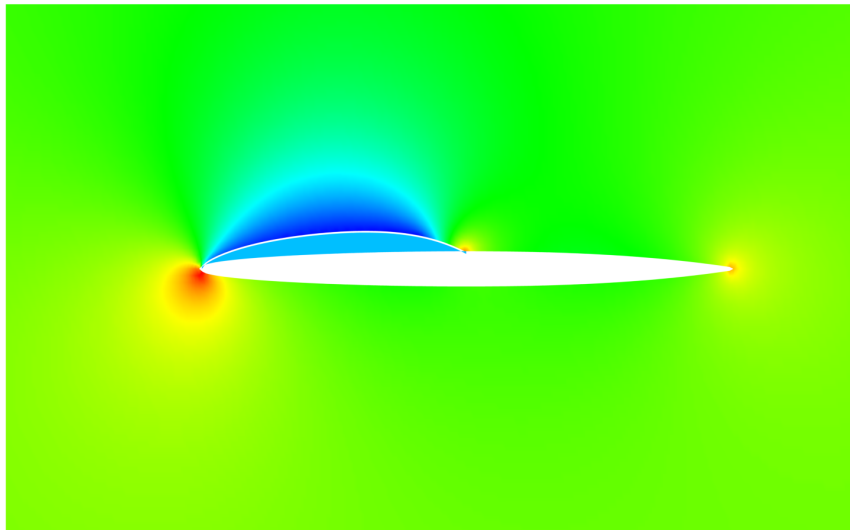NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF NAVAL ARCHITECTURE AND MARINE ENGINEERING
DIVISION OF SHIP AND MARINE HYDRODYNAMICS

Computation of Partially Cavitating flow around 2-D Hydrofoils using BEM

DIPLOMA THESIS

Antonios Kritikos

Supervisor: Kostas Belibassakis
Professor NTUA

Athens, February 2021

National Technical University of Athens
School of Naval Architecture and Marine Engineering
Division of Ship and Marine Hydrodynamics

Computation of Partially Cavitating flow around 2-D Hydrofoils using BEM

DIPLOMA THESIS

of

Antonios Kritikos

Supervisor: Kostas Belibassakis
Professor NTUA

Approved by the committee on:

(Sign)   *(Sign)*   *(Sign)*

....
Kostas Belibassakis   Georgios Triantafyllou   Georgios Papadakis
Professor NTUA   Professor NTUA   Assistant Professor NTUA

Athens, February 2021

(Signature)

....

Antonios Kritikos

Dipl. Naval Architect & Marine Engineer

Acknowledgements

Abstract

Cavitation in marine technology is a complex phenomenon and can hinder the performance and life of marine technology and the ship. Accurate prediction of cavitation and its control is an active endeavor within the engineering community. The purpose of this thesis is to implement a numerical modeling of partial cavitation on two-dimensional hydrofoils. It builds on established methods for this purpose and proposes slight modifications of the modeling to allow minimum user input. The modeling is based on a Boundary Element Method, programmed by the author in Fortran with the help of supervising professors. The method and its accompanied code are included in the text and can be used for extensions of the method. Results of the method agree well with the results by similar BEM methods. These results suggest that boundary element methods are still a very useful tool for modern naval architects and marine engineers, in order to design and operate more efficient propulsors.

Contents

Chapter 1

Introduction

Cavitation is important in connection with various engineering applications and devices. These include pumps, propellers, turbines and hydrofoil wings to name a few. Cavitation occurs when a fluid's pressure drops below the vapour pressure while it's temperature remains constant. This vapour pressure is dependent on the fluid's temperature. In the case of marine applications, this temperature is usually the temperature of the seawater, but may be higher or lower for fluids circulating within the piping systems of a ship(e.g fresh water). The cavity forming within a volume of water is not separated from the water, as a bubbly volume is created within the fluid volume, figure 1.1. The collapse of vapour bubbles, once reaching

Figure 1.1: Sheet cavitation, taken from https://cavity.caee.utexas.edu

areas of higher pressure, can cause erosion damage to machinery and radiate noise. The physics of bubble dynamics in an incompressible fluid have been investigated by many researchers and the most popular model today is the one by Rayleigh-Plesset:

$$R\frac{d^2R}{dt^2} + \frac{3}{2}\left(\frac{dR}{dt}\right)^2 + \frac{4\nu_L}{R}\frac{dR}{dt} + \frac{2\gamma}{\rho R} + \frac{p_0 - p_v}{\rho} = 0 \qquad (1.0.0.1)$$

where p_v is the vapour pressure, p_0 the pressure at infinity, ρ the density of the fluid, ν_L is the kinematic viscosity of the surrounding liquid, γ is the surface tension of the bubble-liquid interfaceand R the radius of the bubble.

Cavitation inception is best described by the nondimensional cavitation num-

ber:

$$\sigma = \frac{p_0 - p_v}{\frac{1}{2}\rho V_\infty}$$

This parameter can be used to achieve similarity between different cavitating conditions. Cavitation can be categorized into the following types:

1. Sheet cavitation: starting close to the leading edge of the blade, a fixed cavity attached to the foil suction side is formed and is generally stable in character

2. Supercavitation: the cavity formed extends beyond the chord length of the body

3. Cloud cavitation: a mist of bubbles is formed, usually as a result of strongly developed sheet cavities which are separated from the blade.

4. Bubble cavitation: distinct bubbles of different sizes are formed usually in the midchord region of a blade

5. Propeller specific cavitation: hub vortex which is produced by the combination of vortices of the blade root and tip vortex which is attached to the tip of the blade

Scope of the present work

The scope of the present thesis is to develop a boundary element method for simulating fixed-length partial cavitation around 2d hydrofoils. It is based on the work of Spyros Kinnas & Neal Fine in the early 1990s. In this respect, panel methods for fully wet flow are treated and presented first and the partial cavitation problem is built on these panel methods. Certain assumptions, in addition to those made by the founders of the method were made, as a pure reproduction of the method didn't produce the expected results, or required cumbersome calibration. Also, such a reproduction is of little use.

1.0.1   Literature review

A short review of the published literature is first presented, with a special focus on results related to marine applications
Attempts to numerically predict cavitation were made very early in the history of fluid dynamics. The hodograph technique introduced by Helmholtz, Kirchoff and Levi-Civita was the first non-linear theory applied to cavitating flows [1]. With this method, the cavity is treated as a *free* streamline with constant pressure. The hodograph technique was difficult to use with arbitrary body geometries, and thus its use was very limited to certain cases.

Linear method

Tulin introduced in 1953[2] the linearized cavity theory for slender bodies. The goal of this theory is to calculate the extent of the cavity by assuming a thin foil geometry, and imposing a distribution of singularities along the length of the body and cavity. The strength of singularities satisfies a set of boundary conditions derived from neglecting second-order effects. Results from subsequent non-linear theories showed the limitations of the linear theory. Linear cavity theory overpredicts the extent of cavitation and results in an increasing cavity thickness with increasing foil thickness, contrary to non-linear cavity theories.

Panel methods

The first successful non-linear method for modeling partially cavitating flows was developed by Uhlman in 1987[3]. He formulated a velocity-based boundary element method, by applying a set of linear boundary conditions for the fluid velocity on the hydrofoil and cavity. The foil and cavity are discretized into straight panels and each panel has a constant strength vortex, closely following the method by Hess & Smith [4]. Since the cavity shape is part of the unknowns, an iterative scheme is applied, in which the cavity shape is initially assumed and is then updated based on a flow-tangency boundary condition on the cavity. The cavity is assumed to reattach to the foil by a line perpendicular to the foil surface. The results obtained were significantly different from the results of the linear theory, correctly predicting the trend of the cavity thickness by increasing foil thickness. Lemmonier & Rowe[5] applied a similar method, in which the surface singularity elements bear source elements as well, apart from the vorticity distribution of Uhlman's method. They also imposed a smooth transition of the flow velocity from the cavity to the foil, contrary to the end-plate model of Uhlman. The system of equations resulting from the discretization of the boundary conditions is closed by requiring least-square minimization of the source and vortex distribution gradient. The method was applied to a hydrofoil, a rounded body and an axisymmetrical body with the cavity on the trailing wake.

Rowe & Blottiaux[6] carried out an evaluation of different partial cavitation closure models (closed, open with constant wake thickness and open with varying wake thickness) by applying the method of Lemmonier & Rowe and extending it to an open model. Open models showed satisfactory capturing of the pressure distribution in the cavity wake and agreement with experimental results for large angles of attack.

Peallat & Pellone[7] proposed a method in which a closure condition of the cavity is not necessary and the shape is obtained by an iterative process. They, too, used the method of Lemmonier & Rowe (Panel method minimization process) for calculating the flow characteristics and used results of the method for predicting the cavity shape and length for a given cavitation number. Results were in good agreement with experiments regarding the relationship between cavity length and cavitation number.

Boundary elements methods in which the boundary conditions are expressed in terms of the flow velocity potential were introduced by Kinnas & Fine[8], [9] and shortly later by Kim & Lee[10]. The method by Kinnas & Fine is the basis

of the present study and a detailed description of the method will be given in following chapters. The two methods differ in the cavity termination modeling and the cavity shape iterative scheme. Kim & Lee employed a termination wall model and the cavity shape is corrected after each iteration using a thin wing theory approximation. Extensions of the non-linear BEM to 3D cavitating flows, of hydrofoils or propellers were immediate. A short list of related papers is presented in Chapter 9 of [1].

Dang & Kuiper [11] extended the method of Kinnas & Fine with a reentrant jet model by adding an extra boundary condition on the jet cross section, requiring the normal velocity to be equal to the fluid velocity on the cavity boundary. Though the method is numerical, it showed that the re-entrant jet thickness is always a certain percentage of the maximum cavity thickness (around 8% to 10%). Results were compared and agreed well with the results of Uhlman's method.

Dang[12] developed a non-linear method for partial sheet cavity flows, in which the cavity surface evolution follows the Eulerian formulation and a Bernoulli integration of Euler's equation is used to update the potential at the cavity surface. The reentrant jet model is reused from previous works.

Viscosity effects play an important role in the cavity detachment point and the cavity wake. BEM of Kinnas& Fine were enhanced by coupling of the method with a boundary layer solver(Brewer & Kinnas [13]). The method is described shortly in [1]. In this way, viscosity effects can be captured with good accuracy and with less computation cost compared to a RANS solver.

Krisnaswamy et al (2000)[14] also applied the reentrant jet model in combination with the potential-based method of Kinnas & Fine. By first acquiring the fixed cavity length solution, the jet height is calculated by the continuity and momentum balance of the fluid volume upstream of the jet. This calculated height is then used to iteratively update the cavity shape. The jet boundary surface remains constant chordwise by introducing a source singularity to counteract the mass flux of the jet. They combined the above method with the method of Brewer & Kinnas to account for viscosity effects and achieve better agreement with experimental results.

A similar reentrant jet model, but without the source employed in Krisnaswamy et al, was applied by Uhlman[15] for simulating the cavity produced by a bluff axisymmetric body. Higher-order panel methods have been formulated for the problem by various of the above researchers, but not applied for comparison with low-order methods.

BEM has been succesful in predicting sheet cavitation of small angles of attack. However, the detachment and closure of the sheet cavity, which strongly influence the sheet cavity dynamics is not easily captured. Furthermore, it is difficult to extend these methods to more complex physical phenomena such as the shedding of the sheet cavity and tip vortex cavitation. For this reason finite element methods have gained popularity in recent years.

3D Methods(FEM, RANS)

Kubota et al. [16] developed a model for cavitating flows, which is based on Reynolds-averaged Navier Stokes equations (RANS). The cavity is treated as a

compressible viscous fluid with varying density. The model is formulated with a coupling of the RANS equations and Rayleigh's equation for bubble dynamics and solved by a finite-difference method. This enabled the study of unsteady effects in cavitation, mainly the shedding of cavitation clouds by separation of the attached cavity from the foil. This model rests on the assumption of hydrodynamic equilibrium and constant bubble number density.

This assumption was lifted by Chen and Heister(1996)[17], thus extending the model of Kubota et al for the case of variable bubble number density by deriving a new constitutive equation for bubble dynamics. The model was applied for axisymmetric headforms and captured unsteady effects and surface pressures satisfactorily.

Kunz et al[18] developed a cavitation model based on multi-phase Navier- Stokes, accounting and solving separately for the liquid and vapor as well as a non- condensable gas field (e.g atmospheric air entrained in the flow field of missiles), mass transfer between liquid and vapor and unsteady effects. The k-ε model was used for turbulence. The method was validated for axisymmetric bodies and a super cavitating fin on the interface of liquid and air.

M. Deshpande, J. Feng, and C. Merkle[19] formulated a solver based on Euler equations for simulating 2d cavity flows. The cavity is treated - similarly to potential methods- as a constant pressure volume. The motivation for this work was to apply a computationally efficient method for cavity flow in more complex geometry, such as cascade hydrofoils, or time-dependent problems.

O. Coutier-Delgosha et al simulated in 2003 [20] the unsteady behaviour of cavitating flows by applying the RANS equations for the mixture of liquid and vapor. A barotropic state law was employed as a cavitation model. Agreement with experimental results was achieved with a modified k-ε turbulence model and the results showed a strong dependence of the cavitating flow dynamic behavior on the turbulence model.

Merkle et al. (1998)[21] introduced an additional equation for the vapor (or liquid) volume fraction including source terms for evaporation and condensation (i.e. bubble growth and collapse). Kunz et al., 2000[18], Schnerr and Sauer, 2001[22] used similar techniques with different source terms.

Singhal et al developed in 2002[23] a new cavitation model, which they called Full Cavitation Model. The name is derived from the inclusion of the major first-order effects affecting the low pressure regions in cavitating flows. They adopted a modified form of the Rayleigh-Plesset equation for bubble dynamics, by adding two empirical constants derived from experimental data covering a wide range of flow conditions. The model has been implemented in modern commercial CFD codes(e.g. CFD-**ACE**+ and ANSYS Fluent) and has been validated against experimental results in various cases of interest(e.g. hydrofoils).

ANSYS Fluent also incorporates the cavitation model developed by Zwart et al[24] which adopts a first-order form of the Rayleigh-Plesset equation and separate empirical corrections for the condensation and vaporization phase change of the fluid, similar to the full cavitation model. Turbulence is treated with the k-ε or SST models.

A comparison of cavitation models is exposed in Coutier & Delgosha [25]. Main difference of these models is within the formulation of the mass fluxes between

vapor and liquid. In the process of benchmarking various cavitation models published earlier, the researchers observed main discrepancies of these models with experimental results in the wake of the cavity, where condensation takes place.

Within the framework of of the 2003 symposium on cavitation, workshop Models and CFD Tools for Computation of Cavitating Flows, Kunz et al[26] simulated the test hydrofoil geometry with unsteady RANS and detached Eddy simulation(DES). DES exhibits certain advantages in flows where separation occurs, as in the case of cavitating flow over hydrofoils. The main disadvantage of the method is the statistically unstationary time history of integrated coefficients (lift and drag). A comprehensive and analytical comparison between the cavitation models of Singhal et al, Zwart et al and Kunz et al was carried out by Morgut et al and published in 2011[27]. Results suggest that these models predict accurately sheet cavitation on hydrofoils if proper calibration of empirical coefficients is carried out. In this respect, they developed an optimization strategy for this calibration.

In 2011 researchers at Nanjing University of Science & Technology [28]applied a parallel algorithm implementing the one-equation LES turbulence model and Schnerr-Sauer cavitation model on a NACA hydrofoil.

A similar method was applied by E. Roohi et al in 2013[29] on a Clark-Y hydrofoil. They combined LES and VOF for the turbulence and flow modeling and either the Sauer or the Kunz model for cavitation. A qualitative agreement with predicted cloud cavity shapes from experiments was observed. When the LES model was substituted with the standard k-ε model results for cloud cavitation dynamics were inaccurate, due to failure of correct prediction of the re-entrant jet and vapour shedding. .

Ji et al applied in 2015[30] the LES model with Schnerr and Sauer cavitation model, resulting in good agreement with experimental results and observations regarding the vortex structure around the cavitating hydrofoil.

Recently (published in 2017) Chahine et al[31] used an Euler–Lagrange model, which accounts for the bubble dynamics in microscale and the large cavity dynamics in macroscale, as well as volumes of transition between these two scales. This model works without employing empirical mass exchange models that are non-physical and need to be calibrated. Thus, it is physically more consistent for representing the processes of sheet and cloud cavitation but it is computationally more expensive compared to other cavitation models.

Various attempts to improve and extend above methods have been made in recent years. Nohmi et al [32]deactivated the cavitation model at the estimated laminar boundary layer zone of the studied hydrofoil, assuming cavity growth on the transition to turbulent zone, in order to achieve more accurate lift coefficients at large angles of attack.

Direct Numerical Simulations of cavitating flows to develop better CFD models were carried out by Tryggvason et al[33] recently.

Xing et al[34] proposed a viscous vorticity equation model and compared it to commonly used RANS models. The method applied a 2d hydrofoil predicted larger cavity than RANS for low cavitation numbers, and thus needs improvements.

9

Experiments

Sheet and cloud cavitation were first studied extensively by Knapp(1955)[35]. One of the earliest experimental works in cavitation related to naval architecture was the work of T. Ito who investigated unsteady cavitation of Marine propellers in 1966[36]. A large number of experimental works have concentrated on understanding the mechanism and dynamics of cloud cavitation (Kawanami et al, 1997[37], de Lange & de Bruin, 1997 [38], T.M. Pham et al, 1999[39], Callenaere et al 2001[40], Laberteaux & Ceccio 2001[41]). These works identified the formation of a reentrant jet on the cavity closure that facilitates cloud cavitation. Leroux et al(2005)[42] found that the pressure wave resulting from the cloud collapse affects the frequency of cloud shedding. Coutier-Delgosha et al (2007)[43] attempted to shed light on the structure and flow inside the sheet cavity by applying x-ray densitometry and thus identified that modeling of the cavity at the macroscopic scale is needed to capture the phenomenon more accurately.

Dular et al[44] validated experimentally (and numerically) the hypothesis made by de Lange and de Bruin that the reentrant jet has a spanwise component if the closure line of the cavity is inclined. Foeth, van Terwisga & van Doorne (2008) [45] verified the above effect of the cavity topology.

Young et al (2013)[46] caried out experiments which showed that unsteady sheet/cloud cavitation substantially increases turbulent velocity fluctuations in the cavitating region around the foil and in the wake.

Ganesh et al, 2016[47], observed that in low cavitation numbers cloud shedding is caused by the propagation of a bubbly shock in the separated cavity flow. Experimental results by Pelz et al (2017)[48] carried out in a convergent–divergent nozzle showed the existence of a critical Reynolds number, dependent on the body roughness and cavitation number, beyond which sheet cavitation becomes unsteady and cloud cavitation is observed.

Propeller cavitation

In the present section we present a short outline of numerical methods for modeling cavitating flow in Marine propellers. The practical design of marine propellers is still carried out up to this day by use of the well known Burill diagram, which gives an estimate of propeller cavitation extent with minimum input from the designer. While it is useful at a preliminary design stage, this method is not accurate for modern applications. A large number of tools have been developed to predict propeller cavitation.

Stern & Vorus[49] developed a partially non-linear method in which the problem of a cavitating marine propeller is separated into a static part and a dynamic one. The static part accounts for the instantaneous cavity fixed relative to the propeller as it translates through the nonuniform wake field. The dynamic part accounts for the evolution of the cavity, and is solved by a lifting surface method. Unsteady motion of the propeller can be accounted for in the model.

Panel methods were readily extended to marine propellers. The works of the team led by Kinnas were successful in analysis of propeller sheet cavitation(Fine & Kinnas, 1992[50], Mueller & Kinnas 1999[51]) as well as design of cavitating propeller blades(Griffin & Kinnas 1998[52]). Further enhancements of the method

were investigated subsequently. Prediction of tip vortex cavitation with BEM needs special treatment as outlined in Lee & Kinnas 2004[53]. BEM analaysis of supercavitating and surface-piercing propellers was carried out by Young & Kinnas 2003[46]. Cavitating waterjet propulsor cavitation study was also carried out by Kinnas & Lee 2007[54]. Results in the above numerical methods were in good agreement with experiments.

Vaz[55] et al developed a nonlinear method for cavitating propellers in which remeshing implemented in previous panel methods is not needed and thus achieves fast convergence. The method was validated against analytical and experimental results. The method underpredicted the cavity extent for low propeller loading. The phd thesis of Vaz[56] includes a short listing of numerical works on cavitating propellers(up to its publication time, 2005)

Rhee et al 2005[57] were among the first who studied propeller cavitation with a RANS solver, by implementing one of the many cavitation models formulated earlier for finite element modeling of cavitating flows. The method agreed with experimental results, but issues were identified at low propeller loading and extent of tip vortex cavitation.

Sun & Kinnas[58] coupled the panel method by the Kinnas group with a viscous boundary layer analysis method based on the strip theory assumption. The method was applied for open, ducted, and water-jet propulsors and showed closer agreement with experimental results compared to the inviscid BEM.

Bensow & Bark 2010 [59] applied a LES of the cavitating flow on a propeller coupled with a two-phase cavitation model. The model predicted the mechanisms of practical importance which lead to undesired effects(e.g blade erosion and noise).

Ji et al[60] developed a Partially-Averaged Navier–Stokes (PANS) computational model, with the k-ε turbulence model and a mass transfer cavitation model to simulate cavitating flow around a marine propeller in a nonuniform wake. The PANS method captured the cavity volume pulsation better than the RANS method, but accuracy depends on the PANS model input coefficients.

The group led by Chahine extended their multiscale two phase model to simulate propeller cavitation[61].

Du & Kinnas[62] have presented a coupled numerical optimization technique with a panel method or a vortex vattice method to design cavitating propellers in nonuniform inflows. The resulting propeller design is more efficient than the lifting line-based optimal.

Yilmaz et al[63] used a LES model on commercial software STAR-CCM+ to predict tip vortex cavitation, while utilizing a mesh adaptive refinement strategy to include the effect of both the hull wake and the rudder. The method is still under development as desired accuracy has not been achieved. Shin & Andersen[64] applied a similar technique for the same problem, by employing DES and mesh refinement, but the model couldn't capture the full tip vortex cavity up to the rudder trailing edge.

The above finite element methods are only indicative of the significant amount of numerical works which were published in the years following the first attempt. A detailed review can be found in the proceedings of the 26th and 27th ITTC conference. In these works, a variety of FEM setups were investigated regarding governing equations(RANS, DES, RANS/BEM) turbulent modeling(k-ϵ, SST k-

ω, LES), cavitation modeling(models described previously), meshing(structured, unstructured, adaptive refinement). Without entering into details on the differences and pros/cons of all these alternative, we can sum up that RANS and coupled RANS/BEM methods capture well sheet cavitation of marine propellers. Regarding turbulent modeling, most researchers prefer the SST k-ω model. For more complex cavitation phenomena, such as tip vortex cavitation, cloud and bubble cavitation, more advanced methods are needed (e.g LES, DES, Eulerian/Lagrangian).

Control of propeller cavitation and prediction of its effects

Building on the well established methods of previous years, a shift in attention has been observed in recent years related to measures to control cavitation(Park et al[65]), improve propulsor cavitation performance(Kehr et al[66]), accurate prediction of unwanted effects, particularly noise radiation(Ando et al[67], Sakamoto et al[68], Fu & Li[69], Belibassakis & Politis[70], Bosschers[71], Lidtke et al[72]), hull pressure fluctuation(Su et al[73], Hur et al[74], Shiraishi et al[75], Kim & Kinnas[76], Kao[77], Kanemaru et al[78],Berger et al[79]) and erosion potential(Jin-Keun Choi et al 2015[80], Melissaris et al[81]).

Chapter 2

Two-dimensional Potential Flow

Boundary element methods although computational in their nature have a solid theoretical foundation. This foundation was laid down in the works of early researchers including Euler, Lagrange and Laplace to name some of the most important. This foundation consists of the dynamics of inviscid fluids. All fluids have a viscosity, so inviscid fluids are only ideal and a simplification of the reality. It was this assumption (or the ignorance of viscosity) that facilitated the first general equations describing fluid dynamics, that is Euler equations.

