application of an iterative method such as Jacobi or Gauss-Seidel, GMRES iterations can also be preconditioned using a Krylov subspace technique [81]. However, in this research multigrid preconditioning is of particular interest. Originally, multigrid methods were developed as acceleration techniques for classical iterative methods such as Jacobi or Gauss-Seidel. As the method proceeds, the errors associated with each step can be decomposed into high and low frequency components. While the high frequency errors are rapidly cancelled out by the solver, convergence is hindered by the presence of the low frequency errors. These errors scale with the element characteristic size, and so by transposing the problem to a coarser domain, the fine grid lower frequency errors become the coarse grid high frequency errors which are in turn rapidly cancelled out by the coarse grid smoothing operator. A correction is then interpolated back to the fine grid, which is smoothed again before repeating the procedure. This process can involve several layers, and is known as linear multigrid. Multigrid can also be applied directly as a solver for a nonlinear problem, where at each grid level a Newton iteration is executed. This procedure is known as the Full Approximation Scheme (FAS), The choice between the two methods for solving nonlinear problems essentially depends on the quality of the available Jacobian, an assessment on the performance of both methods can be found in [60]. An important consideration in multigrid methods is how the problem quantities are transferred between the different levels. In geometric multigrid (GMG), different computational grid layers are generated and the quantities to be transfered are interpolated based on the grid topologies. In algebraic multigrid (AMG), the transfer operators are built using the sparsity pattern of the coefficient matrix solely. AMG is advantageous as once it is implemented it can be used for many problems, GMG is conceptually easier but requires building the grid layers explicitly for different problem (even an automatic agglomeration procedure might require some readjustments). Only GMG will be considered in this work as is it more commonly used in CFD, but also as the cost in memory and application per GMRES cycle is cheaper than AMG.

Knoll et al. [52] were the first to use linear multigrid as a preconditioning strategy for a Newton-Krylov (NKMG) implementation with GMRES as the Krylov method. The motivation for their work was based on the observation that the quality of ILU preconditioning was found to degrade with increasing grid sizes. They solved the 2D incompressible NS for a lid-driven cavity problem using linear multigrid preconditioning with Gauss-Seidel smoothing iterations and found that it outperformed ILU(0). Geuzaine et al. [37] used multigrid to precondition a Newton-GMRES method and solved the NS equations over a NACA 0012 airfoil for different flow regimes on unstructured grids. The NS equations were coupled with either the 1-equation Spallart-Almaras turbulence model. ILU(0) was used as the smoother for the multigrid levels which was found to outperform single grid
ILU(0) in all cases. The solver was compared against the FAS, NKMG was competitive with the FAS for the transonic cases, but showed better performance for the subsonic cases. Since then, NKMG has been applied to several other physics based problems by Knoll et al. in [78, 20, 49], and by other authors to solve the incompressible NS equations [88, 72, 51].

Authors have tried ranking multigrid performance when being used as a solver or as a preconditioner for a NK method. In [60], Mavriplis shows that the performance of multigrid is limited by the approximations made in the linearization of the nonlinear function. This is usually the case for the NS equations, where the convection operator is built using a second-order space reconstruction, and the Jacobian is evaluated using a first order approximation for memory and cost purposes. Mavriplis circumvents the issue by using GMRES and computing the application of the Jacobian on a vector using finite differences between second-order perturbed and unperturbed flux residual. The benefits are demonstrated on several case including a subsonic NACA airfoil, the multigrid smoother in the NK preconditioner uses a 1st order approximation which did not affect convergences rates. Osterlee et al. [71] compared the two MG implementation on a rotating convection-diffusion problem and noted that MG as a preconditioner is more robust as the nonlinear convergence is less affected by MG parameter changes in this configuration. In [58], Mavriplis discusses how the FAS performance degrades for viscous calculations: the prism elements needed to capture boundary layers and wakes increase the stiffness of the system. Mavriplis showed that nonlinear convergence was less affected by the anisotropic properties of the mesh when multigrid was used in a preconditioning configuration for a NK method. An important observation was made by Knoll et al. [52], who noticed that nonlinear convergence was less affected by the quality of the intergrid transfer operators in NKMG than in FAS or linear multigrid. This is of specific concern for unstructured grids: in an ideal structured case, a coarse multigrid level can be built by omitting every second element vertex in all coordinate directions $n_x$ resulting in a $2^n_x$ factor decrease in grid resolution. As the elements between successive grid levels are nested, the problem quantities can be easily be interpolated. For the unstructured case, a coarsening strategy needs to be implemented, this can take the form an element agglomeration procedure (collapsing element faces), an area/volume based weighted interpolation method, or a mesh re-triangulation based on a set of selected vertices from the fine level. While it is possible to implement such strategies, it is difficult to obtain the same nested element property as in the structured case, needed to obtain optimal multigrid convergence behaviour, making NKMG an attractive alternative to FAS or linear MG for highly unstructured cases. The bottom line is that FAS and linear multigrid can be highly efficient as solvers, even more so than a NK method in the case of smooth Partial Differential Equation problems with well defined transfer operators and where an exact Jacobian is available. However, based on the literature, using linear multigrid as a preconditioner to a NK method is likely to be more adequate for solving viscous subsonic NS problems on unstructured grids.
1.3.3 Time-Linearized Solution Procedures

The steady-state solution can in turn be used to build the linearized unsteady equations. Ni et al. [69], showed that by adding a pseudo time-step to the equations, the unsteady perturbations could be marched to convergence by using preexisting steady-state solver. Time-accuracy not being conserved, the resulting scheme can also benefit from convergence acceleration techniques, and so authors have used the solution procedure used in their steady-state code to solve the linear unsteady problem: Clark et al. [27] use a explicit Lax-Wendroff scheme with local time-stepping and multigrid, Marshall et al. [56] use a four stage explicit Runge-Kutta scheme to solve the linearized problem for force response in turbine stage and fan flutter. Sbardella et al. [85] use an backward implicit differencing scheme to solve the 3D linearized NS equations.

Campobasso et al. [18] report cases in which the residuals of the linearized flow computations diverge exponentially due to the presence of instabilities in the steady-state solution. The original code used a Runge-Kutta scheme preconditioned with multigrid and Jacobi smoothing iterations. Campobasso et al. showed that in the case of a non-convergence of the base flow, the use of GMRES with the time-marching procedure and using the multigrid iterations for preconditioning the Krylov vectors, could stabilize the linear harmonic code when coupled with the Recursion Projection Method (RPM). Chassaing et al. [22] also use GMRES to solve the time-linearized problem but without the additional time-marching term. Chassaing et al. computed the unsteady flow for a 2D transonic duct with far-field disturbances and showed that when a large Krylov basis was used, the convergence rates obtained without pseudo time-marching were superior to when including them. In the case were finer computational grids were used, because of large flow separation regions, the instabilities present in the mean flow were found to grow within the pseudo-time solution approach, eventually resulting in the stagnation of the linear residuals. Chassaing et al. found that by omitting the pseudo-time term, the solution procedure was stabilized and the code could converge.

At every GMRES step, a vector of the size of the solution is stored. Because of memory (and performance) considerations, after a user-determined maximum subspace length $m$, GMRES can be restarted. This is done by computing the linear solution at the end of a cycle and setting the residual as the initial vector for the following cycle, this is known as GMRES$(m)$. In the Newton-GMRES context, where the solution of the linearized system is used to pseudo time-march the nonlinear problem, GMRES$(m)$ is generally not used, therefore sacrificing nonlinear convergence rates for CPU-time (at least in the vicinity of the solution). However, in the time-linearized context, for large problems where the solution can not be obtained from the computation of a single Krylov subspace, GMRES$(m)$ will need to be used. As aforementioned, the convergence rates obtained with GMRES are highly dependent on the distribution of the linear system coefficient matrix eigenvalues, and in particular, the ones closest to the origin. As the method proceeds, the eigenvectors associated with these eigenvalues are eventually removed, by means of the Arnoldi method, from the spectrum of the iteration matrix. As GMRES$(m)$ is restarted with only the residual vector at the current step, the information built by the previous cycle Arnoldi procedure is lost. At best, the following cycle will need to recompute an
initial basis before new eigenvectors are actually removed, resulting in a performance
decrease when compared to the convergence rates which could have been obtained with
full GMRES. At worst, the spectral information is not recovered and the iterative solution
procedure will stagnate.

A natural solution is to increase the size of the Krylov subspace, which as previously men-
tioned is not always feasible due to memory constraints. This motivated applied mathe-
maticians to develop methods which somehow incorporate part of the spectral information
accumulated during a GMRES cycle into the following. This led to several GMRES vari-
ants, the most notable contributions of which were made by Morgan [64, 65, 66]. These
methods consist in solving an eigenvalue problem defined by the reduced size problem
obtained with the Arnoldi relation, and appending a set of $k < m$ extracted eigenvectors
to the end of the next GMRES cycle (GMRES-E), or at the beginning (GMRES-IR).
Morgan later developed, a more robust version of GMRES-IR known as GMRES with
deflated restart (GMRES-DR). A comparison was made between the different methods on
ill-conditioned test problems obtained from matrix collections. One of the main results
was the ability of GMRES-DR to reproduce similar convergence rates to classical GM-
RES. This comparison could be made as the linear systems were relatively small, and the
termination criterion was $10^{-6}$ which allowed the solution to be obtained in one GMRES
cycle. To the best of found knowledge, GMRES-DR has not been applied to solve the
time-linearized equations in turbomachinery problems.
1.4 Thesis Objectives

Due to their attractive mechanical features, labyrinth seals are unlikely to be replaced in the foreseeable future. Several authors have investigated their instability mechanisms, whether experimentally or numerically, yet seal flutter remains an ongoing issue in the aeroengine industry. Numerical methods are essential to the design of aeroelastic performance and flutter prediction, but although time-accurate integrated methods are effective in capturing accurately complex nonlinear phenomena, they remain far from being used as a routine tool in industry. From the review of the literature, the computationally cheaper uncoupled time-linearized frequency domain method has not been applied to labyrinth seals, despite the applicability of such methods to cavity flow being more compatible with the methods assumptions than in the case of turbomachinery blades (absence of shockwaves, subsonic low-Mach number surrounding flow, stiff structure). It has been established that neglecting the surrounding cavities alters the results of the aeroelastic analysis, and hence, their representation in the computational domain is needed. Due to a dominant low Mach number distribution in the flow domain, previous authors have encountered convergence issue in their steady-state computations, it is reported that this is likely to alter the quality of flutter predictions. Newton-GMRES methods have proven to be effective in a range of stiff nonlinear problems, this includes complex steady-state CFD problems. However, Newton-GMRES remains uncommon in steady-state turbomachinery problems. GMRES has also been used with or without pseudo-time marching to solve the time-linearized equations. It has been seen, that the performance of GMRES(m) for large linear problems can be improved by adding spectral information extracted from previous cycles.

The research aims to provide numerical methods able to efficiently predict seal flutter and cavity solutions for different geometrical configurations and operating conditions. To that end, the following objectives were set:

1. Develop a time-linearized harmonic method for labyrinth seal flutter.

2. From an algorithmic perspective, accelerate with respect to existing best practices, the convergence rates to solution in cavity problems by:

   (a) Implementing a Newton-GMRES methodology to solve the steady-state solutions for labyrinth seals and their surrounding cavities.

   (b) Implementing a GMRES strategy for solving the unsteady time-linearized problem.

AU3X was developed at Imperial College London by Dr. L. di Mare and collaborators [31, 95, 19], this code will be used as the platform for implementing the methods set out in the objectives. The code handles structured and unstructured grids in 2D/3D. The unsteady RANS equations are solved using a cell-centered finite volume scheme. The solutions procedure is fully implicit: the steady-state equations are solved using Newton’s method with Jacobi subiterations. A FAS implementation is available for structured cases with nested element between successive grids. Different methods can be used to
obtain the inviscid numerical fluxes: Roe fluxes with vector difference splitting, Van Leer flux vectors, or different implementations of the Advection Upstream Splitting Method (AUSM). Second-order accuracy in space is obtained using the Monotonic Upstream-Centered Scheme for Conservation Laws (MUSCL), the flow gradients are evaluated using the weighted least square approach, and spurious oscillations are countered using the van Albada flux limiter. Turbulence closure is obtained by means of the: Cebeci-Smith, Spalart-Allmaras, $k - \epsilon$ or $k - \omega$ models.

As a subsidiary outcome, future AU3X users will benefit from the convergence acceleration techniques developed in this research.

1.5 Organisation

Chapter 2 presents the governing equation for deforming control volumes. It will be seen that by using the harmonic perturbation assumption, the unsteady flow can be obtained by successively solving the steady-state nonlinear and linearized flow problems. That section will also demonstrate that by assuming the unsteadiness to be space-periodic in the circumferential direction, the domain size can be reduced. In Chapter 3, the solution procedures for solving nonlinear and linear problems will be presented. Chapter 4 concerns some more technical aspects relevant to CFD and harmonic grid motion. The steady-state and linearized flow results are presented in Chapters 5 and 6 respectively. Both result chapters treat the subjects of numerical performance and flow analysis in two distinct parts. The conclusions of the thesis are presented in Chapter 7.
Chapter 2

Time-Linearized Harmonic Method about Axisymmetric Background Flows

2.1 Introduction

By assuming a time periodicity in the flow unsteadiness and the perturbations magnitudes to be small, the unsteady flow induced by the vibrating seal can be linearized about a nonlinear base flow. In a similar analysis to Clark [26] and Clark et al. [27], but in the cylindrical coordinate system, this section will demonstrate that by starting from the unsteady RANS equations for deforming control volumes, and by decomposing the flow variables and computational grid coordinates into mean and harmonically varying components, that the nonlinear base flow is in fact obtained by solving the steady-state problem. The solution to the linearized flow problem is then obtained by solving a large complex linear system with coefficients built using the steady-state solution.

Due to the low mass ratio parameter of the problem at hand, the aeroelastic and structural modes are treated as identical. Hence, for a given mode shape nodal diameter number, the flow unsteadiness exhibits the same cyclic symmetry as the seal flexural wave in the circumferential direction. It will be seen that this circumferential space-periodicity, in the frequency domain because of the use of complex algebra, enables the computational domain to be reduced to 2D. As a result, the nonlinear base flow about which the perturbations are linearized can also be 2D, this allows the steady-state problem to be solved in an axisymmetric manner, greatly contributing to reducing the overall computation cost.
2.2 Governing Equations for Deforming Control Volumes

Considering a control volume $V$ undergoing deformation in the $(z,r)$ plane and rotating about the $z$ axis with angular velocity $\Omega$, the three-dimensional nonlinear Navier-Stokes equations in their integral form can be expressed in the cylindrical coordinate system $(z,r,\theta)$. Mass conservation is defined as:

$$
\frac{d}{dt} \iiint_V \rho r \, d\theta \, dz + \oint_{\partial V} \left( \rho u_z - \rho \frac{\partial f}{\partial t} \right) \, r \, d\theta \, dr + 
\oint_{\partial V} \left( \rho u_r - \rho \frac{\partial g}{\partial t} \right) \, r \, d\theta \, dz + \oint_{\partial V} \rho u_\theta \, r \, dr \, dz = 0
$$

(2.1)

Where $f$ and $g$ account for the control volume displacements in the $z$ and $r$ direction. Momentum conservation in the axial, radial, and circumferential directions hold:

$$
\frac{d}{dt} \iiint_V \rho u_z \, r \, d\theta \, dz + \oint_{\partial V} \left( \rho u_z^2 + p - \tau_{zz} - \rho u_z \frac{\partial f}{\partial t} \right) \, r \, d\theta \, dr + 
\oint_{\partial V} \left( \rho u_z u_r - \tau_{rz} - \rho u_z \frac{\partial g}{\partial t} \right) \, r \, d\theta \, dz + \oint_{\partial V} \left( \rho u_z u_\theta - \tau_{\theta z} \right) \, dr \, dz = 0
$$

(2.2)
\[
\frac{d}{dt} \iiint_V \rho u_r \, r \, d\theta \, dr \, dz + \oint_{\partial V} \left( \rho u_r u_z - \tau_{rz} - \rho u_r \frac{\partial f}{\partial t} \right) r \, d\theta \, dr + \\
\oint_{\partial V} \left( \rho u_r^2 + p - \tau_{rr} - \rho u_r \frac{\partial g}{\partial t} \right) r \, d\theta \, dz + \oint_{\partial V} \left( \rho u_r u_z - \tau_{zr} \right) r \, d\theta \, dz \\
= \iiint_V \rho \Omega u_\theta \, r \, d\theta \, dr \, dz \\
\frac{d}{dt} \iiint_V \rho u_\theta \, r \, d\theta \, dr \, dz + \oint_{\partial V} \left( \rho u_\theta u_z - \tau_{z\theta} - \rho u_\theta \frac{\partial f}{\partial t} \right) r \, d\theta \, dr + \\
\oint_{\partial V} \left( \rho u_\theta u_r - \tau_{r\theta} - \rho u_\theta \frac{\partial g}{\partial t} \right) r \, d\theta \, dz + \oint_{\partial V} \left( \rho u_\theta^2 + p - \tau_{r\theta} \right) r \, d\theta \, dz \\
= - \iiint_V \rho \Omega u_r \, r \, d\theta \, dr \, dz
\] (2.3)

Finally, energy conservation is given as:

\[
\frac{d}{dt} \iiint_V \rho e r \, d\theta \, dr \, dz + \oint_{\partial V} \left( \rho u_z h - \tau_{zh} - \rho e \frac{\partial f}{\partial t} \right) r \, d\theta \, dr + \\
\oint_{\partial V} \left( \rho u_r h - \tau_{rh} - \rho e \frac{\partial g}{\partial t} \right) r \, d\theta \, dz + \oint_{\partial V} \left( \rho u_\theta h - \tau_{\theta h} \right) r \, d\theta \, dz = 0
\] (2.4)

Where static pressure \( p \), specific internal energy \( e \) and enthalpy \( h \) for an ideal gas are closed by the following relations:

\[
h = e + \frac{p}{\rho} \tag{2.6}
\]

\[
\frac{p}{\rho} = (\gamma - 1) \left[ e - \frac{1}{2} (u_z^2 + u_r^2 + u_\theta^2) \right] \tag{2.7}
\]

The specific heat ratio is taken to be \( \gamma = 1.4 \). The viscous stress terms are expressed as a function of strain rates and viscosity:

\[
\tau_{r\theta} = \tau_{\theta r} = \mu \left[ r \frac{\partial}{\partial r} \left( \frac{u_\theta}{r} \right) + \frac{1}{r} \frac{\partial u_r}{\partial \theta} \right] \quad \tau_{zz} = 2\mu \left[ \frac{\partial u_z}{\partial z} \right] + \frac{2}{3} \mu \nabla \cdot \mathbf{u}
\]

\[
\tau_{rz} = \tau_{zr} = \mu \left[ \frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \right] \quad \tau_{rr} = 2\mu \left[ \frac{\partial u_r}{\partial r} \right] + \frac{2}{3} \mu \nabla \cdot \mathbf{u}
\]

\[
\tau_{z\theta} = \tau_{\theta z} = \mu \left[ \frac{1}{r} \frac{\partial u_z}{\partial \theta} + \frac{\partial u_\theta}{\partial z} \right] \quad \tau_{\theta\theta} = 2\mu \left[ \frac{1}{r} \frac{\partial u_\theta}{\partial \theta} + \frac{u_\theta}{r} \right] + \frac{2}{3} \mu \nabla \cdot \mathbf{u}
\]

With:

\[
\nabla \cdot \mathbf{u} = \frac{1}{r} \frac{\partial}{\partial r} (ru_r) + \frac{1}{r} \frac{\partial u_\theta}{\partial \theta} + \frac{\partial u_z}{\partial z} \tag{2.8}
\]
The viscous work and conduction terms are given as:

\[
\begin{align*}
\tau_{zh} &= \kappa \frac{\partial T}{\partial z} + u_z \tau_{zz} + u_r \tau_{rz} + u_\theta \tau_{\theta z} \\
\tau_{rh} &= \kappa \frac{\partial T}{\partial r} + u_z \tau_{zr} + u_r \tau_{rr} + u_\theta \tau_{r\theta} \\
\tau_{\theta h} &= \kappa \frac{\partial T}{\partial \theta} + u_z \tau_{z\theta} + u_r \tau_{r\theta} + u_\theta \tau_{\theta\theta}
\end{align*}
\] (2.9)

The total viscosity \( \mu = \mu_l + \mu_t \) is obtained from the molecular \( \mu_l \) and turbulent \( \mu_t \) viscosities. Sutherland’s law is used to evaluate \( \mu_l \), the turbulent contribution \( \mu_t \) is computed using the Cebeci-Smith algebraic model [87], no flow separation regions are present in cavity-seal domains, this motivated the use of the 0-equation model to avoid introducing additional stiffness in the solution of the discretized equations. The thermal conductivity coefficient \( \kappa \) is expressed as:

\[
\kappa = \frac{\gamma}{\gamma - 1} \left( \frac{\mu_l}{Pr_l} + \frac{\mu_t}{Pr_t} \right)
\] (2.10)

Where the laminar and turbulent Prandtl number are taken to be \( Pr_l = 0.7 \) and \( Pr_t = 0.9 \). The conservation laws are conveniently written in the following vector form:

\[
\frac{d}{dt} \iiint_V U \, dV + \oint_{\partial V} \left( F - U \frac{\partial f}{\partial t}, G - U \frac{\partial g}{\partial t}, H \right) \cdot dA = \iiint_V S \, dV
\] (2.11)

About the solution vector of conservative variables \( U \):

\[
U = \begin{bmatrix}
\rho \\
\rho u_z \\
\rho u_r \\
\rho u_\theta \\
\rho e
\end{bmatrix}
\] (2.12)

Where \( dA \) is the outward pointing vector normal to the control volume surface, with surface area magnitude. The fluxes in the three cylindrical coordinate direction \( F, G, H \) are:

\[
F = \begin{bmatrix}
\rho u_z \\
\rho u_z^2 + p - \tau_{zz} \\
\rho u_z u_r - \tau_{rz} \\
\rho u_z u_\theta - \tau_{\theta z} \\
\rho u_z - \tau_{hz} h
\end{bmatrix} \quad G = \begin{bmatrix}
\rho u_r \\
\rho u_r u_z - \tau_{rz} \\
\rho u_r^2 + p - \tau_{rr} \\
\rho u_r u_\theta - \tau_{r\theta} \\
\rho u_r h - \tau_{hr}
\end{bmatrix} \quad H = \begin{bmatrix}
\rho u_\theta \\
\rho u_\theta u_z - \tau_{\theta z} \\
\rho u_\theta u_r - \tau_{r\theta} \\
\rho u_\theta^2 + p - \tau_{\theta\theta} \\
\rho u_\theta h - \tau_{h\theta}
\end{bmatrix}
\] (2.13)
The contributions from the rotational frame are accounted for by the following source term:

\[
S = \begin{bmatrix}
0 \\
0 \\
\rho \Omega u_\theta \\
-\rho \Omega u_r \\
0
\end{bmatrix}
\]  \hspace{1cm} (2.14)

## 2.3 Harmonic Assumption

The previous section described the nonlinear unsteady viscous flow. Linearized flow solvers rely on the assumption that the flow unsteadiness can be treated as a small perturbation about a background base flow. In the case of the linear harmonic method, the background flow is taken to be the nonlinear steady-state solution and the linear perturbation is periodic in time. As a result of the periodicity assumption, the perturbation can then be decomposed into a sum of harmonics.

