Improved algorithms for the computation of induced velocities in propeller design

by

Frédéric BUCHOUX

Submitted to the Department of Ocean Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Ocean Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1995

© Massachusetts Institute of Technology 1995. All rights reserved.

Author ..................................................

Department of Ocean Engineering

June 7, 1995

Certified by .............................................

Justin E. Kerwin
Professor
Thesis Supervisor

Accepted by .............................................

Douglas A. Carmichael
Chairman, Departmental Committee on Graduate Students
Improved algorithms for the computation of induced velocities in propeller design

by

Frédéric BUCHOUX

Submitted to the Department of Ocean Engineering on June 7, 1995, in partial fulfillment of the requirements for the degree of Master of Science in Ocean Engineering

Abstract

The successful Propeller Design Program in use at the Department of Ocean Engineering at the Massachusetts Institute of Technology requires the computation of velocities and circumferential mean velocities induced by the propeller at field points defined by a vortex lattice method. An improved theory to compute the velocities induced by each vortex/source segment defined by the blade lattice, including the effects of loading and thickness, is given as an alternative method to the accurate, but slow, algorithm previously in use. Based on an axisymmetric decomposition for each segment contribution and on vortex and source ring influence functions computation, it reduces the computation running time by a factor of 7 without losing the necessary accuracy.

Thesis Supervisor: Justin E. Kerwin
Title: Professor of Naval Architecture
Acknowledgments

I would like to thank Professor Kerwin who provided the guidance and encouragement needed to complete this challenging project. I also thank my advisor, professor Tomasz Wierzbicki, who was always friendly and available, and everyone in the Ocean engineering department, and particularly among the propeller nuts, who helped me during my work.

I am very grateful to Lamson Corporation and Jacques Lepage who made my stay at M.I.T possible.

I wish to thank everyone who gave me some of their time to send me support through letters, calls, fax or just thoughts. A special thank to my true friends in France and in the US who mean a lot to me and are really worth sharing my life with. I hope the link between us will remain as strong in the future.

Finally, I would like to particularly thank my parents and my sister for their love, advice, confidence and unlimited support during my whole life. I would never have been able to do anything without them.
# Contents

1 Introduction ........................................... 11

2 Background ............................................. 15
  2.1 The propeller blade design process .................. 15
  2.2 The velocity computation ............................. 18
    2.2.1 The velocity induced by a vortex/source element .. 18
    2.2.2 The circumferential mean velocity ................. 20

3 The axisymmetric approach .............................. 25
  3.1 Principle of the axisymmetric theory .................. 25
  3.2 The tangential induced velocity ........................ 26
  3.3 The axisymmetric simple elements ....................... 29
    3.3.1 The vortex ring .................................... 29
    3.3.2 The source ring ................................... 30
    3.3.3 The generalized actuator disk ....................... 31

4 Computation of the loading and thickness effects .......... 35
  4.1 Introduction ......................................... 35
  4.2 The vortex influence computation ....................... 36
    4.2.1 The elementary segment ............................ 36
    4.2.2 The elementary axisymmetric vortex contribution .... 37
    4.2.3 The effects of pitch angle ........................ 38
4.2.4 Comparisons between velocities induced by an elementary rotating vortex segment and a vortex ring ........................................ 40
4.2.5 The final integration: A Romberg scheme .................................. 46
4.3 The source influence computation .................................................. 49
  4.3.1 The elementary axisymmetric source contribution .................... 49
  4.3.2 The final integration of thickness effects ............................... 50
4.4 Conclusion ................................................................................. 53

5 Implementation of the new subroutine in the PBD program ............ 54
  5.1 Possible bugs in the ring subroutine ........................................... 54
    5.1.1 The integration process .................................................... 54
    5.1.2 The single-ring subroutine ............................................... 55
  5.2 Tests inside the PBD environment ............................................. 67
    5.2.1 Modifications in the PBD program ..................................... 67
    5.2.2 Optimizing tests ............................................................ 67

6 Conclusion .................................................................................... 78

A RING/new CMVSEG subroutine code ........................................... 81
B Modifications in the PBD hscmv.f file ......................................... 89
C PBD global velocity outputs comparison tests .................................. 91
  C.1 Grid 10x10 ........................................................................... 91
  C.2 Grid 20x20 ........................................................................... 107
List of Figures

2-1 The approximation fields in VORSEG .............................................. 19
2-2 (a) Circumferential mean velocity and (b) Infinite-element induced ve-
locity ........................................ 21
2-3 Element coordinates used in CMVSEG ........................................... 22
2-4 Structure of CMVSEG subroutine ................................................. 23
2-5 The four integration zones in CMVSEG subroutine ......................... 24

3-1 The tangential circumferential mean velocity for a vortex lattice ..... 27
3-2 The tangential circumferential mean velocity for a horseshoe lattice . 28
3-3 The actuator disk decomposition .................................................. 32
3-4 The computation of an annulus of radial vortex lines .................... 33
3-5 The computation of a finite tube of helical vortex lines ................. 34

4-1 Decomposition of the elementary vortex segment ......................... 37
4-2 Axisymmetric decomposition ..................................................... 39
4-3 Comparison of axial circumferential mean velocities induced by an el-
ementary vortex segment at distance R from the X-axis (CMVSEG)
and a vortex ring of radius R (RING), at varying control points P(X,Rp) 41
4-4 Comparison of radial circumferential mean velocities induced by an
elementary vortex segment at distance R from axis and a vortex ring
of radius R at control points in plane x=R/2 ................................ 42
4-5 $\frac{V_{z/r-CMVSEG}}{V_{z/r-RING}}$ at various control points with varying tolerance 43
4-6 Discontinuities of $\frac{V_{z/r-CMVSEG}}{V_{z/r-RING}}$ and $V_{z/r-CMVSEG}$ with control points
in plane x=0.1R (Tolerance = 5 x 10^{-6}) .......................................... 44
4-7 $V_{z-CMVSEG}$ with varying tolerance, at control points coplanar with the elementary vortex element situated in plane $x=\text{constant}$ and at distance $R$ from the $x$-axis ........................................ 45
4-8 The pitch angles for the regular element ............................... 47
4-9 The axial pitch angle .................................................. 47
4-10 The radial pitch angle ................................................ 48
4-11 A simple tested element .............................................. 50
4-12 Relative error between old $CMVSEG$ and $RING$ subroutine outputs for a specific element and a radially moving control point .......... 51
4-13 Relative error between old $CMVSEG$ and $RING$ subroutine outputs for a specific element and an axially moving control point .......... 52

5-1 Relative error between $CMVSEG$ and $RING$ outputs for control points at $Xp/R=0.01$ from the generated surface ..................... 57
5-2 Axial induced velocities for $CMVSEG$ and $RING$ subroutines at $Xp/R=0.01$ from the surface ........................................ 58
5-3 Relative error between $CMVSEG$ and $RING$ outputs for control points at $Xp/R=0.001$ from the generated surface .............. 59
5-4 Axial induced velocities for $CMVSEG$ and $RING$ subroutines at $Xp/R=0.001$ from the surface ........................................ 60
5-5 Radial induced velocities for $CMVSEG$ and $RING$ subroutines at $Xp/R=0.001$ from the surface ........................................ 60
5-6 Radial source induced velocity for different number of Romberg cycles in $RING$ subroutine ........................................ 61
5-7 Accuracy comparisons between $CMVSEG$ and $RING$ subroutines for a single ring (Radially moving control point) ................. 64
5-8 Accuracy comparisons between $CMVSEG$ and $RING$ vortex outputs for a single ring (Axially moving control point) ............... 65
5-9 Accuracy comparisons between $CMVSEG$ and $RING$ source outputs for a single ring (Axially moving control point) ............... 66
5-10 Axial velocity in PBD vs. tolerance .......................... 71
5-11 Normal velocity in PBD vs. tolerance .......................... 71
5-12 Maximum differences between PBD output, using various tolerance in
the integration .................................................. 72
5-13 Tolerance choice in the integration process .................. 72
5-14 Accuracy in PBD vs. Romberg cycles ....................... 74
5-15 Computation time vs. Romberg cycles ....................... 76
5-16 Computation time vs. tolerance ............................... 77

C-1 Relative errors old CMVSEG/RING with loading and thickness (LT) 92
C-2 Normal velocity (LT with inflow) ................................ 93
C-3 Normal velocity (LT without inflow) ........................... 93
C-4 x-velocity (LT with inflow) .................................... 94
C-5 x-velocity (LT without inflow) .................................. 94
C-6 y-velocity (LT with inflow) .................................... 95
C-7 y-velocity (LT without inflow) .................................. 95
C-8 z-velocity (LT with inflow) .................................... 96
C-9 z-velocity (LT without inflow) .................................. 96
C-10 Relative errors old CMVSEG/RING with loading only (L) ...... 97
C-11 Normal velocity (L with inflow) ............................... 98
C-12 Normal velocity (L without inflow) ........................... 98
C-13 x-velocity (L with inflow) .................................... 99
C-14 x-velocity (L without inflow) .................................. 99
C-15 y-velocity (L with inflow) .................................... 100
C-16 y-velocity (L without inflow) .................................. 100
C-17 z-velocity (L with inflow) .................................... 101
C-18 z-velocity (L without inflow) .................................. 101
C-19 Relative errors old CMVSEG/RING with thickness only (T) ... 102
C-20 Normal velocity (T with inflow) ............................... 103
C-21 Normal velocity (T without inflow) ........................... 103
C-22 x-velocity (T with inflow) ........................................... 104
C-23 x-velocity (T without inflow) ........................................ 104
C-24 y-velocity (T with inflow) ........................................... 105
C-25 y-velocity (T without inflow) ........................................ 105
C-26 z-velocity (T with inflow) ........................................... 106
C-27 z-velocity (T without inflow) ........................................ 106
C-28 Relative errors old CMVSEG/RING with loading and thickness for
lattice 20x20 (LT2) ............................................................ 108
C-29 Normal velocity (LT2 with inflow) .................................... 109
C-30 Normal velocity (LT2 without inflow) ................................ 109
C-31 x-velocity (LT2 with inflow) ........................................... 110
C-32 x-velocity (LT2 without inflow) ........................................ 110
C-33 y-velocity (LT2 with inflow) ........................................... 111
C-34 y-velocity (LT2 without inflow) ........................................ 111
C-35 z-velocity (LT2 with inflow) ........................................... 112
C-36 z-velocity (LT2 without inflow) ........................................ 112
List of Tables

5.1 Divergence domain for the $RING$ subroutine .................. 62
5.2 $Blowing$ domain for the $RING$ subroutine ...................... 62
5.3 Thickness distribution for the tested propeller (Radius $R=D/2$) ... 68
5.4 Critical parameters for $diverging$ control points .................. 70
5.5 Number of cycles used in a run of PBD .............................. 74
5.6 PBD mean running times for both $CMVSEG$ subroutines ............ 77
Chapter 1

Introduction

The design of blade shapes of marine propellers is a major step in the hydrodynamic design of propulsion devices. The constant optimization of the process has proved to enhance the performance of various types of propulsors. Therefore, researchers and engineers are constantly searching for ever-increasing improvements in this process. One of the major concerns propeller designers have to cope with is the computation of the induced velocity in the vicinity of a propeller. In the M.I.T. propeller blade design program, this is currently done according to algorithms that are accurate but tend to waste computation time. The purpose of this thesis is to evaluate induced velocities of propellers using new mathematical approaches and computational methods in order to reduce computation time without affecting the accuracy of the final result.

