# A Full Potential Solver for Lifting Flows on Unstructured Tetrahedral Meshes

Robert Mihael Liegl

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montréal, Québec, Canada

December 2005

*Your file Votre référence*
*ISBN: 0-494-14310-X*
*Our file Notre référence*
*ISBN: 0-494-14310-X*

# ABSTRACT

A Full Potential Solver for Lifting Flows on Unstructured Tetrahedral Meshes

Robert Mihael Liegl

The full potential model, though having known its major development through the 1970's into the 1980's, is currently used in a wide range of applications and it continues to be a valuable CFD tool proving certain advantages. The current approach, using an unstructured tetrahedral mesh and finite volume method, tries to take advantage of integrating a fast initializer (a linear potential Prandtl-Glauert corrected panel model) whose results are computationally very affordable while giving a good start for the actual full potential solver in terms of lift. As a result, the implementation of the Kutta condition will be of significantly lower computational cost, whereas it is reportedly known for damaging convergence. Also, the outer boundary of the computational domain can be set closer to the studied lifting body, as it will be aware of the lifting flow inside. Hence, a smaller computational domain can be used. Another full potential issue addressed here is the slow convergence of transonic solutions and a way of circumventing this problem has been proposed. Overall, the present full potential model implementation is intended as an accurate subsonic tool, with the capability to converge transonic solutions, while being at least one order of magnitude faster than an Euler solution on a similar mesh.

# ACKNOWLEDGEMENTS

A grateful thought goes towards my late Professor Dr. Jaroslav Svoboda. He has set an example. May God reward him in the name of all those regreting him.

Not less, I wish to thank my Professor and thesis supervisor, Dr. Marius Paraschivoiu, for being the kind of teacher that any student would like to have. He has provided me with constant and consistent guidance without restraining my research freedom and initiative. His support was a blend of teaching, motivating and encouraging.

I am also grateful for all the friends I found in Concordia University while being a graduate student here.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF ABBREVIATIONS AND SYMBOLS

| | |
|---|---|
| $\phi$ | velocity potential |
| $\vec{V}$ | flow velocity |
| $\rho$ | density |
| $\gamma$ | specific heats ratio |
| M | Mach number |
| $V_\infty$ | unperturbed flow velocity |
| $\Gamma$ | airfoil or wing bound circulation, vortex intensity |
| $\alpha$ | angle of attack |
| $\beta$ | sideslip angle |
| $\vec{\mathbf{n}}$ | body surface normal |
| $r$ | radius, in polar coordinates |
| $\theta$ | angle, in polar coordinates |
| $\gamma_{frontal}, \gamma_{lateral}$ | wing bound vorticity |
| $D$ | number of panels on a wing model |
| $x,y,z$ | Cartesian coordinates |
| $u,v,w$ | Cartesian components of velocity vector |
| $\mu$ | depending on context: |
| | doublet intensity or |
| | upwind switching function |
| $\vec{d}$ | depending on context, a position vector |
| $G$ | flux function |
| $\tau_i^c, \tau_{i,j}$ | node $i$ attached volume, tetrahedra $j$ attached to node $i$ |
| $\mathbf{R}$ | array of mass flux residual |
| $R_i$ | local mass flux residual |
| $J$ | Jacobian |
| $\delta\Phi$ | array of potential updates |

$\Phi$ array of potential values

$\epsilon$ small perturbation

$R^{\Gamma}$ recirculation residual

$c_i^{lower}$, $c_i^{upper}$ mesh dependent local trailing edge coefficients relating the expected value in the wake face center to the gradient values in the tetrahedra pairs

$c^n$ explicit circulation update coefficient

$c_{i^-}^{R^\Gamma}$ and $c_{i^+}^{R^\Gamma}$ mesh dependent local trailing edge coefficients relating the necessary update in trailing edge node potential to the neighboring recirculation residuals

FP full potential model; in context, denoting also the current implementation

FG refering to the fictitious gas approach

*i,j,k,m,n* (subscripts) indices of vortex segments or collocation points in panel model, or nodes, tetrahedra in full potential model

$n$ (superscript) current Newton step

$w_{indexed}$, $t_{indexed}$ referring to wake nodes or trailing edge nodes

# Chapter 1

# Introduction

## 1.1 The Role of Potential Flow in CFD

Although Navier-Stokes solutions are becoming more common today, simpler solutions under the assumption of a potential flow field, whether in the form of linear potential or full potential models, are still valued for their simplicity and low computational cost, especially for flow simulations where the viscous effects might be neglected in a first analysis or where such effects might be caught with a coupled boundary layer solver.

For an exhaustive review of the development of potential flow models, one can follow references [3] ... [14], [17], [27], [28] for full potential and [29] for linear potential / panel methods. To date we are not aware of a state-of-the-art beyond what has been presented in the above cited references. Also, a short history of transonic CFD from the perspective of *Symposium Transonicum* could be followed in reference [31].

In the following, the generally accepted potential flow paradigm will be presented and used as a starting point, with the specific features of the current approach being outlined.

Linear models, in their basic form do not take into account fluid compressibility and hence are appropriate for use in low speed flow simulations. By using a compressibility correction such as Prandtl-Glauert [29], the validity of such models can be approximately extended within the compressible subsonic range, however, incurring some error. Moreover, the presence of shocks in a real flow and the inability of such models to catch shocks makes them inappropriate for simulating transonic flows. Despite some interesting attempts made to enhance the capability of panel models to simulate high subsonic, and even transonic flows (like [12], [8]), such models have not been used widely. However, the linear potential panel models are still valuable for providing accurate and very fast solutions in low speed aerodynamics and related interdisciplinary applications.

There are various implementations of the linear potential model, but all stem from the basic assumption of a velocity potential with a constant fluid density. The body exposed to the flow will alter the unperturbed flow potential with a perturbation potential, usually modeled with singularities as sources, doublets, vortices or a combination of them. The advantage of such an approach lies in the fact that there is no need to compute the flow throughout the domain but it suffices to calculate the intensities of these singularity elements. The solution reduces to solving a linear system of equations. For lifting flows, a slight complication of the problem is introduced by the wake relaxation, which actually introduces a non-linearity. For many purposes, just assuming a shape for the wake which is realistic enough would be sufficient. For more critical applications, the model might be run several times, with wake relaxation steps in between. Usually, even in such cases a limited number of repetitions will be sufficient.

The full potential model, on the other side, is not prone to the limitations of linear potential assumptions. Introducing the fluid density as a non-linear dependent variable, in addition to the velocity potential, makes compressible flow modeling possible both in subsonic and transonic conditions, as long as the shocks are of low intensity. Actually, for subsonic flows, a full potential solution should be basically equivalent to an Euler solution, while being computationally more affordable. This qualifies the full potential model as the best tool for fast simulations in high subsonic inviscid aerodynamics, for situations where neglecting viscous effects is acceptable.

The potential flow theory seems to have reached its maturity more than 15 years ago and seemingly there is little room for further improvements. Though currently used as engine for a wide range of applications, including standalone aircraft design and optimization tools or integrated as initializer for Navier-Stokes solvers, there was little development in the theory behind potential flow after the 80's. However, as the meshing techniques and CAD modeling tools have evolved, there is a new opportunity for updating the implementations of this model and looking for new applications which, due to the rigid formulation imposed by structured meshes, were impossible or at least impractical before. An interesting approach for generating 3D grids can be seen in the development of a non-linear potential solver emerged at Boeing into what is known as TRANAIR [32], [36].

## 1.2 Overview of the Potential Flow Model

### 1.2.1 General Considerations

A technical model of some real phenomenon should be, within the specifications of accuracy imposed, as simple as possible. The potential flow theory, while trying to model an inviscid, irrotational flow, brings a substantial simplification of the mathematical description by compressing the vectorial velocity into one scalar velocity potential function, hereby noted as $\phi$.

$$\vec{V}=\vec{\nabla}\,\phi \tag{1.1}$$

Therefore, instead of solving for three scalar components of the velocity, the only unknown will be the potential value, which will implicitly include all other parameters of the flow (pressure, density, energy, Mach number). For many applications, the potential flow assumption will realistically stand for the most part of the flow domain, but there are two issues which need closer attention.

The irrotationality of a potential flow will disable the model to catch the lift production by an airfoil (or a tridimensional wing). The second issue is the inability to catch the increase in entropy (and the related vorticity production) in a shock. In many implementations the isentropic shock and the related error are acceptable, though different corrections for this problem have been proposed and can be followed in [19], [28]. The first problem, however, cannot be overlooked and an implementation of a Kutta condition is mandatory.

### 1.2.2 The full potential model

**Equations** For a full potential approach, the compressibility of the fluid will be taken into account, which actually introduces a nonlinearity in the model. The continuity (mass conservation) equation will be:

$$\frac{\partial\rho(\phi)}{\partial t}+\nabla(\rho\nabla\phi)=0 \tag{1.2}$$

The density will be written as:

$$\rho(\phi) = \rho_\infty(1 + \frac{\gamma - 1}{2} M_\infty^2 (1 - \frac{\|\nabla\phi\|_2^2}{V_\infty}))^{1/(\gamma-1)} \tag{1.3}$$

These considerations stand for an isentropic flow, where the momentum and energy equations will be equally satisfied, along with the mass conservation, which is the only conservation equation explicitly formulated in the potential model.
For steady flows, the potential equation (1.2) is further simplified to:

$$\nabla(\rho\nabla\phi) = 0 \tag{1.4}$$

**Boundary conditions** On solid boundaries, body walls, the impermeability condition should be imposed as

$$\frac{\partial\phi}{\partial\mathbf{n}} = 0 \tag{1.5}$$

where **n** is the wall's normal direction.

For a non-lifting flow, the outer boundary will not be significantly perturbed and usually a uniform free flow will be imposed. The farfield will have potential values to match the unperturbed flow, at given Mach number $M_\infty$, pressure $p_\infty$, density $\rho_\infty$.

$$\phi_{boundary} = \vec{V_\infty} \cdot \vec{d} + \phi_0 \tag{1.6}$$

In the case of a lifting flow, the outer boundary should be influenced, both for a 2D flow (airfoil) and 3D flow, with the wake generated intersecting the outer boundary. The actual implementation of a Kutta condition should take into account the outer boundary influence.

**Lifting flows and the Kutta condition** The computational domain around a lifting body can no longer be simply-connected, or it will not be able to catch the

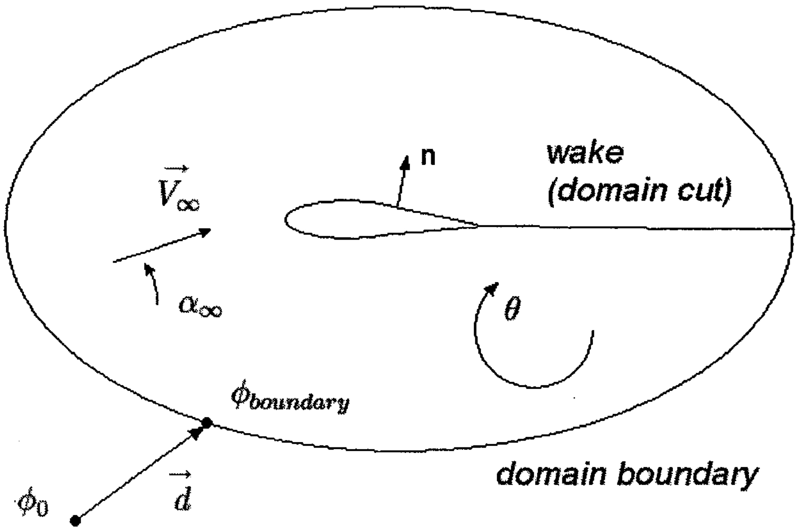

Figure 1.1: Boundary conditions

circulation production. The trailing edge of an airfoil or 3D wing will start a cut through the domain which will meet the outer boundary.

In a 2D flow the cut has only mathematical meaning, being used to impose a jump in potential value which will be equal to the circulation generated by the airfoil. Therefore, the actual shape and position of the cut will have no effect on the solution accuracy. The whole outer boundary will be influenced, but this influence is straightforward to determine. It suffices to add a frontal vortex (or a distribution of frontal vortices) modeling the airfoil. For a single vortex, placed at 1/4 of the airfoil chord, the effect, corrected for compressibility will be, following [27], [28]:

$$\phi_{boundary} = \vec{V_\infty} \cdot \vec{d} + \phi_0 + \frac{\Gamma}{2\pi} \tan^{-1}[\sqrt{1 - M_\infty^2} \tan(\theta - \alpha_\infty)] \tag{1.7}$$

where $\theta$ defines the position of a boundary point, $V_\infty$ and $\alpha_\infty$ define the free flow.

The rotational potential induced by this vortex (or system of vortices) will increase in value from the lower side of the cut, rotating through the domain, up to its maximal value on the upper side of the cut. Through the cut, the rotational potential value is discontinued back to its lowest value.

In 3D flows things are significantly different, the cut having also a physical meaning, i.e. the wake shed in the flow by the trailing edge of a wing. Consequently, some attention will be recommended when modeling the wake. Numerical tests performed both with panel and full potential models suggest that the shape and location of the far wake is of negligible importance while the wake close to the trailing edge influences sensibly the wing-bound circulation. As a rule of thumb, for subsonic (isentropic) flows, a wake leaving the trailing edge at bisector angle and with mild or no variation in the vicinity of the trailing edge will produce results consistent with the inviscid flow theory. However, real flows might behave slightly different both in terms of wake orientation and lift production (the two are related), because of viscous effects. For completion of this considerations, Chapter 3 will readdress the issue in the context of the simulation used for validation.

In terms of lift influence on the outer boundary, both the 3D wing attached circulation and the wake distributed vorticity should theoretically be taken into account. Though the wing influence is sometimes neglected, this will introduce an error, increasing for smaller domains. The wake will intersect the outer boundary and there is no way one could neglect its influence on this boundary. In the implementation chapter, specific details to the current approach will be given on how the outer boundary takes into account both the presence of the wing and its wake.

### 1.2.3 The linear potential simplification

The linear potential model brings about a supplementary simplification consisting in the incompressibility assumption, which will allow the formulation of a linear problem. The panel methods are usually build upon this assumption, though they might eventually allow the simulation of a compressible flow by means of corrections as the Prandtl-Glauert correction. In the incompressibility assumption, equation 1.1 with a constant density will yield the continuity equation as:

$$\nabla^2\phi = 0 \tag{1.8}$$

The presence of a body will be implemented by a distribution of elements which will perturb the potential field. Usual elementary field generators are doublets, sources, vortices. The boundary condition on the body's skin will be:

$$\nabla\phi \cdot \vec{\mathbf{n}} = 0 \tag{1.9}$$

where $\vec{\mathbf{n}}$ is the surface normal.

The equation 1.9 will be imposed either by specifying appropriate values of the potential close to the solid surface, which is a Dirichlet formulation, or by specifying the potential derivative normal to the surface to be zero, known as Neumann formulation. The later is more intuitive from physical point of view.

In practice, the distribution of field generators will be neither uniform, nor continuous. Actually the distribution will be a set of discrete elements, allowing to reduce the problem to a linear system of equations. Hence, the Neumann condition will be imposed in a (balanced) set of collocation points on the solid border, ensuring the impermeability to the flow in these points while providing the system of equations. Depending on the formulation, at least part of the equations will stem from the wall condition written in the collocation points. In some formulations,

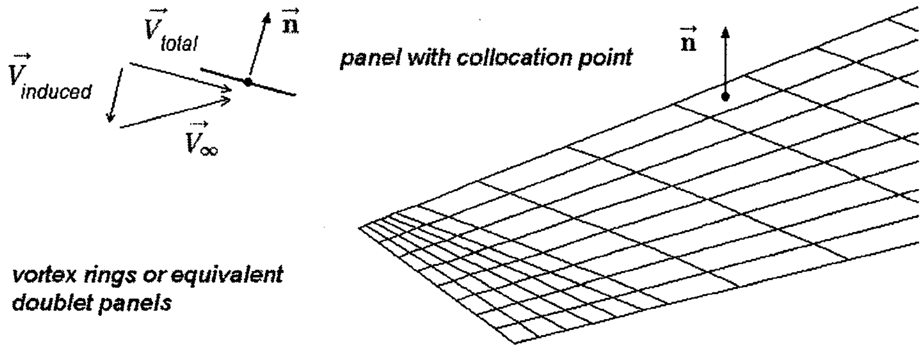

Figure 1.2: Example of a linear potential model for a lifting surface

additional equations might come from summation of vortex segments in panel mesh nodes.

When modeling a wing or body with thickness, the Dirichlet boundary condition for an impermeable solid surface with no internal singularities will reduce to having a constant potential inside the body. A combination of the Neumann and Dirichlet conditions might also be used. Also, the choice of sources and doublets for modeling a given problem is usually not unique. However, the use of doublets or, equivalently, vortices, is mandatory when modeling lifting flows. Such flows will generate a wake behind, which can be modeled by a doublet distribution having constant intensities downstream. The constant doublet intensity is related to the fact that the wake doesn't produce lift. However, for 3D wings, the doublet intensity will vary along the wing span direction. An equivalent approach is to model the wake by trailing vortex filaments.

In the linear potential initializer implemented hereby, solely a Neumann wall condition will be used, as the wing is modeled by a thin lifting surface, featuring only airfoil camber but no thickness. This is accurate enough in terms of lift production, for purpose of initializing the full potential solver.

