



Norwegian University of
Science and Technology

Coupling of a 2D Boundary Element Method with a local analytical solution to deal with geometrical singularities

Einar Biørn-Hansen

Marine Technology

Submission date: June 2016

Supervisor: Marilena Greco, IMT

Norwegian University of Science and Technology
Department of Marine Technology

NORWEGIAN UNIVERSITY OF SCIENCE, NTNU

MASTER THESIS IN MARINE TECHNOLOGY

SPRING 2016

**Coupling of a 2D Boundary Element Method
with a local analytical solution to deal with
geometrical singularities**

Author:

Einar BIØRN-HANSEN

Supervisor:

Prof. Marilena GRECO

June 10, 2016

Abstract

Boundary Element Method, BEM, as a numerical tool to solve boundary values problems, has been developed and tested. The numerical scheme is coupled with known analytical flow models for geometrical singularities, and results compared. Both problems in infinite fluid, and free surface are investigated, and convergence issues addressed. Accuracy and error approximation of the solver is presented.

As an introduction, the BEM solver is applied to BVPs where analytical solutions are known. For the case of a circle, double wedge and rounded square in infinite fluid, the added mass is calculated, and good agreement with analytical solutions are found.

Later, a local flow model is introduced in the presence of geometrical singularities. Both direct and least square matching of the solutions are used. The solutions sensitivity to the local solutions parameters is investigated. In both cases it is found that a low number of constants and matching nodes on the boundary of the body, produce the best results. On a late stage in the work on this thesis, it is found that the reason for the latter, might be that the solver is biased to produce better results for boundary matching nodes, which is discussed in the thesis. Despite that a low number of constants (2), is found to give best matching, a higher number of constants is chosen for the final model. Reason being that using two constants in the local solution, gives poor matching of the velocity in the intersection between the domains. The results from both models show good agreement, and the added mass deviates with about 2% from the "sharpest" rounded square. The order of accuracy is found to be lower than the theoretical order, and the reason for this should be investigated further. The error bands show convergence.

BVPs including the free surface are analyzed in the last part of the thesis. This is done both

using "double body" considerations, and by distributing free space Green's functions over the body and free surface. For bodies with wall sided boundary conditions, both methods produce good results, tested for an oscillating half-cylinder. Analyzing the same BVP for a wedge, which is not orthogonal to the surface, convergence issues in the free surface are experienced for the latter method. On the body surface the results for both methods agree with with the analytical. Addressing the convergence issue, a local flow model is introduced in the intersection between the body and surface. This deals with the convergence issue, but results in deviating values for the velocity potential, giving to low values of the added mass.

The concluding remarks in the thesis, are that the local solutions used on the square cylinders give realistic results in terms of the fluid velocity close to the corner apex. For free surface problems, it found that if only a solution on the boundary of the body is needed, the double body method is preferred. This both because of its better numerical performance and the unnecessary large equation system that has to be solved, using Rankine sources. In the opposite case, a local flow model should be included in the intersection between the body and free surface, if the the boundary conditions are not orthogonal.

Sammendrag

I denne oppgaven er Boundary Element Method, studert inngående og har blitt implementert i Matlab for forskjellige grenseverdi-problem. I tilfeller med geometriske singulariteter, blir en lokal løsning brukt rundt singulariteten, og den globale løsningen blir koblet med den lokale ved kreve kontinuitet av løsningene ved en kunstig grensen mellom de to domenene. Løserens nøyaktighetsgrad, og konvergenssegenskaper blir også undersøkt.

For verifikasjon og sammenligning, blir BEM-løseren først anvendt på problemer der den analytiske løsningen er kjent. For en sirkel, "dobbel kil" og et kvadrat med avrundede hjørner, blir globale verdier som hydrodynamisk tilleggsmasse beregnet. Det blir her funnet at de numeriske løsningene stemmer godt med de analytiske.

I tilfeller med geometrisk singularitet, introduseres et lokalt hastighetspotensial. Undersøkelser av løsningens sensitivitet til de forskjellige variablene i den lokale løsningen, viser at den beste lokale løsningen, er en der få konstanter blir brukt, og med kontinuitets-noder kun langs overflaten av sylindren. Dette blir funnet både når kontinuitets ligningene brukes direkte, og ved bruk av minste kvadraters metode. Det blir til tross for dette valgt å bruke flere konstanter i den endelige løsningen, for å oppnå bedre samsvar mellom hastighetene i det lokale og globale domenet. Resultatene fra analysene, viser at hastigheten rundt hjørnene oppfører seg asymptotisk. Tilleggsmassen funnet med disse modellene er i sammen størrelsesorden som den analytiske Lewis form-massen, og avviker med rundt 2%.

I siste del av oppgaven blir problemer med fri overflate analysert. Det blir sett nærmere på legemer som oscillerer med uendelig frekvens, og tilleggsmassen beregnet. Dette blir funnet både ved å bruke kilder som tilfredsstillende fri overflate betingelsen, og Rankine kilder. For tilfeller der

grensebetingelsene mellom overflaten og legemet er ortogonale, gir begge metodene tilfredstillende løsninger, som stemmer godt overens analytiske verdier.

I tilfellet med en kil som oscillerer i overflaten, oppstår det et konvergens problem når Rankine kilder brukes, for normalhastigheten i overflaten nær kilen. Til tross for dette oppleves det ikke noe problem med hastighetspotensialet over overflaten til legemet, og tilleggsmassen kan også i dette tilfellet beregnes med høy nøyaktighet. For mer realistisk løsning av normalhastigheten i skjæringen mellom kilen og overflaten, introduseres en analytisk lokal model. Ved å gjøre dette oppnås konvergens av normalhastigheten. Resultatet blir da at hastighetspotensialet for alle bunnreisvinkler blir mindre enn tidligere beregnet verdier, noe som resulterer i for lav tilleggsmasse. Dette skyldes sannsynligvis problemer med den lokale modellen som er innført, og burde undersøkes videre i fremtidig arbeid.

MASTER THESIS IN MARINE TECHNOLOGY

SPRING 2016

FOR

Einar Biørn-Hansen

Coupling of a 2D Boundary Element Method with a local analytical solution to deal with geometrical singularities

(Fysisk undersøkelse av slamming last på et 2D legeme)

Slamming is a local nonlinear phenomenon connected with liquid-structure impact that can have local and global consequences on ships, depending on the vessel geometry and operational conditions. Within a strip-theory strategy for slender bodies, the slamming is studied as a series of 2D problems in the cross-sectional planes of the ship. A great part of the 2D analyses examines the slamming for a wedge or a circular cylinder in calm water because ship cross-sections are mostly U/V shaped. In head sea conditions, the 2D problems would be symmetric with respect to the ship axis, while in oblique sea waves they would be asymmetric due to the excitation of roll motion. In this framework, numerical convergence is an issue for potential-flow slamming solvers in case of geometrical singularities.

In the project work, the student examined different 2D Boundary Integral Formulations (BIF) and related Boundary Element Methods (BEMs) available in literature and those used within local slamming analyses. Both an indirect and direct BIFs were implemented as zero-order BEMs. The BEMs were used to investigate a circular and a squared cylinder. In the latter case, a local corner-flow solution was incorporated in the BEM but the results were not fully satisfactory.

Objective

Present master thesis aims to solve the problems raised by the project studies and to investigate the advantages of incorporating a local-flow solution in a BEM in terms of accuracy and numerical convergence.

The work should be carried out in steps as follows:

1. Summarize major findings/outcomes from the project thesis.
2. Complement the literature study of the project on BEM formulations and local corner-flow solutions.
3. Investigate the possible implementation errors in the coupled BEM-local solution and discuss the matching of the local and global solutions. Compare the results against the full BEM solution in the case of a squared cylinder in infinite fluid. Select at least one suitable approach to examine the numerical convergence and apply it when the inflow is aligned with the body axes and possibly with the diagonal.
4. Study the case of an almost squared cylinder, i.e. with small radius of curvature at the four corners, and discuss the numerical convergence when reducing the radius of curvature. Compare the BEM solution with small radius of curvature at the corners with the full BEM solution and the BEM-with local solution for the square cylinder.

5. Study numerically the case of a half surface-piercing cylinder with infinite-frequency free-surface boundary condition. Compare the results against those for a circular cylinder in infinite fluid.
6. Assume a surface-piercing wedge with infinite-frequency free-surface boundary condition. In this case, the body is not wall-sided at the free surface. Investigate the numerical convergence and the possibility of incorporating a local solution at the intersection between the body and the free surface.

The work may show to be more extensive than anticipated. Some topics may therefore be left out after discussion with the supervisor without any negative influence on the grading.

The candidate should in his report give a personal contribution to the solution of the problem formulated in this text. All assumptions and conclusions must be supported by mathematical models and/or references to physical effects in a logical manner.

The candidate should apply all available sources to find relevant literature and information on the actual problem.

The thesis should be organised in a rational manner to give a clear presentation of the work in terms of exposition of results, assessments, and conclusions. It is important that the text is well written and that tables and figures are used to support the verbal presentation. The thesis should be complete, but still as short as possible. In particular, the text should be brief and to the point, with a clear language. Telegraphic language should be avoided.

The thesis must contain the following elements: the text defining the scope (i.e. this text), preface (outlining project-work steps and acknowledgements), abstract (providing the summary), table of contents, main body of thesis, conclusions with recommendations for further work, list of symbols and acronyms, references and (optional) appendices. All figures, tables and equations shall be numerated.

The supervisor may require that the candidate, in an early stage of the work, present a written plan for the completion of the work. The plan should include budget for the use of computer and laboratory resources that will be charged to the department. Overruns shall be reported to the supervisor.

From the thesis it should be possible to identify the work carried out by the candidate and what has been found in the available literature. It is important to give references to the original source for theories and experimental results.

