THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING IN THERMO AND FLUID DYNAMICS

Design and Detailed Analysis of Turbomachinery Blades using Truncated Domains

GONZALO MONTERO VILLAR

Department of Mechanics and Maritime Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden August 2020

Design and Detailed Analysis of Turbomachinery Blades using Truncated Domains
GONZALO MONTERO VILLAR

Thesis for the degree of Licentiate of Engineering 2020:14
ISSN 1652-8565
Department of Mechanics and Maritime Sciences
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Chalmers Reproservice
Göteborg, Sweden August 2020

Design and Detailed Analysis of Turbomachinery Blades using Truncated Domains
GONZALO MONTERO VILLAR
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Abstract

With the continuous growth in air traffic that we are seeing nowadays comes an increase in the requirements needed to be satisfied by an aircraft for it to be certified to fly. These stricter regulations affect aspects such as CO_2 emissions, sound pollution and so on, pushing manufacturers to aim for lighter, more efficient, more robust designs. These required improvements needed to keep up with the regulations might come in two different ways; by improving/optimizing existing technology, or by developing new technological concepts. In either of the two scenarios, numerical tools, such as optimization methods or reliable fluid flow simulations play a paramount role.

In this thesis, the new capabilities implemented into the in-house Computational Fluid Dynamics (CFD) solver, G3D::Flow, are described. These new additions have been put in place with the objective in mind of performing broadband noise predictions of a fan/OGV configuration using hybrid RANS/LES simulations. Some of the additions to G3D::Flow include: phase-lagged pitch-wise and rotor-stator interfaces, sliding grids and synthetic turbulence injection. These methods have been then put together in order to simulate the flow around the ACAT 1 fan/OGV geometry.

In this work, an optimization framework called HAMON is presented. It is based on evolutionary algorithms and can be coupled with meta-modeling techniques to speed up processes where computationally expensive simulations need to be performed, such as 3D CFD simulations. HAMON can be used to fine tune an existing design, or as it has been used here, a black-box approach. It has been able to design counter rotating open rotors with more than acceptable performance were no knowledge about propeller aerodynamics was assumed, giving all the design variables more freedom than probably needed. This black-box approach might be specially useful when optimizing new technologies for which no prior knowledge exist, allowing not only to hopefully find good designs but also to show the trends of what a good design should be like.

Keywords: CFD, Sliding grid, Chorochronic, Phase-lagged boundary conditions, LODI, Synthetic turbulence, Stochastic optimization, Evolutionary algorithms

To my family...

Acknowledgements

First of all, I would like to thank Niklas Andersson, not only for giving me the opportunity to embark on this adventure, but also for sharing so much of his knowledge with me. Where would I be without your all your G3D knowledge?

I would like to express my gratitude to all my colleagues at the Division of Fluid Dynamics for all the fun times and the amazing working environment that have made me feel like home despite being several thousands kilometers away. Special mention goes to the G3D::Flow mafia; to Daniel for all your help, advice, fun that we had working, kebab evenings and also those infinite discussions on whose coding style was better, your perfectly order and fully commented one, or my ordered chaos with no white spaces (I am still have not made up my mind). To Elias, who was the first person I met and shared office with when I came into the division for my Master's thesis, who made me feel welcome from the first moment (despite reminding me more often than probably needed that in Iceland killing Basque people was not totally illegal). Also for all your help (not at all with g3dmesh), advice, and fun that I have had working with you. I think Ananda and Sudharsan also deserve a special mention as both had to suffer me during the teaching of the CFD course. I hope you had as much fun sharing that course with me as I had with you.

I would also like to thank my friends outside Chalmers (the Spanish/Latino mafia) who have made it quite easy for me to occupy my mind with non-work related things, and have kept me distracted as well as positive when things were not going very well. Thanks for all the fun times we have had!

Last but not least, to my family. To my parents, Susana and Rafael, for always supporting me in my decisions, including the one that meant moving away from Spain. To my sister, Berta, who is one of the toughest more determined persons I know, for always giving me words of encouragement and believing in me. And finally, to Naia, my niece, you have been one of the best things that ever happened to our family!

This work has been carried out within the project TurboNoiseBB, which is funded by European Union's Horizon 2020 research and innovation program under grant agreement No. 682 690714. The computational resources used have been provided by the Swedish National Infrastructure for Computing (SNIC) and the Chalmers Centre for Computational Science and Engineering (C^3SE) in Sweden.

Gonzalo Montero Villar
Göteborg, August 2020

NOMENCLATURE

Greek letters

ρ	density
τ_{ij}	viscous stress tensor
ρe_0	total energy density
μ	dynamic viscosity
λ_i	characteristic speed of the i^{th} wave
σ	coupling parameter in LODI outlet, length scale in SEM
Ω_i	rotational speed of the i^{th} row
β_i	phase shift of the i^{th} row
Δt	simulation time step
ϵ^k	intensities of the k^{th} eddy
$\phi^n,\ \psi^n,\ \theta^n,\ \alpha^n$	random angles used in Fourier based synthetic turbulence
κ	wave number

Roman letters

u_i	i^{th} component of the velocity vector
x_i	i^{th} component of the position vector
p	pressure
k	thermal conductivity
T	temperature
C_p	specific heat
P_r	Prandtl number
c	speed of sound
p_∞	desired pressure in LODI based outlet
NO	number of overlapping cells in GGI/sliding grid
$q(i)$	flow state of cell i
f	relative blade passing frequency
N_i	number of blades of i^{th} row
T_{shift_i}	time shift of i^{th} row
N_h	number of harmonics to consider on truncated Fourier series
$\hat{q}_n$	n^{th} temporal Fourier coefficient
t_s	simulation time
j	imaginary unit
$q(t)$	flow variables at time t
$Q(t)$	flow variable reconstructed using Fourier coefficients at time t
N_{tspp}	number of time steps per period
$\hat{q}_{n,k}$	time azimuthal Fourier coefficient
$\mathbf{U}$	eddy convective velocity in SEM
R_{ij}	Reynolds stress tensor

N_k	number of eddies
V_{BOX}	volume of the box of eddies
$\mathbf{x}^k$	position of the k^{th} eddy
$\mathbf{u}'$	synthetic velocity fluctuation
a_{ij}	Cholesky decomposition of R_{ij}
$\hat{u}^n$	amplitude of the n^{th} Fourier coefficient on synthetic turbulence
$E(\kappa)$	energy contained at κ
$a,\ b$	time correlation filter variables
T_{int}	integral time scale
N	number of objective functions
o_m	value of the m^{th} objective function
cd	crowded distance

Abbreviations

CFD	Computational Fluid Dynamics
NSCBC	Navier-Stokes Characteristic Boundary Condition
LODI	Local One Dimensional Inviscid
(U)RANS	(Unsteady) Reynolds Averaged Navier-Stokes
GGI	General Grid Interface
LES	Large Eddy Simulation
DNS	Direct Numerical Simulation
SEM	Synthetic Eddy Method
EA	Evolutionary Algorithm
RBF	Radial Basis Function
CROR	Counter Rotating Open Rotor
LHS	Latin Hypercube Sampling
GA	Genetic Algorithm
DE	Differential Evolution
(D)DES	(Delayed) Detached Eddy Simulation
OGV	Outlet Guide Vane

Thesis

This thesis consists of an extended summary and the following appended papers:

Paper A D. Lindblad, G. Montero Villar, N. Andersson, A. Capitao Patrao, S. Courty-Audren, and G. Napias. “Aeroacoustic Analysis of a Counter Rotating Open Rotor Based on the Harmonic Balance Method”. *2018 AIAA Aerospace Sciences Meeting*. 2018

Paper B G. Montero Villar, D. Lindblad, and N. Andersson. “Multi-Objective Optimization of an Counter Rotating Open Rotor using Evolutionary Algorithms”. *2018 Multidisciplinary Analysis and Optimization Conference*. 2018

Paper C G. Montero Villar, D. Lindblad, and N. Andersson. “Effect of Airfoil Parametrization on the Optimization of Counter Rotating Open Rotors”. *AIAA Scitech 2019 Forum*. 2019

Paper D G. Montero Villar, D. Lindblad, and N. Andersson. “Investigation of Phase-Lagged Boundary Conditions for Turbulence Resolving Turbomachinery Simulations”. *AIAA AVIATION 2020 FORUM*. 2020

Other publications:

Paper E D. Lindblad, N. A. Wukie, G. Montero Villar, and N.s Andersson. “Implementation of a Quasi-Three-Dimensional Nonreflecting Blade Row Interface for Steady and Unsteady Analysis of Axial Turbomachines”. *2018 AIAA/CEAS Aeroacoustics Conference*. 2018

Paper F D. Lindblad, N. Wukie, G. Montero Villar, and N. Andersson. “A Nonreflecting Formulation for Turbomachinery Boundaries and Blade Row Interfaces”. *AIAA Scitech 2019 Forum*. 2019

Paper G A. Capitao Patrao, T. Grönstedt, A. Lundbladh, and G. Montero Villar. Wake Analysis of an Aerodynamically Optimized Boxprop High-Speed Propeller. *Journal of Turbomachinery* (2019)

Contents

Part I
Extended Summary

Chapter 1

Introduction

Predicting the future with absolute certainty is more often than not, impossible. Nevertheless, when it comes to the evolution of air traffic, one thing seems to be clear regardless of which study is analyzed, it is growing and will continue to grow in the foreseeable future. For instance, according to Airbus' global market forecast, the passenger fleet is expected to double by year 2035 in comparison to 2015 and the passenger traffic is expected to grow at a rate of 4.5% CAGR over the same period of time [1]. This increase in air traffic is expected to have considerable environmental and economic impact.

With these predictions in mind, the European Commission has set some goals for the technologies that should be available by the year 2050. Some of these goals focus on the environmental impact of aviation, such as having air vehicles which design and manufacturing allows them to be recycled, a 75% reduction of CO_2 emissions per passenger kilometer, a 65% reduction in perceived noise emissions and a 90% reduction in NO_x emissions with respect to the year 2000 [15]. On the other hand, taking a look at the economic side of it, the International Air Transport Association (IATA) estimated in their annual report, that 24% of all airlines operating costs in the year 2018 were due to jet fuel [20].

Taking these goals and predictions into consideration, it comes as no surprise that the aviation industry is putting some major efforts into improving their products. These efforts have yielded significant improvements over the past decades, much of it thanks to the advances in numerical simulation techniques, such as more reliable and accurate Computational Fluid Dynamics (CFD), as well as a huge increase in computational power. In order to be able to keep up with the regulations, further improvements are needed, which might be achieved in two different ways, by continuously improving designs of existing technologies (which is becoming harder and harder) or by developing new technologies that will enable radical new designs.

Regardless of which of the two ways improvements come, the use and development of computational tools seems to be a key aspect. For instance, more accurate CFD tools which better predict the flow over a certain turbomachinery component, can be helpful not only in the design and improvement process of that particular component, but also in the understanding and learning about the complex fluid physics phenomena that take place, which might lead to the development of a new concept. Moreover, design and optimization tools that do not rely on analytical/empirical methods come in handy, specially when new concepts are being developed. One example of the application of such methods where a new concept could not be designed using analytical methods, as they were not existing, can be seen with the Boxprop [3]. In this case, a fully automated optimization framework that assumed almost no existing knowledge about aerodynamics was able to produce satisfactory designs [8].

1.1 Aim

The objective of this work is to expand and improve the capabilities of the in-house compressible CFD solver, G3D::Flow[1], with the ultimate aim of being able to perform broadband noise simulations of the ACAT 1 fan. The added functionality to G3D::Flow include phase-lagged periodicity, rotor-stator interfaces and synthetic turbulence methods among others. These simulations and acoustic predictions will later be compared and validated using data obtained during an experimental campaign within the European project TurboNoiseBB[2].

An optimization platform, HAMON[3], was also developed. It is based on evolutionary algorithms with the capability of using meta-modelling for speeding up computationally demanding problems, and can be freely obtained online. This tool has been used to optimize several turbomachinery components within the Division of Fluid Dynamics at Chalmers University of Technology.