2.1 Fundamental equations of inviscid fluid dynamics

If one assumes that the external force field consists only of the gravitational force ρg which acts vertically downward, the Euler equations can then be written:

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{1}{\rho}F_i, \quad i = 1, 2, 3 \tag{2.1.0.1}$$

in which:
$\mathbf{V} = (u_1, u_2, u_3)$ is the fluid velocity,
p is the local pressure,
t is time, (x_1, x_2, x_3) is the coordinate system
ρ is the fluid density and
$\mathbf{F} = (F_1, F_2, F_3) = (0, -\rho g, 0)$ is the external force field due to gravity.
Use of indicial notation is being made in the above equation.
The fluid must also satisfy the continuity equation:

$$\text{div}\mathbf{V} = \sum \partial u_i/\partial x_i = 0 \tag{2.1.0.2}$$

Gravity is a conservative force and since our fluid is ideal, one easily becomes predisposed that the velocity vector field is a conservative field. This is proven using Kelvin's theorem of the conservation of circulation. First, the circulation must be defined as the intergral around any closed contour C in the fluid.

$$\Gamma = \int_C u_i dx_i$$

The theorem states that if conservative forces act on a moving ideal fluid, the circulation is constant for any closed material contour moving with fluid. A proof of Kelvin's Theorem can be found in [82] . If we, then, assume that the fluid started from rest, the circulation will remain constantly zero. Thus, so far we have established that

$$\Gamma = \int_C \mathbf{V} d\mathbf{x} = 0 \tag{2.1.0.3}$$

A vector field which satisfies this equation is called a conservative vector field and vector calclulus gives us important tools for such fields. According to[Marsden & Tromba], we will have:

i) $\nabla \times \mathbf{V} = 0$.

ii) $\mathbf{V}$ is the gradient of some scalar function ϕ; that is $\mathbf{V} = \nabla\phi$.

The first of these properties is the vorticity of the fluid, showing that the flow will be irrotational. The second property defines the velocity potential which is a scalar. Using the velocity potential in the continuity equation, it follows that:

$$\sum \frac{\partial}{\partial x_i} \frac{\partial \phi}{\partial x_i} = \sum \frac{\partial^2 \phi}{\partial x_i \partial x_i} = \nabla^2 \phi = 0 \tag{2.1.0.4}$$

This is the governing partial differential equation for the velocity potential of the fluid, and the well known Laplace equation.

2.2 Integral equation of potential flow

So far we have outlined the basic equations governing general potential flows, in the form of partial differential equations. Boundary conditions must be imposed on these equations to acquire a solution. Postponing the discussion of specific boundary conditions to subsequent chapters, we can, nevertheless, define the geometric boundaries of the problem. We are interested in two dimensional flows, so our fluid volume is first reduced to an area. The equations in the preceeding section remain valid with the only difference being the omission of the third component, where applicable. More specifically, if we denote by V the entire fluid area, its geometric boundary consists of:

- a body of outer surface S_B immersed in the fluid.

- the outer boundary far from the immersed body S_∞.
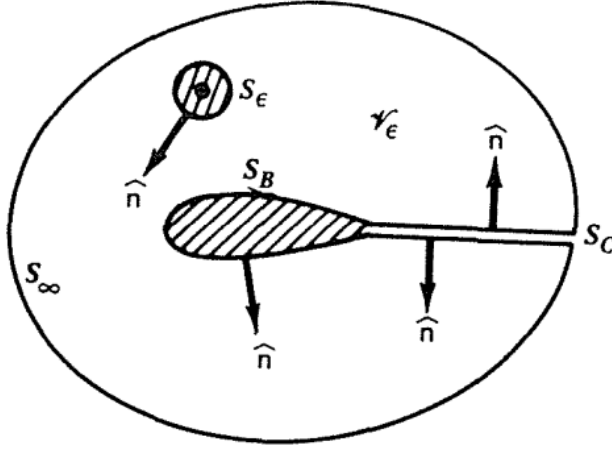
- the wake of the body S_W.

Figure 2.1: Geometry of the problem[83]

The above figure illustrates the notation used for the geometrical boundaries. The unit vector normal to the boundaries denoted by $\mathbf{n}$ is directed exterior to the fluid area.

The divergence theorem states that for a vector $\mathbf{q}$ within an area V of boundary S:

$$\int_S \mathbf{n} \cdot \mathbf{q} \, dS = \int_V \nabla \cdot \mathbf{q} \, dV$$

We can apply this theorem to the vector $\phi \nabla \phi_S - \phi_S \nabla \phi$, where ϕ is the velocity potential and $\phi_S = \frac{1}{2\pi} \ln r$, in which r is the distance of a random point P in the field from an assumed reference :

$$\int_S \hat{\mathbf{n}} \cdot (\phi \nabla \phi_s - \phi_s \nabla \phi) \, dS = \int_V \left(\phi \nabla^2 \phi_s - \phi_s \nabla^2 \phi \right) \, dV. \tag{2.2.0.1}$$

The function ϕ_S is a solution of the Laplace equation, based on the following calculation made in polar coordinates:

$$\Delta \phi_S = \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial \phi_S}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 \phi_S}{\partial \theta^2} = \frac{\partial^2 \phi_S}{\partial r^2} + \frac{1}{r} \frac{\partial \phi_S}{\partial r} + \frac{1}{r^2} \frac{\partial^2 \phi_S}{\partial \theta^2} = \frac{1}{2\pi} \left(-\frac{1}{r^2} + \frac{1}{r} \cdot \frac{1}{r} \right) = 0$$

Since ϕ is the velocity potential, it also satisfies the Laplace equation, so the right hand side integral in (2.2.0.1) vanishes and we have:

$$\int_S \hat{\mathbf{n}} \cdot (\phi \nabla \phi_s - \phi_s \nabla \phi) \, dS = 0. \tag{2.2.0.2}$$

When the point P is within the region, the above integral becomes singular as $r \to 0$ and the region close to the point must be excluded from the calculation. This is done by adding to the region's boundary a circle of radius ϵ centered on point P. In this way, point P is outside the region and equation 2.2.0.2 remains valid for the remaining region. We can thus write:

$$\int_{S+\text{circle } \epsilon} \hat{\mathbf{n}} \cdot (\phi \nabla \phi_s - \phi_s \nabla \phi) \, dS = 0. \tag{2.2.0.3}$$

15

To calculate the integral over the circle we introduce a polar coordinate system at P and since the vector $\mathbf{n}$ points inside the small circle, $\mathbf{n} = -\mathbf{e}_r$, $\mathbf{n} \cdot \nabla \phi = -\partial \phi / \partial r$ and $\mathbf{n} \cdot \nabla \phi_S = -2\pi/r$. Equation 2.2.0.3 now becomes

$$-\int_{\text{circle }\epsilon} \left(\phi \frac{1}{r} - \ln r \frac{\partial \phi}{\partial r}\right) dS + \int_S (\phi \nabla \ln r - \ln r \nabla \phi) \cdot \mathbf{n} dS = 0 \tag{2.2.0.4}$$

At the limit $\epsilon \to 0$, ϕ will approach ϕ_P and $\frac{\partial \phi}{\partial r} \to 0$. The circumference of the small circle around P is $2\pi\epsilon$ and 2.2.0.4 is equivalent to:

$$2\pi \phi_P = \int_S (\phi \nabla \ln r - \ln r \nabla \phi) \cdot \mathbf{n} \ dS$$

or

$$\phi_P = \int_S (\phi \nabla \phi_S - \phi_S \nabla \phi) \cdot \mathbf{n} \ dS \tag{2.2.0.5}$$

With this formula, one can calculate the solution of Laplace's equation ϕ at any point P in V, in terms of the values of ϕ and $\nabla \phi \cdot \mathbf{n}$ on the boundary of V.

The scalar function $\phi_S = \frac{1}{2\pi} \ln r$ is the velocity potential of a point source element. The function $\mathbf{n} \cdot \nabla \phi_S$ is the velocity potential of a doublet. At this point we will conclude the section with a variation of the above formula that will be used in the following chapters. We can separate the boundary far from the body, S_∞, and if the flow at infinity is approximated by a constant stream with speed V_∞ and with angle of attack α relative to the body reference axis, then on S_∞,

$$\phi \approx V_\infty (x \cos \alpha + y \sin \alpha)$$

When the point P is far from the body, the effect of the body and its wake are negligible, so we can write

$$\phi_P = V_\infty (x_P \cos \alpha + y_P \sin \alpha) + \int_{S_B + S_W} (\phi \nabla \phi_S - \phi_S \nabla \phi) \cdot \mathbf{n} \ dS \tag{2.2.0.6}$$

2.3   Bernoulli's equation

An explicit formula for the pressure can be output from Euler's equations. First we assume the flow is steady in relation to time. In this case Euler's become:(using indicial notation, ref [82])

$$u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial}{\partial x_i} (p/\rho + gx_2)$$

We then multiply both sides by the velocity components u_i and summing over i gives

$$u_i u_j \frac{\partial u_i}{\partial x_j} = -u_i \frac{\partial}{\partial x_i} (p/\rho + gx_2)$$

The left hand side of the equation can be written as

$$\frac{1}{2}u_j\frac{\partial}{\partial x_j}u_i u_i = \frac{1}{2}u_i\frac{\partial}{\partial x_i}u_j u_j,$$

where we interchanged the two indices to agree with the partial derivative of the right hand side. Returning to the modified Euler's equations we get

$$u_i\frac{\partial}{\partial x_i}\left(p/\rho + \frac{1}{2}u_j u_j + gx_2\right) = 0 \tag{2.3.0.1}$$

The differential operator $u_i\frac{\partial}{\partial x_i}$ is the well known material derivative in steady flow. The quantity in the parentheses is constant along any streamline:

$$p + \frac{1}{2}\rho V^2 + \rho g x_2 = const. \tag{2.3.0.2}$$

This is the Bernoulli equation for steady flow. This formula is very useful to calclulate the pressure coefficient of a point on the flow field. Suppose we take the second point of the streamline in infinity and neglecting the hydrostatic pressure component $\rho g x_2$ we will have:

$$p + \frac{1}{2}\rho V^2 = p_\infty + \frac{1}{2}\rho V_\infty^2 \implies c_p = \frac{p - p_\infty}{\frac{1}{2}\rho V_\infty^2} = 1 - \frac{V^2}{V_\infty^2} \tag{2.3.0.3}$$

Furthermore, the distribution of the pressure coefficient on the body boundary facilitates the calculation of the lifting force and its moment(let $\mathbf{r}$ be the distance of the element ds from the leading edge of the body):

$$\begin{aligned}
\mathbf{F} &= \int_S p \cdot \mathbf{n}\, ds = \frac{1}{2}\rho V_\infty^2 \int_S c_p \cdot \mathbf{n}\, ds + p_\infty \int_S \mathbf{n}\, ds = \frac{1}{2}\rho V_\infty^2 \int_S c_p \cdot \mathbf{n}\, ds \\
\mathbf{M} &= \int_S p \cdot (\mathbf{r} \times \mathbf{n})\, ds = \frac{1}{2}\rho V_\infty^2 \int_S c_p \cdot (\mathbf{r} \times \mathbf{n})\, ds
\end{aligned} \tag{2.3.0.4}$$

Chapter 3

Non-cavitating flow BEM

In the previous chapter, the foundations of the boundary element method were outlined without mention to specific applications. In this chapter, we make use of the general equations of the previous chapter to calculate the flow characteristics around hydrofoils when they are fully wet(i.e. without cavitation).

3.0.1 Mathematical formulation of the non-cavitating flow

The purpose of this subsection is to restate the governing equations for two dimensional potential flow when applied to non-cavitating flow.

Boundary conditions

Different types of fluid motions result according to the conditions imposed on fluid region boundaries. These conditions are physical in their nature, so, part of the modeling is to translate them to mathematical relations. The studied bodies have a fixed and rigid boundary. This means that there will be no flow through this boundary. This is the no entrance or flow tangency boundary condition. Adhering to notation of the previous chapter this boundary condition means that on S_B:

$$\mathbf{n}\mathbf{V} = \mathbf{n}\nabla\phi = 0. \tag{3.0.1.1}$$

This is a kinematic boundary condition and although very simple provides the basis for solving numerically the boundary problem. The second boundary condition is related to the statement made at the end of section 2.2. That is we require from the solution ϕ to approximate the flow at infinity velocity potential. This is practically equivalent to require that the perturbation potential diminish far from the body. The perturbation potential is the part of the total potential ϕ due to the disturbance the body imposes on the fluid. If we denote it with Φ, then we require that:

$$\lim_{r\to\infty}\nabla\Phi = 0 \tag{3.0.1.2}$$

If a solution to the problem is constructed by using a distribution of singularities like sources and doublets on the body boundary, this condition is automatically met, so it doesn't add any further restriction.

Finally, the Kutta condition is applied on the body's trailing edge. This condition requires that the flow at the trailing edge is smooth and tangent to the bisector of the trailing edge angle. This will occur if the velocity potential is finite at the trailing edge. The physical reasoning of the Kutta condition is that when the body reaches the steady state condition, any vortex generated during initial accelerating(usually called free vortex) will be dissipated in viscous diffusion. So the flow will be irrotational in the surrounding fluid, but the body will have a net circulation about it(what is usually called bound vortex). As noted in the previous chapter, the wake of the body is modeled with a separate boundary S_C. This is necessary in order to include a discontinuity of the potential ϕ across this boundary. This discontinuity is related to the bound vortex explained previously. A consequence of the Kutta condition is the requirement that the wake is modeled with a doublet of constant strength in its entire length. This strength is just the so called *potential jump*, which is the difference in doublet strength of the two adjacent panels on the two sides of the wake. The details of this statement can be found in [83], and omitted here for brevity.

Integral equation for non-cavitating flow

As shown in the previous chapter there is an integral equation for the velocity potential function on the surface of a body within an area of fluid:

$$\phi_P = V_\infty(x_P \cos \alpha + y_P \sin \alpha) + \int_{\mathcal{S}_B + \mathcal{S}_C} [(\mathbf{n} \cdot \nabla \phi)\phi_s - \phi(\mathbf{n} \cdot \nabla \phi_s)]dS \qquad (3.0.1.3)$$

Now, the surface integral portion of this equation can be simplified further by making a few observations. Firstly, the flow tangency boundary condition establishes that, on the surface of the body $S_B, \mathbf{n}\mathbf{V} = \mathbf{n}\nabla\phi = 0$. Consequently, the first term of the surface integral will be zero for the portion of the integral that includes S_B.

Furthermore, the portion of the first term of the integral that includes $S_\mathcal{C}$ can also be eliminated using the following rationale. The velocity component tangential to $S_\mathcal{C}$ can be discontinuous in certain situations (a trailing vortex sheet, for example). However, the component of velocity normal to $S_\mathcal{C}$ is continuous. Since the unit normals on both sides of $S_\mathcal{C}$ have directly opposite directions, their contributions cancel out exactly, thus eliminating the whole first part of the integral in 3.0.1.3 leaving us with

$$\phi_P = V_\infty(x_P \cos \alpha + y_P \sin \alpha) - \int_{\mathcal{S}_B + \mathcal{S}_C} \phi(\mathbf{n} \cdot \nabla \phi_s)dS \qquad (3.0.1.4)$$

Now, the potential ϕ itself can be discontinuous across $S_\mathcal{C}$ (and indeed it is in all cases of lifting flows). Using the concept of the *potential jump* across $S_\mathcal{C}$ defined as:

$$\triangle\phi = \phi^+ - \phi^-$$

we can rewrite our expression for ϕ_P as follows

$$\phi_P = V_\infty(x_P \cos \alpha + y_P \sin \alpha) - \int_{\mathcal{S}_B} \phi(\mathbf{n} \cdot \nabla \phi_s)dS - \int_{\mathcal{S}_C} \triangle\phi(\mathbf{n} \cdot \nabla \phi_s)dS$$

As shown in the Figure below, ϕ^+ Is the value of the potential on the side of $S_{\mathcal{C}}$ for which $\mathbf{n}$ goes into the fluid, while ϕ^- Is the value of the potential on the other side. It is, of course, understood that the integral is now carried out over a single side of $S_{\mathcal{C}}$.
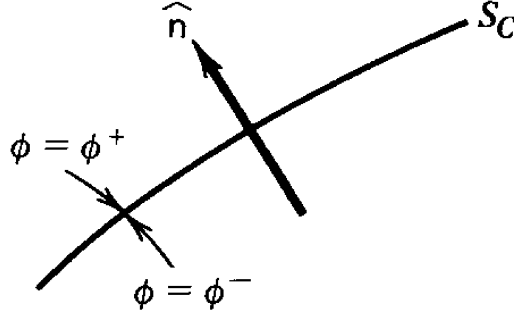
Figure 3.1: Potential Jump across connecting surface S_c

Finally we are left with an equation that can be used in practice for a variety of potential-based panel methods:

$$\phi_P = V_\infty(x_P \cos\alpha + y_P \sin\alpha) - \int_{\mathcal{S}_B} \phi(\mathbf{n} \cdot \nabla\phi_s)dS - \triangle\phi \int_{\mathcal{S}_C} \mathbf{n} \cdot \nabla\phi_s dS \quad (3.0.1.5)$$

3.0.2 Numerical formulation & results

Notice, once more, that equation 3.0.1.5 is an integral equation for ϕ on S_B which can be solved either analytically (difficult) or numerically. In other words, for the numerical solution of Equation 3.0.1.5, we will discretize the surface of the configuration into a series of panels $[1, \ldots, N]$, we will assume a particular variation of the potential ϕ on each panel (constant, linear, quadratic sources and/or doublets) and parametrize this variation with a series of coefficients (the unknowns of our equations). We will obtain a number of independent equations equal to the number of unknowns by allowing the point P and its associated potential ϕ_P to approach points on the surface of the body S_B where the values of the potential are known according to the chosen parameterization. Again, this will result in a set of simultaneous linear equations $Ax = b$ which can be easily solved using standard procedures.

As a reminder, the setup of panels and nodes, together with their numbering for a typical airfoil analysis problem can be found in figure 3.2 below (taken from [83]).
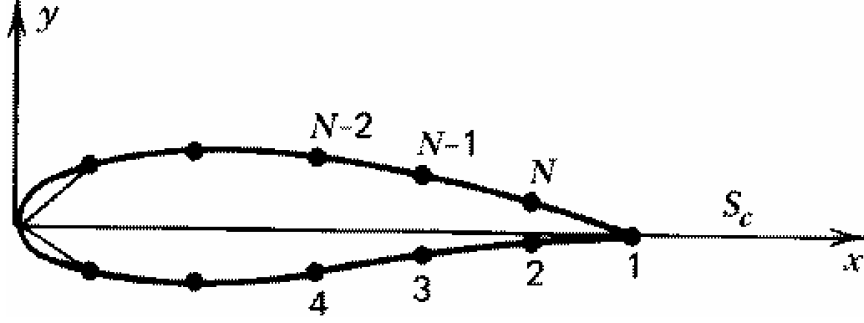
Figure 3.2: Panels on Airfoil

In the solution of the non-cavitating flow problem we used the NACA 4412 hydrofoil. This series of foils has an analytic geometric description, thus making easier the generation of the geometry for the numerical solution. The panel edge points were placed according to cosine spacing in the foil's chord length, in order to have a higher concentration of panels in areas of high pressure gradients (that is, the leading and trailing edges).

Constant-Potential Method

This method is also known as the two-dimensional version of Morino and Kuo. In its simplest incarnation, we choose the potential to be constant on each panel of the configuration, but the values of these constants are allowed to differ from panel to panel. That is:

$$\phi = \phi_j$$

on panel j. That is equivalent to a distribution of doublets with constant strength on each panel. According to the Kutta condition, $\triangle\phi = \phi_N - \phi_1$.

Therefore, we have N unknowns (the values of the potential on each panel), and we need N independent equations to solve for these unknowns. These equations can be obtained by evaluating Equation (3.0.1.5) at the midpoint of each panel. Defining

$$\overline{x}_i = \frac{1}{2}(x_i + x_{i+1})$$
$$\overline{y}_i = \frac{1}{2}(y_i + y_{i+1})$$

we have

$$\phi_i = V_\infty(\overline{x}_i \cos\alpha + y_i \sin\alpha) - \sum_{j=1}^{N} \phi_j \int_{\text{panel}j} \mathbf{n} \cdot \nabla\phi_s dS - (\phi_N - \phi_1) \int_{\mathcal{S}_C} \mathbf{n} \cdot \nabla\phi_s dS$$

(3.0.2.1)

The integrals in this equation need to be evaluated and they are most easily performed in a coordinate system aligned with the panel that is contributing to the potential. Let $(x^*,\ y^*)$ be the coordinates of the midpoint of panel i on a coordinate system fixed on panel j such as the one in figure 3.3.

21
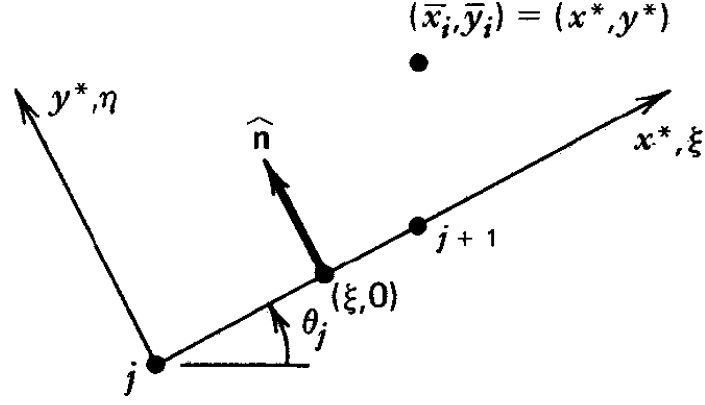
Figure 3.3: Local panel-fixed coordinates

The value of $\mathbf{n} \cdot \nabla \phi_s$ at $(\xi, 0)$ is the y^* component of the velocity due to a source of unit-strength at (x^*, y^*). We can thus obtain

$$\mathbf{n} \cdot \nabla \phi_s \Big|_{\xi,0} = \mathbf{n} \cdot \nabla \left(\frac{1}{2\pi} \ln \sqrt{(x^* - \xi)^2 + y^{*2}} \right) \Big|_{\xi,0} = -\frac{1}{2\pi} \cdot \frac{y^*}{(x^* - \xi)^2 + y^{*2}} \quad (3.0.2.2)$$

and equation (3.0.2.1) can be written

$$\phi_i = V_\infty (\overline{x}_i \cos \alpha + y_i \sin \alpha) + \sum_{j=1}^{N} \phi_j \int_0^{l_j} \frac{y^*}{(x^* - \xi)^2 + y^{*2}} d\xi \qquad (3.0.2.3)$$
$$+ (\phi_N - \phi_1) \int_0^{+\infty} \frac{y^*}{(x^* - \xi)^2 + y^{*2}} d\xi$$

On the panel j,

$$\int_0^{l_j} \frac{y^*}{(x^* - \xi)^2 + y^{*2}} d\xi = \tan^{-1} \frac{y^*}{x^* - \xi} \Big|_{\xi=0}^{\xi=l_j} = \beta_{ij} \ , \qquad (3.0.2.4)$$

where β_{ij} is the angle subtended at (x^*, y^*) by the jth panel. In the case of the wake panel we have,

$$\int_0^{+\infty} \frac{y^*}{(x^* - \xi)^2 + y^{*2}} d\xi = \tan^{-1} \frac{y^*}{x^* - \xi} \Big|_{\xi=0}^{\xi=+\infty} = -\tan^{-1} \frac{y^*}{x^*} = \tan^{-1} \frac{\overline{y}_i}{\overline{x}_i - c} \ ,$$

on the global coordinates. After all integrations are carried out, one arrives at an equation like

$$\sum_{j=1}^{N} A_{ij} \phi_j = b_i \ ,$$

for $i = 1, \ldots, N$ with

$$b_i = -V_\infty (\overline{x}_i \cos \alpha + y_i \sin \alpha)$$

$$A_{ij} = \begin{cases} \frac{1}{2\pi} \beta_{i1} - \delta_{i1} - \frac{1}{2\pi} \beta_{iN+1} & \text{if } j = 1 \\ \frac{1}{2\pi} \beta_{iN} - \delta_{iN} + \frac{1}{2\pi} \beta_{iN+1} & \text{if } j = N \\ \frac{1}{2\pi} \beta_{ij} - \delta_{ij} & \text{otherwise} \end{cases}$$

22

where δ_{ij} is the Kronecker delta and $\beta_{iN+1} = \tan^{-1} \frac{\bar{y}_i}{\bar{x}_i - c}$. Finally, the velocity can be calculated in a variety of ways. The simplest of all is to compute the velocity at the nodes of the discretization by using $V_i = \dfrac{\phi_i - \phi_{i-1}}{d}$, where d is the distance of the midpoints of the i, $i-1$ panels.
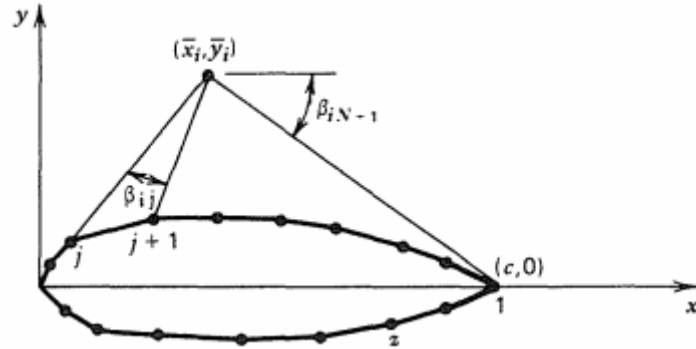
Figure 3.4: Elements of calculation