The method for designing marine propellers has evolved considerably since the marine screw propeller was first used as means of propulsion in the eighteenth century. Traditionally, the blade geometry was developed from the performance characteristics of propeller standard series obtained through systematic model tests. Significant developments in the theory of lifting surfaces applied to propulsors (Prandtl’s lifting line theory [10], and lifting surface problems for propellers [11]) coupled with developments in numerical methods have led to consequent improvements in propeller and design method performance.
The design of propellers is basically conducted in three steps ([6] gives a summary of the fundamental theory). First, a radial and chordwise loading distribution of circulation over the blades is implemented in order to produce the expected thrust. The second step consists of adjusting the shape of the blade that will produce this loading according to propeller lifting surface theory. This is accomplished with respect to the kinematic boundary condition, which requires the computation of velocities induced by the propeller. The first two steps are labelled as the Design problem. The final step is to check if the designed shape will produce the expected thrust as well as the performance of the designed propeller (Analysis problem). The global process is, however, iterative as each step may involve changes in the others.

Based on the prediction of propeller performance by numerical lifting-surface theory [8], numerical methods for propeller design and analysis problems in relative simple flows have been developed by D.S. Greeley and J.E. Kerwin [1]. Recently, these methods have been unified at the M.I.T. Department of Ocean Engineering to allow the design of propeller under more constraints (Coupled Potential/Viscous flow; multi-stage, wake-adapted and ducted propellers) [7]. The procedure consists of decomposing the flow field into an axisymmetric and a non-axisymmetric part and is done in the Propeller Blade Design (PBD) program. The coupling of these two flow problems requires the computation of the velocity induced by the propeller and of the circumferential mean induced velocity. The blade is discretized using a lattice method and is therefore described by a spanwise and chordwise distribution of vortex/source elements. The algorithm currently used computes the influence coefficients due to a vortex-source element (VORSEG). Then, to obtain the circumferential mean induced velocity, another subroutine, named CMVSEG, calls VORSEG at a sequence of angular positions and finds the mean contribution. A brief review of the background of hydrodynamic design of marines propellers blades and of the current computation of induced velocity is given in chapter 2.

The algorithms in use happen to be generally robust but show also other flaws
(time consuming, lack of internal documentation). In this thesis, emphasis has been put on improvements of the quality and efficiency of these procedures. The original CMVSEG subroutine required almost 90% of the computation time in the standard Propeller Blade Design program. To decrease this waste of computation resources and to improve computation performances, a new mathematical and numerical approach had to be investigated. Based on the axisymmetric properties of the computation of the circumferential velocities, the fundamentals of our theory are exposed in chapter 3, as well as the specific case of the circumferential mean tangential velocity.

The influence of both loading effects (vortex singularities) and thickness effects (source distribution) needs to be taken into account in the algorithm. The axisymmetric problem is now decomposed into vortex/source rings that contribute to the axial and radial influence functions, the tangential component being obtained in a different subroutine inside the PBD program. Knowing these contributions allows a numerical integration on the initial segment rather than a circumferential mean integration, which requires much time. Chapter 4 presents the theory adopted and the results for the computation of loading and thickness effects in the case of a single vortex/source segment.

Once the algorithm programmed, it obviously needs to be tested for safety and robustness to prevent any numerical errors due to the assumptions made. Improving the programming style of our algorithms and optimizing the needs for computer resources have been the major goal throughout this whole work. It involves the addition of internal comments to increase the readability of our procedures and a detailed description of all the possible bugs that may occur when using the new subroutine. The new code needs also to be implemented into the PBD program as it is meant to replace the previous version of CMVSEG. Several comparisons tests have been made between the two versions of CMVSEG and the integration process has been optimized with respect to accuracy, efficiency and optimum running-time. This led to a major gain in computation time without losing the necessary accuracy. Results and safety
tests are shown in chapter 5.

Finally, chapter 6 lists conclusions reached during the course of the work and recommendations for possible further improvements in the subroutines as well as in their use.
Chapter 2

Background

2.1 The propeller blade design process

Blade shapes of a wide variety of propeller types can be designed. This process is currently done at the Ocean Engineering Department of the Massachusetts Institute of Technology using the Propeller-Blade-Design Program (PBD). This program and its use has been described and studied in various publications [9].

Using numerical lifting surface theory, the technique, which is based on the work by Greeley and Kerwin [1], happens to be very powerful and easily adaptable to many different propellers. Nevertheless, this theory, established in 1982, supposed a hubless, ductless, single-stage propeller operating in potential flow, which nowadays has proved not accurate and valid enough in certain propeller design problems.

This technique was then recently developed to meet the requirement of an increasing technological demand in propeller design [7]. Various theories, such as the use of B-spline surfaces to represent the shape, have been unified to improve the PBD-10 program to allow the design of single or multi-stage open and ducted propulsors. One of the major recent improvements has been made on the flow treatment.

One assumption in [1] was that the flow was considered as a potential flow, and therefore did not account for vorticity in the incoming flow field. Nevertheless, because of the boundary layer and wake of the upstream vehicle, vorticity is, generally speaking, present in the flow. The vorticity is transported by a velocity field includ-
ing the induced velocities from the propeller. Therefore, the inflow to the propeller happens to be modified by the propeller itself, and we must take into account the difference between the measured nominal inflow and the effective inflow to which the propeller must be designed.

The current PBD-10 program now couples two flow solvers to solve that problem. First, there is a viscous flow solver, namely a Reynolds-averaged Navier-Stokes (RANS) solver, which can explicitly model the transport of vorticity and capture separation. Then a potential flow solver, based on a vortex-lattice lifting surface model, is used because of its better and faster geometric manipulation characteristics (especially to avoid a too-slow regridding process with the RANS solver).

The viscous flow computation is done in an axisymmetric RANS code. This stage will include the presence of a duct or a hub, if required, as well as the vorticity in the incoming flow. The velocity field obtained through the RANS calculation is then input into the potential flow solver, which will treat the inviscid problem around the blades of the propulsor and issue a blade shape for the propeller. The distribution of effective velocity can then be used as new inputs to the viscous flow solver. The iteration is then carried on until convergence is reached.

As shown in [7], the issues at stake here are to determine the accurate relationship between the two flows, how the forces in the two problems are related, and how to evaluate the correct effective velocity from the total velocity distribution given by the potential flow solver.

The first solver solves the flow around an axisymmetric body, including a duct, if present. All flow quantities corresponding to this flow are noted here with a superscript $\circledcirc$. The total velocity will be, for instance, noted as $V^{\circledcirc}$.

The second solver deals with the flow produced by a set of blades operating in a given axisymmetric inflow field. All flow quantities corresponding to this flow (called the hull flow) are noted here with a superscript $\circledast$. The total velocity will be there noted as $V^{\circledast}$.

The total velocity in the blade problem can be decomposed as follows:
\[ V^\circ = V_e^\circ + V_i^\circ + \hat{V}_i^\circ, \quad (2.1) \]

where:

- \( V_e^\circ \) is the effective inflow velocity,
- \( V_i^\circ \) is the circumferential mean induced velocity, and
- \( \hat{V}_i^\circ \) is the circumferentially fluctuating component of the induced velocity.

This equation confirms the definition of the effective inflow velocity, which consists of the total velocity minus the potential flow velocity induced by the blades. If we are given \( V_e^\circ \), we can solve for the propulsor-induced velocity using traditional potential flow methods. But as the effective flow cannot be easily determined or measured, we need to make the important assumption that the effective velocity is axisymmetric and can be obtained from the RANS solver:

\[ V_e^\circ = \tilde{V}_e^\circ = \hat{V}_e^\circ. \quad (2.2) \]

At this stage, it is understood that this assumption is not totally correct if some vorticity is present in the inflow (the time-varying flow field induced by the rotating propeller varies the stretching of the vortex field at a given time). Nevertheless, as the induced velocity upstream of the propeller is mainly dominated by its circumferential mean component (the blade-rate harmonics being more attenuated with increasing distances than the mean component), the resulting velocity field is also nearly axisymmetric. This result gets more accurate with an increasing number of blades, but is not valid very close to the blades. Nevertheless, computing the flow deformation with such accuracy would require solving the complete flow problem with an unsteady three-dimensional code that cannot be completed with sufficient efficiency.

The force field also needs to be determined in order to be applied back in the axisymmetric solver and complete the iterative process. Based on vortex-source lattice methods, the concentrated forces on each element of the lattice can be computed according to Kutta-Joukowski’s law and Lagally’s theorems. This calculation involv-
ing total local velocities should produce the same circumferential mean flows in the axisymmetric solver as in the blade problem:

\[ \bar{V}^\circ = \bar{V}_e^\circ + \bar{V}_i^\circ. \]  

(2.3)

The necessary correct force field and the assumptions needed to achieve it are presented and discussed in more detail in [7]. Once obtained, the force field allows the derivation of an effective velocity by subtracting the circumferential mean blade solution induced velocity from the total velocity of the hull solution:

\[ V_e^\circ = \bar{V}_e^\circ = \bar{V}_e^\circ = \bar{V}^\circ - \bar{V}_i^\circ. \]  

(2.4)

The computation of the induced velocity and the circumferential velocity appears to be a major point in the process of determining the flows around the propeller, and thus in the global design process of the propeller.

2.2 The velocity computation

The propeller blade design requires the computation of the velocities induced at a set of points on the blade by a prescribed spanwise and chordwise distribution of circulation and thickness using a vortex/source lattice method. This procedure is based on the method whose fundamentals are revealed in [7] and that has been recently developed in order to treat more specific flows. The positions and strengths of all the discrete vortex/source elements are determined at an early stage of the process. The computation of the velocity can then be pursued on every one of these single elements.

2.2.1 The velocity induced by a vortex/source element

Knowing the positions and strengths of every vortex/source element, the induced velocity at each control point on the blades due to each vortex/source element can be computed. It is currently done through a subroutine called VORSEG, which has been
Figure 2-1: The approximation fields in *VORSEG*

carried out from Greeley and Kerwin's paper [1]. It is based on the fundamental Biot-Savart law, which allows one to compute the velocity induced at any control point by a three-dimensional vortex element of known strength. The effect of thickness that has been modeled by a source distribution leads also to the computation of the velocity field induced by a straight-line source element of known constant strength. It is conducted in *VORSEG* by integrating the gradient of the source potential along the element. The computation is done according to the position of the point relative to the vortex element taken into consideration (figure 2-1). If the field point is in the suburban field or in the near field, we use the exact formulas as shown in [1]. The only difference comes from the level of accuracy required (single or double precision). The double precision was initially used for the near field computation, but has been afterwards removed (without any notice or comment from the programmer) as all the computation in the PBD program is made in single precision which proved to be sufficient for propeller design. In the far field, the formulas are simplified to save some computation time. The computation is done in a global XYZ-coordinate system.
2.2.2 The circumferential mean velocity

Besides the induced velocity, the current blade shape designing process needs to compute the circumferential mean induced velocity in order to subtract it from the total induced velocity at each control point and then allow the computation of the effective inflow velocity. This is currently done under a subroutine called CMVSEG, which accomplishes the calculation by calling VORSEG at a sequence of angular positions starting with the blade control point and finding the mean induced velocity. In effect, the circumferential mean velocity induced by a vortex/source element at any control point is the same as the velocity induced by an infinite number of elements obtained by rotating the initial element around the axis of symmetry (figure 2-2). This result is equivalent to the one already used in propeller theory to compute the circumferential mean velocity induced by a finite bladed propeller, which happens to be the same as the velocity induced by an infinite bladed propeller [6].

The computation is done by rotating around the X-axis the initial element defined by the coordinates \((R_1, T_1)\) and \((R_2, T_2)\) in the YZ-plane of its two extremities (also defined by their respective x-coordinates \(x_1\) and \(x_2\) that are not relevant to our case as the rotation is around the X-axis). The circumferential integration is done from the initial angles \(T_1\) and \(T_2\) as shown in figure 2-3. The subroutine CMVSEG uses a Romberg method to perform the integration in order to obtain the required accuracy. The structure of the algorithm is exposed in figure 2-4.
Figure 2-2: (a) Circumferential mean velocity and (b) Infinite-element induced velocity.
Discrete Element projected in a plane $X=\text{constant}$

Figure 2-3: Element coordinates used in $CMVSEG$
CMVSEG

Set Initial Angular Increment: -Tz/Tz
(Currently of 3.0E-2)

\[ N_{zone} = 1 \text{ to } 4 \]

Test If Thickness

NO

Output Velocity Array dimension = 3

YES

Output Velocity Array dimension = 6

Set Lower And Upper limits of integration
\[ X_l = -Tz \quad X_r = +Tz \]

1st Integration: Trapezoidal rule
Function called: VORSEG
Integration done on \([X_l,X_r] \]

1st rough value

Romberg Integration Scheme
Each Interval of integration is subdivided in 2 sub-intervals
Romberg integration computation calling VORSEG

Test on convergence (w/ Chosen tolerance)

NO

Convergence?