[1] Webpage: `https://nikander.github.io/g3dflow`

[2] Validation of improved turbomachinery noise prediction models and development of novel design methods for fan stages with reduced broadband noise

[3] Available at `https://github.com/gmonterovillar/HAMON`

Chapter 2
Methodology

In order to perform fluid flow simulations, the in-house solver G3D::Flow is used, which is based on the family of codes developed by Eriksson [14]. G3D::Flow is a finite volume compressible CFD solver, developed and maintained at the division of Fluid Dynamics at Chalmers University of Technology, which uses a three-stage second-order accurate Runge-Kutta algorithm for advancing the solution in time. Regarding the spatial discretization, an upwind-biased third-order accurate scheme is used for the convective fluxes and a second-order accurate central differencing scheme for the diffusive ones. More information on G3D::Flow can be found at [16, 2]

2.1 Governing equations

The equations that dictate the behaviour of fluid flows are usually referred to as Navier-Stokes equations. Here they are presented in their viscous compressible form, for a Newtonian fluid, using tensor notation, in Cartesian coordinates and neglecting body forces. These equations are individually referred to as continuity (2.1), momentum (2.2) and energy (2.3) equations.

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} = 0 \tag{2.1}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial (\rho u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \tag{2.2}$$

$$\frac{\partial (\rho e_0)}{\partial t} + \frac{\partial [(\rho e_0 + p) u_j)]}{\partial x_j} = \frac{\partial (u_i \tau_{ij})}{\partial x_j} + \frac{\partial}{\partial x_j}\Big(k \frac{\partial T}{\partial x_j}\Big) \tag{2.3}$$

where τ_{ij} is the viscous stress tensor, ρe_0 is the total energy density and k is the thermal conductivity, which are defined as

$$\tau_{ij} = \mu\Big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\Big) \tag{2.4}$$

$$\rho e_0 = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u_k u_k \tag{2.5}$$

$$k = \frac{\mu C_p}{P_r} \tag{2.6}$$

where μ is the dynamic viscosity, C_p is the specific heat and P_r is the Prandtl number.

2.2 Boundary treatment

When performing a CFD simulation, the domain is usually trimmed to a region of space which is of special interest. This results in having to apply boundary conditions in the artificially created boundaries of the domain, which will differ depending on the nature of the flow and the conditions under which it is simulated. In order to save computational power the boundaries are placed as close as possible to the area of interest without compromising the results. Correct implementation and use of the boundary conditions is fundamental in order not to contaminate the numerical solution obtained inside of the domain, for instance, by generating nonphysical upstream traveling pressure waves on an outlet that may destroy the solution.

2.2.1 Navier-Stokes Characteristic Boundary Conditions

In this section, the method of imposing boundary conditions referred to as Navier-Stokes Characteristic Boundary Conditions (NSCBC) by Poinsot and Lele[32] is outlined. As the names suggests, the boundary conditions are imposed using the method of characteristics. The different waves travelling across the boundary are assumed to be the same ones that would occur on an Euler simulation (i. e. waves related to viscous phenomenon are discarded). The description is made assuming boundaries normal to the x_1 direction, being the application to x_2 and x_3 direct.

Starting from the governing equations previously presented (Eqs. 2.1, 2.2 and 2.3), one can write an equivalent set of equations in which the terms involving the waves propagating though the boundary can be identified (note that Eq. 2.2, the momentum equation, has been expanded into its three components)

$$\frac{\partial \rho}{\partial t} + d_1 + \frac{\partial(\rho u_2)}{\partial x_2} + \frac{\partial(\rho u_3)}{\partial x_3} = 0 \tag{2.7}$$

$$\frac{\partial \rho u_1}{\partial t} + u_1 d_1 + \rho d_3 + \frac{\partial(\rho u_1 u_2)}{\partial x_2} + \frac{\partial(\rho u_1 u_3)}{\partial x_3} = \frac{\partial \tau_{1j}}{\partial x_j} \tag{2.8}$$

$$\frac{\partial \rho u_2}{\partial t} + u_2 d_1 + \rho d_4 + \frac{\partial(\rho u_2 u_2)}{\partial x_2} + \frac{\partial(\rho u_2 u_3)}{\partial x_3} = -\frac{\partial p}{\partial x_2} + \frac{\partial \tau_{2j}}{\partial x_j} \tag{2.9}$$

$$\frac{\partial \rho u_3}{\partial t} + u_3 d_1 + \rho d_5 + \frac{\partial(\rho u_3 u_2)}{\partial x_2} + \frac{\partial(\rho u_3 u_3)}{\partial x_3} = -\frac{\partial p}{\partial x_3} + \frac{\partial \tau_{3j}}{\partial x_j} \tag{2.10}$$

$$\begin{gathered} \frac{\partial(\rho e_0)}{\partial t} + \frac{1}{2} u_k u_k d_1 + \frac{d_2}{\gamma - 1} + \rho u_1 d_3 + \rho u_2 d_4 + \rho u_3 d_5 + \\ \frac{\partial[(\rho e_0 + p) u_2)]}{\partial x_2} + \frac{\partial[(\rho e_0 + p) u_3)]}{\partial x_3} = \frac{\partial(u_i \tau_{ij})}{\partial x_j} + \frac{\partial}{\partial x_j}\left(k \frac{\partial T}{\partial x_j}\right) \end{gathered} \tag{2.11}$$

contributions from variations in direction normal to the boundary have been replaced by terms which depend on the vector d_i (which can be expressed using characteristic analysis

[39]) as

$$\begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{bmatrix} = \begin{bmatrix} \frac{1}{c^2}\left[\mathscr{L}_2 + \frac{1}{2}(\mathscr{L}_1 + \mathscr{L}_5)\right] \\ \frac{1}{2}(\mathscr{L}_1 + \mathscr{L}_5) \\ \frac{1}{2\rho c}(\mathscr{L}_5 - \mathscr{L}_1) \\ \mathscr{L}_3 \\ \mathscr{L}_4 \end{bmatrix} = \begin{bmatrix} \frac{\partial(\rho u_1)}{\partial x_1} \\ \frac{\partial(c^2 \rho u_1)}{\partial x_1} + (1-\gamma)\mu\frac{\partial p}{\partial x_1} \\ u_1\frac{\partial u_1}{\partial x_1} + \frac{1}{\rho}\frac{\partial p}{\partial x_1} \\ u_1\frac{\partial u_2}{\partial x_1} \\ u_1\frac{\partial u_3}{\partial x_1} \end{bmatrix} \tag{2.12}$$

where c is the speed of sound and $\mathscr{L}_i$ are the characteristic wave amplitudes defined by

$$\begin{bmatrix} \mathscr{L}_1 \\ \mathscr{L}_2 \\ \mathscr{L}_3 \\ \mathscr{L}_4 \\ \mathscr{L}_5 \end{bmatrix} = \begin{bmatrix} \lambda_1\left(\frac{\partial p}{\partial x_1} - \rho c\frac{\partial u_1}{\partial x_1}\right) \\ \lambda_2\left(c^2\frac{\partial \rho}{\partial x_1} - \frac{\partial p}{\partial x_1}\right) \\ \lambda_3\frac{\partial u_2}{\partial x_1} \\ \lambda_4\frac{\partial u_3}{\partial x_1} \\ \lambda_5\left(\frac{\partial p}{\partial x_1} + \rho c\frac{\partial u_1}{\partial x_1}\right) \end{bmatrix} \tag{2.13}$$

where λ_i are the characteristic speeds of the waves with amplitude $\mathscr{L}_i$. These five waves can be interpreted as two acoustic waves (one travelling upstream with λ_1 and one downstream with λ_5) and two vorticity and an entropy wave (these three travelling at $\lambda_2 = \lambda_3 = \lambda_4$).

$$\lambda_1 = u_1 - c; \qquad \lambda_2 = \lambda_3 = \lambda_4 = u_1; \qquad \lambda_5 = u_1 + c \tag{2.14}$$

Based on these characteristic speeds one can determine whether a wave is leaving or entering the domain. For instance, at an inlet were the velocity is in the positive x_1 direction, there are four incoming waves and one outgoing with speed λ_1. This information can be used to formulate boundary conditions, and the solution can be advanced in time using Eqs. 2.7, 2.8, 2.9, 2.10 and 2.11 if the value of the $\mathscr{L}$'s can be determined. As suggested in [32] and [42] the characteristic amplitudes corresponding to outgoing waves are determined from one-sided derivatives using interior points.

2.2.2 Local One Dimensional Inviscid relations

Due to the lack of a method to compute the exact values of the incoming $\mathscr{L}$'s in the general case, the problem is reduced to considering Local One Dimensional Inviscid (LODI) relations. As the name implies, these relations can be obtained by dumping every term related to the x_2 and x_3 dimensions, as well as viscosity from Eqs. 2.7, 2.8, 2.9, 2.10 and 2.11, as well as substituting the d's according to Eq. 2.12, resulting in the following LODI relations. Here expressed as a function of the primitive variables

$$\frac{\partial \rho}{\partial t} + \frac{1}{c^2}\left[\mathscr{L}_2 + \frac{1}{2}(\mathscr{L}_1 + \mathscr{L}_5)\right] = 0 \tag{2.15}$$

$$\frac{\partial u_1}{\partial t} + \frac{1}{2\rho c}(\mathscr{L}_5 - \mathscr{L}_1) = 0 \tag{2.16}$$

$$\frac{\partial u_2}{\partial t} + \mathscr{L}_3 = 0 \tag{2.17}$$

$$\frac{\partial u_3}{\partial t} + \mathscr{L}_4 = 0 \tag{2.18}$$

$$\frac{\partial p}{\partial t} + \frac{1}{2}(\mathscr{L}_1 + \mathscr{L}_5) = 0 \tag{2.19}$$

These relations can also be combined to express time or spatial derivatives of other quantities in order to simplify the imposition of boundary conditions.

Subsonic outlet

In the case of an outlet, there is only one wave going into the domain with an amplitude $\mathscr{L}_1$, the rest of the amplitudes can be determined from interior points using one-sided derivatives as aforementioned. Once all the $\mathscr{L}$'s have been determined, Eqs. 2.7, 2.8, 2.9, 2.10 and 2.11 are used to perform the iteration. Different types of subsonic outlets can be obtained:

1. **Reflecting outlet**: one way of obtaining an exact reflection of the waves at the outlet is to keep the pressure at the outlet constant ($\partial p/\partial t{=}0$). This together with one of the LODI relations, Eq. 2.19, allows to calculate the amplitude of the incoming wave
$$\mathscr{L}_1 = -\mathscr{L}_5 \tag{2.20}$$
where $\mathscr{L}_5$ as well as $\mathscr{L}_2$, $\mathscr{L}_3$ and $\mathscr{L}_4$ has been computed from interior.

2. **Perfectly non-reflecting outlet**: in order to avoid reflections one can set the amplitude of the incoming wave to zero, $\mathscr{L}_1 = 0$. The problem with this implementation, is that since no information is travelling inside the domain, it prevents from setting the right pressure level.

3. **Partially-reflecting outlet**: due to the problem described with the perfectly non-reflecting outlet, some small reflections inside the domain are allowed in order to be able to set the correct pressure level. This is achieved by computing the amplitude of the entering wave as [32]
$$\mathscr{L}_1 = K(p - p_\infty) \tag{2.21}$$
p_∞ being the desired pressure. The used form of K is as described in [34]
$$K = \frac{\sigma(1 - M^2)c}{L} \tag{2.22}$$
where the coupling parameter σ is set to a default value of 0.58, M is the Mach number of the flow and L a representative length of the domain.

Implementation

In order to implement the subsonic outlet boundary condition just described into G3D::Flow, the fluxes normal to the outlet boundary are not computed and instead, they are substituted with a source term for each of the conservation equations as:

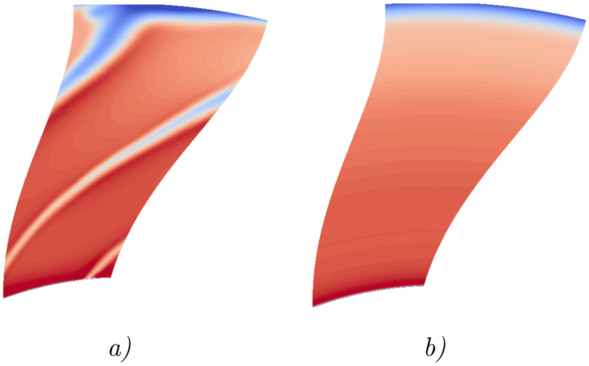
$$S = \begin{bmatrix} d_1 \\ u_1 d_1 + \rho d_3 \\ u_2 d_1 + \rho d_4 \\ u_3 d_1 + \rho d_5 \\ \frac{1}{2} u_k u_k d_1 + \frac{d_2}{\gamma - 1} + \rho u_1 d_3 + \rho u_2 d_4 + \rho u_3 d_5 \end{bmatrix} \Delta V \tag{2.23}$$

where ΔV is the cell volume, and the five source terms in Eq. 2.23 are introduced in the continuity, three momentum and energy equations respectively.