The various elements of this calculation are defined in Figure 3.4. Finally, with the velocities already calculated, one can continue as usual and compute the pressure distribution using Bernoulli's equation, whose result can then be used to integrate the forces on the body. The pressure field resulting from the calculation for an 8 deg angle of attack can be seen in figure 3.5. Having calculated the distribution of doublets, field values of the velocity were obtained by making use of the formulas for the induced velocity due to a constant strength doublet found in [84]. Figure 3.6 is a comparison of the calculated results with the experimental values found in [85]. The assumption for zero viscosity in the boundary element method is the main factor for the deviation of numerical values from the experiment. Nevertheless, we see that the BEM is an effective method in predicting the pressure field at a minimum computational cost.
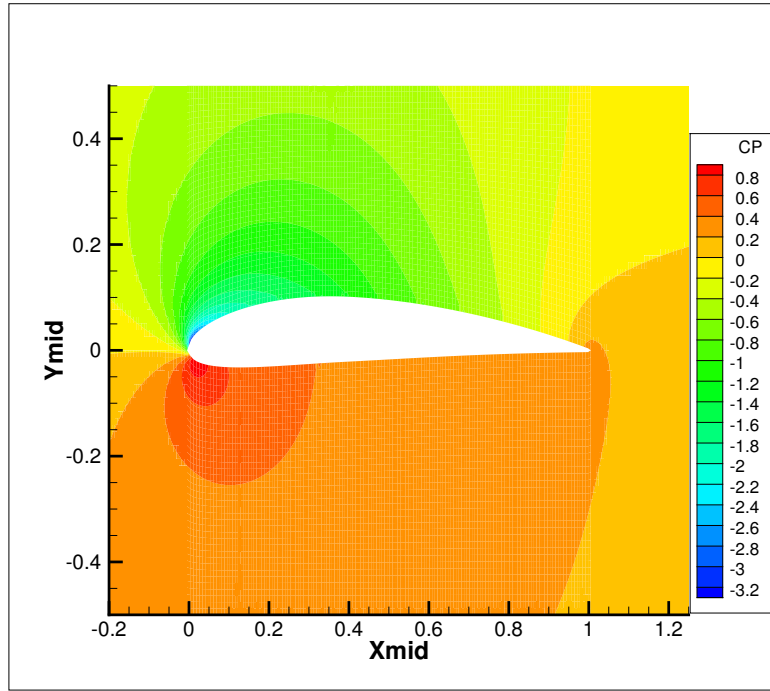
Figure 3.5: Constant strength doublet method field pressures, $\alpha = 8°$
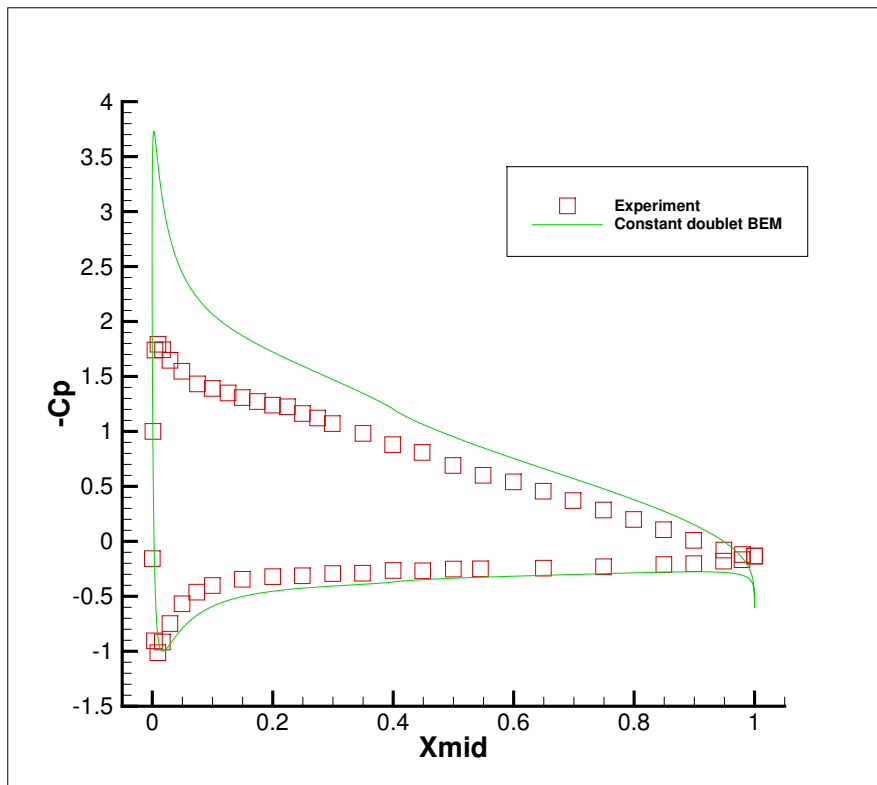
Figure 3.6: Constant strength doublet method body pressure distribution, $\alpha = 8°$

3.0.3 Linear-Potential Method

In the linear potential method the potential is assumed to vary linearly over each panel according to:

$$\phi = \phi_j + \frac{\xi}{l_j}(\phi_{j+1} - \phi_j) \tag{3.0.3.1}$$

on panel j, where ξ is the distance from the jth node on the jth panel, and ϕ_j is the value of ϕ at the jth node. The Kutta condition consequence must now be modified into:

$$\triangle\phi = \phi_{N+1} - \phi_1$$

where ϕ_1 and ϕ_{N+1} are the potentials on the lower and upper sides, respectively, of the trailing edge. Using equation 3.0.2.2 for $\hat{n} \cdot \nabla\phi_s$, and the integral equation for the non-cavitating flow, we thus obtain:

$$\phi_P = V_\infty(x_P \cos\alpha + y_P \sin\alpha) + \frac{1}{2\pi}\sum_{j=1}^{N}\int_{\text{panel } j}\phi\frac{y^*}{(x^* - \xi)^2 + y^{*2}}d\xi$$

$$+ \frac{1}{2\pi}(\phi_{N+1} - \phi_1)\int_0^\infty \frac{y^*}{(\text{x}^* - \xi)^2 + y^{*2}}d\xi \tag{3.0.3.2}$$

in which (x^*, y^*) are the coordinates of point P. On introducing the value of ϕ on the jth panel, we obtain integrals similar to (3.0.2.4) for the steady part of the potential, while the linear part is reduced to:

$$\int_0^{l_j}\frac{y^*\xi}{(x^* - \xi)^2 + y^{*2}}d\xi = \frac{1}{2}y^*\ln\left[(\xi - x^*)^2 + y^{*2}\right] + x^*\tan^{-1}\frac{\xi - x^*}{y^*}\bigg|_0^{l_j} =$$

$$= \frac{1}{2}y^*\ln\left[(l_j - x^*)^2 + y^{*2}\right] + x^*\tan^{-1}\frac{l_j - x^*}{y^*} -$$

$$- \frac{1}{2}y^*\ln\left(x^{*2} + y^{*2}\right) - x^*\tan^{-1}\frac{-x^*}{y^*} = \tag{3.0.3.3}$$

$$= y^*\ln\frac{r_{Pj+1}}{r_{Pj}} + x^*\beta_{Pj}$$

in which β_{Pj} is the angle subtended at point P by the jth panel (or by the connecting surface S_c when $j = N + 1$), and r_{Pj} is the distance from P to the jth node. So we have the equation:

$$\phi_P = V_\infty(x_P\cos\alpha + y_P\sin\alpha) +$$

$$+ \frac{1}{2\pi}\sum_{j=1}^{N}\left\{\left[\phi_j + \frac{x^*}{l_j}(\phi_{j+\text{i}} - \phi_j)\right]\beta_{Pj} + \frac{y^*}{l_j}(\phi_{j+1} - \phi_j)\ln\frac{r_{Pj+1}}{r_{Pj}}\right\} \tag{3.0.3.4}$$

$$+ \frac{1}{2\pi}(\phi_{N+1} - \phi_1)\beta_{PN+1}$$

Equation (3.0.3.4) can be turned into a set of linear equations for the $N + 1$ values of ϕ by letting P become the ith node, $i = 1, 2, 3...N + 1$. However,if we replace the subscript P by i in equation 3.0.3.4, we will get singular log values and

indeterminate angle β_{ij} when $j = i$ or $i - 1$. For this reason, equation (3.0.3.4) is treated differently. The singularity vanishes if, in each application of equation (3.0.3.4) on the node i, we zero out β_{ij} and $\ln r_{ij}$. These deletions must then be compensated by adding in the right hand side of equation (3.0.3.4) the quantity

$$\frac{1}{2\pi}\phi_i(\beta_{ii-1} + \beta_{ii})$$

while

$$\beta_{ii-1} + \beta_{ii} = \tan^{-1}\frac{(y_i - y_{i+1})(x_i - x_{i-1}) - (x_i - x_{i+1})(y_i - y_{i-1})}{(x_i - x_{i+1})(x_i - x_{i-1}) + (y_i - y_{i+1})(y_i - y_{i-1})}, \qquad (3.0.3.5)$$

is the angle subtended at the ith node by the $(i-1)$th and the $(i+1)$th nodes. The details of this calculation can be found in [83] and omitted here for brevity. The details of the calculation for the trailing edge are included for completeness, since they weren't found in some reference.

Let us consider in detail, therefore, the effect on equation 3.0.3.4 of letting the point P approach the trailing edge node. the troublesome terms of equation (3.0.3.4) are those for which $j = i$ or $i - 1$. Thus we write it:

$$\phi_1 = \frac{1}{2\pi}\left[\phi_1 + \frac{x_1^*}{l_1}(\phi_2 - \phi_1)\right]\beta_{11} + \frac{1}{2\pi}\frac{y_1^*}{l_1}(\phi_2 - \phi_1)\ln\frac{r_{12}}{r_{11}}$$
$$+ \frac{1}{2\pi}\left[\phi_N + \frac{x_N^*}{l_N}(\phi_{N+1} - \phi_N)\right]\beta_{1N} + \frac{1}{2\pi}\frac{y_N^*}{l_N}(\phi_{N+1} - \phi_N)\ln\frac{r_{1N+1}}{r_{1N}} + \quad (3.0.3.6)$$
$$+ \frac{1}{2\pi}(\phi_{N+1} - \phi_1)\beta_{1N+1} + \text{regular terms}$$

while x_1^*, y_1^*, $y_N^* \to 0$, $x_N^* \to l_N$. With the help of Fig. 3.7 we have
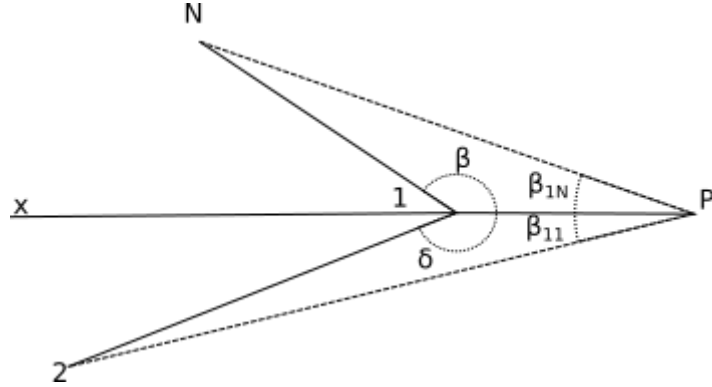
Figure 3.7: Trailing edge influence coefficient

$$\phi_1 \to \frac{1}{2\pi}\phi_1\beta_{11} + \frac{1}{2\pi}(\phi_N + \phi_{N+1} - \phi_N)\beta_{1N} + \frac{1}{2\pi}(\phi_{N+1} - \phi_1)\beta_{1N+1} + \text{r.t.}$$
$$\to \frac{1}{2\pi}\phi_1(\pi - \delta) + \frac{1}{2\pi}\phi_{N+1}(\pi - \beta) + \frac{1}{2\pi}(\phi_{N+1} - \phi_1)\beta_{1N+1} + \text{r.t.} \qquad (3.0.3.7)$$
$$\to -\frac{1}{2\pi}(\phi_1\delta + \phi_{N+1}\beta) + \frac{1}{2}(\phi_1 + \phi_{N+1}) + \frac{1}{2\pi}(\phi_{N+1} - \phi_1)\beta_{1N+1} + \text{r.t}$$

26

and $\beta_{1N+1} \rightarrow -\pi$, so if we move ϕ_1 on the right hand side we get

$$0 = -\frac{1}{2\pi}(\phi_1 \delta + \phi_{N+1} \beta) + \text{regular terms} \tag{3.0.3.8}$$

Having identified and then removed a real problem with the use of equation 3.0.3.4, equations 3.0.3.4 are insufficient to determine the $N+1$ unknowns ϕ_i. Only N equations of the form of 3.0.3.4 are independent, since nodes i and $N+1$ (the two sides of the trailing edge) yield the same result. *The set of equations can be completed if we implement an explicit Kutta condition, appart from its consequence regarding the wake panel. The Kutta condition is equivalent to requiring that the tangential velocities at the midpoints of the two trailing-edge panels are equal:*

$$\frac{\phi_{N+1} - \phi_N}{l_N} = \frac{\phi_1 - \phi_2}{l_1}$$

To setup the algorithm for the computations the following procedure was used. The coordinates of the control point $P(x_i^*, y_i^*)$ on the coordinate system of panel j are:

$$x^* = (x_i - x_j) \cos \theta_j + (y_i - y_j) \sin \theta_j$$
$$y^* = -(x_i - x_j) \sin \theta_j + (y_i - y_j) \cos \theta_j$$

The linear-potential equations can be put in the form of a linear system $[A][\phi] = [b]$. First we set:

$$A_{ij}^{(1)} = \frac{1}{2\pi} \left[\left(1 - \frac{x^*}{l_j}\right) \beta_{ij} - \frac{y^*}{l_j} \ln \frac{r_{ij+1}}{r_{ij}} \right]$$

$$A_{ij+1}^{(2)} = \frac{1}{2\pi} \left(\frac{x^*}{l_j} \beta_{ij} + \frac{y^*}{l_j} \ln \frac{r_{ij+1}}{r_{ij}} \right)$$

for $i, j = 1, ... N$, $j \neq i, i-1$ and equal to 0 for $j = i$ or $i-1$, or when $i = 1$ and $j = N$(the trailing edge).
Then we set for $i = 1, ..., N$:

$$A_{ij} = \begin{cases} -\dfrac{\delta}{2\pi} & \text{when } i = 1, j = 1 \\[2mm] -\dfrac{\beta}{2\pi} & \text{when } i = 1, j = N+1 \\[2mm] A_{i1}^{(1)} + 0 - \dfrac{1}{2\pi}\beta_{iN+1} & \text{when } i \neq 1, j = 1 \\[2mm] 0 + A_{iN+1}^{(2)} + \dfrac{1}{2\pi}\beta_{iN+1} & \text{when } i \neq 1, j = N+1 \\[2mm] A_{ij}^{(1)} + A_{ij}^{(2)} + \delta_{ij}\left(-1 + \dfrac{1}{2\pi}|\beta_{ii-1} + \beta_{ii}|\right) & \text{when } i \neq 1, j = 2, ..., N \end{cases}$$

where δ_{ij} is the Kronecker delta. For $i = N+1$ we have the Kutta condition which translates to:

$$A_{N+1,1} = l_N, \quad A_{N+1,2} = -l_N, \quad A_{N+1,N} = l_1, \quad A_{N+1,N+1} = -l_1$$

To derive the velocity and pressure fields we differentiate equation 3.0.3.1, the formula for the assumed linear variation of the potential along the panel, and take the result as the tangential velocity at the panel midpoint $(\overline{x}_i, \overline{y}_i)$:

$$V_t(\overline{x_i}, \overline{y_i}) \approx \frac{\phi_{i+1} - \phi_i}{l_i}$$

It should be noticed that the absolute value of the angle $\beta_{ii-1} + \beta_{ii}$ was used in the calculation of the linear system coefficients, in order to avoid negative values of the angle.

Regarding the calculation of field point velocities and pressures, formulas for the induced velocity of linear strength doublets were taken, again, from [84]. It should be noted that a somewhat singular behaviour was observed for the points close to the foil. The closest points were placed at a distance $1/300 \cdot$ chord above and below the foil ordinates. This behaviour can be seen in figure 3.8. The pressure field close to the foil elements and the velocity vectors "blow up" near the panel edges (200 panels in total), having a pattern similar to the velocity field created by a point doublet pointing along the foil boundary. A possible explanation for this is related to the flow tangency condition and the spacing of the panels. That is, the flow tangency condition is imposed on the panel midpoints, while the grid of the field point values has 150 equally spaced points for the entire chord of the foil. The spacing of panels follows, on the contrary, the cosine spacing and the produced grid is more dense in the areas of the leading and trailing edges. In other words the distribution of doublets is "smoother" close to the leading edge, and this allows the velocity vectors to better fulfill the flow tangency condition in this area. A relatively dense grid (e.g. 800 panels) is required to produce smooth velocity vectors and pressure values for the entire length of the foil, similar to 3.5
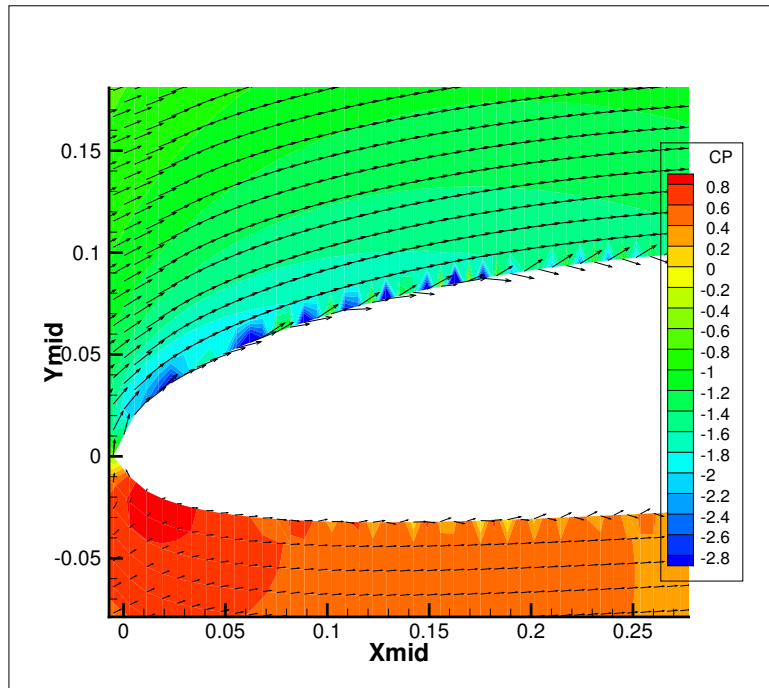
Figure 3.8: Velocity vectors close to the foil

3.0.4 Constant source and dipole method

Previous methods involved the use of doublets only to solve the potential flow around a hydrofoil. in this section we will present a method in which both source and doublet elements are distributed on the body surface and wake. The no entry condition on the body surface will now give predefined values to the source elements strength. Due to same condition, source elements vanished in previous methods. Equation ?? is equivalent to

$$\phi_P = \int_{\mathcal{S}_B + \mathcal{S}_C} [(\mathbf{n} \cdot \nabla\phi)\phi_s - \phi(\mathbf{n} \cdot \nabla\phi_s)]dS \; , \tag{3.0.4.1}$$

where ϕ is the perturbation potential, which is the velocity potential due to the hydrofoil only. The flow tangency boundary condition requires that on the surface of the body, $\frac{\partial\phi}{\partial n} = \hat{\mathbf{n}} \cdot \nabla\phi = -\hat{\mathbf{n}} \cdot \mathbf{V}_\infty$. The normal component of $\nabla\phi$ is continuous along the surface S_C. the unit normals are equal and opposite on the two sides of S_C, so $\int_{\mathcal{S}_C}[(\mathbf{n} \cdot \nabla\phi)\phi_s dS = 0$. Using the concept of the *potential jump* across S_C defined earlier we can write

$$\phi_P = \int_{\mathcal{S}_B} [(\mathbf{n} \cdot \nabla\phi)\phi_s - \phi(\mathbf{n} \cdot \nabla\phi_s)]dS - \int_{\mathcal{S}_C} \triangle\phi(\mathbf{n} \cdot \nabla\phi_s)dS \tag{3.0.4.2}$$

The influence coefficients of doublet elements have been calculated in the constant potential method section. Using the same nomenclature, we have the source element influence coefficient:

$$S_{ij} = \int_{\text{panel } j} \phi_s dS = \frac{1}{2\pi}\int_0^{l_j} \ln r_{ij} dS = \frac{1}{2\pi}\int_0^{l_j} \ln\left[(x^* - t)^2 + y^{*2}\right]^{1/2} dt \tag{3.0.4.3}$$

$$= -\frac{1}{2\pi}\int_{x^*}^{x^* - l_j} \ln\left(X^2 + y^{*2}\right)^{1/2} dX$$

$$= -\frac{1}{4\pi}\left[X\ln\left(X^2 + y^{*2}\right) - 2X + 2y^*\arctan\frac{X}{y^*}\right]_{X=x^*}^{X=x^* - l_j}$$

$$= -\frac{1}{4\pi}\{(x^* - l_j)\ln\left[(x^* - l_j)^2 + y^{*2}\right] - x^*\ln\left(x^{*2} + y^{*2}\right) + 2l_j - 2y^*\beta_{ij}\}$$

Also define $\left(\frac{\partial\phi}{\partial n}\right)_j = V_\infty\left(\cos\alpha\frac{y_{j+1} - y_j}{l_j} - \sin\alpha\frac{x_{j+1} - x_j}{l_j}\right)$.
Keeping the same nomenclature with subsection 3.0.2, we can the write the linear system in the following form:

$$A_{ij}\phi_j + S_{ij}\left(\frac{\partial\phi}{\partial n}\right)_j = 0 \tag{3.0.4.4}$$

and solve for the unknown ϕ_j. The obtained results were similar to the two previous methods.