## 1.3 Current approach formulation and motivation

This might be seen as a shortcoming of the full potential model: it is, by itself, unable to catch the character of a lifting flow. The Kutta condition has to be enforced explicitly. Some Euler models, though still inviscid, overcome this problem in pseudo-time approaches, apparently due to a mechanism which might be related to numerical dissipation and/or local temporary shocks, which together lead to the ability of generating vorticity in the presence of a pressure difference on a sharp edge, the actual behavior depending on several factors, including the fragmentation of the mesh. A more detailed discussion can be followed in [28]

For a full potential model, however, implementing a Kutta condition is mandatory, but the actual problem lies in the fact that the circulation generated by the wing is not known a priori. An exact value should be obtained as part of the solution. Therefore, not only the potential value in each node should be updated, but also the circulation (which is equal to the jump in potential across the wake). The potential throughout the flow domain is updated as a result of monitoring and reducing a mass flux residual while the circulation update comes from monitoring and reducing the recirculation residual at trailing edge level. To avoid any ambiguity, by recirculation we understand that the frontal wing-bound vorticity does not drop to zero at trailing edge level. This implies having a velocity component through the initial (trailing edge adjacent wake). In order to get a good estimation of this trailing edge recirculation residual, a reasonably low mass flux residual throughout the domain is necessary, but updating the circulation will more or less re-increase the mass flux residual and hence damage the convergence. In such circumstances, knowing a priori a good start value for the circulation would greatly help in reducing the run time, as lesser updates and of smaller amplitudes will be requested. The circulation should start with a value which is close to its correct expected value.

The current approach will exploit a linear potential model which will provide, at preprocessor stage, the circulation initial estimate. A panel model in form of a lifting surface with Prandtl-Glauert correction for compressibility has been chosen for this purpose. The lifting surface looks like the best choice, given that the only result demanded from this initializer is a good estimation of circulation. A more elaborated panel model (e.g. modeling wings with thickness) would be wasteful in this context. One point of view which is shared here is that integrating flow models of different accuracy levels (and computational cost) might be benefic for an overall step-wise solution. Such an approach provides a possible way of optimizing the computational effort.

Besides the Kutta condition which tends to slow down convergence, there is another issue with respect to the computational effort, related to the simulation of transonic flows. Full potential solvers are valued for being very fast. This is generally true for subsonic flows and is one of the best arguments in favor of such a model. Having a supersonic inclusion in the flow, however, changes the nature of the equations from elliptic to hyperbolic, slowing down the convergence, which undermines the main advantage of full potential models. While for subsonic flows the behavior is excellent in terms of both speed and accuracy, transonic solutions are less satisfying. These are limitations of the full potential model which have been tackled by different researchers. The accuracy is related to the potential model itself, because the isentropic nature of the full potential shock. Corrections to this problem have been suggested and can be followed in [28]. The speed of convergence, on the other side, remains a problem of full potential transonic flow simulations. Transonic solutions have been reported in most works related to full potential, but they are usually impractically slow. This issue has been addressed and a trade-off to speed up transonic convergence to a level comparable to subsonic performance can be followed in Section 2.4.4. This might be interesting especially for cases where the accuracy of a transonic solution is not important but rather the ability of the solver to provide a fast approximate solution, e.g. the case of a subsonic aircraft design, where the subsonic character of the flow needs to be checked throughout the flight envelope, or the case of a transonic wing for which a stronger shock is anyway undesirable. In such situations, fast transonic solutions are likely to speed up modification decisions in the design process.

The objectives of this thesis might be summarized like:

- development of a linear potential model for use as initializer of the (full potential) main solver
- implementation of a Kutta condition in the the non-linear main solver
- solver validation on complex 3D geometry and calibration for enhancing the ability to model real flows
- implementation and testing of a fast approximate transonic approach

The starting point in the development of the full potential solver is a finite volume approach using unstructured tetrahedral meshes, as described in references [1], [2] and [18]. Overall, this full potential approach has been developed keeping in mind the goal of a fast solver with practical applicability to lifting subsonic or mild transonic flows.

# Chapter 2

# Model implementation

## 2.1 General considerations

The 2D case is more straightforward, as it has only one span-wise constant value for circulation. A good analytical initial estimation of this value can be provided to the full potential code by means of simple formulas. Starting with this value, the full potential code will do the rest of the job in converging a correct solution. However, by providing an estimate which is close to the final value the computational efficiency is improved. Another way of initializing the full potential solver, would be by means of an airfoil-bound distribution of lumped vortices model which could provide a numerical initial estimation of lift. In the absence of a shock in the full potential solution, both analytical and numerical initializing methods bring the circulation close to its final full potential converged solution. The analytical initialization has however the advantage of simplicity, sparing a lumped vortex model run, and therefore has been preferred and adopted for the 2D case. The influence of the airfoil generated lift on the domain is determined using equation 1.7.

For a 3D wing or wing-fuselage combination, there is no accurate analytical approach for producing a good initial guess of the wing-bound circulation. An initializer, in the form of a panel lifting surface, will provide the lift information.

A preprocessor program will import the estimated circulation distribution and do the actual job of building and smoothening the initial approximation of the full potential solution. The preprocessor will also provide a set of influence coefficients, which relate the wing-bound circulation to the potential jump at each node in the wake, this feature being necessary for updating the wake, along with the updated values for the wing circulation.

Finally, the full potential solver is provided with an initial approximation which is close to the correct, converged solution, in terms of circulation. Full potential solvers are known for being very fast in terms of mass flux residual reduction, but their efficiency is usually damaged by the need of updating the circulation. The goal here was to alleviate this inconvenient by carefully initializing the circulation.

The full potential solver uses meshes generated in Gambit (Fluent Inc.) [37], but, in principle, any other CAD tool capable of generating unstructured tetrahedral meshes might be used, with some adaptation required for the processing of the mesh files into the correct format. It is noted that the full potential tetrahedral mesh and the polar panel mesh used in the initializer are completely independent and hence don't restrict each other. There is a modeling freedom for both meshes. The panel model polar mesh generation will be described in the next section. For the tetrahedral full potential mesh no special requirement is made. Basically it would differ from an Euler mesh only through the presence of a wing wake which is materialized in the mesh as a surface cutting the domain (a smooth distribution of triangular faces to which wake adjacent tetrahedra will stick), but the FP will support also coarser meshes, which might not work with an Euler solver. In principle, existing meshes used for Euler models might be reused in the FP model with a slight modification (wake

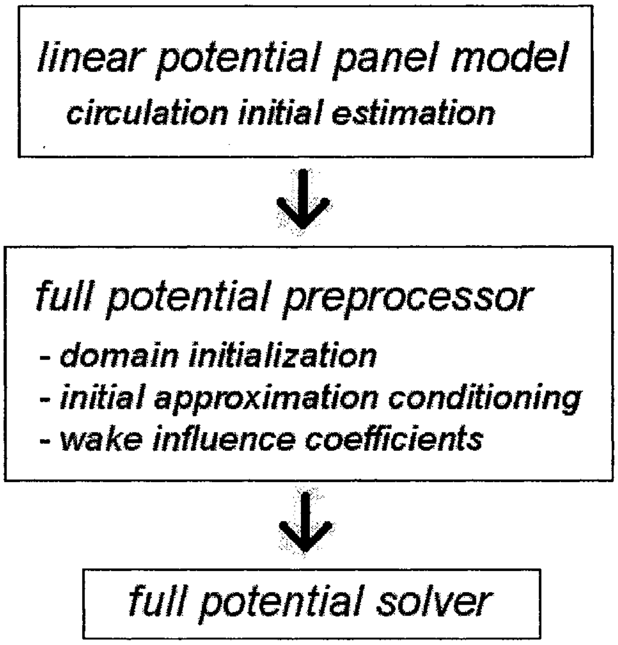
linear potential panel model
circulation initial estimation
full potential preprocessor
- domain initialization
- initial approximation conditioning
- wake influence coefficients
full potential solver

Figure 2.1: Model integration

remeshing) and meshes generated for the FP model might be used unmodified[1] for Euler simulations, provided they are fine enough to avoid generating numerical entropy.

## 2.2 The panel model initializer

No detailed account on linear potential panel models will be given here (reference [29] could be consulted for this purpose, as it is the most recent and exhaustive textbook on this issue). Rather, the particularities of the polar modeling of the lifting surface and the arguments which led to this approach will be highlighted.

[1] The geometry doesn't need to be modified / remeshed, only the boundary condition on the wake should be adjusted to the respective solver

### 2.2.1 Considerations of Kutta conditions for panel models

As formulated for 2D wings, the Kutta condition implies that for a sharp trailing edge and with a steady flow attached to the wing, the vortex intensity will drop to zero at the trailing edge. Before proceeding to deeper considerations on consistently accounting for this condition on 3D wings, a short analysis of the wing-bound vorticity will be attempted.

Consider an infinite-small spot on the thin-wing surface, as shown in Figure 2.2. The Helmholtz theorem implies that the total variation of vortex intensity on the spot is zero, i.e.:

$$\frac{\partial \gamma_{frontal}}{\partial r} = -\frac{\partial \gamma_{lateral}}{r \cdot \partial \theta} \tag{2.1}$$

With the frontal wing-bound vortices oriented along center lines and having the intensity decaying from mid-wing towards wingtip, the wing-bound lateral vortices (normal to center lines) will follow equation 2.1 which relates the variation of frontal vortex intensity with the generation of lateral vortices. The coordinate system has been chosen as the polar system with origin at the intersection of the leading and trailing edge supporting lines.

At this point it should be emphasized that not only the frontal, but also the lateral vortices (being also wing-bound) are capable of generating aerodynamic forces acting on the wing, as long as they are not aligned with the unperturbed flow. For straight wings, or wings with small sweep angle the effect might be negligible but when calculating wings with greater sweep there is no reason to ignore it. The lateral wing-bound vortices are in part accountable for the different behavior of forward vs. aft swept wings, as their contribution to lift production is inverted with the sign of sweep angle. Aligning a priori the lateral vortices with the free

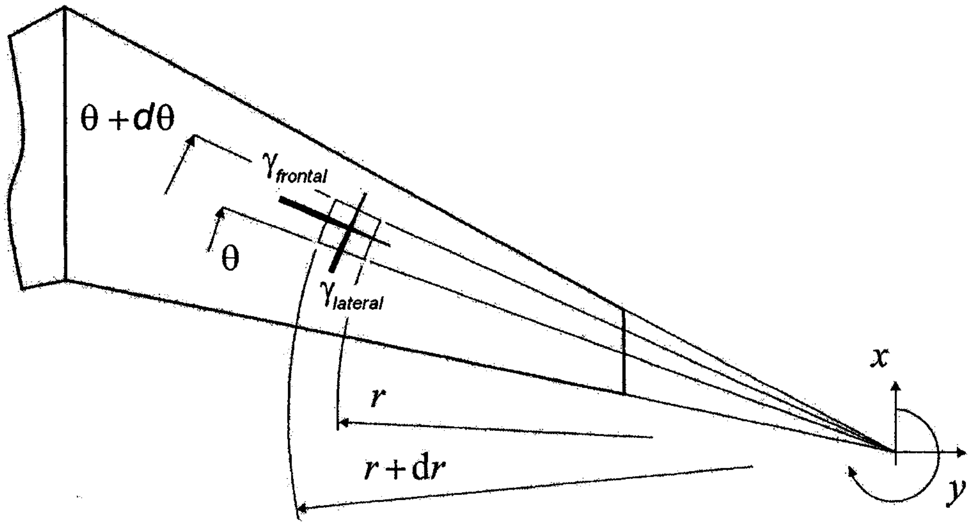

Figure 2.2: Wing in polar coordinates

flow means hiding their contribution to (positive or negative) lift. Moreover, the calculated lateral-vortices-induced downwash is influenced by the orientation of the lateral vortices. Flow oriented lateral vortices would underestimate the downwash and the induced drag of an aft-swept wing while overevaluating them for a forward-sweep. The hypothesis of flow oriented lateral vortices is desirable for keeping things simple when calculating straight and low-taper wings, but it is somehow arbitrary for swept and/or taper wings and it infringes upon the Kutta condition as shown in the following.

For a 3D wing, the Kutta condition as seen in the polar model, would ask for the wing-bound vortices to be oriented as for reaching the trailing edge with no frontal component, i.e., the connection of wing-bound vortices with the wake should occur only through vortices that are normal to the trailing edge. Having in the immediate vicinity of the trailing edge lateral wing-bound vortices that are not perpendicular to the trailing edge is susceptible of introducing modeling errors as it is inconsistent with the (sometimes only 2D considered) Kutta condition. Further on the same idea, there is no reason to believe that only in the vicinity of the trailing edge the bound lateral vortices should submit a normal-to-center-line orientation, while one could have them arbitrarily oriented inside the thin-wing surface. It is generally accepted in current panel methods that the frontal (2D similar) vortices are oriented along center-lines. Equation 2.1 implies that the lateral vortices are generated normal to center-lines, i.e. not arbitrarily oriented to some convenient direction. Subsequent to their production, the lateral vortex filaments will gradually bend to keep the local normal-to-center-line orientation and eventually reach the trailing edge still being center-line perpendicular, as the trailing edge also lies on a center-line. Lateral vortices are generated all along the wing chord, thus their filaments (and intensities) are added gradually towards the trailing edge, in the same manner that the trailing vortices of the classical horseshoe-modeling are superimposed. The significant difference consists in the geometry that the filaments should submit.

Modeling the wing-bound vortices this way has some important consequences. The lateral vortices won't just induce a downwash on the wing but also contribute to, or reduce its lift (to be more precise, for center lines with $\theta > 270°$ the contribution to lift will be negative and for $\theta < 270°$ it will be positive). This means that for sufficiently high sweep angles, where both the leading and trailing edges are located $\theta > 270°$ or $\theta < 270°$ all lateral vortices will contribute the same way to decrease or, respectively, increase the total wing lift. One could herein find some explanations for the better aerodynamic performance of forward swept wings when compared to their aft counterparts, these considerations standing under the assumptions of a completely attached flow.

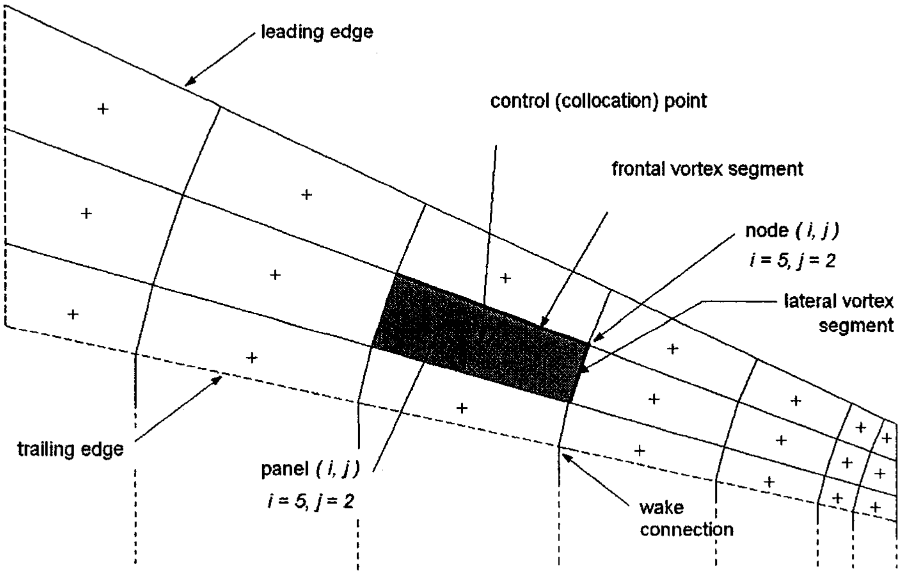

Figure 2.3: Planar polar mesh

### 2.2.2 The polar wing model

Generating the polar lumped vortex mesh for numerical computation is straightforward. A simple way is suggested below and illustrated in Figure 2.3.

One could use the intersection of the lines supporting the leading and the trailing edges of the right wing as origin for the polar coordinate system superimposed on a Cartesian system having x-axis forward and y-axis to the right (Figure 2.2). The nodes of the mesh will then be generated by keeping one polar coordinate constant and varying the other one in steps. To be more specific, two consecutive nodes limiting a frontal vortex segment will have the same $\theta$ but consecutive values for the radius r, while two consecutive nodes limiting a lateral vortex segment will have the same radius r and consecutive values for the angular coordinate $\theta$. The values for the radius increase inward from wingtip to mid-wing and the angular coordinate increases from trailing edge to leading edge. The considerations are valid for

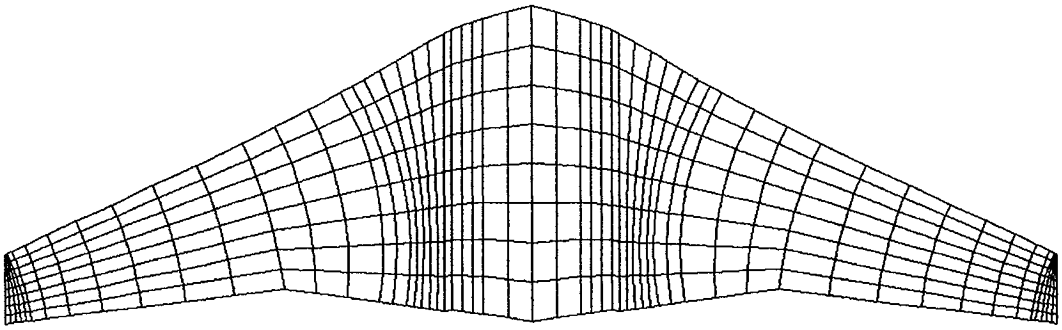

Figure 2.4: Multisegment wing

modeling the right wing. The left wing could then be modeled through a simple mirroring technique, which could be achieved after converting the mesh data of the right wing in Cartesian coordinates and translating the Cartesian coordinate system in a wing-symmetric position on the mid-wing chord.