The two possible excitation sources in time-linearized flow problems are far-field incoming disturbances at the boundaries and/or structural motion. In the present case, only the latter is considered. The flow domain is first discretized, the structural deformation of the boundaries is then constrained to the interior nodes defining the system excitation source. The decomposition of the nodal coordinates into mean and time varying components gives:

\[
\begin{align*}
\bar{z}(\theta, t) & = \bar{z} + f(\theta, t) \\
\bar{r}(\theta, t) & = \bar{r} + g(\theta, t) \\
\bar{\theta}(t) & = \bar{\theta}
\end{align*}
\]  \hspace{1cm} (2.15)

Where the overbar denotes the mean nodal position, \( f \) and \( g \) are smooth continuous function which account for the nodal perturbation in the \( z-r \) directions. The \( \theta \) component has no time-varying contribution, as the grid deformation occurs solely in the \( z-r \) plane. However, for the grid motion to represent a given mode shape with its associated nodal diameter, the boundary walls need to be able to deflect in the circumferential direction. To that end, the values of \( f \) and \( g \) are made dependant on the azimuthal position. An illustration of the computational grid in the \((r, \theta)\) plane is provided in Figure 2.2a, with the perturbed grid resulting from the wall deflections in Figure 2.2b. Expanding the nodal perturbation functions into Fourier series gives:

\[
\begin{align*}
f(\theta, t) & = \Re \left[ \sum_{k=-\infty}^{\infty} \hat{f}_k e^{i(w_k t + k\theta)} \right] \\
g(\theta, t) & = \Re \left[ \sum_{k=-\infty}^{\infty} \hat{g}_k e^{i(w_k t + k\theta)} \right]
\end{align*}
\]  \hspace{1cm} (2.16)
Where \( \hat{f}_k \) and \( \hat{g}_k \) are the complex amplitudes defining the grid motion resulting from a \( k \)-nodal number diameter mode shape with associated excitation frequency \( w_k \). In uncoupled harmonic methods, the structural modes are studied individually:

\[
\begin{align*}
&f_k(\theta, t) = \Re \left[ \hat{f}_k e^{i(w_k t + k\theta)} \right] \\
&g_k(\theta, t) = \Re \left[ \hat{g}_k e^{i(w_k t + k\theta)} \right]
\end{align*}
\] (2.17)

In a similar approach, the conservative flow variables are decomposed as the following:

\[
\begin{align*}
&U(\theta, t) = \bar{U} + \Re \left[ \hat{u} e^{i(w_k t + k\theta)} \right] \\
&\bar{U} = \text{non-linear steady-state solution.}
\end{align*}
\] (2.18)

Here, \( \bar{U} \) is the non-linear steady-state solution. The flow unsteadiness is described by the magnitude and phase of the complex perturbation \( \hat{u} \). Using a Taylor series and introducing the arrays \( x = [z, r, \theta]^T \) and \( \hat{f} = [\hat{f}, \hat{g}, 0]^T \), the axial flux can be approximated as:

\[
F(U, x, \dot{x}) = \bar{F}(\bar{U}) + \frac{\partial \bar{F}}{\partial U} \hat{u} e^{i(w_k t + k\theta)} + \frac{\partial \bar{F}}{\partial x} \hat{f} e^{i(w_k t + k\theta)} + i w_k \frac{\partial \bar{F}}{\partial \dot{x}} \hat{f} e^{i(w_k t + k\theta)}
\] (2.19)

Where \( \bar{F}(\bar{U}) \) and \( \frac{\partial \bar{F}}{\partial U} \) are respectively the steady-state flux and flux Jacobian, \( \frac{\partial \bar{F}}{\partial x} \) is the sensitivity of the flux to the nodal displacement \( \hat{f} \) and \( \frac{\partial \bar{F}}{\partial \dot{x}} \) its sensitivity to the nodal velocity \( \hat{f} \). The second RHS term of equation (2.19) accounts for the perturbation of the flux function due to the flow perturbation itself. The third and fourth term relate to the flux perturbations induced by the nodal motion. Differentiating with respect to time, the nodal velocity can be expressed in terms of nodal displacement:

\[
F(U, x, \dot{x}) = \bar{F}(\bar{U}) + \frac{\partial \bar{F}}{\partial U} \hat{u} e^{i(w_k t + k\theta)} + \frac{\partial \bar{F}}{\partial x} \hat{f} e^{i(w_k t + k\theta)} + i w_k \frac{\partial \bar{F}}{\partial \dot{x}} \hat{f} e^{i(w_k t + k\theta)}
\] (2.20)

Correspondingly, the radial flux is expressed as:

\[
G(U, x, \dot{x}) = \bar{G}(\bar{U}) + \frac{\partial \bar{G}}{\partial U} \hat{u} e^{i(w_k t + k\theta)} + \frac{\partial \bar{G}}{\partial x} \hat{f} e^{i(w_k t + k\theta)} + i w_k \frac{\partial \bar{G}}{\partial \dot{x}} \hat{f} e^{i(w_k t + k\theta)}
\] (2.21)

The circumferential flux and source term contributions are perturbed solely about the flow unsteadiness:

\[
\begin{align*}
H(U) &= \bar{H}(\bar{U}) + \frac{\partial \bar{H}}{\partial U} \hat{u} e^{i(w_k t + k\theta)} \\
S(U) &= \bar{S}(\bar{U}) + \frac{\partial \bar{S}}{\partial U} \hat{u} e^{i(w_k t + k\theta)}
\end{align*}
\] (2.22)
2.4 Linearized Governing Equations

Substituting flow and nodal coordinate decompositions into equation (2.11) and regrouping by the zeroth $e^{i(w_{k}t+k\theta)}$ power terms gives the steady-state equations:

$$\oint_{\partial V} (\vec{F}, \vec{G}, \vec{H}) \cdot d\vec{A} = \int\int\int_{V} \vec{S} dV$$ (2.23)

Which are independent of any perturbed quantities. Collection of the 1st order terms results in the linearized flow equations:

$$\int\int\int_{V} \left( iw_{k}I - \frac{\partial \vec{S}}{\partial \vec{U}} \right) \hat{\vec{u}} dV + \oint_{\partial V} \left( \frac{\partial \vec{F}}{\partial \vec{U}} \hat{\vec{u}}, \frac{\partial \vec{G}}{\partial \vec{U}} \hat{\vec{u}}, \frac{\partial \vec{H}}{\partial \vec{U}} \hat{\vec{u}} \right) \cdot d\vec{A}$$

$$= -iw \int\int\int_{V} \hat{\vec{U}} dV + iw \oint_{\partial V} \hat{\vec{U}} \hat{\vec{f}} \cdot d\vec{A} + \int\int\int_{V} \vec{S} dV$$ (2.24)

Which are a complex variable linear system about the perturbation $\hat{\vec{u}}$. All grid motion related quantities are regrouped into the forcing term of equation (2.24), and as a result are only need to be computed once prior to solving the system. The steady-state solution is heavily involved in the definition of the system, it intervenes in the LHS flux Jacobian and linearized sources terms as well as in the forcing term. The diagonal term $iw_{k}I$ ensures the coupling between real and imaginary parts of the solution.

2.5 Circumferential Contribution Approach

Applying the divergence theorem to equation (2.24), the linearized equations can be rewritten in their differential form:

$$\left( iw_{k}I - \frac{\partial \vec{S}}{\partial \vec{U}} \right) \hat{\vec{u}} + \frac{\partial}{\partial z} \left( \frac{\partial \vec{F}}{\partial \vec{U}} \hat{\vec{u}} \right) + \frac{\partial}{\partial r} \left( \frac{\partial \vec{G}}{\partial \vec{U}} \hat{\vec{u}} \right) + \frac{1}{r} \frac{\partial}{\partial \theta} \left( \frac{\partial \vec{H}}{\partial \vec{U}} \hat{\vec{u}} \right) = b$$ (2.25)

Where $b$ is the forcing term in differential form. The linearized flow equation were derived by regrouping the first order $e^{i(w_{k}t+k\theta)}$ power terms, thus the fourth term of equation (2.25) is rewritten as:

$$\frac{\partial}{\partial \theta} \left( \frac{\partial \vec{H}}{\partial \vec{U}} \hat{\vec{u}} e^{i(w_{k}t+k\theta)} \right) = i k \frac{\partial \vec{H}}{\partial \vec{U}} \hat{\vec{u}} e^{i(w_{k}t+k\theta)}$$ (2.26)
When substituted back into equation (2.25):

\[
(iwI \frac{\partial \bar{S}}{\partial \bar{U}} + \frac{ik}{r} \frac{\partial \bar{H}}{\partial \bar{U}}) \hat{u} + \frac{\partial}{\partial z} \left( \frac{\partial \bar{F}}{\partial \bar{U}} \hat{u} \right) + \frac{\partial}{\partial r} \left( \frac{\partial \bar{G}}{\partial \bar{U}} \hat{u} \right) = b
\]  

(2.27)

The circumferential contribution is now represented by a diagonal term, i.e. the flux evaluation will require no neighbouring cell (Figure 2.2d). The linearized system is solved on a 2D computational grid, allowing the computation of the steady-state background flow to take place in the axisymmetric frame:

\[
\oint_{\partial V} (\bar{F}, \bar{G}) \cdot d\bar{A} = \iiint_{V} \bar{S}dV
\]  

(2.28)

Where for a control volume \(i\), \(d\bar{A}_i = [2\pi r_i dr_i, 2\pi r_i dz_i]\).

2.6 Summary

Due to the circumferential cyclic symmetry of the seal nodal diameter, the time-linearized unsteady solution is obtained by successively solving the steady-state axisymmetric equations and a linear complex system. The linearized system is built using the steady-state quantities. All quantities relevant to the computational grid motion are present solely in the linear system forcing term, and hence, only need to be computed once. The following Chapter will describe the methods used to solve the nonlinear steady-steady and linearized unsteady flow equations.
Figure 2.2: Illustration of flow domain reduction to 2D axisymmetric computational grids using the harmonic assumption - \((r,\theta)\) plane. a) Original 3D computational grid. b) 3D harmonic grid (unperturbed vertices are shown for comparison. c) 3D harmonic grid - Enlarged view. d) 2D axisymmetric grid with linearized circumferential flux contributions acting as a diagonal term.
Chapter 3

Solution Procedure

3.1 Introduction

In Chapter 2, the influence of the steady-state solution in the definition of the unsteady flow equations (2.24) was demonstrated. The first part of this Chapter will describe the iterative methods used to overcome the numerical difficulties inherent to steady-state flow computations in cavities discussed in section 1.3.2. The motivation for developing the solver is twofold: 1) Computational efficiency is needed when carrying out parametric studies for industrial applications. 2) Solving the problem to a low degree of error will yield a more accurate definition of the unsteady linear system. In the second part, the methods used to solve the time-linearized problem will be covered.

After discretization\(^1\) of the steady-state governing equations (2.28), a nonlinear set of equations is obtained:

\[
\bar{R}(\bar{U}) = 0 \tag{3.1}
\]

\(R(U)\) is the nonlinear residual\(^2\), or nonlinear function, of the current state \(U\). Explicit methods have restrictive stability bounds and for stiff problems their performance degrades rapidly because of their poor convergence rates. Newton-type implicit methods are not limited by stability constraints and allow larger time steps: the solution will require far less nonlinear function evaluations than their explicit counterpart. The drawback is the increased cost per iteration, as each time step requires the solution of a large linear system. Equation (3.1) is linearized about a state \(n\):

\[
\frac{\partial R(U^n)}{\partial U} \delta u^n = -R(U^n) \tag{3.2}
\]

\(^1\)A description of the spatial discretization will be given in Chapter 4.

\(^2\)For clarity, the overbar will be omitted throughout this chapter and unless mentioned otherwise all variables are considered to be steady-state.

45
where $\frac{\partial R}{\partial U}$ is the Jacobian of the residuals with a sparsity pattern dependant on the grid discretization as well as the order of the reconstruction stencil. The linear system (3.2) is solved about the nonlinear update or correction $\delta u$:

$$U^{n+1} = U^n + \delta u^n$$  \hfill (3.3)

A Newton procedure starts from an initialized solution $U^0$ from which the nonlinear function and Jacobian at this state are computed, the linearized system is then solved for $\delta u^0$ from which the solution at state $n = 1$ is obtained. These steps are carried out repeatedly until equation (3.1) holds true. In practice the algorithm terminates when $||R(U^n)||_2 = \varepsilon$, where $\varepsilon$ a user-defined termination criterion. The NS equations are highly nonlinear, several linear systems of type (3.2) need to be solved for $N = N_{\text{nodes}} \times N_{\text{eqs}}$ unknowns, where $N_{\text{eqs}}$ is the number of state quantities stored at one grid point and $N_{\text{nodes}}$ is the number of grid points. In CFD applications $N$ is typically large, as the complexity of direct matrix inversion methods (i.e. Gaussian elimination) scale with $O(N^3)$, the need for fast linear solvers becomes quickly apparent. For this reason, it is custom to partially solve the linear system at every Newton step using an iterative method, greatly reducing the computing time per iteration. This procedure is referred to as the Inexact-Newton method and the degree to which the linear system is solved is controlled by the following linear convergence criterion:

$$||J^n \delta u^n + R(U^n)||_2 \leq \eta ||R(U^n)||_2$$  \hfill (3.4)

where the Jacobian has been rewritten as $J$. The LHS quantity of equation (3.4) is the Euclidian norm of the linear residual. At an inner iteration $k$, the linear residual is defined as: $r_k = J^n \delta u^n_k + R(U^n)$. $\eta \in [0, 1]$ is the forcing term. Condition (3.4) is hard to meet for the iterative solver, as these linearized systems are typically ill-conditioned. To relax the system, the Pseudo-transient Continuation scheme is introduced: for the steady-state problem the transients are not of interest, therefore a pseudo time-step can be added to equation (3.2):

$$\left( I + \frac{J^n}{\Delta t} \right) \delta u^n = -R(U^n)$$  \hfill (3.5)

increasing the diagonal dominance of the linear system. The linear convergence behaviour is improved as a result of better conditioning, however the nonlinear convergence of the Newton method deteriorates as the LHS no longer represents the true linearization of the nonlinear function.

Classical ways of iteratively solving linear systems are based on a matrix splitting approach which fall within the stationary methods taxonomy. Considering a linear system $Ax = b$.

---

3 Nonlinear iterations are referred to outer iterations. The iterations use to solve the linear system which arises at each newton step are referred to inner iteration.

4 A matrix $A^{n \times n}$ is diagonally dominant if $|a_{jj}| > \sum_{i=1,i \neq j}^{n} |a_{ij}|$, with $j = 1, \ldots, n$.
and taking $A = M + N$, the following iterative scheme can be defined:

$$x_{k+1} = M^{-1}(b - Nx_k)$$

(3.6)

Depending on the definition of $M$ and $N$, different well-know iterative schemes are obtained: for Jacobi iterations $M = \text{diag}(A)$, for Gauss-Seidel $M = L + \text{diag}(A)$ where $L$ is the lower triangular part of $A$. Unlike direct methods where the Jacobian needs to be built explicitly (i.e. pivoting), only the result of the application of (some parts of) the Jacobian on a vector is needed. This leads to the matrix-free scheme where the Jacobian-vector product is approximated using numerical fluxes, the entries needed at other steps in the linear procedure are stored in a separate array (e.g. $\text{diag}(A)$ for a Jacobi scheme).

Linear systems may also be solved using the more recent projection methods which are categorized as non-stationary methods. These approximate the solution to $Ax = b$ by extracting a solution $x_m$ from a search subspace $K$ of dimension $m \leq N$, while imposing $m$ constraints. One way of defining the constraints is to impose $m$ orthogonality conditions on the residual, the subspace containing the vectors fulfilling such condition is the constraints subspace $L$. The projection problem is then defined as:

$$\text{Find } x_m \in x_0 + K \text{ such that } r_m = b - Ax_m \perp L$$

(3.7)

where $x_0$ is an initial solution. Krylov methods are a projection method where the search subspace is taken to be:

$$K_m (A, r_0) = \text{span}(r_0, Ar_0, A^2r_0, \ldots, A^{m-1}r_0)$$

(3.8)

GMRES is a Krylov-type method, the main technical features of the Newton-GMRES method are covered in the following section. The important topic of preconditioning is discussed in section 3.3. As the efficiency of an implicit method is related to that of the linear solver, it would be interesting to recycle the solver for the unsteady flow linear problem, this is covered in section 3.4. Finally, deflation methods for GMRES are discussed.

### 3.2 Newton-GMRES Method

#### 3.2.1 GMRES: Overview

GMRES is a Arnoldi-based algorithm where the approximate solution $x_m \in x_0 + K_m$ is found such that the Euclidean norm of the resulting residual $r_m$ is minimized over the Krylov subspace. The basic version of GMRES can be summarized in four steps, namely: 1) initialization 2) building the Krylov basis 3) minimizing the residual norm 4) forming
the approximate solution.

The basis \( V_m = [v_0, \ldots, v_{m-1}] \in K_m \) is introduced. In step 1), the first vector of \( V \) is set to the normalized residual:

\[
v_0 = \frac{r_0}{\beta}
\]

where \( \beta = ||r_0||_2 \) (3.9)

When GMRES is used as the linear solver for a Newton method, the initial solution is taken to be \( x_0 = 0 \), and so \( r_0 \) is the evaluation of the nonlinear function at the current outer step. The Krylov base in (3.8) is rewritten as:

\[
K_m (A, v_0) = \text{span} (v_0, Av_0, A^2v_0, \ldots, A^{m-1}v_0)
\]

(3.10)

In step 2), the Arnoldi procedure is used to build the orthonormal base \( V \in K_m \):

1: for \( j = 0 \) to \( m - 1 \) do begin
2: \( v_{j+1} := Av_j \)
3: for \( i = 0 \) to \( j \) do begin
4: \( h_{ij} = v_{j+1} \cdot v_i \)
5: \( v_{j+1} := v_{j+1} - h_{ij}v_i \)
6: end
7: \( h_{j+1,j} = \sqrt{v_{j+1} \cdot v_{j+1}} \)
8: \( v_{j+1} := v_{j+1} / h_{j+1,j} \)
9: end

Algorithm 1: Arnoldi orthogonalization procedure

As the method proceeds, a new search vector is obtained by applying the Jacobian to \( v_j \). In lines 3-6, the resulting \( v_{j+1} \) is orthogonalized against all previous vectors before being normalized at line 8. The matrix representation of this procedure gives the Arnoldi relation:

\[
AV_m = V_{m+1}\bar{H}_m
\]

(3.11)

Where \( \bar{H}_m \) is a \((m+1) \times m\) real matrix with \( h_{ij} \) entries obtained from the Arnoldi loop in lines 4 & 7. From the iteration ranges in Algorithm 1, it can be seen that \( \bar{H}_m \) is a upper-Hessenberg matrix⁵.

As \( V_m \in K \), there exists a vector \( y \) such that the approximation solution can be expressed as:

\[
x_m = x_0 + V_m y
\]

(3.12)

Step 3) consists in finding the vector \( y \) which minimizes the value of \( ||r_m||_2 = ||b - Ax_m||_2 \).

⁵A matrix \( A \) is upper-Hessenberg if \( a_{ij} = 0 \) when \( i > j + 1 \)
Substituting equation (3.26) into \( ||r_m||_2 \) holds:

\[
||r_m||_2 = ||r_0 - AV_m y||_2
\]  

(3.13)

Making use of the Arnoldi relation and equation (3.9), this is recast as:

\[
||r_m||_2 = ||\beta v_0 - V_{m+1} \bar{H}_m y||_2 \\
||r_m||_2 = ||V_{m+1}(\beta e_0 - \bar{H}_m y)||_2
\]  

(3.14)

Where \( e_0^{(m+1)} = (1, 0, \ldots, 0)^T \), the columns of \( V_{m+1} \) are normalized and so the vector \( y \) minimizing the residual norm can be found by solving the following \( (m+1) \times m \) least square or reduced problem:

\[
\bar{H}_m y = \beta e_0
\]  

(3.15)

which is an inexpensive procedure as \( m \ll N \). A straight-forward way to do this is to perform a QR-decomposition on \( \bar{H}_m \). However, here Givens rotations are applied during the Arnoldi loop to the columns of \( \bar{H}_m \) making the matrix upper-triangular \([83]\), to unalter equation (3.15) the same rotations must be applied to \( \beta e_0 \): at every Arnoldi step \( k \) the least square RHS is populated with an additional entry \( (\gamma_0, \gamma_1, \ldots, \gamma_{k+1}) \). It can be shown \([83]\) that the last entries absolute value is actually the linear residual norm at that step \( |\gamma_{k+1}| = ||r_m||_2 \). This is one of the very attractive features of GMRES, where matrix splitting methods fix an arbitrary iteration count and hence no information about the degree to which the linear system is being solved, GMRES breaks down when the linear termination criterion (3.4) is met at no additional cost (i.e. without explicitly computing the residual). As it will be shown in the following section, this level of control over the linear convergence is particularly useful in the context of Newton-GMRES.

Finally in step 4), the solution is built using equation (3.26). When a Newton method is wrapped around GMRES, \( x_m \) is the Newton correction, the vector of conservative quantities is then updated using \( U^{n+1} = U^n + x_m \).