2.3 Turbomachinery simulations

When performing turbomachinery-like CFD simulations, it is quite common to encounter a situation where two or more blade rows need to be simulated, which might or might not be in relative motion to one another. The most widely used approach to perform these computations is to use a mixing-plane interface together with rotational periodic boundary conditions for steady state problems. If this is the case, the simulation set up is the same independently of the blade count in each of the rows and the mesh structure at the interface in between them, as a mixing-plane will perform an azimuthal average of the flow properties in order to transfer information between domains (in the case where the blade rows are in relative motion, a change in frame of reference is also performed). By using this approach, one major drawback is the fact that a lot of the information contained in one domain is lost when transferred to the adjacent one. This is illustrated in Fig. 2.1, where a) represent the flow field right upstream of the mixing-plane, and b) what the downstream domain receives after the flow field has gone through the averaging process performed by the mixing-plane.

After seen the example illustrated by Fig. 2.1, one can clearly see that even though this approach might be useful to perform steady state computations, it is not desirable on a transient simulation, where for instance the impact of the wake impinging a second blade row is sought for, as that information is lost. When these type of simulations are to be carried out, one set up that is always valid is to simulate the entire 360° wheel, but this becomes, more often than not, prohibitive due to computational resources. If the full wheel simulation is to be avoided, two different scenarios exist. In the first one, both domains cover the same pitch, this can be achieved by using one blade passage in each domain if the blade count is the same, or by using a different number of blade passages on each domain which add up to covering the same tangential sector (for instance one rotor and two stator blades in a configuration with 20 by 40 blades). In the second scenario, the blade count is different and using a combination of blade passages that cover the same tangential sector is not possible (either because there is not an existing configuration, i.e. if one of the blade counts is a prime number, or because it requires of too many blades to be simulated, being the computational resources a limiting factor).



a) *b)*

Figure 2.1: *Generic wake on a RANS turbomachinery CFD simulation; a) flow upstream of the mixing-plane, b) information transferred to the downstream domain by the azimuthal average performed by the mixing-plane.*

In the first case scenario described (same tangential sector) a solution not to lose the information via the averaging procedure is to use a General Grid Interface (GGI) when the blade rows are not in relative motion, or a sliding-grid interface when a relative motion exist. This allows for the use of standard rotational periodicity in the pitch-wise boundaries. In the second scenario (different tangential sector), the so called phase-lag boundary conditions can be used to avoid the full wheel simulation. In the following subsections this two approaches are outlined.

2.3.1 General grid interface, sliding grid

When the tangential sector covered by two domains is the same, one can use a GGI or sliding grid (depending on whether they are in relative motion or not) to transfer the information between them. One can think about the GGI as an interpolator which connects cells on both sides of the interface in order to perform the flux calculation. Figure 2.2 shows an schematic on how the mesh on two sides of a GGI might look. As it can be seen, there is no need to have matching nodes, or same cell count and the domain can be shifted by an angle (like in the figure shown). The sliding grid interface follows the same concept, with the difference that both the upstream and downstream domains are in relative motion to one another, therefore every time step the overlap between cells needs to be recomputed (to account for the rotations) as well as a change in frame of reference of the flow state when introducing values in the ghost cells for flux calculation.

Implementation

G3D::Flow uses a ghost cell approach to treat the interfaces, and when using a third order accurate upwind biased scheme for convective fluxes, two cells both upstream and downstream of the face across which the flux is being calculated (the face in between $c2$ and $c3$ as depicted in Fig. 2.3 a) and b)) are utilized. The first step is to find all the cells on the other side of the interface that overlap with a given cell (both layers, i.e. $c3$

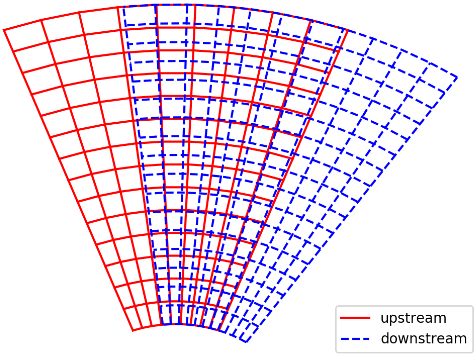


Figure 2.2: *Mesh distribution schematic on both sides of a GGI.*

and $c4$ in Fig. 2.3 a)) and the overlapping area is calculated. This is done only once as a preprocessing step in case of a GGI being used. On the other hand, if a siding grid interface is used, the overlap is checked for every iteration as the domains are fictionally rotated with different angular speeds. When computing the fluxes on the upstream domain (left of the interface in Fig. 2.3) two different strategies are adopted; one for the face which is located at the interface (Fig. 2.3 a)) and another one for the face on step inside the domain (Fig. 2.3 b)).

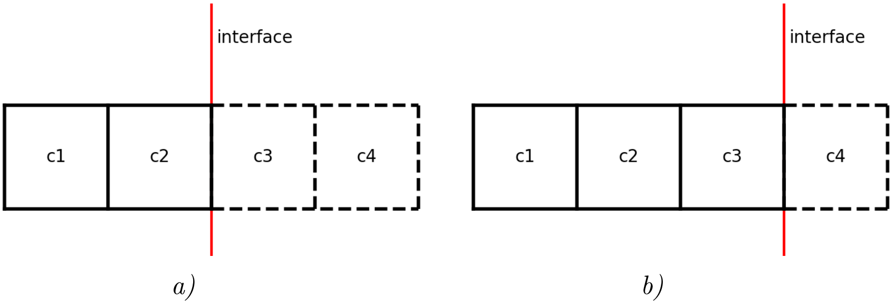


Figure 2.3: *Schematic of the cells used when computing fluxes across an interface in G3D::Flow. Cells drawn with dashed lines represent ghost cells. a) interface face and b) face one step inside the domain.*

For the flux calculation corresponding to the face at the interface (Fig. 2.3 a)) several flux calculations are performed to ensure that the interface is conservative as,

$$F_{c2i} = \sum_{n=1}^{NO_i} \mathcal{F}\Big(q(c1_i), q(c2_i), q(c3_{in}), q(c4_{in})\Big) \frac{A_{in}^{intersec}}{A_i} \tag{2.24}$$

where the subscript i refers to the face in question, F_{c2i} the resultant flux onto cell $c2$ though face i, NO_i the amount of neighbouring cells overlapping, $c3_{in}$ and $c4_{in}$ are the

n^{th} cell overlapping face i closer to the interface and one step in the neighbouring domain respectively (see Fig. 2.3 a)), $A_{in}^{intersec}$ is the overlapping area between face i and cell $c3_{in}$, A_i is the area of face i, $\mathcal{F}$ is the convective flux operator which takes four cells as input and $q(j)$ is the flow state of cell j.

For the flux computation one step inside the domain (see Fig. 2.3 b)) a unique flux calculation is performed by introducing an average flow state of all the overlapping cells in the ghost cell as,

$$F_{c3i} = \mathcal{F}\Big(q(c1_i), q(c2_i), q(c3_i), \sum_{n=1}^{NO_i} q(c4_{in}) \frac{A_{in}^{intersec}}{A_i}\Big) \tag{2.25}$$

If the sliding grid is used, the flow state introduced into the ghost cells are transformed to the frame of reference of the opposite domain.

2.3.2 Chorochronic method

When the tangential sector covered is not the same, one can use the so called phase-lag boundary conditions. One of such methods is the chorochronic method, for which a short overview is given here, for more details the interested reader is referred to [17]. Two different treatments have to be performed with this method in order to be applicable to turbomachinery simulations, namely; the pitchwise boundaries and the blade row interface. The notation used here mostly follows that on [30].

Pitch-wise boundaries

One can compute the relative blade passing frequency for the first blade row as

$$f_1 = \frac{N_2|\Omega_1 - \Omega_2|}{2\pi} \tag{2.26}$$

where Ω_i and N_i represent the rotational speed and the number of blades of the i^{th} blade row respectively. Indexes 1 and 2 must be swapped in Eq. 2.26 in order to compute the relative blade passing frequency for the second blade row, f_2. This is the frequency at which a given blade sees the opposite row's blades, and hence the frequency related to the periodicity of the flow properties on the former blade row (leaving aside turbulent flow instabilities which might have their own frequency). Due to the difference in pitch between the blade rows, two continuous blades see the same flow field with a phase shift, β, which can be computed as [17],

$$\beta_1 = -\frac{2\pi(N_1 - N_2)\text{sign}(\Omega_1 - \Omega_2)}{N_1} \tag{2.27}$$

Again, swapping indexes allows for the calculation of the phase shift for the second blade row. This phase shift can be converted into a time shift as

$$t_{\text{shift}_i} = \pm\frac{\beta_i}{2\pi f_i} \tag{2.28}$$

where the use of the positive or negative sign in Eq. 2.28 is determined based on the tangential direction in which the phase shift is applied.

Due to the time periodic nature of the flow, a truncated temporal Fourier series can be used to efficiently store it, where only N_h harmonics are considered. For every cell belonging to the pitch-wise boundary (two cell layers on each side as illustrated in Fig. 2.3), the n^{th} temporal Fourier coefficient is calculated as ,

$$\begin{aligned} \hat{q}_n &= f_1 \int_{t_s-\frac{1}{f_1}}^{t_s} q(t)e^{-2j\pi n t f_1} dt \\ \hat{q}_n &= f_2 \int_{t_s-\frac{1}{f_2}}^{t_s} q(t)e^{-2j\pi n t f_2} dt \end{aligned} \tag{2.29}$$

where t_s is the current simulation time, j is the imaginary unit, $q(t)$ are the flow variables at time t for a given cell, and depending which blade row the cell belongs to, the first or the second equation in Eq. 2.29 will be used. Due to the discrete nature of CFD methods, Eq. 2.29 can be rewritten as,

$$\begin{aligned} \hat{q}_n &= f_1 \sum_{i=it-N_{tspp1}}^{it} q(i\Delta t)e^{-2j\pi n i \Delta t f_1}\Delta t \\ \hat{q}_n &= f_2 \sum_{i=it-N_{tspp2}}^{it} q(i\Delta t)e^{-2j\pi n i \Delta t f_2}\Delta t \end{aligned} \tag{2.30}$$

where N_{tsppi} is the number of time steps required to complete a period associated to the frequency computed for the i^{th} blade row with Eq. 2.26, Δt is the simulation time step and it is the current iteration of the simulation.

In order to avoid storing the time signal over one full time period before the temporal Fourier coefficients in Eq. 2.30 can be computed, a moving average technique is used to update them [17]. This technique updates the Fourier coefficients by adding the contribution of the current flow solution to all coefficients and subtracting the value one period back in time obtained using the coefficients. By using this updating technique, one can clearly see that when it has fully converged, due to the periodicity of the signal, the reconstructed value one period back in time and the current value should be the same, hence the Fourier coefficients remain unchanged. This moving average based technique can be expressed as,

$$\begin{aligned} \Delta\hat{q}_n &= \frac{1}{N_{tspp1}}\Big(q(t_s) - Q(t_s - \frac{1}{f_1})\Big)e^{-2j\pi n t_s f_1} \\ \Delta\hat{q}_n &= \frac{1}{N_{tspp2}}\Big(q(t_s) - Q(t_s - \frac{1}{f_2})\Big)e^{-2j\pi n t_s f_2} \end{aligned} \tag{2.31}$$

where $\Delta\hat{q}_n$ is the update of the n^{th} temporal Fourier coefficient, and $Q(t)$ is the value obtained when reconstructing the flow properties at time t using the temporal Fourier coefficients computed with Eq. 2.30. Note that the subtraction in Eq. 2.31 is only done when the simulation time exceeds one time period associated with the relative blade passing frequency of the corresponding blade row computed with Eq. 2.26.