For non-planar wings (dihedral, camber), a third Cartesian z-coordinate can be added to each node after planar modeling. For asymmetric wings (aileron deflection, uneven flap retraction or other reason) individual modeling of each wing's geometry would be necessary.

For modeling multi-segment wings, each wing segment has its own polar coordinate system, the mesh of each section obeying to the local coordinates. At segment interface, the mesh adapts gradually from one system to the next (Figure 2.4). For modeling wing-fuselage combinations a classical mesh for the hidden portion of the wing can be generated, with the geometrical incidence lowered in order to catch the loss in circulation due to the fuselage. A torsion of the wing can actually be modeled throughout the wing span, with local angles imposed in each segment-interface section and having the angle varying continuously in-between. A particular case would be the wing with little or no taper ratio. The origin of the polar system would be to

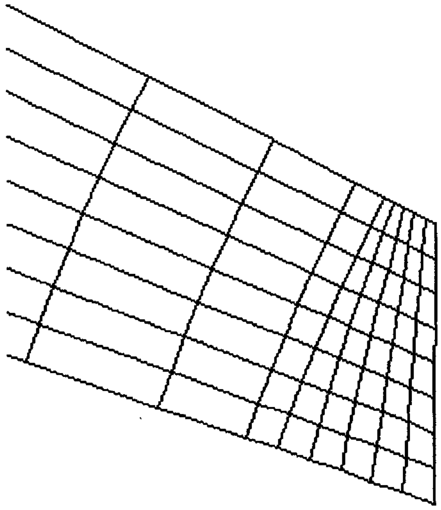

Figure 2.5: Wingtip mesh

far from the wing, if the taper ratio is too small or infinitely far for parallel leading and trailing edges. However, wing designs with nonzero but very small taper ratios are unlikely to occur as they would benefit neither from the better aerodynamics of taper wings nor from the simpler fabrication technology of wings with constant chord. For the later, the modeling could follow a simplified pattern, using Cartesian coordinates with one direction parallel to the leading and trailing edges (similar to center-lines in the polar model) for the frontal vortices and the other direction perpendicular to the first, supporting the lateral vortices. The other considerations would be similar to the polar model.

One more issue related to the generation of the vortex mesh for high sweep angles would be the need of a locally supplementary fragmented modeling of the wingtip (as exemplified in Figure 2.5 for an aft-swept wing), due to the misalignment of wingtip side and the regular lateral vortices as generated in the polar model. This wingtip rotated mesh should better account for the 3D character of the flow.

With regard to the chord-wise placement of the frontal lumped vortices and the control (collocation) points, the rule of placing the frontal vortex at 1/4 and the collocation point at 3/4 of the panel chord seems to be largely accepted in lifting surface models. A slightly different approach will however be preferred here. With the sum of panels covering the planform of the wing, vortices will be placed on panel sides while collocation points go to panel centers. For the last row of collocation points, in trailing edge vicinity, a closer-to-trailing-edge position is chosen. Actually, an optimal position for the collocation points might be experimentally obtained from a retro-calibration with feedback from the full potential results, judging on the error of the initial estimate of circulation. The issue is however of secondary importance, as the panel model solution is meant for initialization and the correct solution will be eventually provided by the full potential solver. This approach has been chosen for its advantage of a better geometric match between the lifting surface and the vortex rings attached to panels (the wireframe will stick to the surface). It should be noted that the vortex rings around the panels are completely equivalent with doublets uniformly distributed over the same panels, having the strength equal to that of the vortex ring. This equivalence will be used in the full potential preprocessor for initialization.

**Wake modeling** For the wake model, the choice most consistent with potential flow theory is to have, in the trailing edge vicinity, the trailing vortex filaments (or equivalently doublet panels) aligned with the lifting surface (which is here equivalent to the bisector surface of a wing with thickness). The wake influence on the wing decreases rapidly aft and, for an isolated wing, there is no critical constraint on the far wake model. Things could be different when modeling a combination of several lifting surfaces (e.g. wing + horizontal tail). In this case, a longer portion of the wake should be given attention regarding shape and position, as it influences the stability and trim of the studied lifting surfaces system. A wake relaxation with several steps might be a reasonable solution to this issue. Additional aspects on wake evolution might be followed in related references, e.g. [24], [25].

### 2.2.3 The linear system of equations

Numerical calculations can be made similar to the usual lumped-vortex models, i.e., solving a linear system of equations whose solution provides values for both the frontal and lateral vortices. While modeling the right wing, (Figures 2.2 and 2.3), the discrete equivalent of Equation 2.1, for mesh node $(r_i, \theta_j)$ will be:

$$\Gamma_{frontal}(r_i, \theta_j) - \Gamma_{frontal}(r_{i-1}, \theta_j) = \Gamma_{lateral\ generated}(r_i, \theta_j) \tag{2.2}$$

$$\Gamma_{lateral\ cumulated}(r_i, \theta_j) = \Gamma_{lateral\ generated}(r_i, \theta_j) + \Gamma_{lateral\ cumulated}(r_i, \theta_{j-1}) \tag{2.3}$$

where the value of $r$ increases with index, from wingtip to mid-wing, while the value of $\theta$ decreases with increasing index, from leading edge to trailing edge. The indexes are equivalent as $(r_i, \theta_j) \leftrightarrow (i, j)$. 2.2 for the wing tip nodes becomes:

$$\Gamma_{frontal}(r_{wingtip}, \theta_j) = \Gamma_{lateral\ generated}(r_{wingtip}, \theta_j) \tag{2.4}$$

with index equivalence $(r_{wingtip}, \theta_j) \leftrightarrow (1, j)$ for the wingtip nodes. Equation 2.3 for the first $\theta$ aft leading edge will feature no anterior cumulated lateral vortex:

$$\Gamma_{lateral\ cumulated}(r_i, \theta_1) = \Gamma_{lateral\ generated}(r_i, \theta_1) \tag{2.5}$$

The right wing will be modeled with $m$ nodes span-wise and $n$ nodes chord-wise (not including the nodes on the mid-wing chord and trailing edge but including the wingtip nodes). The left wing modeled through mirroring and adding $n$+1 nodes on the mid-wing chord and, for each wing, $m$ nodes on the trailing edge, would result in a mesh composed of $D$ frontal and $D$ lateral vortex segments on each wing, where $D$=$m$ x $n$. This gives $D$ panels on each wing.

Equations 2.2 or 2.4 written for all nodes of both wing sides will provide 2x$D$ of the equations for building the linear system. The other 2x$D$ will be provided by balancing the total induced velocity in each collocation point with the perturbation velocity requested by the lifting surface geometry and angle of attack. The total induced velocity in a collocation point is the sum of all frontal and lateral vortex influences. These individually induced velocities come in form of influence coefficients multiplying the vortex intensities, and obey the Biot-Savart law, which has been complemented with a Prandtl-Glauert correction to account for compressibility effects. Together, a 4x$D$ equations linear system is obtained, having all frontal and lateral vortex intensities as unknowns.

This approach focuses on lateral vortex generated intensities rather than lateral vortex cumulated intensities and will consider lateral vortices of length varying between one segment (generated near the trailing edge, of $\Gamma_{lateral\ generated}(i,j{=}n)$ intensity), up to $n$ segments (generated near the leading edge, ending at trailing edge, of $\Gamma_{lateral\ generated}(i,j{=}1)$ intensity). Consequently, a lateral vortex filament would begin in some node $(i,\ j{=}1\ldots n)$ and end on the node $(i,n{+}1)$ on the trailing edge. Hereby, the extended lateral vortex is defined as the vortex having the filament of the lateral vortex continued with the filament of the corresponding wake vortex. This approach is used for solving the system of equations.

Another approach will focus on lateral vortex cumulated intensities and therefore on lateral vortices of one segment length. It is applied to build the panel vortex rings, or equivalently the panel attached doublets, which will be used by the full potential preprocessor. This equivalence between vortex rings and panels with constant doublet distribution (Figure 2.6) is the basis for transferring the lift information from the panel model to the full potential model.

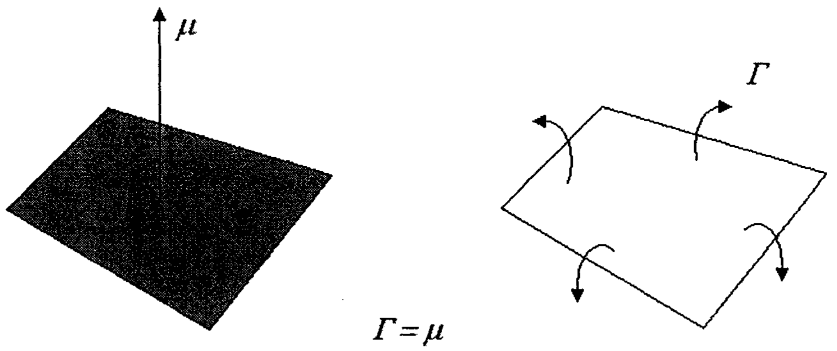

Figure 2.6: Doublet - vortex ring equivalence

## 2.3 The full potential preprocessor

As the initializer doesn't compute the whole domain around the lifting body but only the wing bound singularities, the full potential preprocessor will import this data and perform the actual domain initialization.

The preprocessor will initialize the wake nodes with a potential jump corresponding to the estimation of lift, varying span-wise along with the wing bound circulation, but keeping a constant value along x-axis. During later update, in the full potential code, the wake potential jump will obey the same rule of span-wise variation and x-wise constant value. An exception from this rule can be made for wing-fuselage combinations, where a wingroot wake 'turbulence' could be modeled, leading to both a span-wise (y-wise) and longitudinal (x-wise) variation of the wake intensity (potential jump). This is a simple mean of modeling the wingroot specific wing-fuselage boundary layer interference, which is responsible for the turbulence and recirculation in and behind the wingroot area. At implementation level, this may take the form of a wake map, which will modify the potential jump value with a 'turbulence' coefficient, effective only in the wingroot wake. Hence, in this case, the value of the potential jump through the wake will be a function of ($x$,$y$) coordinates. If no wingroot 'turbulence'(recirculation) is modeled, then only a span-wise variation of the wake potential jump will be present. If modeled, the intensity of the wingroot wake turbulence will increase with increasing lift (or angle of attack). However, a realistic calibration of this model should rely on experimental and / or viscid flow simulation data.

The preprocessor will provide a set of influence coefficients, which relate the wing-bound circulation to the potential jump in the wake nodes, this feature being necessary for updating the wake, along with the updated values for the wing circulation. Each node in the wake will be influenced by two consecutive nodes on the trailing edge (or only one node in the case of y-coordinate coincidence). The two trailing edge nodes influencing a given wake node will be the nodes closest to the wake node, in terms of y-coordinate (span position), as shown in figure 2.8.

The influence coefficients will be calculated interpolating on the trailing edge, between the two (left and right) corresponding trailing edge node(s) as to yield:

$$\phi(w_k) = \phi(t_l) \tag{2.6}$$

$$\phi(w_i) = \frac{\phi(t_m) \cdot d_{left} + \phi(t_n) \cdot d_{right}}{d} \tag{2.7}$$

The outer boundary of the computational domain will be initialized with the correct

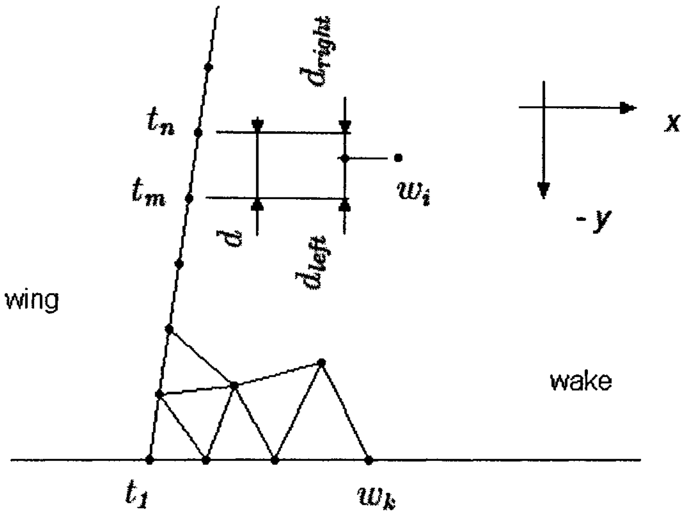

Figure 2.7: Wake influence coefficients

potential values reflecting the presence of a lifting flow. This makes possible both an increase of the final solution accuracy (as the boundary is made aware of the presence of lift) and a reduction in computational effort, because a smaller domain can be exploited. The circulatory component of the potential on outer boundary nodes comprises two influences, one of the wing-bound circulation, the other of the wake carrying vorticity. Both influences will be calculated using the results imported from the initializer.

The wing attached quadrilateral panels (as defined in the panel model) and the wake attached triangular faces will be assigned doublets whose intensities are, for the first, imported from the panel code, for the second calculated in the preprocessor, by means of interpolating on the span-wise wake intensity distribution (actually the span-wise wing-bound circulation). The cumulated influence of wing and wake carried doublets will build the circulatory component of potential value in the outer boundary nodes. To this, the unperturbed flow potential component will be added.

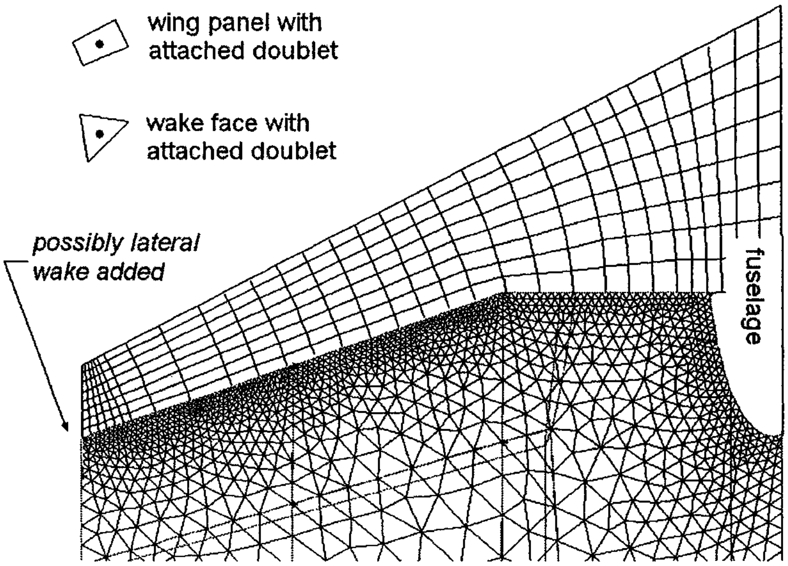
wing panel with attached doublet
wake face with attached doublet
possibly lateral wake added
fuselage

Figure 2.8: Wing and wake attached doublets

Following [29], the influence of a wing-bound or wake-bound doublet on a node in the outer boundary will be, for an incompressible flow:

$$\phi_x(x,y,z) = \frac{\mu_x}{4\pi}(x-x_0)[(x-x_0)^2+(y-y_0)^2+(z-z_0)^2]^{-3/2} \tag{2.8}$$
$$\phi_y(x,y,z) = \frac{\mu_y}{4\pi}(y-y_0)[(x-x_0)^2+(y-y_0)^2+(z-z_0)^2]^{-3/2} \tag{2.9}$$
$$\phi_z(x,y,z) = \frac{\mu_z}{4\pi}(z-z_0)[(x-x_0)^2+(y-y_0)^2+(z-z_0)^2]^{-3/2} \tag{2.10}$$

with $(x, y, z)$ the coordinates of the influenced node and $(x_0, y_0, z_0)$ the coordinates of the doublet position (for wing-bound doublets the panel center, for wake-bound doublets the center of a wake mesh face). Note that $(\mu_x, \mu_y, \mu_z)$ is the doublet intensity.

For compressible flows a geometric correction will be applied, with respect to the

$\overrightarrow{f_\infty}$ unperturbed flow direction as:

$$\phi_x(x,y,z) = \frac{\mu_x}{4\pi} d_x[d_x^2 + d_y^2 + d_z^2]^{-3/2} \tag{2.11}$$

$$\phi_y(x,y,z) = \frac{\mu_x}{4\pi} d_y[d_x^2 + d_y^2 + d_z^2]^{-3/2} \tag{2.12}$$

$$\phi_z(x,y,z) = \frac{\mu_x}{4\pi} d_z[d_x^2 + d_y^2 + d_z^2]^{-3/2} \tag{2.13}$$

where $\overrightarrow{d}$ is a corrected position vector,

$$d_x = \frac{(x - x_0)}{\sqrt{1 - (M_\infty f_\infty(x))^2}} \tag{2.14}$$

$$d_y = \frac{(y - y_0)}{\sqrt{1 - (M_\infty f_\infty(y))^2}} \tag{2.15}$$

$$d_z = \frac{(z - z_0)}{\sqrt{1 - (M_\infty f_\infty(z))^2}} \tag{2.16}$$

The total potential induced by a doublet in a node will be

$$\phi_{node}(x,y,z) = \phi_x(x,y,z) + \phi_y(x,y,z) + \phi_z(x,y,z) \tag{2.17}$$

The doublet influence is computed using farfield formulas, i.e., considering concentrated doublets placed in the center of wing panels or wake faces, which makes the preprocessor faster. It has been shown ([29] pp. 251-254) that considering distributed doublets increases computational cost, while, for the purpose of outer boundary initialization, the concentrated doublets provide the same accuracy. For ensuring this accuracy, wake faces near the outer boundary (wake-boundary intersection) have been split into 4 triangular subfaces, each getting a correct, individually interpolated (with respect to y-axis position) doublet strength. This measure brings all boundary points out of the range where a difference would be noticeable between the concentrated and distributed doublet influence.