3.0.5 Comparison of methods and grid independence

In the following, we make a basic comparison of the three methods presented above with regard to field results and rate of convergence with increasing number of

panels. We also establish the grid independence of results obtained. Te comparison is made for the NACA 4412 foil, with an 8° angle of attack. We observed that the constant source and dipole method gives better field results for a small number of panels. More specifically, a comparison of this method with the linear dipole method can be seen in figure 3.9, in which the same area of the foil is plotted and the number of panels is 200 in both cases. As can be seen, the constant source dipole method results in flow velocities very near tangent to the foil surface, in the vicinity of the boundary. In figure 3.10 the lift coefficient is plotted against the number of panels used in the BEM. Firstly, we observe the independency of results from the size of the panel grid. We, also, see that dipole methods converge faster with an increasing number of panels.
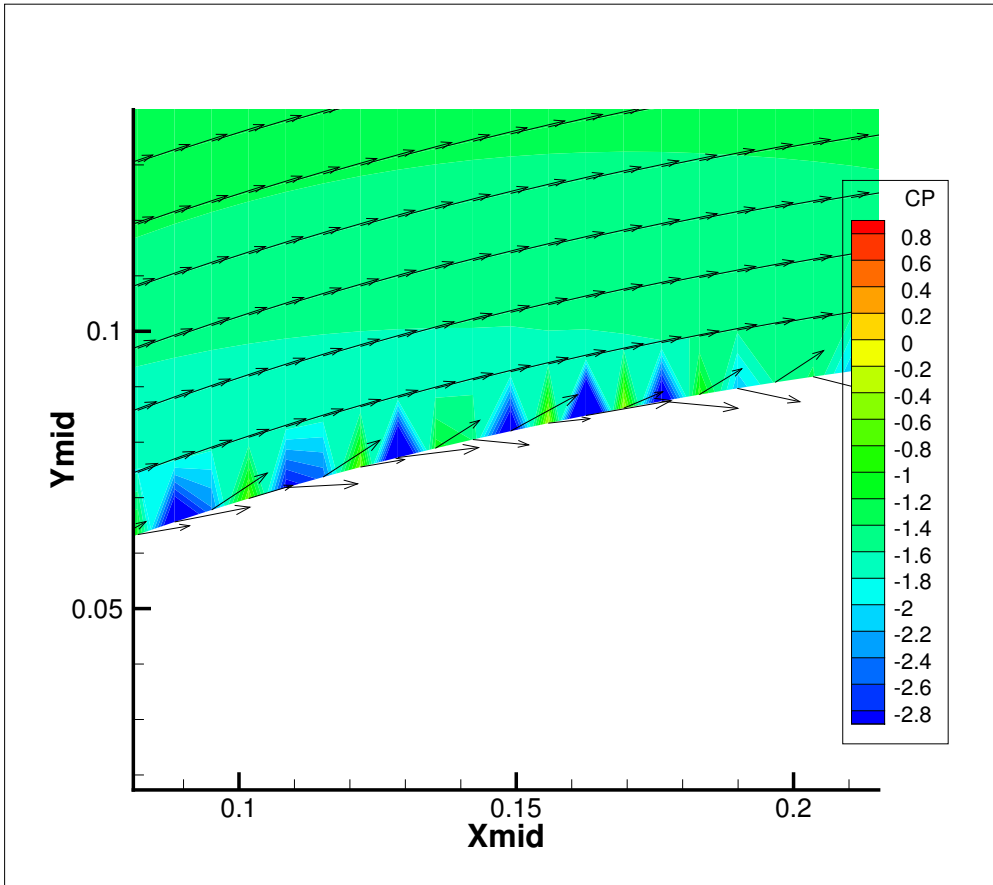
(a) Linear potential method
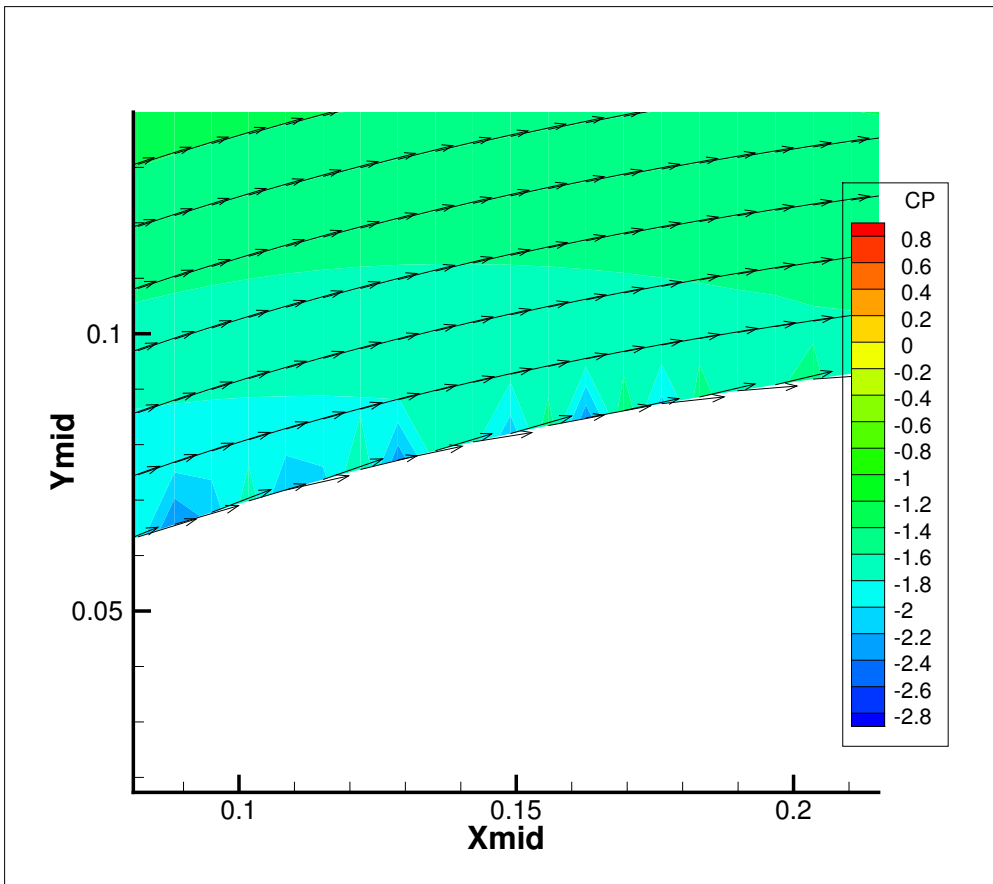
(b) Constant source and dipole method

Figure 3.9: Comparison of methods in field values close to the foil
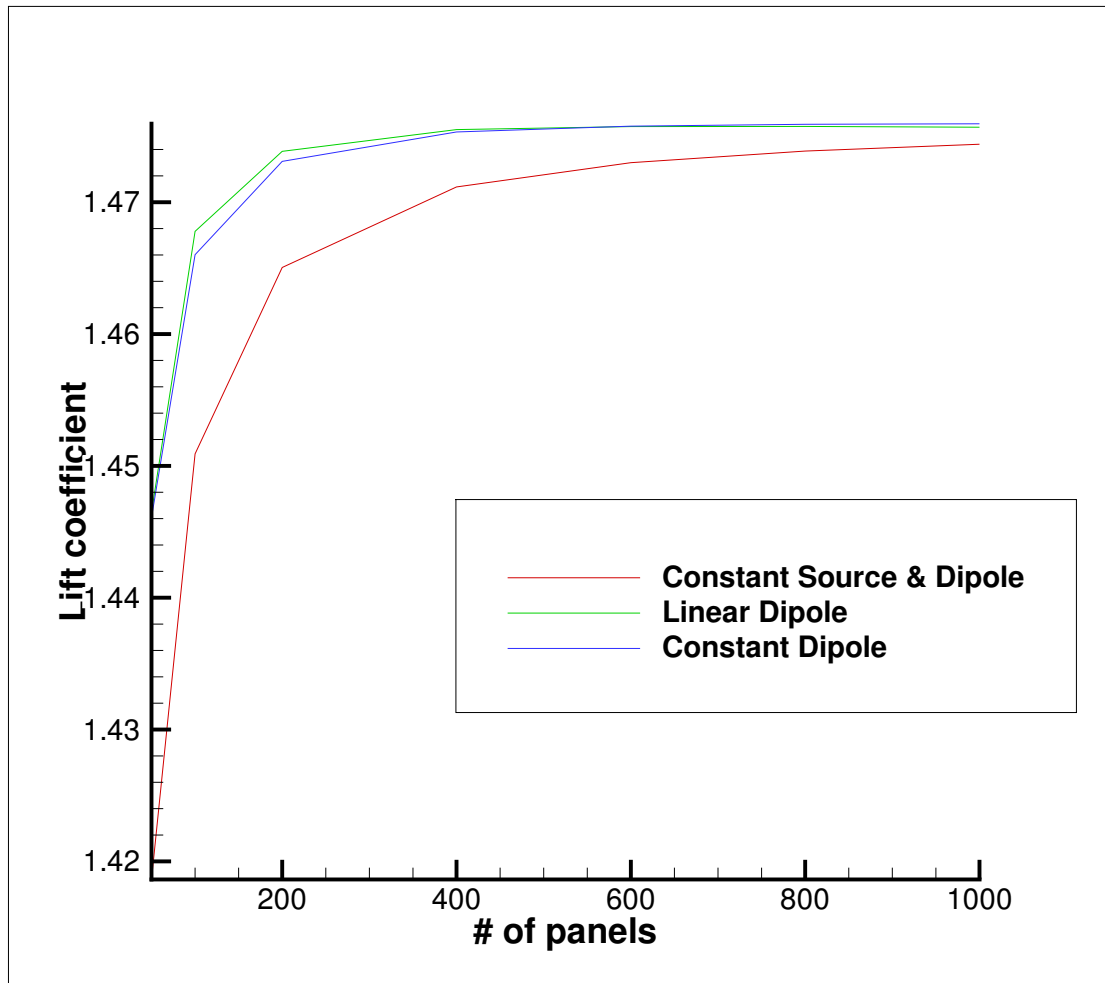
Figure 3.10: Convergence of C_L vs number of panels

Chapter 4

Partial cavitation BEM modelling

In the modelling of the flow around a partially cavitating foil, it is assumed that the fluid is inviscid and incompressible and that the resulting flow is irrotational. In the following, a shift towards the use of the perturbation potential is made and for this reason it will be denoted with lowercase ϕ, interchanging it with the uppercase Φ notation of chapter 2. Thus, the total velocity flow field q can be expressed in terms of either the total potential, Φ, or the perturbation potential, ϕ, as follows

$$q = \nabla \Phi = U_\infty + \nabla \phi. \tag{4.0.0.1}$$

The total and perturbation potentials are related as follows

$$\phi = \Phi(x,\ y) - \Phi_{in}(x,\ y)\ ,$$

where the inflow velocity potential Φ_{in} corresponds to the uniform inflow:

$$\Phi_{in}(x,\ y) = U_\infty(x \cos\alpha + y \sin\alpha)\ .$$

The perturbation potential ϕ will satisfy Laplace's equation in the domain outside the hydrofoil and the cavity

$$\nabla^2 \phi = 0.$$

In addition to the above, ϕ must comply with the following boundary conditions:
Kinematic boundary condition: The flow must be tangent to the wetted blade and cavity surface, which is equivalent to:

$$\frac{\partial \phi}{\partial n} = -U_\infty \cdot n$$

where n is the unit normal to the blade surface.
Dynamic boundary condition on the cavity: The pressure must be constant on the cavity surface. Using Bernoulli's equation and based on the definition of the cavitation number $\sigma = \frac{p_\infty - p_v}{\frac{\rho}{2}U_\infty^2}$, the magnitude of the total velocity on the cavity surface, q_c, is found to be constant:

$$q_c = U_\infty \sqrt{1 + \sigma}.$$

Kutta condition: $\nabla\phi$ finite at the trailing edge.

Condition at infinity: $\nabla\phi \to 0$ at infinity.

Cavity termination model: A termination model is applied at the end of the cavity. A *pressure recovery termination model* is employed, by which the velocity (i.e. pressure) follows a prescribed algebraic law in the transition zone between T and L.

$$q_{tr} = U_\infty \sqrt{1 + \sigma}(1 - f\ (sf))\ ,$$

where $f(s_f)$ is defined as follows:

$$f(s_f) = \begin{cases} 0 & \text{if } s_f < s_T, \\ A(\frac{s_f - s_T}{s_L - s_T})^\nu & \text{if } s_T \le s_f \le s_L, \end{cases} \tag{4.0.0.2}$$

where s_f is the arc-length of the foil measured beneath the cavity, measured from the cavity leading edge, and $A(0 < A < 1)$ and $\nu(\nu > 0)$ are constants. In the formulation outlined in [9], these constants are taken arbitrary. **In the present study, we observed that these constants cannot be completely arbitrary. As will be shown in the results below, the value of A affects, as expected, the value of the flow velocity just before and after termination point L. Furthermore, it was found that a smooth transition of the flow velocity from the cavity to the wet surface is achieved by adding it as an extra boundary condition.** More specifically, we require that the flow velocity just before point L equals the velocity just after L. In mathematical language:

$$\lim_{x \to L^-} \nabla\phi = \lim_{x \to L^+} \nabla\phi \tag{4.0.0.3}$$

We will name this condition as cavity wake boundary condition for easier reference in the following. The pressure law termination model, enhanced with this boundary condition, ensures that computed pressure will smoothly decrease from the cavity to the wet surface, which is more consistent with reality.

In addition to the above boundary conditions, we assume that the cavity height becomes zero at its trailing edge:

$$h(s_L) = 0.$$

Based on Green's theorem, the perturbation potential, ϕ_P, at any point P on the flow field away from the body is given by

$$2\pi\phi_P = \int_S \left(-\phi \frac{\partial \ln r}{\partial n} + \frac{\partial \phi}{\partial n} \ln r \right) ds - \int_W \triangle\phi_W \frac{\partial \ln r}{\partial n} ds \tag{4.0.0.4}$$

where r is the distance from the surface element ds, on the combined foil/cavity surface S, to the point P, and $\triangle\phi_W$ is the potential jump in the wake.
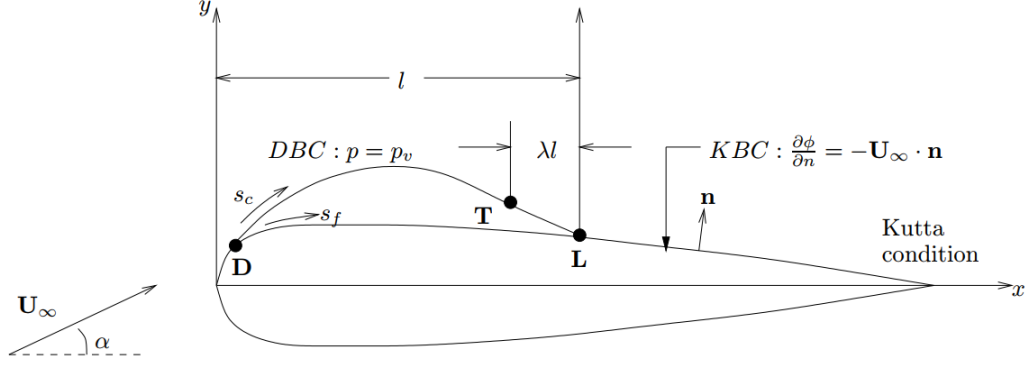
Figure 4.1: Geometry of partial-cavitating hydrofoil

4.1 Numerical Formulation

The formulation outlined above is the basis for the application of boundary element methods (BEM) to obtain a numerical solution to the problem. It can be used to develop a potential-based (i.e. based on Green's identity for the perturbation potential) BEM for the nonlinear analysis of the flow around partially hydrofoils. In the following, we describe such a BEM for partial cavitation with a fixed cavity length.

4.1.1 Cavity Shape for fixed Cavity Length

The perturbation potential ϕ_P at a point on the foil or the cavity satisfies Green's formula 4.0.0.4. The kinematic boundary condition is used to calculate the value of $\partial\phi/\partial n$ in the wet part of the foil . The velocity on the cavity including the transition zone is given by the following algebraic law:

$$\frac{\partial\phi}{\partial s_c} + \frac{\partial\Phi_{in}}{\partial s_c} = q_c(1 - f(s_f)) \tag{4.1.1.1}$$

where s_c (cf. Figure 4.1) is the arclength of the cavity. Integrating 4.1.1.1 we have the following formula for ϕ on the cavity:

$$\phi(s_c) = \phi(0) - \Phi_{in}(s_c) + \Phi_{in}(0) + q_c \int_0^{s_c} (1 - f(s_f))ds_c \tag{4.1.1.2}$$

The cavity surface, and thus s_c is determined by implementing the model iteratively. In the first iteration, the cavity panels coincide with the foil surface underneath the cavity. At each successive iteration, the vertices of the cavity panels are updated by calculating the cavity height at each point. The cavity height h_c is normal to the present iteration cavity surface and will be nonzero until the kinematic boundary condition is be satisfied. It can be shown that the following relationship is valid up to first order in h_c [8]:

$$q_c(1 - f(s_f))\frac{dh_c}{ds_c} = \frac{\partial\phi}{\partial n} + \frac{\partial\Phi_{in}}{\partial n}. \tag{4.1.1.3}$$

Combining 4.1.1.3 with the zero cavity height requirement at its trailing edge gives the cavity closure condition:

$$\int_0^{s_{c_L}} \frac{\partial \phi}{\partial n} \frac{ds_c}{1 - f(s_f)} = -\int_0^{s_{c_L}} \frac{\partial \Phi_{in}}{\partial n} \frac{ds_c}{1 - f(s_f)},$$

where s_{c_L} is the total arclength of the cavity surface. The incremental cavity thickness can be computed by numerically integrating 4.1.1.3 with the calculated values of $\partial \phi / \partial n$. The shape of the cavity surface and the new cavity arc length, s_c, can then be calculated and used subsequently by the following equation:

$$h_c = \int_0^{s_{c_L}} \frac{1}{q_c(1 - f(s_f))} \left(\frac{\partial \phi}{\partial n} + \frac{\partial \Phi_{in}}{\partial n} \right) ds_c. \tag{4.1.1.4}$$

The value of $\phi(0)$ in 4.1.1.2 is the value of the perturbation potential at the cavity detachment point, D, as shown in Figure 4.1.

The above described method will be now presented in more detail.

We need to build a numerical method to solve the integral equation 4.0.0.4 subject to the kinematic and dynamic boundary conditions. The cavity and the foil are discretized into N straight panels. N_c of these panels make the cavity while the remaining $N_w = N - N_c$ panels make the wetted part of the foil.

The continuous source and dipole distributions on each panel are approximated by constant strength elements. We then have N unknown dipole strengths, N unknown source strengths and the unknown q_c, making a total of $2N+1$ unknowns. Equation 4.0.0.4 is applied at the midpoints of the panels. The Kutta condition is numerically equivalent to the Morino condition:

$$\triangle \phi_W = \phi_{TE}^+ - \phi_{TE}^- ,$$

where ϕ_{TE}^+ and ϕ_{TE}^- are the potentials at the upper and lower trailing-edge panels respectively. Equation 4.0.0.4 can be written in discretized form:

$$2\pi \phi_i = \sum_{j=1}^{N} \int_{S_j} \left[-\phi_j \frac{\partial \ln r_{ij}}{\partial n} + \left(\frac{\partial \phi}{\partial n} \right)_j \ln r_{ij} \right] ds - \int_W \Delta \phi_W \frac{\partial \ln r_{iw}}{\partial n} ds , \tag{4.1.1.5}$$

where $i = 1, .., N$ panels, r_{ij} is the distance of panel i from panel j (or the wake in case of r_{iw}), S_j the surface of panel j and ϕ_j, $\left(\frac{\partial \phi}{\partial n} \right)_j$ the constant doublet and source strengths of j-panel, respectively.

We then apply equation 4.1.1.2 on the N_c cavity surface panels to get:

$$\phi_j = \phi_{n_D} - \Phi_{in}(\vec{x}_j) + \Phi_{in}(\vec{x}_{n_D}) + q_c \sum_{k=1}^{j} (1 - f(s_{f,k})) \Delta s_{c,k} , \tag{4.1.1.6}$$

where:

- n_D is the number of the panel of the cavity detachment point, with position $\vec{x}_{n_D}$,

- $\vec{x}_j$ is the position of the cavity panel j under consideration,

- the summation on the last term is taken over the cavity panels upstream of panel j and

- $f(s_{f,k})$, $\Delta s_{c,k}$ are the termination model function value and cavity panel length of cavity panel k, respectively(k starting from 1 on first cavity panel).

The unknown $\phi(n_D)$ is expressed via a cubic lagrange extrapolation in terms of the unknown potentials on the wetted panels in front of the cavity.

On the N_w fully wetted panels the kinematic boundary condition can be written as:

$$\left(\frac{\partial \phi}{\partial n}\right)_j = -\vec{n}_j \cdot \nabla \Phi_{in}|_{\vec{x}=\vec{x}_j} \tag{4.1.1.7}$$

where $\hat{n}_j$ is the unit normal to panel j and $\nabla \Phi_{in}|_{\vec{x}=\vec{x}_j}$ is the inflow velocity at panel j. By substituting the above equations on the discritized governing equation we will get a linear system with unknowns the N_c $\left(\frac{\partial \phi}{\partial n}\right)_j$, the N_w ϕ_j and q_c. For the following we denote:

- $D_{ij} = \frac{1}{2\pi}\int_{S_j}\left(-\frac{\partial}{\partial n}\ln r_{ij}\right)dS$ the unit doublet influence coefficient on panel i

- $S_{ij} = \frac{1}{2\pi}\int_{S_j}\ln r_{ij}dS$ the unit source influence coefficient on panel i,

- D_{iw} the influence coefficient of a unit doublet along the wake and

- δ_{ij} the kronecker delta.

Then

$$\phi_i = \sum_{j=1}^{N}\phi_j D_{ij} + \sum_{j=1}^{N}\left(\frac{\partial \phi}{\partial n}\right)_j S_{ij} + \Delta\phi_w D_{iw}$$

$$\Rightarrow \sum_{j=1}^{N}\phi_j\left(D_{ij} - \delta_{ij}\right) + \sum_{j=1}^{N}\left(\frac{\partial \phi}{\partial n}\right)_j S_{ij} + \Delta\phi_w D_{iw} = 0$$

The linear system can be simplified by applying the Kutta condition $\Delta\phi_w = \phi_N - \phi_1$.

$$\sum_{j=1}^{N}\phi_j\left(D_{ij,w} - \delta_{ij}\right) + \sum_{j=1}^{N}\left(\frac{\partial \phi}{\partial n}\right)_j S_{ij} = 0$$

where $D_{ij,w} = D_{ij} + D_{iw}(\delta_{Nj} - \delta_{1j})$. The above equation provides N equations. Up to this point we also have N_c equations from equation 4.1.1.6 and N_w equations from 4.1.1.7 making a total of $N + N_c + N_w = 2N$ equations.

To solve the linear system we need one more equation, which is produced by the cavity closure condition:

$$\sum_{j\in N_c}\left(\frac{\partial \phi}{\partial n}\right)_j\frac{\Delta s_{c,k}}{1 - f(s_{f,k})} = -\sum_{j\in N_c}\vec{n}_j \cdot \nabla \Phi_{in}|_{\vec{x}=\vec{x}_j}\frac{\Delta s_{c,k}}{1 - f(s_{f,k})}$$

37

The cavity shape after each iteration is calculated by numerical integration of equation 4.1.1.4. Once the height of the cavity $h_{c,j}$ is established for each cavity panel, coordinates of cavity nodes are updated by the following transformation(k the number of iteration):

$$x_j^{(k+1)} = x_j^{(k)} - h_{c,j} \sin \theta_j^{(k)}$$
$$y_j^{(k+1)} = y_j^{(k)} + h_{c,j} \cos \theta_j^{(k)}$$

The influence coefficients have been calculated in previous sections, so they will be presented here only for completeness. If the collocation point i has (x^*, y^*) coordinates with respect to the coordinate system aligned with panel j of length l_j, with the origin on node j, we have the following influence coefficients for constant strength source and doublet elements:

$$D_{ij} = \frac{1}{2\pi} \int\limits_{S_j} \left(-\frac{\partial}{\partial n} \ln r_{ij} \right) dS = \frac{\beta_{ij}}{2\pi}$$

$$S_{ij} = \frac{1}{2\pi} \int\limits_{S_j} \ln r_{ij} dS$$

$$= -\frac{1}{4\pi} \{ (x^* - l_j) \ln [(x^* - l_j)^2 + y^{*2}] - x^* \ln (x^{*2} + y^{*2}) + 2l_j - 2y^* \beta_{ij} \}$$

$$D_{iw} = \beta_{iw} = \frac{1}{2\pi} \arctan \frac{y}{c - x},$$

where β_{ij} is the angle subtended by panel j from collocation point (x^*, y^*) and c the chord of the foil. When the collocation point i is on the midpoint of panel j the coefficients become $D_{ii} = 0.5$ and $S_{ii} = \frac{1}{2\pi} l_j \left(\ln \frac{l_j}{2} - l_j \right)$. The cavity wake boundary condition is used to calculate the coefficient A in an iterative method. We first assume a value for A_0 and use it as input to the numerical model. The linear system presented above is solved and we then calculate the value of the functions $g(A_0) = |\nabla \phi_{L-1} - \nabla \phi_L|$ and $g'(A_0) = \frac{dg}{dA}(A_0)$. The derivative function is easily calculated by a finite difference method. Having calculated these functions, we can apply a Newton Raphson method to solve for A the equation:

$$g(A) = |\nabla \phi_{L-1} - \nabla \phi_L| = 0$$

Then, A will converge to the value of the iteration:

$$A_{n+1} = A_n - \frac{g(A_n)}{g'(A_n)}$$

4.2   Application of model & results

4.2.1   Geometry of hydrofoil section

The above modeling has been applied to the two dimensional NACA 16006 hydrofoil section, for which there is extensive literature on the subject matter. The

geometry of this section was based on the offset data from the book "Theory of wing sections" from Abbott and Doenhoff[86]. It must be noted that the section has been modified in the trailing edge. The thickness of the trailing edge has been reduced to zero from the original finite one. This modification is necessary to apply the pressure type Kutta condition. Application of the pressure type Kutta condition on the finite thickness trailing edge causes the solution to greatly diverge close to the trailing edge.

Since the available data points of the section geometry are few, we first applied a cubic spline interpolation to produce an adequate number of panels for the method. The resulting geometry had changes in the sign of the curvature, negatively affecting the smoothness of the solution. This was rectified by best-fitting a polynomial function using a weighted least-squares numerical method. The shape of the wake panel is considered to be a straight line connecting the trailing edge with infinity.

Panel boundaries were spaced by means of cosine spacing from foil trailing edge to longitudinal position of cavity trailing edge and from cavity trailing edge to foil leading edge.

4.2.2 Background of cavity wake boundary condition

The need for an additional boundary condition arised when the model was applied for a 3° angle of attack and constant paramaters of the cavity termination model:

- $A = 0.5$, $\nu = 1$ and $\lambda = 0.1$.

- The cavity detachment point is at the foil's leading edge and the length of the cavity is $0.5c$.