Finally, when values are needed in order to transfer information between both sides of the pitch-wise boundaries, the temporal Fourier coefficients are used to reconstructed the desired flow properties with the corresponding time shift computed according to Eq. 2.28 as,

$$Q(t_s + t_{\text{shift}_i}) = \sum_{n=-N_h}^{N_h} \hat{q}_n e^{2j\pi n f_i (t_s + t_{\text{shift}_i})} \tag{2.32}$$

When using the periodic chorochronic boundary condition, the standard set up is the same as when using a standard periodic boundary condition (see Fig. 2.4 a)). In a normal periodic boundary condition this configuration generates no problems, but it might generate convergence issues when using the chorochronic one due to an underlying error feedback loop.

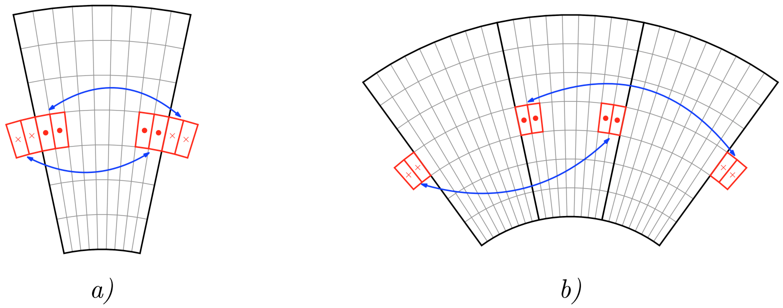


Figure 2.4: *Pitch-wise boundary set up strategy. a) single sector with coupled periodic boundaries, b) multiple sectors with decoupled periodic boundaries.* ● *represents the cells where the temporal Fourier coefficients are sampled and X the ghost cells where the phase shifted values are introduced. The blue arrows show which Fourier coefficients are used for phase shifting which values.*

As the values need to be phase-lagged, one has to wait until the moving average of the temporal Fourier coefficients is somehow converged in order to be able to reconstruct the phase shifted values correctly. In figure 2.4, the cells represented with a full dot are the cells where the temporal Fourier coefficients are computed, and the cells drawn with a cross, the ghost cells where the phase-shifted values are introduced. Before the convergence occurs, an error introduced by the wrong reconstruction of the phase shifted values on the left boundary (crossed cells on the left), is going to be propagated via the flux calculation to the cells where the coefficients are calculated (dot cells on the left) in the standard single sector set up (see Fig. 2.4 a)). This newly introduced error on the Fourier coefficients on the left side is going to be propagated to the right side boundary when the ghost cells values on the right are computed from the coefficients on the left. On can see how this is an ongoing error feedback loop that deteriorates the convergence of the moving average. In order to try to alleviate this, the set up illustrated in Fig. 2.4 b) is used. Here more than one sector is simulated (in this case three, but it could also be done with two, or any number greater than one) and as it can be seen from the figure, the feedback loop is broken, as the cells where the Fourier coefficients are sampled and the ones where the phase shifted values are introduced are not next to each other (this meaning that the error is not directly propagated via the flux calculation). In this case where the periodic boundaries have been decoupled, the algorithm used is exactly the same as described above, with the only difference that β computed with Eq. 2.27 is doubled to account for the extra sector, and therefore the time shift (see Eq. 2.28) as well. A double passage strategy was used in [4, 10] where sampled flow signal improvements was reported and [31] used three sectors in rotor-wake stator interaction computations. In the former two works whether they were simply adding sectors by still having the boundaries coupled or whether they were decoupled to break the feedback loop was not specified.

Blade-row interface

In a turbomachinery blade-row interface, the occurring interacting spinning modes are described by the well known Tyler-Sofrin modes [40],

$$m_{n,k} = nN_1 + kN_2 \tag{2.33}$$

Each of these modes spins at the following frequencies in the frame of reference of the first and second blade row respectively,

$$\begin{aligned} \omega_n &= nN_2(\Omega_1 - \Omega_2) \\ \omega_k &= kN_1(\Omega_2 - \Omega_1) \end{aligned} \tag{2.34}$$

The flow signal at the interface is considered to have a double periodic nature, in time and space, at least for the deterministic part of it. Therefore a truncated double Fourier series is used to represent and store it, where only the first N_h harmonics are considered. A first step required in order to compute the time-azimuthal coefficients for every radial position, $\hat{q}_{n,k}(r)$, is to compute the temporal Fourier coefficients for every cell belonging to the interface in the same manner as described for the pitch-wise boundaries in Sec. 2.3.2. Once the temporal coefficients are computed, the time-azimuthal ones for every radial span can be constructed as,

$$\begin{aligned} \hat{q}_{n,k}(r) &= \frac{N_1}{2\pi} \int_0^{\frac{2\pi}{N_1}} \hat{q}_n(\theta, r) e^{jm_{n,k}\theta} d\theta \\ \hat{q}_{n,k}(r) &= \frac{N_2}{2\pi} \int_0^{\frac{2\pi}{N_2}} \hat{q}_k(\theta, r) e^{jm_{n,k}\theta} d\theta \end{aligned} \tag{2.35}$$

for each of the blade rows respectively. Finally, when the reconstructed values are needed in order to transfer information between blade rows, the flow state can be obtained using the time-azimuthal Fourier coefficients computed using Eq. 2.35 as,

$$\begin{aligned} Q(r, \theta, t) &= \sum_{n=-N_h}^{N_h} \sum_{k=-N_h}^{N_h} \hat{q}_{n,k}(r) e^{j\omega_n t - jm_{n,k}\theta} \\ Q(r, \theta, t) &= \sum_{n=-N_h}^{N_h} \sum_{k=-N_h}^{N_h} \hat{q}_{n,k}(r) e^{j\omega_k t - jm_{n,k}\theta} \end{aligned} \tag{2.36}$$

2.4 Synthetic turbulence

When performing turbulence resolving simulations such as hybrid RANS/LES, LES or DNS, the inflow boundary conditions are not as straight forward as when performing (U)RANS simulations. In the later case, it is enough with specifying the mean profiles of the flow quantities, whereas on the former case, a time dependent turbulent flow field needs to be provided. This is done to avoid having to deal with a large inlet so that the simulation can generate the turbulent flow field by the time the flow reaches the area of interest. In order to tackle this issue, several methods for introducing these fluctuations exist, some of them being; using data from a precursor simulation, Fourier based methods, synthetic eddy methods and so on. The interested reader is referred to a comprehensive review of this methods by [38].

2.4.1 Synthetic Eddy Method

Here, a short overview of the Synthetic Eddy Method (SEM) introduced in [21] is given, for the full details the interested reader is referred to the original work. SEM represents the turbulent fluctuations at the inlet as a contribution from different discrete eddies that are convected with the mean flow.

For simplicity, let the dimensions of the inlet plane be defined as $[0, 0], [0, L_y], [0, L_z]$ in x, y and z direction respectively so that it is a plane orthogonal to the x axis , $\sigma(\mathbf{x})$ the prescribed turbulence length scales, $\mathbf{U}$ the average convection velocity for the eddies and R_{ij} the desired prescribed Reynolds stress tensor. Moreover let σ_i^{max} be the equal to $\max(\sigma_i(\mathbf{x}))$ over the entire inlet plane. First step is to create a box surrounding the inlet plane with dimensions:

$$[-\sigma_x^{max}, \sigma_x^{max}]x[-\sigma_y^{max}, L_y + \sigma_y^{max}]x[-\sigma_z^{max}, L_z + \sigma_z^{max}]$$

This is the box inside which turbulent eddies are allowed to exist and to be convected. In order to ensure that the inlet plane is statistically covered by eddies, one can compute the total number of eddies, N_k, as,

$$N_k = \frac{V_{BOX}}{min(\sigma_x(\mathbf{x})\sigma_y(\mathbf{x})\sigma_z(\mathbf{x}))} \tag{2.37}$$

where V_{BOX} is the volume of the box of eddies just computed.

Once the number of eddies and the size of the box have been determined, the algorithm for computing the velocity fluctuations at every point of the inlet is the following. First, initialize random position for every eddy inside the boundaries of the box, $\mathbf{x}^k$, and assign random intensities to each of them ϵ^k, which take discrete values of -1 or 1. Then repeat the following steps for each physical time step:

1. Convect the eddies through the box using the convection velocity $\mathbf{U}$ as $\mathbf{x}^k = \mathbf{x}^k + \mathbf{U}\Delta t$, where Δt is the physical time step. If any eddy steps out of the box boundaries, it is regenerated randomly at the opposite face of the box with new random intensities ϵ.

2. Compute velocity fluctuations $\mathbf{u}'(\mathbf{x})$ according to Eq. 2.39.

In order to compute the fluctuations $\mathbf{u}'(\mathbf{x})$, first an initial fluctuation, $\mathbf{u}''(\mathbf{x})$, is computed and then scaled according to the prescribed Rij. The contribution of all the eddies needs to be accounted for when computing the initial fluctuation at an inlet cell located at $\mathbf{x}$ as,

$$u_i''(\mathbf{x}) = \frac{1}{\sqrt{N_k}} \sum_{k=1}^{N_k} \epsilon_j^k f_j(\mathbf{x} - \mathbf{x}^k) \tag{2.38}$$

where $\mathbf{x}$ is the position of the inlet point for which the fluctuations are being computed and $f_i(\mathbf{x})$ is a shape function used to represent the fluctuations induced by the turbulent eddies. Once this fluctuation is been computed it is scaled as,

$$u_i'(\mathbf{x}) = a_{ij} u_j''(\mathbf{x}) \tag{2.39}$$

where a_{ij} is the Cholesky decomposition of the prescribed Reynolds stress tensor and reads as follows,

$$a_{ij} = \begin{pmatrix} \sqrt{R_{11}} & 0 & 0 \\ R_{21}/a_{11} & \sqrt{R_{22} - a_{21}^2} & 0 \\ R_{31}/a_{11} & (R_{32} - a_{21}a_{31})/a_{22} & \sqrt{R_{33} - a_{31}^2 - a_{32}^2} \end{pmatrix} \tag{2.40}$$

In this work the used shape function reads,

$$f_i(\mathbf{x} - \mathbf{x}^k) = \sqrt{\frac{V_{BOX}}{\prod_{i=1}^{3} \sigma_i}} \prod_{i=1}^{3} f\Big(\frac{x_i - x_i^k}{\sigma_i}\Big) \tag{2.41}$$

and

$$f(x) = \begin{cases} \sqrt{1.5}(1 - |x|) & \text{if } |x| < 1 \\ 0 & \text{else} \end{cases} \tag{2.42}$$

There might be situations where synthetic fluctuations are imposed in which the simulation also contains periodic boundaries that are in contact with the inlet or RANS/LES interface, such as a turbulent channel flow. For those simulations, the SEM method does not provide a periodic velocity fluctuation field, which might results in reflections emanating from the periodic boundaries which might become important in, for instance, aeroacoustic simulations. In order to address this issue, whenever periodic boundaries are used, the eddies are allowed to induce a perturbation across the boundaries by creating copies of them, as illustrated in Fig. 2.5. This treatment ensures that the fluctuating velocity field will be fully periodic.

2.4.2 Fourier based synthetic fluctuations

An overview of a method to imposed synthetic fluctuations as a superimposition of random Fourier modes is described [7, 12, 5, 6]. In order to prescribe anisotropic fluctuations, this method generates an isotropic turbulent field on the principal components of the desired Reynolds stress tensor and then transforms them to the computational reference system. Here an overview of the method is presented.