All the above considerations apply for the whole wing and the whole wake, not only the portions inside the full potential computational domain, i.e., for the case of a symmetric flow, where only half of the wing is studied in the full potential code, the other's half influence should be included in the outer boundary potential. Furthermore, a sufficiently long wake portion outside the domain should be considered, when initializing the outer boundary. This boundary is actually reflecting everything happening inside and outside the numerical domain.

The last task of the preprocessor is to ensure a smoothening of the full potential initial approximation, eliminating sharp gradients at the interface of the wake and the outer boundary with the interior of the domain, while diffusing the potential intensity into the domain. This step is not only meant to produce a better initial approximation but failing to smoothen the initialization might prevent convergence because of sharp gradients which are left close to the wake. The smoothening takes place while running several times through all segments (node pairs) in the domain and diffusing the potential from the nodes with higher absolute value to nodes with lower absolute potential value. Wake and outer boundary nodes are protected by node-type flags during the smoothening, to prevent altering their initialized values. With a pointer based array access, the smoothening is very fast, while sparing from the full potential code run time by providing a better initialization.

## 2.4 The full potential solver

### 2.4.1 Domain discretization

The implementation will use a finite volume approach, applied to an unstructured tetrahedral mesh, as developed in [1], [2] and [18]. The node centered approach has been preferred, as this allows a good fragmentation of the mesh with a relatively small problem dimension. For an unstructured tetrahedral mesh, the ratio of number of nodes (storing the unknowns) to the number of tetrahedral cells can be as low

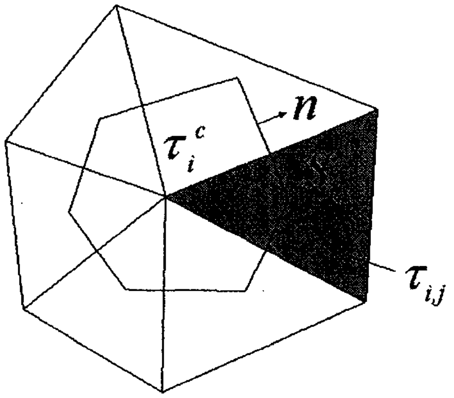

Figure 2.9: Section through a control volume

as 1:5. The full potential equation in integral form is written for a node centered control volume as:

$$\int_{\tau_i} \nabla \cdot G(\phi) dA = 0 \tag{2.18}$$

where $G(\phi) = \rho\nabla\phi$ with $\nabla(\phi) = (u,v,w)^T$ the gradient (velocity vector). The index $i$ marks the current node (and adjacent control volume). The sum of all control volumes will cover the computational domain $\tau$.

The potential values are stored at nodes, while the gradient and all the physical dependent variables are evaluated in tetrahedra. Thus, the discretized mass flux equation through a cell's walls will be:

$$\int_{\tau_i^c} \nabla \cdot G(\phi) dA = \int_{s_i} \rho\nabla\phi \cdot \mathbf{n} ds = \sum_{\tau_{i,j}} \rho_j(\nabla\phi)_j \cdot S_{i,j} \tag{2.19}$$

where $S_{i,j}$ is the part of the $i^{th}$ control cell vectorized boundary inside the tetrahedron $j$.

The boundary conditions are those presented in Section 1.2.2

### 2.4.2 Kutta condition

Both for 2D and 3D computations, during the mesh generation, a smooth cut of the domain is also generated, which will impose to the adjacent tetrahedra to align one face with the domain cut. For 2D meshes, the cut will simply be a line extending between the airfoil trailing edge and the domain outer boundary, the exact orientation of the cut playing no physical role and having no influence on the results. For 3D meshes, the surface will also play the role of a wake containing trailing vorticity. The wake geometry will need to obey certain constraints, where a special attention needs to be given to the orientation in trailing edge vicinity.

For modeling the vorticity content of the wake, each node belonging to it will have two potential values attached, one corresponding to the lower face and one to the upper. Or, equivalently, it has a mean value and a jump value. The jump value at a given span ($y$ coordinate) will be equal to the wing bound circulation at the same span location. Therefore, the wake needs to be updated along with the wing bound circulation, which is done by means of influence coefficients, relating each wake point to one or two (in terms of $y$ coordinate) point(s) on the trailing edge. Actually, the trailing edge points, which serve both as front of the wake and closure for the wing circulation will be the reference storage for lift information.

For wing-fuselage combinations, the values of potential jump throughout the wake might contain one more coefficient, which comes as a wake dissipation map (a function of $(x,y)$). This allows the implementation of a wake dissipation model, which is meant to catch the loss of organized vorticity into turbulence in wingroot area. Such a model is of a rather empirical nature and could be the subject of a calibration, following experimental data and/or Navier Stokes solutions for typical cases.

In all wake points, including trailing edge, the mean value of the potential is allowed

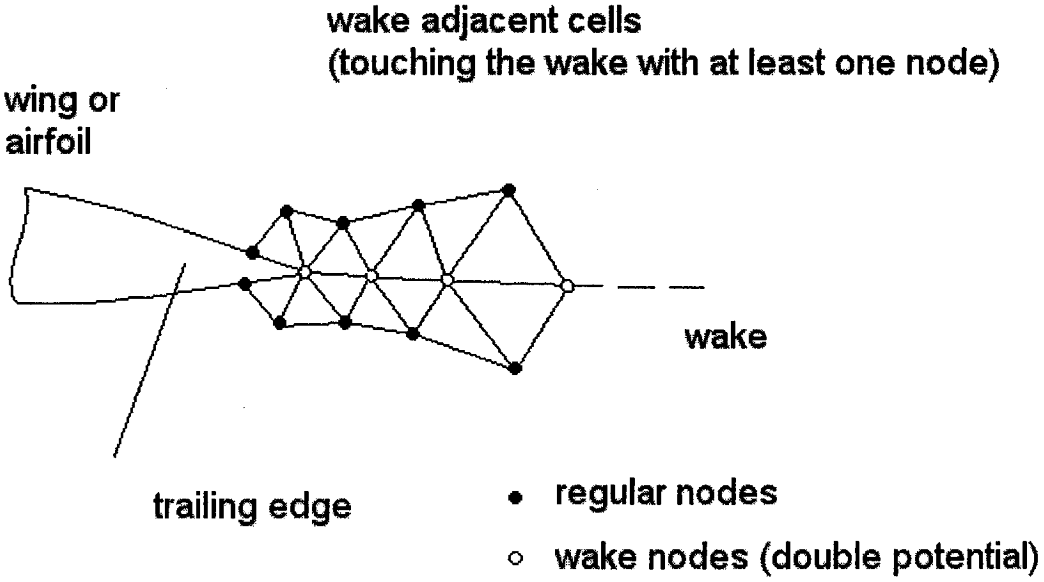
wake adjacent cells
(touching the wake with at least one node)
wing or airfoil
wake
trailing edge
● regular nodes
○ wake nodes (double potential)

Figure 2.10: Section through the wake

to evolve with the mass flux residual reduction, while the jump value is updated along with the circulation evaluation.

When calculating the gradient for a wake-adjacent tetrahedron, the nodes lying on the wake will be assigned a potential value $\phi = \phi_{mean} \pm 0.5 \cdot \phi_{jump}$, the algorithm deciding the sign upon the type of tetrahedron (+ for tetrahedra above the wake, - for tetrahedra below the wake). However, all the wake adjacent tetrahedra will behave as all regular tetrahedra in terms of mass conservation, which will be satisfied through the wake. The triangular faces on the wake are permeable, no boundary condition is applied to these faces, except the already mentioned potential jump.

### 2.4.3 Algorithms

**Mass flux reduction algorithms**

An iterative scheme is needed in order to reduce the mass flux residual in equation 2.19 below an acceptable value.

**Explicit approach** Explicit schemes have simple implementations but are rather inefficient computationally. The simplest explicit algorithm would be:

$$\phi_i^{n+1} = \phi_i^n + c^n R_i \tag{2.20}$$

where $R_i$ is the local residual, $c^n$ is an adaptable step, whose value will vary as needed to keep the solution converging, upon monitoring the residual. This approach has not been used here.

**Implicit approach** Due to variable density in the full potential equation, a Newton based method has to be used. A linearization of the problem will build each step in a Newton method.

$$J^n \cdot \delta\Phi^n = -\mathbf{R}^n \tag{2.21}$$

where $J$ is the Jacobian matrix, $\delta\Phi$ the array of potential updates, $\mathbf{R}$ the array of mass flux residuals and $n$ the current Newton step. After updating the potential, the mass flux is recalculated and new residual and Jacobian are found.

$$\Phi^{n+1} = \Phi^n + \delta\Phi^n \tag{2.22}$$

where $\Phi^n$ is the array of potential values at step $n$. With a sparse Jacobian matrix, a GMRES solver [30] will be used for solving each linear step.

*The Jacobian Matrix*

For subsonic flows, the Jacobian will be build as follows:

- *diagonal elements* will measure the variation of mass flux over a cell with respect to the variation of potential value in the cell's center node.

$$J(i,i) = \frac{\partial R_i}{\partial \phi_i} \approx \frac{R_i(\phi + \epsilon) - R_i(\phi)}{\epsilon} \tag{2.23}$$

- *non-diagonal elements* will measure the variation of mass flux over a cell with respect to the variation of potential value in the peripheral nodes.

$$J(i,j) = \frac{\partial R_i}{\partial \phi_j} \approx \frac{R_i(\phi_j + \epsilon) - R_i(\phi_j)}{\epsilon} \tag{2.24}$$

Here, $i$ denotes the center and $j$ the neigboring nodes. Non-diagonal terms are calculated and stored on basis of mesh segments (neighboring node pairs). Each node $i$ will be bound by mesh segment to its $j$ neighbors. In some situations a second order Jacobian, though apparently more expensive, produces better convergence.

### Trailing edge recirculation reduction algorithms

The residual measuring the trailing edge recirculation needs to be defined (using a Kutta condition). A common Kutta condition is based on velocity (or pressure equality), but this approach will not be used here. Rather, a Kutta condition pertaining to the wake orientation (in trailing edge vicinity) will be preferred, as this allows more flexibility in modeling real flows (see comments in Chapter 3). Hence, the flux through the first row of wake faces (trailing edge adjacent) will be monitored, in order to quantify the residual.

During the Newton steps, the recirculation residual will be monitored all along the trailing edge. A recirculation residual array (with as many elements as tetrahedra pairs along trailing edge), will be fed to the circulation update algorithm.

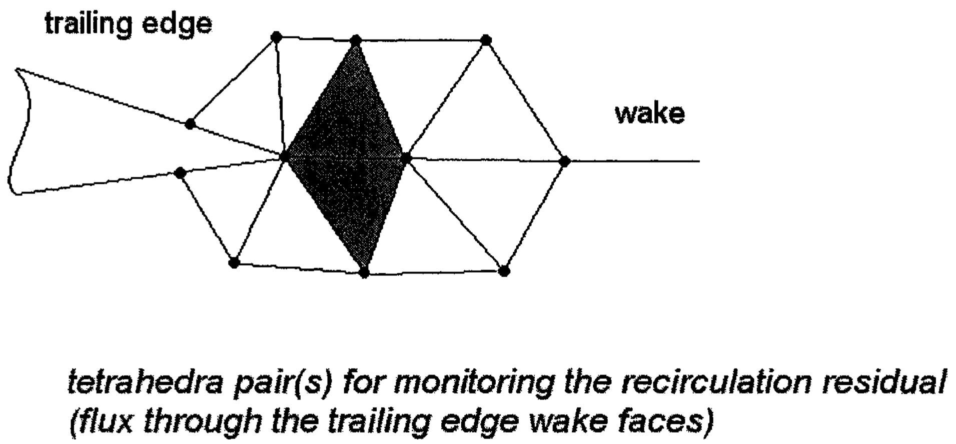
trailing edge
wake
tetrahedra pair(s) for monitoring the recirculation residual
(flux through the trailing edge wake faces)

Figure 2.11: Recirculation residual monitoring

With the recirculation residual evaluated as flux through the domain cut, in trailing edge vicinity, caution is mandatory with respect to the wake (cut) orientation, even for 2D flows. To be more precise, the cut should be aligned to the assumed trailing edge flow direction, both for 2D and 3D simulations. The physical meaning of this would be that the wake is basically aligned with the flow, which will not allow any flux through the wake in trailing edge vicinity. The fact that downstream a flux through the wake is allowed is not a physical requirement but rather a way to swallow the numerical discrepancy between the flow orientation and the position of the wake. Following existing implementations of the Kutta condition for full potential solvers, this flux existence across the wake is accepted, as long as it doesn't manifest itself in the trailing edge vicinity. For a rigorous, but rather expensive approach, this far wake discrepancy can be avoided by running a panel model with wake relaxation which would provide to the meshing application the actual shape of the far-wake, or, alternately, a mesh relaxation / wake-flow alignment might be performed in the FP solver.

**The 2D case** For studying a lifting airfoil, one supplementary unknown will be added to the problem, the airfoil-bound circulation. Upon monitoring a trailing edge Kutta condition related residual, an iterative updating of circulation will run parallel to the mass flux residual reduction algorithm. The processes will not interfere otherwise than through a (low enough) mass flux residual triggering a recirculation residual check and circulation update, and the circulation, when updated, disturbing the potential field and possibly increasing again the mass flux residual. Because of this damaging effect of circulation update on mass flux convergence, a good start value is desirable, as well as an optimal step in circulation update.

For updating the circulation value, both explicit and implicit approaches might be used.

*Explicit 2D circulation algorithm*

When the mass flux residual descends below an intermediate value, which might be fixed or dependent on the last evaluated recirculation residual, a reevaluation of the later will be triggered and its new value will be used for updating the circulation:

$$\Gamma^{n+1} = \Gamma^{n} + c^{n} R^{\Gamma} \tag{2.25}$$

Smaller values for $c^n$ will need more circulation updating steps but will perturb less the mass flux residual reduction. An overestimated value might induce an oscillatory behavior for the circulation while having the disadvantage of perturbing unnecessarily the mass flux residual reduction. The $c^n$ coefficient will be adjusted by the algorithm during circulation iterations, as to ensure a fast circulation refinement ($c^n$ sufficiently high), while avoiding oscillations in residual values ($c^n$ sufficiently low). A start value for $c^n$ will be around $mac \cdot 10e-4 ... mac$ to $10e-3$ (smaller values going with finer meshes), with $mac$ being the mean aerodynamic chord of the studied wing.

*Implicit 2D circulation algorithm*

An implicit approach might be formulated as the solution of a non-linear algebraic equation with the circulation as unknown. The secant method, the Newton-Raphson method, the method of tangent parabolas are possible candidates. The Newton method has been used, as the derivative of recirculation residual with respect to circulation value is computationally affordable, needing only a small perturbation of the potential field.

$$\Gamma^{n+1} = \Gamma^{n} - R^{\Gamma} \cdot \frac{\partial R^{\Gamma}}{\partial \Gamma} \tag{2.26}$$

$$\frac{\partial R^{\Gamma}}{\partial \Gamma} \approx \frac{R^{\Gamma}(\Gamma + \epsilon) - R^{\Gamma}(\Gamma)}{\epsilon} \tag{2.27}$$

The explicit approach has been preferred because it leads to a faster convergence.