Supervisor :Marilena Greco
Submitted :16 January 2016
Deadline :10 June 2016

Marilena Greco
Supervisor

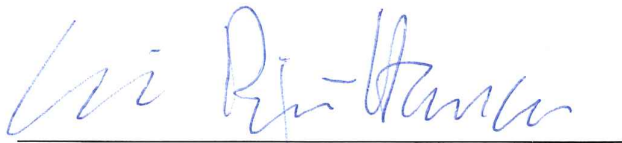
Preface

This thesis is the result of the individual work done during the spring of 2016, and is the concluding work of the Master of Science degree in Marine Technology at the Norwegian University of Science, NTNU. The work has been done with Professor Marilena Greco as supervisor.

The topic was initially proposed by DNV GL, to be a study of slamming loads on 2 dimensional bodies. The focus of the thesis has later been shifted, and it is here investigated how BEM can be used to solve various BVPs in infinite fluid and in the free surface, and how it can be coupled with local solutions to deal with geometrical singularities and convergence issues.

Studying the Boundary Element Method theory and applications has been very time consuming, but also very educational. Implementing it in Matlab proved to be a challenging task, and there were several misunderstandings that had to be clarified to get meaningful results.

I would like to take this opportunity to thank Jens Helmers in DNV GL, for initially proposing such an interesting topic, and for being available for answering my questions. I would also like to thank my supervisor, Marilena Greco. Thank you for all your help and support, I am very grateful.



Einar Biørn-Hansen

Trondheim, 2016-06-10

Nomenclature

| | |
|--------------|---|
| 2D | Two dimensional |
| BEM | Boundary Element Method |
| BIE | Boundary Integral Equation |
| BVP | Boundary Value Problem |
| LHS | Left hand side |
| TLC | Top left corner |
| GCI | Grid Convergence Index |
| RHS | Right hand side |
| STD | Standard Deviation |
| α | Internal angle |
| δ | Dirac's delta function |
| ω | Circle frequency [rad/s] |
| Φ | Velocity potential |
| Φ_{h0} | Approximate exact solution |
| Φ_{loc} | Local velocity potential |
| Φ_n | Normal velocity |
| Σ_m | Matching boundary between local and global domain |

| | |
|----------------|---|
| Σ_{S_B} | Matching boundary between local and global domain on body surface |
| Σ_{S_F} | Matching boundary between local and global domain on free surface |
| θ | Dead rise angle |
| a | Size of local domain |
| A_C | Control Area |
| B | Width wedge |
| C | Boundary surface |
| F | Fraction between the refinement factor, p and GCI for three grids |
| F_S | Safety factor |
| g | Gravitational acceleration |
| G | Free space Green's function |
| G_n | Normal derivative of free space Green's function |
| h | grid spacing |
| \mathbf{I} | Identity matrix |
| N_C | Number of constants in local solution |
| \mathbf{n} | Unitary normal vector |
| p | Order of accuracy |
| r | Grid refinement factor |
| S_0 | Sea bottom surface |
| S_B | Body surface |
| S_F | Free surface |
| S_∞ | Vertical control surface |
| \mathbf{U} | Current velocity |

| | |
|---------------|-----------------|
| V | Velocity |
| V_n | Normal velocity |
| \mathbf{x} | Field point |
| $\mathbf{x0}$ | Singular point |

Contents

| | |
|---|------------|
| Abstract | i |
| Sammendrag | iii |
| 1 Introduction | 2 |
| 1.1 Motivation | 2 |
| 1.2 Background | 2 |
| 1.3 Scope of Work | 3 |
| 2 Theory | 4 |
| 2.1 Boundary Element Method - Direct formulation | 4 |
| 2.1.1 Green's 2nd identity and Green's functions | 5 |
| 2.1.2 Boundary Element Formulation | 8 |
| 2.1.3 Discretization of the boundary and system of equations | 9 |
| 2.1.4 Local solution | 11 |
| 2.1.5 Matching equations | 13 |
| 2.1.6 Coupled Method | 15 |
| 2.1.7 Free surface effects | 16 |
| 3 Numerical aspects -Error and Convergence | 20 |
| 3.1 Local error, Order of Accuracy and Grid Convergence Index | 20 |
| 3.2 Sensitivity to parameters in local potential solution | 23 |
| 4 Comments on computer code | 25 |
| 4.1 Geometry.m | 25 |

| | | |
|----------|---|-----------|
| 4.2 | matcalc.m | 26 |
| 4.3 | Corner.m | 26 |
| 4.4 | Leastsquare.m | 26 |
| 4.5 | Sysmat.m | 26 |
| 4.6 | Sensitivity | 27 |
| 4.7 | Convergence | 27 |
| 5 | Infinite fluid problems | 28 |
| 5.1 | Boundary Value Problem | 28 |
| 5.2 | Rounded Square | 30 |
| 5.3 | Square, direct matching | 35 |
| 5.3.1 | Sensitivity and convergence | 35 |
| 5.3.2 | Results | 39 |
| 5.4 | Square, Least square matching | 46 |
| 5.4.1 | Sensitivity and convergence | 47 |
| 5.4.2 | Results | 51 |
| 5.5 | Oscillating double | 57 |
| 6 | Free surface problems | 61 |
| 6.1 | Boundary Value Problem | 61 |
| 6.2 | Cylinder with draft equal to radius | 62 |
| 6.3 | Wedge oscillating in the free surface | 65 |
| 6.3.1 | Convergence and Sensitivity | 65 |
| 6.3.2 | Results | 67 |
| 6.4 | Local analysis | 72 |
| 6.4.1 | Convergence and sensitivity | 72 |
| 6.4.2 | Results | 74 |
| 7 | Discussion and Concluding Remarks | 79 |
| 8 | Suggestions for further work | 84 |
| A | Divergence Theorem in the plane | 86 |

| | |
|----------------------------------|------------|
| <i>CONTENTS</i> | xiii |
| B Method of Least squares | 87 |
| C Solution Details | 88 |
| D Central Difference | 91 |
| E Matlab code | 93 |
| E.1 Geometry | 93 |
| E.2 Matcalc.m | 96 |
| E.3 Corner | 100 |
| E.4 Least Sqaure | 106 |
| E.5 Sysmat | 113 |
| E.6 Sensitivity | 114 |
| E.7 Convergence | 115 |
| Bibliography | 118 |

List of Figures

| | | |
|------|---|----|
| 2.1 | Example of control area Pozrikidis (2002) | 6 |
| 2.2 | Singular point within control area Faltinsen (1990) | 7 |
| 2.3 | Arbitrary body discretized into N constant elements Pozrikidis (2002) | 10 |
| 2.4 | Definition of coordinate system for corner Liang et al. (2015) | 12 |
| 2.5 | Matching nodes distributed around corner, and along boundary | 14 |
| 2.6 | Integration surfaces in two dimensions | 18 |
| 3.1 | Example of Discretized boundary Dijkstra et al. (2011) | 21 |
| 5.1 | Object in constant current | 29 |
| 5.2 | Geometry of square with rounded corners with $r=0.5$ and location of control nodes. | 30 |
| 5.3 | Velocity potential for rounded square in constant current, $U=3$ | 32 |
| 5.4 | Velocity potential for square and circle in constant current, $U=3$ | 33 |
| 5.5 | Normalized added mass of a rounded square for different radii of curvature, circle and square with sharp corners | 34 |
| 5.6 | Φ_{loc} at 10 global nodes closest to the local domain of the top left corner, plotted with global velocity potential | 37 |
| 5.7 | Control nodes for convergence study | 38 |
| 5.8 | Convergence plot for control node 1, with approximate true solution | 39 |
| 5.9 | Node numbering, direction of integration and intersection between local and global domain | 40 |
| 5.10 | Global velocity potential, $N_B = 4$, $a = 0.09$ | 41 |
| 5.11 | Local velocity potential, $N_B = 4$, $a = 0.09$ | 42 |

| | |
|---|----|
| 5.12 Local and global velocity for $N_B = 4$ | 43 |
| 5.13 Local and global velocity for $N_B = 2$ | 44 |
| 5.14 Velocity potential in surrounding fluid | 45 |
| 5.15 Velocity of surrounding fluid | 46 |
| 5.16 Standard deviation for all combination of N_B and N_C | 48 |
| 5.17 Local solution at 10 global nodes closest to the local domain of the top left corner, using $N_B = 2, N_{out} = 0, N_C = 2, a = 0.07$ | 49 |
| 5.18 Convergence plot for control node 1, including approximated exact solution | 50 |
| 5.19 Global velocity potential, $N_B = N_C = 4, a = 0.09$ | 52 |
| 5.20 Local velocity potential, $N_B = N_C = 4, a = 0.09$ | 53 |
| 5.21 Global and local velocity for $N_B = N_C = 4$ | 54 |
| 5.22 Local and global velocity for $N_B = N_C = 2$ | 55 |
| 5.23 Velocity potential in surrounding in fluid | 56 |
| 5.24 Velocity of surrounding fluid | 57 |
| 5.25 Geometry of wedge Liang et al. (2015) , and node numbering and direction of inte- gration | 58 |
| 5.26 Added mass in heave for double wedge with 20° dead rise angle, oscillating in infinite fluid | 59 |
| 5.27 Velocity potential for double wedge with 20° dead rise angle, oscillating in infinite fluid | 60 |
| 6.1 Geometrical description of boundary value problem | 62 |
| 6.2 Velocity potential and normal velocity of cylinder oscillating with infinite frequency in the free surface | 63 |
| 6.3 Velocity potential for half cylinder oscillating in free surface, found by double body considerations | 64 |
| 6.4 Control nodes and node numbering | 66 |
| 6.5 Velocity potential distribution over wedge surface, for oscillation in free surface, analyzed by double body method | 68 |
| 6.6 Normal velocity on S_F and S_B using Rankine source distribution | 69 |