3.2.2 Globalization Strategy

In equation (3.5), a pseudo time-step was introduced to obtain better conditioned linear systems, this also stabilizes the Newton method which is needed in the first stages of the computation, when the Jacobian is still inaccurate because of the errors induced by the initialization of the solution arrays. A small pseudo time-step \( \Delta t \) increases the LHS diagonal terms of equation (3.5), and as a result damps the effect of the Jacobian in the solution process of the linearized problem. The linear correction terms obtained are thus more conservative and prevent unphysical quantities from arising when updating the solution (i.e. negative pressures).

As aforementioned, this increase in robustness is at the expense of poor nonlinear conver-
gence behaviour, as the linear system being solved does not correspond to the true system linearization. Once the main features of the flow are established, and the variations in the effect of the Jacobian are less significant, the time-step needs to be increased in order to recover a better nonlinear convergence behaviour. For this reason a nonlinear convergence strategy or globalization strategy must be implemented. First, the notion of local time-stepping is introduced:

\[
\left( \frac{I}{\Delta t^n_i} + J^n \right) \delta u^n = -R(U^n)
\]  (3.16)

For steady-state calculation the time-accuracy does not need to be preserved, each node \( i \) has its own varying time-step \( \Delta t^n_i \), which changes at each iteration \( n \). The time step change is controlled by the non-dimensional Courant-Friedrich-Lewy (CFL) number:

\[
\text{CFL} = \left( \frac{|u_i| + a_i}{\Delta x_i} \right) \Delta t_i
\]  (3.17)

which is constant throughout the computational grid at a given iteration. It is built using the Jacobian wave speeds \( |u_i| + a_i \) where \( a_i \) is the speed of sound, \( \Delta x_i \) is the characteristic length of the element. At the start of the computation the CFL number is initialized with \( \text{CFL} = 0.1 \) and is ramped up to a maximum value \( \text{CFL} = 10.1 \). These parameters are very conservative and have been effective in providing a robust start-up phase for the numerical experiments carried out in this work.

In the main phase of the computation the CFL is ramped up to larger values. It would be favourable to include a measure of the nonlinear method progress in the evaluation of the CFL such that in the vicinity of the solution, when the variations in the application of the Jacobian are small, the CFL tends quickly towards its maximum value. One way to proceed is to use the Switched-Evolution-Relaxation (SER) scheme [67], which increases the CFL based on the nonlinear residual decrease:

\[
\text{CFL}^{n}_{\text{ser}} = \text{CFL}^{n_0} \cdot \frac{||R(U^n)||_2}{||R(U^{n_0})||_2}
\]  (3.18)

where \( \text{CFL}^{n_0} \) is the SER scheme starting value and \( \zeta \) is a tuning parameter which amplifies the CFL rate of change. Two additional constraints are prescribed: 1) the CFL is capped at \( \text{CFL}_{\text{max}} \), as it will be seen shortly a fully Newton method is not always the most efficient. 2) The CFL cannot decrease, to prevent stagnation if local minima or momentary nonlinear residual overshoots occur in the solution process. \( \text{CFL}_{\text{max}}, \text{CFL}^{n_0} \) and \( \zeta \) are case dependent. In theory there exists an optimal \( \Delta t^n \) sequence tending towards infinity such that quadratic convergence could be obtained, however this rarely the most efficient strategy from a CPU-time perspective.

Removing entirely the time-step is counterproductive, as this results in additional work imposed on the linear solver to met the termination criterion. For GMRES, additional
work means increasing the Krylov subspace length, which is costly as the computational cost scales quadratically with every additional Krylov vector (because of the orthogonalization step). This could be acceptable if the linear residual drop corresponded to an equivalent nonlinear decrease, however, this is rarely the case as: 1) the NS equations remain very nonlinear 2) the Jacobian often does not correspond to the true linearization of the nonlinear flux function (more in section 3.2.4).

3.2.3 Forcing Term

A similar reasoning to the time-step choice can be made for the forcing term $\eta$. A small forcing term ($\eta < 10^{-2}$) will result in an excessive amount of work which, again, is unlikely to be the most efficient strategy. This is known as oversolving and must especially be avoided at large CFL number. Authors have worked on adaptive forcing terms to prevent the linear solver from oversolving. They consist of evaluating the forcing term based on the agreement between linear and nonlinear models [33]. Safeguards are implemented to prevent the forcing from decreasing too quickly in case of a temporary agreement in the models. However, for engineering applications with large linear systems, fixing the forcing term is usually more effective. In this work the forcing term is set to the constant value of:

$$\eta = 0.1$$  (3.19)

3.2.4 Finite Difference Jacobian

Increasing the time-step, and so decreasing the diagonal dominance of the LHS, increases the workload of the linear solver. This is acceptable (to some extent) as long as the Newton correction obtained is accurate, in the aim of obtaining a corresponding decrease between linear and nonlinear residuals when getting closer to the solution. While this is rarely the case because of the problem nonlinearity, it would nonetheless be favorable to amortise the cost of the linear iterations. For this, one must ensure that the linear problem solved corresponds to the true linearized problem. This rises the issue of the nonlinear function Jacobian accuracy.

Considering a control volume $i$ and its neighbour $j$, using Roe’s flux differencing scheme [80], the convective flux at the boundary interface after accumulation of the axial and radial flux components in the face normal direction takes the following form:

$$\Psi(U_i, U_j) = \frac{1}{2} [\Psi(U_i) + \Psi(U_j)] - \frac{1}{2} |\tilde{A}_\Psi| (U_i - U_j)$$  (3.20)

$\tilde{A}_\Psi = \frac{\partial \Psi}{\partial U}$ is the $5 \times 5$ flux Jacobian matrix. The differentiation of equation (3.20) first two
terms is straightforward, however the last term derivative leads to the following relation:

$$\frac{\partial \tilde{A} \Delta U}{\partial U_i} = \frac{\partial \tilde{A}}{\partial U_i} \Delta U + \tilde{A} \frac{\partial \Delta U}{\partial U_i}$$  (3.21)

The first term is a third rank tensor for which the evaluation is computationally expensive, it is usually omitted in the linearization of the nonlinear function which leads to a mismatch between the linearized systems RHS and LHS. This will hinder nonlinear convergence in the larger time-step region. Evaluating the Jacobian in this manner will be referred to as using the analytical Jacobian.

Only the result of a Jacobian-vector product is needed in GMRES (Algorithm 1). Instead of hand-linearizing equation (3.20), the application of the Jacobian can be computed by using a first-order finite difference between unperturbed and perturbed nonlinear functions:

$$J \delta u \approx \frac{R(U + \epsilon \delta u) - R(U)}{\epsilon}$$  (3.22)

where $\epsilon$ is a small factor. Using the finite difference Jacobian to compute the Krylov search vectors is known as the Jacobian-Free Newton-Krylov (JFNK) method [48]. It was originally developed to avoid explicitly building the Jacobian matrix, Here the application of the analytical Jacobian to a vector is obtained using numerical fluxes, nonetheless it is commonly accepted, especially in CFD because of the difficulties in obtaining a true hand-linearized Jacobian (aforementioned approximations in the convection operators but also the likelihood of approximations made in the linearization of the viscous wall boundaries and/or turbulent viscosity), that NK methods are more competitive when using finite difference Jacobians. Another advantage is their ability to easily approximate the linearization of the spatial second-order convection operator by evaluating perturbed/unperturbed $R(U)$ using the distance-2 neighbour stencil. The linearization of the boundaries is managed implicitly by recomputing the perturbed nonlinear flux functions at these interfaces. Evaluating the Jacobian-vector product this way will be referred to as the numerical Jacobian.

A comparison is made between the analytical Jacobian implementation of AU3X and the finite difference Jacobian. This is done by considering free-stream subsonic viscous flow over a NACA0012 airfoil shown in Figure 3.1b. The computational grid is shown in Figure 3.1a and consists of 12288 quadrilateral elements with 12 prism layers at the airfoil wall. The convection operator is built using first-order accuracy in space so that the comparison can concentrate on the approximations made in the linearization of the flux Jacobian. The start-up phase consists of 500 Newton-Jacobi iterations which end with a CFL value of 10, the solver then switches to Newton-GMRES/SER with parameters $CFL_{\text{max}} = 10000$ and $\zeta = 3.5$. Linear and nonlinear residuals for the main computation phase are shown in Figure 3.2. Up to values of CFL= 800, the convergence behaviour of the two Jacobian implementations is near identical, as the CFL increases, the nonlinear convergence rate of the numerically evaluated Jacobian increases with respect to the analytical case. In the numerical Jacobian case, it can be observed that the linear residual drops correspond
Figure 3.1: Subsonic flow over a NACA0012 airfoil: a) Computational grid - 12288 quadrilateral elements. b) Flow Mach number distribution.

better to the following nonlinear residual norm. At maximum CFL values, both methods on average need 140 Krylov vector evaluations to meet the linear termination criterion, the analytical Jacobian is $\approx 1.2 \times$ faster to compute than a finite difference Jacobian. However, due to the better accuracy of the linear correction obtained with the numerical Jacobian, it can be seen from Figure 3.3 that the overall speed-up factor to $\|R(U)\|_2 = 10^{-10}$ when compared to the analytical implementation is $\approx 1.38 \times$. In Figure 3.3, ”Equivalent RHS evaluation” is the CPU-time normalized by the time needed to compute the nonlinear flux function. The numerical Jacobian will be used to calculate the Krylov vectors for the steady-state cavity-seal cases, however, the analytical Jacobian remains useful as it meets the preconditioning strategy requirements (Section 3.3). A comparison between numerical and analytical Jacobian which reaches the same conclusions can be found in [91], a CFD example of a Newton-GMRES implementation using a analytical Jacobian can be found in [16].

The difficulty in using finite difference Jacobians comes from the evaluation of the scaling factor $\epsilon$: the factor must be large enough to avoid the presence of floating point roundoff errors but small enough to accurately represent the differentiation. Authors obtain $\epsilon$ in different ways [15, 48], Nielsen et al. [70] suggest the use of the following formula to compute $\epsilon$:

$$
\epsilon = \frac{\sqrt{\epsilon_m}}{||\delta u||_2}
$$

(3.23)

Where $\epsilon_m$ is usually taken to be machine zero which for a 64-bit double precision corresponds to $\approx 10^{-16}$. Chisholm et al. [23] experimented with different values of $\epsilon_m$, in the aim to reduce the dependency of $\epsilon$ on the largest components of $\delta u$ and propose the value of $\sqrt{\epsilon_m} = 10^{-6}$ which will be used.
Figure 3.2: NACA0012 - Newton-GMRES $l_2$ residual norm convergence comparison between numerically and analytically obtained Jacobian-vectors products.
3.3 Preconditioning

3.3.1 Background

The linearized systems obtained from the discretization of the NS equations are ill-conditioned, this results in increased Krylov basis sizes with associated computational costs and memory overheads. If left unaccounted for, this will severely degrade the linear solver performance, and hence, the overall nonlinear method time to convergence. The linear behaviour depends on the spectral distribution of the coefficient matrix, in particular, convergence is hindered by the presence of eigenvalues close to the origin. To improve linear convergence rates, a preconditioning step is applied, this is done by applying a linear transformation to the left of a linear system $Ax = b$, or in the present case to the right:

$$AM^{-1}\hat{x} = b \quad \text{with} \quad \hat{x} = Mx$$

Where $M^{-1}$ should be computationally cheap to obtain and apply, and the resulting system should be better conditioned and so easier to solve. For the extreme case where $M = A$, the eigenvalues of the resulting transformed coefficient matrix (i.e. $I$) are $1$, however, this amounts to exactly solving $Ax = b$ which is not cheap. Instead, $M$ is taken to be an easily invertible lower order approximation to $A$. For example, $M = \text{diag}(A)$ could be an adequate choice during the start-up phase when the coefficient matrix exhibits strong diagonal dominance because of the transient term. The Krylov subspace is...
rewritten:

$$K_m (AM^{-1}, v_0) = \text{span}\{v_0, AM^{-1}v_0, (AM^{-1})^2v_0, \ldots, (AM^{-1})^{m-1}v_0\}$$ (3.25)

The Arnoldi orthogonalization procedure including right preconditioning gives:

1: for $j = 0$ to $m - 1$ do begin
2: \[ w := M^{-1}v_j \]
3: \[ v_{j+1} := Aw \]
4: for $i = 0$ to $j$ do begin
5: \[ h_{ij} = v_{j+1} \cdot v_i \]
6: \[ v_{j+1} := v_{j+1} - h_{ij}v_i \]
7: end
8: \[ h_{j+1,j} = \sqrt{v_{j+1} \cdot v_{j+1}} \]
9: \[ v_{j+1} := v_{j+1}/h_{j+1,j} \]
10: end

Algorithm 2: Right-preconditioned Arnoldi orthogonalization procedure

To account for right preconditioning, the solution of the least-squares problem $y$ now minimizes the residual vector $||b - AM^{-1}\tilde{x}||_2$, accordingly the preconditioner must be taken into account when computing the solution to the linearized problem by using:

$$x_m = x_0 + M^{-1}V_m y$$ (3.26)

### 3.3.2 Linear Multigrid Preconditioning

It can be seen from the modified Arnoldi relation in Algorithm 2, that the application of right-preconditioning is the result of the matrix-vector product $w = M^{-1}v_j$. Here, $M^{-1}$ does not need to be explicitly built, but could rather be a function which approximates the solution to the linear system:

$$Mw = v_j$$ (3.27)

such as an iterative procedure. In this work, linear multigrid is considered, the method is invoked by calling a subroutine $w = MG(w, v_j)$. The two stages in linear multigrid are the application of a smoother, and the coarse-grid correction step. Jacobi iterations (equation (3.6)) are used to smooth the linear systems on all grid levels. Their application is denoted by the operator $\mathcal{S}$:

$$w = \mathcal{S}(v, w_0, v)$$ (3.28)

where $v$ is the number of smoothing operations performed on the initial solution $w_0$. In-erectly solving (3.27) is an approximation to the application of the Jacobian inverse on a vector, thus, it is reasonable to use a Jacobian approximate in the smoothing operations.
Using an exact Jacobian for all grid levels would actually be counterproductive, as non-linear convergence rates are not sensitive to approximations made in the preconditioner. The analytical Jacobian described in section 3.2.4 will be used in the smoothing iterations of the multigrid preconditioner.

Considering the system $Ax = b$ for which $x$ is the exact solution. At an intermediate step in the iterative procedure, for a iterate $x_k$, this would rewritten as $Ax_k = b + r_k$. By subtracting these two expressions the following system is obtained:

$$ Ae_k = r_k \quad (3.29) $$

where $e_k = x_k - x$ is the solution error. Equation (3.29) is transferred to a coarser grid or restricted on which it is smoothed. The coarse grid correction is then prolonged back to the fine grid and added to the current solution. Algorithm 3 illustrates the standard multigrid V-cycle procedure for $l = 0, \ldots, N_{\text{lev}}$ grids, with $l = 0$ being the finest level.

1: if $l = N_{\text{lev}}$ then
2:    Smooth: $w^{(l)} = S(v^{(l)}, 0, v^{(l)})$
3: else
4:    Pre-smooth: $w^{(l)} = S(v^{(l)}, 0, v^{(l)})$
5:    Compute residual: $r^{(l)} = v^{(l)} - M^{(l)}w^{(l)}$
6:    Restrict residual: $r^{(l-1)} = R^{(l)}(r^{(l)})$
7:    Call $w^{(l-1)} = MG(w^{(l-1)}, r^{(l-1)})$
8:    Coarse-grid correction: $w^{(l)} = w^{(l)} + P(w^{(l-1)})$
9:    Post-smooth: $w^{(l)} = S(v^{(l)}, w^{(l)}, v^{(l)})$
10: end

Algorithm 3: Linear Multigrid V-cycle pseudo-code for approximately solving $Mw = v_j$

This procedure can be repeated for $n_{\text{cp}}$ cycles per Krylov vector. $P^{(l)}$ and $R^{(l)}$ are the prolongation and restriction operators at a grid level $l$. The coarse level coefficient matrices $M^{(l)}, l > 0$ are built at every outer Newton step and are held constant throughout the GMRES cycle. As it has been established that the analytical Jacobian will be used in the preconditioner, building the coarse levels operators consists in restricting the state quantities about which the perturbations are linearized.

The difficulty in multigrid for unstructured grids resides in obtaining the intergrid transfer operators, in the case of GMG these are built by using the topological information of the coarser grid levels. For structured grids, the coarse meshes are obtained by omitting every second element vertex in each direction, resulting in a sequence of nested elements. The intergrid transfer operators are then based on injection or linear interpolation of the concerned quantities. For cell-centered schemes, usually the fine element variables are summed up and injected into the coarse element in which they are nested. For the unstructured case, different coarsening strategies exist, they can be regrouped into two types: methods which try and conserve the nested property of the structured case, and
methods which build the successive grids independently of one another. In the present work, an overset mesh technique is used [61]. The grids levels are built independently, the intersection areas between overlapping elements of successive grids are used as weighted averages to build injection-type transfer operators. A quantity $\Theta^{(l)}$ is restricted-to or prolonged-from a grid level $l - 1$ using the following rules:

$$
\mathcal{R}^{(l)}: \quad \Theta_k^{(l-1)} = \sum_{i=1}^{n(k)} \frac{A_{ik}}{A_k^{(l-1)}} \Theta_i^{(l)}
$$

$$
\mathcal{P}^{(l)}: \quad \Theta_k^{(l-1)} = \sum_{i=1}^{n(k)} \frac{A_{ik}}{A_k^{(l)}} \Theta_i^{(l-1)}
$$

(3.30)

Where $n(k)$ is the number of elements a control volume $k$ overlaps on a successive coarser/finer grid level. $A_k$ is the element area and $A_{ik}$ is the intersection area between two overlapping elements. Illustrations of the restriction and prolongation processes are presented in Figure 3.4 and 3.5. This procedure guarantees coarser grid level regularity.

![Figure 3.4: Illustration of the unstructured restriction process of fine grid quantities (left) to a coarse grid element (right)](image-url)
3.3.3 Implementation Aspects

This section concerns some more technical aspects relevant to the implementation of unstructured multigrid.

The intersection areas are computed prior to the flow computations. An ADT binary search tree [11] is used to efficiently obtain the nearest coarse neighbours of a fine grid element. The tree object is built using the coarse element cell-center coordinates, upon request of the coarse neighbours, the coordinates of a 2D box are passed to the ADT function which returns the indexes of the coarse elements residing within this range. The box coordinates are determined using the fine element cell center and characteristic length. The coarse elements returned are then tested for intersection, in the occurrence of this event, a polygon-clipping algorithm is used to calculate the intersection area. The area is then stored, as well as the index mapping between fine and coarse elements. In a third array which represent the number of overlaps per element \( n(k) \), the corresponding fine element entry is incremented. This procedure is repeated for all following grid sequences. The arrays are read at runtime before the start of the computation and are then used in the prolongation and restriction subroutines.

An important consideration to take into account when solving large-scale problems on a distributive memory environment is how to partition the domain efficiently. This is especially true in the case of finite volume discretizations. For a computation on \( p \) processors, a partitioner first obtains *load-balancing* by dividing the domain into \( p \) subdomains...
with equal number of elements. Secondly, the partitioning must be such that the communication time between processors is reduced. The exchange time between processors depends on the size of the subdomain boundaries, but also on the number of neighboring partitions. While these two tasks are manually manageable for structured computational grids, the same cannot be said of the unstructured case, where obtaining optimal communication/load-balancing ratios is a complex task. In the cavity-seal case, typically 2/3 of the computational domain elements are clustered around and inside the inner seal passage. The topology of the seal flow domain all but guarantees that any trivial decomposition technique will provide an efficient partition. To alleviate the issue, the open source program METIS developed by Karypis et al. [47] is used to partition the computational grids. The program uses a multilevel algorithm which consist in collapsing the input mesh graph into successive coarser graphs until a sufficiently reduced graph, which can easily be partitioned, is obtained. The partition is then projected back onto the finer levels and refined on each level until the original graph level. An example of a METIS domain decomposition for a straight-through seal domain with cavities is presented in Figure 3.6 and 3.7. The coarse level grid elements are partitioned based on their corresponding fine element cell-center closest neighbours. The coarse grid partition halos were assigned so that the information needed by the neighbouring subdomains of successive grids was accessible by the multigrid intergrid transfer operators.

### 3.4 Time-Linearized solver

The previous sections described the solution procedure for solving the discretized steady-state equations. This section and the following focus on solving the time-linearized flow problem defined by equation (2.24). For the nonlinear problem, it has been seen that an efficient linear solver is key to obtaining an efficient nonlinear solution procedure. The previous sections in this Chapter aimed towards this, and so it is natural that such methods, where applicable, be recycled to solved the time-linearized problem.

The time-linearized equations (2.27) after accumulation of the LHS fluxes in the face normal directions are recast as:

\[
\left( w_k I + \frac{k}{r} \frac{\partial \mathbf{R} \theta}{\partial \mathbf{U}} \right) \mathbf{i} + \frac{\partial \mathbf{R}}{\partial \mathbf{U}} \] \hat{\mathbf{u}} = b \quad (3.31)

Sisto et al. [69] suggest the use of a pseudo-time step \( \frac{d\hat{\mathbf{u}}}{d\tau} \) to march the linearized equations to convergence in a similar procedure to solving the steady-state equations:

\[
\frac{d\hat{\mathbf{u}}}{d\tau} + \left( w_k I + \frac{k}{r} \frac{\partial \mathbf{R} \theta}{\partial \mathbf{U}} \right) \mathbf{i} + \frac{\partial \mathbf{R}}{\partial \mathbf{U}} \] \hat{\mathbf{u}} = b \quad (3.32)

60
Figure 3.6: METIS domain decomposition on 20 processors of a straight-through labyrinth with surrounding cavities.

Figure 3.7: METIS domain decomposition on 20 processors of a straight-through labyrinth with surrounding cavities. - Seal inner-volume view.
The implicit formulation is obtained by using the Backward Euler scheme on equation (3.31):

\[
\{ I + \Delta \tau \left[ \left( w_k I + \frac{k \partial R_{\theta}}{r \partial U} \right) i + \frac{\partial \tilde{R}}{\partial U} \right] \} \Delta \hat{u} = \Delta \tau b - \left[ \left( w_k I + \frac{k \partial R_{\theta}}{r \partial U} \right) i + \frac{\partial \tilde{R}}{\partial U} \right] \hat{u}^n
\]

(3.33)

Where \( \Delta \hat{u} = \hat{u}^{n+1} - \hat{u}^n \) is the flow perturbation correction. Here, the solution is marched until \( \| \Delta \hat{u} \|_2 \approx 0 \). Unlike the steady-state problem, the LHS coefficient matrix is not updated at every outer-iteration. However, the forcing term must be recomputed with the new iterate \( \hat{u}^n \).