**The 3D case** For 3D wings, the wing-bound circulation will vary along the span. Several values need to be considered, at different span locations. The updated values of the circulation will also be imposed on the wake nodes in the form of a potential jump, using the influence coefficients relating each wake node to one or two trailing edge nodes. The later are used for wing bound circulation reference, storing the updated circulation values (and potential jump). The approach chosen here, will fragment the circulation distribution down to the level of the mesh, i.e., the residual will be evaluated in each wake-bound tetrahedra pair along trailing edge (i.e. tetrahedra with 2 nodes on the trailing edge and one face on the wake), and the update of the trailing edge nodes potential jump will take place directly, related to the recirculation residual in the neighboring trailing edge wake faces. Subsequently, all wake nodes, other than trailing edge nodes, will be updated using the influence coefficients generated in the preprocessor. Monitoring the residual all along the trailing edge,

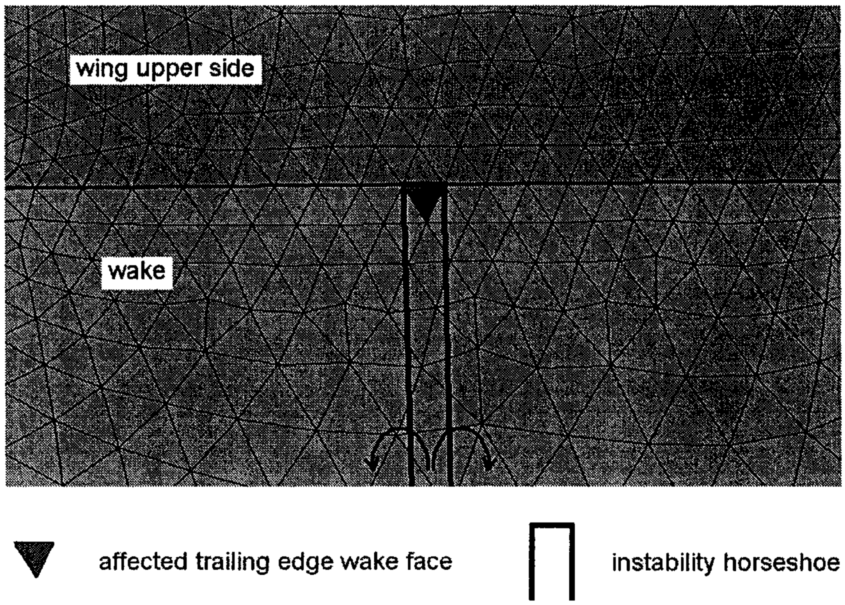

Figure 2.12: Circulation distribution instability with associated trailing horseshoe vortex

and not only in a limited number of locations, helps improving the stability of the circulation refining algorithm. Instabilities might still occur (such a behavior seems to be related to counterrotating parasite trailing vortices which will superimpose the correct solution, appearing in pairs on neighboring trailing edge nodes), but as they are reduced to the mesh size the algorithm will detect and smoothen them. Figure 2.12 shows the effect of the instability.

*Explicit 3D circulation algorithm*

Nodes potential updates will depend on the residual in the tetrahedra pairs containing the node. Each tetrahedra pair is assigned a residual measuring the flux through their common face.

$$R_i^\Gamma = \left(c_i^{upper}(\nabla\phi)_i^{upper} + c_i^{lower}(\nabla\phi)_i^{lower}\right) \cdot \mathbf{n}_i \cdot S_i \tag{2.28}$$

where $S_i$ is the wake face $i$ area and $\mathbf{n}_i$ is its normal. The gradient comes from the $i$th tetrahedra pair. Depending on the local mesh geometry, $c_i^{upper}$ and $c_i^{lower}$ relate the expected value in the wake face center to the gradient values in the tetrahedra pairs.

The update of the potential in trailing edge nodes will take place like:

$$\phi_i^{n+1} = \phi_i^n + c^n(c_{i^-}^{R^\Gamma} \cdot R_{i-1}^\Gamma + c_{i^+}^{R^\Gamma} \cdot R_i^\Gamma) \tag{2.29}$$

for $i = 2...(m-1)$, where $m$ is the dimension of the circulation problem (the number of trailing edge nodes) and $(m-1)$ the number of trailing edge tetrahedra pairs evaluating the recirculation residual. For the wingroot node and the wingtip node, there is only one wake face providing residual information:

$$\phi_1^{n+1} = \phi_1^n + c^n c_{1^+}^{R^\Gamma} \cdot R_1^\Gamma \tag{2.30}$$

$$\phi_m^{n+1} = \phi_m^n + c^n c_{m^-}^{R^\Gamma} \cdot R_{m-1}^\Gamma \tag{2.31}$$

where $c_{i^-}^{R^\Gamma}$ and $c_{i^+}^{R^\Gamma}$ are mesh dependent local coefficients relating the necessary update in trailing edge node potential to the neighboring recirculation residuals.

*Implicit 3D circulation algorithm*

Implicit algorithms will need a Jacobian relating the change in recirculation residual for each trailing edge face to the corresponding small perturbation in potential value for each trailing edge node. The calculation of this Jacobian is computationally expensive. A more efficient way would be to use an approximative but constant Jacobian which could be build using a panel model approach. In this case, the Jacobian would contain the influence coefficients relating the normal induced speed (generating circulation residual) in collocation points placed in the centers of the trailing edge wake faces, to the intensities of additional trailing vortices (reflecting the need of potential jump change in the wake). Though the Jacobian is constant for all steps (and hence needs to be calculated only once), the RHS of the problem (the recirculation residuals) will change between steps. Such an approach might need a smaller number of circulation updating steps than the explicit approach but will perturb the mass flux convergence more severely. This could eventually lead to the need of a smoothening process in the wake vicinity, similar to the one used in the preprocessor, applied at each step, in order to avoid sharp potential gradients at trailing edge level, following the update in potential jump. For these reasons, the explicit approach presented previously has been preferred in the actual implementation of circulation update algorithm.

*Residual free circulation algorithm*

For completeness, a residual free approach in recirculation reduction is also presented. Residual free doesn't mean that there is no recirculation residual at the trailing edge, during convergence, but rather that there is no need to evaluate it. Hence, the criterion to stop the algorithm is assuming convergence upon detecting a slowdown in wake potential jump update. This approach might seem simpler but involves a trade-off with the workload in mesh generation and preprocessing tasks. As shown in figure 2.13, imposing a wall-type condition on the initial wake (e.g. first row of wake faces, those who touch the trailing edge), will move the recirculation downstream on the wake mesh. Upon monitoring the potential difference on the (hence doubled) trailing edge points, now viewed rather as regular wall points, not initial wake points, this potential difference will be iteratively imposed to the downstream wake by means of the same influence coefficients as used in the other approaches. As the wake update slows down in magnitude with decreasing recirculation, this will indicate a circulation convergence, without explicitly monitoring

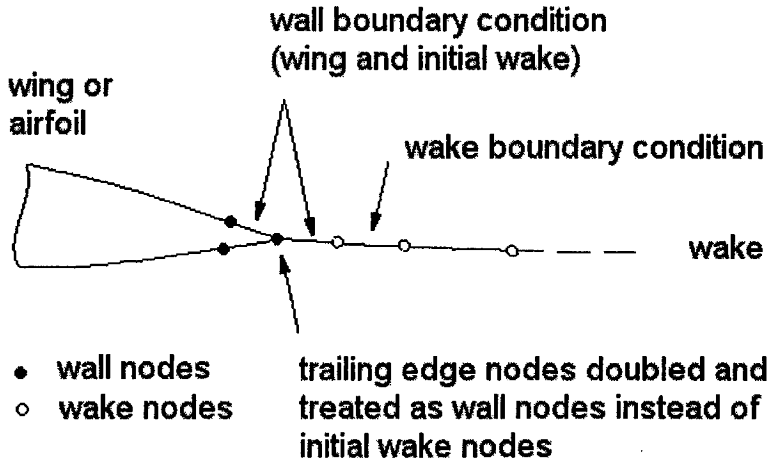

Figure 2.13: Residual free approach

the recirculation residual.

### 2.4.4 Fictitious gas transonic approximation

An approximate and much faster alternative to the standard FP transonic approach will be proposed here.

The idea of using modified gas properties above Mach 1 for circumventing the hyperbolic character of the flow appears in full potential works in the context of designing shockfree airfoils and wings. Reference [15] is an example. There, the gas compressibility is lowered above the critical conditions, hence producing an elliptical 'supersonic' flow. The approach taken here is slightly different, in that the compressibility is not modified right at M=1, but at a higher Mach number, hereby noted as $M_{threshold}$. Such a model is enabled to catch shocks, as it will actually produce three subdomains in the flow: subsonic, low (true) supersonic, and high (elliptical) 'supersonic'. The iterations might start with a lower value for $M_{threshold}$ (e.g. 1.02...1.05) or even 1.0, being subsequently increased. Thus, the solution will adapt gradually.

The fictitious compressibility is lowered above $Mach_{threshold}$ as:

$$\rho_{fictitious} = (\rho_{real} - \rho_{threshold}) \cdot c_{fictitious} \tag{2.32}$$

where $c_{fictitious}$ is the coefficient $< 1$ which actually fixes the degree of rigidity of the fictitious gas, $\rho_{real}$ is the value according to equation 1.3 and $\rho_{threshold}$ is the threshold value, corresponding to $Mach_{threshold}$.

Such a solution is useful in signaling, at lower computational cost, the presence of a transonic flow with shock, which is usually not desirable in any design, especially for stronger shocks. Waiting for a standard full potential solution to converge is wasteful if the only purpose is to check a design for the absence of shocks.

As the method proves to have no difficulty to converge both in terms of mass flux residual and circulation residual in the case of lifting flows (whereas standard full potential solutions might get even slower for lifting cases because of the interference between lift related circulation and shock position), it seems to be most useful for highly lifting flows at moderate speeds, which might accidentally produce small supersonic inclusions and shocks close to the leading edge on the upper side.

# Chapter 3

# Results and validation

In this chapter results for 2D and 3D flow simulations will be presented, to demonstrate the feasibility of our approach. The computational cost for obtaining the FP solutions is measured on a platform with the following characteristics: Intel Pentium 4 CPU 2.60GHz, 1.50GB RAM, Windows XP.

## 3.1 NACA0012 airfoil

The solver is first tested on a simple 2D flow. Nevertheless, a 3D computational domain is used. The mesh has 13066 nodes, 52967 tetrahedrons and 16570 boundary faces. The solver is designed for 3D flows but 2D flows can be simulated using a 3D mesh which is built on a 2D geometry placed between 2 parallel walls. The pressure coefficient in figures 3.2, 3.3, 3.4 and 3.5 is compared to Euler results obtained with the commercial solver Fluent, on the same mesh (for subsonic flows). For transonic cases, where the Euler solution failed to converge, a refined mesh had to be used. The mesh is good for FP simulations but might be too coarse/unbalanced for Euler simulations even in subsonic, as it seems to produce numerical entropy affecting the solution.

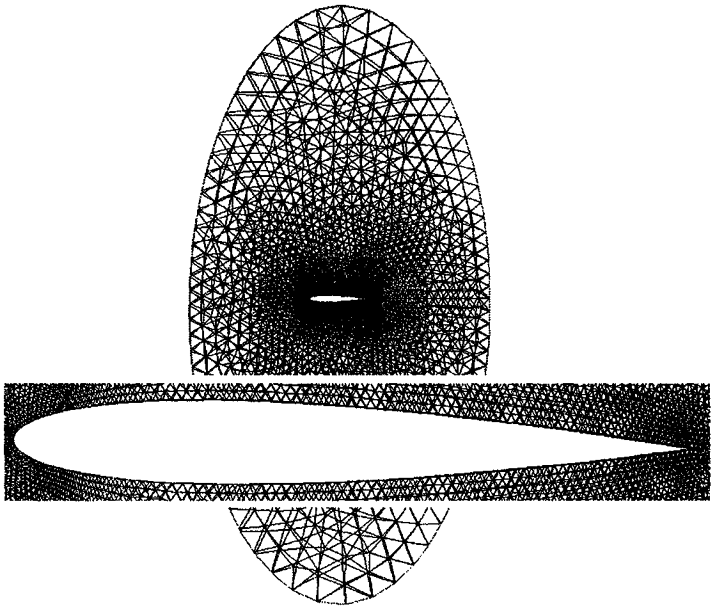

Figure 3.1: 3D mesh slice for 2D flow simulation over the NACA0012 airfoil

Figure 3.6 compares the computational cost of FP FG solutions for different $M_{threshold}$ Mach numbers, for a non-lifting flow at Mach 0.8. A lower compressibility threshold (and hence smaller hyperbolical domain) is helpful if fast solution check is desired. Transonic FG FP solutions, though not accurate give a good approximation of the $C_p$ with shock while converging fast. A possible application of such a solver could be in preliminary supercritical airfoil design. Obviously a Navier-Stokes solution will provide the most accurate transonic evaluation but a FG FP solver could help to rapidly identify and sort transonic solutions, where the approximate shock position is also provided.

Figure 3.7 compares the convergence of FP solutions at Mach 0.5, for 3 cases: non-lifting, lifting subsonic and lifting with transonic flow on the upper leading edge. It seems that the circulation is refined fast (though its initialization influences the start mass flux residual), without significantly disturbing the overall convergence. The computational cost is affected by the apparition of transonic flow, which is however

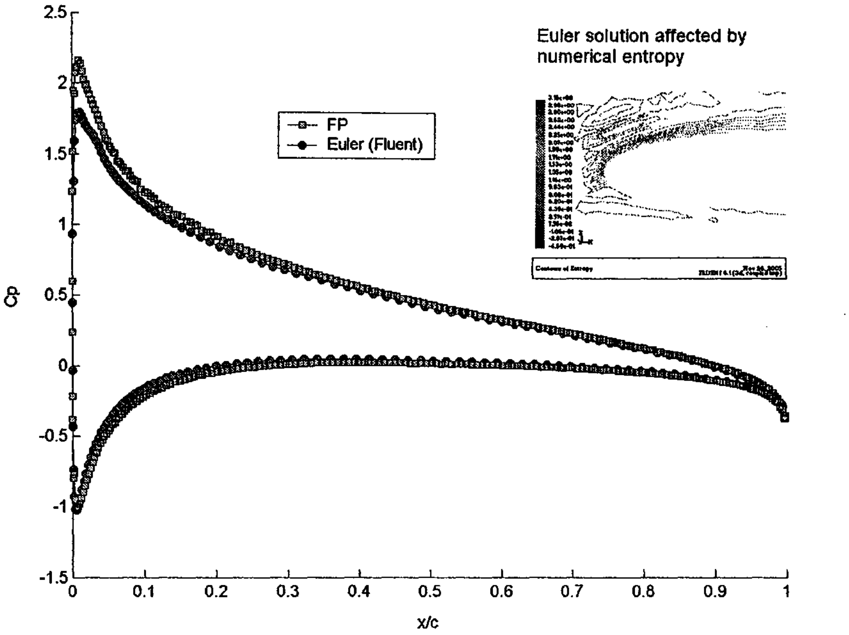

Figure 3.2: Pressure coefficient distribution at Mach 0.3, 5° incidence

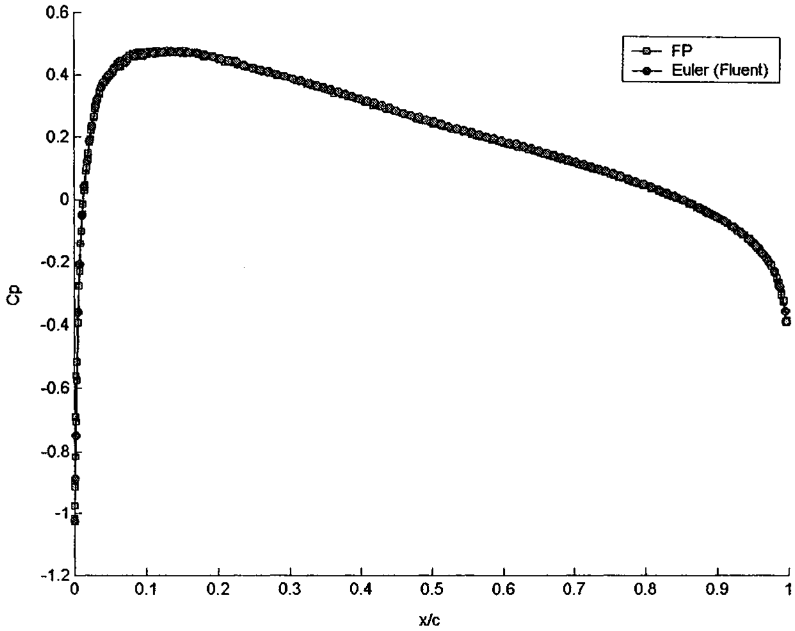

Figure 3.3: Pressure coefficient distribution at Mach 0.5, 0° incidence

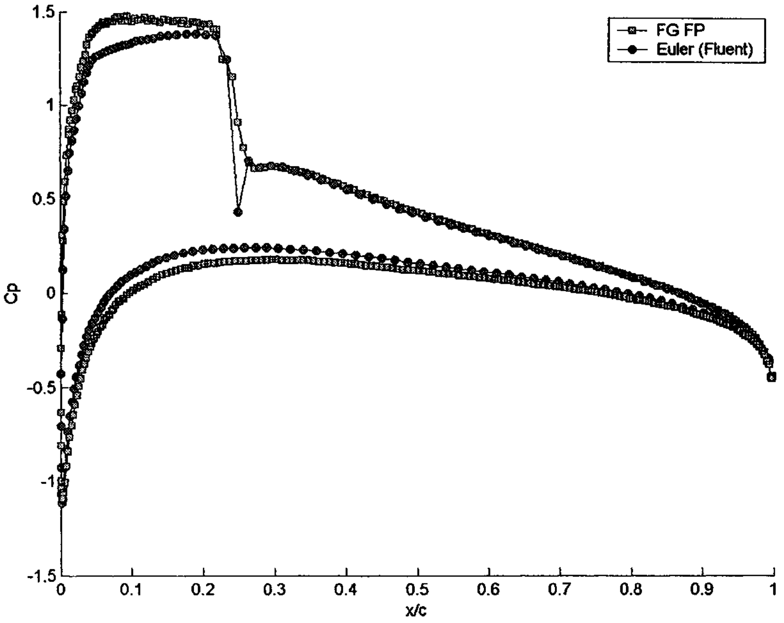

Figure 3.4: Pressure coefficient distribution at Mach 0.68, 3° incidence, FG transonic