The pressure coefficient distribution in this case can be seen in figure 4.2. The resulting solution in this case produces a steep velocity gradient after the cavity trailing edge and the calculated cavity heights are reduced to negative values close to the cavity trailing edge. This, of course, is not physically possible. This breakdown is more evident when plotting the tangent to the foil flow velocity. According to figure 4.2 and figure 4.3, we observed that, while varying A, the flow velocity remains constant in the greatest part of the wet foil and the cavity. Significant difference can be observed only in the region from point T to point L and in a short length after the cavity closure. After having a closer look, one can deduce that $\lim_{x \to L^-} \nabla\phi$ is inversely proportional to A, while $\lim_{x \to L^+} \nabla\phi$ seems to diverge asymptotically either to 0 or to $+\infty$ in the the opposite direction compared to the left side limit. This means that there should be a value of A for which these two limits will be very close. The convergence of the iterative method outlined earlier confirms this statement.

The treatment of A as a variable better agrees with the method outlined in [Kinnas & Fine, 1990], in which the velocity at the end of the cavity q_L is included in the cavity termination model, but is calculated as an extrapolation of the velocity in the wet part of the foil downstream of the cavity.
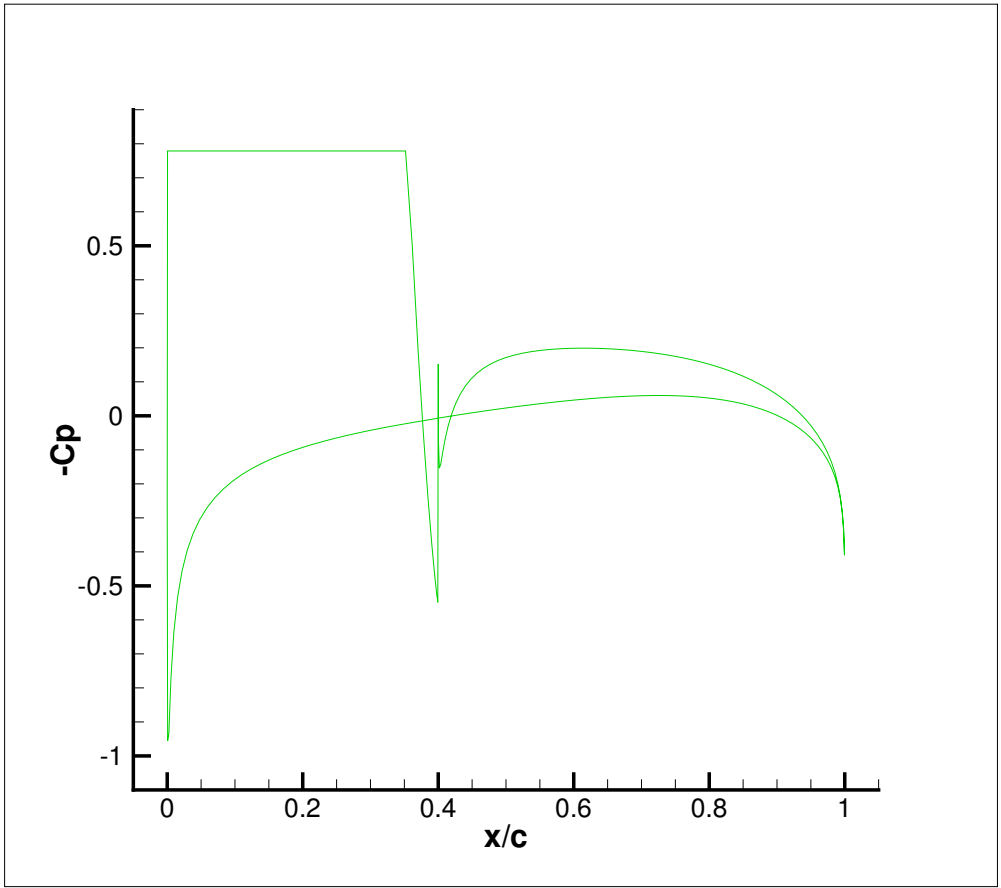
Figure 4.2: Pressure coefficient after first iteration for NACA 16006 and 3° angle of attack
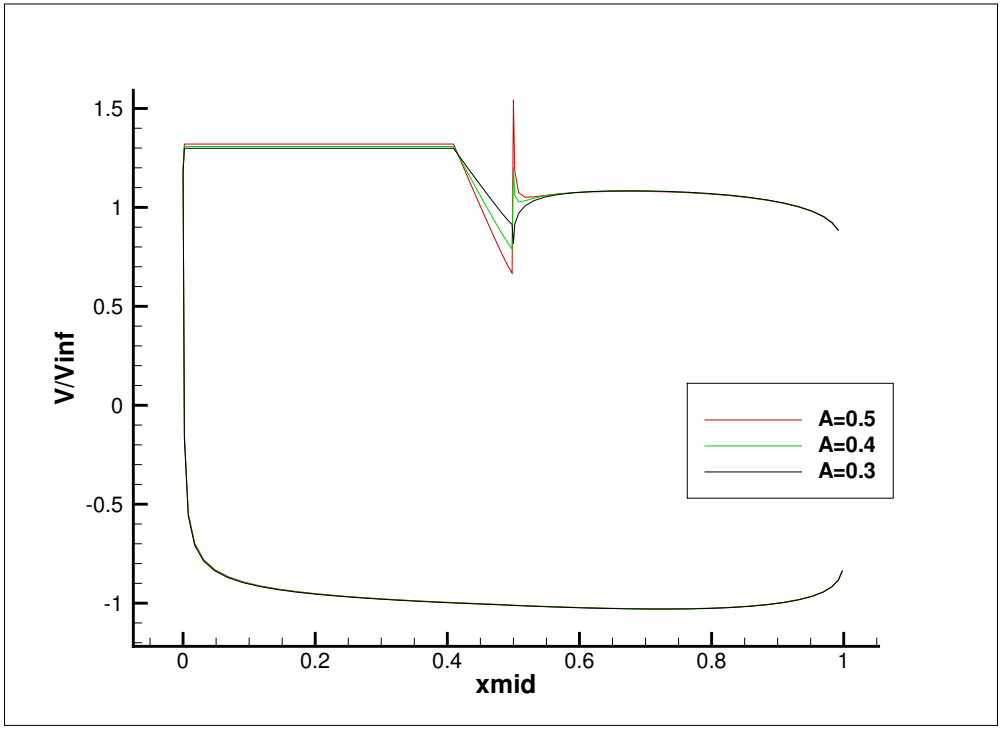
Figure 4.3: Flow velocity $\frac{V}{V_\infty}$ in both sides of the foil
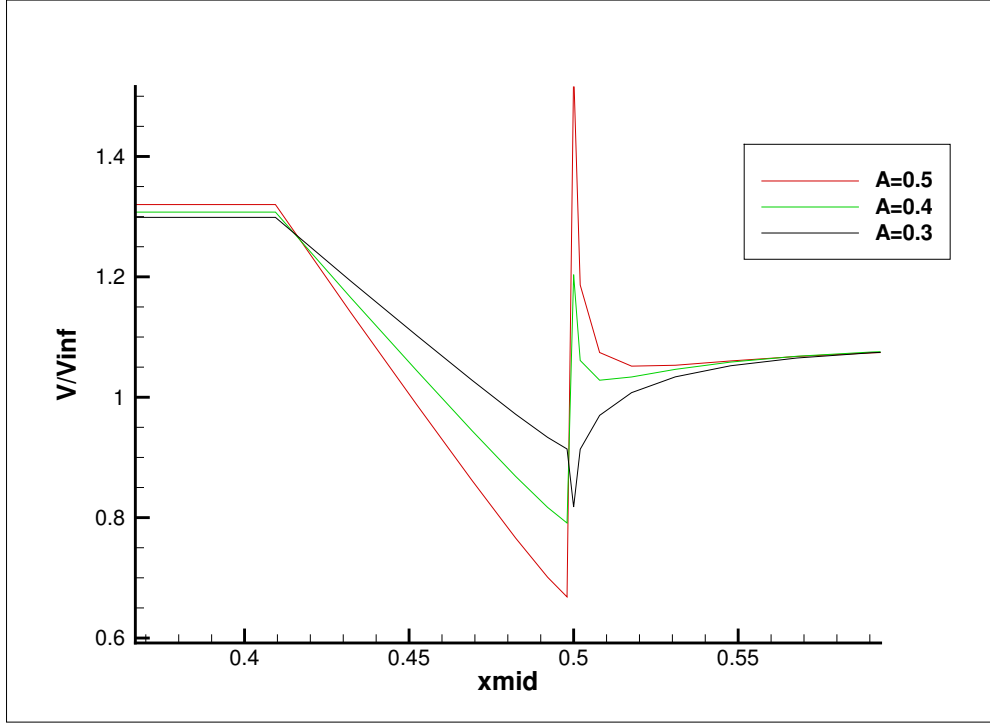
Figure 4.4: Flow velocity $\frac{V}{V_\infty}$ in the cavity transition length and its wake

4.2.3 Background for a restriction on ν parameter

As shown in equation (4.0.0.2), the cavity termination model has a parameter ν. This parameter affects the shape of the cavity. Testing of the model showed that when $\nu < 1$ the produced cavity shape displays a steep curvature gradient at a specific point, resembling to an edge. This happens from the first iteration for the cavity shape and becomes more pronounce with subsequent iterations. Figure confirms the above. The resulting cavity shape after a few iterations can be seen in figure 4.5. For this reason, the model was applied with values for $\nu > 1$.

4.2.4 Input data of the model

The model has been tested for various angles of attack, different geometries and lengths of the cavity. The parameters of the cavity termination model are $A_0 = 0.5$, $\nu = 2$ and $\lambda = 0.1$. The cavity detachment point is at the foil's leading edge. The foil consists of 100 straight panels.

4.2.5 Results

In this subsection, results for a NACA 16006 foil, $4°$ angle of attack and length of cavity $0.5c$ are presented. More results are attached in the appendix of the thesis. The pressure coefficient distribution after 3 iterations of the method is shown in figure 4.8. The cavity shape after the first iteration and the converged cavity shape are shown figure 4.6. From figure 4.7, one can verify the convergence of

41
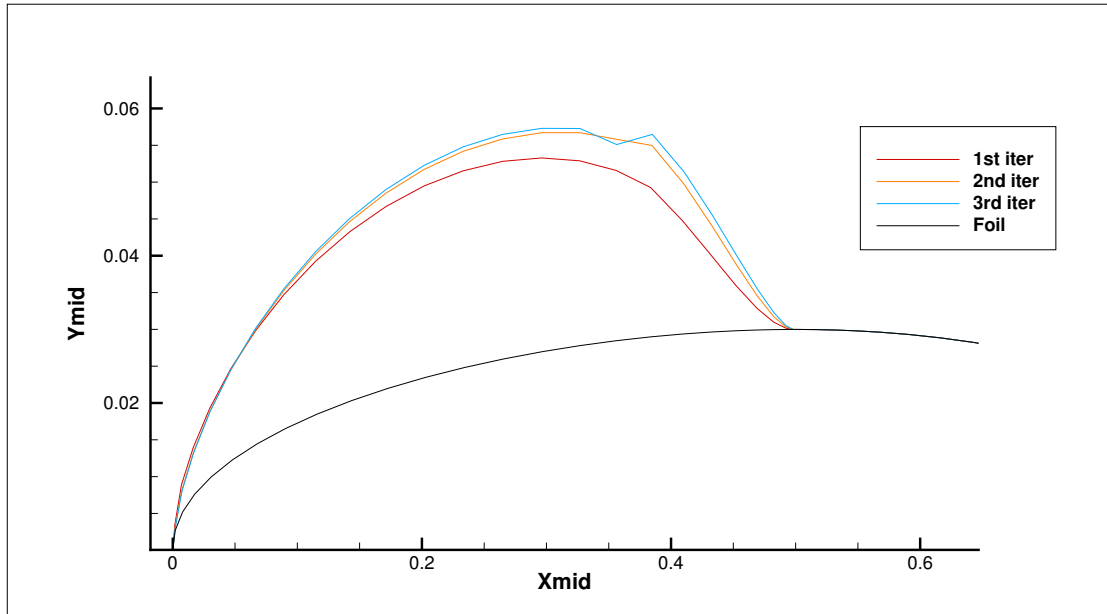
Figure 4.5: Cavity shape in first three iterations, with $\nu = 1$.

the cavitation number with increasing number of iterations. We observe that the cavitation number is very close to the converging value from the third iteration.
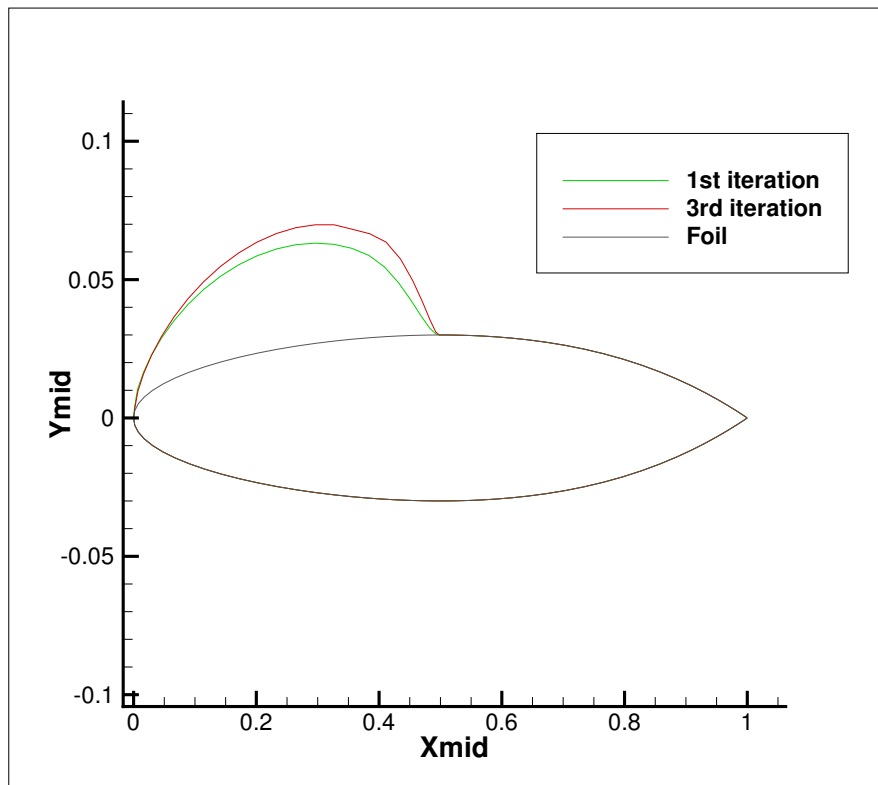
Figure 4.6: Cavity shape after first and third iteration for NACA 16006 and 4° angle of attack

The model has been tested for grid independency. In the 4° case, results
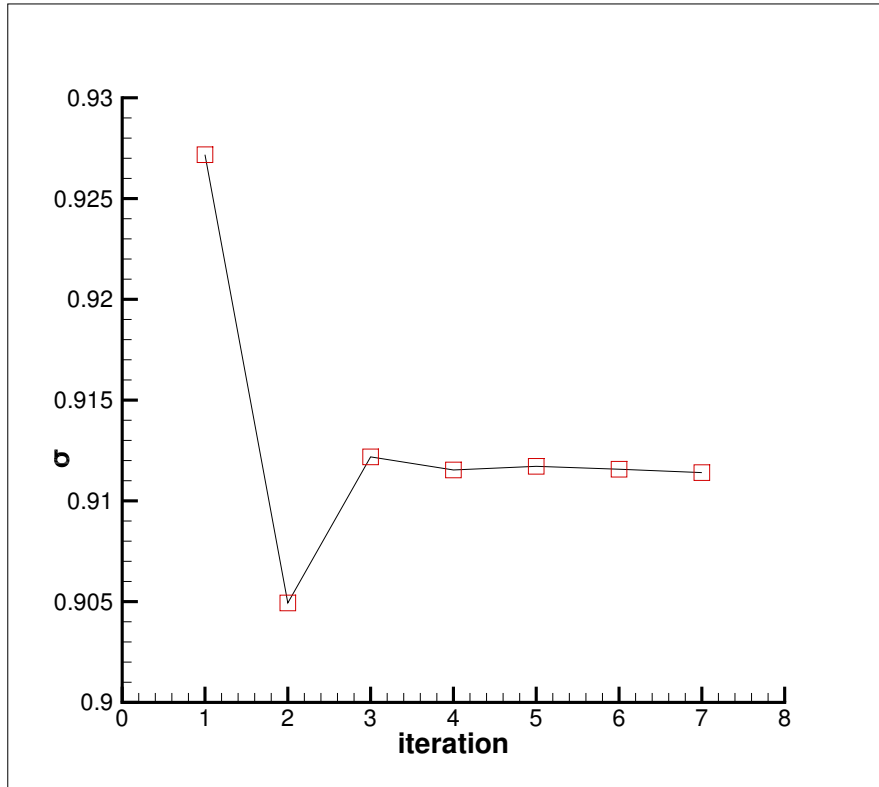
Figure 4.7: Convergence of the cavitation number with increasing number of iterations

do not change significantly with increasing number of panels, as can be seen in the following table comparing the calculated cavitation number after the sixth iteration in increasing density of the panels. The converging trend of the cavitation number with increasing number of panels can be seen also in figure 4.9.

# of panels	cavitation number
100	0.92719
150	0.91997
200	0.91600
250	0.91241
300	0.91157
350	0.91177
400	0.91142
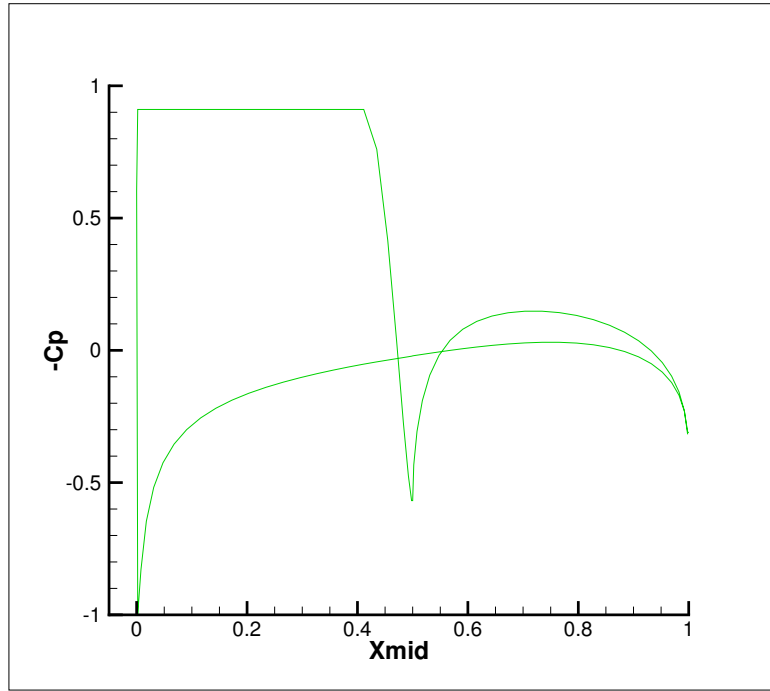
Figure 4.8: Pressure coefficient after third iteration for NACA 16006 and 4° angle of attack
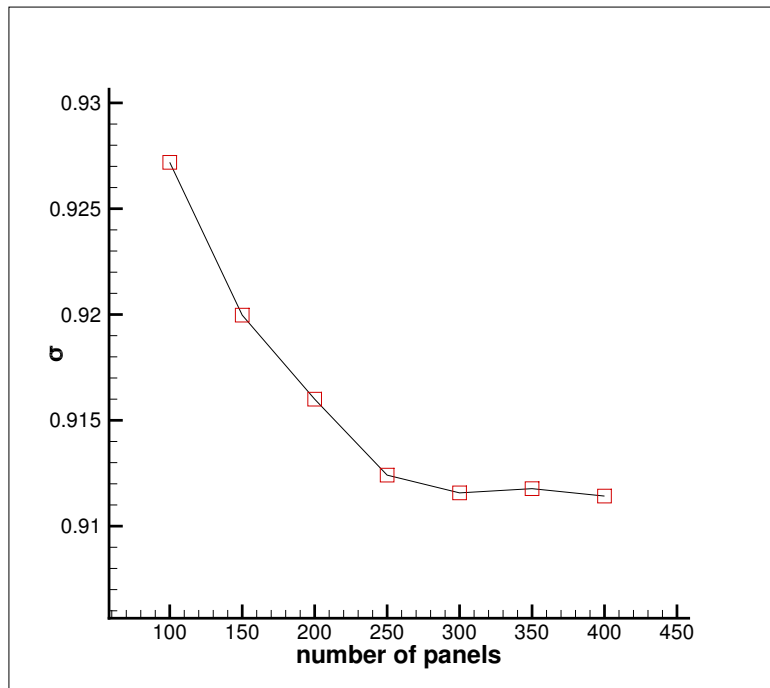
Figure 4.9: Cavitation number after sixth iteration for increasing number of panels(NACA 16006 and 4° angle of attack)

44
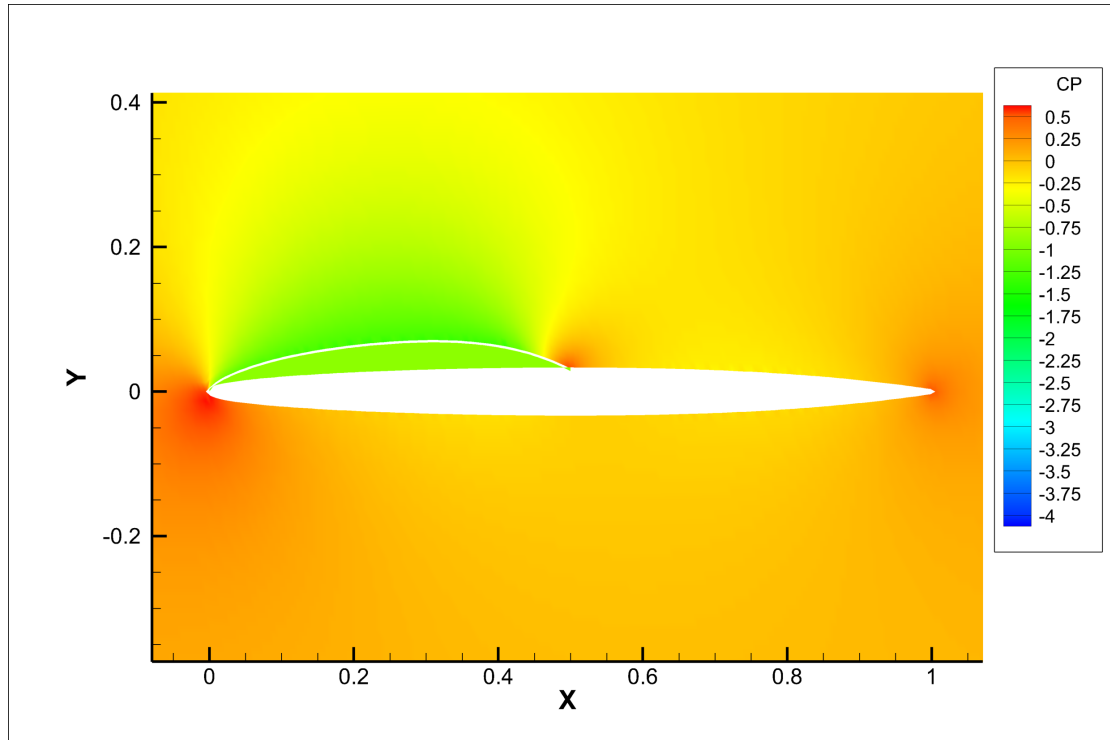
Figure 4.10: Field values of the pressure coefficient for NACA 16006 and 4° angle of attack