6.7 Velocity potential on the free surface and body of oscillating wedge 70

6.8 Added mass for oscillating wedge in free surface 71

6.9 Global and local velocity potential and normal velocity 73

6.10 Global velocity potential of oscillating wedge, with local solution in the intersection 75

6.11 Local velocity potential of oscillating wedge, with local solution in the intersection 75

6.12 Global normal velocity of oscillating wedge, with local solution in the intersection 76

6.13 Local normal velocity of oscillating wedge, with local solution in the intersection . 76

6.14 Added mass in heave of oscillating wedge, analyzed using with local flow model in
intersection between the free surface and body surface 78

7.1 Location of outer matching nodes and global nodes where STD is calculated. . . . 81

List of Tables

| | | |
|------|--|----|
| 5.1 | GCI, p and F for rounded square in constant current | 31 |
| 5.2 | Minimum and maximum values of Φ for different radii of curvature | 33 |
| 5.3 | Added mass for different radii of curvature | 35 |
| 5.4 | Standard deviation for different numbers of outer domain matching nodes | 36 |
| 5.5 | Standard deviation for different numbers of boundary matching nodes, and local domain size. | 36 |
| 5.6 | Observed order of accuracy, GCI and approximated true solution for 5 nodes on the surface on the cylinder | 38 |
| 5.7 | Global and local velocity in intersection between local and global domain for top left corner | 45 |
| 5.8 | Added mass of oscillating square cylinder in infinite fluid | 46 |
| 5.9 | Standard deviation for different combinations of N_B and N_{out} , using $N_C = 2$ | 47 |
| 5.10 | Minimum standard deviation for different combinations of N_C , N_B and a. | 48 |
| 5.11 | Observed order of accuracy, GCI and fraction for 5 nodes on the surface on the cylinder | 50 |
| 5.12 | Global and local velocity in intersection between local and global domain for top left corner | 56 |
| 5.13 | Added mass of oscillating square cylinder in infinite fluid, analyzed with $N_B = N_C = 4$ using least square matching | 57 |
| 5.14 | Largest deviation between analytical and numerical added mass in heave for double wedge | 59 |

| | | |
|-----|---|----|
| 6.1 | Element data for problem with cylinder in oscillating in free surface, using free space Green's functions | 63 |
| 6.2 | Standard deviation between analytical solution and numerical values obtained using the double body method and Rankine source distribution | 64 |
| 6.3 | Added mass of half cylinder oscillating in free surface | 65 |
| 6.4 | p , GCI and F for wedge oscillating in the free surface, analyzed using double body . | 66 |
| 6.5 | p , GCI and F for wedge oscillating in the free surface, analyzed using Rankine source distribution. | 67 |
| 6.6 | Standard deviation between local and global solution, for velocity potential and normal velocity at 10 global nodes closest to intersection | 73 |
| 6.7 | E , GCI and p at surface boundary and free surface control nodes | 74 |
| 6.8 | Global and local velocity potential and normal velocity at the intersections between the domains | 77 |

Chapter 1

Introduction

1.1 Motivation

Numerous problems within hydrodynamics, involve problems which are difficult or impossible to solve analytically. Facing such problems, numerical methods are often necessary. If potential theory is assumed, one such numerical alternative is the Boundary Element Method, BEM, which can be used to solve linear partial differential equations, such as the Laplace equation.

One great advantages with using BEM is the efficiency. This is because in BEM, one only has to discretize the boundary of the domain, instead of the whole domain, as is the case for other numerical methods. This results in smaller, but dense system matrices to be solved.

1.2 Background

Prior to implementing BEM, an extensive literature study on the Boundary Element Method is conducted. In addition to this, literature in marine hydrodynamics is reviewed to find suitable models to describe the various BVPs, and theoretical background on how BEM can be applied to hydrodynamic applications.

When using numerical tools to approximate the solution of continues problems, one has to

make sure the solution is convergent, prior to making any conclusions. Literature on how to approximate the observed order of accuracy and errors is therefor also reviewed.

1.3 Scope of Work

The focus of this thesis is to make use of BEM to solve relevant boundary value problems in hydrodynamics. This includes flow around, and oscillation of smooth bodies and bodies with geometrical singularities. For the latter an analytical corner flow model is introduced, and coupled with the global solution. The continuity equations are enforced both directly and using the method of least squares. Lastly BVPs where the free surface is present, is considered, both for the case of wall sided and non-wall sided boundary conditions between the body and the free surface.

In the thesis, the work done is presented in the following 7 chapters

- **Chapter 2:** Findings from the literature study of BEM and relevant hydrodynamic applications.
- **Chapter 3:** Numerical aspects of BEM.
- **Chapter 4:** Comments on implementation.
- **Chapter 5:** Results form BVPs in infinite fluid.
- **Chapter 6:** Results from BVPs in free surface.
- **Chapter 7:** Final comments and concluding. remarks
- **Chapter 8:** Further work.

Further, details such as mathematical theorems and particular solution details are included in the appendix.

Chapter 2

Theory

2.1 Boundary Element Method - Direct formulation

The Boundary Element Method, BEM, is a numerical method, which can be used to solve partial differential equations such as the Laplace equation, which will be the focus of this thesis. Hence it is from here on assumed that all flows are irrotational, incompressible and inviscid. This results in only one governing equation, namely the Laplace equation.

$$\nabla^2\Phi = 0 \tag{2.1}$$

Where Φ is an unknown velocity potential. This equation must also satisfy the natural and essential boundary conditions. It is further in this thesis only focused on two dimensional bodies, yielding the Laplace equation on component form as:

$$\frac{\partial^2\Phi}{\partial x^2} + \frac{\partial^2\Phi}{\partial y^2} = 0 \tag{2.2}$$

The idea behind BEM, is to express the solution in terms of a boundary distribution of the fundamental solution of the PDE, in this case equation 2.1. The fundamental solutions are Green's functions, which expresses the field, in terms of a local source. The densities of the sources are

then computed, by requiring boundary conditions to be satisfied.

In the following, Green's functions, and Green's second identity are introduced. Following this is how they together can be used to find a Boundary Integral Equation, BIE, for the velocity potential at the boundary of a body, shown.

2.1.1 Green's 2nd identity and Green's functions

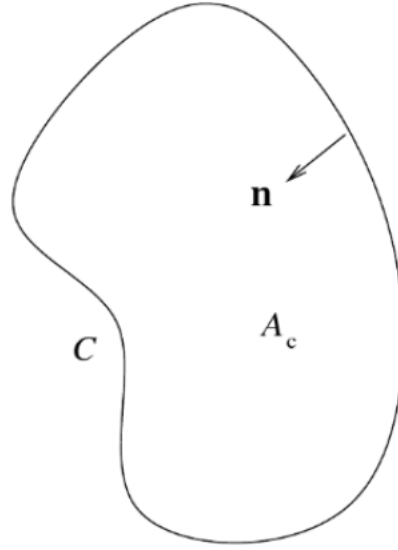
To derive a BIE for the unknown Φ or Φ_n , Green's second identity is needed. It states that for any twice differentiable function Ψ , the following holds:

$$\Psi \cdot \nabla^2 \Phi - \Phi \cdot \nabla^2 \Psi = \nabla \cdot (\Psi \cdot \nabla \Phi - \Phi \cdot \nabla \Psi) \quad (2.3)$$

If Ψ also satisfies the Laplace equation, i.e. is a harmonic equation 2.3 yields the reciprocal relationship for harmonic functions.

$$0 = \nabla \cdot (\Psi \cdot \nabla \Phi - \Phi \cdot \nabla \Psi) \quad (2.4)$$

The integral form of equation 2.4, can be found by integrating over the solution domain. By denoting the control area A_C , and using the divergence theorem as defined in Appendix A, the integral is reduced from a surface integral to a line integral, around the closed surface, C . The normal vector is as defined in figure 2.1, pointing into the fluid domain. This results in:

Figure 2.1: Example of control area [Pozrikidis \(2002\)](#)

$$0 = \int_{A_c} \nabla(\Psi \nabla \Phi - \Phi \nabla \Psi) dA = \int_C (\Phi \cdot \nabla \Psi - \Psi \cdot \nabla \Phi) \cdot \mathbf{n} \cdot dl \quad (2.5)$$

A Green's function, G is the fundamental solution of the PDE. For the Laplace equation, G is a function which satisfies the Laplace equation everywhere in the domain, except in a point \mathbf{x}_0 , where it is singular. It can be defined as [Pozrikidis \(2002\)](#)

$$\nabla^2 G(\mathbf{x}, \mathbf{x}_0) + \delta(\mathbf{x}, \mathbf{x}_0) = 0 \quad (2.6)$$

Here \mathbf{x} is a field point, δ is the Dirac's delta function, which is zero everywhere, except in \mathbf{x}_0 , where it is infinity. Bold indicates vector. By substituting $\nabla^2 G(\mathbf{x}, \mathbf{x}_0) = -\delta(\mathbf{x}, \mathbf{x}_0)$ into 2.3 for Ψ , and integrating using the divergence theorem, the following result is obtained:

$$\int_{A_c} \Phi(\mathbf{x}) \delta(\mathbf{x}, \mathbf{x}_0) dA_c(\mathbf{x}) = \int_C (\Phi(\mathbf{x}) \cdot \nabla G(\mathbf{x}, \mathbf{x}_0) \cdot \mathbf{n}(\mathbf{x}) - G(\mathbf{x}, \mathbf{x}_0) \cdot \nabla \Phi(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})) \cdot dl(\mathbf{x}) \quad (2.7)$$

To be able to solve this system, an expression for G is needed. In addition to satisfying equation 2.6, it should satisfy the boundary conditions of the BVP. In the case of an infinite solution space, the far-field conditions that the potential and the normal derivative goes to zero, should be satisfied. In the case of an infinite solution space, i.e no interior boundary exists, G can be found by Fourier transform of equation 2.6, to be Pozrikidis (2002):

$$G(\mathbf{x}, \mathbf{x}_0) = -\frac{1}{2\pi} \ln(r) \quad (2.8)$$

Where r , is the radial distance between the field point \mathbf{x} , and the singularity \mathbf{x}_0 . I.e $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$. The value of integral on the LHS of equation 2.7, depends on the location of the singular point \mathbf{x}_0 . There are three different case to consider:

1. The singular point is inside A_c
2. The singular point is on C
3. The singular point is outside A_c .

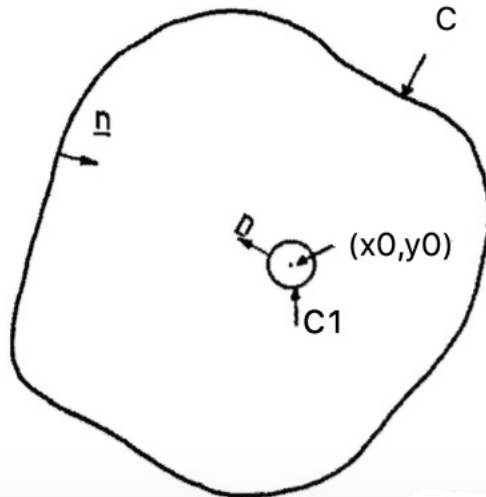


Figure 2.2: Singular point within control area Faltinsen (1990)

The first case is illustrated in figure 2.2. The surface integral is split into a small circle C_1 surrounding the singularity, and C surrounding the rest of the domain. By integrating 2.6, using the

divergence theorem, for the three different cases, the following is obtained Pozrikidis (2002).

$$\int_{A_c} \Phi(\mathbf{x}) \delta(\mathbf{x}, \mathbf{x}_0) dA_c(\mathbf{x}) = \begin{cases} 1, & \text{When } \mathbf{x}_0 \text{ inside } A_c \\ \frac{1}{2}, & \text{When } \mathbf{x}_0 \text{ on } C \\ 0, & \text{When } \mathbf{x}_0 \text{ outside } A_c \end{cases} \quad (2.9)$$

When the singular point is on the boundary, the line integral on the RHS of equation 2.7, is a principle value integral. These three values, when multiplied with 2π from the Greens function, are often referred to as angles in the literature. They will from here on be denoted α .

2.1.2 Boundary Element Formulation

Using the findings from the previous section, we are able to formulate equations for Φ and Φ_n in the fluid domain. The integral equation 2.7, then reduces to a distribution of Green's functions over the boundary of the body.

$$\alpha \cdot \Phi(\mathbf{x}_0) = \int_C \left(\Phi(\mathbf{x}) \cdot \nabla G(\mathbf{x}, \mathbf{x}_0) - G(\mathbf{x}, \mathbf{x}_0) \cdot \nabla \Phi(\mathbf{x}) \right) \cdot \mathbf{n}(\mathbf{x}) \cdot dl(\mathbf{x}) \quad (2.10)$$

By using the appropriate interior angle, and substituting $G(\mathbf{x}, \mathbf{x}_0) = -\frac{1}{2\pi} \ln(r)$, the following expression for Φ on the boundary of the body is obtained.

$$\pi \cdot \Phi(\mathbf{x}_0) = \int_C \left(-\Phi(\mathbf{x}) \cdot \nabla \ln(r) + \ln(r) \cdot \nabla \Phi(\mathbf{x}) \right) \cdot \mathbf{n}(\mathbf{x}) \cdot dl(\mathbf{x}) \quad (2.11)$$

BEM, is based on that either information of Φ , or Φ_n on the boundary is known. If the normal derivative of the velocity potential on the boundary is known, i.e a Neumann boundary, the equation reduces to a *Fredholm integral equation of the second kind*, this means that an equation of the form 2.12, must be solved.

$$\int_C \ln(r) \cdot \nabla \Phi(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = F(\mathbf{x}_0) \quad (2.12)$$

where

$$F(\mathbf{x0}) = \pi\Phi(\mathbf{x0}) + \int_C \Phi \cdot \nabla \ln(r) \cdot \mathbf{n}(\mathbf{x}) \cdot d\mathbf{l}(\mathbf{x}) \quad (2.13)$$

In the opposite case, where Φ is known on the boundary, a Dirichlet boundary, the equation to be solved is a *Fredholm integral equation of the first kind*, that is

$$\pi\Phi(\mathbf{x0}) = - \int_C \Phi(\mathbf{x}) \cdot \nabla \ln(r) \cdot \mathbf{n}(\mathbf{x}) \cdot d\mathbf{l}(\mathbf{x}) + H(\mathbf{x0}) \quad (2.14)$$

where

$$H(\mathbf{x0}) = \int_C \ln(r) \cdot \nabla \Phi(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) d\mathbf{l}(\mathbf{x}) \quad (2.15)$$

2.1.3 Discretization of the boundary and system of equations

To be able to solve equation 2.11 in practice, a boundary discretization is needed. This is done, by dividing the boundary into a discrete set of boundary elements, as shown in figure 2.3, for constant elements.

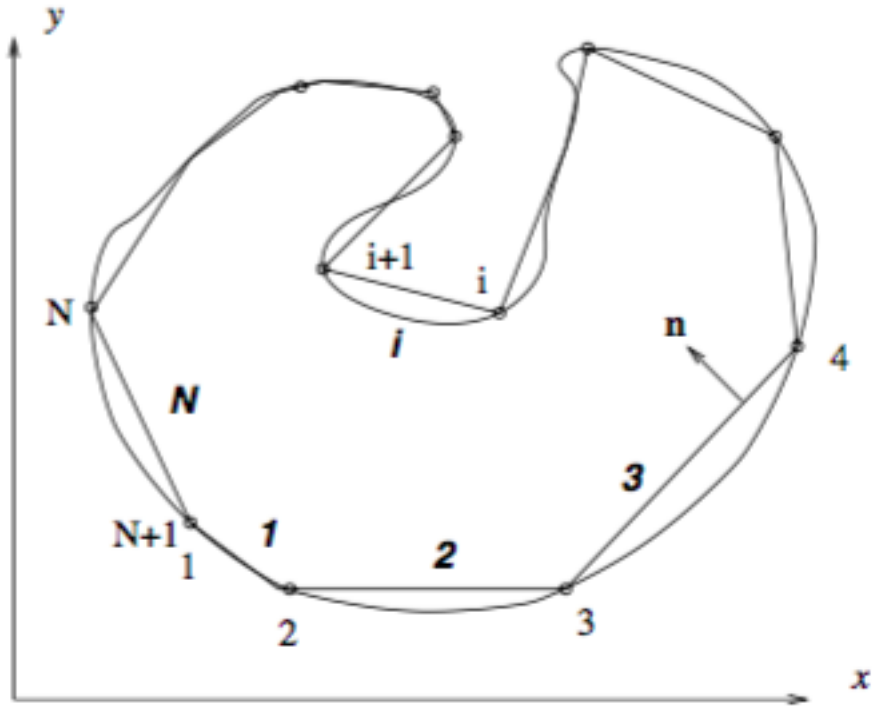


Figure 2.3: Arbitrary body discretized into N constant elements [Pozrikidis \(2002\)](#)

The elements the body are discretized into, can be of different order. The lowest order of elements, are constant elements. In this case an element has two endpoints, and the boundary conditions are satisfied, and the solution found, in the center of the element. By placing the singular point, \mathbf{x}_0 in the center of element i , the following equation is produced:

$$\pi \cdot \Phi_i = \sum_{j=1}^N \int_{E_j} \left(-\Phi(\mathbf{x}) \cdot \nabla \ln(r) + \ln(r) \cdot \nabla \Phi(\mathbf{x}) \right) \cdot \mathbf{n}_j \cdot d\mathbf{l}(\mathbf{x}) \quad (2.16)$$

In the case of constant elements, Φ and Φ_n are assumed to be constant over each element, yielding

$$\pi \cdot \Phi_i = - \sum_{i=1}^N \Phi_j \int_{E_i} \mathbf{n}_j \cdot \nabla \ln(r) d\mathbf{l}(\mathbf{x}) + \sum_{j=1}^N \mathbf{n}_j \nabla \Phi_j \int_{E_j} \ln(r) d\mathbf{l}(\mathbf{x}) \quad (2.17)$$

This can be written in a compact form as:

$$\alpha_i \Phi(\mathbf{x}_0) = - \sum_{j=1}^N \bar{H}_{ij} \Phi_j + \sum_{j=1}^N G_{ij} \Phi_j \quad (2.18)$$

where

$$\bar{H}_{ij} = \int_{E_j} \frac{\partial \ln(r)}{\partial n} dl = \int_{E_j} \frac{\partial}{\partial n} \ln\left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right) dl \quad (2.19)$$

$$G_{ij} = \int_{E_j} \ln(r) ds = \int_{E_j} \ln\left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right) dl \quad (2.20)$$

By doing this for all elements on the body, an equation system with N equations with N unknowns are obtained. By solving the system of equations, the unknown Φ or Φ_n can be found on the midpoint on each element. The resulting matrix equation is:

$$[H]\Phi = [G]\Phi_n \quad (2.21)$$

Where, by utilizing the identity matrix I :

$$[H] = \bar{H} + \pi \cdot I \quad (2.22)$$

In the case of either a pure Dirichlet or Neumann boundary, either the left hand side or right hand side of equation 2.21 is known, and can be computed into a vector directly, resulting in the standard matrix equation to be solved. If the BVP is a mixed problem, consisting of both Neumann and Dirichlet boundary conditions, the matrices must be sorted, such that all unknown quantities is assembled into one matrix, and the known into another.