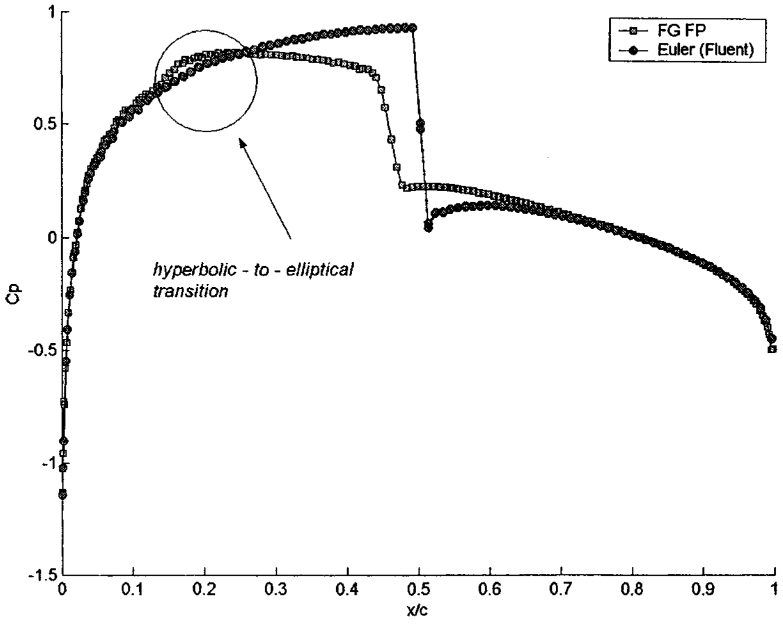

Figure 3.5: Pressure coefficient distribution at Mach 0.8, 0° incidence, FG transonic

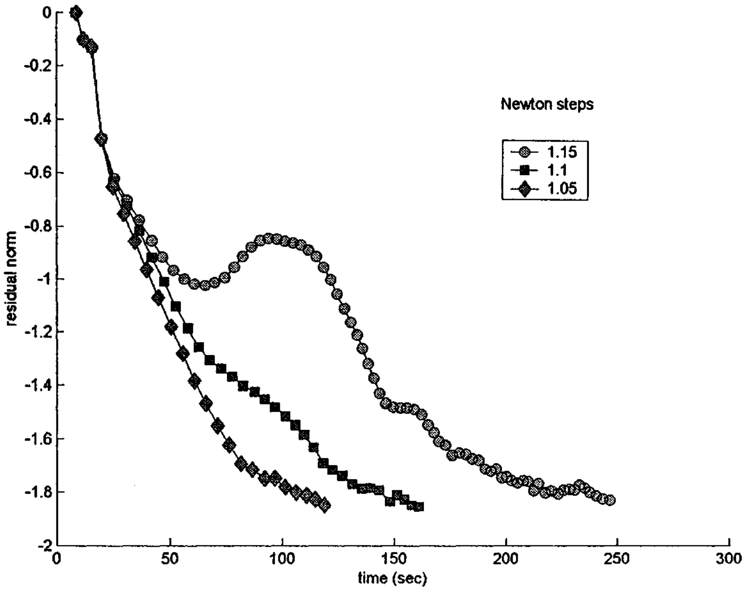

Figure 3.6: The effect of $M_{threshold}$ on convergence, in transonic FG approach

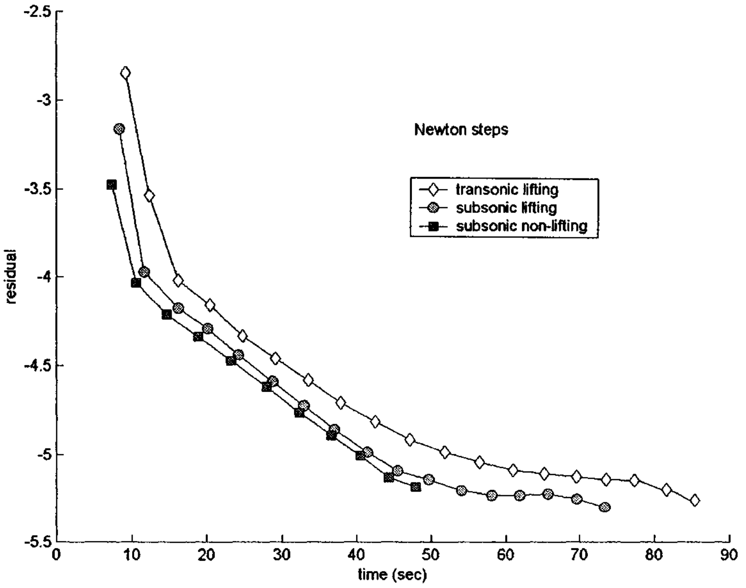

Figure 3.7: Compared convergence at Mach 0.5: non-lifting, subsonic lifting (3°) and transonic lifting (7°) flows

handled with a good convergence, due to the FG approach.

When compared to Euler (Fluent, implicit coupled solver) solutions (based on the 10e-3 Fluent default convergence criterion), the FP solutions (based on a 10e-2 convergence criterion) seem to converge roughly 15 times faster. However, it is reminded that the 3D Fluent solver has been used for a 2D problem. If modeled with a real 2D (planar) mesh, the Fluent Euler solution is less expensive than in the 3D approach. A reason for the fast FP convergence is the low problem dimension, with only one variable (potential value) / node (the approach is node centered), on a mesh having the ratio number-of-tetrahedra to number-of-nodes more than 4.

## 3.2 DLR F4 wing-body combination

A mesh consisting of 189268 nodes, 1063829 tetrahedrons and 51349 boundary faces has been used for full potential flow simulations, with a polar initializer mesh of 8 (chord) x 32 (half-span) panels, consisting of several wing segments. Figure 3.8 shows the unstructured mesh. A fine mesh envelope has been constructed on the leading edge of the wing and a wing-fuselage envelope separates the mesh close to the body from the remote coarse mesh. This helps obtaining a balanced and economical mesh with local refinement around critical areas. Figure 3.10 shows the polar mesh used in the initializer to model the wing.

Results presented for a high subsonic case at Mach 0.6 and lift coefficient $C_L = 0.5$, are compared to Bombardier FANSC results in [21], [20]. Caution in interpreting the results is however advisable, when comparing different approaches (experimental, Navier-Stokes and the current FP). In [20] a certain scatter in the angle of attack for producing a given lift coefficient at a given Mach number is noticeable,

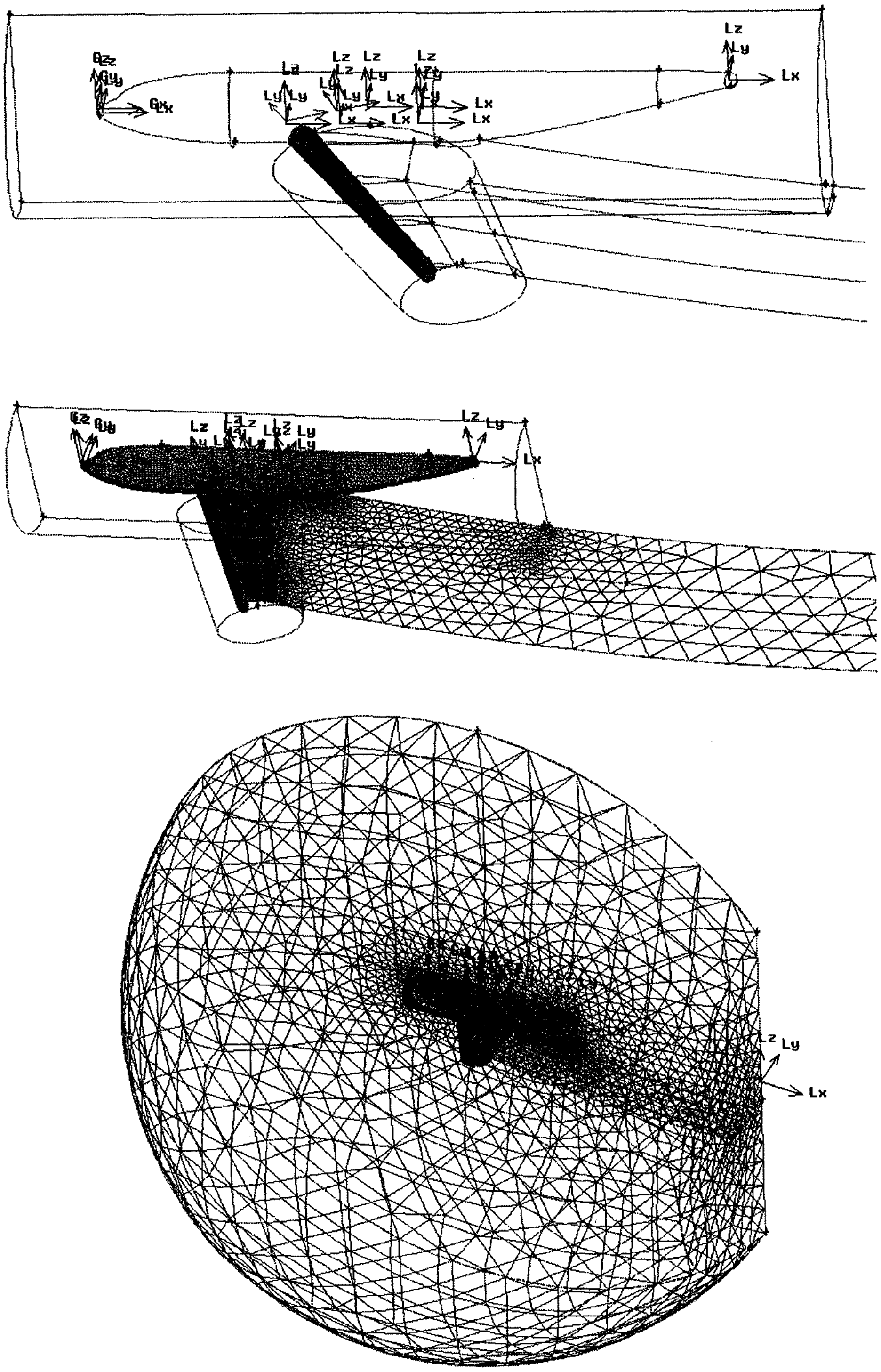

Figure 3.8: Stages in building the DLR F4 unstructured mesh

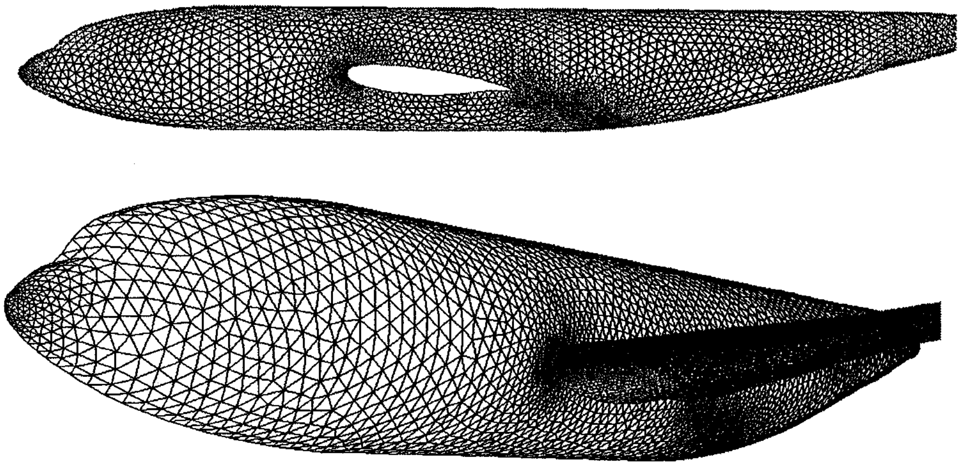

Figure 3.9: Surface mesh on the DLR F4 fuselage

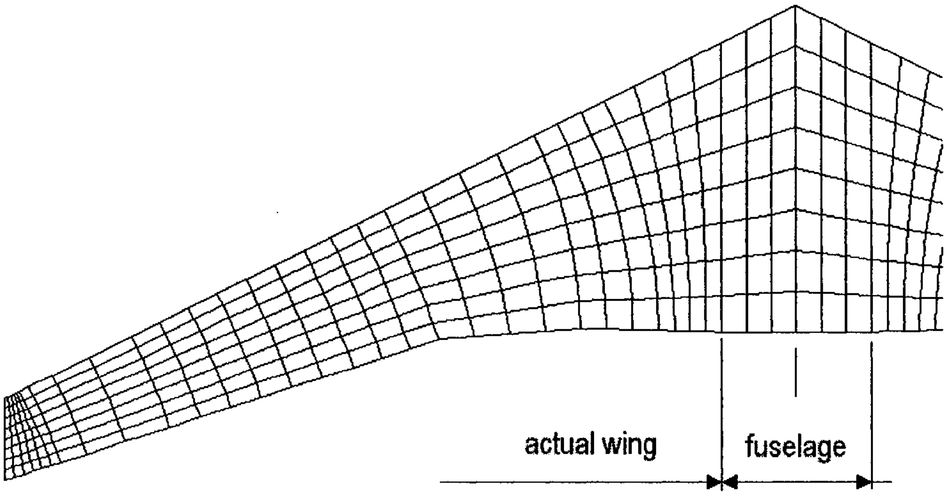

Figure 3.10: Polar mesh for the linear potential lift initializer

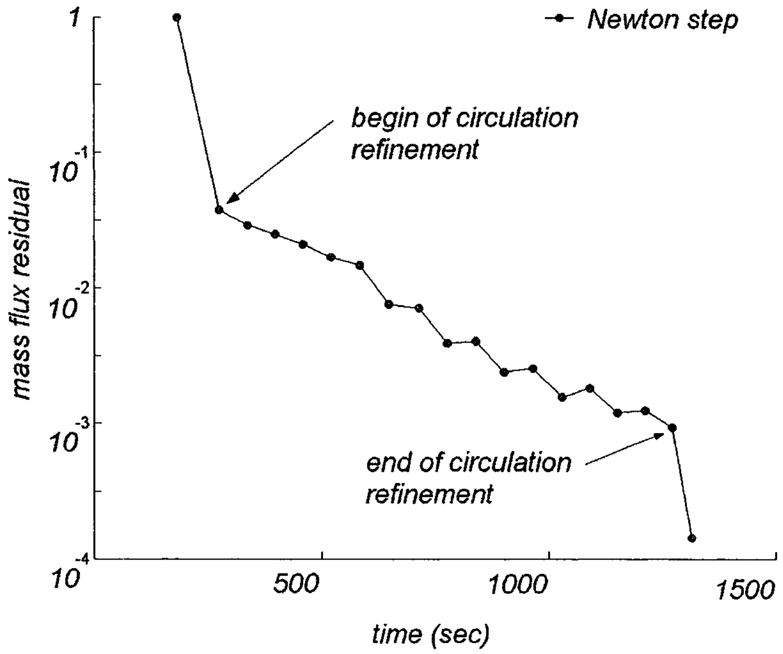

Figure 3.11: Convergence history

between different results. Wind tunnel experiments tend to request a higher angle of attack than Navier-Stokes simulations, possibly due to lower Reynolds numbers. Also to be noted, the full potential mesh used in our approach is built on the rigid DLR-F4 geometry, while experimental data come from a wind-tunnel flexible model which is prone to elastic deformations. Beyond this geometrical incertitude, a clear tendency is however noticeable, regarding inviscid flow simulations. The FP model, tends to overestimate lift when compared to viscous flows (experimental and Navier-Stokes simulations). The same tendency is noted in [23], which presents results from MGAERO Euler code. This fact is predictable and stems from the assumption of inviscid flow. Regarding the current FP approach, some corrections to calibrate the model to yield realistic results are suggested.

The trailing-edge-bisector orientation of the initial wake, which has been imposed, though correct from potential flow theoretical point of view, is not realistic. In real flows, there is neither a perfectly sharp trailing edge, nor a fixed stagnation point attached to such a sharp edge. Hence, the orientation of the trailing edge wake is dependent on the actual flow conditions, which could be modeled by giving the wake start an orientation which is relaxed from the trailing edge bisector position and/or allowing a certain recirculation-related flux through the trailing edge adjacent wake faces. When run without correction, for a fuselage angle of attack of -0.3°, the FP model produces a lift coefficient of 0.6468. Calibrated with a flux through the trailing edge wake faces corresponding to 7.3% of the free flow velocity (value which is mesh-fineness dependent), the model produces for the same angle of attack a lift coefficient of 0.5040, which is in good concordance with the 0.5 value predicted by Navier-Stokes results in [20]. Clearly, a better correction, both more physically meaningful and with no mesh sensitivity, is using the wake orientation. In this case the wake should be deflected upward by approximately 5° to produce similar results. The value of 5° is probably slightly overevaluated due to neglecting the wing tunnel model deformation, where the wing has an elastic twist going below -0.4° towards wingtip ([22]).