Comparison with published results

The results obtained are compared with published results from other researchers. Since the present method follows the method by Kinnas & Fine, a benchmarking is neccessary. In figure 4.11, the calculated pressure coefficient is compared to the published results in [8], for the NACA 16006 foil, 4° angle of attack and length of cavity $l_c/c = 0.5$. A comparison of the same configuration is made with results from [10]. It must be noted that Lee et al. adopt a velocity-based boundary element method and the cavity termination model is the end-plate Riabouchinsky model. This termination model requires that the thickness of the cavity vanishes abruptly at the end of the cavity, forming a boundary resembling an end-plate. In order to get comparable results in this case, the present method was used with very small length of transition zone, with $\lambda = 0.03$.
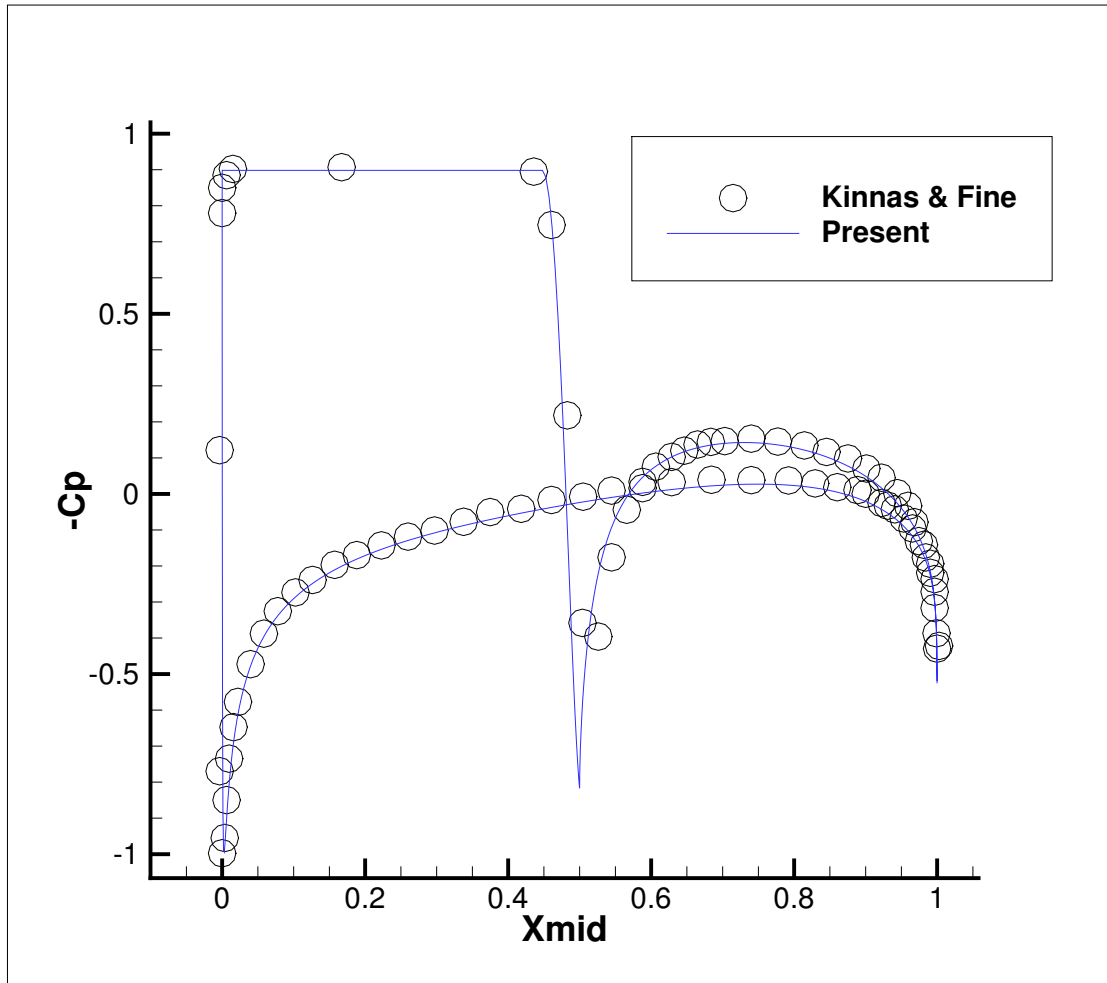
Figure 4.11: Comparison of results with Kinnas & Fine
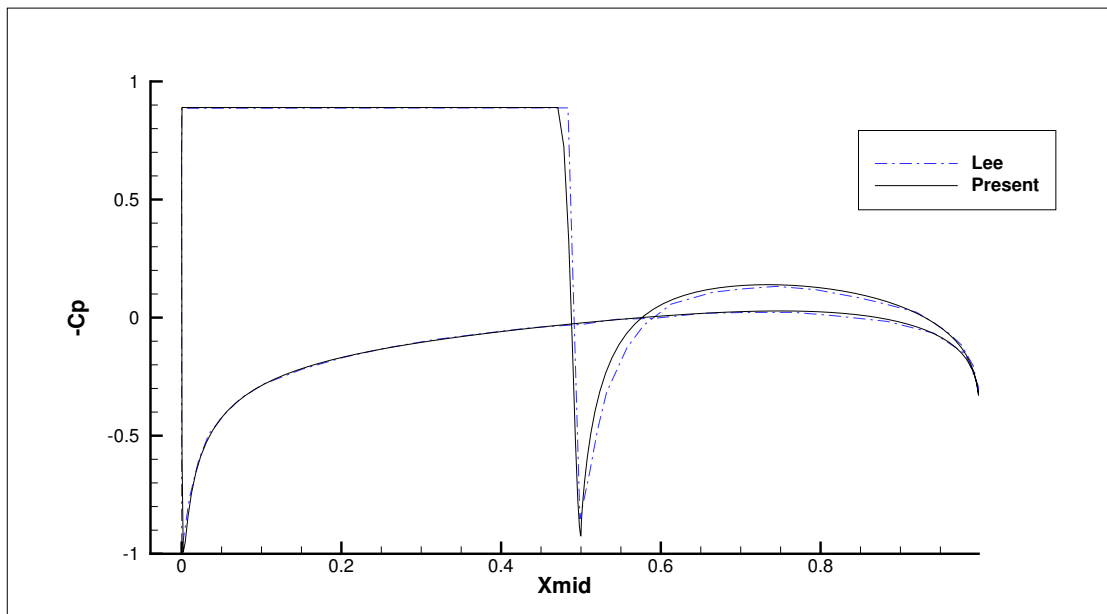
Figure 4.12: Comparison of results with Lee et al

Chapter 5

Conclusions and suggestions for future work

In the present thesis a boundary element method for determining the cavitation number and cavity shape of a two-dimensional hydrofoil with a given cavity length is presented. The method is based on the work of Kinnas & Fine, and modified on the cavity closure model. While maintaining the linearity of system of equations for the boundary problem, an iterative scheme for the cavity's velocity at its trailing edge is implemented by enforcing a cavity wake boundary condition, thus minimizing the required input from the user. The method is validated against published results of similar methods.

Also the present thesis can be used as a reference for programming low-order panel methods, as the discretization of relevant equations has been exposed with detail. Some suggestions for future work can be drawn from the work of other researchers, as the problem has been treated extensively:

- the treatment of the fixed cavitation number problem, in which the cavity shape and length are calculated

- the implementation of a different cavity termination model, such as the re-entrant jet model

- the inclusion of viscosity effects by coupling of the method with a RANS solver or boundary layer solver. Viscosity effects play an important role in cavity detachment and in the cavity wake.

- extension of the method to three-dimensional hydrofoils and marine propellers for prediction of sheet cavitation

- investigation and extension of the method for unsteady movement(e.g heaving) of the hydrofoil within the fluid domain

Bibliography

[1] MH Aliabadi and PH Wen. *Boundary element methods in engineering and sciences*, volume 4. World Scientific, 2011.

[2] Marshall P Tulin. Steady two-dimensional cavity flows about slender bodies. *David W. Taylor Model Basin, Washington DC, USA, Department of the Navy, Research and Development Report 834*, 1953.

[3] James S. Uhlman. The Surface Singularity Method Applied to Partially Cavitating Hydrofoils. *Journal of Ship Research*, 31(02):107–124, 06 1987.

[4] John L Hess and A_M O Smith. Calculation of potential flow about arbitrary bodies. *Progress in Aerospace Sciences*, 8:1–138, 1967.

[5] H Lemonnier and A Rowe. Another approach in modelling cavitating flows. *Journal of Fluid Mechanics*, 195:557–580, 1988.

[6] A Rowe, O Blottiaux, et al. Aspects of modeling partially cavitating flows. *Journal of ship research*, 37(01):34–48, 1993.

[7] Jean Marc Peallat, Christian Pellone, et al. Experimental validation of two- and three-dimensional numerical analysis of partially cavitating hydrofoils. *Journal of ship research*, 40(3):211–223, 1996.

[8] Spyros A Kinnas and Neal E Fine. Non-linear analysis of the flow around partially or super-cavitating hydrofoils by a potential based panel method. In *Boundary Integral Methods*, pages 289–300. Springer, 1991.

[9] Spyros A Kinnas and Neal E Fine. A numerical nonlinear analysis of the flow around two-and three-dimensional partially cavitating hydrofoils. *Journal of Fluid Mechanics*, 254:151–181, 1993.

[10] CS Lee, Young Gil Kim, and JT Lee. A potential-based panel method for the analysis of a two-dimensional super-or partially-cavitating hydrofoil. 1992.

[11] Jie Dang and Gert Kuiper. Re-entrant jet modeling of partial cavity flow on three-dimensional hydrofoils. 1999.

[12] Jie Dang. Numerical simulation of unsteady partial cavity flows. 2002.

[13] Wesley Huntington Brewer. *A computational and experimental study of viscous flow around cavitating propulsors*. PhD thesis, Massachusetts Institute of Technology, 1995.

[14] Paddy Krishnaswamy. Flow modelling for partially cavitating hydrofoils. 2001.

[15] James S Uhlman et al. A note on the development of a nonlinear axisymmetric reentrant jet cavitation model. *Journal of ship research*, 50(03):259–267, 2006.

[16] Akihiro Kubota, Hiroharu Kato, Hajime Yamaguchi, et al. A new modelling of cavitating flows: A numerical study of unsteady cavitation on a hydrofoil section. *Journal of fluid Mechanics*, 240(1):59–96, 1992.

[17] Yongliang Chen and SD Heister. Modeling hydrodynamic nonequilibrium in cavitating flows. 1996.

[18] Robert F Kunz, David A Boger, David R Stinebring, Thomas S Chyczewski, Jules W Lindau, Howard J Gibeling, Sankaran Venkateswaran, and TR Govindan. A preconditioned navier–stokes method for two-phase flows with application to cavitation prediction. *Computers & Fluids*, 29(8):849–875, 2000.

[19] Manish Deshpande, Jinzhang Feng, and Charles L Merkle. Cavity flow predictions based on the euler equations. 1994.

[20] O Coutier-Delgosha, JL Reboud, and Y Delannoy. Numerical simulation of the unsteady behaviour of cavitating flows. *International journal for numerical methods in fluids*, 42(5):527–548, 2003.

[21] Charles L Merkle. Computational modelling of the dynamics of sheet cavitation. In *Proc. of the 3rd Int. Symp. on Cavitation, Grenoble, France, 1998*, 1998.

[22] Jürgen Sauer and Günter H Schnerr. Unsteady cavitating flow-a new cavitation model based on a modified front capturing method and bubble dynamics. In *Proceedings of 2000 ASME fluid engineering summer conference*, volume 251, pages 1073–1079, 2000.

[23] Ashok K Singhal, Mahesh M Athavale, Huiying Li, and Yu Jiang. Mathematical basis and validation of the full cavitation model. *J. Fluids Eng.*, 124(3):617–624, 2002.

[24] Philip J Zwart, Andrew G Gerber, Thabet Belamri, et al. A two-phase flow model for predicting cavitation dynamics. In *Fifth international conference on multiphase flow, Yokohama, Japan*, volume 152, 2004.

[25] Olivier Coutier-Delgosha, François Deniset, Jacques André Astolfi, and Jean-Baptiste Leroux. Numerical prediction of cavitating flow on a two-dimensional symmetrical hydrofoil and comparison to experiments. 2007.

[26] Robert F Kunz, Jules W Lindau, Todd A Kaday, and Leonard J Peltier. Unsteady rans and detached eddy simulations of cavitating flow over a hydrofoil. In *Fifth international symposium on cavitation, Osaka, Japan*, 2003.

[27] Mitja Morgut, Enrico Nobile, and Ignacijo Biluš. Comparison of mass transfer models for the numerical prediction of sheet cavitation around a hydrofoil. *International Journal of Multiphase Flow*, 37(6):620–626, 2011.

[28] Mu Zhang, Junjie Tan, Xinjian Ma, Wenjun Yi, and Zhihua Chen. The parallel algorithm study of cavitation flow on the hydrofoil. In *2011 International Conference on Computer Science and Service System (CSSS)*, pages 863–866. IEEE, 2011.

[29] Ehsan Roohi, Amir Pouyan Zahiri, and Mahmood Passandideh-Fard. Numerical simulation of cavitation around a two-dimensional hydrofoil using vof method and les turbulence model. *Applied Mathematical Modelling*, 37(9):6469–6488, 2013.

[30] B Ji, XW Luo, Roger EA Arndt, Xiaoxing Peng, and Yulin Wu. Large eddy simulation and theoretical investigations of the transient cavitating vortical flow structure around a naca66 hydrofoil. *International Journal of Multiphase Flow*, 68:121–134, 2015.

[31] Chao-Tsung Hsiao, Jingsen Ma, and Georges L Chahine. Multiscale tow-phase flow modeling of sheet and cloud cavitation. *International Journal of Multiphase Flow*, 90:102–117, 2017.

[32] Motohiko Nohmi, Tomoki Tsuneda, Byungjin An, Takayuki Suzuki, Satoshi Watanabe, and Shin-ichi Tsuda. Cavitation CFD Prediction for NACA0015 Hydrofoil Flow Considering Boundary Layer Characteristics. In *Proceedings of the 10th International Symposium on Cavitation (CAV2018)*. ASME Press, 12 2018.

[33] Gretar Tryggvason, Jiacai Lu, and Ming Ma. Direct Numerical Simulations of Gas-Liquid Flows. In *Proceedings of the 10th International Symposium on Cavitation (CAV2018)*. ASME Press, 12 2018.

[34] Lu Xing, Chunlin Wu, and Spyros A. Kinnas. VISVE, a Vorticity Based Model Applied to Cavitating Flow around a 2-D Hydrofoil. In *Proceedings of the 10th International Symposium on Cavitation (CAV2018)*. ASME Press, 12 2018.

[35] Robert T Knapp. Recent investigations of the mechanics of cavitation and cavitation damage. *Transactions of the ASME*, 77:1045–1054, 1955.

[36] Tatsuo Ito. An experimental investigation into the unsteady cavitation of marine propellers. *Papers of the Ship Research Institute, Tokyo, Japan, Paper No. 11*, 1966.

[37] Y Kawanami, Hiroharu Kato, Hajime Yamaguchi, M Tanimura, and Y Tagaya. Mechanism and control of cloud cavitation. 1997.

[38] DF De Lange and GJ De Bruin. Sheet cavitation and cloud cavitation, re-entrant jet and three-dimensionality. *Applied scientific research*, 58(1):91–114, 1997.

[39] TM Pham, Frédérique Larrarte, and Daniel H Fruman. Investigation of unsteady sheet cavitation and cloud cavitation mechanisms. *J. Fluids Eng.*, 1999.

[40] Mathieu Callenaere, Jean-Pierre Franc, Jean-Marie Michel, and Michel Riondet. The cavitation instability induced by the development of a re-entrant jet. *Journal of Fluid Mechanics*, 444:223, 2001.

[41] KR Laberteaux and SL Ceccio. Partial cavity flows. part 1. cavities forming on models without spanwise variation. *Journal of Fluid Mechanics*, 431:1, 2001.

[42] Jean-Baptiste Leroux, Olivier Coutier-Delgosha, and Jacques André Astolfi. A joint experimental and numerical study of mechanisms associated to instability of partial cavitation on two-dimensional hydrofoil. *Physics of fluids*, 17(5):052101, 2005.

[43] Olivier Coutier-Delgosha, B Stutz, Alexandre Vabre, and Samuel Legoupil. Analysis of cavitating flow structure by experimental and numerical investigations. *Journal of Fluid Mechanics*, 578:171, 2007.

[44] Matevž Dular, Rudolf Bachert, Christian Schaad, and Bernd Stoffel. Investigation of a re-entrant jet reflection at an inclined cavity closure line. *European Journal of Mechanics-B/Fluids*, 26(5):688–705, 2007.

[45] Evert-Jan Foeth, Tom van Terwisga, and Cas van Doorne. On the collapse structure of an attached cavity on a three-dimensional hydrofoil. *Journal of Fluids Engineering*, 130(7), 2008.

[46] YL Young and SA Kinnas. Analysis of supercavitating and surface-piercing propeller flows via bem. *Computational Mechanics*, 32(4):269–280, 2003.

[47] Harish Ganesh, Simo A Makiharju, and Steven L Ceccio. Bubbly shock propagation as a mechanism for sheet-to-cloud transition of partial cavities. *Journal of Fluid Mechanics*, 802:37–78, 2016.

[48] PF Pelz, T Keil, and TF Groß. The transition from sheet to cloud cavitation. *Journal of Fluid Mechanics*, 817:439–454, 2017.

[49] Frederick Stern, William S Vorus, et al. A nonlinear method for predicting unsteady sheet cavitation on marine propellers. *Journal of Ship Research*, 27(01):56–74, 1983.

[50] Neal E Fine. *Nonlinear analysis of cavitating propellers in nonuniform flow*. PhD thesis, Massachusetts institute of Technology, 1992.

[51] AC Mueller and Spyros A Kinnas. Propeller sheet cavitation predictions using a panel method. 1999.

[52] Paul Elliott Griffin. Computational techniques for the analysis and design of cavitating propeller blades. Master's thesis, University of Texas at Austin, 1998.

[53] Hanseong Lee, Spyros A Kinnas, et al. Application of a boundary element method in the prediction of unsteady blade sheet and developed tip vortex cavitation on marine propellers. *Journal of Ship Research*, 48(01):15–30, 2004.

[54] Spyros A Kinnas, Hanseong Lee, Thad J Michael, and Hong Sun. Prediction of cavitating waterjet propulsor performance using a boundary element method. Technical report, TEXAS UNIV AT AUSTIN, 2007.

[55] GNVB Vaz and Johan Bosschers. Modelling three dimensional sheet cavitation on marine propellers using a boundary element method. In *Sixth International Symposium on Cavitation CAV2006, Wageningen, The Netherlands*, 2006.

[56] Guilherme Nuno Vasconcelos Beleza Vaz, José Alberto Caiado Falcão de Campos, Johan Bosschers, Luís Rego da Cunha de Eça, and Instituto Superior de Ciências do Trabalho e da Empresa. *Modelling of sheet cavitation on hydrofoils and marine propellers using boundary element methods*. 2005.

[57] Shin Hyung Rhee, Takafumi Kawamura, and Huiying Li. Propeller cavitation study using an unstructured grid based navier-stoker solver. 2005.

[58] H Sun and SA Kinnas. Simulation of sheet cavitation on propulsor blades using a viscous/inviscid interactive method. In *CAV2006: Sixth International Symposium on Cavitation, Wageningen, The Netherlands*, 2006.

[59] Rickard E Bensow and Göran Bark. Implicit les predictions of the cavitating flow on a propeller. *Journal of fluids engineering*, 132(4), 2010.

[60] Bin Ji, Xianwu Luo, Yulin Wu, Xiaoxing Peng, and Hongyuan Xu. Partially-averaged navier–stokes method with modified k–ε model for cavitating flow around a marine propeller in a non-uniform wake. *International Journal of Heat and Mass Transfer*, 55(23-24):6582–6588, 2012.

[61] Chao-Tsung Hsiao, Jingsen Ma, and Georges L Chahine. Simulation of sheet and tip vortex cavitation on a rotating propeller using a multiscale two-phase flow model. In *Fourth int. symp. mar. propulsors, Austine, Texas, USA*, 2015.

[62] Weikang Du and Spyros Kinnas. Coupling a numerical optimization technique with a panel method or a vortex lattice method to design cavitating propellers in non-uniform inflows. In *Proceedings of the 10th International Symposium on Cavitation (CAV2018). ASME Press*, 2018.

[63] Naz Yilmaz, Batuhan Aktas, Mehmet Atlar, Patrick A. Fitzsimmons, and Mario Felli. An experimental and numerical investigation of propeller-rudder-hull interaction in the presence of tip vortex cavitation (tvc). *Ocean Engineering*, 216:108024, 2020.

[64] Keun Woo Shin and Poul Andersen. CFD Analysis of Propeller Tip Vortex Cavitation in Ship Wake Fields. In *Proceedings of the 10th International Symposium on Cavitation (CAV2018)*. ASME Press, 12 2018.

[65] Sang-Il Park, Seung-Jae Lee, Geuk-Sang You, and Jung-Chun Suh. An experimental study on tip vortex cavitation suppression in a marine propeller. *Journal of Ship Research*, 58(3):157–167, 2014.

[66] Young-Zehr Kehr, Huan-Jia Xu, Jui-Hsiang Kao, and Yan-Jhang Lin. Investigation of The Improvements in Cavitation And Pressure Pulse Due to The Endplate Propeller. In *Proceedings of the 10th International Symposium on Cavitation (CAV2018)*. ASME Press, 12 2018.

[67] Satoko Ando, Koyu Kimura, Kohei Segawa, and Koutaku Yamamoto. Study on the Hybrid Method of CFD and Bubble Dynamics for Marine Propeller Cavitation Noise Prediction. In *Proceedings of the 10th International Symposium on Cavitation (CAV2018)*. ASME Press, 12 2018.

[68] Nobuaki Sakamoto and Hikaru Kamiirisa. Propeller Cavitation Noise Radiated from Single and Twin-Screw Cargo Liners: CFD Prediction and Full Scale Validation. In *Proceedings of the 10th International Symposium on Cavitation (CAV2018)*. ASME Press, 12 2018.

[69] Huiping Fu and Jie Li. Hydro-Acoustic Analysis of the Cavitating Model Propeller PPTC in Oblique Flow. In *Proceedings of the 10th International Symposium on Cavitation (CAV2018)*. ASME Press, 12 2018.

[70] Kostas Belibassakis and Gerasimos Politis. Generation and propagation of noise from cavitating marine propellers. In *Proceedings of the Sixth International Symposium on Marine Propulsors*, 2019.

[71] Johan Bosschers. Propeller tip-vortex cavitation and its broadband noise. 2018.

[72] Artur K Lidtke, Thomas Lloyd, and Guilherme Vaz. Acoustic modelling of a propeller subject to non-uniform inflow. In *Proc. 6th Int. Symp. Mar. Propulsors, SMP'19, Rome, Italy*, 2019.

[73] Y Su, S Kim, W Du, SA Kinnas, M Grekula, J Hallander, and Q Li. Prediction of the propeller-induced hull pressure fluctuation via a potential-based method: Study of the rudder effect and the effect from different wake alignment methods. In *Proceedings of the 5th International Symposium on Marine Propulsors (SMP'17), Espoo, Finland*, pages 12–15, 2017.

[74] J Hur, H Kim, and H Lee. Numerical study on the effect of turbulence and cavitation model for propeller induced hull pressure fluctuation. In *Proceedings of the 10th International Symposium on Cavitation-CAV2018 Baltimore, Maryland, USA*, pages 834–837, 2018.

[75] Koichiro Shiraishi, Yuki Sawada, Daijiro Arakawa, and Kunihiro Hoshino. Experimental Estimation for Pressure Fluctuation on Ship Stern Induced by Cavitating Propeller Using Cavity Shape Measurements. In *Proceedings of the 10th International Symposium on Cavitation (CAV2018)*. ASME Press, 12 2018.

[76] Seungnam Kim and Spyros A Kinnas. Prediction of unsteady developed tip vortex cavitation and its effect on the induced hull pressures. *Journal of Marine Science and Engineering*, 8(2):114, 2020.

[77] Jui-Hsiang Kao et al. An improved numerical computation of hull pressure fluctuations due to unsteady sheet cavitation of a propeller. *Journal of Ship Research*, 62(03):101–114, 2018.

[78] Takashi Kanemaru, Daijiro Arakawa, Yuki Sawada, Koichiro Shiraishi, and Jun Ando. A practical calculation method of pressure fluctuation induced by cavitating propeller using a simple surface panel method "sqcm".

[79] Stephan Berger. *Numerical analysis of propeller-induced higher-order pressure fluctuations on the ship hull*. Technische Universität Hamburg-Harburg, 2018.

[80] Jin-Keun Choi and Georges L Chahine. Experimental and numerical study of cavitation erosion resistance of a polyurea coating layer. In *Fourth International Symposium on Marine Propulsors SMP*, volume 15, 2015.

[81] Themistoklis Melissaris, Norbert Bulten, and Tom van Terwisga. On Cavitation Aggressiveness and Cavitation Erosion on Marine Propellers using a URANS Method. In *Proceedings of the 10th International Symposium on Cavitation (CAV2018)*. ASME Press, 12 2018.

[82] John Nicholas Newman. *Marine hydrodynamics*. The MIT press, 2018.

[83] Jack Moran. *An introduction to theoretical and computational aerodynamics*. Courier Corporation, 2003.

[84] Joseph Katz and Allen Plotkin. *Low-speed aerodynamics*, volume 13. Cambridge university press, 2001.

[85] Robert M Pinkerton. *Calculated and measured pressure distributions over the midspan section of the NACA 4412 airfoil*, volume 563. NACA, 1936.

[86] Ira H Abbott and Albert E Von Doenhoff. *Theory of wing sections: including a summary of airfoil data*. Courier Corporation, 2012.

[87] Ion Paraschivoiu. *Aérodynamique subsonique*. Presses inter Polytechnique, 1998.

[88] John Carlton. *Marine propellers and propulsion*. Butterworth-Heinemann, 2018.

[89] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical Recipes in Fortran 90: Numerical recipes in Fortran 77V. 2. Numerical recipes in Fortran 90*. Cambridge University Press, 1996.