Another approach, suggested by Chassaing et al. [22] consists in directly solving the linear system by applying GMRES to equation (3.31). This will be referred to as solving the time-linearized equations without pseudo-time marching. By removing the time-step, Chassaing et al. showed that given a sufficiently large Krylov basis, the performance of GMRES without pseudo-time marching was superior to when including \( \frac{du}{dt} \). In the case of finer computational grids, the convergence of the solution using GMRES with pseudo-time marching was found to stagnate, which was not the case when omitting the time-step. Based on Chassaing et al. findings, GMRES will be used to solve the linear system (3.31) without pseudo-time marching.

Additional vectors are allocated for storing the imaginary part of the solution vectors, auxiliary variables and forcing term RHS. Modifying GMRES to support complex variables is relatively straightforward: special care needs to be taken in the Arnoldi orthogonalization step when computing the dot products (Figure 1 line 4.), which become complex inner products, and in the Givens rotation for obtaining the \( l_2 \) residual norm. The multigrid procedure described in Section 3.3.2 will be used to precondition the linear iterations. As the non-pseudo time-marching methods requires a large Krylov basis, the number of multigrid cycle per Krylov vector \( n_{cy} \) will help increase the residual drop per Krylov vector evaluation. Despite this, given the size of the problem, increasing the number of multigrid cycles is unlikely to be sufficient, and so GMRES will need to be restarted. For a Krylov basis of size \( m \), GMRES(m) is performed by computing the linear solution \( x_m \) at the end of a GMRES cycle and setting the initial solution of the following cycle to \( x_0 = x_m \).

In the numerical experiments carried out, it was noted that while evaluating the Jacobian-vector products numerically was effective in creating the Arnoldi vectors spanning the Krylov subspace, this was not the case when computing the initial vector \( v_0 = (b - Ax_0)/\|b - Ax_0\|_2 \) of a restarted GMRES iteration. In the lower residual stages of the computations (\( l_2 < 10^{-7} \)), when applying directly the finite difference Jacobian to the solution vector to solution obtained from a GMRES cycle, it was found that \( \|r_m\|_2 \neq \|\beta e_m - H_m y\|_2 \), i.e. the Arnoldi relation (3.11) no longer holds. The difference in these two quantities is small, the discrepancy comes from the roundoff errors introduced by the finite difference perturbation factor \( \epsilon \), which does not manage to consistently scale all the components of \( x_m \). In the Newton-GMRES context, where GMRES is not restarted, the accuracy of the solution is effectively controlled by the evaluation of the nonlinear flux function, and the sole impact of these errors will be in the nonlinear convergence.
rates because of the application of the Jacobian not being truly constant. In the time-linearized problem, where no outer quantity is to be updated, these errors directly impact the accuracy of the solution. This behaviour is exacerbated in the restarted context where the initial vectors are contaminated by the round-off errors of the previous cycles which are in turn amplified and propagated to the following cycles. This could be observed in the linear convergence histories, where after a certain threshold, overshoots were present at every restart iterations. A mathematical analysis was carried out by Choquet et al. [25] on the effect of round-off errors arising from finite difference Jacobian evaluation on the residual of restarted GMRES. Choquet et al. suggest the use of the second order finite difference when building the initial vectors of GMRES cycles. However, in their work GMRES was used as the linear solver to an implicit Newton method with less number of restart needed to meet the linear termination criterion than in the solution of large linear problems. For these reasons, the analytical Jacobian described in section 3.2.4, due to its application to a vector being constant, will be used in the solution of the time-linearized problem.

3.5 GMRES with Deflated Restart

It can be shown that the convergence of GMRES is dependent on the eigenvalue distribution of the coefficient matrix $A$ [83], or in the case where right-preconditioning is applied $M^{-1}A$. The presence of small eigenvalues hinders linear convergence. Removing or deflating the eigenvectors associated with the smallest eigenvalues from the Krylov subspace improves the convergence rates of GMRES. Deflation occurs naturally in the case where the Krylov basis is sufficiently large [29], however, for large problems this is not always feasible. As mentioned in section 3.4, the classical way to circumvent this issue is the use of GMRES(m). Unfortunately, this is to the detriment of the convergence rates because: 1) The size of the basis $m$ might not be large enough for automatic deflation to occur. 2) The information built from the previous GMRES cycle is lost as the procedure is restarted from the single vector $r_0$.

This motivated authors to develop methods which reincorporate spectral information made available during GMRES into the following restarted cycle. Amongst these, Morgan introduced the GMRES-E method or GMRES with eigenvectors [65]. The method makes use of the property of the Arnoldi iterations to extract an approximation to the $m^{th}$ dominant eigenvalues and eigenvectors of $A$ (or $M^{-1}A$). This is done by solving the following eigenvalue problem:

$$H_m g_i = \Theta_i g_i$$

Where $H_m$ is the matrix obtained from the $m^{th}$ first rows of the Hessenberg matrix $\tilde{H}_m$. $\Theta_i$ are the approximate eigenvalues of $A$, $m$ of them will be obtained from the eigenvalue problem. The associated eigenvectors are obtained using $y_i = V_m g_i$. The eigenpairs $(y_i, \Theta_i)$ are know as the Ritz values. In GMRES-E, the set of $k < m$ eigenvectors obtained from a cycle with eigenvalues closest (in magnitude) to the origin are added to the end of
the following GMRES cycle Krylov subspace:

\[ K_m (A, r_0) = \text{span} \left( r_0, Ar_0, A^2r_0, \ldots, A^{m-k-1}r_0, y_0, y_1, \ldots, y_{k-1} \right) \] (3.35)

The first \( m - k \) iterations are carried out as in regular GMRES, in the remainder of the range, no \( Av_i \) product occurs and the eigenvectors \( y_i \) are directly orthonormalized in the Arnoldi procedure against the previous Krylov vectors.

While the convergence behaviour using GMRES-E is improved, the first \( m - k \) portion of the algorithm does not benefit from the augmented subspace. Morgan [64] and Le Calvez et al. [17] concurrently worked on a method to address this matter, which Morgan named GMRES-DR. The methods consists in adding the spectral information at the beginning of the following subspace. As all the following Krylov vectors will be orthogonalized against this information, the more accurate Harmonic Ritz values are used. The Ritz values are the result of the eigenvalue problem built using the Arnoldi relation \( V^H_m A V = 0 \). The Harmonic Ritz values on the other hand, are obtained from the problem built using the Arnoldi relation in (3.11) which includes the full Hessenberg matrix (with the \( m + 1 \) row). It can be shown [66], that the eigenvalue problem obtained from the full Arnoldi relation holds:

\[ (H_m + \alpha^2 f e_m^T) \tilde{g}_i = \tilde{\Theta}_i \tilde{g}_i \quad \text{with} \quad \alpha = h_{m+1,m} \] (3.36)

Where the tilde means that the eigenpair are harmonic Ritz values. The vector \( f \) of size \( m \times 1 \) is obtained by solving the system \( H_m^H f = e_m \). As it will be seen shortly, in GMRES-DR the Hessenberg matrix obtained at the end of a cycle is no longer a Hessenberg matrix and so Givens rotations can not be applied directly. The system defining \( f \) is not upper-triangular and so a QR factorization is performed, which is an inexpensive procedure (\( m \ll N \)). The \( k \) first eigenvectors \( \tilde{g}_i \) of size \( m \times 1 \) associated with the smallest eigenvalues are orthonormalized and inserted into a matrix \( G \) of size \((m+1) \times (k+1)\). The last column of \( G \) is populated with the reduced residual \( c = \beta e_{m+1} - \bar{H}_m y \) of size \((m+1) \times 1\) obtained from the least-squares problem at the end of the previous GMRES cycle:

\[ G_{k+1} = \begin{bmatrix} G_k \\ 0_{1 \times k} \end{bmatrix} \quad c = \beta e_{m+1} - \bar{H}_m y \] (3.37)

The last column of \( G_{k+1} \) containing the reduced residual is orthonormalized against all previous columns. The matrix \( G \) is then used to build the matrices needed in the next GMRES cycle:

\[ V_{k+1} = V_m G_{k+1} \]
\[ \bar{H}_{k+1} = G_{k+1}^H \bar{H}_m G_k \] (3.38)

An optional step consists in re-orthogonalizing the columns of \( V_{k+1} \) for higher accuracy. The residual norm is set to \( \beta = ||c||_2 \) and the following GMRES cycle is applied to form the remaining \([k+1 : m] \) portion. The vector \( Av_i \) formed at \( i = k + 1 \) in the Arnoldi
loop will be orthogonolized against the previous column of $V$ which now contain the spectral information and the linear residual obtained at the end of the previous cycle. The pseudo-code for GMRES-DR presented in Algorithm 4.

1: set $\beta = r_0 / \| r_0 \|_2$, and apply the Arnoldi procedure to compute $V_{m+1}$ and $\tilde{H}_m$.
2: **Loop**
3: Solve the least square problem $\tilde{H}_m y = \beta e_{m+1}$.
4: Compute $x_m := x_0 + V_m y$.
5: Compute $c := \beta e_{m+1} - \tilde{H}_m y$.
6: If $\| r_m \|_2 = \| c \|_2 \leq \epsilon_{\text{tol}}$.
7: **Break**
8: **Else**
9: Solve $H_m^H f = e_m$.
10: Compute the eigenpairs $(\tilde{g}, \tilde{\Theta})$ of $H_m + \alpha^2 f e_m^T$ with $\alpha = h_{m+1,m}$
11: Determine the $k$ eigenvectors with eigenvalues closest to the origin.
12: Populate $G_k$ and orthonormalize.
13: Append $c = \beta e_{m+1} - \tilde{H}_m y$ forming the $k+1$ column of $G_{k+1}$
14: Orthonormalize the $k+1$ column of $\tilde{G}$ against previous columns.
15: Compute $V_{k+1}^{\text{new}} = V_m G_{k+1}$ and $\tilde{H}_{k+1}^{\text{new}} = G_{k+1}^H \tilde{H}_m G_k$.
16: Orthonormalize $V_{k+1}^{\text{new}}$.
17: Prolong $V_{k+1}^{\text{new}}$ and $\tilde{H}_{k+1}^{\text{new}}$ to $V_m$ and $\tilde{H}_m$ using the Arnoldi loop.
18: **End if**
19: **End Loop**

Algorithm 4: Pseudo-code of Morgan’s GMRES-DR($m,k$) procedure

Where $\epsilon_{\text{tol}}$ is the stopping criterion for the linear iterations. The application of the matrix containing the residual and spectral information to the Hessenberg matrix (equation (3.38)) of the previous GMRES cycle will yield a dense matrix, and hence the matrix obtained at the end of a GMRES-DR cycle will be upper-block Hessenberg. By applying a QR factorization to $\tilde{H}_m^{\text{new}}$, the first $k+1$ of the next cycle GMRES can be made triangular, and by storing $Q$, the following columns of the Hessenberg matrix can also be made triangular using Givens rotations for the $[k+1:m]$ portion. This allows the computation of the residual $l_2$ norm during a cycle to break the procedure if $\epsilon_{\text{tol}}$ is met. However a second $\tilde{H}_m$ will need to be stored to solve the eigenvalue problem in the case where the code does not break.

A benefit of GMRES-DR is that only $m - k$ Arnoldi iterations are performed in the main GMRES loop. The first $k$ portion performs computation on the reduced size problem which is computationally cheap, and performs a dense Matrix-vector products with re-orthogonalization to produce $V_{k+1}^{\text{new}}$ which remains cheaper that evaluating $k+1$ $Av_i$ products (and orthonormalizing them).

For real value problems, the eigenvalues extracted are usually both complex and real: the size of the arrays storing the quantities needed in the augmented process ($G, \tilde{H}^{\text{new}}, V^{\text{new}}$) must be made flexible to account for imaginary and real part of the eigenvectors. However,
the harmonic problem being already in the complex domain, the dimension $k$ can be fixed, and second arrays for the augmented matrices are allocated to store the imaginary components.

The eigenvalue value problem (3.36) is solved using the $cg$ routine for complex general matrices of the eigenvalue/eigenvector library EISPACK.

### 3.6 Summary

The steady-state nonlinear equations are solved using an inexact Newton-GMRES method with pseudo-transient continuation. The linear iterations terminate using the inexact formulation with a forcing term of $\eta = 0.1$. A pre-computing phase occurs at small CFL numbers to cancel the errors due to the initialization of the solution arrays. For the main computations, the time-stepping strategy is controlled by means of the SER scheme which is made flexible by means of the tuning factor $\zeta$, and the starting/maximum CFL numbers. GMRES is right-preconditioned using a $n_{cy}$ linear multigrid V-cycles, the unstructured case is handle using overlapping element areas as weighted-averages for the intergrid transfer operators which are obtained in a pre-processing stage. The Arnoldi vectors are obtained using finite difference Jacobian using Nielsen’s perturbation factor with a machine zero value of $\epsilon_{mach} = 10^{-12}$. The smoothing operations in the preconditioner are performed by means of point-wise Jacobi iteration which use the analytical Jacobian-vector product obtained with numerical fluxes. The partitioning of the unstructured domain is obtained using the METIS library.

The harmonic problem is solved by applying complex GMRES directly to the linearized equations without pseudo-time marching. The Arnoldi vectors are obtained using the analytical Jacobian and preconditioned by applying linear multigrid. GMRES will need to be restarted, better convergence behavior can be obtained by using GMRES-DR which consists in incorporating previous cycle spectral information at the beginning of the following GMRES cycle.
Chapter 4

Numerical Integration

This chapter provides information about the computation of the nonlinear residual function. The discrete form of the time-linearized equations and the process for building the linear system RHS are covered. Finally, the procedure for obtaining the harmonically perturbed computational grids is discussed.

4.1 Spatial discretization

All variables throughout this section refer to steady-state quantities.

4.1.1 Convective Operator

Using a unstructured mesh generator, the domain is discretized into $N_{\text{nodes}}$ control volumes. Considering a control volume $i$ with $n_f$ faces of area $\Gamma_{ij}$ (as shown in Figure 4.1), the convective contribution of the steady-state equation (2.28), is given as:

\[ R_c^i = \sum_{j} [F^c(U_i, U_j) + G^c(U_i, U_j)] n_{ij} \Gamma_{ij} \]  

(4.1)

Which after accumulation of the radial and axial flux components in the face normal directions $n_{ij} = (n_z, n_r)$, can be recast as:

\[ R_c^i = \sum_{j} \Psi(U_i, U_j) \Gamma_{ij} \]  

(4.2)
Here, $\Psi(U_i, U_j)$ is obtained at a control volume boundary interface between elements $i$ and $j$ using Roe’s flux differencing method [80]:

$$
\Psi(U_i, U_j) = \frac{1}{2} [\Psi_{ij}(U_i) + \Psi_{ij}(U_j)] - \frac{1}{2} |\hat{A}_\Psi| (U_i - U_j)
$$

(4.3)

The flux Jacobian obtained at the boundary face $|\hat{A}_\Psi| = T^{-1}|\Lambda|T$ is diagonalized into right eigenvectors $T$ and the diagonal matrix of eigenvalues $\Lambda$ evaluated using Roe’s density-based weighted average. To avoid the presence of zero eigenvalues which lead to non-physical expansion shocks, Harten’s entropy correction [42] is included in the computation of $\Lambda$. The cell-center contributions $\Psi_{ij}(U_i)$ and $\Psi_{ij}(U_j)$ of the interface flux are now given as:

$$
\Psi_{ij}(U_j) = \begin{bmatrix}
\rho(u_n - u_f) \\
\rho u_z(u_n - u_f) + p_m z \\
\rho u_r(u_n - u_f) + p_m r \\
\rho(u_n - u_f) h
\end{bmatrix}
$$

(4.4)

where $u_n = \mathbf{u} \cdot \mathbf{n}$ and $u_f$ is the velocity of the face $ij$. In the steady-state problem, $u_f = 0$ for all quantities computed using equation (4.1), the velocities of the control volume faces have been conveniently defined here as they will be required subsequently.

### 4.1.2 MUSCL Scheme

Second-order accuracy of the convective spatial operator is obtained by means of Van Leer’s MUSCL scheme [67]. The underlying idea consist in reconstructing the variables
$U_i$ and $U_j$ needed for computing the flux (4.3) at a control volume interface using a linear piece-wise extrapolation. The reconstructed variables $U_i^+$ and $U_j^-$ are obtained by extrapolating the cell-centered quantities to the interface center $x_{ij}$:

$$U_i^+ = U_i + \frac{\Phi_i}{4} \left[ (1 - \kappa \Phi_i) \Delta_i + (1 + \kappa \Phi_i) (U_j - U_i) \right]$$

(4.5)

$$U_j^- = U_j - \frac{\Phi_j}{4} \left[ (1 - \kappa \Phi_j) \Delta_j + (1 + \kappa \Phi_j) (U_j - U_i) \right]$$

(4.6)

with:

$$\Delta_i = 2 \nabla U_i \cdot (x_{ij} - x_i) - (U_j - U_i)$$

$$\Delta_j = 2 \nabla U_j \cdot (x_{ij} - x_j) - (U_j - U_i)$$

and the parameter $\kappa = 0$ which leads to the second-order upwind scheme. The solution gradients $\nabla U$ are computed using the weighted least-squares method [10]. $\Phi$ is the flux limiter, the aim is to obtain the largest possible value of $\Phi$ while respecting the monotonicity principle: the reconstructed variables $U_i^+$ and $U_j^+$ must be bounded by the minimum and maximum values of the interface neighbouring elements. Limiters are needed around solution discontinuities to avoid spurious oscillations and reconstructing the variables with unphysical quantities. The drawback is that limiters are known to severely hinder convergence behaviour especially when near-constant flow regions are present. These regions, for which the flow solution is smooth, do not require any limiting, however $\Phi$ is affected by machine level noise resulting in intermittent limiter activations. In the present case, this causes two issues: 1) The baseline iterative method must converge before the solution procedure proposed in Chapter 3 can be applied. 2) This behaviour will also affect the accuracy of the second-order finite difference Jacobian operator. An option is to freeze the limiter once the main feature of the flow are obtained, although there is no real guarantee of this being successful. Instead, the flux limiter proposed by Venkatakrishnan [93], which is renown for alleviating these issues [68], is implemented. $\Phi_i$ and $\Phi_j$ then take the following form:

$$\Phi_i = \frac{1}{\Delta_-} \left[ \frac{(\Delta_+^2 + \epsilon^2) \Delta_- + 2 \Delta_- \Delta_+}{\Delta_+^2 + 2 \Delta_-^2 + \Delta_+ \Delta_- + \epsilon^2} \right]$$

(4.7)

where:

$$\Delta_- = \Delta_i$$

$$\Delta_+ = \begin{cases} U_i^{\text{max}} - U_i & \text{if } \Delta_- > 0 \\ U_i^{\text{min}} - U_i & \text{if } \Delta_- < 0 \end{cases}$$

$U_i^{\text{min}}$ and $U_i^{\text{max}}$ are the minimum and maximum quantities over all neighbouring and current element $i$, the perturbation factor is fixed at $\epsilon = 0.05(U_i^{\text{max}} - U_i^{\text{min}})$ where here the minimum and maximum solution quantities are evaluated over the entire domain. A comparison is made in Figure 4.2 with the van Albada flux limiter on the NACA0012.
Figure 4.2: NACA0012 - Second-order Newton-GMRES $l_2$ residual norm convergence comparison between van Albada and Venkatakrishnan flux limiters. The finite difference Jacobian is evaluated using second-order perturbed and unperturbed nonlinear function with the relevant limiter.

case described in Section 3.2.4, it can be observed that the solution procedure using van Albada stagnates at $\approx ||R(U)||_2 = 10^{-4}$. Both the nonlinear RHS term and the finite difference Jacobian are computed to second-order accuracy with the relevant limiter.

### 4.1.3 Viscous Operator

The viscous contributions given in section 2.2 are discretized on the same control volumes used for the convective operator. The flow solution gradients at an interface needed to build the viscous stress and conduction terms are approximated using the solution quantities on both sides. The interface normal gradient quantities are obtained as:

$$
\frac{\partial U_{ij}}{\partial n} = \frac{U_j - U_i}{(x_j - x_i) \cdot n_{ij}}
$$

(4.8)

The tangential gradient quantities along the interface of direction $l_{ij}$ are then computed as:

$$
\frac{\partial U_{ij}}{\partial l} = \frac{\partial U_{ij}}{\partial n} \cdot l_{ij} - \left[ \frac{\partial U_{ij}}{\partial n} \cdot l_{ij} - \frac{U_j - U_i}{(x_j - x_i)} \right] l_{ij}
$$

(4.9)
4.1.4 Boundary Conditions

The free-stream boundary condition types rely on the direction of quantities known as the characteristics, this is be done by checking the signs of the convective flux Jacobian eigenvalues at the concerned boundary interface. The boundary variables are then either extrapolated from the domain or specified depending on if the characteristics are travelling inside or outside of the domain. In the present cavity case, subsonic conditions occur and the outlet/inlet boundaries. For the inlet boundary, 4 eigenvalues are positive and 1 is negative, therefore one characteristic variable is extrapolated from the domain and the 4 remaining are prescribed. Conversely, for the outlet boundary 4 variables are extrapolated and 1 is specified.

At the solid wall boundaries, flow tangency is imposed such that \( \mathbf{u}_n \cdot \mathbf{n} = 0 \), consequently the convective flux contribution at this interface is obtained as:

\[
\Psi_{\text{wall}}(\mathbf{U}_i) = \begin{bmatrix}
0 \\
p_{nz} \\
p_{nr} \\
0 \\
0
\end{bmatrix}
\] (4.10)

The no-slip condition applies at the wall interfaces, the viscous boundary shear stresses are obtained using an automatic wall function in similar fashion as [43].