YES

Store Value into Output Array

Change Integration Zone
Input new Zone angular increment

\[ N_{zone} < 4 \]

\[ N_{zone} = 4 \]

Output = Circumferential Mean velocity vector induced by \([A_1,A_2] \) on P

Figure 2-4: Structure of CMVSEG subroutine
The number of circumferential intervals in the Romberg integration method is progressively doubled until convergence is achieved. The required number of these intervals increases as the angle to the initial element gets smaller. On the other hand, the integration path requires fewer intervals to fulfill the same convergence rate as we integrate further from the element. It is therefore useful to divide the integration into four distinct zones as exposed in figure 2-5. The accuracy is then still fulfilled, and the global computation takes less time as the number of integration intervals is decreased. Although the results obtained are very accurate, the number of VORSEG calls happens to be amazingly important and therefore leads to a waste of computation time.

Figure 2-5: The four integration zones in CMVSEG subroutine
Chapter 3

The axisymmetric approach

3.1 Principle of the axisymmetric theory

As shown in chapter 2, the circumferential mean velocity induced by an element is obtained by computing the mean value of the velocities induced by this element while rotated at various angular positions until convergence is reached. This approach consists in performing the numerical integration on the rotation of the element, as we know how to compute the contribution of a single element. As the computation time depends on the number of angular intervals of integration (that happens to be really large), the main idea of a different approach is to avoid this time-wasting numerical integration. CMVSEG is evaluating analytically the single induced velocity and numerically the angular integration:

$$\bar{V} = \frac{Z}{2\pi} \int_0^{2\pi} \bar{V}_{\text{seg}} d\theta$$  \hspace{1cm} (3.1)

where:

- \(Z\) is the number of blades of the propeller and,
- \(\bar{V}_{\text{seg}}\) is the velocity induced by a vortex segment (VORSEG Output)

Of course, we cannot avoid a numerical integration as we cannot solve the whole problem analytically. But we can face the problem in a different manner.
The problem is obviously axisymmetric. Once the vortex element has been rotated around the axis, we obtain an axisymmetric vortex surface distribution. The velocity induced by this surface is the same as the one induced by the rotating elements, and then can easily be related to the circumferential mean induced velocity. The approach followed in this work is to determine the axisymmetric contribution analytically and to perform the numerical integration on the element.

\[ \vec{V} = 2 \int_{\text{element}} (dV)_{\text{axis}} dl. \]  

(3.2)

Dealing with the surface distribution rather than the rotation of the single element avoids the angular numerical integration. In effect, the axisymmetric surface can be decomposed into singular axisymmetric vortex elements (ring, cone, tube) of which we know how to compute their influence functions according to their shape, position relative to the control point and other characteristics. The final numerical integration is than conducted on the initial element and should be less time-consuming than the angular computation to reach the required convergence, as each elementary axisymmetric contribution does not depend on its position on the element (whereas in CMVSEG, depending on the angular integration zone, the number of integration intervals varies widely to be really high when close to the element).

### 3.2 The tangential induced velocity

It is important to notice that the tangential velocity induced by the whole propeller does not require the same computation as the radial and axial induced velocities. The influence functions of a single element are needed in order to be applied to every single element, obtained by the vortex-lattice method that describes the real blade of the propeller. The total contribution of all these single elements allows one to evaluate the circumferential mean velocity induced by the propeller on a control point. Nevertheless, we can evaluate the global tangential circumferential mean induced velocity directly according to Kelvin's theorem.
Figure 3-1: The tangential circumferential mean velocity for a vortex lattice

The blades of the propeller are, as seen before, described with a vortex lattice method. As shown in figure 3-1, if we consider a field point P inside a panel element and if we call $\Gamma_i$ the spanwise elements upstream on the blade, we can easily compute the induced tangential velocity at P according to Kelvin’s theorem:

$$-2\pi r_P \bar{u}_t = \sum_{i=1}^{M} \Gamma_i. \quad (3.3)$$

Introducing the non-dimensionalized circulations $G_i = \frac{\Gamma_i}{2\pi RV_s}$ (with $V_s$ the flow velocity and $R$ the characteristic length of the propeller), we end up with the following
\begin{equation}
\frac{\bar{u}_t}{V_s} = - \frac{Z}{r_p/R}.
\end{equation}

For the PBD program, the calculation is even simpler. In effect, vortex horseshoes are used to discretize the blade (figure 3-2). The choice of horseshoes proved significant improvements in accuracy, computation times and fluidity in the numerical approximations. Therefore, the PBD program needs only to compute the influence function of vortex horseshoes. According to equation 3.4, we have:

For a control point inside the horseshoe:

\begin{equation}
\frac{\bar{u}_t}{V_s} = - \frac{Z}{r_p/R}.
\end{equation}

For a control point outside the horseshoe:

\begin{equation}
\frac{\bar{u}_t}{V_s} = 0.
\end{equation}

Therefore, at any control point we can apply this result to any horseshoe vortex.
(obtained from each spanwise lattice element). It just consists in testing the position of the control point with the horseshoe to be considered. According to their relative position, we know what the velocity contribution at this point will be. We can then add the contribution of every horseshoe to obtain the tangential component of the circumferential velocity induced by the whole blade. The problem now is to solve for the axial and radial components of the circumferential mean velocity.

3.3 The axisymmetric simple elements

Before doing any work on the axisymmetric vortex surface distribution, it is useful to examine simple axisymmetric elements whose influence functions can be easily computed. The main simple elements that will be taken into account in our case are the vortex ring and the generalized actuator disk.

3.3.1 The vortex ring

A vortex ring is obtained from a vortex filament which forms a circle of radius \( r' \). The induced velocity is computed according to the Biot-Savart law:

\[
dv = \frac{\Gamma}{4\pi} \frac{\mathbf{R} \times ds}{R^3},
\]

(3.7)

where \( dv \) is the velocity vector induced by an element of length \( ds \) at the control point \( P(x, r, \theta) \), \( ds/ds \) is the unit vector tangential to the vortex element, and \( \mathbf{R} \) is the radius vector from the vortex element to the control point. This expression is then integrated around the circumference defined by the ring in order to obtain the whole vortex ring-induced velocity. The tangential component \( (w) \) is zero in the case of the ring. The complete derivation of the induced velocities is shown in [4]. The final results, for a ring in the plane \( x=0 \), give the following axial \( (u) \) and radial \( (v) \) influence functions \( (\Gamma/2\pi R = 1) \) expressed as a function of elliptic integrals:

\[
\bar{u} = \frac{1}{r'\sqrt{x^2 + (r + 1)^2}} \left\{ K(k) - \left[ 1 + \frac{2(r - 1)}{x^2 + (r - 1)^2} \right] E(k) \right\},
\]

(3.8)
\[
\bar{v} = \frac{-x}{rr'} \sqrt{x^2 + (r + 1)^2} \left\{ K(k) - \left[ 1 + \frac{2r}{x^2 + (r - 1)^2} \right] E(k) \right\}, \quad (3.9)
\]

\[
\bar{w} = 0, \quad (3.10)
\]

where \(x\) and \(r\) have been non-dimensionalized as \(x/r'\) and \(r/r'\).

The elliptic integrals are given as follows:

\[
K(k) = \int_0^{\pi/2} \frac{1}{\sqrt{1 - k^2 \sin^2 \alpha}} d\alpha, \quad (3.11)
\]

\[
E(k) = \int_0^{\pi/2} \frac{1}{\sqrt{1 - k^2 \sin^2 \alpha}} d\alpha, \quad (3.12)
\]

with

\[
k^2 = \frac{4r}{x^2 + (r + 1)^2}. \quad (3.13)
\]

### 3.3.2 The source ring

A source ring is a distribution of sources of constant strength along a circle. Using the same notation as for the vortex ring, the velocity induced by an element of strength \(q \, ds\) of the source ring of radius \(r'\) at the point \(P\) defined by the vector \(R\) is:

\[
dv = \frac{qds}{4\pi R'^2} \frac{R}{R}. \quad (3.14)
\]

The complete derivation of the induced velocities leads to express the axial and radial influence functions in terms of the same complete elliptic integrals (with the
The elliptic integrals $K$ and $E$, and therefore the various influence functions at stake, can easily be computed. From now on, the axial and radial velocities induced by a vortex/source ring are computed through a subroutine called the $RING$ subroutine.

### 3.3.3 The generalized actuator disk

The theory of the generalized actuator disk has been derived by Hough and Ordway [3]. The main idea is to compute the induced velocities of an infinite-bladed propeller with arbitrary circulation based upon the classical vortex system representation. They have also developed expressions for the velocity induced by an infinite set of helical vortices uniformly distributed around a circumference [6]. The generalized actuator disk formulas are not of direct use in our case, but their derivation is made in two steps that will help in the course of our derivation. The bound vortices influence and the free vortices influence are evaluated separately to be added afterwards and give the global influence function (figure 3-3).

#### The bound vortices

The bound blade vortex lines are radial lines. For an infinite number of lines, we have therefore the expression of a disk of radial vortex distribution. The important result here is that these bound vortices induce only tangential velocity. For a continuous distribution on a disk located in the plane $x = 0$ with total circulation of $\mathcal{Z}$, we have the following induced velocity component for a circumference of radius $r'$ and a
control point defined as \( P(x,r,\theta) \):

\[
\begin{align*}
    u &= 0 \quad \text{(3.18)} \\
    v &= 0, \quad \text{(3.19)} \\
    w &= \frac{Ze}{2\pi^2r^{3/2}} \int_0^r \frac{Q_{1/2}(\omega)}{\tau_v^{3/2}} d\tau_v, \quad \text{(3.20)}
\end{align*}
\]

with \( Q_{1/2} \) Legendre function of the second kind and half-integer order with argument \( \omega \):

\[
\omega = 1 + \frac{x^2 + (r - r')^2}{2rr'}.
\]  

Therefore, in our case a disk of such vortex lines will not induce any radial or axial velocity and should not be taken into account. We can go even further. In fact, with two well-chosen disks (coplanar, concentric but of different radii), we can derive the velocity induced by an annulus or radial vortex lines (figure 3-4). The induced velocity in this case is obviously tangential.