Appendix A

FORTRAN Code Listing and usage instructions

```fortran
!     Last change:  Antonis Kritikos, 19, January 2021
!     ****************************************************************
!     *                      PROGRAM Caviation cvt                   *
!     *     MORINO PANEL METHOD FOR SINGLE-ELEMENT LIFTING           *
!     *     AIRFOIL IN TWO DIMENSIONAL PARTIALLY CAVITATING INCOMPRESSIBLE FLOW   *
!     *                                                              *
!     ****************************************************************
include "nrtype.f90"
include "mod_naca_geometry.f90"
include "nr.f90"

include "nrutil.f90"

include "ludcmp.f90"
include "lubksb.f90"
include "lfit.f90"
include "gaussj.f90"
include "covsrt.f90"

program source_dipole_cavitation
use naca_geometry
use nrtype
use nr, ONLY: spline,splint,ludcmp,lubksb,lfit
!use nrutil
implicit none
real(sp)::ld,lc,af,ni,lamda,d,qc,sl,st
real(sp),dimension(4):: la
real(sp),allocatable, dimension(:):: dli,vtang
INTEGER(I4B), allocatable, DIMENSION(:):: indx
integer:: nlower,nupper, naca_m,naca_p,naca_t,ninterp,m,ii,iter
integer:: npanels,nnodes, neqs,nc,nw,ncav,pointA,pointT,pointL,nwet
real(sp)::    angle_of_attack,vinf,chord, cosalf,sinalf,chisq
real(sp),allocatable,dimension(:):: sf,fsf
!
real(sp),allocatable,dimension(:)::x,y,costhe,sinthe,cp,q,dist,sr
real(sp),allocatable,dimension(:,:):: f_log,f_b,covar  ! auxiliary data
real(sp),allocatable,dimension(:,:):: a2,a1,D1,Dstar,S ! linear system coefficients
real(sp),allocatable,DIMENSION(:):: xp,yp,xcav,ycav,dsc,coscav,sincav,b,b2,xa,ya,sig,ac,acav,sigc
logical(lgt), allocatable,dimension(:):: maska
logical(lgt):: flagA
!
OPEN(5,FILE= 'in_bem')
OPEN(6,FILE= 'out_bem')
open(17, file= 'sigma-conv.dat')
!pi   =   4.*ATAN(1.)
!
WRITE(*,10)
WRITE(6,10)
10 FORMAT(//&
'============================program_cvt_==========================='/ &
'_Morino_potential_panel_method_for_single_element_lifting_airfoil_in_____'/ &
'_two_dimensional_partially_cavitating_incompressible_flow._____'/ &
'_Automatic_data_generation_for_NACA_4_digit_series_is_included._____'/ &
'_Reference:_J.Moran,_Introduction_to_Theoretical_and_Computational_Aerodynamics'/ &
'_Notes:_____'/ &
'_Input_data_are_written_in_file:__in_bem,_ingeom_____'/ &
'_Output_data_are_written_in_file:_out_bem_____'/ &
'_Also:_out_geom.dat:_section_geometrical_data,__out_pres.dat:_pressure_results_'/ &
'_Data_for_TECPLOT_are_created_in_file:_tecplot_geometry_and_pressures.dat_____'/ &
'==================================================================='/)
!
m=6
ALLOCATE(ac(m),maska(m),covar(m,m),acav(m))
maska(1:m) =.true.
CALL read_input    ! read panel number and NACA data
ALLOCATE(x(nnodes),y(nnodes), costhe(npanels),sinthe(npanels),xcav(ncav+1),ycav(ncav+1),dsc(ncav))
```

```fortran
allocate(coscav(ncav),sincav(ncav),dist(npanels))
allocate(sf(ncav),fsf(ncav))
CALL create_boundary_element_geometry
neqs = 2*npanels+1
write(6,*) 'pointL,pointA,pointT,ncav', pointL,pointA,pointT,ncav
ALLOCATE(f_log(npanels,npanels),f_b(npanels,npanels+1),a2(neqs,neqs),b(neqs),a1(neqs,neqs))
allocate(D1(npanels,npanels),Dstar(npanels,npanels),S(npanels,npanels),indx(neqs),b2(neqs))
ALLOCATE(xp(npanels),yp(npanels),cp(nnodes),q(2*npanels),sr(npanels))
allocate(dli(npanels),vtang(npanels))
!

DO ii=1,iter
flagA=.false.
DO

call find_linear_system_coefficients
call ludcmp(a2(1:neqs,1:neqs),indx,d) !LU decomposition
call lubksb(a2,indx,b) ! solution of the linear system
call find_foil_velocities_and_pressures
call ANR
IF (flagA) EXIT


ENDDO
call cavshape
write(17,*) ii,-cp(pointA+1)
ENDDO

call find_forces_and_moments


allocate(sigc(ncav+1))
sigc=1.
call lfit(xcav,ycav,sigc,acav,maska,covar,chisq,fff)
call tecplot_geometry_and_pressures
!
call tecplot_field_point_velocities_and_pressures

write(*,*) af
WRITE(*,*) ' <program Cavitation_BEM>, OK'

!-----------------------------
CONTAINS
!-----------------------------
!
!
!-----------------------------

     subroutine read_input
      use nrtype
      use nr, ONLY: spline
      ! read input data and calculate parameters for subroutine body
      ! set parameters of body shape and panel distribution
      ! input
      !       nupper no.of panels on upper surface
      !       naca
      read(5,*) nlower,nupper    ! nlower,nupper no.of panels on lower and upper foil surface
      read(5,*) angle_of_attack  ! input angle of attack in degrees
      read(5,*) vinf             ! velocity at infinity (m/s)
      read(5,*) chord            ! input foil chord in (m)
      read(5,*) naca_m,naca_p,naca_t  ! naca series hydrofoil number, example: 4 4 12 (for NACA 4412)
      read(5,*) ld,lc,lamda,af,ni
      read(5,*) iter
      write(*,11)  nlower,nupper,angle_of_attack,vinf,naca_m,naca_p,naca_t,chord
      write(6,11)  nlower,nupper,angle_of_attack,vinf,naca_m,naca_p,naca_t,chord
      11 format ( '================================================='/          &
            ' no. of panels on lower surface nlower = ',i5/              &
            ' no. of panels on upper surface nupper = ',i5/              &
            ' angle of attack (deg)                = ',f7.2/             &
            ' velocity at infinity (m/s)           = ',f7.2/             &
            ' naca series hydrofoil number, NACA  = ',2i1,i2/            &
            ' chord (m)                           = ',f7.3/              &
            '=========================================='/)
      !
      npanels = nlower+nupper
      nnodes  = npanels+1
      ncav=floor((lc-ld)*npanels/2.)
      nwet=npanels-ncav
      write(*,12)  npanels,nnodes
      write(6,12) npanels,nnodes,ncav
      12 format ( ' ==========================================='/          &
            ' total number of panels    = ',i5/                    &
            ' total number of nodes     = ',i5/                    &
            ' total number of cavity points = ',i5                 &
            '==========================================='/)
      !
      cosalf  =cos(angle_of_attack*pi/180.)
      sinalf  =sin(angle_of_attack*pi/180.)
      call naca1(ninterp,xa,ya,sig)
      !      allocate(y2(ninterp)) ! if user prefers  spline geometry
      !      call spline(xa,ya,yp1,ypn,y2)
      call lfit(xa,ya,sig,ac,maska,covar,chisq,fff)
            !
      WRITE(*,*) ' <read_input>, OK'
      endsubroutine read_input
```

```fortran
  !-------------------------------
subroutine create_boundary_element_geometry
   !use nrtype
   use nr, ONLY: splint
   integer:: npoints,nstart,n1,n2,j
   integer:: nsurf,i,n,nside !nu=n points used for every face
   real(sp):: sign,sc,fract1,dx,dy,fract2,xmid,xmid1,ymid,ymid1,ds
   ! set coordinates of body nodes, start with lower surface
   npoints  = nlower
   nc=floor((lc-ld)*npoints)
   nw=npoints-nc

   sign     =-1.0
   nstart   = 0
   ! loop on lower and upper foil nodes
   do nsurf=1,2
   do n=1,npoints
   IF (nsurf==1) THEN
   n1=n
   n2=n-nw
   else
   n1=n
   n2=n-nc
   ENDIF

   fract1   = float(n1-1)/float(nc)
   fract2   = float(n2-1)/float(nc)
   ! fract   = float(n-1)/float(npoints)
   !     if(nsurf==1) sc     = 0.5*(1.+cos(pi*fract))
   !     if(nsurf==2) sc     = 0.5*(1.-cos(pi*fract))
   IF (((n<=nw).and.(nsurf==1)).or.((n<=nc).and.(nsurf==2))) THEN
   nside=1
   else
   nside=0
   ENDIF
   IF(nsurf==1) sc       = (0.5*(1-lc)*(1.+cos(pi*float(n1-1)/float(nw)))+lc)*nside+ &
   & + 0.5*lc*(1.+cos(pi*float(n2-1)/float(nc)))*(1-nside)
   IF(nsurf==2) sc       = (0.5*(1-lc)*(1.-cos(pi*float(n2-1)/float(nw)))+lc)*(1-nside)+ &
   & + 0.5*lc*(1.-cos(pi*float(n1-1)/float(nc)))*nside
   i        = nstart+n
   ! [calculate coordinates of node i using iso-cosine chord spacing
   ! between leading edge-cavity detachment-trailing edge for both faces]
   IF (naca_t>0.) THEN
      CALL naca_coordinates(naca_m,naca_p,naca_t,chord,sc,sign, x(i),y(i))
      else
       x(i)=sc
       call comp(x(i),y(i),ac,fff,sign)
   ENDIF
   enddo

   ! end loop on nodes on lower face, upper face next:
   npoints  = nupper
   sign     = 1.0
   nstart   = nlower
   enddo
   ! end loop on foil nodes
   x(nnodes)=x(1)  ! nodes should define a closed contour
   y(nnodes)=y(1)
   !     ------------set slopes of panels, loop on panels-----------------
   DO i=1,npanels
   dx=x(i+1)-x(i)
   dy=y(i+1)-y(i)
   dist(i) = sqrt(dx*dx+dy*dy)
   sinthe(i)=dy/dist(i)
   costhe(i)=dx/dist(i)
   ENDDO
   ! write results:
   OPEN(14,FILE='out_geom.dat')
   OPEN(12,FILE='out_geom_cav.dat')
   !write(*,10)
   !write(14,10)
10 format('  body shape, nodal points'/'  i',4x,'x',9x,'y')
   DO i=1,nnodes
   write(14,11)  x(i),y(i)
   write(*,11)  x(i),y(i)
   ENDDO
   CLOSE(14)
11 format(2f10.6)
   !===============================================================
   !======== SOS definition of "boundary" points===============!!!!
   !===============================================================
   n=npanels
   pointA = 0.5*n+floor(ld*n*0.5)+1
   pointT = 0.5*n+floor((1-lamda)*lc*n*0.5)
   pointL=0.5*n+floor(lc*n*0.5)+1
   j=1
   DO i=pointA,pointL
   xcav(j) = x(i)
   ycav(j) = y(i)

   j=j+1
   ENDDO
   DO j=1,ncav
   dx=xcav(j+1)-xcav(j)
```

```fortran
        dy=ycav(j+1)-ycav(j)
        xmid=.5*(xcav(j)+xcav(j+1))
        ymid=.5*(ycav(j)+ycav(j+1))
        IF (j==1) THEN
        xmid1=xcav(1)
        ymid1=ycav(1)
        ELSE
        xmid1=.5*(xcav(j-1)+xcav(j))
        ymid1=.5*(ycav(j-1)+ycav(j))
        ENDIF

        dsc(j)   = sqrt((xmid1-xmid)*(xmid1-xmid)+(ymid1-ymid)*(ymid1-ymid))
        ds = sqrt(dx*dx+dy*dy)
        coscav(j)=dx/ds
        sincav(j)=dy/ds
        WRITE(12,*) xcav(j),ycav(j),dsc(j)
        ENDDO


        WRITE(*,*) ' <create_boundary_element_geometry>, OK'
      ENDSUBROUTINE create_boundary_element_geometry

subroutine find_linear_system_coefficients
 integer:: i,j,j1,k,nd
 ! internals:
 real(sp):: xcp,ycp,dxj,dyj,dxj1,dyj1
 real(sp):: x2t,y2t,dx,dz,th,X1,X2,Z,Z2,R1,R2,th1,th2
 real(sp):: inv2pi,sum1,rhs
 real(sp):: xcavcp,ycavcp

 real(sp),dimension(4):: t
 real(sp), allocatable, dimension(:):: F,dpin,f1
 logical(lgt):: flag1
  real(sp):: s0,s1,s2,t1,t2

 allocate(F(ncav),dpin(ncav),f1(npanels))
  ! calculate coefficients of the linear system
  a2  = 0. ! initialize coefficients
  b = 0.
  b2=0.
  fsf = 0.
  sf=0.
  flag1=.true.
  nd=pointA
  !

  inv2pi  = 1./(pi)
  OPEN(25, file='linear coeff.dat')
  open(27, file='fsf.dat')
  write(27,*) ii

  10 format(2i5,f10.6)

   != lagrange interpolation for phi(0), taken from Neal Fine phd

        t(1)=0.
        DO i=2,4
        t(i)=t(i-1)+0.5*(dist(pointA-6+i)+dist(pointA-5+i))
        ENDDO
        s0=t(4)+0.5*dist(pointA-1)
        s1=t(4)
        s2=t(3)

        t1=(s2-2*s0)/(s2*s2-3*s0*s0)
        t2=1/(s2**3-3*s0**2*s2)
        r2=s1/s2*(s1**2-3*s0**2)/(s2**2-3*s0**2)
        r1=s2*r2*(s2-2*s0)-s1**2+2*s0*s1
         la(1)=s0**2/r1*(1-2*s0*t1)
         la(2)=-s0**2/r1*(2*s0*r1*t2-2*s0*t1*r2+r2)
         la(3)=1+s0**2/r1*(2*r1*t2*s0+2*t1*s0*(1-r2)+r2-1)
         la(4)=2*s0 **3*(t1/r1*(s1-s2*r2)+s2*t2)-s0**2/r1*(s1-s2*r2)+s0

  ! compute cavity length of each point and cavity termination function f

        sf(1)=dsc(1)
        DO j1=2,ncav
        xcavcp = 0.5*(xcav(j1)+xcav(j1+1))
        !ycavcp = 0.5*(ycav(j1)+ycav(j1+1))
        sf(j1)=sf(j1-1)+dsc(j1)

        IF ((xcavcp>=lc-lamda).and.flag1) THEN
        st=sf(j1)
        flag1=.false.
        ENDIF
        ENDDO


        sl=sf(ncav)+0.5*dist(pointA+ncav-1)

        DO j1=1,ncav
        IF (sf(j1)>=st) THEN
        fsf(j1)=af*((sf(j1)-st)/(sl-st))**ni

        ENDIF
        write(27,*)  xcav(j1), dsc(j1),sf(j1),sl,st,fsf(j1)
        ENDDO
```

```fortran
! Fj, delta phi j definition
      DO j=1,ncav
      sum1=0.
      xcavcp = 0.5*(xcav(j)+xcav(j+1))
      ycavcp = 0.5*(ycav(j)+ycav(j+1))

      DO k=1,j
      sum1 = sum1+(1-fsf(k))*dsc(k)
      ENDDO
      F(j)=sum1
      dpin(j)=vinf*((xcavcp-x(pointA))*cosalf+(ycavcp-y(pointA))*sinalf)
      write(25,*) xcav(j),F(j),dpin(j)
      ENDDO

 !=================\ INFLUENCE COEFFICIENTS \==============
 DO i=1,npanels
  xcp  =.5*(x(i)+x(i+1))    ! control point position
  ycp  =.5*(y(i)+y(i+1))
  rhs=0.
  f_b(i,npanels+1) = atan(ycp,(x(1)-xcp))*inv2pi

  sum1=0.

  DO j=1,npanels
   dxj     =xcp-x(j)
   dyj     =ycp-y(j)
   dxj1    =xcp-x(j+1)
   dyj1    =ycp-y(j+1)
   x2t    = x(j+1)-x(j)
   dx     = x2t
   y2t    = y(j+1)-y(j)
   dz     = y2t
   th     = atan(dz,dx)

   X1= dxj*cos(th)+dyj*sin(th)
   Z = -dxj*sin(th)+dyj*cos(th)
   X2= x2t*cos(th)+y2t*sin(th)
   Z2=0.

   R1=SQRT(X1*X1+Z*Z)
   R2=SQRT((X1-X2)**2+Z*Z)


   th1=atan(Z,X1)
   th2=atan(Z,X1-X2)

   IF(i/=j) THEN
   f_b(i,j)   =   atan2(dyj1*dxj-dxj1*dyj,dxj1*dxj+dyj1*dyj)*inv2pi
   ELSE
   f_b(i,j)   = -0.5*2
   ENDIF

   IF(j==1) THEN
    D1(i,j)       =  f_b(i,j)-f_b(i,npanels+1)
    ELSE IF(j==npanels) THEN
    D1(i,j)       =  f_b(i,j)+f_b(i,npanels+1)
    ELSE
    D1(i,j)       =  f_b(i,j)
   END IF

    !source influence coefficient
   IF (j==i) THEN
   !S(i,j) = 2*X1*log(R1)*inv2pi !
   S(i,j) = (dist(j)*log(dist(j)/2)-dist(j))*inv2pi
   ELSE
   S(i,j) = inv2pi*(X1*log(R1)-(X1-X2)*log(R2)-X2+Z*(th2-th1)) !
   ENDIF
   a2(i,j)=D1(i,j)
   a2(i,j+npanels)=S(i,j)
  ENDDO
  ENDDO


    ! ==============\ KINEMATIC BC & DYNAMIC BC\==============
    DO   i=npanels+1,2*npanels
    j=i-npanels
    IF ((j<pointA).or.(j>=pointL)) THEN
    a2(i,i)=1.
    b(i)=vinf*(cosalf*sinthe(j)-sinalf*costhe(j))
    ELSE
    DO k=1,3

    a2(i,nd-k)= la(k)
    ENDDO

    j1=j-pointA+1 !
    a2(i,j)=-1.
    a2(i,2*npanels+1)=F(j1)+la(4)  !
    b(i)=dpin(j1)  !

    ENDIF


    ENDDO
```

```fortran
      !===== cavity closure condition =============
      sum1=0.
      i=2*npanels+1
      DO j=pointA,pointL-1
      j1=j-pointA+1
      xcp  =.5*(xcav(j1)+xcav(j1+1))
      ycp  =.5*(ycav(j1)+ycav(j1+1))
      a2(i,j+npanels)=dsc(j1)/(1.-fsf(j1))
      sum1=sum1+vinf*dsc(j1)*(-cosalf*sincav(j1)+sinalf*coscav(j1))/(1.-fsf(j1))
      ENDDO
      b(i) = -sum1
      write(6,*) 'b_closure=',b(i)

a1=a2
b2=b
open(8,file='linear-coefficients.dat')
 DO i=1,neqs
 DO j=1,neqs
 WRITE(8,11) i,j,a2(i,j)
 ENDDO
 ENDDO
 11 format(i3,2x,i3,2x,2f12.7)




 WRITE(*,*) '_<find_linear_system_coefficients>,_OK'
endsubroutine find_linear_system_coefficients



      !------------------------------
    subroutine find_foil_velocities_and_pressures
      ! calculate pressure distribution
      ! locals:
      integer:: i
      real(sp):: xmid,ymid,xmid1,ymid1

      !
      ! retrieve solution from a-matrix:
      OPEN(26, file='solution.dat')
      do i=1,npanels
      xmid=.5*(x(i)+x(i+1))
      ymid=.5*(y(i)+y(i+1))
      q(i) = b(i)+vinf*(xmid*cosalf+ymid*sinalf)
      sr(i)=b(i+npanels)
      write(26,*) xmid,q(i),sr(i)
      ENDDO
      qc=b(neqs)
      write(6,*) 'qc=',qc

      ! find tangential velocity vtang and cp at midpoint of ith panel
       do i=1,npanels-1    ! loop on control points
      xmid=.5*(x(i)+x(i+1))
      ymid=.5*(y(i)+y(i+1))
      xmid1=.5*(x(i+1)+x(i+2))
      ymid1=.5*(y(i+1)+y(i+2))
      dli(i+1)   = sqrt((xmid1-xmid)*(xmid1-xmid)+(ymid1-ymid)*(ymid1-ymid))
      vtang(i+1) = (q(i+1)-q(i))
      vtang(i+1)= vtang(i+1)/dli(i+1)
      cp(i+1)  = 1.-vtang(i+1)*vtang(i+1)/(vinf*vinf)
      xp(i+1)  = x(i+1)
      yp(i+1)  = y(i+1)
      ENDDO
      cp(1)=(cp(npanels)+cp(2))/2
      cp(nnodes)=cp(1)
      vtang(1)=(vtang(npanels)+vtang(2))/2
      xp(1)=x(1)
      yp(1)=y(1)


      !
      ! write results:
      OPEN(15,FILE='out_pres.dat')
      write(*,10)
      write(15,10)
      10 format('__body_pressures_at_control_points',/'__i_mid',3x,'x',9x,'y',9x,'Cp',9x,'dl',9x,'vt')
      DO i=2,npanels-1
      write(15,11) i, xp(i),yp(i),-cp(i),dli(i),vtang(i)
      !write(*,11)  i, xp(i),yp(i),cp(i)
      ENDDO
      CLOSE(15)
      11 format(i3,5x,3(f10.4,2x),f10.8,2x,f10.6)
      !
      IF(nlower==nupper) THEN
      OPEN(15,FILE='out_pres_dif.dat')
      write(15,12)
      12 format('__chordwise_pressure_differences_'/'__i_mid',3x,'xmid',6x,'(Cp+)-(Cp-)')
      DO i=1,nlower
      write(15,13) i,(xp(nlower+1-i)+xp(nlower+i))/2 , cp(nlower+1-i)-cp(nlower+i)
      ENDDO
      CLOSE(15)
      13 format(i3,5x,2f10.6)
      ENDIF
      WRITE(*,*) '_<find_foil_velocities_and_pressures>,_OK'
```

```fortran
ENDSUBROUTINE find_foil_velocities_and_pressures


subroutine find_forces_and_moments
  ! calculate and print cl,cd and cm
  ! locals:
  integer:: i
  real(sp):: cfx,cfy,cm,cm_ac,xmid,ymid,dx,dy ,cl,cd,cp1
  cfx = 0.
  cfy = 0.
  cm  = 0.
  do i=1,npanels
  xmid =.5*(x(i)+x(i+1))
  ymid =.5*(y(i)+y(i+1))
  dx   = x(i+1)-x(i)
  dy   = y(i+1)-y(i)
  cp1  = (cp(i)+cp(i+1))/2.
  cfx  = cfx + cp1*dy
  cfy  = cfy - cp1*dx
  cm   = cm  + cp1*(dx*xmid-dy*ymid)
  ENDDO
  cfx  = cfx/chord
  cfy  = cfy/chord
  cl   = cfy*cosalf-cfx*sinalf
  cd   = cfx*cosalf+cfy*sinalf
  cm   = cm/(chord*chord)
  cm_ac= cm + 0.25*cfy
  write(*,10) angle_of_attack,vinf,chord,cl,cd,cm,cm_ac
  write(6,10) angle_of_attack,vinf,chord,cl,cd,cm,cm_ac
  10 format((//'=================================================='/   &
            '␣␣␣␣␣␣i␣n␣t␣e␣g␣r␣a␣t␣e␣d␣␣␣c␣o␣e␣f␣f␣i␣c␣i␣e␣n␣t␣s␣␣␣␣␣'/   &
            '=================================================='/   &
            '␣␣␣␣␣␣␣␣␣␣␣␣alpha␣␣␣␣␣␣=␣',f10.3,'␣degrees'/   &
            '␣␣␣␣␣␣␣␣␣␣␣␣vinf␣␣␣␣␣␣=␣',f10.3,'␣m/s'/   &
            '␣␣␣␣␣␣␣␣␣␣␣␣chord␣␣␣␣␣␣=␣',f10.3,'␣m'/   &
            '␣␣␣␣␣␣␣␣␣␣␣␣Clift␣␣␣␣␣␣=␣',f10.5,/   &
            '␣␣␣␣␣␣␣␣␣␣␣␣Cdrag␣␣␣␣␣␣=␣',f10.5,/   &
            '␣␣␣␣␣␣␣␣␣␣␣␣Cmoment_le_=␣',f10.5,/   &
            '␣␣␣␣␣␣␣␣␣␣␣␣Cmoment_ac_=␣',f10.5,/   &
            '==================================================')
  !
  WRITE(*,*) '␣<find_forces_and_moments>,␣OK'
 endsubroutine find_forces_and_moments
  !
  !
  !------------------------------

  ! find field point velocities and pressure coefficients at control point (xcp,ycp)
subroutine field_point_velocities_and_pressures(xcp,ycp,u,v,cp)
  real(sp),INTENT(in):: xcp,ycp
  real(sp),INTENT(out):: u,v,cp
  ! locals:
  integer:: j
  real(sp):: inv2pi,dz,dx,th
  real(sp)::xt,yt,y2t,x2t,X1,X2,Z2,Z,R,R1,R2,ULd,WLd,UXd,Wd,us_star,vs_star,us,vs
  real(sp)::dxj,dyj,dxj1,dyj1,f_log,f_b

  !
  inv2pi  = 1./(2*pi)
  u    = vinf*cosalf
  v    = vinf*sinalf
  DO j=1,npanels  ! loop on boundary elements, (xcp,ycp) should be outside body!
  !q(j) = b(j)
  xt     = xcp-x(j)
  yt     = ycp-y(j)
  x2t    = x(j+1)-x(j)
  dx     = x2t
  y2t    = y(j+1)-y(j)
  dz     = y2t
  th     = atan2(dz,dx)
  X1= xt*cos(th)+yt*sin(th)
  Z = -xt*sin(th)+yt*cos(th)
  X2= x2t*cos(th)+y2t*sin(th)
  Z2=0
  R1=SQRT(X1*X1+Z*Z)
  R2=SQRT((X1-X2)*(X1-X2)+Z*Z)
  dxj     =xcp-x(j)
  dyj     =ycp-y(j)
  dxj1    =xcp-x(j+1)
  dyj1    =ycp-y(j+1)
  f_log    = .5*log((dxj1*dxj1+dyj1*dyj1)/(dxj*dxj+dyj*dyj))*inv2pi
  f_b      = atan2(dyj1*dxj-dxj1*dyj,dxj1*dxj+dyj1*dyj)*inv2pi
  us_star  = -f_log
  vs_star  =  f_b
  ! COMPUTE THE VELOCITY INDUCED AT THE ITH
  ! COLLOCATION POINT BY THE JTH PANEL
  ULd = inv2pi*Z*(1/(R1*R1)-1/(R2*R2))
  WLd = -inv2pi*(X1/(R1*R1)-(X1-X2)/(R2*R2))
  UXd =  ULd*cos(-th)+WLd*sin(-th)
  Wd  = -ULd*sin(-th)+WLd*cos(-th)
  us       =  us_star*costhe(j)-vs_star*sinthe(j)
  vs       =  us_star*sinthe(j)+vs_star*costhe(j)
  !b1  = sr(j)
  u  = u+q(j)*UXd+us*sr(j)
  v  = v+q(j)*Wd+vs*sr(j)
  ENDDO
```

```fortran
   ! influence of wake panel
   R=SQRT((xcp-x(1))*(xcp-x(1)) +(ycp-y(1))*(ycp-y(1)))
   UXd=inv2pi*(ycp/(R*R))
   Wd=-inv2pi*(xcp-x(1))/(R*R)
   u=u+UXd*(q(npanels)-q(1))
   v=v+Wd*(q(npanels)-q(1))

   !
   cp  = 1.-(u*u+v*v)/(vinf*vinf)
   endsubroutine field_point_velocities_and_pressures



   !=================================================

 SUBROUTINE tecplot_geometry_and_pressures
   ! locals
   integer::i
   OPEN(20, file='tecplot_geometry_and_pressures.dat')
   write(20,10)
   10 FORMAT('TITLE_=_"_plot_foil_geometry,_pressure_coefficients_or_q"' )
   write(20,11)
   11 FORMAT('VARIABLES_=_"Xmid","Ymid","-Cp","q"_')
   WRITE(20,12) npanels
   12 FORMAT('_ZONE_T="__plot_geometry,_pressure_coefficients_or_q_",_I=',I5,',_F=POINT_' )
   DO i =1,npanels
   WRITE(20,13) x(i), y(i),-cp(i),q(i)
   13 FORMAT(4(1x,f12.5))
   ENDDO
   !
   CLOSE(20)
   WRITE(*,*)  '_<tecplot_geometry_and_pressures>,_OK'
 ENDSUBROUTINE tecplot_geometry_and_pressures


 SUBROUTINE tecplot_field_point_velocities_and_pressures
   ! locals
   integer:: i,j,x_points,y_points_face,y_points_back,x_cav,y_points_cav
   real(sp):: x_start,x_end,y_start,y_end,dx,dy, xcp,ycp,u,v,cp
   real(sp):: sc,sign, xface,yface, xback,yback,yc
   !
   x_points      = 300
   y_points_face = 40
   y_points_back = 40
   OPEN(21, file='tecplot_field_point_velocities_and_pressures.dat')
   write(21,10)
   10 FORMAT('TITLE_=_"_plot_field_point_velocities_and_pressures"' )
   write(21,11)
   11 FORMAT('VARIABLES_=_"Xmid","Ymid","UX","VX","CP"_')
   !
   x_start  = -chord/2.
   x_end    = 3.*chord/2
   y_start  = -chord/2.
   y_end    =  chord/2.
   !
   dx   = (x_end-x_start)/(x_points-1)
   !
   ! face grid:
   WRITE(21,12) y_points_face,x_points
   12 FORMAT('_ZONE_T="__face_grid_",_I=',I5,',_J=',I5,',_F=POINT_' )
   DO i=1,x_points
   xcp     = x_start+(i-1)*dx
   yface  = 0.
   IF(xcp>0.and.xcp<chord) THEN
   sc   = xcp/chord
   sign = -1.
   !CALL naca_coordinates(naca_m,naca_p,naca_t,chord,sc,sign, xface,yface)
   call comp(sc,yface,ac,fff,sign)
   xface=sc
   yface =yface-chord/300. ! avoid coincidence of field points with foil face
   xcp   =xface
   ENDIF
   dy   = (yface-y_start)/(y_points_face-1)
   DO j=1,y_points_face
   ycp  = y_start+(j-1)*dy
   call field_point_velocities_and_pressures(xcp,ycp,u,v,cp)
   WRITE(21,15) xcp,ycp,u,v,cp
   15 FORMAT(5(1x,f12.5))
   ENDDO
   ENDDO

   ! back grid:
   WRITE(21,13) y_points_back,x_points
   13 FORMAT('_ZONE_T="__back_grid_",_I=',I5,',_J=',I5,',_F=POINT_' )
   DO i=1,x_points
   xcp    = x_start+(i-1)*dx
   yback  = 0.
   IF(xcp>0.and.xcp<chord) THEN
   sc   = xcp/chord
   sign =  1.
   !CALL naca_coordinates(naca_m,naca_p,naca_t,chord,sc,sign, xback,yback)
   call comp(sc,yback,ac,fff,sign)
   call comp(sc,yc,acav,fff,sign)
   IF ((xcp>0).and.(xcp<x(pointL))) THEN
    yback=yc
```

```fortran
     ENDIF
   xback=sc
   yback =yback+chord/300. ! avoid coincidence of field points with foil back
   xcp   =xback
   ENDIF
   dy    = (y_end-yback)/(y_points_back-1)
   DO j=1,y_points_back
   ycp   = yback+(j-1)*dy

   call field_point_velocities_and_pressures(xcp,ycp,u,v,cp)
   WRITE(21,15) xcp,ycp,u,v,cp
   ENDDO
   !
   ENDDO
   !
   ! cavity
   x_start  = x(pointA+1)
   x_end    = x(pointL-1)
   y_points_cav=20
   x_cav=100
   dx    = (x_end-x_start)/(x_cav-1)
   WRITE(21,14) y_points_cav,x_cav
14 FORMAT(' ZONE T=" cavity grid ", I=',I5,', J=',I5,', F=POINT ' )
   DO i=1,x_cav
   xcp    = x_start+(i-1)*dx
   yback  = 0.
   sc    = xcp/chord
   sign  =  1.
   !CALL naca_coordinates(naca_m,naca_p,naca_t,chord,sc,sign, xback,yback)
   call comp(sc,yback,ac,fff,sign)
   call comp(sc,yc,acav,fff,sign)
   xback=sc
   yback =yback+chord/300
   xcp   =xback

   dy    = (yc-yback)/(y_points_cav-1)
   DO j=1,y_points_cav
   ycp   = yback+(j-1)*dy

   call field_point_velocities_and_pressures(xcp,ycp,u,v,cp)
   cp=1-(qc/vinf)**2
   WRITE(21,15) xcp,ycp,u,v,cp
   ENDDO
   !
   ENDDO


   CLOSE(21)
   WRITE(*,*)  ' <tecplot_field_point_velocities_and_pressures>, OK'
 ENDSUBROUTINE tecplot_field_point_velocities_and_pressures

   !
   !=================================================
 FUNCTION kd(i,j)

    IMPLICIT NONE
    integer:: i,j
    integer:: kd
    kd=0
    IF (i==j) THEN
    kd=1
    ENDIF

   END FUNCTION kd
   !=================================================


   !=================================================
 subroutine cavshape

    REAL(sp)::  Pin,sum1,dx,dy,xmid,xmid1,ymid,ymid1,ds
    real(sp),dimension(ncav):: hc
    integer:: j,j1,k


    open(27,file='hc.dat')

    DO j=1,ncav

    sum1=0.
    DO k=1,j
    Pin=vinf*(-cosalf*sincav(k)+sinalf*coscav(k)) !-vinf*(cosalf*sinthe(i)-sinalf*costhe(i))
    sum1 = sum1+(sr(k+pointA-1)+Pin)/(1.-fsf(k))*dsc(k)
    ENDDO
    hc(j)=sum1/qc
    write(27,*)  ii, xcav(j),hc(j)
    ENDDO

    DO j=2,ncav-1
    j1=pointA+j-1

     x(j1)=x(j1)-(1.-0.5*kd(2,j))*hc(j)*sinthe(j1)
     y(j1)=y(j1)+(1.-0.5*kd(2,j))*hc(j)*costhe(j1)
     xcav(j)=xcav(j)-(1.-0.5*kd(2,j))*hc(j)*sincav(j)
     ycav(j)=ycav(j)+(1.-0.5*kd(2,j))*hc(j)*coscav(j)
```

```fortran
      enddo

      DO j=1,ncav
      dx=xcav(j+1)-xcav(j)
      dy=ycav(j+1)-ycav(j)
      xmid=.5*(xcav(j)+xcav(j+1))
      ymid=.5*(ycav(j)+ycav(j+1))
      IF (j==1) THEN
      xmid1=xcav(1)
      ymid1=ycav(1)
      ELSE
      xmid1=.5*(xcav(j-1)+xcav(j))
      ymid1=.5*(ycav(j-1)+ycav(j))
      ENDIF

      dsc(j)    = sqrt((xmid1-xmid)*(xmid1-xmid)+(ymid1-ymid)*(ymid1-ymid))
      ds = sqrt(dx*dx+dy*dy)
      coscav(j)=dx/ds
      sincav(j)=dy/ds

      j1=pointA+j-1
      dist(j1) = ds
      sinthe(j1)=dy/dist(j1)
      costhe(j1)=dx/dist(j1)
      ENDDO


   endsubroutine  cavshape


      SUBROUTINE fff(x,arr)
      USE nrtype
      IMPLICIT NONE
      REAL(SP),INTENT(IN) :: x
      REAL(SP), INTENT(OUT) :: arr(:)
      !SOS: basis functions should be m in number!!!
      arr(1)  = x**0.47
      arr(2)  = x
      arr(3)  = x**2
      arr(4)  = x**3
      arr(5)  = x**4
      arr(6)  = (1-x)**0.5
      !

      ENDSUBROUTINE fff

      subroutine ANR
       real(sp):: dh, f1,df,fh,err
       integer:: n1
       n1=pointL
       err=1e-6
       dh=1e-2
       f1=vtang(n1-1)-vtang(n1)
       IF (abs(f1)<err) THEN
       flagA=.true.
       ELSE
       af=af+dh
       call find_linear_system_coefficients
       call ludcmp(a2(1:neqs,1:neqs),indx,d)  !LU decomposition
       call lubksb(a2,indx,b) ! solution of the linear system
       call find_foil_velocities_and_pressures
       fh=vtang(n1-1)-vtang(n1)
       df=(fh-f1)/dh
       af=af-dh-f1/df
       ENDIF

      endsubroutine

ENDPROGRAM source_dipole_cavitation


      !----------------------------------------------------------------------
      !    Module: naca_geometry.f90,        Author: Gerasimos K. Politis
      !    Revision date: Thursday, May 24, 2018,  7:21 PM
      !----------------------------------------------------------------------

      module naca_geometry
      USE nrtype
      implicit none
      integer:: m
      !
      private
      ! public functions:
      public:: naca_coordinates,naca1,comp
      !integer, parameter :: sp  = kind(1.0d0)
      !
      contains
      !
      !-------------------
      subroutine naca_coordinates(naca_m,naca_p,naca_t,chord,sc,signum, x,y)
      integer,intent(in):: naca_m,naca_p,naca_t
      real(sp),intent(in):: chord,signum,sc
      real(sp),intent(out):: x,y
      real(sp):: yc,yt,tantheta,theta
      ! find foil geometry
      IF (naca_m==0) THEN
      call naca4s(naca_t,  sc, yc,yt)
```

```fortran
      x=sc
      y=signum*yt
      ELSE
      call naca4(naca_m,naca_p,naca_t,  sc, yc,yt,tantheta)
      theta  = atan(tantheta)
      x      =   (sc - signum*yt*sin(theta))*chord
      y      =   (yc + signum*yt*cos(theta))*chord
      ENDIF
      endsubroutine naca_coordinates

      !-------------------
      ! evaluates thickness and camber for naca 4 digit airfoil
      subroutine naca4(naca_m,naca_p,naca_t,  x, yc,yt,tantheta)
      integer,intent(in):: naca_m,naca_p,naca_t
      real(sp),intent(in):: x  ! s/c
      real(sp),intent(out):: yc,yt,tantheta
      ! internals
      real(sp):: m,p,t,fact
      !
      m  = naca_m/100.
      p  = naca_p/10.
      t  = naca_t/100.
      !
      if(x<=p) then
      yc       = m*(x/p)*(2.-x/p)
      tantheta = 2*m*(1.-x/p)/p
      else
      fact     = (x-p)/(1-p)
      yc       = m*(1.-fact*fact)
      tantheta = 2*m*(p-x)/((1-p)*(1-p))
      endif
      !
      yt  =5.*t*(.2969*sqrt(x)-x*(.126+x*(0.3516-x*(.2843-x*.1036))))
      !yt  = 5.*t*(0.2969*sqrt(x)-0.126*x-0.3537*x**2+0.2843*x**3-0.1015*x**4)
      if(x<=0..or.x>=1.) then
      yt=0.
      yc=0.
      endif
      endsubroutine naca4

      subroutine naca4s(naca_t,  x, yc,yt)
      integer,intent(in):: naca_t
      real(sp),intent(in):: x  ! s/c
      real(sp),intent(out):: yc,yt
      ! internals
      real(sp):: t
      !

      t  = naca_t/100.
      !
      yt  =5.*t*(.2969*sqrt(x)-x*(.126+x*(0.3537-x*(.2843-x*.1015))))
      !yt  = 5.*t*(0.2969*sqrt(x)-0.126*x-0.3537*x**2+0.2843*x**3-0.1015*x**4)
      if(x<=0..or.x>=1.) then
      yt=0.
      yc=0.
      endif
      endsubroutine naca4s

      !=================================================
       SUBROUTINE naca1(n,xa,ya,sig)
       integer:: i
       integer, intent(inout):: n
       real(sp), allocatable, dimension(:):: xa,ya,sig

       open(7,file='ingeom.dat')
       read(7,*) n
       allocate(xa(n),ya(n),sig(n))
       DO i=1,n
        read(7,*) xa(i),ya(i)
        sig(i)=1.
       ENDDO
       sig(1)=0.01
       sig(n)=0.01
!      yp1 = (ya(2)-ya(1))/(xa(2)-xa(1))
!      ypn = 1.86*(ya(n)-ya(n-1))/(xa(n)-xa(n-1))
!      ypn=10e+30
!      yp1=ypn

       END SUBROUTINE naca1
      !=================================================




      SUBROUTINE comp(x,y,a,funcs,sign)
      REAL(sp), INTENT(IN):: x,sign
      REAL(sp), INTENT(IN), DIMENSION(:):: a
      REAL(sp), INTENT(OUT):: y
      REAL(sp) ff
      INTEGER J
      INTERFACE
        SUBROUTINE funcs(x,arr)
        USE nrtype
        IMPLICIT NONE
        REAL(SP),INTENT(IN) :: x
```

```fortran
  REAL(SP), DIMENSION(:), INTENT(OUT) :: arr
  ENDSUBROUTINE funcs
ENDINTERFACE
REAL(sp), DIMENSION(size(a)):: arr


 ff=0.

 CALL funcs(x,arr)
 DO J=1,size(a)
 ff=ff+a(J)*arr(J)
 ENDDO
 y=sign*ff


END SUBROUTINE comp


end module naca_geometry
```