4.2 Time-Linearized Equations: Linear System Forcing Term

It can be recalled that the differential form of the time-linearized equations was given in Section 3.4 as:

\[
\left( w_k \mathbf{I} + \frac{k}{r} \frac{\partial \mathbf{R}}{\partial \mathbf{U}} \right) \mathbf{i} + \frac{\partial \mathbf{R}}{\partial \mathbf{U}} \dot{\mathbf{u}} = \mathbf{b}
\] (4.11)

The complex forcing term \( \mathbf{b} \) at a control volume \( i \) is built using the steady-state quantities and the real and imaginary parts of the nodal deflections \( \mathbf{v}_{ij} = (v_{re}, v_{im}) \):

\[
b_i = -\frac{\partial \mathbf{R}_i}{\partial \mathbf{X}} \mathbf{v}_{ij} - \frac{\partial \mathbf{R}_i}{\partial \mathbf{X}} \dot{\mathbf{v}}_{ij} + iw_k \sum_{j}^{n_f} \mathbf{v}_{ij} \cdot \mathbf{n}_{ij} \Delta \mathbf{U}_{ij} - i\Delta \dot{V}_i \mathbf{U}_i + \mathbf{\dot{S}}_i
\] (4.12)

Where \( \Delta \dot{V} \) is the change between perturbed and unperturbed element volume and \( \mathbf{\dot{S}} \) is the rotational frame source term contribution induced by \( \Delta \mathbf{V} \). The first term of \( \mathbf{b} \) is the sensitivity of the residual function to the nodal displacement, it is obtained using finite

\footnote{For the 2D axisymmetric case the flux Jacobian eigenvalues are \( (u_n - a, u_n, u_n, u_n, u_n + a) \)}
difference in a similar fashion to equation (3.22), by perturbing the $R(\bar{U}, \bar{x})$ about the nodal displacements:

$$\frac{\partial R}{\partial \bar{x}} \bar{v} = \frac{R(\bar{U}, \bar{x} + \epsilon \bar{v}) - R(\bar{U}, \bar{x})}{\epsilon}$$ (4.13)

Here, the perturbation factor is obtained as $\epsilon = \frac{||v||_2}{10^6}$. The second term is the residual sensitivity to the nodal velocities, as $\dot{v} = iw_k \bar{v}$, the face normal velocities defined in equation (4.10) are computed as $u_f = iw_k \bar{v} \cdot n$. The differentiation of the residual function with respect to the face velocities then holds:

$$\frac{\partial R}{\partial \dot{x}} \dot{v} = \frac{R(\bar{U}, \epsilon u_f) - R(\bar{U}, 0)}{\epsilon}$$ (4.14)

With $\epsilon$ evaluated using the Euclidean norm of $u_f$. The sensitivities to nodal displacements/velocities are evaluating to second-order accuracy using the Venkatakrishnan flux limiter. In the complex operator $\hat{A}$, the steady-state quantities about which the perturbation $\hat{u}$ are linearized, using the analytical Jacobian described in section 3.2.4, are also evaluated using second-order accuracy. The turbulent viscosity $\mu_t$ obtained with the Cebeci-Smith model is held constant in the application of the linearized viscous flux to the complex amplitude $\hat{u}$. As no flow separation regions are present in cavity-seal domains, keeping the turbulent viscosity frozen was deemed an acceptable assumption.

### 4.3 Harmonic Grids

The computational grid nodal displacement quantities remain to be determined, this is done in two steps. First, the steady-state mean flow mesh is generated and the mode shapes of the seal rotor are obtained by a FE program. Secondly, the mode shape deflections are interpolated to the CFD mesh wall vertices before being propagated to the interior nodes. This is done by means of a pre-existing routine which was originally developed for computing harmonic grids for turbomachinery blade problems. The CFD grid deflections obtained must conform to the boundary conditions set by the given mode shape at the rotor wall vertices, and must go to zero at the far field wall boundaries with ideally a smooth distribution of the interior nodes. To this end, the deformed grid is described by a Laplace equation of the form:

$$\nabla^2 x = 0$$ (4.15)

which is solved using Jacobi iterations. The nodal deflections are then obtained by subtracting this result from undeflected vertex positions. A modification is made to the code to allow the wall vertices to slide along the seal stator boundary, this was done to avoid any overlaps between control volume edges and the flow domain wall boundaries at the fin tip regions. An example is shown on a straight-through seal computational grid in Figure 4.3 with its perturbed counterpart in Figure 4.4. Enlarged views of the inlet and outlet fin regions are shown in Figure 4.5 and 4.6.
Figure 4.3: Straight-through seal: original computational grid - 163634 quadrilateral elements.

Figure 4.4: Straight-through seal in Figure 4.3: perturbed computational grid.
Figure 4.5: Straight-through seal - Inlet fin tip. a) Original computational grid b) Perturbed computational grid. The labyrinth seal is supported at the outlet.

Figure 4.6: Straight-through seal - Outlet fin tip. a) Original computational grid b) Perturbed computational grid. The labyrinth seal is supported at the outlet.
Chapter 5

Results I: Steady-State

5.1 Introduction: Case Description

In this Chapter, the results for the steady-state computations are presented. The Chapter is divided into three Sections. In the current Section, the computational test cases are described. The following Section presents the performance of the Newton-GMRES method described in Chapter 3. Finally, results about the steady-state flow behaviour are shown.

The computational cases consist of three cavity-seal configurations: Case A & B are straight-through and stepped labyrinth seals with simplified cavities, Case C is the same seal as in Case A but with slightly more realistic high and low pressure cavity representations. Cases A & B are tested for 2 different seal clearances, each with 6 different pressure ratios which were chosen to represent typical engine operating conditions. The set-up for Case C reproduces the case discussed by di Mare et al. [30]. The reason for choosing this case is twofold: 1) In the paper by di Mare et al., a coupled time-accurate analysis of flutter is carried out on this case, and hence it will be used to validate the time-linearized method discussed in Chapter 2. 2) The case will be used to assess the performance of the iterative methods implemented when dealing with more complicated solution features which are mainly due to the higher flow recirculation regions present in the cavities. Case C is tested for 1 seal clearance and 1 pressure ratio, adding up to a total of 5 computational grids for 25 different steady-state computations. The case descriptions are summarized in Table 5.1. The case terminology is given the format LXX, where L is the case configuration (A, B or C), and XX accounts for the seal clearance. For instance, case A05 is a straight-through seal with simplified cavities and a 0.5mm clearance which is the case investigated by Phibel [73] and Phibel et al. [74]. Evidence of case A05 being aeromechanically unstable during experimental rig tests has been reported. The computational domains, including information on the whereabouts of the rotating boundaries, inlet and outlet boundaries, are presented in Figure 5.1.
The computational grids are obtained using the mesh generator GAMBIT. The grids are unstructured with quadrilateral elements. A mesh density study for case A05 was carried out by Phibel in [73] between 165,234 and 363,539 element grids. The difference in the two grids was found to change the mass flow rate by 1% and so the coarser grid was deemed sufficient. These numbers were for grids which had been extruded over a 10° sector (with 1 mesh element per degree) and so the 2D grids would be reduced by a factor of 10. Nonetheless, the present work aims towards building 2D computational grids with as many elements as the full grids in [73]. The grids for case B are made larger to evaluate performance of the iterative methods implemented and in particular the ability of the finite difference Jacobian to deal with large problems. The maximum size grid is for case B03 which is discretized into 332478 quadrilateral elements. A prism layer mesh is generated in the inner seal regions on the rotating and static parts to capture the boundary layer which affects the leakage flow across the labyrinth seal. The total boundary layer height is taken to be 20% of the given seal clearance and the height ratio parameter is set at 1.1. Cases A & C have 16 prism layers and cases B have 8. Size functions are used at the fin tip wall vertices to refine this regions, resulting in $\approx \frac{2}{3}$ of the grid elements being clustered in the seal inner volume and surrounding regions. For these parameters, the dimensionless distance $y^+$ values for the wall boundary neighbouring elements were less than 10, with the upper range located at the fin tips. The mesh properties for cases A/B/C including elements at the fin tip are given in Table 5.1. The computational grids for cases A05, B05 and C05 are presented in Figures 5.2, 5.6 and 5.7.

<table>
<thead>
<tr>
<th>Case</th>
<th>Seal Type</th>
<th>Cavity</th>
<th>Fin No.</th>
<th>Clearance (mm)</th>
<th>Seal radius (m)</th>
<th>$p_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A03</td>
<td>straight-through</td>
<td>S</td>
<td>4</td>
<td>0.3</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>A05</td>
<td>straight-through</td>
<td>S</td>
<td>4</td>
<td>0.5</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>B03</td>
<td>stepped</td>
<td>S</td>
<td>3</td>
<td>0.3</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>B05</td>
<td>stepped</td>
<td>S</td>
<td>3</td>
<td>0.5</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>C05</td>
<td>straight-through</td>
<td>R</td>
<td>4</td>
<td>0.5</td>
<td>0.3</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.1: Computational test cases (S = simplified, R = realistic)

<table>
<thead>
<tr>
<th>Case</th>
<th>No. of elements</th>
<th>No. of elements on fin tip</th>
<th>No. of quad layers</th>
</tr>
</thead>
<tbody>
<tr>
<td>A03</td>
<td>149257</td>
<td>38</td>
<td>16</td>
</tr>
<tr>
<td>A05</td>
<td>163634</td>
<td>44</td>
<td>16</td>
</tr>
<tr>
<td>B03</td>
<td>332478</td>
<td>18</td>
<td>8</td>
</tr>
<tr>
<td>B05</td>
<td>266228</td>
<td>18</td>
<td>8</td>
</tr>
<tr>
<td>C05</td>
<td>178111</td>
<td>28</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 5.2: Characteristics of computational grids for cases A, B and C
Unless mentioned otherwise, all steady-state cases are computed using 3 multigrid levels. Due to the level of unstructuredness of the grids, the domains are divided into subdomains each of which aiming to be discretized with a factor of 4 decrease in resolution per grid level. The level of control on the unstructured mesh generator for this goal being relatively poor, minimum and maximum ratio tolerances were set to 4.2 and 3.8 to prevent low performance of the multigrid routine. This range was deemed acceptable, as the multigrid method is less sensitive to the quality of the transfer operators in a preconditioning configuration than when being used as a solver in linear multigrid or FAS. The number of prism layers, being a user input, is divided by 4 at every grid level. In a pre-processing stage, the overlapping elements areas are obtained and stored with the connectivity arrays which are then read at runtime. The multigrid levels are presented for case A05 in Figures 5.2, 5.3 and 5.4. Enlarged views of the fin tip and inner seal cavity regions are shown in Figure 5.5.

The computations were performed on a compute node with two socket Intel Haswell CPU (E5-2650v2) clocked at 2.6Ghz and running on all 20 physical cores. All communications between the different processors are handled using the Message Passing Interface (MPI). As described in Section 3.3.3, the finer level computational grids are partitioned using the METIS library, the coarse level elements are then assigned a processor based on their closest neighbour fine element core number.
Figure 5.1: Computational domains - Cases A, B and C.
Figure 5.2: Multigrid levels for case A05 - Level 0: 163634 elements

Figure 5.3: Multigrid levels for case A05 - Level 1: 43880 elements

Figure 5.4: Multigrid levels for case A05 - Level 2: 13623 elements
Figure 5.5: Multigrid levels (enlarged views) for case A05
Figure 5.6: Computational grid for case B05 - Level 0: 266228 elements

Figure 5.7: Computational grid for case C05 - Level 0: 178111 elements
5.2 Performance of Iterative Methods

In this section, the performance of Newton-GMRES for solving the steady-state equations is discussed. The performance of cases A & B, for which the solutions constitute the base for the flutter parametric study carried out in the following Chapter, is assessed for lowest and highest pressure ratio $p_r = 1.1, 3.5$. Case C being only computed for pressure ratio $p_r = 3.2$, is treated separately. This case will be used to assess the performance of Newton-GMRES for more complicated flow features and to evaluate the multigrid preconditioning strategy in the nonlinear solver. The nonlinear method used for a benchmark is the steady-state iterative solver originally present in AU3X which is a fully implicit Newton method with point-wise Jacobi linear iterations. It is worth mentioning here that this is also the iterative solver used in previous work to obtain steady-state solution for cavity-seal configurations [84, 74, 73, 30]. The nonlinear flux function is evaluated using Roe’s scheme with vector difference splitting and is second-order accurate in space by means of the MUSCL scheme coupled with the Venkatakrishnan flux limiter. The application of the Jacobian of the residual function on a vector is computed using the first order linearized numerical fluxes (analytical Jacobian). The pseudo-transient continuation method is used with a maximum CFL $= 10$ with 20 Jacobi sweeps per Newton outer-iterations. In the numerical experiments carried out, because of the mismatch between the order in space of the nonlinear function and its Jacobian, increasing the CFL did not provide any benefits in nonlinear convergence. While it might be argued that the choice of this benchmark solver, being relatively basic, might not provide a fair comparison, Newton-Jacobi does have compelling advantages: the setup cost is low as no multigrid levels are to be generated, the algorithm needs few input parameters, has as low computational cost-per-iteration and is very robust. Newton-GMRES needs more parameters, and some knowledge of the algorithm is needed in the case where a breakdown might occur. For these reasons, this section also aims to provide a set of conservative parameters for a robust Newton-GMRES solution procedure in seal/cavities domains.

**Start-up phase**  The solution arrays are initialized with a uniform distribution. At this stage, the system is solved using Newton-Jacobi, with the pseudo-transient term controlled by the CFL number which is ramped up from 0.1 to 10. For all cases, these parameters provided a globally convergent and robust scheme.

Newton-GMRES could also have been applied in these early stages of the computation. However, the finite difference Jacobian would need to be evaluated using first-order nonlinear functions as the flux limiter will exhibit strong variations because of the initialization errors. This would lead to two $R(U)$ evaluations per Newton outer iteration (without counting the perturbed $R(U + \epsilon \delta u)$ needed for forming the Jacobian): a second order nonlinear function evaluation for the linear system RHS, and a first order evaluation for the unperturbed finite difference Jacobian contribution. A better choice is to use the first order analytical Jacobian which is more efficient at lower CFL values. A minimum number of Krylov vectors also needs to be set to ensure that the different flow variable equations are solved to an equivalent linear tolerance. Alternatively, this can be done by decreasing
the forcing term. In the present cases at low CFL values, the ratio of cost-per-iteration to nonlinear residual decrease, made Newton-Jacobi a more efficient candidate for start-up.

In Figure 5.8, a typical start-up phase nonlinear convergence history is presented. The start-up phase aims to reach a stage in which the variations of the Jacobian effects on a vector become less important, the better agreement between nonlinear and linear models allowing the use of larger time-steps. This stage is reached when the main features of the flow are established. For flow computations over NACA airfoils, this usually occurs when the position and strength of the shockwave have stabilized. In the case of seal-cavities, where no shocks are present, another criterion must be determined. Here, this is done by monitoring the mass flux through the inlet/outlet boundaries of the domain which are shown for case A05 in Figure 5.9. In the present work, mass flow rates are nondimensionalized and are presented by means of the Flow function $F$ defined as:

$$F = \frac{\dot{m}\sqrt{T_i}}{A_c P_i}$$

(5.1)

Where $\dot{m}$ is the mass flow rate at the concerned boundary, $T_i$ and $P_i$ are the total temperature and pressure at the inlet boundary, and $A_c$ is the seal clearance area. In Figure 5.9, it can be observed in the first 10000 iterations that the initialization errors are rapidly cancelled by the iterative method, resulting in a rapid decrease in the relative error between the mass flow at the inlet/outlet boundaries. After a momentary better agreement, the boundary mass fluxes start to converge, with a lower rate, towards their final value. At this stage, the residual level is controlled by the poor accuracy of the solution. This can also be seen in Figure 5.8, where the nonlinear residual norm quickly drops in the first 10000 iterations, before converging with a lower rate. At the end of the startup phase, the relative error between the mass fluxes was inferior to 1% which was found to be a suitable threshold to further increase the time-steps and effectively switching to the Newton-GMRES scheme. For the case shown in Figure 5.8, the SER scheme could be started earlier (around 15000 iterations), as no local minima are present in the residual norm. However, this is case dependant, and so would involve tuning of the SER quantities for different pressure ratios/computational domains. For this reason, the more conservative threshold of 80000 iterations was chosen, to provide an efficient startup phase for all cases.
Figure 5.8: Start-up phase - Nonlinear convergence history for case A05 with $p_r = 3.5$.

Figure 5.9: Flow function at Inlet/Outlet boundaries and relative error for case A05 with $p_r = 3.5$. 
Cases A and B  After start-up, the iterative solver switches to Newton-GMRES and the CFL number is controlled by the SER scheme. The initial CFL value is set to 100 and capped at $\text{CFL}_{\text{max}} = 2000$, the tuning factor is set to $\zeta = 1.5$. At this stage, less variations are present in the flux limiter, and so the finite-difference Jacobian-vector product are computed using the second order in space nonlinear function evaluation at a given Newton-step and its perturbed counterpart. The GMRES cycles are preconditioned using all three levels with $n_{cy} = 1$ multigrid V-cycles per Krylov vector. $v = 2$ pre/post-smoothing Jacobi iterations are performed in the multigrid preconditioner on all grid levels, with the exception of the coarsest level where $v^{(2)} = 4$, and so 8 consecutive Jacobi iterations, were applied to amortise the cost of the restriction and prolongation process. The increase in number of Jacobi sweeps on the coarsest level is acceptable as the grid resolution is reduced by a factor of $\approx 16 \times$. The forcing term is fixed at a value of $\eta = 0.1$. The maximum Krylov subspace size was set to $m = 150$, although this number has not been attained in any of the numerical tests carried out.

The nonlinear residual convergence histories obtained with Newton-GMRES for cases A & B at pressure ratios $p_r = 1.1$ and $p_r = 3.5$ are presented in Figure 5.10. At these pressure conditions, the flow Mach number at the outlet seal fin correspond to the subsonic regime for $p_r = 1.1$, and to sonic conditions for the $p_r = 3.5$ case where the flow is choked. For case A05, the exit fin this corresponds to $M_{p_r=1.1} = 0.18$ and $M_{p_r=3.5} = 1.04$. The high/low pressure cavities are dominated by low subsonic flow with $M < 0.1$. Cases A03, B03 and B05 have similar Mach number distributions which are presented in the following section.

All cases reach full convergence at $\| \tilde{R}(\tilde{U}) \|_2 \approx 10^{-12}$. The convergence history of Newton-Jacobi is shown for comparison. Because of the cheaper Newton-Jacobi cost per outer iteration, but also because the time spent in a Newton-GMRES iteration is variable, a fairer comparison is shown in Figure 5.11 by plotting nonlinear residual norm against CPU-time. On average, cases A reached full convergence in 14.43 hours, the shortest computation time being for case A05 with $p_r = 3.5$ which converged in 7.45 hours, and the longest for case A03 with $p_r = 1.1$ which took 14.61 hours. Cases B05 with $p_r = 1.1$ and $p_r = 3.5$ converged in 35.63 hours and 26.40 respectively. The largest case B03 converged in 54.36 hours for pressure ratio $p_r = 1.1$ and in 26.75 hours for $p_r = 3.5$.

On average, the speed-up obtained from using Newton-GMRES for cases A with $p_r = 3.5$ is $18.27 \times$. This factor was obtained based on the CPU-time needed for Newton-GMRES to reach the nonlinear residual level of Newton-Jacobi at the end of the computation. For $p_r = 1.1$, the order of magnitude of the candidate preconditioned Krylov vectors to be perturbed are smaller, and hence more sensitive to the round-off errors introduced by the finite difference Jacobian perturbation factor, the speed-up factor was $14.43 \times$. This increase in convergence rates, when compared to Newton-Jacobi, is obtained by imposing larger time-steps. The gains in nonlinear residual drops obtained from the CFL increase are possible due to the accuracy of finite difference Jacobian which evaluates the linearization of the second-order operator convection operator more accurately than the Jacobian used in the Jacobi scheme. The minimization of the residual norm property in GMRES also helps towards this. Finally, the CPU-time spent in the linear iteration is reduced by means
of the multigrid preconditioning and the forcing term $\eta = 0.1$ which is kept at a small value.

The speed-up factors obtained for cases B are less dramatic: cases B05, produced similar speed-up results for both operating conditions which averaged $4.33 \times$ and the larger grid cases B03 obtained $3.05 \times$ and $4.77 \times$ speed-up for pressure ratios $p_r = 1.1$ and $p_r = 3.5$.

Here, convergence is limited by two issues which also apply to cases A, but are exacerbated in the case of larger grids: 1) The solution variable of the energy equation is larger than the other components typically by 1 or 2 orders of magnitude, and so the Hessenberg matrix entries (i.e. the inner-product between current/previous Krylov vectors) will be more representative of this component. This is also the case when obtaining the norm of the initial residual vector which populates the first entry of the reduced problem RHS. As a result, the reduced problem being mostly defined by the energy component, the linear residual norm decrease obtained after application of Givens rotations does not correspond to an equivalent decrease in all components of the solution. The linear solution will still converge, but differs slightly from the solution to the problem obtained from the linearization of the nonlinear function, leading to poor nonlinear convergence. 2) Similarly, because of the Krylov vector norm present in the evaluation of the finite Jacobian perturbation factor, the smaller components of $\delta u$ are not scaled as effectively, introducing round-off errors. This is worsened in the case where the entries of the vector to be perturbed are small (case B03, $p_r = 1.1$). Both issues originate from a non-representative inner product due to a variation in magnitude between components of the different flow equations. A solution to both issues consists in applying a diagonal scaling matrix to the initial residual vector, in an effort to bring closer the components order of magnitude. Scaling strategies are discussed in [14, 23]. In addition, a study on the performance of the finite difference Jacobian for different values of $\epsilon_{\text{mach}}$, as carried out by Chisholm et al. in [23], could help towards obtaining better performance for large grid cases.

Despite this, the rates for Cases B remain acceptable. The lowest speed-up factor obtained was for case B03 with $p_r = 1.1$, however, in the parametric study ranging $p_r = 1.1 - 3.5$ (see following Section 5.3), all following speed-up factors were similar to the ones obtained for $p_r = 3.5$. For this reason, these suggestions were not carried out in the present work.
Figure 5.10: Nonlinear convergence history against outer iterations: Newton-GMRES and baseline solver - Case A & B for lowest \( (p_r = 1.1) \) and highest \( (p_r = 3.5) \) pressure ratios.
Figure 5.11: Nonlinear convergence history against CPU-time (hours): Newton-GMRES and baseline solver - Case A & B for lowest ($p_r = 1.1$) and highest ($p_r = 3.5$) pressure ratios.
Case C  In this section, the performance of Newton-GMRES is assessed on the same straight-through seal as in case A05 but with more realistic cavity representations. The pressure ratio is set to \( p_r = 3.2 \) for which the flow reaches \( M = 0.58 \) at the exit fin. Higher flow recirculation regions are expected in this case because of the more intricate features of the domain boundaries. Consequently, the number of Newton-Jacobi iterations in the start-up is increased to 150000 iterations to provide a better initial solution for the Newton-GMRES scheme. Again, the starting CFL number for the SER scheme is set to \( \text{CFL} = 100 \), the maximum CFL number is set to \( \text{CFL}_{\text{max}} = 500 \) so that the \( \eta = 0.1 \) linear tolerance is more easily met than in cases A/B, and a smoother transition to the capped \( \text{CFL}_{\text{max}} \) value is prescribed by setting \( \zeta = 1.0 \). Every Krylov vector is preconditioned by applying \( n_cy = 1 \) multigrid V-cycle with the same Jacobi iteration pattern as in cases A & B.