**The free vortices**

The second part of Hough and Ordway actuator disk derivation deals with the trailing free vortices. Those consist of infinite helical lines distributed on the previous circumference with a pitch angle \( \beta \). The velocity induced at \( P(x,r,\theta) \) by a continuous
Velocity induced by an Annulus of Radial Vortex lines Uniformely distributed

Velocity induced by an Disk of radial bound Vortex lines Radius R2

Velocity induced by an Disk of radial bound Vortex lines Radius R1

Figure 3-4: The computation of an annulus of radial vortex lines distribution on a circumference of radius $r'$ with total circulation of $Z$ are given as follows:

\[ u = \frac{Z}{4\pi} \left[ \frac{K_1}{\pi r' \tan \beta} \right] \]  
\[ v = \frac{-Z}{4\pi^2 \sqrt{rr'} \tan \beta} Q_{1/2}(\omega) \]  
\[ w = \frac{Z}{4\pi^2 r} K_2, \]

with

\[ K_1 = \pi + \frac{x}{2\sqrt{rr'}} Q_{-1/2}(\omega) + \frac{\pi}{2} \Lambda_0(s, t), \]

\[ K_1 = \frac{x}{2\sqrt{rr'}} Q_{-1/2}(\omega) - \frac{\pi}{2} \Lambda_0(s, t). \]
The induced velocity by a finite tube of helical lines, located at \(x + L\), is identical to the induced velocity by an infinite tube of infinite helical lines located at \(x\), with the radial inequality signs inversed.

\[ K_2 \text{ is identical to } K_1 \text{ with the radial inequality signs inversed.} \]

\( \Lambda_0(s,t) \) is the Heuman Lambda function with arguments:

\[
\begin{align*}
  s &= \arcsin \frac{x}{\sqrt{x^2 + (r - r')^2}} \\
  t &= \sqrt{\frac{4rr'}{x^2 + (r + r')^2}}.
\end{align*}
\]  

(3.27)

These expressions are given for a semi-infinite tube. We can, therefore, derive the result for a finite tube of length \(L\) by subtracting the effects of a semi-infinite tube of radius \(R\) at position \(x\) to the effects of another semi-infinite tube of same radius at position \(x + L\) (using the same pitch angle). This allows us to evaluate the influence of a finite tube at any control point in space (figure 3-5). Finally, for a pitch angle of 90 degrees, the helical lines become straight lines parallel to the axis of symmetry. According to the equations (3.22-3.24), we can notice that the induced velocity is only tangential in that case (as \(\tan \beta \to \infty\) when \(\beta \to 90^\circ\)).
Chapter 4

Computation of the loading and thickness effects

4.1 Introduction

The main idea in our method is to use the influence functions of the axisymmetric elements considered in the previous chapter, to compute the axial and radial circumferential mean induced velocities for any segment. A vortex/source segment, when rotated around the axis of symmetry, will generate an axisymmetric surface, that can be decomposed into axisymmetric elements. The most simple description of the surface would be to use vortex/source rings. For the source distribution, this choice seems relevant: The sources are distributed along the segment which generates a source surface distribution when rotating around the axis. As a source has no specific direction (as a dipole or a vortex), the surface can be decomposed into source rings centered around the axis, and the computation of thickness effect can be easily made. Nevertheless, for the vortex influence computation, the direction of the vortex is an important information. When using CMVSEG, the vortex vector is tangent to the element, whereas, in the case of rings, the vortex vector is tangent along the filament that forms the ring. Therefore, simply using vortex rings to described the surface, as with the source distribution, would not take into account the effects of the position of the vortex vector (and hence of the element). The derivation of the vortex influence
needs to be conducted with more attention to keep every relevant information.

4.2 The vortex influence computation

4.2.1 The elementary segment

The numerical computation of lifting surface theory requires many approximations for practical purposes. This is naturally related to the discrete representation of governing equations at stake, which are normally continuous. Nevertheless, as long as the level of accuracy is obtained, these approximations are justified, even recommended as they usually help in minimizing the added computational time expenses. Therefore, we can first deal with an elementary vortex element that will be rotated around the axis of symmetry (taken to be the x-axis from now on) and give an elementary vortex surface distribution. This element, which can have any position in space, can be decomposed in several ways that may lead to simpler computation of the induced velocity. As shown in figure 4-1, the element $A_1A_2$ is defined by the positions of its two extremities $A_1(x_1,R_1,T_1)$ and $A_2(x_2,R_2,T_2)$. The circles $C_1$ and $C_2$ are respectively centered on the axis of symmetry at positions $x_1$ and $x_2$, with radius $R_1$ and $R_2$. We can then decompose the element into three parts: an element parallel to the x-axis of symmetry $A_1'A_1$, a circumferential component $A_1'A_2$, following the circle $C_1'$ (centered on the axis at $x_2$ with radius $R_1$) and a radial component $A_2'A_2$ on the disk defined by the circle $C_2$.

Of course, there are a lot of other ways to decompose this element. We could, for instance, decompose it as a line on the tube defined by $C_1$ and $C_1'$ and a radial vortex line on the disk defined by $C_2$. We could also chose a different circumferential part (for example, on $C_2$ rather than $C_1$), while keeping a decomposition in three parts: axial, circumferential and radial. It is also true that the velocity induced by these three elements will not be exactly the same as the initial element induced velocity or as for a different decomposition. Nevertheless, we are deliberately dealing with an elementary element, thus very small. Though slightly different from the exact result,
this will give a good approximation of the velocity induced by the element itself and will ease the circumferential mean problem. When the element is rotated around the x-axis, $A_1A'_1$ will describe a tube $T$ of radius $R_1$, $A'_1A'_2$ will describe a ring $R$ of radius $R_1$ and $A'_2A_2$ will radially describe an annulus $An$. Therefore, the induced velocity by the rotating element can be decomposed in the contributions of three parts: the tube $T$, the ring $R$, the annulus $An$.

### 4.2.2 The elementary axisymmetric vortex contribution

Once the axisymmetric problem has been decomposed to ease the computation, we can now see the contributions of each axisymmetric part.

- The influence of the tube $T$ can be deduced from the Hough and Ordway formulas. In our case, the tube is described by an infinite set of vortex straight lines (pitch angle $\beta=0$). Therefore, the induced velocity here will be only tangential.
• The annulus $A_n$, as shown in the previous chapter, contributes only to the tangential component of the induced velocity.

• The ring $R$ will induce only radial and axial velocities at the control point.

This axisymmetric decomposition is summarized in figure 4-2. Naturally, the effect of the pitch angle needs to be taken into account in the ring contribution. It is also important to point out again that the figure is showing the decomposition in a large scale, whereas the decomposition is done for elementary segments, and therefore elementary axisymmetric parts ($R_1$ and $R_2$ are slightly different as well as $x_1$ and $x_2$).

### 4.2.3 The effects of pitch angle

The effect of pitch angle needs to be taken into account at this point. For instance, if one of the extremities of the element is rotated around the axis of symmetry, the pitch angle will vary and we will get different results for the induced velocity. The relevant parameter at this stage is the circumferential part in our decomposition, namely $A_1'A_2'$. This contribution can be easily computed from the initial element $A_1A_2$ (length $dl$) using the angles $\alpha$ and $\beta$ (figure 4-1).

As we are considering an elementary of length $dl$, $A_1'A_2'$ can be evaluate as follows:

$$A_1'A_2' = \sin \alpha \times \sin \beta \times dl. \quad (4.1)$$

This length needs to be non-dimensionalized by the circumference of the ring to give the correct fraction of the ring effects that needs to be taken into account. Finally, the radial and axial contribution of our elementary element is equal to:

$$d\vec{V}_v = \vec{V}_{v_{\text{ring}}} \times \frac{\sin \alpha \sin \beta dl}{2\pi R_{\text{ring}}}, \quad (4.2)$$

with $\vec{V}_{v_{\text{ring}}}$ the axial or radial velocity induced by the vortex ring of radius $R_{\text{ring}}$. $\sin \alpha$ and $\sin \beta$ can be evaluated as we know the cartesian coordinates of the different points involved in our decomposition. Nevertheless, as we will see in the ultimate step, we can compute them in an easier and faster way during the final integration.
Figure 4-2: Axisymmetric decomposition
4.2.4 Comparisons between velocities induced by an elementary rotating vortex segment and a vortex ring

Before computing the circumferential mean velocity induced by a larger vortex segment, the influence function of a single ring can be compared to the outputs of CMVSEG for a well-chosen element. Actually, if we take a rather small element in the CMVSEG computation, its rotation will give an elementary surface that can be approximated by a ring in its limit. Therefore, we can compare the outputs of the RING subroutine (modified to take into account the pitch angles) and the CMVSEG subroutine. This test was conducted for several elements with various positions in space and gave similar results. Figure 4-3 shows the comparisons between axial velocity outputs for an element of length $10^{-4}$ in a plane $x=$constant (taken to be zero for instance) for CMVSEG subroutine and a ring of radius $R$ ($R$ computed according to the position of the element) for the RING subroutine. This shows that the outputs are similar at any position except when the control point is far from the element. But in this latter case, the induced velocity tends to zero and therefore the differences may come from the machine approximations errors, and are in any case of no importance as the velocities are very small. Similar results have been obtained for the radial components (figure 4-4).

This test revealed some discontinuities in the CMVSEG outputs. Besides, a comparison of the ratios of the outputs shows similar conclusions for tests depending on the tolerance allowed in the two subroutines. As shown in figure 4-5, the ratio of axial velocities from both subroutines has significant discontinuities, even for low tolerances (and then more accuracy required). Even if the results for a large tolerance cannot be taken for granted because of their lack in accuracy, we can see that for the tolerance set in the CMVSEG subroutine in use ($5 \times 10^{-4}$), the absolute value of the ratio is not exactly equal to one and varies discontinuously as the control point moves away from the ring. This is nevertheless due to the approximations in the CMVSEG subroutine. Figure 4-6 gives zoomed plots of the axial velocity for CMVSEG and
Figure 4-3: Comparison of axial circumferential mean velocities induced by an elementary vortex segment at distance R from the X-axis (CMVSEG) and a vortex ring of radius R (RING), at varying control points P(X,Rp)
Figure 4-4: Comparison of radial circumferential mean velocities induced by an elementary vortex segment at distance R from axis and a vortex ring of radius R at control points in plane x=R/2
Figure 4-5: \(-\frac{V_{CMVSEG}}{V_{CMVSEG}}\) at various control points positions with varying tolerance

RING subroutines on the positions where the ratio shows discontinuities. The change comes from a sudden step in the CMVSEG output values, which is due to the change of integration zone (as explained in chapter 2) in order to keep up with the accuracy required (figures 4-7). These differences, though much noticeable if we compute the relative error or the ratio of the outputs of both subroutines, are, in any case, small compared to the effective values of the velocities, and are not very much significant, either in the use of CMVSEG or in the use of the RING approximation.

The ring approximation for the computation of circumferential axial and radial velocities seems to be a relevant choice in our case. Of course, as will be shown later, a lot of care has to be taken to prevent the errors in the use of the ring subroutine,
Figure 4-6: Discontinuities of $-\left(\frac{V_{x/r-CMVSEG}}{V_{x/r-RING}}\right)$ and $V_{x/r-CMVSEG}$ with control points in plane $x=0.1R$ (Tolerance = $5 \times 10^{-6}$)
Figure 4-7: $V_x^{CMVSEG}$ with varying tolerance, at control points coplanar with the elementary vortex element situated in plane $x=\text{constant}$ and at distance $R$ from the $x$-axis especially in the neighborhood of the ring where the computation may show significant flaws. Nevertheless, for control points in a field not too close to the rings at stake in our computation, we can use the theory exposed to conduct the integration upon any element.
4.2.5 The final integration: A Romberg scheme

The final step in our computation is to integrate the contributions of the elementary segments to obtain the axial and radial velocities of any element (i.e. not necessarily small). According to equation (4.2) and taking into account the number of blades of the propeller, we have for the total induced velocity

\[ \bar{V}_{z/r} = Z \int_0^D \bar{V}_{z/r-\nu_{\text{ring}}}(l) \times \frac{\sin \alpha(l) \sin \beta(l)}{2\pi R_{\text{ring}}(l)} \, dl \]  

with

- \( \bar{V}_{z/r} \), the axial or radial circumferential mean velocity induced by the vortex segment of length D,
- \( \bar{V}_{z/r-\nu_{\text{ring}}}(l) \), the radial or axial velocity induced by a vortex ring of radius \( R_{\text{ring}}(l) \) defined by the point M, at position l on the element (figure 4-8),
- \( \sin \alpha(l), \sin \beta(l) \) as defined in the previous section 4.2.3,
- \( Z \), the number of blades.