2.1.4 Local solution

If geometrical singularities are present on the body, BEM does not very well describe the flow field close to the singularity. To deal with this, several different approaches are possible. One

is to introduce a local coordinate system in the apex, and then describe the potential in a small area surrounding the apex, using known analytical corner flow models. In the case of plane two-dimensional flows, the complex potential is defined as:

$$F(z) = \Phi + i\Psi \quad (2.23)$$

For the case of a corner, the potential flow can be described by the complex potential $F(z) = z^m$ Newman (1977). By using polar coordinates, $z = r e^{i\theta}$ this can be written $F(z) = r^m e^{im\theta}$, which can be formulated as:

$$F(z) = r^m (\cos(m\theta) + i \sin(m\theta)) \quad (2.24)$$

To satisfy the impermeability condition, values of m which makes the imaginary part of equation 2.24 (the streamlines) become zero are necessary. By using the coordinate system defined in figure 2.4, one can by inspection see that the following values of m satisfies the zero Neumann condition Liang et al. (2015)

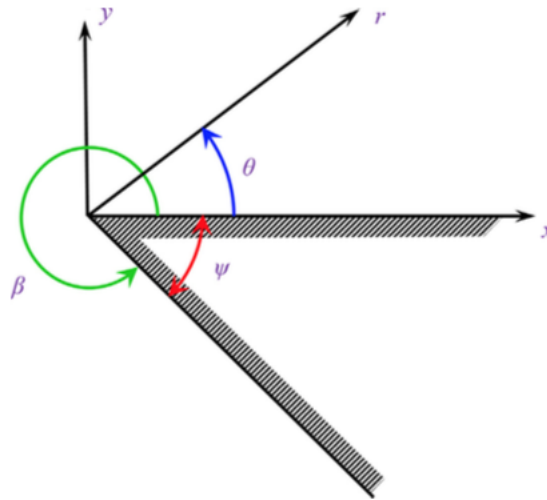


Figure 2.4: Definition of coordinate system for corner Liang et al. (2015)

$$m = \frac{j\pi}{2\pi - \psi} \quad (2.25)$$

Where j is any integer, zero, positive or negative. The negative integers can be excluded by requiring that the fluid velocity flux through a surface with radius r enclosing the apex, approaches zero as $r \rightarrow 0$ [Liang et al. \(2015\)](#). The local solution is meant to be matched with a global solution for the rest of the domain. Therefore a number of unknown constants are introduced to the corner potential. Since all the m 's satisfies the zero Neumann condition, the local potential can be written as the following sum, [Liang et al. \(2015\)](#), changing the variable $F(z)$ to Φ_{loc} .

$$\Phi_{loc} = C_0 + \sum_{i=1}^{N_c-1} C_i r^{m_i} \cos(m_i \theta) \quad (2.26)$$

Where N_c are the number of terms included in the local solution.

2.1.5 Matching equations

By using a local analytical solution to describe the potential around a corner, new unknowns have been introduced. The system now consists of N equations, with $N + N_c$ unknowns. Since this system is underdetermined, artificial matching boundaries, Σ_m are introduced, at the intersections between the local and global domains. By requiring continuity of Φ or Φ_n on the matching boundaries, new equations are produced. [Figure 2.5](#) illustrates a body with 4 corners. Here matching nodes have been distributed around the corners, and along the boundary of the body.

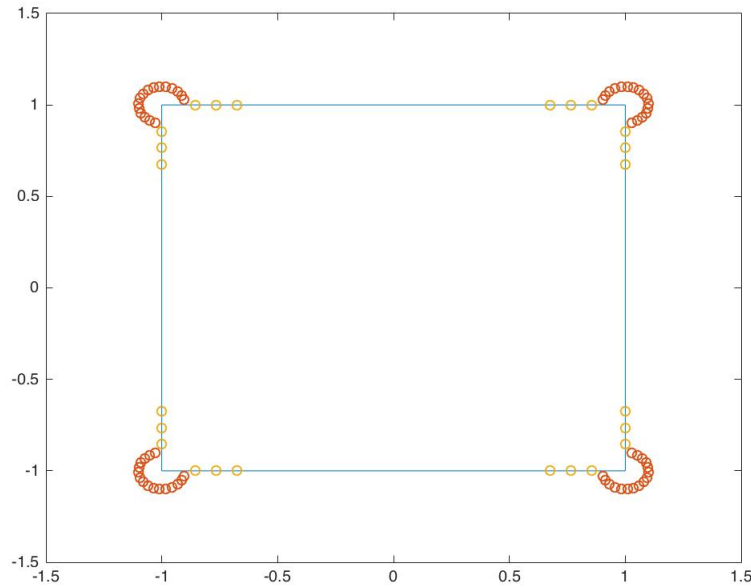


Figure 2.5: Matching nodes distributed around corner, and along boundary

When it comes to the matching equations there are several possibilities. One of which is direct matching, which means that we state that the global and local solution have to be the same on the matching nodes, that is:

$$\Phi_{loc} = \Phi \quad \text{on } \Sigma_m \quad (2.27)$$

In this case, the number of nodes on Σ_m has to be the same as the number of unknown constants, to have an unique system of equations.

Another possibility is the use the method of Least Squares, as described in appendix B. Here, the constants which minimizes the error committed when approximating a polynomial to the discrete values of the potential on the matching boundaries, are found. A great advantage by using this method, is that the number of constants *does not* have to be the same as the number of matching nodes. For the local potential described by equation 2.26, this results in N_c equations of the form:

$$\begin{aligned}
\frac{\partial E}{\partial C_0} &= 2 \sum_{i=1}^N (\Phi_i - (C_0 + \sum_{j=1}^{N_c} C_j r_i^{m_j})) \cdot 1 = 0 \\
&\vdots \\
\frac{\partial E}{\partial C_j} &= 2 \sum_{i=1}^N (\Phi_i - (C_0 + \sum_{j=1}^{N_c} C_j r_i^{m_j})) \cdot (-r_i^{m_j} \cos(m_j \theta_i)) = 0 \\
&\vdots \\
\frac{\partial E}{\partial C_{N_c}} &= 2 \sum_{i=1}^N (\Phi_i - (C_0 + \sum_{j=1}^{N_c} C_j r_i^{m_j})) \cdot (-r_n^{m_j} \cos(m_j \theta_i)) = 0
\end{aligned} \tag{2.28}$$

A complication occurs, if matching nodes which are not located on the boundary of the body are used. This is because, in BEM the unknown Φ or Φ_n is firstly found on the boundary of the domain, and then in the rest of the fluid domain. By using matching nodes which are not on the boundary, we thus have to express the potential at these nodes, as a function of the potential at the boundary nodes. By denoting the potential on Σ_{m_i} in the outer domain as $\bar{\Phi}$, the potential at these nodes can be written as

$$\bar{\Phi}_{\Sigma_{m_i}} = f(\Phi_1, \dots, \Phi_N, C_0, \dots, C_{N_c}) \tag{2.29}$$

Where the function f is the discrete form of equation 2.11 with $\alpha = 2\pi$. $\bar{\Phi}$ is then implemented in equation 2.28, to find the constants which minimizes the error.

2.1.6 Coupled Method

To be able to use the BIF derived using Greens second identity, the integral must be around a closed surface. Hence the local solution must be coupled with the global, and solved simultaneously. By extension of equation 2.18, this yields the following BIF:

$$\alpha \Phi = - \int_{C_{glob}} \Phi \frac{\partial}{\partial n} \ln(r) dl + \int_{C_{glob}} \frac{\partial \Phi}{\partial n} \ln(r) dl - \int_{C_{loc}} \Phi_{loc} \frac{\partial}{\partial n} \ln(r) dl + \int_{C_{loc}} \frac{\partial \Phi_{in}}{\partial n} \ln(r) dl \tag{2.30}$$

By discretizing the global part of the boundary, and using the fact that $\frac{\partial \Phi_{loc}}{\partial n} = 0$, the BIF can be written in compact form as:

$$\alpha\phi(x_i, y_i) = - \sum_{j=1}^N H_{ij}\phi_j + \sum_{j=1}^N G_{ij}\phi_{nj} - \int_{S_{in}} \Phi_{loc} \frac{\partial}{\partial n} \ln(r) dl \quad (2.31)$$

If N_C constants are used in the local solution, all unknowns can be assembled into a system matrix with dimension $(N + N_C) \times (N + N_C)$. Here the N first rows and columns corresponds to the unknown Φ or Φ_n at the N global nodes, and the N_C last columns to the unknown constants in the local solution. The last N_C equations are the matching equations, either direct or least square. The RHS of the matrix equation, is a vector \mathbf{b} , which N first terms corresponds to the global domain, and the terms $N+1$ to $N+N_C$ to the matching equations, for the direct case:

$$b_{N+i} = 0 = \Phi_{out} - \Phi_{loc} \quad i = 1, \dots, N_C \quad (2.32)$$

The system matrix is then solved for the unknown Φ and/or Φ_n and the unknown constants C_0, \dots, C_{N_C}

2.1.7 Free surface effects

Numerous problems within hydrodynamics, include problems where the free surface is present and must be accounted for. This is for instance the case in slamming, where bodies with large velocities are forced through the free surface. In the following are methods to analyzing 2 dimensional bodies oscillating with infinite frequency in free surface outlined. These problems are relevant for cases such as high frequency ship vibrations.

If linear theory is applied, the free surface condition yields:

$$-\omega^2\Phi + g \frac{\partial\Phi}{\partial y} = 0 \quad \text{on } y = 0 \quad (2.33)$$

Where ω is the frequency and g the gravitational acceleration. For the case of $\omega \rightarrow \infty$, the frequencies are much larger than g , and the free surface condition simplifies to:

$$\Phi = 0 \quad \text{on } z = 0 \quad (2.34)$$

This means that there is no velocity in the tangential direction of the surface, only in the normal direction. Hence no waves are generated.

There are several methods to analyze problems where bodies are oscillating in the free surface. One of which, is to use Green's functions which satisfies the free surface condition. For the infinite frequency case, this can be shown to be: [Faltinsen \(1990\)](#)

$$g(x, y) = \ln(r) - \ln(r') \quad (2.35)$$

where

$$\ln(r) = \ln\left(\sqrt{(x - x_0)^2 + (y - y_0)^2}\right) \quad (2.36a)$$

$$\ln(r') = \ln\left(\sqrt{(x - x_0)^2 + (y + y_0)^2}\right) \quad (2.36b)$$

This is in the literature referred to as solving the double body problem. The reason being that, sources are distributed over the closed surface, and over an image surface with positive y coordinates. Equation 2.36 represents the source potential as a combination of a source in infinite fluid, and an image sink above the free surface.