The nonlinear convergence history of the Newton-GMRES method applied to case C is shown in Figure 5.12, full convergence is reached. The Newton-Jacobi reference solver is applied until full convergence of the nonlinear residual. The Newton-GMRES computation time to \( \| \bar{R}(\bar{U}) \|_2 = 3.2 \times 10^{-12} \) was reached in 16.63 hours which represents a 20.68× speed-up factor when compared to Newton-Jacobi.

In the maximum CFL value stages of the SER scheme, the linear solver GMRES preconditioned with \( n_cy = 1 \) multigrid V-cycle on 3 grid levels with \( (\nu(0), \nu(1), \nu(2)) = (2, 2, 4) \), needs to build on average 18 Krylov vectors to meet the linear tolerance set by the forcing term. A comparison is made when using only 1 or 2 grid levels, this is presented in Figure 5.13 in terms of nonlinear residuals against outer iterations and CPU-time. Here, to assess the influence of the third level coarse grid correction, and to ensure that any increase in CPU-time is not induced by the additional Jacobi smoothing iterations on levels 1 and 2 which are larger than level 3 by a factor of \( \approx 4 \times \) and \( \approx 16 \times \), the Jacobi pre/post smoothing iterations are maintained at \( (\nu(0), \nu(1)) = 2 \). The nonlinear residual norm per outer iteration for all cases displays identical behaviour, this was expected as the preconditioner does not affect the Newton correction obtained from a GMRES cycle. A better indicator of the quality of preconditioning is the CPU-time also shown in Figure 5.13, as the preconditioner only affects the linear convergence of GMRES, and so affects the global CPU-time of the Newton-GMRES method. The 3 level case is more efficient by a factor of 1.66× and 1.28× when compared to the 1 and 2 level cases, demonstrating the benefits of the coarse grid correction. Although the cost of preconditioning per Krylov vector is cheaper in the cases with 1 and 2 grid levels, the quality of the preconditioning, due to the absence or reduction in accuracy of the coarse grid correction, imposes more work on the Arnoldi loop to meet the termination criterion. This translates to an increase in the Krylov subspace size in an effort to compensate for the lower preconditioning quality. As a result, the increase in CPU time comes from the cost of the orthogonalization procedure. The GMRES cycles linear convergence are shown in Figure 5.15, where the increase in Krylov subspace for the 1 and 2 level cases can be observed. In the maximum CFL value of the computations, the number of Krylov directions on average increases from 18 vectors for the 3 level case to 47 and 26 vectors for the 1 and 2 level cases respectively.

The case where more preconditioning is applied is assessed by increasing the number of
multigrid cycles to \( n_{cy} = 5 \), the same \((v^{(0)}, v^{(1)}, v^{(2)}) = (2, 2, 4)\) Jacobi iterations pattern is applied, resulting in a preconditioning cost per Krylov vectors \( \approx 5 \times \) as important. The increase in number of multigrid grid V-cycle results in a reduction of the Krylov subspace from 18 to 9 vectors on average in the maximum CFL region, which can be observed in Figure 5.15. However, the gains in CPU time spent in the orthogonalization procedure do not compensate the time spent in the preconditioner, as it can be seen from Figure 5.13, the CPU-time needed for convergence of the nonlinear residual is \( 1.11 \times \) more important as for the case with \( n_{cy} = 1 \). Here, the effects of preconditioning are limited by the approximations made in the first order analytical Jacobians used in the multigrid procedure. The time spent in the multigrid preconditioner could also be reduced by obtaining better load-balancing on the coarser levels, this could be attained by considering element agglomeration strategies when building the prolongation and restriction operators.

Another indicator of the quality of the preconditioning, are the Ritz values obtained by solving the eigenvalue problem defined by the Hessenberg matrix built in the Arnoldi process. For right-preconditioning, the harmonic Ritz values obtained will approximate the first \( m^{th} \) dominant eigenvalues of \( M^{-1}A \), and so ideally, all eigenvalues obtained would be clustered at 1. Again, this is is not possible because of the approximations made in the different grid level Jacobians when compared to the finite difference Jacobian used to compute the Krylov vectors, but also because the linear systems to precondition would need to be solved exactly, resulting in a highly inefficient inexact Newton-GMRES method. Nonetheless, including preconditioning should yield a closer distribution of the eigenvalues around unity, and globally further away from the origin where eigenvalues obtained from ill-conditioned coefficient matrices reside. The Ritz values are obtained by solving the eigenvalue problem defined in Equation (3.36), where \( \tilde{H}_m \) is real for the steady-state case. The eigenvalue distribution for all grid level cases and grid level case 3 with \( n_{cy} = 5 \) are shown in Figure 5.14, they were obtained from the converged solution state. The spectrum of GMRES without preconditioning is also shown for comparison. For the purpose of illustration, the length of the subspace was set to the maximum size \( m = 150 \) which, with the exception of the non-preconditioned case, as aforementioned, is more than the number of Kyrlov vectors needed for meeting the linear tolerance. From Figure 5.14, it can be observed that the eigenvalues of the case without preconditioning are tightly clustered close to the origin. The preconditioned eigenvalues are distributed further away from the origin, and shifted closer towards unity with increasing number of grid levels. In the case where more preconditioning is applied, for \( n_{lev} = 3 \) and \( n_{cy} = 5 \), the outlying eigenvalues are the furthest away from the origin. This case also exhibits an eigenvalue located very close to unity. These results are in agreement with the GMRES linear convergence rates presented in Figure 5.15.
Figure 5.12: Nonlinear convergence history: Newton-GMRES and baseline solver - Case C with $p_r = 3.2$.

Figure 5.13: Newton-GMRES Nonlinear convergence history: Multigrid cases levels $n_{lev} = 1, 2, 3$ with $n_{cy} = 1, 5$ - Case C with $p_r = 3.2$.  
91
Figure 5.14: First 150 approximate eigenvalues of $M^{-1}A$ and $A$ (no preconditioning) obtained from converged solution - Multigrid case levels $n_{lev} = 1, 2, 3$ with $n_{cy} = 1, 5$ - Case C with $p_r = 3.2$.

Figure 5.15: GMRES Linear convergence history - Multigrid cases levels $n_{lev} = 1, 2, 3$ with $n_{cy} = 1, 5$ - Case C with $p_r = 3.2$. 

92
5.3 Flow Analysis

In this section, results for the steady-state flow behaviour of the cases described in Section 5.1 are presented. As the nonlinear function $\bar{R}(U)$ defining the steady-state solution has not been modified, this section aims to provide information about the nonlinear background flow about which the perturbations are linearized in Chapter 6, rather than code validation. Validation work for the steady-state solver in AU3X can be found in [95].

Newton-GMRES is applied to all 6 pressure ratios for cases A & B with identical start-up, SER and multigrid parameters as described in Section 5.2. The nonlinear convergence history per Newton outer-iteration are presented in Figure 5.16, all cases reach full convergence. The convergence history for case C with $p_r = 3.2$ was shown in Figure 5.12.

In Figure 5.17, the flow functions are presented for cases A & B with 0.3mm and 0.5mm seal clearances, these were obtained by fixing the inlet boundary total pressure and temperature and decreasing the pressure at the outlet boundary. An analytical solution obtained from a correlation model based on various reported data is also shown for comparison. The flow functions exhibit a sharp increase from pressure ratios $p_r = 1.1$ to $p_r = 1.4$ before slowly decreasing. At $p_r = 2.4$, sonic conditions at the exit seal fin are on the verge of being reached, this point is attained before $p_r = 3.2$ after which the exit fin is effectively choked. The trends in flow functions obtained from the steady-state solver are consistent with the analytically obtained solution and the CFD analysis carried out by Phibel in [73].

The pressure distributions on the rotor surfaces of seal A03 and A05 at different operating conditions are shown in Figure 5.18. The highest pressure drops occur at the exit fins for $p_r = 2.4, 3.2$ and 3.5. The pressure distributions within the seal are identical for $p_r = 3.2$ and 3.5, as the exit fin is choked, the LP cavity information can not travel back upstream. In the $p_r = 2.4$ case, for which the flow at the exit fin is approaching choked conditions, the pressure distribution within the seal is close to conditions $p_r = 3.2, 3.5$. The flow entering the seal clearance regions is characterized by a momentary pressure increase on the rotor surface, this is due to the annular jets formed at upstream seal clearances, most of the flow is carried over to the following clearance, but a portion impinges the fin creating the static pressure increase. The carry over flow also drives the recirculation regions within the seal cavities. This behaviour can be observed in Figure 5.24, where the computed streamlines are presented for the inner seal region of case A05. Figure 5.22 shows the streamlines of case B05 in the same region, the stator and rotor being stepped, the jets emanating from upstream fins drive two recirculation zones in the following seal cavity, no stagnation point is present on the downstream fins because of the difference their heights which reduces the flow carry over. The static pressure distribution for case B05 with $p_r = 3.5$ is shown in Figure 5.19.

The computed streamlines for the entire computational domain of cases A05 and B05 are presented in Figure 5.23 and 5.21. As expected, case C05 exhibits a more complex flow pattern due to the increased number of recirculation zones present in secondary flow regions (Figure 5.20). Finally, Mach number distributions in the inner seal region of case A05 are presented in Figure 5.25 at different pressure ratios.
Figure 5.16: Nonlinear convergence history - Cases A & B.
Figure 5.17: Cases A & B: flow function against pressure ratios. - Comparison between CFD and analytical method.
Figure 5.18: Static pressure distribution for different pressure ratios on seal rotor - cases A

Figure 5.19: Static pressure distribution - case B05 - $p_r = 3.5$
Figure 5.20: Computed streamlines - Case C05 - $p_r = 3.2$. The velocity field has been scaled by the maximum value.
Figure 5.21: Computed streamlines - Case B05 - $p_r = 3.5$. The velocity field has been scaled by the maximum value.

Figure 5.22: Computed streamlines - Case B05 inner seal volume - $p_r = 3.5$. The velocity field has been scaled by the maximum value.
Figure 5.23: Computed streamlines - Case A05 - $p_r = 3.5$. The velocity field has been scaled by the maximum value.

Figure 5.24: Computed streamlines - Case A05 inner seal volume - $p_r = 3.5$. The velocity field has been scaled by the maximum value.
Figure 5.25: Flow Mach number distribution - Case A05 inner seal volume. The aspect-ratio has been changed for clarity.
Chapter 6

Results II: Linearized Flow

6.1 Introduction

In this Chapter, the results concerning the time-linearized method for flutter predictions are presented. The description of the seal rotor vibrational mode shapes are presented in this section. The following section addresses the matter of method validation by using the nonlinear background flow of Case C described in the previous Chapter. Section 6.3 discusses the numerical aspects of the proposed time-linearized method solution procedure. Finally, results for the flutter parametric study carried out over steady-state pressure ratios and nodal diameter vibrational patterns on cases A & B for both clearances are presented.

The first 5 mode shapes for the seal rotor of cases A and B are presented in Figure 6.1. Their associated natural frequencies are given in Table 6.1. The seal vibration features are the output of an axisymmetric Finite Element analysis which was carried out in the cylindrical coordinate system. Upon inspection of the mode shapes obtained, it was found that no circumferential components were present in the real deflection terms, whereas for the imaginary terms this was the only contribution. The imaginary \( \theta \)-components were however 1 or 2 orders of magnitude smaller than the real terms, and as the tangential deflection components and the forces created by the unsteady pressures are perpendicular, these do no affect the aerodynamic work imposed on the structure (the primary output of interest). The proposed method considers only 2D deflections, and so these contributions were neglected in the following computations, although the code is implemented to support \((z,r)\)-direction imaginary harmonic grid motion. The deflected harmonic grids are obtained following the procedure explained in Section 4.3. The mode shapes for case A were actually obtained from the case C geometry, and the vertex deflections from the base to mid-level of the rotor part were interpolated to ensure no deflection is accounted for at the rotor base vertices.
Figure 6.1: Rotor deformation for the first five nodal diameter numbers: a) Case A. b) Case B.

<table>
<thead>
<tr>
<th>Nodal Diameter</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>± 1</td>
<td>498.8</td>
<td>267.2</td>
<td>483.5</td>
</tr>
<tr>
<td>± 2</td>
<td>563.5</td>
<td>342.5</td>
<td>570.9</td>
</tr>
<tr>
<td>± 3</td>
<td>532.9</td>
<td>824.6</td>
<td>529.6</td>
</tr>
<tr>
<td>± 4</td>
<td>1012.9</td>
<td>1548.8</td>
<td>967.0</td>
</tr>
<tr>
<td>± 5</td>
<td>1685.3</td>
<td>2310.7</td>
<td>1590.3</td>
</tr>
</tbody>
</table>

Table 6.1: Natural frequencies (Hz) for the first five nodal diameters modes - Cases A, B and C.
6.2 Validation

Because of the confidential nature of aero-engine experimental and numerical flutter results, material providing sufficient levels of information needed for setting up validation cases is rather scarce. A time-accurate computation of seal flutter was carried out for case A by Phibel [73] and Phibel et al. [74], however, in these studies only the second nodal diameter $ND = 2$ is considered and the mode shapes were created by introducing a rotor pivot-point and using trigonometric functions to obtain the seal deflections. The case studied by di Mare et al. [30] has been chosen for validation, as it provides results for different nodal diameter wave numbers which correspond to mode shapes obtained from the same FE program used to obtain the ones presented in Figure 6.1. A time-accurate flutter computation was conducted on a straight-through labyrinth seal of radius $R = 0.3\text{m}$ and $0.5\text{mm}$ clearance with realistic HP and LP cavities representations, the steady-state background flow is computed for a pressure ratio of $p_r = 3.2$ across the domain free-stream boundaries, the study was corroborated by experimental results. During rig-tests, the data obtained from the pressure transducers and strain-gauges indicated that the seal rotor was subject to aeroelastic instability for travelling wave numbers $ND = \pm 2, \pm 3, 4$. The strains recorded for nodal diameters $ND = \pm 4$ were irregular, i.e. the labyrinth seal was found to be on the verge of instability. The solver comprised a vertex-base finite volume discretization scheme for the unsteady compressible RANS, turbulent viscosity was obtained by using the 1-equation Baldwin-Lomax model. The computational domain is similar to that of case C, in [30] the 2D domain was discretized into 39000 triangular element before being extruded over a $10^\circ$ sector (with 1 mesh element per degree). The flutter predictions obtained were consistent with the experimental observations.

The validity of the presented time-linearized method is assessed. The steady-state solution for case C and the harmonic grid quantities are used to build the unsteady linear system RHS defined in equation (4.12). In the time-accurate computation by di Mare et al. [30], the ratio between the maximum seal rotor displacement $|\hat{z}_b|$ and the seal clearance $c$ is such that $|\hat{z}_b|/c < 10\%$ for all travelling waves, and so the mode shapes for the linearized flow computations have been scaled accordingly. The nonlinear function sensitivity to nodal displacements and velocities is obtained using finite difference between perturbed and unperturbed second-order in space residual function with the Venkatakrishnan flux limiter. The iterative solution procedure described in Section 3.4 is used to compute the flow harmonic perturbations. The Krylov subspace size is set to $m = 150$ after which GMRES is restarted, no deflation is used in this case. All 3 multigrid levels are used with $n_{cg} = 3$ and identical Jacobi iteration pattern as the steady-state calculations $(\nu^{(0)}, \nu^{(1)}, \nu^{(2)}) = (2, 2, 4)$. The complex operator consists of the conjoint application of the $1^{st}$ order linearized fluxes (about steady-state quantities), the circumferential linearized contribution and the $i\omega_k I$ diagonal terms, on the real and imaginary components of the flow perturbations. The termination criterion for the linear procedure is set at $\epsilon_{tol} = 10^{-10}$, against which the Arnoldi reduced problem RHS last entry (after Givens rotations) is compared.

The linear residual convergence history for the 5 forward travelling nodal diameter waves
is shown in Figure 6.2. All cases converge to the linear tolerance in about 6500 iterations which represents a $\approx 0.62\text{h}$ of computing time, with the exception of the ND = +3 case which converged in $0.82\text{h}$ after $\approx 9000$ iterations. On average this represents 3% of the CPU-time needed for the steady-state computation of case C, the comparatively lower times are obtained due the implemented solution procedure but also because of the linearity of the problem which is not subject to RHS and LHS mismatches.

![Figure 6.2: Linear residual norm of case C with $p_r = 3.2$ for first 5 forward travelling wave number against: a) Krylov vector evaluation (linear iterations). b) CPU-time.](image)

In uncoupled aeroelastic methods where the wall boundary motion is harmonically imposed, the output of interest is generally the amount of aerodynamic work fed to the structure. Positive aerodynamic work indicates energy being passed from the fluid flow to the structure and implies that the structural part of concern will be subject to aeroelastic instability, conversely, negative work will have stabilizing effects. Here, the aerodynamic work imposed on the seal rotor boundary surface $\Gamma_b$ for a vibration cycle of time $T$, is obtained by means of the following equation:

$$W_b = \int_T \int_{\Gamma_b} p_b \cdot v_b \, d\Gamma \, dt \quad (6.1)$$

Where $p_b$ is the outward pointing vector of static pressure magnitude at the rotor boundaries, $v_b$ is the velocity vector of the rotor wall vertices. From the relations in equations
(2.17) and (2.18), the time-domain components can be recast in terms of their mean and harmonic components as:

\[
\begin{align*}
\mathbf{p}_b &= \bar{\mathbf{p}}_b + \Re \left[ \hat{\mathbf{p}}_b e^{i(w_k t + k\theta)} \right] \\
\mathbf{v}_b &= \Re \left[ iw_k \hat{\mathbf{z}}_b e^{i(w_k t + k\theta)} \right]
\end{align*}
\]  

Substituting these relations into equation (6.1) and integrating over a vibration cycle gives:

\[
W_b = w_k \int_T \int_{\Gamma_b} \Re \left[ i\hat{\mathbf{z}}_b^H \hat{\mathbf{p}}_b \right] \, d\Gamma \, dt = -w_k \int_T \int_{\Gamma_b} \left[ \Re(\hat{\mathbf{z}}_b) \Im(\hat{\mathbf{p}}_b) - \Im(\hat{\mathbf{z}}_b) \Re(\hat{\mathbf{p}}_b) \right] \, d\Gamma \, dt \quad (6.3)
\]

It can be recalled that in the present case only the real components of the harmonic grid deflections are non-zero, and so only the first term of equation (6.3) contributes to the aerodynamic work on the rotor surface. In turbomachinery, the aerodynamic work on the structure is more commonly presented by means of the logarithmic decrement \(\delta\), which for a modal mass of 1 is given as:

\[
\delta = -\frac{W_b}{(w_k |\hat{\mathbf{z}}_b|)^2} \quad (6.4)
\]

Where \(|\hat{\mathbf{z}}_b|\) is the maximum seal displacement. The logarithmic decrement is monitored at every GMRES restart iteration. The history of the damping factor throughout the solution procedure is shown in Figure 6.3 against the linear residual norm for all forward travelling wave numbers. From Figure 6.3, it can be seen that at \(l_2 = 10^{-6}\) the final value of \(\delta\) is reached for all nodal diameter numbers, the stage at which this linear residual is reached represents 75.8% of the computation time needed to reach \(\epsilon_{\text{tol}}\). Nonetheless, the termination criterion \(\epsilon_{\text{tol}} = 10^{-10}\) will be maintained for all following computations.

The final logarithmic decrement values for all forward/rearward travelling wave numbers are shown in Figure 6.4a. In 6.4b a comparison is made with the time-accurate computations carried out by di Mare et al. [30], here the values of \(\delta\) have been normalized by their maximum value (for time-linearized and time-accurate methods this occurred at \(\text{ND} = -1\)). The time-linearized method captures the same stability trend as the computation carried out by di Mare et al. Both methods exhibit maximum negative damping at \(\text{ND} = \pm 2\) with the time-linearized method being slightly more sensitive to the direction of the travelling wave. The magnitudes of \(\text{ND} = \pm 4\) obtained with the present method are smaller than those obtained with the validation case, however this is consistent with the experimental observations of the labyrinth seal being only marginally unstable for that particular mode.
Figure 6.3: Logarithmic decrement history against linear residual norm - Case C with \( p_r = 3.2 \). The x-axis has been reversed for clarity.

Figure 6.4: Logarithmic decrement for rearward and forward travelling wave numbers - case C with \( p_r = 3.2 \): a) Time-linearized results. b) Comparison with time-accurate results from di Mare et al., the damping factors have been normalized by their maximum value (\( ND = -1 \) for both cases).
6.3 Performance of Iterative Methods

In this section, different numerical aspects of the proposed solution procedure of the time-linearized are investigated. The effects of the background flow on the convergence of the linear solver is assessed, this is shown for case A05 at a selected nodal diameter number for lower and highest steady-state pressure ratio conditions. The case of stiffer and larger problems is also discussed.

Background flow effects The linearized flow solution procedure is assessed using the background flow of case A05 with \( p_r = 1.1, 3.5 \) for the forward travelling wave number \( ND = 1 \). The parameters of GMRES(m) are set to the same values as in the validation section 6.2. A comparison is also made with the deflated restart method GMRES-DR\((m, k)\) for which the total Krylov maximum size is maintained at \( m = 150 \) and the size of eigenvector subset added to the begining of the following GMRES cycle Krylov basis is fixed at \( k = 30 \).

The number of Krylov vector evaluation needed for GMRES(m) to reach \( \epsilon_{\text{tol}} = 1 \times 10^{10} \) are shown in Figure 6.5a. The \( p_r = 1.1 \) case needs \( 1.59 \times \) more Krylov vector evaluations than for \( p_r = 3.5 \). This behaviour is best explained by considering the first 150 approximate Ritz values of the complex operator \( \hat{A} \) which are shown in Figure 6.5d. These were obtained for the first GMRES cycles of both cases without any multigrid preconditioning, the level of residual decrease is naturally inferior to when preconditioning is included, and so the eigenvalues shown will be a poorer approximations to the true eigenvalues of \( \hat{A} \) than of \( \mathcal{M}^{-1}\hat{A} \) if preconditioning was included. Nonetheless, showing the spectral information this way provides a better comparison of the pressure ratio effects on the conditioning of the coefficient matrix. Both spectrum lie with the unit circle centered at (1,0), and as expected, very close to the origin as no preconditioning is applied. However, the effective center of the eigenvalues of case \( p_r = 3.5 \) is closer to unity which explains the difference in convergence rates shown Figure 6.5a.