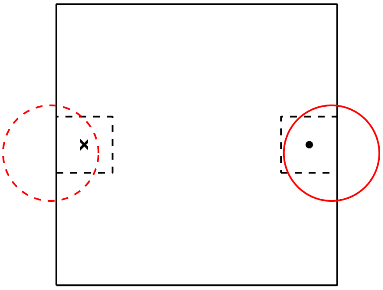


Figure 2.5: *Schematic of the treatment of periodic boundaries when SEM is utilized. The full dot and the cross represent cells that are affected by the presence of the original eddy (solid line circle) and the periodic copy of it (dashed line circle) respectively.*

Isotropic fluctuations

The turbulent fluctuations are expressed as contributions from different random Fourier modes as,

$$\mathbf{u'}_{iso}(\mathbf{x}) = 2\sum_{n=1}^{N} \hat{u}^n \mathrm{cos}(\boldsymbol{\kappa}^n \mathbf{x} + \psi^n)\boldsymbol{\sigma}^n \tag{2.43}$$

where N is the number of Fourier modes used, $\hat{u}^n$, $\boldsymbol{\kappa}^n$, $\boldsymbol{\sigma}^n$ and ψ^n are the amplitude, wave number vector, direction and phase of the n^{th} mode, and $\mathbf{x}$ is the coordinate vector of the grid node where fluctuations are to be introduced. To be able to compute $\boldsymbol{\kappa}^n$, $\boldsymbol{\sigma}^n$ with Eq. 2.44 three random angles are drawn, namely α^n, ϕ^n and θ^n. The random numbers here needed with their probability distributions and allowed ranges are shown in Tab. 2.1.

$$\begin{aligned} \kappa_1^n &= \mathrm{sin}(\theta^n) cos(\phi^n) & \sigma_1^n &= \mathrm{cos}(\phi^n)\mathrm{cos}(\theta^n)\mathrm{cos}(\alpha^n) - \mathrm{sin}(\phi^n)\mathrm{sin}(\alpha^n) \\ \kappa_2^n &= \mathrm{sin}(\theta^n) sin(\phi^n) & \sigma_2^n &= \mathrm{sin}(\phi^n)\mathrm{cos}(\theta^n)\mathrm{cos}(\alpha^n) + \mathrm{cos}(\phi^n)\mathrm{sin}(\alpha^n) \\ \kappa_3^n &= \mathrm{cos}(\theta^n) & \sigma_3^n &= -\mathrm{sin}(\theta^n)\mathrm{cos}(\alpha^n) \end{aligned} \tag{2.44}$$

Table 2.1: Probability distributions and allowed ranges for random variables.

variable	prob.	range
ψ^n	$1/(2\pi)$	$0 < \psi^n < 2\pi$
ϕ^n	$1/(2\pi)$	$0 < \phi^n < 2\pi$
θ^n	$1/(2\mathrm{sin}(\pi))$	$0 < \theta^n < \pi$
α^n	$1/(2\pi)$	$0 < \alpha^n < 2\pi$

For more datailed information on the geometrical and physical meaning of the different random variables, the reader is referred to [12, 11].

In order to compute the amplitude of the Fourier modes, $\hat{u}^n$, the range of wave numbers is first defined. The highest wave number is determined from the grid size and the smallest from the turbulent length scale [11]. Once the range is determined, it is distributed among the N modes on equally large sectors $\Delta\kappa$, and the energy contained by each wave number, $E(\kappa)$, is taken from a modified von Kármán spectrum [12, 11]. The amplitude is finally computed as,

$$\hat{u}^n = \sqrt{E(\kappa)\Delta\kappa} \tag{2.45}$$

Anisotropic fluctuations

The approach presented above has been extended to be able to impose anisotropic fluctuations [36, 5, 7], and used and described in [12, 11]. As previously mentioned the main idea behind this method is to generate isotropic fluctuations on the principal directions of the prescribed Reynolds stress tensor, and then transform them to the computational reference system.

In order to achieve this, Eq. 2.43 is transformed into,

$$\mathbf{u'}_{aniso}(\mathbf{x}^*) = 2\sum_{n=1}^{N} \hat{u}^n cos(\boldsymbol{\kappa}^{*n}\mathbf{x}^* + \psi^n)\boldsymbol{\sigma}^{*n} \tag{2.46}$$

where the superscript * denotes principal coordinates and the wave number vector and direction are scaled according to

$$\boldsymbol{\kappa}^{*n} = \kappa_i^{*n} = \sqrt{\frac{1}{\lambda_i}} R_{ji}\kappa_j^n \quad ; \quad \boldsymbol{\sigma}^{*n} = \sigma_i^{*n} = \sqrt{\lambda_i} R_{ji}\sigma_j^n \tag{2.47}$$

where $\boldsymbol{\lambda}$ are the eigenvalues of the desired Reynolds stress tensor. Finally $\mathbf{u'}_{aniso}(\mathbf{x}^*)$ is transformed to the computational frame of reference, this transformation matrix can be obtained from the eigenvectors of R_{ij}. A complete description of the transformation procedures can be found in [7].

Time correlation

Both the isotropic an anisotropic synthetic generation methods described above use Fourier modes in order to provide the turbulent field with a spatial correlation. In order to ensure that the computed perturbations also have a correlation in time a time filter is applied as follows,

$$(\mathbf{U}')^k = a(\mathbf{U}')^{k-1} + b(\mathbf{u}')^k \tag{2.48}$$

where k denotes the simulation time step, $\mathbf{U}'$ the actual fluctuation that is introduced in the computational domain or RANS/LES interface, $\mathbf{u}'$ is taken from either the isotropic or anisotropic version of the method and a and b are defined as,

$$a = e^{-\Delta t/T_{int}} \qquad b = \sqrt{1-a^2} \tag{2.49}$$

T_{int} being the integral time scale and Δt the simulation time step.

2.5 HAMON - an optimization platform

HAMON is an optimization platform developed in Python whose optimization mechanisms are based on Evolutionary Algorithms (EAs) [18]. It can be used to deal with single- and multi- objective constrained and unconstrained optimization problems. In cases where the objective function is too expensive to evaluate (such as in optimization where the objective function values are obtained from a 3D CFD simulation), it offers the possibility to use Radial Basis Functions (RBFs) to try and speed up the optimization process.

The procedure implemented when using HAMON as an optimization tool, together with CFD simulations and RBFs is illustrated in Fig. 2.6, in this case applied to the multi-objective aerodynamic optimization of a Counter Rotating Open Rotor (CROR).

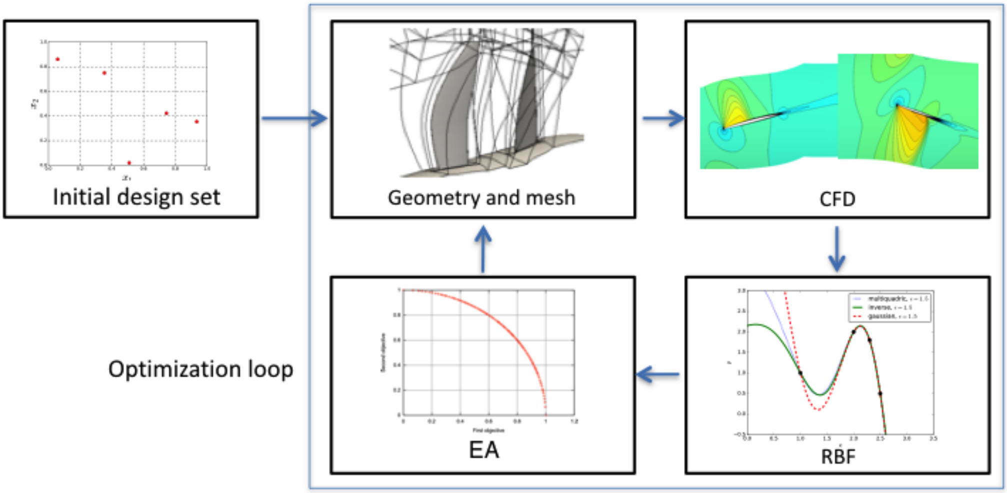


Figure 2.6: *Optimization procedure implemented in HAMON when RBFs are used.*

This entire procedure can be broken down into the following steps, once the design parameters have been defined;

1. **Initial design set:** an initial random sampling of the allowed design space is carried out, in this case by using a Latin Hypercube Sampling (LHS) technique.

2. **Geometry and mesh:** for a given number of designs an automated process generates the corresponding geometries and meshes.

3. **CFD:** the generated designs in the previous steps are simulated using 3D CFD simulations and the objective function values post-process in order to assess the performance of each design.

4. **RBF:** based on all the CFD evaluations performed, a response surface is fitted to allow to estimate the performance of a new design as an algebraic expression (which is of course way faster than the CFD evaluation).

5. **EA:** the generated RBF is passed to the evolutionary algorithm to optimize based on it. A set of optimal designs are found by the optimization process (this is based on RBF prediction not CFD evaluation).

6. **Iterate:** the newly found designs by the EA are put through steps two and three in order to check their "true" performance based on CFD. With the newly obtained information on the designs, the RBF is updated aiming at improving its prediction capabilities (step four). The updated RBF is passed to the EA to optimize (step five). This process is repeated until a satisfactory design is found or the RBF shows no further improvement in its prediction capabilities despite being given information about new designs.

2.5.1 Evolutionary algorithms

HAMON uses evolutionary algorithms as its optimization method, Genetic Algorithms (GAs) and Differential Evolution (DE) being implemented. Both EA methods can handle single- and multi-objective, as well as constrained and unconstrained optimization problems. These two types of evolutionary algorithms are stochastic optimization methods inspired by biological evolution which are population-based algorithms. Both methods start with a pool of candidates or individuals that conform the population (in HAMON it is randomly initialized), which is advanced through generations through selection, mutation and recombination processes [41].

Single-objective genetic algorithm

In HAMON a standard genetic algorithm with binary encoding (meaning that the design variables are encoded in an array of ones and zeros called chromosome) is implemented for single-objective problems. The processes used for advancing the population from one generation to the next are the following ones;

1. **Selection** (tournament selection of size two): two individuals are randomly chosen from the population and compared against each other. The one with better performance will move on with a user defined probability (typically between 0.7 and 1). This selection process resembles the "survival of the strongest" seeing in nature.

2. **Crossover** (one-point crossover): after two individuals have come out of tournament selection, they are given the chance (with a user defined probability) of generating an offspring by combining their chromosomes (see Fig. 2.7). If the offspring is successful the two children move on, otherwise the parents do. This process mirrors the reproduction seen in nature.

3. **Mutation**: the individuals coming out of the crossover process (either the parents or the children) get each of their chromosomes swapped with a user-defined probability.

4. **Elitism**: before putting a population through the three processes aforementioned, a copy of the best performing individual is stored to ensure that it is not lost. After the population has gone through selection, crossover and mutation, the saved copy is swapped by a random individual of the population provided it performed worse.

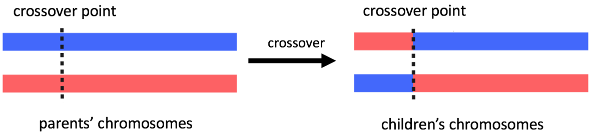


Figure 2.7: *Illustration of one-point crossover.*

Once these four processes are performed on a population, the population is considered to have advanced one generation. This is performed iteratively until a stopping criteria is being satisfied. This criteria is usually a maximum number of generations, or a relative value of the objective function.

Multi-objective genetic algorithm

The main difference when dealing with multi-objective problems as opposed to single-objective ones, is that comparing two individuals is not always straightforward. The situations where one of the individuals performs better on one objective function but worse on another one does not have a clear solution (this direct comparison is for instance used in the selection or elitism processes). Due to these situations where a clear best individual can not be chosen, the solution to a multi-objective problem is no longer an individual, but instead a group of them, called pareto-front. In order to deal with multi-objective problems, the NSGA-II algorithm proposed by Deb et al. [13] has been chosen to be implemented in HAMON.

In order to address the issue of comparing two individuals, the dominance operator ($\prec$) was introduced in the NSGA-II algorithm. Let N be the number of objective functions, i_1 and i_2 the two individuals to be compared and $o_m(j)$ the value of the mth objective function of the j^{th} individual. Assuming all the objective functions are to be minimized (if this is not the case the corresponding greater than or lower than sign is used for each objective function) then individual i_1 dominates individual i_2 if:

$$i1 \ \prec \ i2 \ \leftrightarrow o_m(i_1) \leq o_m(i_2), \ \forall m \in \{1, ..., N\} \ \wedge \ \exists k \in \{1, ..., N\} \ : \ o_k(i_1) < o_k(i_2) \tag{2.50}$$

The dominance operator is used to classify the individuals of a population into ranks, being rank one the best. Individuals that are not dominated by any of the individuals of the population belong to rank one, individuals which are dominated by only rank one individuals are rank two and so on.