In order to use Green's second identity, one must integrate over a closed surface. One possible subdivision of the the surface is illustrated in figure 2.6

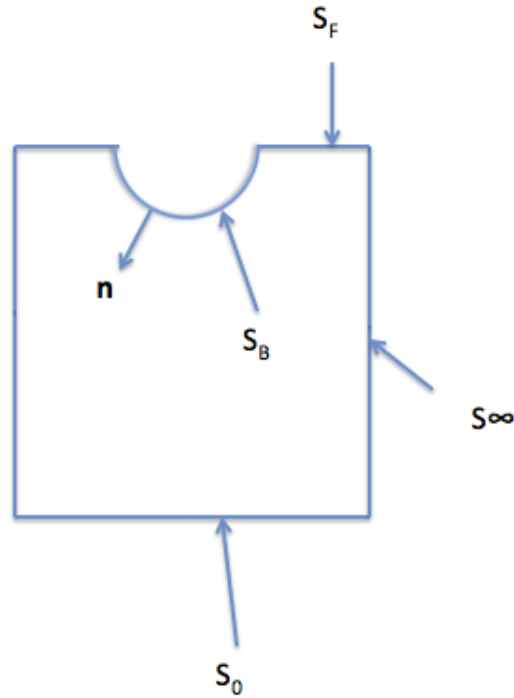


Figure 2.6: Integration surfaces in two dimensions

For deep water, it can be shown that that the integration over the surface $S = S_B + S_F + S_0 + S_\infty$ reduces to a integral over the mean wetted surface [Faltinsen \(1990\)](#). This is because other than at the location of the source (x_0, y_0) , the Green's function satisfies the Laplace equation everywhere in the fluid domain, and the appropriate boundary conditions. The linear free surface condition is $\Phi = 0$ on $z=0$, and hence also $G = 0$ on $z=0$. This means that the integral reduces to:

$$\int_{S_F} \left(-\frac{\partial g}{\partial n} \Phi + \frac{\partial \Phi}{\partial n} g \right) = 0 \quad (2.37)$$

It can also be shown that the integration over S_∞ disappears, and if deep water is assumed there is no contribution from S_0 . [Faltinsen \(1990\)](#). The result is that Φ at (x_0, y_0) can be written as:

$$\Phi(x_0, y_0) = \int_{S_B} \left(-\frac{\partial g}{\partial n} \Phi + \frac{\partial \Phi}{\partial n} g \right) \quad (2.38)$$

Another method to solve problems including surface piercing bodies, is to make use of the free

space Green's functions. The motivation for doing this, is that in the case of the frequency not being 0 or ∞ , the expression for the Green's function is very complicated, and not trivial to evaluate. In this case regular Rankine sources are distributed over the closed surface. Also here, the integrals over S_∞ and S_0 disappears, but integration over the free surface must be done. This means that Φ at (x_0, y_0) equals:

$$\Phi(x_0, y_0) = \int_{S_B + S_F} \left(-\frac{\partial G}{\partial n} \Phi + \frac{\partial \Phi}{\partial n} G \right) \quad (2.39)$$

It is here important that S_F is satisfactory long, such that the effects of S_∞ are negligible. To avoid an unpractical large equation system, one possibility is here to use grid stretching.

In the latter of the two methods, the BVP to be solved is on a general Robin form, meaning that there is both Dirichlet and Neumann boundary conditions present. Care must therefore be shown when solving this problem. For the case of wall sided boundary conditions, which means that the two boundaries where the boundary condition change from one to the other is perpendicular to each other, the problem can be solved directly. While if this is not the case, convergence problems will typically arise in the intersection between the two boundaries.

Chapter 3

Numerical aspects -Error and Convergence

Errors are unavoidable, when using numerical models approximate continues problems. The reason for this is that we try to solve a continues problem, using discrete equations. The most important source in error in BEM, is the discretization error, connected with approximating the physical boundary with a numerical boundary. Other sources of error are round-off errors and errors due to approximating integrals using numerical techniques. In order to be confident that the results obtained from a numerical method are reliable, one must somehow be able to measure error, and verify that it becomes smaller, as the number of elements are increased. When the error becomes smaller than a certain threshold, one can conclude that the solution has converged.

3.1 Local error, Order of Accuracy and Grid Convergence Index

The most important source of error in BEM is the *discretization error*, which is defined as the difference between the exact solution of the governing equation, and the exact solution of the discrete problem [Ferziger and Perić \(2002\)](#). For the geometry in figure 3.1, this means the difference between integrating equation 3.1 over the exact boundary, and equation 3.2 over the numerical boundary *and* approximating the variation of Φ and Φ_n using shape functions. In the case of constant elements this means either 1 or 0.

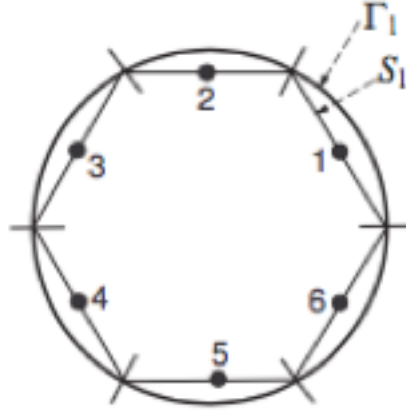


Figure 3.1: Example of Discretized boundary [Dijkstra et al. \(2011\)](#)

$$\alpha \cdot \Phi(\mathbf{x}\mathbf{0}) = \int_{\Gamma} \left(\Phi(\mathbf{x}) \cdot \nabla G(\mathbf{x}, \mathbf{x}\mathbf{0}) - G(\mathbf{x}, \mathbf{x}\mathbf{0}) \cdot \nabla \Phi(\mathbf{x}) \right) \cdot \mathbf{n}(\mathbf{x}) \cdot dl(\mathbf{x}) \quad (3.1)$$

$$\alpha_i \cdot \Phi_i = \sum_{j=1}^N \Phi_j \int_{S_j} n_j \cdot \nabla \ln(r) dl(\mathbf{x}) - \sum_{j=1}^N n_j \nabla \Phi_j \int_{S_j} \ln(r) dl(\mathbf{x}) \quad (3.2)$$

Since all equations in this thesis are solved in a direct manner, i.e not by using iterations, there is no iteration error, which on the contrary would necessary to account for. In addition to this, round-off errors due to the arithmetic limitations of the computer and quadrature errors if the integrals are evaluated using Gauss quadrature, are important to be aware of.

The truncation error is the error committed when approximating an infinite sum by a finite sum. In BEM this is related to discretizing the boundary into a finite sum of elements. In error analysis, it is differentiated between the local and global error. The local error is the error at a solution point after one iteration, while the global truncation error is the cumulative error after many iterations. For problems with smooth boundaries, the quality of an approximation is described in terms of its order, which relates the truncation error to the grid spacing of some order, p . [Ferziger and Perić \(2002\)](#). In BEM, the local error will be one order higher than the global error. For the case of constant elements, p can be shown to be 3 and 2, respectively. [Dijkstra et al.](#)

(2011).

One simple way to estimate the discretization error is based on Richardson's extrapolation. This method is used to obtain higher order estimates of the exact continuous solution Φ_{exact} , from a series of lower order discrete values. If the solution is approximated with the function $\Phi(h)$, the error may be defined as:

$$E = \Phi(h) - \Phi_{exact} = Ch^p + H.O.T \quad (3.3)$$

Where C is a constant and H.O.T stands for Higher Order Terms. For certain numerical methods such as BEM, the theoretical order of accuracy is known. The observed numerical order, will likely be lower, due to boundary conditions, complicated geometries, and the mesh the used in the solver. If the grid is refined systematically by a factor r, an estimate of the the observed order of accuracy is: [Ferziger and Perić \(2002\)](#)

$$p = \frac{\log\left(\frac{\phi_2 - \phi_1}{\phi_3 - \phi_2}\right)}{\log(r)} \quad (3.4)$$

Here Φ_3 is the solution from coarsest mesh, and Φ_1 from the finest. This equation is valid only if the two finest grids are fine enough. Another way of saying this is that the error, must be within the asymptotic range of convergence. If a second order solution is assumed, and Φ is computed on the same node, using two different spacings h_1 and h_2 , Richardson extrapolation can be used to approximate the continuum solution, Φ_{h0} . This is obtained by expanding the series approximation of the function, for both spacings. The resulting approximation is:

$$\Phi_{h0} \cong \Phi_1 + \frac{\Phi_1 - \Phi_2}{r^p - 1} \quad (3.5)$$

The difference between Φ_{h0} and Φ_1 is an error estimate. However, this requires detailed considerations of the caveats which lead to it [Ferziger and Perić \(2002\)](#). Using Φ_1 and Φ_2 , an alternative error approximation is:

$$E_1 = \frac{\epsilon}{r^p - 1} \quad (3.6)$$

where

$$\epsilon = \frac{\Phi_2 - \Phi_1}{\Phi_1} \quad (3.7)$$

To check if the solution is within the asymptotic range of convergence, the Grid Convergence Index, GCI, as suggested by [Roache \(1998\)](#), is used. It is defined as:

$$GCI = \frac{F_s * |\epsilon|}{r^p - 1} = |E_1| \cdot F_s \quad (3.8)$$

Here F_s is a safety factor, which is 1.25 if three or more grids have been used, and 3 if only two have been used. It is important to note that the GCI is not an error estimate, but F_s time the error estimator, representing an error band. If the solution is within the asymptotic range of convergence, equation 3.9 represents the relationship between the three grids. By dividing the RHS by the LHS of equation 3.9 one gets a fraction F , as in equation 3.10, which should be close to unity.

$$GCI_{21} = r^p \cdot GCI_{32} \quad (3.9)$$

$$F = \frac{r^p \cdot GCI_{32}}{GCI_{21}} \quad (3.10)$$

3.2 Sensitivity to parameters in local potential solution

Φ_{loc} as defined in equation 2.26 is a sum, where the constants C_0, \dots, C_{N_c} are unknown and determined by matching with the global solution. How many constants to include, the location of the matching nodes, and the size of the local domain, have to be set a priori, to conducting an

analysis. The solutions sensitivity to different combinations of these parameters must hence be investigated.