We can also evaluate the two angles in terms of known characteristics. On one hand, we can notice that \( \alpha(l) \) is the angle between the element and the x-axis (figure 4-8). Therefore, it is independent of the position on the element and can be evaluated from the coordinates of the extremities of our element \( A_1A_2 \) as follows:

\[ \sin \alpha(l) = \sin \alpha = \frac{M M'}{A_1 M} = \frac{A'_1 A_2}{A_1 A_2} = \frac{D_t}{D} \]  

with \( D_t = \sqrt{(y_2 - y_1)^2 + (z_2 - z_1)^2} \) the length of the element projected in a transverse plane as shown in figure 4-9.
Figure 4-8: The pitch angles for the regular element

Figure 4-9: The axial pitch angle
On the other hand, the angle $\beta(l)$ is dependent on the position on the element as it measures the radial pitch angle (and the radial component in the decomposition we follow varies with the position on the element)(figure 4-10). Nevertheless, we can compute it easily as we can evaluate the coordinates $x_M, y_M, z_M$ of the point $M$, at position $l$ on the element:

$$\sin \beta(l) = \frac{|\vec{O'M} \times \vec{A_1'A_2}|}{|\vec{O'M}| |\vec{A_1'A_2}|} = \frac{S_M}{D_l \times R_{ring}(l)}$$

(4.5)

with $S_M = y_M(z_2 - z_1) - z_M(y_2 - y_1)$ and $|\vec{O'M}| = R_{ring}(l)$.

Therefore, equation 4.3 becomes:
\[ \vec{V}_{v-z/r} = \frac{Z}{2\pi D} \int_0^D \vec{V}_{z/r-\nu\text{ring}(l)} \frac{S_M(l)}{R_{\text{ring}}(l)^2} dl \]  \hspace{1cm} (4.6)

\[ \vec{V}_{z/r-\nu\text{ring}(l)} \] can be computed with the RING subroutine, \( R_{\text{ring}}(l) \) and \( S_M(l) \) are deduced from the coordinates of M and of the extremities of the element as shown above. The numerical integration is done following the Romberg method [5] to ensure the right result accuracy. Of course, other faster numerical integrations can be used, but the Romberg method is a safe warranty for accuracy. Other methods could be used (especially according to the position of the control point) to increase the speed of the algorithm: for example, with a control point far from the element, a rougher integration may be enough to evaluate the influence functions (which will probably be close to zero) and would, then, necessitate less precision (and less running time) in the computation process. Nevertheless, at this stage, a Romberg integration is a safer method, and other methods should only be studied when the subroutine runs within the global PBD program, where significant differences (in computation time and accuracy, particularly) can be noticed in a more relevant way.

### 4.3 The source influence computation

#### 4.3.1 The elementary axisymmetric source contribution

As mentioned at the beginning of this chapter, the effects of thickness are more straight to compute. Thickness is represented by a source distribution along the considered segment. When this segment is rotating around the axis, each source describes a circle centered on the axis and generates a source ring. There is no pitch angle effects to be taken into account here as the source has no specific direction (as for a dipole). For an elementary segment of length \( dl \), the circumferential mean induced velocity in the axial and radial directions is simply the same as for a source ring:
$d\vec{V}_S = \vec{V}_{Sring} \times \frac{dl}{2\pi R_{ring}},$ \hspace{1cm} (4.7)

with $\vec{V}_{Sring}$ the axial or radial velocity induced by the source ring of radius $R_{ring}$.

### 4.3.2 The final integration of thickness effects

The final integration is conducted, as for the vortex influence, using a Romberg method:

$$\vec{V}_{S-x/r} = \frac{Z}{2\pi} \int_0^D \frac{\vec{V}_{x/r-Sring}(l)}{R_{ring}(l)} dl. \hspace{1cm} (4.8)$$

The computation of the vortex and source circumferential mean induced velocities for any element is done in a new CMVSEG subroutine, also called RING subroutine by extrapolation from the single ring computation. The final expressions of the vortex and source induced velocities have been tested on various vortex/source segments, not necessarily small as previously done. A simple choice is a 2D element in the plane $x=0$ as shown in figure 4-11. With well-chosen control points, the induced velocities
resulting from equations (4.6) and (4.8) have been compared with the outputs from the previous CMVSEG subroutine. A good way to test far field and near field control points is to move a control point far from the element, and the generated surface due to the rotation of the segment, toward the surface. Two relevant cases are when the control point varies radially (with $x=\text{constant}$) and axially (toward the generated surface). Relative errors between the two algorithms are shown in figures 4-12 and 4-13 for an element of length $D=2$ ($R_1 = 1$, $R_2 = \sqrt{2}$), and proved that the new theory is very accurate (relative errors of the order of $10^{-5}$). Neverthe less, for control points very close to the surface, the outputs seem to differ more significantly. This case will be more precisely studied in chapter 5, as well as the other possible bugs that may occur in the new subroutine.

Figure 4-12: Relative error between old CMVSEG and RING subroutine outputs for a specific element and a radially moving control point.
Figure 4-13: Relative error between old CMVSEG and RING subroutine outputs for a specific element and an axially moving control point
4.4 Conclusion

The main asset in our method is to combine both the effects of loading and thickness in the same subroutine. There is no extra time when the two effects are to be taken into account, as the same elliptic functions are used in both cases. The effects of thickness, which are generally speaking less significant than those due to circulation, will not increase the computation running time, which is a major saving in the use of PBD. Besides, as the integration is made upon the element, there is no specific cases as with the regular CMVSEG, where, for particular control points and elements, the number of integration on the rotating angle was tremendously high to reach the required accuracy, leading to a waste of computation time. The new RING subroutine should decrease the running-time without losing the accuracy of the results. As the accuracy seems to be correct with the new algorithm on single calls, the subroutine can be now tested inside the PBD program to allow comparisons in efficiency, accuracy and computation time in the case of practical uses of PBD in the blade design process.
Chapter 5

Implementation of the new subroutine in the PBD program

5.1 Possible bugs in the ring subroutine

At this stage, preventing the errors that may occur in the use of the new subroutine is a good way to strengthen computation safety. Although those errors may never happen in the actual use of the new code, it is nevertheless important to give a strong structure to the algorithm, at least to allow further users to fix problems that may occur. The following is a list of all the errors that may be encountered.

5.1.1 The integration process

1. The Zero segment

The algorithm is not valid for an element of length $D=0$, as this length is a denominator in the final expression. Hopefully, this case will never happen as a Zero segment does not have any influence on the flow computation.

2. The direction of the segment

The integration is conducted on the length of the element. Therefore, there will
be no problem due to the position of the element (if the element is perpendicular to the axis, for instance). The computation of the pitch angles involves the use of the length $D_t$ (equation 4.4, 4.5). Yet, this length vanishes in the global expression. Therefore, there will be no problem for elements parallel to the axis of symmetry (i.e. $D_t=0$).

5.1.2 The single-ring subroutine

1. The axial control point

The computation of the vortex/source ring influence functions does not allow the field point to be on the x-axis. This is not a worry as, in the PBD program, the field points are on the blades and never on the axis. Nevertheless, we can also compute analytically (and easily) the induced velocities using Biot-Savart law. Because of symmetry, the velocity will be only axial and, for a ring of radius $R$ and a field point at distance $x$ from the plane of the ring, the axial influence functions due to a vortex and a source ring have the following expressions:

\[
\tilde{u}_V = -\frac{\pi}{R(1 + \left(\frac{x}{R}\right)^2)^{3/2}},
\]

\[
\tilde{u}_S = \frac{\pi \left(\frac{x}{R}\right)}{R(1 + \left(\frac{x}{R}\right)^2)^{3/2}}.
\]

These expressions can then be included in the subroutine to prevent any difficulty in case this specific situation may occur.

2. The Zero ring

A lot of care needs to be taken in case the radius of the ring happens to be zero. Of course, it seems odd to consider a ring of zero radius. Yet, we are
computing the circumferential mean velocity, and the rings involved are not material rings. And when the element is crossing the x-axis, at this intersection point, the radius of the vortex ring computed in the whole integration process is zero. Once again, this is not supposed to happen in the use of PBD as the blades (and hence the discrete segments) never crosses the axis of symmetry. Still, a simple test helps to prevent any computation error. Indeed, the induced velocity in that case is only tangential because the element is in a plane that contains the axis, and there is nothing to worry for our subroutine which gives out the axial and radial components. This more general situation will happen when the expression $s = z_2y_1 - y_2z_1$ is zero (equivalent to say $O\bar{A}_1 \times O\bar{A}_2 = 0$). This condition will take care of any element in a plane containing the axis, and will avoid the computation in the specific case where the radius of the ring is zero.

3. The control point on the ring

A last source of worry in the ring subroutine is the numerical approximation of the elliptic functions involved (equations (3.11),(3.12)). The relevant parameter at this point is $k^2$ as defined in (3.13), and problems arise when $k$ reaches the neighborhood of its upper limit 1. In that case, the elliptic integrals show singularities as shown in [4] and, as a result, in the proximity of singular points, the axial and radial velocities show logarithmic and $\frac{1}{2}$-singularities, which will cause problems in the integration process.

Therefore, for field points on the ring, the numerical approximation may blow up, as the computer will not allow the computation (as for a zero denominator or a logarithm of a variable expression close to one). Though this situation is unlikely to happen in the PBD program (unless the element has locally a zero pitch -i.e. $\alpha = \pi/2$ in our system of notations-, or the panel distribution is so fine that control points, that are positioned in the middle of the panels, happen to be very close to the vortex segments), this singularity needs to be kept in
Figure 5-1: Relative error between CMVSEG and RING outputs for control points at Xp/R=0.01 from the generated surface

mind. Furthermore, for control points in the very close proximity of the element or of the generated surface (and hence the generating rings), the computation may show significant differences with the real results, due mainly to numerical approximations in the evaluation of the elliptic functions at stake.

This problem can be illustrated using the previous studied segment (figure 4-11). Instead of considering a control point moving in the surface plane x=0 (and which would give undefined velocities when the control point is on the surface), we can chose a control point very close to the plane x=0 and moving parallel to the surface. Figure 5-1 gives the relative errors between both the old and new subroutines for such a point moving in the plane x/R = 10^{-2}. With error values ranging from 10^{-4} to 10^{-3} (The peak at 10^{-2} is due to axial velocities close to zero, which is then not a worry, the error computation becoming irrelevant in that case), it proves that the new subroutine gives very accurate results.
Figure 5-2: Axial induced velocities for CMVSEG and RING subroutines at Xp/R=0.01 from the surface.

Figure 5-2 illustrates this accuracy for the axial vortex induced velocity.

If the control point is moved to the plane x/R = 10^{-3}, the relative errors become really significant (figure 5-3) and should not be ignored any longer. Nevertheless, figures 5-4 and 5-5 prove that the outputs from both subroutines are still very close, even though not exactly equal. Increasing the number of cycles in the Romberg method could be a good solution to get more accurate results. As shown in figure 5-6, the results are smoother with a higher number of cycles, but still different from CMVSEG outputs. This figure suggests also that such a high precision may not be very useful, all the more as the new code is meant to be implemented in a global program, that will call the code about 50,000 times. Those slight differences should not be statistically very significant as all the contributions will be added together in a PBD run. It is nevertheless interesting to state a criterion for these near-field situations.
Figure 5-3: Relative error between CMVSEG and RING outputs for control points at Xp/R=0.001 from the generated surface
Figure 5-4: Axial induced velocities for CMVSEG and RING subroutines at $X_p/R=0.001$ from the surface

Figure 5-5: Radial induced velocities for CMVSEG and RING subroutines at $X_p/R=0.001$ from the surface
Figure 5-6: Radial source induced velocity for different number of Romberg cycles in *RING* subroutine
Table 5.1: Divergence domain for the \textit{RING} subroutine

<table>
<thead>
<tr>
<th>Variables</th>
<th>x=0</th>
<th>r=1</th>
<th>\textit{Divergence Domain}</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>$1^\pm 10^{-3}$</td>
<td>1</td>
<td>$1^\pm 2 \times 10^{-3}$</td>
</tr>
<tr>
<td>x</td>
<td>0</td>
<td>$\pm 1.5 \times 10^{-3}$</td>
<td>$\pm 2 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

Table 5.2: \textit{Blowing} domain for the \textit{RING} subroutine

<table>
<thead>
<tr>
<th>Variables</th>
<th>x=0</th>
<th>r=1</th>
<th>\textit{Blowing Domain}</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>$1^\pm 6 \times 10^{-4}$</td>
<td>1</td>
<td>$1^\pm 6 \times 10^{-4}$</td>
</tr>
<tr>
<td>x</td>
<td>0</td>
<td>$\pm 5.5 \times 10^{-4}$</td>
<td>$\pm 5.5 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

\textbf{A divergence criterion}

To establish a near-field domain where the computation of the elliptic functions may diverge, the experiments should be made on a single ring. A ring of radius $R=r'$ can be approximated in the old \textit{CMVSEG} using a very small segment at distance $R$ from the axis. The characteristic value to consider is $k$ with:

$$k^2 = \frac{4r}{x^2 + (r + 1)^2}.$$  \hspace{1cm} (5.3)

$k$ is a function of $x$ (actually $x = (x_{ring} - x_{CP})/r'$) and $r$ ($r/r'$) and has its maximum at 1. The numerical computation of the elliptic functions blows up for $k=1$, which happens only when $x = 0$ and $r = 1$. A good way to test the \textit{RING} subroutine is to set $x$ (or $r$) at its critical value 0 (or 1) and make $r$ vary around 1 (or $x$ around 0). Figures 5-7, 5-8, 5-9 show the range of values of $x$ and $r$ where the divergence occurs. Table 5.1 gives the range of the domain of divergence, and table 5.2 gives more accurately the space domain (included in the divergence domain) where the \textit{RING} subroutine is unable to compute the elliptic functions.