The crowded distance parameter (*cd*) is another concept introduced in NSGA-II. It aims to quantify the density of the surroundings of an individual in the objective functions space as follows: Let $\mathcal{A}_m$ be the set containing individuals with a common rank sorted according to the value of their objective function m, M the size of $\mathcal{A}_m$ and o_m^{max} and o_m^{min} the maximum and minimum values of the objective function m in that set respectively. The crowded distance for each of the individuals i , $cd(i)$, belonging to the set is calculated as follows,

```
for each i, set $cd(i) = 0$
for $m = 1$ to $N$
    $cd\big(\mathcal{A}_m(1)\big) = cd\big(\mathcal{A}_m(M)\big) = \infty$
    for $i = 2$ to $(M - 1)$
        $cd\big(\mathcal{A}_m(i)\big) = cd\big(\mathcal{A}_m(i)\big) + \Big[o_m\big(\mathcal{A}_m(i+1)\big) - o_m\big(\mathcal{A}_m(i-1)\big)\Big] / \big(o_m^{max} - o_m^{min}\big)$    (2.51)
```

Taking the concepts of crowded distance and rank into account, when comparing two individuals the one with lower rank is considered the better performing one. In the case where they belong to the same rank, the one with the largest crowded distance is chosen, thus aiming at exploring in the objective function space.

The NSGA-II implemented in HAMON still uses tournament selection (of size two), crossover and mutation processes. These are done in the same manner as for the single-objective case (except from the difference on how individuals are compared in tournament selection). The elitism process is the one that differs the most. NSGA-II applies an elitism process that involves the newly generated population through mutation, crossover and selection (here called Q), and the previous population (here called P). Both populations are combined into a new one with twice the size, PQ, and their individuals are ranked and the crowded distance computed. The population P of the new generation (which is of the same size as the original one) is obtained by the process illustrated in Fig. 2.8. First, rank one individuals of population PQ are introduced in the new population P. Later on, the individuals of rank two are added and so on. This process proceeds until adding all the individuals of a certain rank will max out the size of the new population P of the next generation. Then, only the individuals with larger crowded distance are added until the new population is of the right size, and the rest of the individuals are discarded.

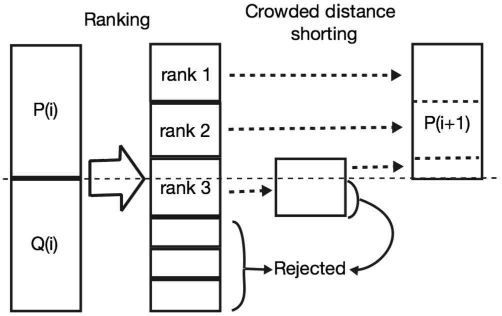


Figure 2.8: *Elitism process in the NSGA-II algorithm. Here i represents the generation number of the population.*

Differential evolution

As well as genetic algorithms, differential evolution is a stochastic optimization method that falls under the evolutionary algorithms category [37]. In HAMON's implementation of the DE, the general algorithm for both single- and multi-objective problems is the same as that of the GA; but mutation, recombination (analogous to crossover) and selection are performed differently. The main processes in DE are explained here with a notation similar to the one used by Rai [33]. As previously mentioned, each individual's design variables are encoded in their chromosome when using GA. On the other hand, on DE each individual can be considered a parameter vector containing its design variables:

$$\mathbf{X_{i,n}} = \left[x_{1,i,n}, x_{2,i,n}, ..., x_{D,i,n}\right]$$

where $\mathbf{X_{i,n}}$ is the parameter vector of the i^{th} individual of nth generation's population, D is the number of variables and $x_{j,i,n}$ is the j^{th} design variable of that individual.

The following three processes are used to evolve $\mathbf{X_{i,n}}$ to the next generation's $\mathbf{X_{i,n+1}}$:

1. **Mutation**: three individuals are randomly selected from the population, $\mathbf{X_{a,n}}$, $\mathbf{X_{b,n}}$ and $\mathbf{X_{c,n}}$, with $a \neq b \neq c \neq i$. Then a new individual $\mathbf{W}$ is created as:

$$\mathbf{W} = \left[w_1, w_2, ..., w_D\right] = \mathbf{X_{a,n}} + F(\mathbf{X_{b,n}} - \mathbf{X_{c,n}}) \tag{2.52}$$

   where $0 < F < 1$ is a user define parameter. If any of the variables of $\mathbf{W}$ was to fall outside the allowed design range, they are properly adjusted.

2. **Recombination**: the newly created individual ($\mathbf{W}$) is given the chance to combine itself with the original one ($\mathbf{X_{i,n}}$) to generate a new individual ($\mathbf{Z}$) as:

$$\mathbf{Z} = \left[z_1, z_2, ..., z_D\right], \text{ where } z_j = \begin{cases} w_j & \text{if } \ r_j \leq C \\ x_{j,i,n} & \text{if } \ r_j > C \end{cases} \tag{2.53}$$

   where r_j is uniformly distributed random number, and $0 < C < 1$ a user defined parameter.

3. **Selection**: the last step is to make $\mathbf{X_{i,n+1}}$ a copy of the best performing individual when comparing $\mathbf{Z}$ and $\mathbf{X_{i,n}}$. Selection process is only done in single-objective optimization, as it can be interpreted as an elitism process as well, since it ensures that one generation is never worse than the previous one. There is no need to do this in multi-objective optimization, as the same process as in the NSGA-II algorithm is used (see Fig. 2.8) and it already compares the old and the new generation in order to form the best performing possible one.

Chapter 3
Vaidations and results

In this section some validation cases used to verify the correct implementation of the different methods described in chapter 2 are presented.

3.1 LODI based outlet

Here the behaviour of the three different types of pressure outlets described in Sec 2.2.2 are tested, namely; perfectly-reflecting, perfectly non-reflecting and partially-reflecting outlet. In order to test the response, a half sine wave velocity pulse is excited at the inlet and convected through the domain. The results at three different time instances can be seen in Fig. 3.1 in the form of pressure fluctuations along the domain.

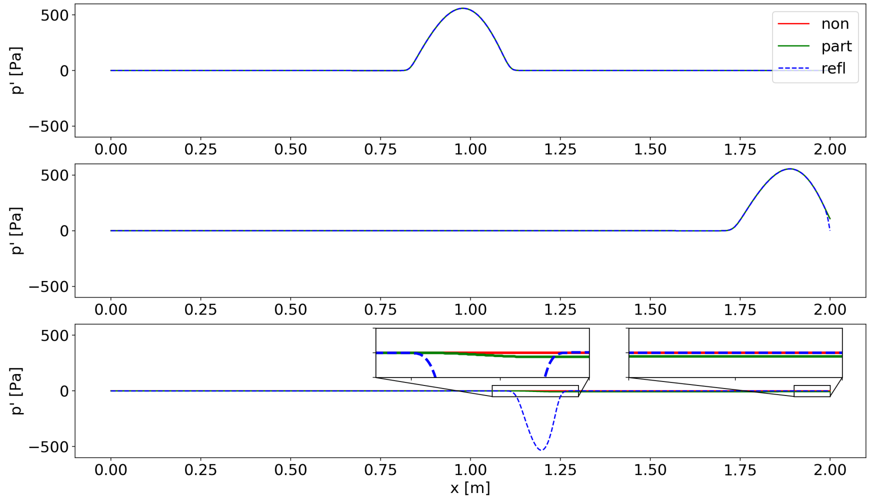


Figure 3.1: *Pressure fluctuation comparison between the three LODI based outlets; perfectly non-reflective, partially-reflective and perfectly-reflective labeled "non", "part" and "refl" respectively in the figure. Three different time instances; top: wave travelling from left to right in the middle of the domain, middle: wave reaching the outlet and bottom: original wave has exited the domain, reflections might have bounced back.*

At the time of the top figure, the wave is still travelling inside the computational domain and since the pressure used to initialize the solution is the same as the one specified for the outlet boundary, the three solutions are identical. In the middle figure, the wave is reaching the location of the outlet and a minor difference, almost negligible, can be observed. In the bottom one, the original wave has already left the computational domain, therefore whatever pressure fluctuation is remaining inside the domain is non physical and has been caused by the outlet boundary condition. It can clearly be seen in the middle of the domain how the perfectly reflecting formulation has left an almost undamped wave with an amplitude reaching almost 500 Pa, same as the original wave. By taking a closer look at the area near the outlet where the zoom in is shown, one can see that the partially-reflecting formulation struggles a little bit to keep the exact pressure level at the outlet. Even though this difference is not of great importance (6 Pa deviation out of the around 500 Pa of the incoming wave), it can also be appreciated in the second zoomed in figure closer to the center of the domain. The best performing formulation in this case, as expected, is the perfectly non-reflective one, as it is able to maintain the right pressure level at the outlet with zero reflections happening at the outlet. As previously mentioned in Sec. 2.2.2, the problem with this formulation is that since no information from the outside is allowed to travel inside the domain, setting the right pressure level is not possible (except in a case like this one, where the pressure level was right from the initialization), therefore the partially-reflective one is been set as default in G3D::Flow.

3.2 Sliding grid

To test the implementation of the GGI and the sliding grid a URANS simulation of a slightly modified version of the VINK compressor [22, 19] is performed. The blade count of the compressor has been modified to have 50 rotor and 90 stator blades instead of the 51 and 88 of the original design. This change allows to perform a simulation with sliding grid and standard periodic boundary conditions, where 5 rotors and 9 stators are considered, as both cover a 36° sector. Furthermore, in order to reduce the computational resources, only a spanwise sector that covers around 15% of it has been used (same as used in Paper D). To validate the correct implementation the flow properties right before and right after the interface are sampled along a constant radius line and compared. Results for the axial component of the velocity can be seen in Fig. 3.2.

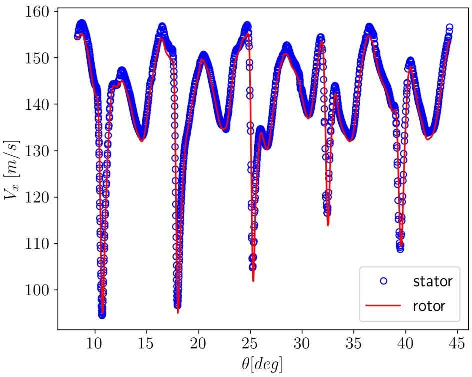
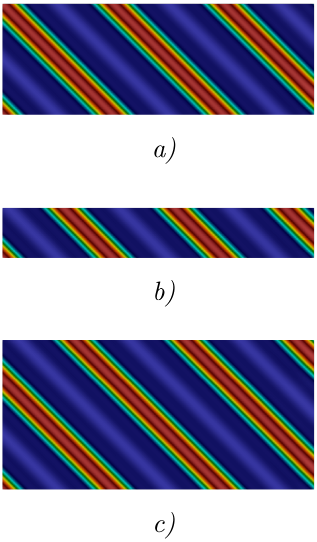


Figure 3.2: *Axial velocity comparison before and after the sliding grid interface.*

It can clearly be seen from the figure, how the five wakes emanating from the five rotors appear and how the agreement between the rotor and stator domain is good. Axial velocity is only shown here for brevity, but it has been checked that the other two velocity components (accounting for change in frame of reference) as well as thermodynamic properties, such as temperature, density and pressure have the same level of agreement.

3.3 Chorochronic periodicity and boundary decoupling

The correct implementation of the chorochronic pitch-wise boundaries, as well as the assumption that the decoupling of the boundaries (achieved by introducing extra sectors) improves the convergence of the temporal Fourier coefficients' moving average is verified here. To do so, a test case where a vortical gust is introduced is used, and three simulations are performed; standard periodic (this is used as benchmark), standard chorochronic set up and decoupled chorchronic set up (see Fig. 2.4). The axial contours of the three simulations for the test case can be seen in Fig. 3.3, where the phase shift applied in the pitch-wise boundaries can clearly be seen.



a)

b)

c)

Figure 3.3: *Pitch-wise chorochronic and decoupling assessment. Axial velocity contours; a) standard periodic, b) standard chorochronic and c) decoupled chorochronic. Same colorbar in all three contour plots.*

To perform a more quantitative comparison the flow properties are extracted over a line that goes through the middle of the domain and the results are shown in Fig. 3.4 where an excellent agreement can be seen.