One criterion to find the best combination of these variables, is to find the local solution which best matches the global, at the 10 global nodes closest to the local domain, proposed by Jens Helmers in DNV GL. To this end, Φ_{loc} is calculated at the 10 global nodes, and the standard deviation, STD between the two solutions is found, using equation 3.11

$$STD = \sum_{i=1}^{i=N} \sqrt{\frac{1}{N} (\Phi_i - \Phi_{loc_i})^2} \quad (3.11)$$

By doing this for all possible combinations, of N_C , a , N_{out} and N_B , and identifying the minimum, this can be taken to be the best fit solution.

Chapter 4

Comments on computer code

In this thesis, various BVPs using BEM are solved. To this end BEM is implemented in Matlab. Even though all problems are solved uniquely, many of the programs have very similar structure and only details such as geometry, boundary conditions, local flow model etc. separate them. Therefore only a part of the Matlab code is included in the appendix. What the specific programs do, and implementation details of some representative programs are described shortly in the following. For the full code, the reader is referred to appendix [E](#).

To solve a particular BVP, a series of functions are made, each solving a specific problem. All the functions are then run by a main program. The codes presented here, are *not* from the same problem, but rather examples of functions used in different BVPs, presented here to give the reader an idea of the general approach to solve the problems.

4.1 Geometry.m

The function `geometry.m` discretizes the geometry. Here the end points of all elements are defined, along with the normal and tangential vector and collocation points. Further the size of the local domain is set, and the location of boundary and outer matching nodes are defined. Lastly the boundary condition vector at the collocation points are calculated.

4.2 **matcalc.m**

In function `matcalc.m`, integration over the global domain is performed. For each element, the integrated values of the G and G_n are assembled into matrices. Depending on whether Φ or Φ_n is known on the collocation point, the matrices are sorted, saving all known elements in one matrix, and unknown in another.

The integrals evaluated in this function, are solved analytically. For details on how this is done, the reader is referred to [Appendix C](#).

4.3 **Corner.m**

In function `corner.m` integration over the local domain is done. The function is only utilized in cases where the local corner solution are used. The integrals evaluated here, are very complex to solve analytically, and are hence solved numerically. Details of this is given in [C](#).

4.4 **Leastsquare.m**

In function `LeastSquare.m` least square matching between the local and global solution is performed.

4.5 **Sysmat.m**

The function `Sysmat.m` takes in all the matrices and vectors from the main program, and assembles the system matrix, which is solved for the unknown Φ , Φ_n and C 's.

4.6 Sensitivity

To check the solutions sensitivity to the parameters in the local solution, the main program is looped, and different combinations of NC , a , N_{out} and N_B are tested. For each combination the STD between the global solution at the 10 nodes closest to the local domain, and the local solution at these nodes are calculated. The solution with the least STD is chosen.

4.7 Convergence

In the convergence study the main program is looped. For each iteration the grid is refined by a factor of 2, and Φ and/or Φ_n are calculated and saved for 5 control nodes on the body surface and/or the free surface. For the saved values, p , GCI , F and Φ_{h0} are calculated. In the calculations of the convergence variables, the location of the control nodes are at the same geometrical location each time.

Chapter 5

Infinite fluid problems

In this chapter, are results for the analysis's done on 2 dimensional bodies in infinite fluid. Firstly a square with rounded corners is analyzed. The results here are compared to square with sharp corners and a circle. Following this, are square cylinders, where a local corner flow model is introduced. The effect of different number of matching nodes, location of matching boundaries, and matching the local and global solution directly versus least square method is investigated. Lastly, an oscillating double wedge is analyzed for different dead rise angles. Results from the convergence studies are also presented.

5.1 Boundary Value Problem

The Boundary value problem, BVP, for the first part of this chapter is defined in the following. The problem consists of a 2D cross sectional geometry in infinite fluid. The body is either in a constant current as shown in figure 5.1 for a circle, or is oscillating.

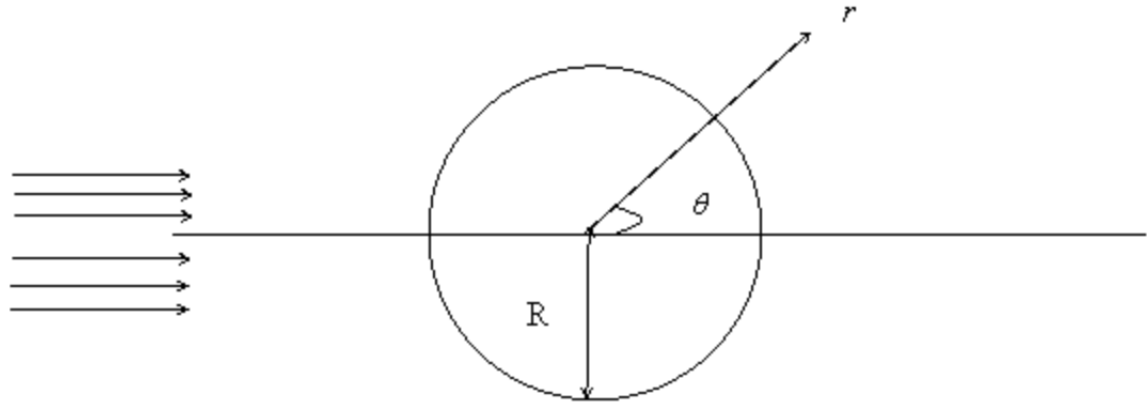


Figure 5.1: Object in constant current

Assuming ideal fluid, the problem is reduced to one governing equation, namely the Laplace equation as defined in equation 2.1. For the current problems, the potential is split into two parts, the current part Ux , and an unknown part Φ_s giving the total potential as:

$$\Phi = \Phi_s + Ux \quad (5.1)$$

By requiring the cylinder to impermeable, the problem is reduced to a pure Neumann problem, with boundary condition:

$$\frac{\partial \Phi_s}{\partial n} = -n_x U \quad (5.2)$$

Where n_x is the outward normal derivative in x direction. For all current problems, the current velocity is set to $U=3$. For problems including oscillating cylinders, the problem is somewhat simplified. Here the potential only consists of one term, which is unknown, and the body boundary condition is $\frac{\partial \Phi}{\partial n} = -n_x$.

5.2 Rounded Square

To visualize how Φ changes, going from from a circle geometry to a square, the square cylinder is modeled as a "rounded square". To model it, a circle with radius less or equal to half of the length of the square is put into each corner. At the two points in the corner where the circle and the square intersect, the circle geometry, replaces the sharp corner. By gradually reducing the circle radius, the corners become "sharper and sharper". An example of the geometry is given in figure 5.2 for $r=0.5$. The model have a total length, $L=2$, and thus a corner radius which is $0 < r < 1$. The results are compared to regular square with "sharp" corners.

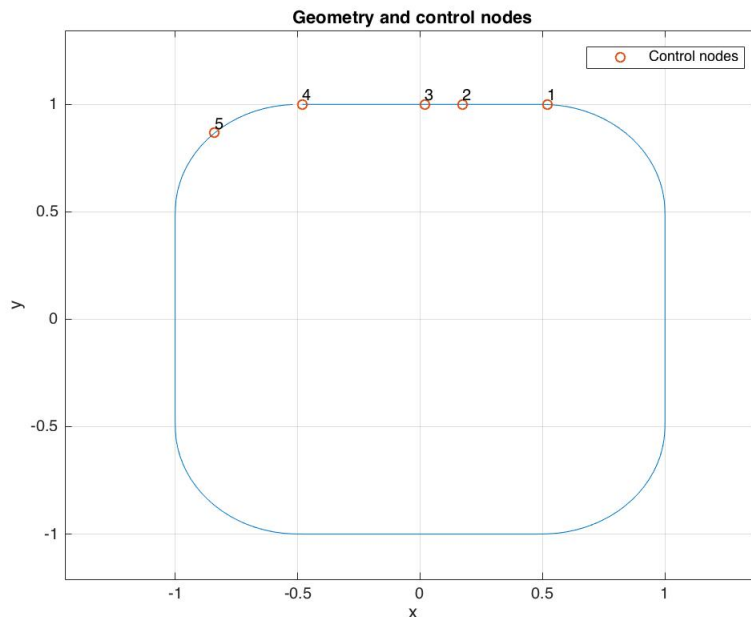


Figure 5.2: Geometry of square with rounded corners with $r=0.5$ and location of control nodes.

Before any results can be presented, it must be ensured that the solutions is within the asymptotic range of convergence. A brief check is here done for the case of $r=0.5$. 3 different grids are used, all refined with a factor of 2 for each iteration. GCI and p are calculated at the 5 control nodes, illustrated in figure 5.2. The coarsest grid has $N=160$ nodes and the finest $160 \cdot 2^2 = 640$ nodes. The analysis is done for the current problem. The result are given in table 5.1.

Table 5.1: GCI, p and F for rounded square in constant current

| Node | p | GCI_{32} | GCI_{21} | F |
|------|-------|-----------------------|-----------------------|--------|
| 1 | 2.003 | 0.0001282 | $3.199 \cdot 10^{-5}$ | 1.0005 |
| 2 | 2.002 | 0.0001514 | $3.778 \cdot 10^{-5}$ | 0.9998 |
| 3 | 1.942 | $9.177 \cdot 10^{-5}$ | $2.388 \cdot 10^{-5}$ | 1.0002 |
| 4 | 1.99 | $9.382 \cdot 10^{-6}$ | $2.362 \cdot 10^{-6}$ | 0.9999 |
| 5 | 2.129 | $6.594 \cdot 10^{-5}$ | $1.508 \cdot 10^{-5}$ | 0.9997 |

From table 5.1, it is observed that p is satisfactory close to theoretical value of 2, for all control nodes except node 5, which is higher. All error bands are satisfactory small, and F close to unity. Hence the solution is concluded to be within the asymptotic range of convergence. The largest error-band using 160 elements is about 0.015%, this is therefor found to be a fine enough grid to analyze this problem.