The divergence domain gives the extreme values of the variables $x$ and $r$ to be used in the elliptic functions subroutine. If the control point happens to be so
close to the ring that \( x \) and \( r \) are within this divergence domain, then numerical divergence should be expected. A possible way to avoid this problem is to consider a control point very close to the ring (hence to the vortex distribution surface generated by the rotating element) to be on the ring (which is justified as the domain is very narrow). Küchemann gives then convenient ways to integrate logarithmic and \( \frac{1}{x} \)-singularities. Another way to solve the problem is to expand the expression of the ring induced velocities in series when \( k \) is close to one, and integrate it on a well-chosen surface as done in [2]. Nevertheless, this would imply to distinguish specific near-field cases. This extra computation time may not be needed as, in PBD, even for fine lattice grids, the control points are not that close to the panel segments.

Eventually, in the case of our algorithm, we can make a simple assumption. The influence of a ring to a close control point can be approximated as the influence of an straight infinite vortex line: A control point at a distance \( r \) really small from the ring will see the ring as an infinite line and the induced influence functions are going to be very large (in \( 1/r \)). For a control point on a ring, according to Biot-Savart law, the induced velocity will be axial. For an infinite vortex line, the induced velocity on the line is set to be zero. If we pursue the comparison between the ring and the infinite line in a near-field situation, we can make the assumption that the velocity for a point on a ring can be set to be finite as a rough approximation. In the case of our integration, if the control point happens to be exactly on one of the integration ring, the contribution of this ring to the global induced velocity at this point will be negligible compared to the influence of the other close rings, which will induced really large velocities (that can be computed with our algorithm as we are no longer in the divergence domain). Therefore, to avoid any problem in a case of a point inside the divergence domain, we can just set the induced velocity to be zero when \( k \) happens to reach its critical value (\( =1 \)). This test has been added to the algorithm to prevent any numerical error in the use of the new algorithm.
Figure 5-7: Accuracy comparisons between CMVSEG and RING subroutines for a single ring (Radially moving control point)

in the PBD program whenever this critical case may occur and proved to be a rather good assumption as shown in the next section where the algorithm has been included in the PBD code and tested in accuracy and running-time with several blade lattices under various loading and thickness conditions.
Figure 5-8: Accuracy comparisons between CMVSEG and RING vortex outputs for a single ring (Axially moving control point)
Figure 5-9: Accuracy comparisons between CMVSEG and RING source outputs for a single ring (Axially moving control point)
5.2 Tests inside the PBD environment

5.2.1 Modifications in the PBD program

The new fortran version of CMVSEG is listed in appendix A. It has been built according to the previously shown theory and have replaced the old version of CMVSEG inside the PBD subroutine called vorseg.f. The arguments have been made the same as for the previous CMVSEG, so that no further modifications need to be made inside the global PBD program, especially inside other calling subroutines.

As the new CMVSEG does not consider the tangential component of the induced velocities, this computation has to be done elsewhere. According to the results from chapter 3, the tangential circumferential mean velocity needs to be computed on behalf of the relative position of the control point and the horseshoe considered. This modification consists in adding the correct tangential velocity to outputs from the new CMVSEG according to the control point and horseshoe indices. A lot of care has had to be taken as, inside the PBD program, the velocity computation may differ between subroutines (some compute the opposite velocity - using for instance -Γ, as the circulation coefficient, instead of Γ). This modification has been made in the hscmv.f file and is listed in appendix B.

5.2.2 Optimizing tests

Once the accuracy of the new code checked up, the algorithm has been optimized within the PBD environment, as the aim of this work had practical goals and was made to compute circumferential mean velocities in real situations of propeller design with as much efficiency as possible. The parameters that can be changed are involved in the integration process. Namely, the tolerance allowed for the integration results and the number of cycles used for the integration can be optimized to enable a good compromise between accuracy and computation time.
Table 5.3: Thickness distribution for the tested propeller (Radius $R=D/2$)

<table>
<thead>
<tr>
<th>Location on blade ($=r/R$)</th>
<th>0.2</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thickness ($=t/D$)</td>
<td>0.02</td>
<td>0.018</td>
<td>0.015</td>
<td>0.01</td>
<td>0.006</td>
<td>0.003</td>
<td>0.002</td>
</tr>
</tbody>
</table>

A- Accuracy and tolerance

The first tests deal with the accuracy of the outputs using the new subroutine compared to the old accurate CMVSEG outputs. Figures 5-10 and 5-11 show comparisons for axial and radial global velocity outputs which are the relevant values in our case. This first test was made on a $10 \times 11$ grid on each blade, which is the most common used lattice grid, and was including the circulation and thickness effects. The number of propeller blades is set to 5, the loading coefficient, when used, is $C_{circ}=0.03$ and the thickness on the blade, when used, is set as described in table 5.3. The outputs measured include also the inflow and the effects of rotation, except when otherwise noticed. The control point number is decided inside the PBD program which describes the blade in a spanwise direction first and then moves to the next chordwise row (for instance, for a $10 \times 10$ grid, the first row of ten control points is at the root with control point #1 at the leading edge and control point #10 at the trailing edge. The last row is at the tip with the same chordwise description). The results are really close for both subroutines and for several tolerances. The maximum differences between both outputs according to the tolerance in the integration (figure 5-12) are even very low for a high tolerance of 0.5 and decrease as the tolerance decreases. As the computation is made in single precision, a tolerance of $5 \times 10^{-6}$ gives the most accurate results for the outputs. A comparison with a computation made at $5 \times 10^{-4}$ reveals that this last value of tolerance is sufficient to obtain accurate results, whereas if we take it to a higher value again the outputs begin to be slightly different from the most exact computation (figure 5-13).

Several tests have been made on PBD outputs, to compare the two algorithms in several situations, using predefined parameters: no loading and thickness only, no
thickness effects and only circulation, both effects coupled. The results are shown for different lattices in appendix C (tolerance=${5 \times 10^{-4}}$, Romberg cycles=10). The results prove to be very satisfactory under any set of conditions and type of blade lattice. Actually, an important test is the case of a grid of $20 \times 20$, which is the current finest grid that is being used in the propeller design program. A finer grid is a more critical case, as the control points are obviously closer to vortex/source grid segments, and therefore we may experience the *divergence* domain, where the *RING* subroutine will be less accurate on the elliptic functions computation. A diagnosis of this computation proved that this situation happens only at the edges of the blade where the lattice is really finer. Nevertheless, the assumption stated in the previous section, which consists in comparing the ring that may be critical to a vortex line and set this specific velocity to be zero, proved to be really efficient. It prevents from any blow up in the computation and, as shown in appendix C, the comparisons with the old *CMVSEG* outputs are really accurate. The values of the relative errors are of the same order of magnitude ($10^{-3}$, which is very satisfactory) in the divergence case and when the exact computation can be done (i.e. where the value of $k$ is not critical.).

The assumption done seems to be relevant. The algorithm proves to be really strong and accurate, even with a fine grid. We can also add that the fine grid is an extreme situation and that the most computations are done with simpler grid which proved not to reach such critical cases.

In the case of a $10 \times 10$ grid, table 5.4 lists the critical parameters $x$ and $r$ and the parameter $k$ as defined in section 5.1.2.3 for all control points where the Romberg integration does not converge after 11 cycles (tolerance=${5 \times 10^{-4}}$). None of the control point happens to be in the *divergence* domain. $k$ is never equal to 1 (and even for $k=0.999999$, the numerical computation of elliptic functions is defined). The divergence is there mainly due to values of $k$ being so close to 1 that the integration would require more cycles to fulfill the accuracy. But, as we will see in the next section, increasing the number of cycles will not be more useful when running PBD in single precision and the global outputs are, in any case, not significantly influenced by these *diverging* control points.
Table 5.4: Critical parameters for *diverging* control points

<table>
<thead>
<tr>
<th>r</th>
<th>x</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.874295</td>
<td>2.31744E-03</td>
<td>0.997748</td>
</tr>
<tr>
<td>0.878197</td>
<td>1.30062E-03</td>
<td>0.997895</td>
</tr>
<tr>
<td>0.893735</td>
<td>-2.51732E-03</td>
<td>0.998424</td>
</tr>
<tr>
<td>0.914469</td>
<td>2.00939E-02</td>
<td>0.998946</td>
</tr>
<tr>
<td>0.914556</td>
<td>-1.88755E-03</td>
<td>0.999003</td>
</tr>
<tr>
<td>0.935953</td>
<td>-2.52978E-03</td>
<td>0.999452</td>
</tr>
<tr>
<td>0.956069</td>
<td>2.00809E-02</td>
<td>0.999695</td>
</tr>
<tr>
<td>0.956092</td>
<td>-6.87567E-03</td>
<td>0.999742</td>
</tr>
<tr>
<td>0.971782</td>
<td>1.47027E-02</td>
<td>0.999870</td>
</tr>
<tr>
<td>0.971759</td>
<td>1.90133E-02</td>
<td>0.999851</td>
</tr>
<tr>
<td>0.971835</td>
<td>-6.44661E-03</td>
<td>0.999893</td>
</tr>
<tr>
<td>0.971832</td>
<td>7.03210E-04</td>
<td>0.999898</td>
</tr>
<tr>
<td>0.971825</td>
<td>-3.42347E-03</td>
<td>0.999896</td>
</tr>
<tr>
<td>0.984434</td>
<td>7.93264E-03</td>
<td>0.999961</td>
</tr>
<tr>
<td>0.984432</td>
<td>1.12367E-02</td>
<td>0.999953</td>
</tr>
<tr>
<td>0.984416</td>
<td>1.23511E-02</td>
<td>0.999950</td>
</tr>
<tr>
<td>0.984380</td>
<td>2.83733E-03</td>
<td>0.999968</td>
</tr>
<tr>
<td>0.984459</td>
<td>1.71390E-03</td>
<td>0.999969</td>
</tr>
<tr>
<td>0.984465</td>
<td>-1.37426E-03</td>
<td>0.999969</td>
</tr>
<tr>
<td>0.996822</td>
<td>7.31043E-04</td>
<td>0.999999</td>
</tr>
<tr>
<td>0.996835</td>
<td>3.73275E-03</td>
<td>0.999997</td>
</tr>
<tr>
<td>0.996777</td>
<td>-9.43484E-04</td>
<td>0.999999</td>
</tr>
<tr>
<td>0.996800</td>
<td>5.11717E-03</td>
<td>0.999995</td>
</tr>
<tr>
<td>0.996830</td>
<td>-3.13090E-03</td>
<td>0.999997</td>
</tr>
<tr>
<td>0.996845</td>
<td>-7.13567E-04</td>
<td>0.999999</td>
</tr>
</tbody>
</table>
Figure 5-10: Axial velocity in PBD vs. tolerance

Figure 5-11: Normal velocity in PBD vs. tolerance
Figure 5-12: Maximum differences between PBD output, using various tolerance in the integration

Figure 5-13: Tolerance choice in the integration process
B- Integration method

The Romberg method is well known for its accuracy to perform a numerical integration. It is nevertheless not the fastest integration method as the higher the number of cycles to reach the accuracy is, the slower is the process. For instance, in our case, a different method (such as a Gauss integration) could be considered in the case of far-field control points which require less accuracy in the velocity computation as their velocity contribution is really small compared to near-field points. For a 10 x 10 grid, a diagnosis of the Romberg integration has been made (table 5.5). It proves that most of the computation is done with a maximum of 2 or less cycles, (1 cycle corresponding to a trapezoidal rule computation and one Romberg cycle). Therefore, it does not seem very interesting to add a far-field/near-field test inside the subroutine in order to decide on the choice of the integration method, as most of the Romberg calls are already very fast. The addition of such a test would on the contrary certainly increase the running-time.