From Figure 6.5b, it can be seen that using GMRES(m) with deflation decreases the CPU-time to convergence by a factor of \( 1.66 \times \) for \( p_r = 1.1 \) and \( 1.45 \times \) for \( p_r = 3.5 \). The computations for case A05 are not particularly stiff for the iterative solver, here the gains obtained when using GMRES-DR are mostly due to the first 30 Krylov vector which are obtained from the augmented procedure instead of the regular Arnoldi iteration which includes multigrid iterations. However, the higher benefits observed for \( p_r = 1.1 \) suggest that the gains also come from recycling previous cycle spectral-information, resulting in better linear convergence at the first Arnoldi steps after restart than with GMRES(m).

The logarithmic decrement convergence history for both pressure ratios cases using GMRES(150) and GMRES-DR(150,30) is shown in Figure 6.5c as a function of CPU-time. It can be observed, especially for \( p_r = 1.1 \), that a faster approximate solution is obtained when restarting GMRES in the deflated frame.
Figure 6.5: Influence of the nonlinear background flow on the time-linearized iterative solution procedure with and without deflated restart - case A05 with $p_r = 1.1$ and 3.5 for ND = 1: a) Linear residual norm against Krylov vector evaluation. b) Linear residual norm against CPU-time. c) Logarithmic decrement history against CPU-time. d) First 150 approximate eigenvalues of non-preconditioned complex operator $\hat{A}$
**Linear iterative method stagnation**  The cases investigated so far were handled by the iterative solver without any particular numerical difficulties, in this section the issue of stagnation of the linear iterative procedure, which occurs for particularly ill-conditioned and/or large cases, is addressed.

The parametric computations for which the aeroelastic results are presented in the following section, consisted of a study on cases A & B using steady-state pressure ratio ($n_{pr} = 6$) and vibrational mode shapes ($n_{ND} = 10$) as varying parameters, this amounts to a total of 240 runs. During the numerical study, it was found when using GMRES(150) with 3 grid levels and $n_{cy} = 3$, that in some cases the variations in the solution unknowns stagnate at high residual levels, this is due to the Arnoldi method failing to build the search space from which the approximate solution can be extracted. This breakdown occurred for 3 nodal diameters of cases A03 at $p_r = 1.1$, for 50/60 B05 cases with successful computations for $ND = \pm 1, \pm 2$ for which the excitation frequencies are the smallest, and for all of the largest case B03. The residual levels at which the solver stagnates ranged from $10^{-4}$ to $10^{-7}$, with on average residual levels which were below the tolerance needed for convergence of the rotor damping factor. For cases A03, this issue originates from the ill-conditioning of the complex operator built using the steady-state quantities for the lowest pressure ratio conditions. The non-convergence of the higher nodal diameter of case B05 suggest that, in addition to low pressure background flows, stronger coupling between real and imaginary component of the solution (due to the higher values of $i\omega_k I$) influence the effectiveness of the linear solver. These complications also affect cases B03, but here the stagnation behaviour is also highly affected by the problem grid size for which the maximum Krylov subspace size is not sufficient enough to approximate a solution. A stagnating computation for each of the cases A03, B05 and B03 are selected, the linear residual norms of GMRES(150) with $n_{cy} = 3$ are shown in Figure 6.6.

The natural solution to this problem is to apply more preconditioning to the Krylov vectors: the number of multigrid cycles is increased to $n_{cy} = 10$. Here, the increase of $n_{cy}$ will be more beneficial than in the steady-state problem as the same Jacobian is used to build the Krylov vectors and to precondition them in the multigrid iterations. The linear convergence history obtain using GMRES(150) with $n_{cy} = 10$ is shown for the selected cases in Figure 6.6. The additional preconditioning relieves the stagnation behaviour encountered as all cases now converge. The convergence rates for the largest case B03 are, however, quite poor. Again, this is best explained by considering the eigenvalues distribution of $\mathcal{M}^{-1}\hat{\mathcal{A}}$ which are presented in Figure 6.8 for case A03 and B03 with $n_{cy} = 3, 10$. For the $n_{cy} = 3$ stagnating cases, it can be seen that some approximate eigenvalues lie outside of the unit circle (6 for A03 & 5 for B03). The errors in the solution during a iterative procedure a current step $e_k$, can be related to the initial errors $e_0$ by means of the iteration matrix $e_k = G^k_0 e_0$ defined as $G_h = I - \mathcal{M}^{-1} \hat{\mathcal{A}}$. For a classical iterative method, it can be shown [82] that the solution will converge if the spectral radius of the iteration matrix follows $\rho(G_h) < 1$, i.e the eigenvalues of $\mathcal{M}^{-1}\hat{\mathcal{A}}$ must reside within the unit circle centered at unity. Guaranteed convergence, even if $\rho(G_h) \geq 1$, is a property of GMRES (given a sufficiently large Krylov subspace), nonetheless, outlying eigenvalue

---

1The spectral radius of a matrix $\mathcal{A}$ is defined as $\rho(\mathcal{A}) = \max \lambda \in \sigma(\mathcal{A}) | \lambda |$, where $\sigma(\mathcal{A})$ is the set of all eigenvalues of $\mathcal{A}$
will severely hinder convergence rates. For $n_{cy} = 10$ multigrid iterations, all eigenvalues of the preconditioned complex operator of case A03 reside within the unit circle. On the other hand, despite the additional application of preconditioning iterations, 2 outlying eigenvalues of $\mathcal{M}^{-1}\hat{A}$ remain for case B03, which justifies the convergence behaviour seen in Figure 6.6.

The capabilities of GMRES-DR(150,30) with $n_{cy} = 3$ is assessed, the convergence history is also shown in Figure 6.6. Again, CPU-time is used to compare the performance obtained using GMRES-DR against GMRES(150) with $n_{cy} = 10$ due to the higher preconditioning cost, the fewer Arnoldi iteration per GMRES-DR cycle, and the benefits of spectral information incorporation. This is shown in Figure 6.7. For cases A03 and B05, similar performance is obtained when using deflation at every restart cycle. However, for case B03, incorporating the 30 Krylov vectors associated with the smallest eigenvalues of $\mathcal{M}^{-1}\hat{A}$ decrease the time to $\epsilon_{tol} = 10^{-10}$ by a factor of $3.24\times$. Further increases in performance are obtained when using GMRES-DR(150,30) with $n_{cy} = 10$, case B03 reaches the linear termination criterion in $\approx 5\times$ the CPU-time needed without deflated restart. The first 30 first eigenvalues of $\mathcal{M}^{-1}\hat{A}$ closest to the origin needed to build the augmented quantities for the 1 restarted cycle of GMRES-DR(30,150) are shown in Figure 6.9 for case A03. The dominant eigenvalue associated with the eigenvector shown is Figure 6.10 is also highlighted.
Figure 6.6: Linear residual norm against Krylov vector evaluation using GMRES(150)/GMRES-DR(150,30) with $n_{cy} = 3,10$ for a selection of stagnating cases.
Figure 6.7: Linear residual norm against CPU-time using GMRES(150)/GMRES-DR(150,30) with $n_{cy} = 3,10$ for a selection of stagnating cases.
Figure 6.8: First 150 approximate eigenvalues of preconditioned complex operator $\mathcal{M}^{-1}\hat{A}$ for $n_{cy} = 3, 10$. - Cases A03 and B03.
Figure 6.9: First 150 approximate eigenvalues of preconditioned complex operator $\mathcal{M}^{-1}\hat{A}$ for $n_{cy} = 10$ - Case A03, $p_r = 1.1$, ND = 2. a) 30 first dominant $\lambda(\mathcal{M}^{-1}\hat{A})$ used in GMRES-DR(150,30). b) Enlarged view with first dominant eigenvalue $\lambda(\mathcal{M}^{-1}\hat{A})$, the associated eigenvector is shown in Figure 6.10.

Figure 6.10: Pressure eigenvector magnitude associated with dominant $\lambda(\mathcal{M}^{-1}\hat{A})$ with $n_{cy} = 10$ - Case A03, $p_r = 1.1$, ND = 2. The pressure magnitude has been normalized.
6.4 Flutter Results

The time-linearized equations for cases A & B for both clearances are solved using GMRES-DR(150,30) until \( l_2 < \epsilon_{\text{tol}} \) for all 6 steady-state pressure ratios and forward/rearward nodal diameter wave numbers.

**Flutter predictions** The work imposed by the surrounding unsteady pressures on the labyrinth seal rotor surface is shown by means of the logarithmic decrement \( \delta \) in Figure 6.11. The influence of the background flow on the aeroelastic damping behaviour is clearly noticeable: pressure ratios \( p_r = 3.2, 3.5 \) for which the steady-state flow at the exit fin is choked, have near identical stability values. The \( p_r = 2.4 \) cases for which the flow is on the verge of reaching sonic conditions at the downstream seal clearance, have stability trends which closely follow the behavior of \( p_r = 3.2, 3.5 \). This is consistent with the flow function distribution shown in Figure 5.17. Regardless of the sign of the damping factor, most magnitudes of \( \delta \) increase with increasing pressure ratio. For instance, this can clearly be observed for cases B at ND = \( \pm 1, \pm 5 \), where the damping curves indicate that for the first pressure \( p_r = 1.1 \) the seal rotor is marginally unstable, however instability increases with pressure ratio. Conversely, it can be noticed in cases A at ND = \( \pm 1 \) that higher stability is obtained with increasing pressure ratios. Some exceptions to this behaviour are noticed, such as cases B03 ND = \( \pm 2, \pm 3 \) and B05 ND = \( \pm 3 \) for instance. For cases B03 ND = \( \pm 3 \) this is less of concern as the seal is stable for all pressure ratios. However for the remaining cases, there is a switch in the damping sign from \( p_r = 1.7 \) to 2.4, this point will be addressed subsequently. Cases A are only stable in the first vibration mode ND = \( \pm 1 \), with the rearward travelling wave providing a slightly higher damping values. The logarithmic decrement then decreases to its minimum value at ND = \( \pm 3 \) for which the labyrinth seal will be the most unstable, before tending towards the stability line \( \delta = 0 \) for higher excitation frequencies. These results can be rearranged and presented in terms of the MRTP function. The MRTP is a non-dimensional form of the mass flow rate, it is similar to the flow function in equation 5.1, however the mass flow rate is not normalized by the clearance area \( A_c \). Presenting the results in this way provides a more intuitive way of comparing the flow condition effects on stability. As in Figure 6.11, Figure 6.12 indicates that from the near-incompressible regime to choked conditions, regardless of seal clearance, the seal of cases A at ND = \( \pm 2, \pm 3, \pm 4, \pm 5 \) is fed work from the surrounding fluid, i.e. seal A is prone to aeroelastic instability in these vibrational modes. It is also noticed that with the exception of ND = \( \pm 1 \), the direction of the travelling wave has little effect on the damping values. Figure 6.13 presents logarithmic decrement of cases B with varying MRTP at different nodal diameters. Unlike cases A, the stability trends are not governed by the seal geometry (or rather its vibrational mode shape), as they are more sensitive to the flow features. The smallest magnitudes of \( \delta \) occur for background flow conditions at \( p_r = 1.1 \), with unstable behaviour for ND \( \pm 1 \). The comparatively smaller magnitude obtained at ND \( \pm 5 \) indicates that the seal is likely to be only marginally unstable for these conditions. It can be noticed that the rearward traveling waves produce slightly higher damping magnitudes in the unstable modes.
Figure 6.11: Logarithmic decrements at different steady-state pressure ratios and nodal diameter numbers. - Cases A & B
Figure 6.12: Logarithmic decrement against MRTP for different nodal diameters - cases A
Figure 6.13: Logarithmic decrement against MRTP for different nodal diameters - cases B
Instability mechanisms  A set of cases exhibiting different logarithmic behaviour is selected, these are shown in Figure 6.14 for all pressure ratios. The set consist of case A05 for rearward travelling wave ND = −1 which is stable for $p_r = 1.1$ and follows a monotonic $\delta$ increase with increasing pressure ratio. Conversely, the unstable case B03 with ND = −3 holds a monotonically decreasing $\delta$ with increasing pressure ratio. The rearward/forward ND ± 2 travelling waves for case B03 which display non-monotonic damping behaviour, crossing the stability boundary before or after $p_r = 2.4$ depending on the travelling wave direction. The remaining case is B03 which is stable for all $p_r$ in the ND = −3 vibrational mode but also exhibit a non-monotonic damping factor distribution.

Figure 6.14: Logarithmic decrement against background flow pressure ratios $p_r = 1.1-3.5$ for selected cases.

In his review on the mechanisms causing labyrinth seal failures, Alford [6] discusses two vibration phenomena. The first is the general self-excited flutter case discussed in Section 1.2 for which the pressure fluctuations caused by the seal motion in the interfin seal cavities impose positive work in this particular region, this is then either countered by the forces imposed on the rotor surfaces in the downstream LP and upstream HP cavities, or leads to aeroelastic instability in the event of these forces not managing to compensate the vibration. The second mechanism is caused by an acoustically coupled vibration which occurs when the natural frequency of a given nodal diameter is resonant with frequency of the acoustic oscillations of an adjacent cavity. These acoustic modes are developed from energy which is fed by the pressure fluctuations caused by the change in seal clearance. For resonance conditions, the forces imposed on the downstream/upstream seal surface generally induce a high increase in the vibration amplitude.

The work imposed by the surrounding fluid flow on the labyrinth seal is computed at
different surfaces. This is done at the downstream/upstream seal rotor surfaces, and in the seal interfin region. The results for the selected cases are presented in Figure 6.15 in terms of work contribution-per-location as a percentage of the total work on the rotor $W_b$. Case A05 ND = -1 shows positive interfin work contribution which increases with increasing pressure ratio, however this is compensated by the adjacent cavities, resulting in a stable configuration. Case B03 is unstable in the vibration mode ND = -5, the interfin cavity work is not compensated by the surrounding cavities, this describes the general flutter mechanism which worsens with increasing pressure conditions. Cases B03 for ND±2, which cross the stability line $\delta = 0$ in Figure 6.14, exhibit stabilizing contribution in the interfin cavities for all pressure ratios. The cases are stable from $p_r = 1.1$ up until a threshold which varies depending on the travelling wave direction. The destabilizing contribution from the adjacent cavities are countered by the interfin regions, all magnitude gradually increase until a peak positive amplitude from the downstream cavity which occurs at $p_r = 2.4$. Unlike the rearward travelling wave case, for ND = +2, the interfin seal region still achieves stabilization of the structure. For the remaining pressure conditions, the seal is instable due to the contributions imposed on the HP/LP cavity rotor surfaces. The unsteady pressure amplitude and phase for this case are shown for the ND = -2 case with $p_r = 1.7, 2.4, 3.2$ in Figure 6.16. A change in the sign of the unsteady pressure phase with respect to the seal motion can be observed in the inlet seal inner cavity from $p_r = 1.7$ to 2.4. From Figure 6.15 it can be seen that for the remaining $p_r = 3.2$, the interfin stabilizing contribution then decreases, resulting in a increase in magnitude of the negative damping factor (Figure 6.11). For comparison, Case B03 for ND = -5 which is subject to instability due to the positive work contributions taking place in the interfin region, is shown in Figure 6.17, the pressure amplitudes are higher which justifies the higher negative damping when compared to ND = -2. B03 ND = -3, was found to be stable for all pressure ratios, however the logarithmic decrement distribution was non-monotonic. The work distribution seems to indicate that an acoustic mode, which is not in resonant condition with this particular mode shape frequency, has developed in the downstream cavity, the seal is thus largely stabilized by the interfin region.

These different cases demonstrates the need for downstream/upstream cavity representations in seal flutter predictions as they may either stabilize positive interfin work contributions, or represent the primary source of aeroelastic instability.
Figure 6.15: Work contributions imposed by the surrounding fluid flow on different seal surfaces - Selection of cases showing different aeroelastic mechanisms. Positive work $W > 0$ indicates a destabilizing contribution.
Figure 6.16: Case B03 - $Pr = 1.7$, 2.4 stable & $Pr = 3.2$ unstable, ND = $-2$. a,c) Unsteady pressure amplitude. b,d) Unsteady pressure phase.
Figure 6.17: Case B03 - $p_r = 1.7, 2.4, 3.2$ unstable, ND = $-5$. a,c) Unsteady pressure amplitude. b,d) Unsteady pressure phase.
Influence of clearance and pressure ratio  The influence of labyrinth seal clearance and pressure ratio is discussed. An analytical model developed by Ehrich [32] shows that tighter seal clearances favor instability. This was corroborated by experimental evidence by Ehrich and also by Lewis et al. [54]. A similar result is determined on the influence of pressure ratio across the seal: higher pressure ratios influencing instability. Not all of the results obtained follow these trends, again a selection of different cases is made and presented in Figure 6.18.

Figure 6.18: Logarithmic decrement against background flow pressure ratios \( p_r = 1.1 - 3.5 \) - Influence of seal clearance and pressure ratio.

The unstable cases B03 and B05 (with the exception of \( p_r = 1.1 \)) for ND = +5 are examples which conforms with these studies: both damping factors decreases with increasing pressure ratio, with the tighter seal clearance configuration exhibiting a 1.9\( \times \) factor increase, with respect to the larger clearance, in magnitude for the highest pressure ratio. The work distribution at different rotor surfaces presented in Figure 6.19, clearly indicates an instability due to the general flutter mechanism with destabilizing forces occurring at the seal interfin surface, the higher unsteady pressure amplitudes taking place in the tighter clearance cases causing the higher \( \delta \) values.

The unstable A03 & A05 cases are considered for ND = +2, from Figure 6.18 a non-monotonic distribution of \( \delta \) about pressure ratio is noticed, with slightly smaller damping values for the tighter seal. The work distribution of both seal clearances is shown for the forward travelling wave numbers is shown in 6.20a-b, with the exception of the stable ND = +1 cases, the unstable behaviour obtained for the remaining nodal diameters numbers originate from the upstream HP cavities. From Figure 6.20c-d, it can be seen that the destabilizing contributions from the upstream HP cavity increase with increasing pressure.
ratio. Both cases observe stabilizing contributions from the interfin region. The influence of the surrounding cavities, which is not accounted for in Ehrich’s studies, explains the behaviour of the case \( \delta \) distribution in Figure 6.18. The unsteady pressure amplitude and phase for case A05 ND = +2 is shown in Figure 6.21 for \( p_r = 3.5 \), this is also shown for the stable ND = +1 case for comparison.

Finally, case B for ND = +3 which is stable for the tighter clearance, becomes unstable for the larger clearance from \( p_r = 2.4 \). The work distribution for these cases is shown in Figure 6.22: stabilizing aerodynamic contributions occur in the interfin region of B03 as well as for case B05 until \( p_r = 3.2 \), Case B05 becomes unstable at \( p_r = 3.2, 3.5 \) for which energy is being fed from the surrounding flow to the structure in the interfin seal cavities. Here the explanation for this behaviour going against the aforementioned trends, is that the increase in leakage flow of the background steady-state flow, which increases the steady-state Mach number of the annular jets present at the seal fin tips, is not taken into account by Erich. This behaviour is consistent with the observations made by Phibel [73]. A change in the aerodynamic damping of the downstream LP cavity from negative to positive contribution is observed for case B03 at \( p_r = 1.7-2.4 \) and at \( p_r = 2.4-3.2 \). This induces a unsteady pressure phase sign change which can be observed in Figures 6.23 and 6.24. The phase sign change in the interfin cavities of both clearance, occurs first for the inlet seal cavity (at \( p_r = 2.4 \)) followed by the outlet interfin region at \( p_r = 3.2 \). Resonant conditions between a LP cavity acoustic mode frequency and the free-vibration frequency of the rotor, characterised by a peak work contribution (Figure 6.22), occur at \( p_r = 2.4 \) for the larger clearance case.

From these results, due to possible work contributions imposed on the seal rotor by the surrounding HP/LP cavities and because of the changes in steady-state background flow seal clearance variations can induce, it can be said that general conclusions on the effects of flow conditions and/or seal geometry on stability behaviour are difficult to establish.
Figure 6.19: Work contributions of surrounding fluid flow on different seal locations - Case B03 & B05 ND = +5.

Figure 6.20: Work contributions imposed by the surrounding fluid flow on different seal surfaces - Case A05 ND = 1, 2. Positive work $W > 0$ indicates a destabilizing contribution.
Figure 6.21: Case A05 - $p_r = 3.5$, ND = +1 stable & ND = +2 unstable. a,c) Unsteady pressure amplitude. b,d) Unsteady pressure phase. Aerodynamic work distribution per seal surface is presented in Figure 6.20b.
Figure 6.22: Work contributions of surrounding fluid flow on different seal locations - Case B03 & B05 ND = +3.
Figure 6.23: Case B03 - $p_r = 1.7, 2.4, 3.2$ stable, ND = +3. a,c) Unsteady pressure amplitude. b,d) Unsteady pressure phase. Aerodynamic work distribution per seal surface is presented in Figure 6.22a.
Figure 6.24: Case B05 - $p_r = 1.7$ stable & $p_r = 2.4, 3.2$ unstable, ND = +3. a,c) Unsteady pressure amplitude. b,d) Unsteady pressure phase. Aerodynamic work distribution per seal surface is presented in Figure 6.22c.
Comparison with Abbott criterion  A comparison for cases A & B is made between the flutter predictions obtained using the time-linearized method and the stability criterion developed by Abbott [3]. In his paper, Abbott presents an analysis in which the outcome states that the instability behaviour of a labyrinth seal depends both on the support side and the ratio of interfin acoustic frequency to the mechanical frequency of the seal rotor: a labyrinth seal supported at the exit can only be unstable if the acoustic frequency in the interfin cavities is greater than the seal mechanical frequency. Conversely, a inlet-supported seal is only unstable if the interfin acoustic frequency is less than the mechanical frequency. The acoustic frequency $w_{ac_i}$ in a interfin cavity $i$ of radius $R_i$ is defined as:

$$w_{ac} = n \frac{a_i \pm u_{si}}{2\pi R_i}$$  \hspace{1cm} (6.5)

where $n$ is the nodal diameter number of the assessed mode shape and the sign of the slip velocity $u_s$ is determined by the direction of the travelling wave. Originally, this analysis applied to cantilever-type supported seals, however an analogy was made by Ziegler [97] between the support-side and the smallest change in seal clearance which is determined from the mode shape. For all case A mode shapes, the effective seal support is located at the outlet. The acoustic frequencies of all travelling waves for case A03 and A05 for the highest pressure $p_r = 3.5$ are shown in Table A.1 and A.2 of the appendix. The instability predictions obtained using the time-linearized method conform with Abbott’s analysis.