Now that the correct behaviour is been validated for both the standard pitch-wise

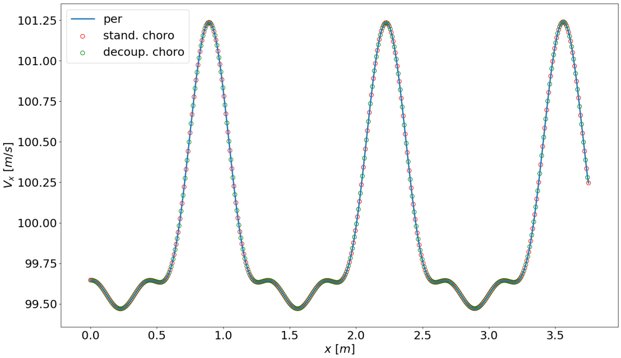


Figure 3.4: *Axial velocity comparison. For visibility enhancement not all the points have been plotted for standard and decoupled chorochronic, and they have been alternated.*

and decoupled chorochronic, what is left to check is the convergence rate of the moving average for these two set ups. In order to measure the convergence rate, ϵ, as defined in Eq. 3.1 is used,

$$\epsilon = \frac{q(t) - Q(t-T)}{q(t)} \tag{3.1}$$

where $q(t)$ is the flow variable at time t, T is the time period of the signal and $Q(t-T)$ is the reconstructed variable using the temporal Fourier coefficients one period back in time. The comparison of the convergence rate is shown in Fig. 3.5 where it can be clearly seen that the decoupled set up converges much faster.

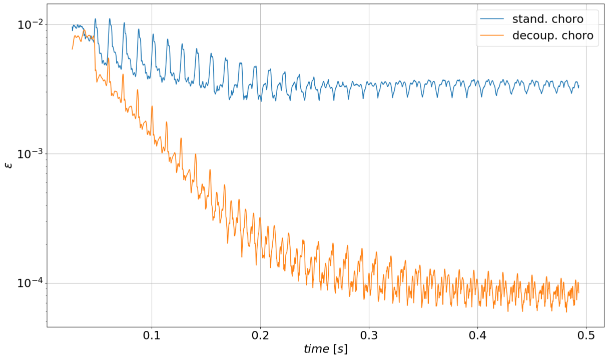


Figure 3.5: *Comparison of the convergence of the moving average between both chorochronic pitch-wise set up strategies.*

A more thorough validation of the implementation of the entire chorochronic method (including the rotor-stator interface) is presented in Paper D.

3.4 Synthetic turbulence

Here a validation test for the two methods for generating synthetic fluctuations at the inlet is presented, namely; the synthetic eddy method and the method based on Fourier coefficients. In this case the homogeneous decaying isotropic turbulence test case has been chosen, where the normal components of the Reynolds stress tensor are equal to one. A domain of sizes 6π by 2π by 2π in the x, y and z directions respectively is used which is discretized using 384 cells in the stream-wise direction and 128 in the y and z directions. A mean velocity of 20 m/s, integral length scale of 0.5 m and the Smagorinsky model is used as a subgrid scale model. Periodic boundary conditions are used on the y and z directions and the flow statistics are averaged over three through flows.

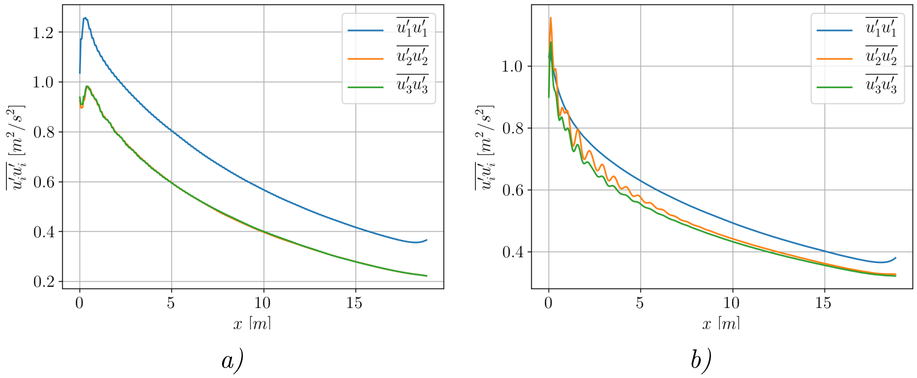


Figure 3.6: *Normal Reynolds stress components along the length of the domain. a) SEM and b) Fourier based synthetic turbulence.*

Figure 3.6 shows the results obtained with both methods for the normal stresses (the off-diagonal stresses were below 3% for both cases and are not shown here for brevity). Taking a look at the results obtained with SEM (Fig. 3.6 a)) one can observe that the values start relatively close to the desired value but there is a spike close to the inlet. The reason of why is this happening is still unclear to the author but similar things have been reported by other authors. In the original work by Jarrin et al. [21], the individual stresses are not reported, but a similar kink can be seen in the turbulent kinetic energy. In the work by Matha et al. [26], this kink appears again even though the synthetic turbulence generation method is different. A very similar behaviour is seen in the results obtained with the Fourier based method, but in this case some oscillations of the $\overline{u'_2 u'_2}$ and $\overline{u'_3 u'_3}$ stresses is seen along the domain. The reason behind this is not understood yet, but one possible explanation is that this method does not generate a periodic inlet perturbation field.

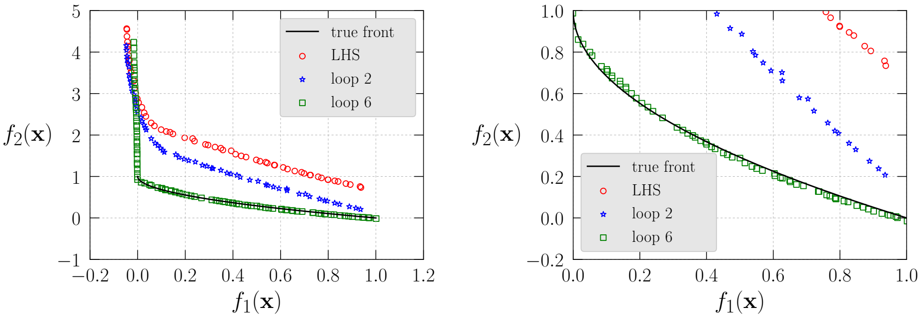
3.5 HAMON

In this section a validation case where HAMON is used together with RBFs is presented (see Fig. 2.6). The chosen test case is a well known function that is often used as a benchmark for multi-objective unconstrained problems as it has an analytical solution. The function is often referred to as the ZDT 1 function [43], which definition is given in Table 3.1.

Table 3.1: Definition of ZDT 1.

Function	variables, m	Variable range	Functions to minimize ($f_1(\mathbf{x})$ & $f_2(\mathbf{x})$)
ZDT 1	30	[0, 1]	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})\left[1 - \sqrt{x_1/g(\mathbf{x})}\right]$ $g(\mathbf{x}) = 1 + 9\left(\sum_{i=2}^{m} x_i\right)/(m-1)$

In this case the optimization has been performed using the multi-objective DE in HAMON, but very similar results were also observed using the NSGA-II algorithm. The results for the optimization can be seen in Fig. 3.7 where the evolution of the predictive capabilities of the RBF are shown. It can clearly be seen that when the RBF is fed only the designs from the latin hypercube sampling, the pareto-front obtained using the RBF is very far from thee true one. The more information is given to the RBF, the closer it gets to the true front, being really close by loop number 6.



a) Entire pareto-front *b) Close up view*

Figure 3.7: *Evolution of the DE pareto-front over several optimization loops.*

A more extensive validation of the different possibilities that HAMON offers can be found in B.

3.6 TurboNoiseBB

Within the European project TurboNoiseBB the aim is the broadband noise prediction on the ACAT1 fan, more specifically the fan-Outlet Guide Vane (OGV) interaction. Due to the number of fan and OGV blades, it becomes extremely expensive (essentially prohibitive) to perform a (D)DES/LES simulation where a sliding mesh and standard periodic boundary condition can be used without altering the original blade count. Moreover, Chalmers' role in the project is to perform DDES simulations using phase-lagged boundary conditions keeping the original blade count. In one of the simulations the wake and its turbulent properties are synthesized from a precursor RANS simulation, and on the second one, both the fan and the OGVs are taken into account. The chorochronic method described in Sec. 2.3.2 is utilized.

3.6.1 Synthesized wake simulation

In this simulation only the OGV inside the bypass is considered. A schematic of the set up used for this simulation can be seen in Fig. 3.8.

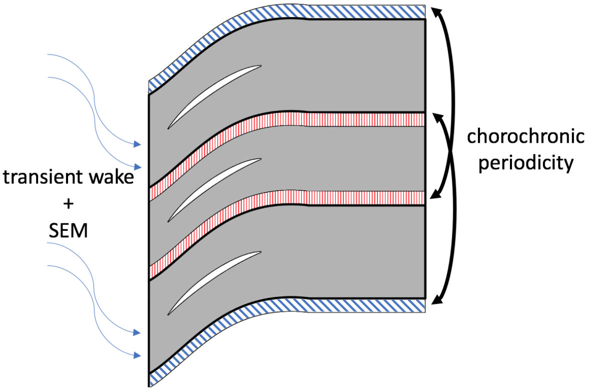


Figure 3.8: *Schematic set up of the simulation where only the OGVs are simulated. The areas colored with red lines represent the cells where the temporal Fourier coefficients are computed and the ones colored by blue lines the ghost cells where the phase shifted values are introduced. The arrows on the right show which coefficients are connected with which ghost cells.*

As can be seen on the figure, decoupled chorochronic pitch-wise boundaries are used where the error feed back loop is broken (as described in Sec. 2.3.2). This decoupling of the boundaries is considered to become even more important now that a DDES simulation is performed, as the signal is going to be more distorted than on URANS simulations. In order to emulate the effect of the rotor wakes impinging on the OGV blades, the mean flow properties of the wake are extracted from a steady state RANS simulation and they are superimposed with synthetic turbulence fluctuations generated using the SEM method described in Sec. 2.4.1. This simulation is currently still running and therefore acoustic predictions are not available yet. Figure 3.9 shows Q-criterion isosurfaces colored by vorticity magnitude where one can clearly see the three OGV blades and the impinging fan wakes.

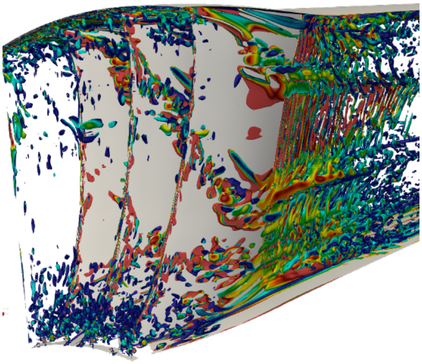


Figure 3.9: *Q criterion isosurfaces colored by vorticity magnitude in the ACAT 1's OGV blades when the fan wakes are synthesized and introduced at the inlet.*

3.6.2 Fan-OGV simulation

In this case, the fan domain is also considered in the simulation, therefore the chorochronic rotor-stator interface is also used. Figure 3.10 shows an schematic of the simulation set up.

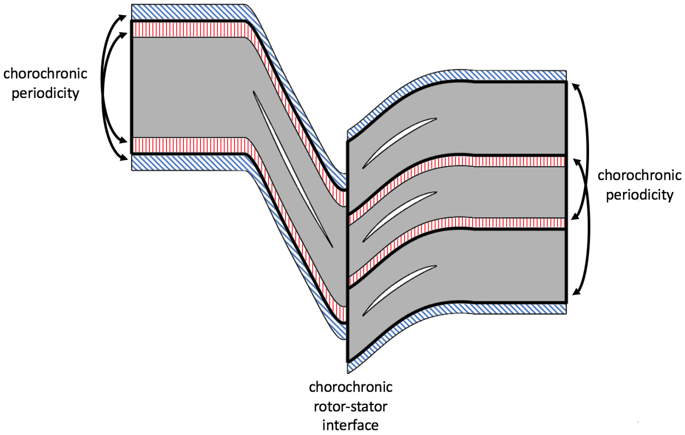


Figure 3.10: *Schematic set up of the simulation where both fan and OGV domains are simulated. The areas colored with red lines represent the cells where the temporal Fourier coefficients are computed and the ones colored by blue lines the ghost cells where the phase shifted values are introduced. The arrows on the right show which coefficients are connected with which ghost cells.*

Similarly as for the case where only the OGVs are simulated, the periodic decoupling is sought for, but it is not considered to be as important on the fan domain as it is on the OGV domain. Therefore only one fan blade is discretized. Due to the smaller cells found on the tip gap of the fan, performing the simulation with the explicit solver became prohibitive, therefore the dual time stepping method implemented in G3D::Flow is being currently used [35]. Figure 3.11 shows Q-criterion isosurfaces on the suction side of the fan blade, where the separation that has appeared also in steady state RANS and simulations by other project partners can be observed. The simulation is still not finished and is currently running, therefore acoustic predictions are not presented here.