Figure 5-14 shows comparisons between several choice for the maximum number of cycles in the Romberg method. Although a choice of Ncycles=1 does not give accurate results, the order of magnitude is respected, which proves that a rough integration process gives results of the same order. As it seems not to be much time-consuming, keeping a Romberg scheme seems to be a good choice. Besides, there is no significant difference if we increase the number of cycles. Compared to the outputs from the previous CMVSEG, there is no significant gain in accuracy between $N_{cycles} = 11$ and $N_{cycles} = 15$. There is nevertheless a difference in the computation running-time as we will see in the next section.
Table 5.5: Number of cycles used in a run of PBD

<table>
<thead>
<tr>
<th>Ncycles used</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ncalls</td>
<td>24648</td>
<td>13446</td>
<td>2108</td>
<td>585</td>
<td>144</td>
<td>86</td>
<td>42</td>
<td>21</td>
</tr>
<tr>
<td>Ncycles used</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>Ncalls</td>
<td>9</td>
<td>15</td>
<td>11</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>15</td>
</tr>
</tbody>
</table>

Figure 5-14: Accuracy in PBD vs. Romberg cycles
C- Running-time

Computation time improvements are the main goal for our new code. The previous CMVSEG was very time-wasting when used inside PBD, which can call it up to 50,000 times in one run for a 10 × 10 grid. For instance, for a 10 × 10 grid, with loading and source effects, the PBD program requires up to 5 minutes and 30 seconds to solve the blade design problem. If a 20 × 20 grid is used, it can go up to 52 minutes. Now that the accuracy of our new routine is ensured, the running-time becomes a very important parameter.

First of all, in our study, the running-time of a program was very hard to evaluate in an absolute way. It depends actually of the load of the machine and of the net it is connected to. Nevertheless, it still gives relative orders of computation times that give a good idea of the actual running-time (all the more as the PBD running times, using the old CMVSEG, were measured in the same conditions). As a consequence, any given times are mean value of computation times of several runs of PBD in the same conditions.

The following tests were made on a 10 × 10 grid and were taking into account both loading and thickness effects. Figure 5-15 shows the PBD running time as a function of the number of Romberg cycles in the integration method. The time slightly increases for a number of cycles less than 11, the minimum time being of the order of 43s (only trapezoidal rule). Then, the increase is more significant. The order of time magnitude is 45 seconds, which proves the new subroutine to be much more efficient than the previous CMVSEG code. There is not a significant loss in time when using a maximum number of cycles of 10 or 11. On the other hand, increasing the number of cycles beyond these values wastes more computation time. Besides, increasing too much the number of cycles of integration would lead to subdivide the critical segment in so small subsegments that their contribution would be no longer significant, as the computation is made in single precision. Therefore, a safe choice of number of cycles would be between 8 and 11 cycles. It would ensure a correct accuracy as shown in the
Figure 5-15: Computation time vs. Romberg cycles

previous section and the running time would remain very satisfactory for this value. Figure 5-16 plots the running time as a function of the tolerance allowed inside the integration method. With same order of time magnitude, a tolerance up to $5 \times 10^{-5}$ does not influence much the code running time. For higher accuracy, the lost in computation time is more significant. Nevertheless, it has been seen in the Accuracy and tolerance section that the differences in accuracy between a tolerance of $5 \times 10^{-4}$ and $5 \times 10^{-6}$ are not significant. A good choice, then, is a tolerance of $5 \times 10^{-4}$ as the computation time and the accuracy are really satisfactory.

Final experiments have been made on $10 \times 10$ and $20 \times 20$ grids, with loading and thickness effects, a tolerance of $5 \times 10^{-4}$ and a maximum number of Romberg cycles of 10. The accuracy results are shown in Appendix C. The running times are listed in table 5.6 as well as the time improvement factor which is of the order of 7.
Figure 5-16: Computation time vs. tolerance

Table 5.6: PBD mean running times for both CMVSEG subroutines

<table>
<thead>
<tr>
<th>Grid</th>
<th>Old CMVSEG(seconds)</th>
<th>new CMVSEG(seconds)</th>
<th>Improvement Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 × 10</td>
<td>330</td>
<td>48</td>
<td>6.88</td>
</tr>
<tr>
<td>20 × 20</td>
<td>3080</td>
<td>447</td>
<td>6.9</td>
</tr>
</tbody>
</table>
Chapter 6

Conclusion

The computation of the circumferential mean velocity induced by a propeller is a major step in the design of propulsor devices. It is the fundamental link between the hub and the hull flows, which form the basic principle of the study of the flow around the propeller and of the design of its blades. The subroutine that computes the circumferential mean velocity for any segment of the discretized blade grid is therefore a key issue in the PBD program.

Despite a very high accuracy, the CMVSEG code used until now was much time-consuming. This inconvenient aspect needed obviously to be improved. This implied a whole new mathematical approach, using the axisymmetrical characteristics of the circumferential mean velocity computation. Instead of integrating numerically on the rotation of the element as done in the previous version of CMVSEG, the problem has been decomposed into vortex and source rings, to take into account both the loading and the thickness effects in the induced influence functions. The axial and radial velocities induced by rings can be analytically computed, and the final integration is being made on the element. Ultimately, the tangential component of the velocity, which is much easier to compute with our horseshoe panel method, is added to give the total circumferential mean induced velocity for every lattice segment (and hence for the whole propeller by adding all these contributions). This new algorithm proves to keep up with the accuracy required for the PBD outputs and shows a meaningful
save in computation time (The PBD code runs about 7 times faster with the new algorithm for CMVSEG) which is a major improvement in the design of propellers.

This algorithm was built to be implemented inside the global PBD program. Therefore, it has been adapted to the environment and the requirements of the Propeller Blade Design program. Naturally, it can be adopted for other applications. In this purpose, a complete diagnosis of every possible bugs that may occur in the use of the new CMVSEG has been made to strengthen the robustness of the algorithm. Most of these situations can be easily solved. Still, in the case of control points on the considered rings, the computation of the elliptic functions cannot be fulfilled. This case does not happen in the use of the PBD program (or only a few times among a huge number of calls for really fine blade grids), and proved to be easily solved by comparing this case with an infinite vortex/source line. Nevertheless, if wanted, a more specific near field computation could be added to perform an analytical integration in the case of these critical control points, using series expansions as shown in [2]. This near field computation would require several steps: state the relative positions of control points and rings, define an elemental domain around the control point to expand the elliptic functions, conduct the analytical integration with as much accuracy as possible. It would certainly imply extra computation time, but may be necessary for further applications. These specific isolated cases are nevertheless not much significant inside the PBD program, all the more as they happen for the extreme cases of fine grids and only at the edges of the blade where the positions of the control points (and therefore the induced velocities computation) is still being studied as they may not be as relevant as for other control points.

The last recommendation deals with the general programming style and the robustness of the algorithms that are programmed or improved. During this work, several programs showed a significant lack in internal comments and little care in critical situations that may occur during the computation. This was the case, for instance, for the VORSEG subroutine where the use of double precision in the near
field has been removed without any notice. For any modification in a subroutine or whenever an error may happen in the computation (like a division by zero, even for obvious cases), adding internal comments and error warnings is a good attitude for any programmer and may save many comprehension efforts for future users. It is another simple way to make science even more efficient.
Appendix A

RING/new CMVSEG subroutine code

SUBROUTINE CMVSEG(XP,YP,ZP,X1,Y1,Z1,X2,Y2,Z2,VX,VY,VZ,*
     SX,SY,SZ,LSIG,NBLADE)
C------------------------------------------------------
C Programmer: Frederic BUCHOUX
C Date: 06/02/95
C This subroutine computes the circumferential mean
C velocities induced by a Vortex/Source Segment
C A1(X1,Y1,Z1)-A2(X2,Y2,Z2) at a control point P(XP,YP,ZP)
C =======================================================
C It works by integrating axial and radial rings contributions
C along the element:
C Romberg Integration from 0 To D (length of segment) of
C VORTEX:
C [Vvortex-ring(M)/(2*PI*RM)]*[SM/(RM*RA)]*[RA/D]dl
C..............Vring/(2*PI*RM)=Influence function of a "point" on a ring
C..............SM/(RM*RA)=Sin(Radial pitch angle)
C..............RA/D=Sin(Axial pitch angle)
C The outputs are in the global coordinate system
C CMVSEG does not compute the tangential velocity
C ------------------------------------------------------
LSIG = Test if thickness or not

SOURCE:

\[ V_{\text{source}} \cdot \text{ring}(M)/(2*\pi*RM)]\text{d}l \]

The outputs are in the global coordinate system

No tangential velocity in the case of the source influence functions

LOGICAL CONVG, LSIG

PARAMETER(ZERO=0.0E00, HALF=0.5E00, ONE=1.0E00, TWO=2.0E00)

PARAMETER(PI=3.1415927E00, FOUR=4.0E00)

Set the tolerance

PARAMETER(TOL=5E-4)

DIMENSION A(12,12,4), V(4), SUM(4)

VX=ZERO
VY=ZERO
VZ=ZERO
UR=ZERO
UT=ZERO
SX=ZERO
SY=ZERO
SZ=ZERO
SR=ZERO
ST=ZERO

Compute the useful values

AX=X2−X1
AY=Y2−Y1
AZ=Z2−Z1

Ri= Radius of point i in a plane X=cst
RA= Length of the element projected in plane X=cst
D=Length of element
R1=SQRT(Y1**2+Z1**2)
R2=SQRT(Y2**2+Z2**2)
RP=SQRT(YP**2+ZP**2)
TP=ATAN2(ZP,YP)
RA=SQRT(AY**2+AZ**2)
D=SQRT(AX**2+AY**2+AZ**2)
SSCALE=FLOAT(NBLADE)/(TWO*PI)
VSSCALE=SSCALE/D
C----NOTE: Non valid if D=0 (length of element =0!!!)
C
C Check if thickness needs to be taken into account
C
NVC=2
IF (LSIG) NVC=4
C
C Set the lower and upper limits of integration
C {Integration on the length of the element}
C
DL=0
DR=D
C
C Zero the element of the Romberg matrix
C
DO 100 J=1,NVC
  DO 110 N=1,12
    DO 120 M=1,12
      A(N,M,J)=ZERO
  120 CONTINUE
  110 CONTINUE
  100 CONTINUE
C
C Test if Radial Segment crosses the X-axis
C (Incompatible with Subroutine RING)
C
S=Z2*Y1-Y2*Z1
IF (S.EQ.0) GOTO 300
C---------------------------------------------------------------
C  First iteration= trapezoidal rule
C---------------------------------------------------------------