For cases B, because the rotor arm supporting the seal lies bellow the intermediate fin, the effective support side determined by the mode shapes was found to vary depending on the nodal diameter number. Abbott’s criterion was found to be inapplicable to ND = ±2 due to the change in clearance being equal on both inlet/outlet sides. All the aerodynamic damping results obtained with the present method are in accordance with the criterion with the exception of ND ± 3. For this particular vibration mode, the outlet fin change in clearance was found to be $\approx 1.5$ of the inlet change in clearance, which questions the applicability of this criterion to more intricate seal configurations.
Chapter 7

Conclusion

7.1 Summary of Thesis Achievements

This research was motivated by the need for the aero-engine industry to efficiently conduct labyrinth seal aeroelastic performance studies. To this end, the thesis aimed towards providing a set of numerical methods able to achieve this goal in a reasonable amount of time and computational resources. In previous research, as well as in this thesis, it has been demonstrated that the need for surrounding HP/LP cavity representation in the aeroelastic analysis of labyrinth seal is critical. The flow features inherent to these cavity domains cause various numerical challenges which have been addressed from both the point of view of the methods used to model such problems and from the algorithmic procedures needed to solve them.

From the point of view of the methods, the time-linearized harmonic approach greatly contributes towards obtaining better computational efficiency, by the linearity assumption and the pre-determined frequency of the flow unsteadiness. Because of the spatial-periodicity assumption in the flow circumferential unsteadiness, the harmonic problem can be reduced to two-dimensions. This allows the computation of the steady-state nonlinear background solution to take place in the axisymmetric frame, further reducing the cost of the overall method. The results obtained with the proposed method were consistent with time-accurate computations and experimental observations, validating these assumptions for cavity-seal flutter predictions.

From the algorithmic perspective, Newton-GMRES demonstrated its ability to efficiently obtain fully-converged steady-state turbomachinery cavity solutions. This was attained by using a more accurate evaluation of the nonlinear function linearization conjointly with the application of larger time-steps which were controlled by the progress of the nonlinear solution. The multigrid coarse grid correction, proved to be a reliable way
of preconditioning the Newton linearized systems of such problems, reduced the overall CPU-time. The computation time when using Newton-GMRES on the medium sized grid cases \((N_{\text{node}} \approx 150000)\) resulted in a \(\approx 20 \times\) factor decrease when compared to the benchmark Newton-Jacobi method generally used for turbomachinery cavity-seals. In the larger grid cases \((N_{\text{node}} > 300000)\), while the implemented method produced a notable speed-up (by \(\approx 4 \times\)), additional features (which are discussed further) need to be added to reach similar acceleration factors. These grids are, however, rather large for two dimensional cases and the medium grid sizes studied in this work suffice. This is especially true in the present problem as: 1) Time-linearized flutter methods are less likely to encounter numerical diffusion issues because of the density of the linear system RHS (unlike far-field disturbance cases). 2) Shockwaves, which are known to severely influence flutter predictions for turbomachinery blades, are not present in seal-cavity domains and hence particularly fine grids able to model their position and strength are not required.

For the solution procedure of the time-linearized unsteady equations, Campobasso et al. [18] showed that using multigrid preconditioned GMRES in the pseudo-time frame could stabilize otherwise diverging problems. Effectively, the addition of the pseudo time-step when solving the system in an implicit fashion about the flow perturbation corrections is simply a means of marching the solution using better conditioned coefficient matrices. Chassaing et al. [22] showed that removing the pseudo-time step and using sufficiently large Krylov subspaces could also stabilize linear codes but with better convergence rates. However, this was demonstrated on a transonic duct case using grid sizes which allowed the use of quite large Krylov subspaces, resulting in better conditioned linear systems than in the cavity case. Here, by preconditioning the Krylov vectors using linear multigrid and removing the pseudo time-step, the medium size cases time to convergence took on average 40 minutes on 20 processors, representing 4% of the computation time needed for obtaining the steady-state base flow. For stiffer cases where the complex operators are built using mean flow quantities with lower pressure ratio across the domain free-stream boundaries, the maximum Krylov subspace size was found to be insufficient, resulting in stagnation of the linear solution procedure. This was relieved by increasing the number of multigrid V-cycles. For larger computational grids, while increasing the number of multigrid cycles also overcame the stagnation issues, restarted GMRES encountered difficulties when recomputing the Krylov basis, leading to poor convergence behaviour (\(\approx 6\) hours to meet the termination criterion). By incorporating the eigenvectors associated with the smallest eigenvalues of the preconditioned complex operator at the beginning of the following Krylov basis, GMRES-DR could recover acceptable rates (\(\approx 6 \times\) reduction compared to when increasing the number of multigrid cycle solely). Starting from a steady-state solution, beside the set-up costs associated with obtaining the harmonic grids, the numerical solution procedure proposed for the time-linearized equations permits the computation of all 10 nodal diameters to takes place in about 6 hours of computing time on 20 processors. The same computations for the larger grid cases can be carried out in 10 hours. If only the aerodynamic damping criteria, indicating the occurrence of seal flutter, are of interest, these computing times can further be reduced by \(\approx 80\%\). In comparison, the time needed for the aerodynamic damping criterion to converge for 1 nodal diameter using a time-accurate flutter method on similar sized grids (over a \(10^\circ\) sector) takes on a
equal number of processors $\approx 20$ hours [73], making the proposed methods attractive for large-scale/parametric flutter studies.

This was shown by conducting a study on both straight-through and stepped labyrinth seal configurations, by varying steady-state pressure ratio, clearance and nodal diameter mode shape. A major outcome was the paramount importance of the surrounding seal cavity representation in flutter predictions, as their aerodynamic work contributions can either stabilize labyrinth seals which would otherwise flutter due to the unsteady pressure fluctuations in the interfin region, or represent the main source of aeroelastic instability. The study also showed that the common “increasing-pressure/tighter-clearance favours instability” idea does not always hold, because of the HP/LP cavity contributions, but also because of the changes in the steady-state leakage flow larger seal clearances induce. The trends obtained with the time-linearized method also allows to capture possible peak work contributions from the HP/LP rotor surfaces, characteristic of resonant conditions between cavity acoustic modes and seal mechanical frequency.

### 7.2 Potential Applications

The proposed time-linearized flutter method was originally developed to predict seal rotor instabilities, however, the cyclic symmetry assumption in the flow circumferential unsteadiness makes the method attractive for any flutter problem in annular domains. The most direct application would concern labyrinth seal stator flutter which has also been reported as being an engine development issue [5]. Although it is unlikely that this influences the work fed to the seal structure, the presence of far-field disturbances could be accounted for by implicitly managing the linearization of the incoming signal at the boundary faces of the complex operator. The flutter parametric study carried out in this work considered steady-state pressure ratio, seal clearance and vibrational pattern as varying parameters, however, by varying the excitation frequency, the code could be used to assess the effects of natural frequency change or mechanical damping on the stability trends of a given labyrinth seal. Also, an alternative to using the steady-state solution as the background flow for the time-linearized method, could be to use the time-average of a pre-existing time-accurate solution about which the flow perturbation could then be linearized.

Although Newton-GMRES methods are widely used in the solution of steady-state external flows over NACA airfoils, their use for solving turbomachinery problems remain infrequent. This is partly due to the somewhat more constraining nature of rotating/internal turbomachinery flows which favors the use of more conservative solution procedures, but also because of the level of familiarization needed with the different features of Newton-GMRES which are more case specific. However, it has been shown that while a set of Newton-GMRES parameters able to provide peak performance on a specific case can necessitate some optimization work, a set which allows robust and overall more efficient computations over a range of boundary conditions can certainly be determined. This has been demonstrated in the present work on cavity domains, but could also be applied to turbomachinery blades for instance.
The linear multigrid routine has been shown to be an effective means of preconditioning the Krylov vectors used for solving the linearized systems of the steady-state computation. Multigrid preconditioned GMRES could very well be used in the solution of the linear steps of a time-accurate calculation.

The solution procedure used for solving the unsteady problem could be applied to any linearized flow code. The addition of the deflation methods, being a purely algorithmic feature, could be added to existing GMRES-based time-linearized methods to relieve convergence issues due to large and/or stiff problems.

The implementation of the presented time-linearized method into pre-existing axisymmetric CFD codes is relatively non-intrusive: the iterative solution procedure can be encapsulated, with the exception of the call to the routine which computes the application of the linearized residual function on the real and imaginary parts of the perturbation quantities. The routine computing the linearized circumferential flow contributions does not need to be modified to account for different flux vector splitting schemes, as they are now defined as diagonal source terms. The linear system RHS is computed once prior to the iterative solution procedure. The sensitivity of the residual function to the nodal displacement and velocities is obtained by perturbing the code’s nonlinear function. Other RHS quantities are built in a loop over element and edges which involves harmonic grid deflections and background flow quantities. The most troublesome aspect would be in the preconditioning stage if the code does not have existing multigrid data structures which can be nontrivial to implement and partition. As the complex operator is constant throughout the linear computations, an ILU($m$) factorization could be computed once prior to the iterative process, however, judging by the high number of multigrid cycles needed in some of the cases studied in the present work, the fill-in level $m$ of the approximate factors is likely to be high. Alternatively, in numerical experiments carried out during this work, it has been noticed that using a high number of point-wise Jacobi iterations and restarting GMRES using deflation is sufficient in obtaining acceptable convergence behaviour.

### 7.3 Future Work

While the evaluation of the turbulent viscosity using the 0-equation Cebeci-Smith model is justified in the cavity case due to the absence of flow separation regions, other turbomachinery cases will require the use of more elaborate turbulence models which generally require the solution of 1 (Spalart-Allmaras) or 2 ($\kappa - \omega$) additional equations. These terms have a strong influence on the problem nonlinearities, and so the SER parameters will need to be varied accordingly. The linear residual magnitudes of these additional terms differ highly from the magnitudes of the mean flow residuals, which might lead to the misrepresentation of smaller solution components in the linear model (i.e. the reduced problem obtained by the Arnoldi relation). Should this event occur (this can be checked by explicitly computing the linear residual at the end of a GMRES cycle and comparing the residual drop of the various flow components), a GMRES scaling strategy should be implemented. It has also been pointed out during this work that the absence of scaling
affects the convergence rates of cases with particularly large grids (especially at very low pressure ratio conditions), and the effectiveness of the finite difference perturbation factor. Although in the present work the solver overcame these numerical difficulties, in the case where the steady-state solver should be used for large 3D problem with more a more accurate representation of turbulence, scaling is likely to be required.

Using intersecting areas between overlapping elements of successive grid levels as a weighted average for building the multigrid transfer operators proved to effective. Multigrid convergence rates are less sensitive to the quality of the intergrid transfer operators when used in a preconditioning configuration, however, CPU-times could be reduced by considering more elaborate unstructured coarsening strategies which could provide more optimal load-balancing and processor communication time on all grid levels. For this reason, it would be interesting to compare the present coarsening strategy with element agglomeration methods which consist in building the coarser grid levels by collapsing fine element faces. Element agglomeration methods being a more automatic procedure (although some readjustment in the obtained computational grids might be required), this would also contribute to reducing the computational set-up cost.
Bibliography


Appendix A

Flutter prediction Comparison
with Abbott’s Criteria

This Appendix section provides a comparison of the flutter predictions obtained for cases A & B with the time-linearized method against Abbott’s stability parameter. Abbott states that a labyrinth seal can only be unstable if:

- $w_{ac} < w_k$ for a inlet supported seal.
- $w_{ac} > w_k$ for a outlet supported seal.

Where $w_k$ is the mechanical frequency for a particular mode shape and $w_{ac}$ is the acoustic frequency in the interfin cavities which is given as:

$$w_{ac} = n \frac{a_i \pm u_{si}}{2\pi R_i} \quad (A.1)$$

Where $u_{si}$ and $a_i$ are the split velocity and speed of sound in a seal interfin cavity $i$. The following Tables provide the frequency comparison and effective side of support for both clearance and for the highest steady-state pressure ratio condition $p_r = 3.5$. 

145
<table>
<thead>
<tr>
<th>ND=-1</th>
<th>ND=+1</th>
<th>ND=-2</th>
<th>ND=+2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cavity 1</td>
<td>409.2</td>
<td>382.9</td>
<td>818.3</td>
</tr>
<tr>
<td>Cavity 2</td>
<td>408.3</td>
<td>385.7</td>
<td>816.7</td>
</tr>
<tr>
<td>Cavity 3</td>
<td>406.5</td>
<td>388.1</td>
<td>813.1</td>
</tr>
<tr>
<td>Natural frequency (Hz)</td>
<td>498.8</td>
<td></td>
<td>563.5</td>
</tr>
<tr>
<td>Time-Linearized</td>
<td>S</td>
<td>S</td>
<td>U</td>
</tr>
<tr>
<td>Support-side</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Abbott criterion</td>
<td>n/a</td>
<td>n/a</td>
<td>OK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ND=-3</th>
<th>ND=+3</th>
<th>ND=-4</th>
<th>ND=+4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cavity 1</td>
<td>1227.6</td>
<td>1148.8</td>
<td>1636.8</td>
</tr>
<tr>
<td>Cavity 2</td>
<td>1225.0</td>
<td>1157.1</td>
<td>1633.4</td>
</tr>
<tr>
<td>Cavity 3</td>
<td>1219.7</td>
<td>1164.5</td>
<td>1626.3</td>
</tr>
<tr>
<td>Natural frequency (Hz)</td>
<td>532.9</td>
<td></td>
<td>1012.9</td>
</tr>
<tr>
<td>Time-Linearized</td>
<td>U</td>
<td>U</td>
<td>U</td>
</tr>
<tr>
<td>Support-side</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Abbott criterion</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ND=-5</th>
<th>ND=+5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cavity 1</td>
<td>2046.0</td>
</tr>
<tr>
<td>Cavity 2</td>
<td>2041.7</td>
</tr>
<tr>
<td>Cavity 3</td>
<td>2032.9</td>
</tr>
<tr>
<td>Natural frequency (Hz)</td>
<td>1685.3</td>
</tr>
<tr>
<td>Time-linearized</td>
<td>U</td>
</tr>
<tr>
<td>Support-side</td>
<td>O</td>
</tr>
<tr>
<td>Abbott criterion</td>
<td>OK</td>
</tr>
</tbody>
</table>

Table A.1: Cavity acoustic frequencies against labyrinth seal natural frequencies - Comparison with Abbott’s criteria - A03 for \( p_r = 3.5 \). S = Stable, U = Unstable, I = Inlet, O = Outlet
<table>
<thead>
<tr>
<th>Cavity</th>
<th>ND=−1</th>
<th>ND=+1</th>
<th>ND=−2</th>
<th>ND=+2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>404.1</td>
<td>384.7</td>
<td>808.1</td>
<td>769.5</td>
</tr>
<tr>
<td>2</td>
<td>404.3</td>
<td>387.6</td>
<td>808.6</td>
<td>775.3</td>
</tr>
<tr>
<td>3</td>
<td>403.1</td>
<td>389.5</td>
<td>806.2</td>
<td>779.1</td>
</tr>
<tr>
<td>Natural frequency (Hz)</td>
<td>498.8</td>
<td></td>
<td>563.5</td>
<td></td>
</tr>
<tr>
<td>Time-linearized</td>
<td>S</td>
<td>S</td>
<td>U</td>
<td>U</td>
</tr>
<tr>
<td>Support-side</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Abbott criterion</td>
<td>n/a</td>
<td>n/a</td>
<td>OK</td>
<td>OK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cavity</th>
<th>ND=−3</th>
<th>ND=+3</th>
<th>ND=−4</th>
<th>ND=+4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1212.3</td>
<td>1154.3</td>
<td>1616.4</td>
<td>1539.1</td>
</tr>
<tr>
<td>2</td>
<td>1212.9</td>
<td>1163.0</td>
<td>1617.2</td>
<td>1550.7</td>
</tr>
<tr>
<td>3</td>
<td>1209.4</td>
<td>1168.7</td>
<td>1612.6</td>
<td>1558.3</td>
</tr>
<tr>
<td>Natural frequency (Hz)</td>
<td>532.9</td>
<td></td>
<td>1012.9</td>
<td></td>
</tr>
<tr>
<td>Time-linearized</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
</tr>
<tr>
<td>Support-side</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Abbott criterion</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cavity</th>
<th>ND=−5</th>
<th>ND=+5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2020.5</td>
<td>1923.9</td>
</tr>
<tr>
<td>2</td>
<td>2021.5</td>
<td>1938.4</td>
</tr>
<tr>
<td>3</td>
<td>2015.7</td>
<td>1947.8</td>
</tr>
<tr>
<td>Natural frequency (Hz)</td>
<td></td>
<td>1685.3</td>
</tr>
<tr>
<td>Time-linearized</td>
<td>U</td>
<td>U</td>
</tr>
<tr>
<td>Support-side</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Abbott criterion</td>
<td>OK</td>
<td>OK</td>
</tr>
</tbody>
</table>

Table A.2: Cavity acoustic frequencies against labyrinth seal natural frequencies - Comparison with Abbott’s criteria - A05 for $p_r = 3.5$. S = Stable, U = Unstable, I = Inlet, O = Outlet.
<table>
<thead>
<tr>
<th></th>
<th>ND=-1</th>
<th>ND=+1</th>
<th>ND=-2</th>
<th>ND=+2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cavity 1</td>
<td>381.41</td>
<td>356.32</td>
<td>762.81</td>
<td>700.83</td>
</tr>
<tr>
<td>Cavity 2</td>
<td>386.85</td>
<td>361.88</td>
<td>773.7</td>
<td>723.75</td>
</tr>
<tr>
<td>Natural frequency (Hz)</td>
<td>267.2</td>
<td></td>
<td></td>
<td>342.5</td>
</tr>
<tr>
<td>Time-linearized</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
</tr>
<tr>
<td>Effective Support-side</td>
<td>O</td>
<td>O</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Abbott criterion</td>
<td>OK</td>
<td>OK</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>ND=+3</th>
<th>ND=+4</th>
<th>ND=+5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cavity 1</td>
<td>1144.2</td>
<td>1051.2</td>
<td>1525.6</td>
</tr>
<tr>
<td>Cavity 2</td>
<td>1160.6</td>
<td>1085.6</td>
<td>1547.4</td>
</tr>
<tr>
<td>Natural frequency (Hz)</td>
<td>824.6</td>
<td></td>
<td>1548.8</td>
</tr>
<tr>
<td>Time-linearized</td>
<td>S</td>
<td>S</td>
<td>U</td>
</tr>
<tr>
<td>Effective Support-side</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>Abbott criterion</td>
<td>n/a</td>
<td>n/a</td>
<td>OK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>ND=-5</th>
<th>ND=+5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cavity 1</td>
<td>1907.0</td>
<td>1752.1</td>
</tr>
<tr>
<td>Cavity 2</td>
<td>1934.3</td>
<td>1809.4</td>
</tr>
<tr>
<td>Natural frequency (Hz)</td>
<td>2310.7</td>
<td></td>
</tr>
<tr>
<td>Time-linearized</td>
<td>U</td>
<td>U</td>
</tr>
<tr>
<td>Effective Support-side</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>Abbott criterion</td>
<td>OK</td>
<td>OK</td>
</tr>
</tbody>
</table>

Table A.3: Cavity acoustic frequencies against labyrinth seal natural frequencies - Comparison with Abbott’s criteria - B03 for \( p_r = 3.5 \). S = Stable, U = Unstable, I = Inlet, O = Outlet
<table>
<thead>
<tr>
<th>Cavity 1</th>
<th>ND=-1</th>
<th>ND=+1</th>
<th>ND=-2</th>
<th>ND=+2</th>
</tr>
</thead>
<tbody>
<tr>
<td>377.67</td>
<td>353.44</td>
<td>755.33</td>
<td>706.88</td>
<td></td>
</tr>
<tr>
<td>383.14</td>
<td>364.93</td>
<td>766.27</td>
<td>729.86</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cavity 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>377.67</td>
</tr>
<tr>
<td>383.14</td>
</tr>
<tr>
<td>267.2</td>
</tr>
<tr>
<td>342.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Natural frequency (Hz)</th>
<th>267.2</th>
<th>342.5</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Cavity 1</th>
<th>ND=-3</th>
<th>ND=+3</th>
<th>ND=-4</th>
<th>ND=+4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1133</td>
<td>1060.3</td>
<td>1510.7</td>
<td>1413.8</td>
<td></td>
</tr>
<tr>
<td>1149.4</td>
<td>1094.8</td>
<td>1532.5</td>
<td>1549.7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cavity 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1133</td>
</tr>
<tr>
<td>1149.4</td>
</tr>
<tr>
<td>824.6</td>
</tr>
<tr>
<td>1548.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Natural frequency (Hz)</th>
<th>824.6</th>
<th>1548.8</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Time-linearized</th>
<th>U</th>
<th>U</th>
<th>U</th>
<th>U</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Effective Support-side</th>
<th>O</th>
<th>O</th>
<th>n/a</th>
<th>n/a</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Abbott criterion</th>
<th>OK</th>
<th>OK</th>
<th>n/a</th>
<th>n/a</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Cavity 1</th>
<th>ND=-5</th>
<th>ND=+5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1888.3</td>
<td>1767.2</td>
<td></td>
</tr>
<tr>
<td>1915.7</td>
<td>1824.7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cavity 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1888.3</td>
</tr>
<tr>
<td>1915.7</td>
</tr>
<tr>
<td>2310.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Natural frequency (Hz)</th>
<th>2310.7</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Time-linearized</th>
<th>U</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Effective Support-side</th>
<th>I</th>
<th>I</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Abbott criterion</th>
<th>OK</th>
<th>OK</th>
</tr>
</thead>
</table>

Table A.4: Cavity acoustic frequencies against labyrinth seal natural frequencies - Comparison with Abbott’s criteria - B05 for $p_r = 3.5$. S = Stable, U = Unstable, I = Inlet, O = Outlet