To drive at a conclusion, when modeling 'well behaved' flows, with no separation other than upper trailing edge, the wake deflection should generally fit into a range not higher than the trailing edge angle, starting from the bisector position upward, as pictured in figure 3.12. This rule of thumb can be completed with the recommendation of increasing deflection for lower Reynolds number of the real flow to be modeled and keeping it smaller for high Reynolds numbers. The considerations above allow adjusting a potential model to evaluate lift more accurately, but for a good drag prediction one would need more than that. A possible solution could be integrating the potential model with a boundary layer solver and correcting the pressure on upper trailing edge from a close-to-stagnation value as predicted by the

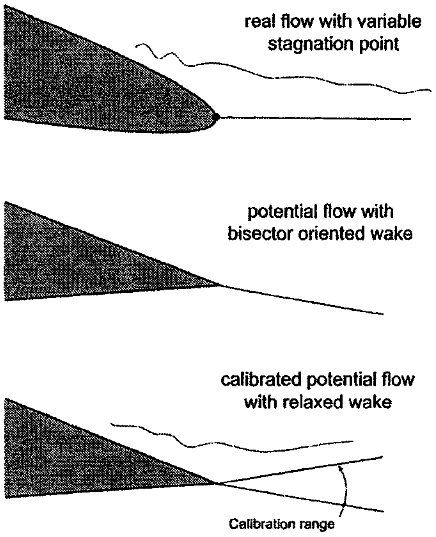
real flow with variable stagnation point
potential flow with bisector oriented wake
calibrated potential flow with relaxed wake
Calibration range

Figure 3.12: Calibration through wake deflection

potential flow to a lower value (closer to the freeflow static pressure), over the area past flow separation (if detected by the boundary layer solver). This model enhancement was beyond the scope of the thesis. The results in fig. 3.14 to 3.21 show the pressure distribution at four wing sections, with the calibration implemented by a recirculation flux through a bisector oriented starting wake.

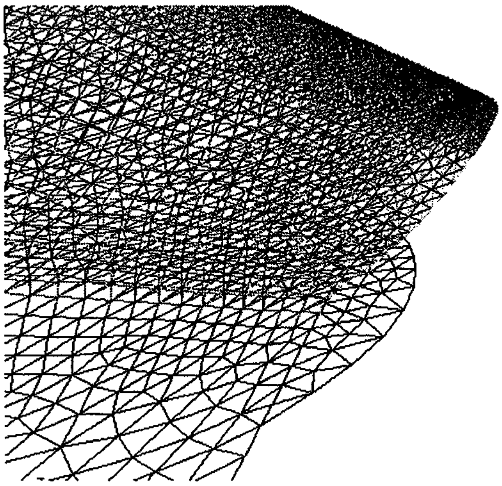

Figure 3.13: Wingtip with trailing edge wake and lateral wake - view from below

One more issue related to wingtip lifting flows will be shortly addressed. In a real (or simulated viscous) lifting flow, the wingtip vortex will never be concentrated in a dimensionless filament, but distributed in a cornet-shaped lateral flow separation. A potential model will not catch this by itself and hence will tend to produce very high speeds close to the wingtip-trailing edge intersection. This problem appears only in potential models which need to compute the whole flow field, e.g. the FP model presented here. Panel models use a discrete distribution of singularities, with velocities evaluated only in a discrete set of collocation points, hence the problem is circumvented. A whole-field potential flow simulation needs to tackle in some way this issue. Herein, for the FP solver, a lateral flow separation is modeled using a similar approach to the wake model. Basically, it is a lateral wake extension (Figure 3.13), which plays the role of laterally diffusing the potential difference appearing between upper and lower trailing edge and avoid excessively sharp gradients. Hence, a simple model enhancement will allow a more realistic flow field representation.

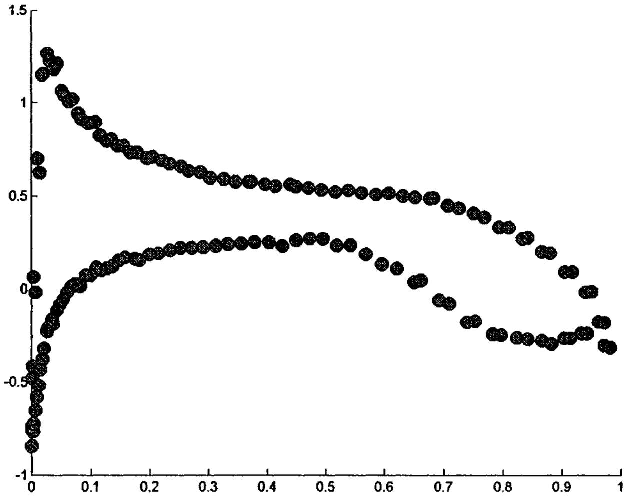

Figure 3.14: FP pressure distribution, wing section 0.185

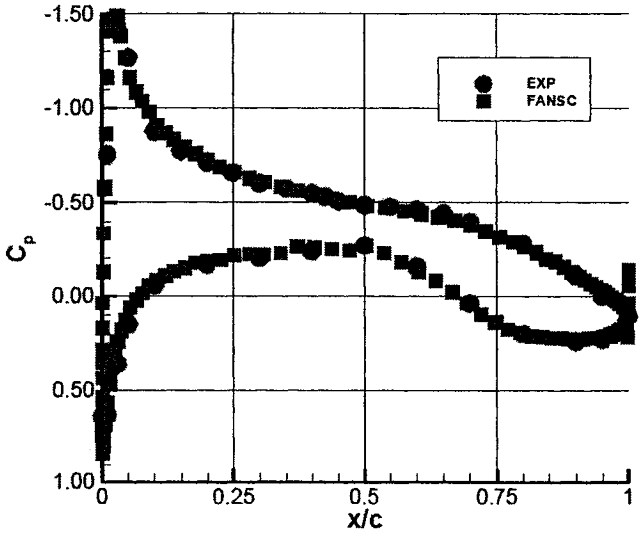

Figure 3.15: Bombardier FANSC [20], wing section 0.185

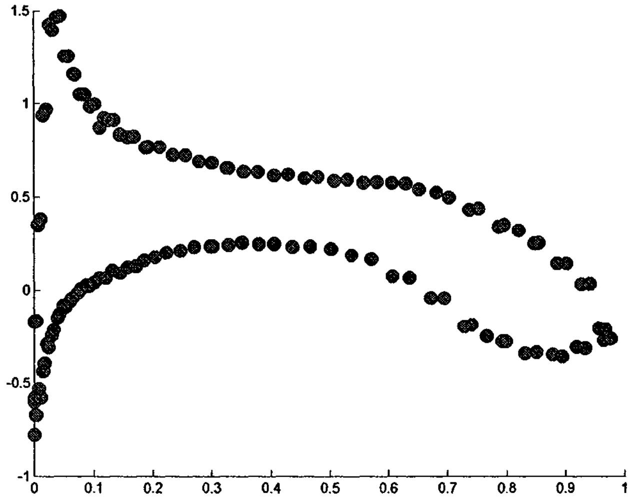

Figure 3.16: FP pressure distribution, wing section 0.331

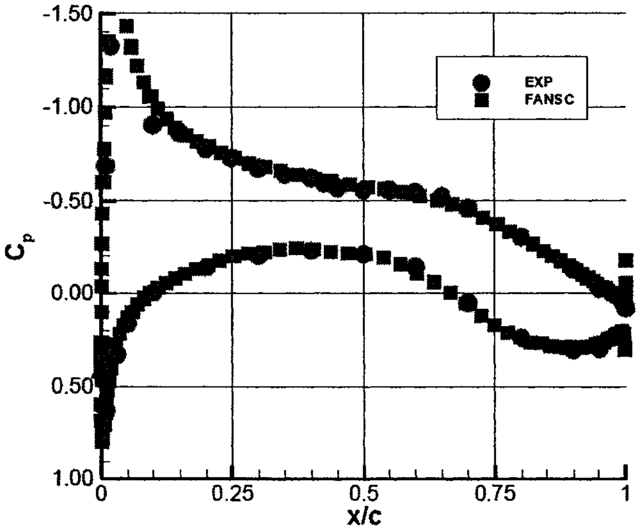

Figure 3.17: Bombardier FANSC [20], wing section 0.331

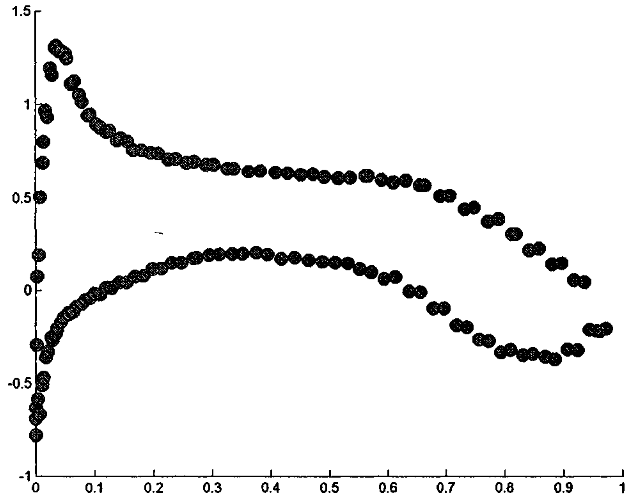

Figure 3.18: FP pressure distribution, wing section 0.512

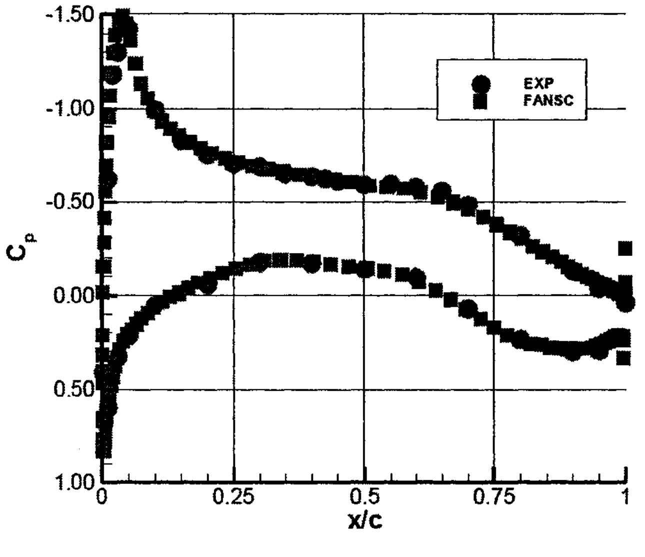

Figure 3.19: Bombardier FANSC [20], wing section 0.512

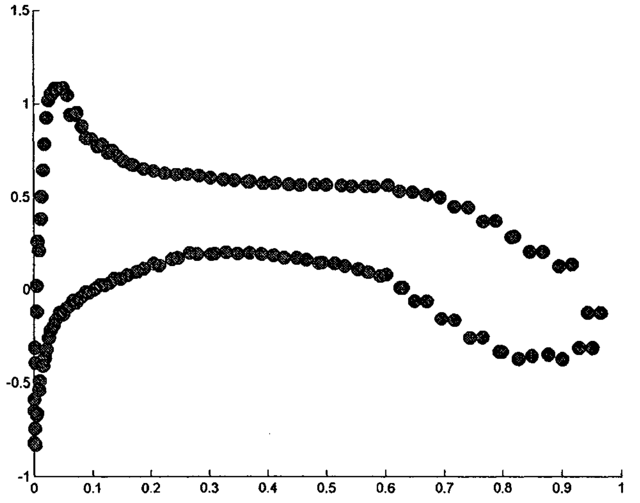

Figure 3.20: FP pressure distribution, wing section 0.844

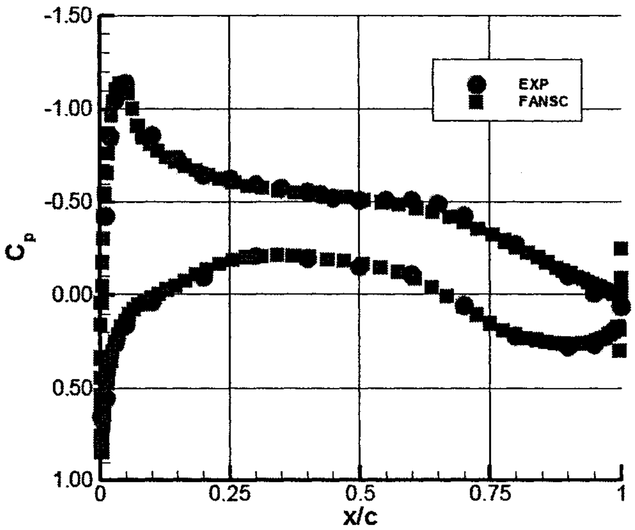

Figure 3.21: Bombardier FANSC [20], wing section 0.844

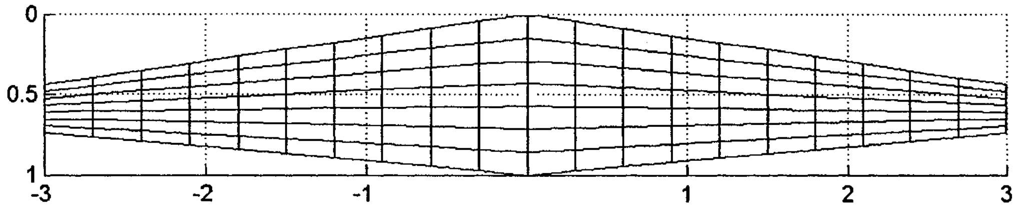

Figure 3.22: Planar straight wing (5° sweep at 25% chord) with Tornado mesh

## 3.3 Results on the polar linear potential model

The polar panel model used as an initializer for the FP solver, might also be used as a standalone tool, for any application modeling wings, where lumped vortex / panel models are usable. The effect of the polar meshing employed in the model, when tested against a classical panel implementation, is not expected to produce similar results but rather an evaluation of the differences in behavior is targeted here. Comparative results for the polar model vs. Tornado code, [35] are presented. Tornado is 'A Vortex Lattice MATLAB Implementation for Linear Aerodynamic Wing Applications', developed by Tomas Melin at Royal Institute of Technology (KTH) / Sweden as subject of a Master Thesis.

3 planar (untwisted, with no dihedral) wing cases (straight, forward sweep and aft sweep) have been considered.

While for straight wings the two models produce comparable [1] results, it seems that the polar model has an increased sensitivity for the different behavior of forward vs. aft swept wings, i.e. the better aerodynamic performance of forward sweep when compared to aft sweep is visible in the results. This comes from the contribution

[1] a slight difference in results even for straight wings might be expected because of compressibility effects, with the polar model having a Prandtl-Glauert correction implemented

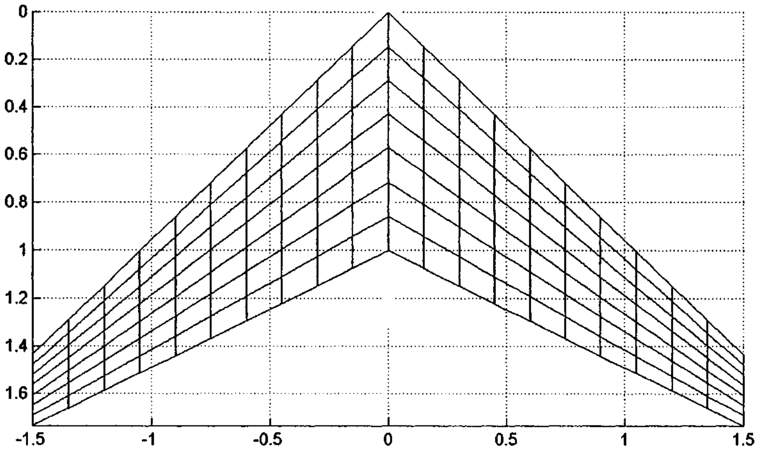

Figure 3.23: Planar aft swept wing (40° sweep at 25% chord) with Tornado mesh

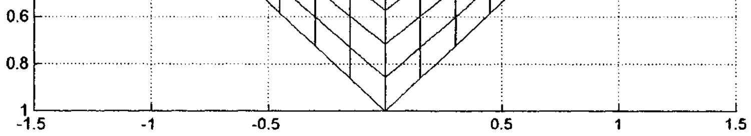

Figure 3.24: Planar forward swept wing (-30° sweep at 25% chord) with Tornado mesh

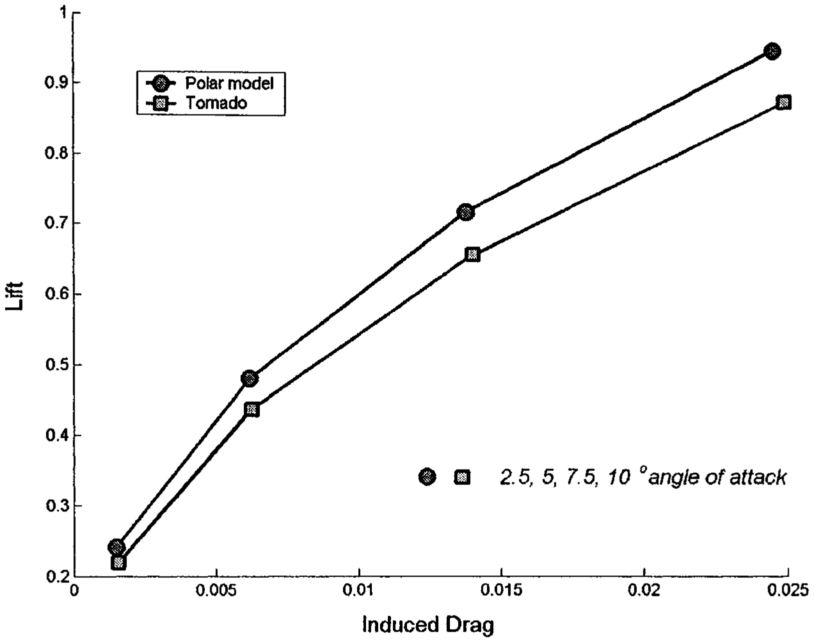

Figure 3.25: Straight wings compared

of lateral vortices, which, ahead of the trailing edge where they become part of the wake, are considered wing-bound (as they don't belong to a flow separation over the wing) and influence lift due to their misalignment to the free flow. As shown in section 2.2, aligning the lateral vortices to the unperturbed flow is judged as an arbitrary simplification which will introduce modeling errors for wings with higher sweep and/or taper. The orientation of lateral vortices, besides the effect on lift, has also an indirect effect on the induced drag: the wing self-induced downwash. A classical panel model will tend to overestimate the downwash of a forward swept wing and to underestimate it for an aft swept wing. Hence, both effects will rather contribute to hiding the influence of wing sweep.