H=DR-DL
BA=HALF*H
CALL RING(X1,R1,S,XP,RP,V(1),V(2),V(3),V(4))
DO 150 J=1,NVC
   A(1,1,J)=BA*V(J)
150 CONTINUE
CALL RING(X2,R2,S,XP,RP,V(1),V(2),V(3),V(4))
DO 160 J=1,NVC
   A(1,1,J)=A(1,1,J)+BA*V(J)
160 CONTINUE

C---------------------------------------------------------------
C  Romberg Cycles
C---------------------------------------------------------------

MK=1
DO 200 NK=1,10
   NKK=NK
   DO 210 J=1,NVC
      SUM(J)=ZERO
210 CONTINUE
   DM=DL+BA
   DO 220 M=1,MK
      XM=X1+DM/D*(X2-X1)
      YM=Y1+DM/D*(Y2-Y1)
      ZM=Z1+DM/D*(Z2-Z1)
      RM=SQRT(YM**2+ZM**2)
      SM=YM*AZ-ZM*AY
      CALL RING(XM,RM,SM,XP,RP,V(1),V(2),V(3),V(4))
      DO 225 J=1,NVC
         SUM(J)=SUM(J)+V(J)
225 CONTINUE
220 CONTINUE
225 CONTINUE
DM=DM+H
220 CONTINUE
DO 230 J=1,NVC
A(NK+1,1,J)=HALF*A(NK,1,J)+SUM(J)*BA
230 CONTINUE
H=BA
BA=HALF*BA
F=ONE
DO 240 IZ=1,NK
NZ=NK−IZ+1
F=F*FOUR
DO 245 J=1,NVC
A(NZ,IZ+1,J)=(F*A(NZ+1,IZ,J)−A(NZ,IZ,J))/(F−ONE)
245 CONTINUE
240 CONTINUE
CONVG=.TRUE.
C........Test the convergence defined by TOL=tolerance allowed
DO 250 J=1,NVC
IF (ABS(A(1,NK+1,J)−A(1,NK,J)).GT.TOL) CONVG=.FALSE.
250 CONTINUE
IF(CONVG) GO TO 300
MK=MK+MK
200 CONTINUE
300 CONTINUE
C-----------------------------------------------
C Outputs = Axial and Radial velocities
C-----------------------------------------------
VX=VX+A(1,NKK+1,1)*VSCALE
UR=UR+A(1,NKK+1,2)*VSCALE
SX=SX+A(1,NKK+1,3)*SSCALE
SR=SR+A(1,NKK+1,4)*SSCALE
C-----------------------------------------------
C Y− And Z−Cartesian Velocities(NOT INCLUDING THE TANGENTIAL COMPONENT)
C-----------------------------------------------
VY=VY+UR*COS(TP)
VZ=VZ+UR*SIN(TP)
SY=SY+SR*COS(TP)
SZ=SZ+SR*SIN(TP)
END

SUBROUTINE RING( XV, RV, SV, XF, RF, UX, UR , SX, SR )

C
C This subroutine computes the velocities induced by a vortex/source ring
C located at XV with radius RV with unit circulation
C divided by RV*RV(vortex)
C or RV(source) at a control point (XP,RP).
C The algorithm is from the solution given
C in terms of elliptic functions by Kucheman and Weber's
C "Aerodynamics of Propulsion."
C The elliptic functions are evaluated with the
C routine QCALC.
C
C RV is the radius of the vortex ring.
C XF is the distance to the field point.
C SV is the contribution of the radial angle to the vortex influence function
C RF is the radius of the field point.
C UX is the axial induced velocity at the field point.
C UR is the radial induced velocity at the field point.
C
PARAMETER(ZERO=0.0E00, ONE=1.0E00, TWO=2.0E00)
PARAMETER(PI=3.1415927E00)
R=RF/RV
X=(XV-XF)/RV
RK=TWO*SQRT(R/(X*X+R*R+2.0*R+1.0))

C----------------------
C Test if Control Point on the axis (==>RF=R=RK=Zero)
C----------------------
IF (RF.EQ.ZERO) THEN
  UX=PI/(SQRT((1+X**2)**3))
UR=ZERO
SX=-(PI*X)/(SQRT((X**2+1)**3))
SR=ZERO
GOTO 1000
ENDIF

C---------------------
C Test if control point on the ring (RK=1)
C In that case the elliptic functions cannot be computed
C Set the velocities to be a finite value (=zero)
C But does not influence the global integration
C as velocities due to other close rings induced larger contributions.
C---------------------
IF (RK.EQ.1) THEN
UX=ZERO
UR=ZERO
SX=ZERO
SR=ZERO
RETURN
ENDIF

C--------------------
C IF Control Point not on X-axis
C--------------------
Z=TWO/(RK*RK)-ONE
CALL QCALC(Z,ELE,ELK)
F=RK*RK/(TWO*(ONE-RK*RK))
UX=(RK/(TWO*SQRT(R)))*(ELK-ELE*(ONE+F*(ONE-ONE/R)))
UR=(X*RK/(TWO*SQRT(R**3)))*(ELK-ELE*(ONE+F))
SX=-(X*RK*F*ELE)/(2*R*SQRT(R))
SR=(RK/(2*R*SQRT(R)))*(ELK-ELE*(1-(R-1)*F))
C----Change RV->RV**3(VORTEX) or RV**2(SOURCE)
C----([dividing by RV only] gives the influence functions
C----of a vortex/source ring)
1000 UX=(UX*SV)/(RV**3)
UR=(UR*SV)/(RV**3)
SX=SX/(RV**2)
SR=SR/(RV**2)
SUBROUTINE QCALC(Z,ELE,ELK)

C  ELLIPTIC FUNCTIONS ROUTINE

C  INPUT
C  Z  -  ARGUMENT  [ UNDEFINED FOR Z= +/-1 ]

C  OUTPUT
C  ELE, ELK  -  ELLIPTIC FUNCTIONS

REAL KPRMES
PARAMETER(ONE=1.0DO0, TWO=2.0DO0)
KPRMES=ONE-(TWO/(Z+ONE))
A1=KPRMES
A2=A1*A1
A3=A2*A1
A4=A2*A2
ALO=LOG(ONE/KPRMES)
ELE=1.0000000000+.44325141463*A1+.06260601220*A2
  +.04757383546*A3+.01736506451*A4+
  (.24998368310*A1+.09200180037*A2+
  .04069697526*A3+.00526449639*A4)*ALO
ELK=1.38629436112+.09666344259*A1+.03590092383*A2
  +.03742563713*A3+.01451196212*A4+
  (.5000000000+.12498593597*A1+.06880248576*A2
  .03328355346*A3+.00441787012*A4)*ALO

RETURN
END
Appendix B

 Modifications in the PBD hscmv.f file

[...: to beginning of hscmv.f]

C.Modifl....Calculation of tangential velocities from horseshoes
C..........Test the horseshoes(M2,N2) and control points(M,N) indices
C..........to evaluate their relative positions
     IF ((M.EQ.M2).AND.(N.GE.N2)) THEN
C..........If field point inside horseshoe, set the value vor V't
     VT=-FLOAT(NBLADE)/RP
     ELSE
C..........If field point outside horseshoe, set V't=zero
     VT=0.
     END IF
C...end Modifl...........

C..........Add up CMV velocities if .CMV output file is needed
C..........Remember that HSCMV makes
C..............NEGATIVE Horseshoe Influence Function's!!
     IF ((IMODE.EQ.3).OR.(IMODE.EQ.5).OR.(IMODE.EQ.6)) THEN
         VVCMV(1,N,M)=VVCMV(1,N,M)-GSP(N2,M2)*HDUM1
     END IF
VVCNMV(2,N,M) = VVCNMV(2,N,M) - GSP(N2,M2) * HDUM2
VVCNMV(3,N,M) = VVCNMV(3,N,M) - GSP(N2,M2) * HDUM3

C...Modif2...........Add Tangential Velocity contribution to CMVSEG outputs
VVCNMV(2,N,M) = VVCNMV(2,N,M) - (SINTP*VT*GSP(N2,M2))
VVCNMV(3,N,M) = VVCNMV(3,N,M) + (COSTP*VT*GSP(N2,M2))

C...end Modif2.............

END IF

C............Subtract CMV from LBV HIF’s
HIF(1,J,I) = HIF(1,J,I) + HDUM1
HIF(2,J,I) = HIF(2,J,I) + HDUM2
HIF(3,J,I) = HIF(3,J,I) + HDUM3

C..Modif3......Add Tangential Velocity contribution
to horseshoe influence functions when needed
C...............Remember that HSCMV makes negative HIF’s!
HIF(2,J,I) = HIF(2,J,I) + (SINTP*VT)
HIF(3,J,I) = HIF(3,J,I) - (COSTP*VT)

C...end Modif3...........

[...-> to end of hscmv.f..]
Appendix C

PBD global velocity outputs comparison tests

C.1 Grid 10x10

The results with inflow take into account the inflow (in the x-direction) and the flow motion due to the rotation of the propeller (y- and z-directions). The results without inflow give the outputs for the velocity induced only by the singularities (vortices, sources, or both). The normal velocity is the relevant output in PBD: It is the velocity normal to the blade, which is required for the kinematic boundary condition.
Figure C-1: Relative errors old CMVSEG/RING with loading and thickness (LT)
Figure C-2: Normal velocity (LT with inflow)

Figure C-3: Normal velocity (LT without inflow)
Figure C-4: x-velocity (LT with inflow)

Figure C-5: x-velocity (LT without inflow)
Grid 10x10, Tol=5E-4, Ncycles=10, Loading and thickness effects

Figure C-6: y-velocity (LT with inflow)

Grid 10x10, tol=5E-4, Ncycles=10, Loading and thickness effects

Figure C-7: y-velocity (LT without inflow)
Figure C-8: $z$-velocity (LT with inflow)

Figure C-9: $z$-velocity (LT without inflow)
Figure C-10: Relative errors of old CMVSEG/RING with loading only (L)

Figures showing relative errors for different parameters against control point number.
Figure C-11: Normal velocity (L with inflow)

Figure C-12: Normal velocity (L without inflow)
**Figure C-13: x-velocity (L with inflow)**

**Figure C-14: x-velocity (L without inflow)**
Figure C-15: y-velocity (L with inflow)

Figure C-16: y-velocity (L without inflow)
Figure C-17: z-velocity (L with inflow)

Figure C-18: z-velocity (L without inflow)
Figure C-19: Relative errors old CMVSEG/RING with thickness only (T)
Figure C-20: Normal velocity (T with inflow)

Figure C-21: Normal velocity (T without inflow)
Figure C-22: x-velocity (T with inflow)

Figure C-23: x-velocity (T without inflow)
Figure C-24: y-velocity (T with inflow)

Figure C-25: y-velocity (T without inflow)
Grid 10x10, Tol=5E-4, Ncycles=10, Thickness only

0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
-0.3
-0.35
0 20 40 60 80 100 120
Control Point Number

Figure C-26: z-velocity (T with inflow)

Grid 10x10, tol=5E-4, Ncycles=10, Thickness effects only

0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
-0.3
-0.35
0 20 40 60 80 100 120
Control Point Number

Figure C-27: z-velocity (T without inflow)
C.2 Grid 20x20
Figure C-28: Relative errors old CMVSEG/RING with loading and thickness for lattice 20x20 (LT2)
Figure C-29: Normal velocity (LT2 with inflow)

Figure C-30: Normal velocity (LT2 without inflow)
Figure C-31: x-velocity (LT2 with inflow)

Figure C-32: x-velocity (LT2 without inflow)
Figure C-33: y-velocity (LT2 with inflow)

Figure C-34: y-velocity (LT2 without inflow)
Grid 20x20, Tol=5E-4, Ncycles=10, Thickness and loading

Figure C-35: z-velocity (LT2 with inflow)

Grid 20x20, tol=5E-4, Ncycles=10, Loading and thickness effects

Figure C-36: z-velocity (LT2 without inflow)
Bibliography