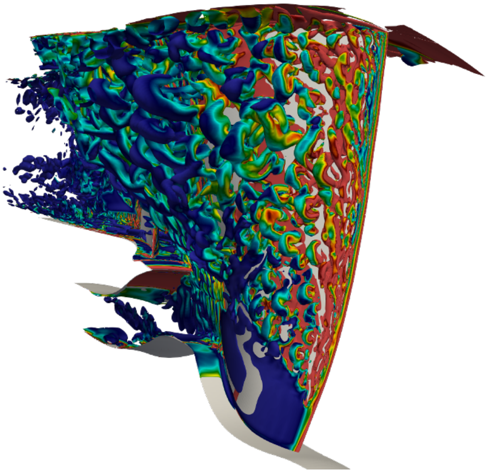


Figure 3.11: *Q criterion isosurfaces colored by vorticity magnitude on the suction side of the ACAT 1's fan blade.*

Chapter 4
Division of work

4.1 Paper A

D. Lindblad, G. Montero Villar, N. Andersson, A. Capitao Patrao, S. Courty-Audren, and G. Napias. “Aeroacoustic Analysis of a Counter Rotating Open Rotor Based on the Harmonic Balance Method”. *2018 AIAA Aerospace Sciences Meeting.* 2018

The geometry was designed by Alexandre and it was later meshed by Gael and Suk-kee. The implementation of the rotor-stator interface as well as the pitch-wise boundary conditions used into the G3D::Flow solver was done by both Daniel and Gonzalo with support from Niklas. The simulation, post-processing and writing of the article was done by Daniel.

4.2 Paper B

G. Montero Villar, D. Lindblad, and N. Andersson. “Multi-Objective Optimization of an Counter Rotating Open Rotor using Evolutionary Algorithms”. *2018 Multidisciplinary Analysis and Optimization Conference.* 2018

The decision of which design variables to use as well as the ranges allowed for them was decided by Gonzalo and Daniel. The set up of the CFD simulation in STAR-CCM+ was done by Daniel. The development of the optimization framework, the automation of geometry generation and meshing, performing the optimization, post-processing and writing of the article was done by Gonzalo. Niklas provided support along the way.

4.3 Paper C

G. Montero Villar, D. Lindblad, and N. Andersson. “Effect of Airfoil Parametrization on the Optimization of Counter Rotating Open Rotors”. *AIAA Scitech 2019 Forum.* 2019

The design variables used as well as the ranges allowed for them was decided by Gonzalo and Daniel. The set up of the CFD simulation in STAR-CCM+ was done by Daniel. The development of the optimization framework, the automation of geometry generation (when using NACA profiles) and meshing, performing both optimizations, post-processing and writing of the article was done by Gonzalo. Daniel extended the existing geometry generation code to allow for CST profiles. Niklas provided support along the way.

4.4 Paper D

G. Montero Villar, D. Lindblad, and N. Andersson. “Investigation of Phase-Lagged Boundary Conditions for Turbulence Resolving Turbomachinery Simulations”. *AIAA AVIATION 2020 FORUM.* 2020

The mesh was generated by Gonzalo. The rotor-stator interface as well as the pitch-wise boundary conditions used were implemented into G3D::Flow by Gonzalo and Daniel with support from Niklas. The simulation set up, running and post-processing as well as writing the article was done by Gonzalo.

References

[1] Airbus. *Global Market Forecast 2016-2035*. 2016.

[2] N. Andersson. *A Study of Subsonic Turbulent Jets and Their Radiated Sound Using Large-Eddy Simulation*. PhD thesis, Chalmers University of Technology, 2005.

[3] R. Avellan and A. Lundbladh. *Air propeller arrangement and aircraft*. US Patent App. 13/519,588. 2009.

[4] T. Biesinger, C. Cornelius, C. Rube, A. Braune, R. Campregher, P. G. Godin, G. Schmid, and L. Zori. "Unsteady CFD methods in a commercial solver for turbomachinery applications". *ASME Turbo Expo 2010: Power for Land, Sea, and Air*. American Society of Mechanical Engineers Digital Collection. 2010, pp. 2441–2452.

[5] M. Billson. *Computational techniques for turbulence generated noise*. PhD thesis, Chalmers University of Technology, 2004.

[6] M. Billson, L.-E. Eriksson, and L. Davidson. "Jet noise prediction using stochastic turbulence modeling". *9th AIAA/CEAS aeroacoustics conference and exhibit*. 2003, p. 3282.

[7] M. Billson, L.-E. Eriksson, L. Davidson, and P. Jordan. "Modeling of synthetic anisotropic turbulence and its sound emission". *10th AIAA/CEAS aeroacoustics conference*. 2004, p. 2857.

[8] A. Capitao Patrao. *On the Aerodynamic Design of the Boxprop*. PhD thesis, Chalmers University of Technology, 218.

[9] A. Capitao Patrao, T. Grönstedt, A. Lundbladh, and G. Montero Villar. Wake Analysis of an Aerodynamically Optimized Boxprop High-Speed Propeller. *Journal of Turbomachinery* (2019).

[10] S. Connell, M. Braaten, L. Zori, R. Steed, B. Hutchinson, and G. Cox. "A Comparison of Advanced Numerical Techniques to Model Transient Flow in Turbomachinery Blade Rows". *ASME 2011 Turbo Expo: Turbine Technical Conference and Exposition*. American Society of Mechanical Engineers. 2011, pp. 1241–1250.

[11] L. Davidson. *Fluid mechanics, turbulent flow and turbulence modeling*. 2015.

[12] L. Davidson and M. Billson. Hybrid LES-RANS using synthesized turbulent fluctuations for forcing in the interface region. *International journal of heat and fluid flow* **27**.6 (2006), 1028–1042.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on* **6**.2 (2002), 182–197.

[14] L. E. Eriksson. Development and validation of highly modular flow solver versions in g2dflow and g3dflow. *Volvo Aero Corporation, Trollhättan, Sweden, Technical Report* 9970-1162 (1995).

[15] *Flightpath 2050, Europe's Vision for Aviation*. https://ec.europa.eu/transport/sites/transport/files/modes/air/doc/flightpath2050.pdf. Accessed: August 2020.

[16] G3D::Flow homepage (accessed 2020). `https://nikander.github.io/g3dflow/`.

[17] G. A. Gerolymos, G. J. Michon, and J. Neubauer. Analysis and Application of Chorochronic Periodicity in Turbomachinery Rotor/Stator Interaction Computations. *Journal of propulsion and power* **18**.6 (2002), 1139–1152.

[18] GitHub repository for HAMON (2020). `https://github.com/gmonterovillar/HAMON`.

[19] GitHub repository for VINK (2017). `https://github.com/nikander/VINK`.

[20] *IATA Annual Review 2019.* https://www.iata.org/contentassets/c81222d96c9a4e0bb4ff6ced0126f0bb/iata-annual-review-2019.pdf. Accessed: August 2020.

[21] N. Jarrin, S. Benhamadouche, D. Laurence, and R. Prosser. A synthetic-eddy-method for generating inflow conditions for large-eddy simulations. *International Journal of Heat and Fluid Flow* **27**.4 (2006), 585–593.

[22] M. Lejon, T. Grönstedt, N. Glodic, P. Petrie-Repar, M. Genrup, and A. Mann. "Multidisciplinary design of a three stage high speed booster". *ASME Turbo Expo 2017: Turbomachinery Technical Conference and Exposition.* American Society of Mechanical Engineers Digital Collection. 2017.

[23] D. Lindblad, G. Montero Villar, N. Andersson, A. Capitao Patrao, S. Courty-Audren, and G. Napias. "Aeroacoustic Analysis of a Counter Rotating Open Rotor Based on the Harmonic Balance Method". *2018 AIAA Aerospace Sciences Meeting.* 2018.

[24] D. Lindblad, N. Wukie, G. Montero Villar, and N. Andersson. "A Nonreflecting Formulation for Turbomachinery Boundaries and Blade Row Interfaces". *AIAA Scitech 2019 Forum.* 2019.

[25] D. Lindblad, N. A. Wukie, G. Montero Villar, and N.s Andersson. "Implementation of a Quasi-Three-Dimensional Nonreflecting Blade Row Interface for Steady and Unsteady Analysis of Axial Turbomachines". *2018 AIAA/CEAS Aeroacoustics Conference.* 2018.

[26] M. Matha, C. Morsbach, and M. Bergmann. "A comparison of methods for introducing synthetic turbulence". *ECCOMAS–ECFD 2018-6th European Conference on Computational Mechanics (Solids, Structures and Coupled Problems)/7th European Conference on Computational Fluid Dynamics.* 2018.

[27] G. Montero Villar, D. Lindblad, and N. Andersson. "Effect of Airfoil Parametrization on the Optimization of Counter Rotating Open Rotors". *AIAA Scitech 2019 Forum.* 2019.

[28] G. Montero Villar, D. Lindblad, and N. Andersson. "Investigation of Phase-Lagged Boundary Conditions for Turbulence Resolving Turbomachinery Simulations". *AIAA AVIATION 2020 FORUM.* 2020.

[29] G. Montero Villar, D. Lindblad, and N. Andersson. "Multi-Objective Optimization of an Counter Rotating Open Rotor using Evolutionary Algorithms". *2018 Multidisciplinary Analysis and Optimization Conference.* 2018.

[30] M. Olausson. *Turbomachinery aeroacoustic calculations using nonlinear methods.* PhD thesis, Chalmers University of Technology, 2011.

[31] M. Olausson and L.-E. Eriksson. "Rotor wake/stator broadband noise calculations using hybrid RANS/LES and chorochronic buffer zones". *15th AIAA/CEAS Aeroacoustics Conference (30th AIAA Aeroacoustics Conference).* 2009, p. 3338.

[32] T. J. Poinsot and S. K. Lele. Boundary Conditions for Direct Simulations of Compressible Viscous Flows. *Journal of Computational Physics* **101**.1 (1992), 104–129.

[33] M. M. Rai. Single-and Multiple-Objective Optimization with Differential Evolution and Neural Networks. *VKI lecture series: introduction to optimization and multidisciplinary design* **58** (2006).

[34] D. H. Rudy and J. C. Strikwerda. Boundary conditions for subsonic compressible Navier-Stokes calculations. *Computers & Fluids* **9**.3 (1981), 327–338.

[35] E. M. V. Siggeirsson. *Aerodynamics of an Aeroengine Intermediate Compressor Duct: Effects from an Integrated Bleed System.* PhD thesis, Chalmers University of Technology, 2020.

[36] A. Smirnov, S. Shi, and I. Celik. Random flow generation technique for large eddy simulations and particle-dynamics modeling. *J. Fluids Eng.* **123**.2 (2001), 359–371.

[37] R. Storn and K. Price. Differential Evolution–a Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces. *Journal of Global Optimization* **11**.4 (1997), 341–359.

[38] G. R Tabor and MH Baba-Ahmadi. Inlet conditions for large eddy simulation: A review. *Computers & Fluids* **39**.4 (2010), 553–567.

[39] K. W. Thompson. Time dependent Boundary Conditions for Hyperbolic systems. *Journal of Computational Physics* **68**.1 (1987), 1–24.

[40] J. M. Tyler and T. G. Sofrin. *Axial flow compressor noise studies.* Tech. rep. SAE Technical Paper, 1962.

[41] M. Wahde. “Biologically Inspired Optimization Methods: an Introduction”. WIT Press, 2008. Chap. 3, pp. 35–67.

[42] A. Widenhorn, B. Noll, and M. Aigner. “Accurate boundary conditions for the numerical simulation of thermoacoustic phenomena in gas-turbine combustion chambers”. *ASME Turbo Expo 2006: Power for Land, Sea, and Air.* American Society of Mechanical Engineers Digital Collection. 2006, pp. 347–356.

[43] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation* **8**.2 (2000), 173–195.