Related to potential flows, some additional considerations on wakes, wingtip flow separation and the modeling thereof will be attempted here.

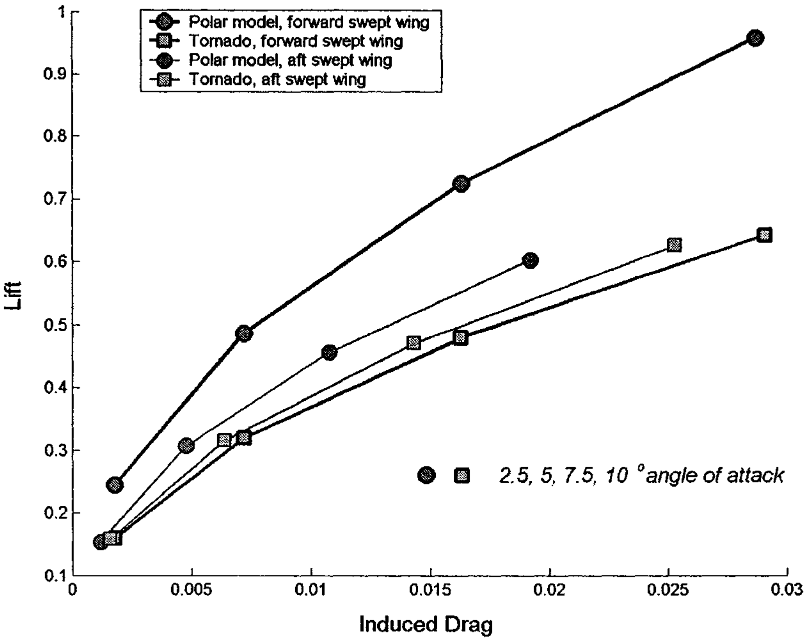

Figure 3.26: Swept wings compared

A common tendency is to view trailing edge generated wakes as basically different phenomena when compared to flow separations occurring on wingtip side (lateral wakes), but their common feature is they are shear sheets transporting vorticity downstream. Both cases will not necessarily present a pressure equality between the upper and lower side. Clear examples to support this affirmation are trailing edge wake deflections, or even wake roll up along trailing edge at high angles of attack on forward swept wings [26], or, for the lateral wingtip wake, the cornet-shaped evolution which testifies of the strong pressure difference between its faces. Some implementations of the Kutta condition for inviscid flow simulations are prone to the 'pressure equality condition' between upper and lower side of the trailing edge, which follows a 2D philosophy. This is actually accurate enough, for many wings of practical interest, with low or moderate sweep angles, where a pressure equality assumption won't damage the results significantly. However, this pressure equality condition shouldn't be seen as infallible in modeling wakes of wings with higher sweep angle. For instance, a forward swept wing with high sweep angle will keep a positive (lifting) pressure difference on its trailing edge. However, it is rather conventional to view the sharp edge at the rear of the wing as 'trailing edge' as its behavior is somewhere between a 2D-type trailing edge and a 3D wing lateral side, which actually sheds a vorticity sheet with a pressure difference between sides. If it is a trailing edge wake or a lateral flow separation is up to convention.

# Chapter 4

# Conclusions

At the beginning of this last chapter, a quotation from reference [17] seems to fit well: *Due to the advantages of unstructured grids, the concept was applied to the full potential equation. This is a new area for finite volume full potential solvers, and should provide a starting point for future research in this area.* The conclusion comes from a Ph.D. Thesis by R.E. Neel at Virginia Polytechnic Institute and State University, completed in 1997.

The current work has tried to integrate a finite volume, node centered full potential approach for tetrahedral unstructured meshes with a linear potential panel model initializer to produce a fast solver for inviscid external lifting flows. The initializer is capable of estimating compressible flow generated lift which is subsequently used as start approximation for the full potential solver.

A polar meshing technique for modeling lifting surfaces has been proposed and justified against classical models with freeflow oriented lateral vortices.

A full potential solver with an implicit mass flux residual reduction algorithm based on a GMRES linear solver and a Kutta condition implementation based on explicit recirculation residual reduction algorithm has been presented. The single variable full potential approach proves to be robust and achieves fast mass flux residual reduction, while the initialized lift seems to have a considerable effect on accelerating the airfoil/wing bound circulation convergence. As this, the solver is a very affordable tool, both in terms of speed and hardware requirements, for simulating subsonic flows.

When compared to its alternative, the Euler model, the full potential model needs roughly 5 times less computational resources, in terms of memory use and processor time. Depending on the specific implementation, this comparison might differ. Also, the behavior of the full potential model seems to be more robust in situations where an Euler code would produce less accurate results or even fail to converge, e.g. coarse or unbalanced meshes. Regarding specifically the current approach it isn't prone to incorrect outer boundary conditions (unawareness of lift inside the domain) which affects smaller sized domains in other solvers, as it gets the lift information from the initializer within a good accuracy and has the domain outer boundary 'expecting' the presence of a lifting flow.

Pertaining to future work, there is still room for improving the transonic performances of the solver. The standard full potential approach is usually quite slow when compared to the subsonic behavior. An approximate 'fictitious gas' approach has been suggested, which tries to speed up convergence by lowering the compressibility above a certain supersonic Mach threshold (e.g. Mach=1.05 - 1.2), hence creating a fictitious supersonic flow with subsonic characteristics (elliptical behavior), above the compressibility threshold. This algorithm is very efficient in terms of convergence but the accuracy of the solution is affected. Though not yet thoroughly tested, a combination of the two algorithms might prove efficient, i.e. starting the transonic solution with the fictitious gas algorithm, localizing the shock and subsequently switching to the standard full potential algorithm. This would make the standard full potential solution computationally more affordable.

Also a possible future work objective might be the implementation of a wake (and mesh) relaxation, during the full potential solution. This could be done upon monitoring the flux through all wake faces, in the same manner it was monitored through trailing edge wake faces for circulation solution, yet hereby not for purpose of updating the wing-bound circulation but for relaxing the wake as to align it with the flow, throughout its surface. This issue is actually of lesser importance, except for situations where the wake shed by a wing might run in the vicinity of another critical object (e.g. empennage). If this occurs, an accurate shape and position of the wake are desirable.

As the current approach uses no preconditioning in the linear step, a future review of this work might also consider this issue.

# Bibliography

[1] M. Paraschivoiu, X.-C. Cai, M. Sarkis, D.P. Young and D.E. Keyes, An implicit multimodel compressible flow formulation based on the full potential equation and the Euler equations. *AIAA Paper*, 99-0784, 1999.

[2] X.-C. Cai, M. Paraschivoiu, M. Sarkis, Explicit Multi-Model Compressible Flow Formulation Based on the Full-Potential Equation and the Euler Equations. *Proc. of the 11th Intl. Conference on Domain Decomposition Methods*, pp. 157-174, 1999.

[3] D. A. Caughey and A. Jameson, Numerical Calculation of Transonic Potential Flow about Wing-Body Combination, *AIAA Journal*, vol. 17, 1979, pp 175-180.

[4] T. L. Holst, Fast, Conservative Algorithm for Solving the Transonic Full-Potential Equation, *AIAA Journal*, vol. 18, 1980, pp 1431-1439.

[5] T. L. Holst, Implicit Algorithm for the Conservative Transonic Full-Potential Equation Using an Arbitrary Mesh, *AIAA Journal*, vol. 17, 1979, pp 1038-1045.

[6] T. L. Holst, Numerical Solution of Transonic Wing Flowfields, *AIAA Journal*, vol. 21, 1983, pp 863-871.

[7] M. Hafez, Artificial Compressibility Methods for Numerical Solutions of Transonic Full Potential Equation, *AIAA Journal*, vol. 17, 1979, pp 838-844.

[8] L.L. Erickson, S.M. Strande, A Theoretical Basis for Extending Surface-Paneling Methods to Transonic Flow, *AIAA Journal*, vol. 23, 1985, pp 1860-1867.

[9] W. H. Mason, On the Use of the Potential Flow Model for Aerodynamic Design at Transonic Speeds, *AIAA Paper*, 95-0741, 1995.

[10] D.J. Kinney, M.M. Hafez, P.A. Gelhausen, Validation of a New Unstructured Full Potential Formulation, *AIAA Paper*, 95-1765, 1995.

[11] M. Madson, S. Moyer, A. Cenko, TRANAIR Computations of the Flow About a Generic Wing/Pylon/Finned-Store Configuration, *AIAA Paper*, 94-0155, 1994.

[12] B. Uhl, J. Ostertag, G. Guidati, S. Wagner, Application of the Dual-Reciprocity Method to Three-Dimensional Compressible Flows Governed by the Full-Potential Equation, *AIAA Paper*, 99-0168, 1999.

[13] N.M. Chaderjian, J.L. Steger, The Numerical Simulation of Steady Transonic Rotational Flow Using a Dual Potential Formulation, *AIAA Paper*, 85-0368, 1985.

[14] J.E. Mercer, E.M. Murman, Application of Transonic Potential Calculations to Aircraft and Wind Tunnel Configurations, *AGARD Conference Proceedings No. 285 Subsonic/Transonic Configuration Aerodynamics*, May 1980.

[15] H. Sobieczky, Design of Advanced Technology Transonic Airfoils and Wings, *AGARD Conference Proceedings No. 285 Subsonic/Transonic Configuration Aerodynamics*, May 1980.

[16] G. Redeker, N. Schmidt, R. Mueller, Design and Experimental Verification of a Transonic Wing for a Transport Aircraft, *AGARD Conference Proceedings No. 285 Subsonic/Transonic Configuration Aerodynamics*, May 1980.

[17] R. E. Neel, *Advances in Computational Fluid Dynamics: Turbulent Separated Flows and Transonic Potential Flows*, Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, August 1997

[18] Tao Xu, *A study of an ILU Preconditioned Multi-Model Formulation for Compressible Flows*, M.A.Sc. Thesis, Graduate Department of Mechanical Engineering, University of Toronto, 2002

[19] Bao Quy Nguyen, *A Modified Potential Method for Transonic Flow Problems*, M.Eng. Thesis (Mechanical), Graduate Studies, Concordia University Montreal, 1986

[20] Proceedings of the 1st AIAA CFD Drag Prediction Workshop, *Preceding the 19th APA Conference*, Anaheim, CA June 9-10 2001, (DLR F4 wing fuselage combination), http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/Workshop1/workshop1.html

[21] E. Laurendeau, J. Boudreau, DLR-F4 Results using Bombardier Aerospace Full-Aircraft Navier-Stokes Code FANSC, *AIAA CFD Drag Prediction Workshop*, Anaheim, CA June 9-10 2001

[22] A Selection of Experimental Test Cases for the Validation of CFD Codes, AGARD-AR-303 vol II, August 1994

[23] G. Waller, Prediction of DLR-F4 Model Drag Using MGAERO Euler Code, AIAA Drag Prediction Workshop, 9-10 June, 2001, Anaheim, CA

[24] E. Oezger and I. Schell and D. Jacob, On the Structure and Attenuation of an Aircraft Wake, *Journal of Aircraft*, vol. 38, 2001, pp 878-887.

[25] J.E. Hackett, M.R. Evans, Vortex Wakes Behind High-Lift Wings, *Journal of Aircraft*, vol. 8, 1971, pp 334-340.

[26] C. Breitsamter, B. Laschka, Vortical Flowfield Structure at Forward Swept-Wing Configurations, *Journal of Aircraft*, vol. 38, No.2, March-April 2001, pp 193-207.

[27] C. Hirsch, Numerical Computation of Internal and External Flows, volume 1, John Wiley & Sons, New York, 1988.

[28] C. Hirsch, Numerical Computation of Internal and External Flows, volume 2, John Wiley & Sons, New York, 1990.

[29] J. Katz and A. Plotkin, Low-speed aerodynamics, Cambridge University Press, Cambridge, Second Edition, 2001.

[30] Y. Saad, M.H. Schultz, A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Comp.*, 7(1986), pp.856-869.

[31] D. A. Caughey, A. Jameson, Development of Computational Techniques for Transonic Flows: an Historical Perspective, Symposium Transonicum IV, Goettingen, September 2002

[32] F.T. Johnson, E.N. Tinoco, N.J. Yu, Thirty Years of Development and Application of CFD at Boeing Commercial Airplanes, Seattle, *AIAA Paper*, 2003-3439, 2003.

[33] A. Jameson, CFD for Aerodynamic Design and Optimization: Its Evolution over the Last Three Decades, *AIAA Paper*, 2003-3438, 2003.

[34] A. Jameson, The role of CFD in Preliminary Aerospace Design, FEDSM2003-45812, *Proceedings of FEDSM'03 4TH ASME JSME Joint Fluids Engineering Conference*, Honolulu, Hawaii USA, July 6-11, 2003.

[35] Tomas Melin, *A Vortex Lattice MATLAB Implementation for Linear Aerodynamic Wing Applications*, Master Thesis, Department of Aeronautics, Royal Institute of Technology (KTH), Sweden, December 2000

[36] Boeing, TRANAIR User's Manual (Version F00), *D6-57030*, REV D, 1994.

[37] Fluent Inc., Gambit User's Manual

# Appendix A

# Density upwinding

Due to the hyperbolic character in the supersonic region of the flow, a form of artificial compressibility is applied. Density upwinding is triggered above a 'cutoff' subsonic Mach number close to the sonic threshold. Hence, the standard value calculated from equation 1.3 will be affected by the values in neighboring upwind tetrahedra cells. Each cell having a gradient exceeding the cutoff Mach value will have a density bias proportional to the difference in density value with its upwind cells. The density bias will also depend on the flow orientation at each cell's level and on a switching function $\mu$, as follows:

$$\rho_j^{upwind} = \rho_j^{standard} + \mu \cdot \vec{V} \cdot \sum^{i} \mathbf{n}_i \left(\rho_j - \rho_i\right) \tag{A.1}$$

$$\mu = \nu_0 \max \left\{ 0, 1 - \frac{M_c^2}{M^2} \right\} M^2 \tag{A.2}$$

where $M$ is the Mach number of the current tetrahedra and $M_c$ is the Mach cutoff value. The parameter $\nu$ controls the amount of upwinding in the affected cells. $M_c$ and $\nu$ can be fixed or adjusted during the simulation. Typically, $M_c \geq 0.95$ while $\nu = 1 \ldots 3$. Depending on the specific problem studied, adding more dissipation might speed up convergence but will diffuse the shock. Choosing an inappropriate value for $\nu$ might get the algorithm instable. A way of increasing the dissipation could be by adopting for each cell the maximal value for the switching function $\mu$ from all immediate neighbors.

For the results presented in chapter 3 the values of 2.5 for $\nu$ and 0.95 for $M_c$ have been used.

# Appendix B

# General algorithm

```
import wing lift information (3D) or compute airfoil lift estimation (2D)
initialize solution. initialize Newton step count $n$=0.
repeat:
—> $n=n+1$
—> compute potential gradient
—> compute mass flux (residual) $\mathbf{R}^n$
—> if ($\|\mathbf{R}^n\|_2$ ≤circulation evaluation threshold).AND.(circulation not frozen*) then
—> —> trigger circulation evaluation and update algorithm
—> endif
—> compute Jacobian $J^n$
—> compute linear solution (GMRES)
—> update potential $\Phi^{n+1} = \Phi^n + \delta\Phi^n$
—> (for fictitious gas approach only - update $M_{threshold}$)
check overall convergence:
until ( ($\|\mathbf{R}^n\|_2$ ≤ mass flux convergence threshold).AND.(circulation frozen*) )
postprocessor
```

* circulation frozen upon convergence or to initial (linear potential) value

# Appendix C

# Circulation algorithm

The circulation residual reduction algorithm is presented below:

```
evaluate recirculation residual
evaluate recirculation residual descent
if (stagnation[1] or oscillatory behavior[2] detected) then
—> decrease explicit coefficient: $c^n < c^{n-1}$
update trailing edge potential jump (for 3D)
(eq. 2.29, 2.30, 2.31)
smoothen spanwise circulation distribution (for 3D)
update domain cut (for 2D) (equation 2.25)
update wake (for 3D)
(re)compute potential gradient (to account for updated circulation)
(re)compute mass flux (residual) $\mathbf{R}^n$
if (recirculation residual < convergence threshold) then freeze circulation
return control to main loop
```

[1] stagnation - meaning that the recirculation residual descends to slow or increases

[2] oscillatory behavior - (recirculation residual changes sign between steps) might be converging but perturbs unnecessarily the mass flux residual convergence