The code uses the following subroutines from the book 'Numerical Recipes in Fortran 90: The Art of Scientific Computing[89] ':

- nrutil

- nr

- nrtype

- lfit

- spline

- splint

- mnewt

- covsrt

- tridag

- lubksb

- ludcmp

- locate

- gaussj

User input is through two files named `in_bem` and `ingeom.dat`. File `in_bem` must be filled with following values for each line:

```
LINE1:   nlower nupper
LINE2:   angle_of_attack
LINE3:   vinf
LINE4:   chord
LINE5:   naca_m naca_p naca_t
LINE6:   ld lc lamda af ni
LINE7:   iter
```

The following list explains each of the above values.

> `nlower`: number of panels on the lower face(pressure side) of the foil.
>
> `nupper`: number of panels on the upper face(suction side) of the foil.
>
> `angle_of_attack`: angle of attack of foil.
>
> `vinf`: Velocity of hydrofoil in m/s.
>
> `chord`: chord length of the hydrofoil in m.
>
> `naca_m`: max camber % of NACA 4 digit hydrofoil. If not using this series value must be zero.
>
> `naca_p`: max camber position % of NACA 4 digit hydrofoil. If not using this series value must be zero.
>
> `naca_t`: thickness % of NACA 4 digit hydrofoil. If not using this series value must be zero.
>
> `ld`: x/c of cavity detachment point.

`lc`: length of cavity over chord.

`lamda`: length of cavity termination transition zone over chord.

`af`: parameter A of cavity termination model. Must be $0 < A < 1$.

`ni`: paramater ν of cavity termination model. Must be $\nu > 0$.

`iter`: number of iterations of the method.

File `ingeom.dat` is used in order to interpolate a given set of points for an arbritrary symmetric hydrofoil geometry. It must contain the number of points to interpolate in the first line and next lines will be the x and y coordinates of the points. Points should start from the trailing edge and finish on the leading edge.

For the reader's ease of reference an example of the input files is included below and the output file in this case, which includes both the updated cavity shape and pressure distribution on the hydrofoil surface.
File `in_bem`:

```
40 40 !  nlower,nupper
4.  !  input angle of attack in degrees
1.  !  velocity at infinity (m/s)
1.  !  input foil chord in (m)
0 0 0 !  not equal 0 if NACA 4 digit series geometry used
0.  0.5 0.1 0.3 2.  !  ld,lc,lamda,af,ni
3 !iterations
```

File `ingeom.dat`:

```
17
1.0000 0.00000
0.9500 0.00707
0.9000 0.01259
0.8000 0.02099
0.7000 0.02635
0.6000 0.02917
0.5000 0.03000
0.4000 0.02927
0.3000 0.02709
0.2000 0.02332
0.1500 0.02067
0.1000 0.01729
0.0750 0.01516
0.0500 0.01255
0.0250 0.00903
0.0125 0.00646
0.0000 0.00000
```

File `tecplot_geometry_and_pressures.dat` is one of the output files:

```
TITLE = " plot foil geometry, pressure coefficients or q"
VARIABLES = "Xmid","Ymid","−Cp","q"
 ZONE T="   plot geometry, pressure coefficients or q ", I=   80, F=POINT
        1.00000       −0.00000      −0.28061       0.89238
        0.99692       −0.00046      −0.27627       0.88712
        0.98776       −0.00179      −0.19447       0.87616
        0.97275       −0.00389      −0.13083       0.85946
        0.95225       −0.00658      −0.08098       0.83725
        0.92678       −0.00966      −0.04212       0.81001
        0.89695       −0.01290      −0.01266       0.77841
        0.86350       −0.01611       0.00849       0.74327
        0.82725       −0.01912       0.02228       0.70556
        0.78911       −0.02180       0.02973       0.66628
        0.75000       −0.02408       0.03194       0.62650
        0.71089       −0.02593       0.03003       0.58726
        0.67275       −0.02735       0.02514       0.54958
        0.63650       −0.02839       0.01828       0.51441
        0.60305       −0.02910       0.01038       0.48260
        0.57322       −0.02955       0.00228       0.45491
        0.54775       −0.02981      −0.00523       0.43198
        0.52725       −0.02993      −0.01151       0.41433
        0.51224       −0.02998      −0.01619       0.40234
        0.50308       −0.02999      −0.01902       0.39628
        0.50000       −0.02999      −0.02023       0.39323
        0.49692       −0.02999      −0.02212       0.38718
        0.48776       −0.02998      −0.02536       0.37525
        0.47275       −0.02993      −0.03042       0.35777
        0.45225       −0.02981      −0.03707       0.33521
        0.42678       −0.02958      −0.04546       0.30819
        0.39695       −0.02919      −0.05621       0.27745
        0.36350       −0.02861      −0.06972       0.24384
        0.32725       −0.02780      −0.08601       0.20827
        0.28911       −0.02674      −0.10510       0.17171
        0.25000       −0.02543      −0.12747       0.13515
        0.21089       −0.02384      −0.15390       0.09959
        0.17275       −0.02199      −0.18561       0.06597
        0.13650       −0.01988      −0.22439       0.03522
        0.10305       −0.01754      −0.27298       0.00816
        0.07322       −0.01499      −0.33576      −0.01448
        0.04775       −0.01226      −0.42016      −0.03211
        0.02725       −0.00939      −0.53952      −0.04432
        0.01224       −0.00640      −0.71992      −0.05092
        0.00308       −0.00332      −0.98098      −0.05187
        0.00000        0.00000       0.65795      −0.04668
        0.00304        0.00455       0.91395      −0.03464
        0.01119        0.01314       0.91395      −0.01504
        0.02564        0.02146       0.91395       0.01162
        0.04574        0.03012       0.91395       0.04518
        0.07098        0.03867       0.91395       0.08490
        0.10072        0.04672       0.91395       0.12989
        0.13422        0.05396       0.91395       0.17911
        0.17065        0.06011       0.91395       0.23143
        0.20910        0.06494       0.91395       0.28562
        0.24862        0.06827       0.91395       0.34044
        0.28825        0.07001       0.91395       0.39464
        0.32700        0.06924       0.91395       0.44722
```

0.36437	0.06872	0.91395	0.49706
0.39951	0.06585	0.91395	0.54268
0.43015	0.05800	0.72357	0.58072
0.45508	0.04798	0.31569	0.60829
0.47403	0.03844	-0.08830	0.62583
0.48787	0.03173	-0.36759	0.63557
0.49692	0.02999	-0.50174	0.63989
0.50000	0.02999	-0.50174	0.64206
0.50308	0.02999	-0.34756	0.64701
0.51224	0.02998	-0.21044	0.65774
0.52725	0.02993	-0.09313	0.67465
0.54775	0.02981	-0.00435	0.69759
0.57322	0.02955	0.05889	0.72605
0.60305	0.02910	0.10184	0.75927
0.63650	0.02839	0.12922	0.79631
0.67275	0.02735	0.14451	0.83612
0.71089	0.02593	0.14995	0.87758
0.75000	0.02408	0.14678	0.91952
0.78911	0.02180	0.13551	0.96076
0.82725	0.01912	0.11618	1.00018
0.86350	0.01611	0.08863	1.03668
0.89695	0.01290	0.05259	1.06931
0.92678	0.00966	0.00771	1.09725
0.95225	0.00658	-0.04652	1.11987
0.97275	0.00389	-0.11122	1.13676
0.98776	0.00179	-0.18884	1.14775
0.99692	0.00046	-0.28496	1.15298

where `Xmid` and `Ymid` (first two columns)are the hydrofoil geometry coordinates, which have been updated in the cavity region to follow the cavity shape. `-Cp` (third column) is the minus pressure coefficient and `q` (fourth column) is the local fluid velocity potential. From this output file, one can deduce that the cavitation number in this case is 0.91395. The code also produces the output file `tecplot_field_point_velocities_and_pressures.dat` which contains values of the pressure coeeficient and velocity in a field away from the hydrofoil. Both output files can be directly loaded into TECPLOT software for results plotting.