For the current problem, Φ for the rounded square with different corner radii are found, and presented in figure 5.3. The radius of curvature is varied from a maximum of 0.9 to a minimum of 0.01. The number of elements, as can be seen in the plots, are in some cases a lot higher than 160. The reason for this is that the size of neighboring elements should not differ much [Faltinsen and Tymocha \(2009\)](#). A minimum of elements are need to describe the changing geometry of the circle section the corner. Thus, when the radius of curvature is small, the elements here becomes small, and because no stretching of the grid is preformed, the result is a high number of elements. For comparison, Φ for a circle and a square cylinder with sharp corners are also found, given in figure 5.4.

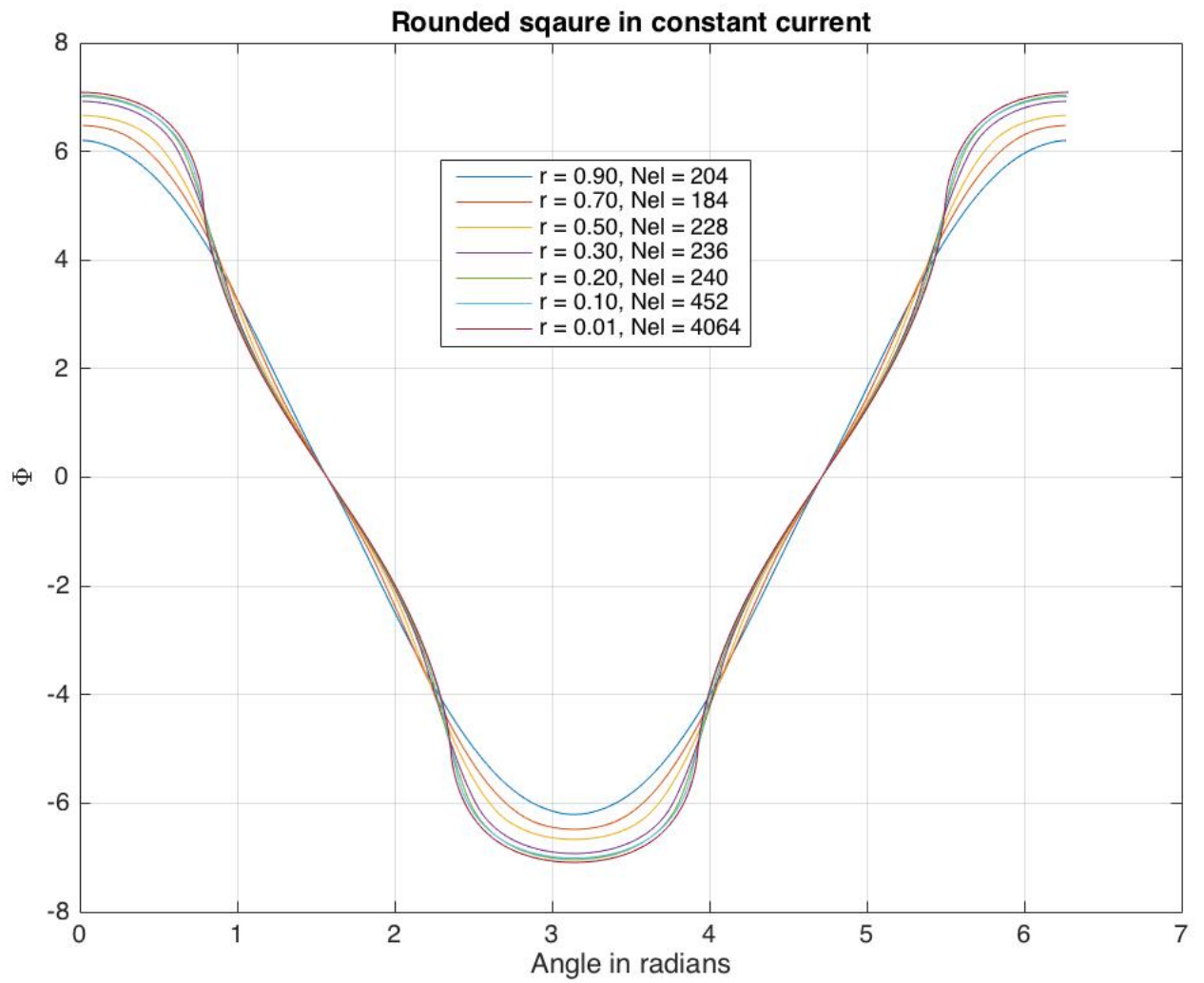


Figure 5.3: Velocity potential for rounded square in constant current, $U=3$.

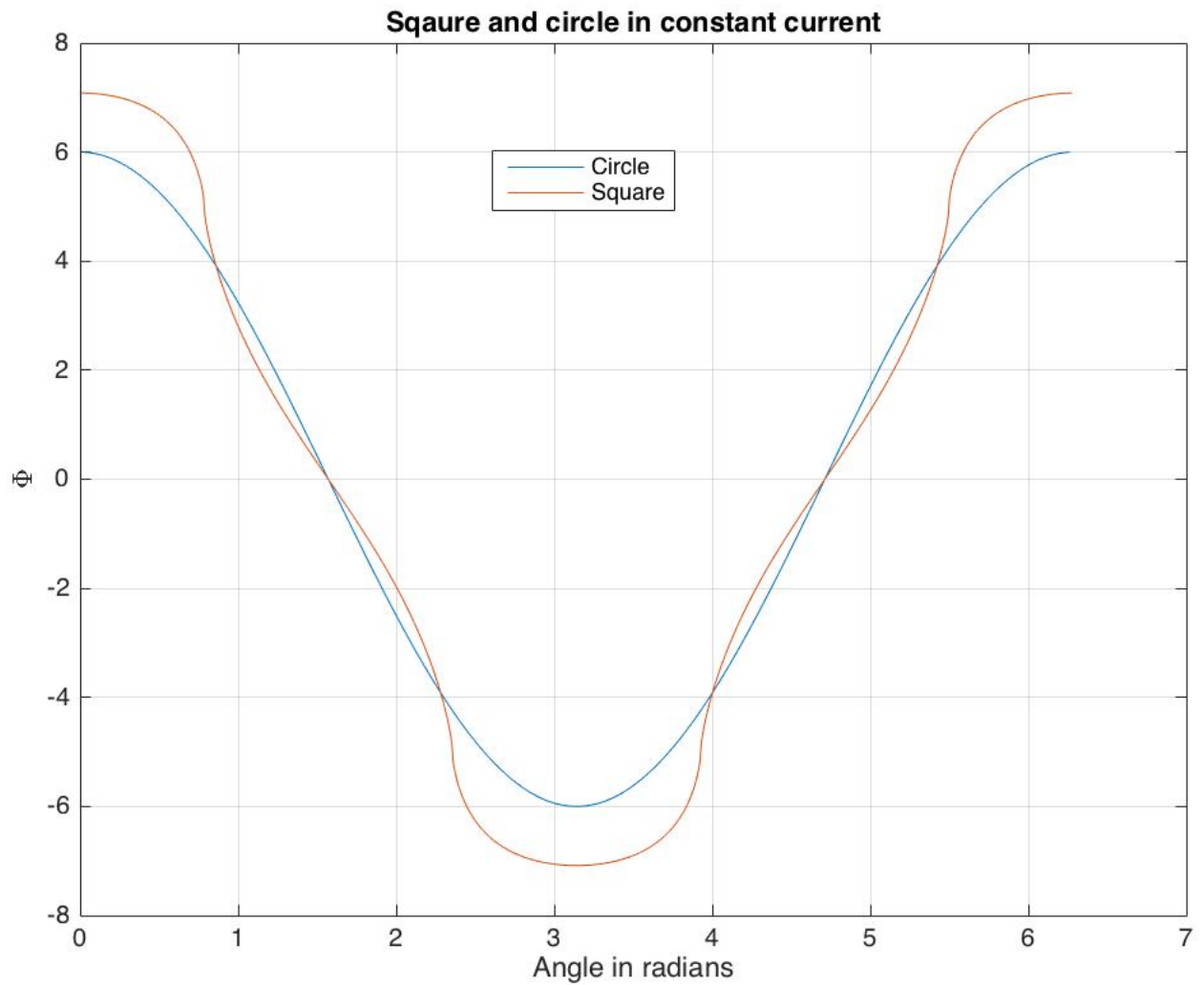


Figure 5.4: Velocity potential for square and circle in constant current, $U=3$.

In figure 5.3 and 5.4 it is observed larger extremes of Φ , going from a geometry of a circle, to a square. The extremes found for all radii are given in table 5.2. Here, $r=1$ represents a circle and $r=0$, a square with sharp corners.

Table 5.2: Minimum and maximum values of Φ for different radii of curvature

| r | 1.0 | 0.9 | 0.7 | 0.5 | 0.3 | 0.2 | 0.1 | 0.01 | 0 |
|--------------|--------|--------|-------|--------|--------|--------|-------|--------|--------|
| Φ_{max} | 5.998 | 6.112 | 6.48 | 6.693 | 6.894 | 6.986 | 7.06 | 7.086 | 7.088 |
| Φ_{min} | -5.998 | -6.112 | -6.48 | -6.693 | -6.894 | -6.986 | -7.06 | -7.086 | -7.088 |

It is of interest to find the added mass of the the cylinder oscillating with infinite frequency. The added mass is found in the same manner as described in [Faltinsen \(1990\)](#) and compared to the Lewis form added mass for a circle and square. All values are normalized by $\rho\pi(L/2)^2$. The resulting values of the added mass are given in [figure 5.5](#).

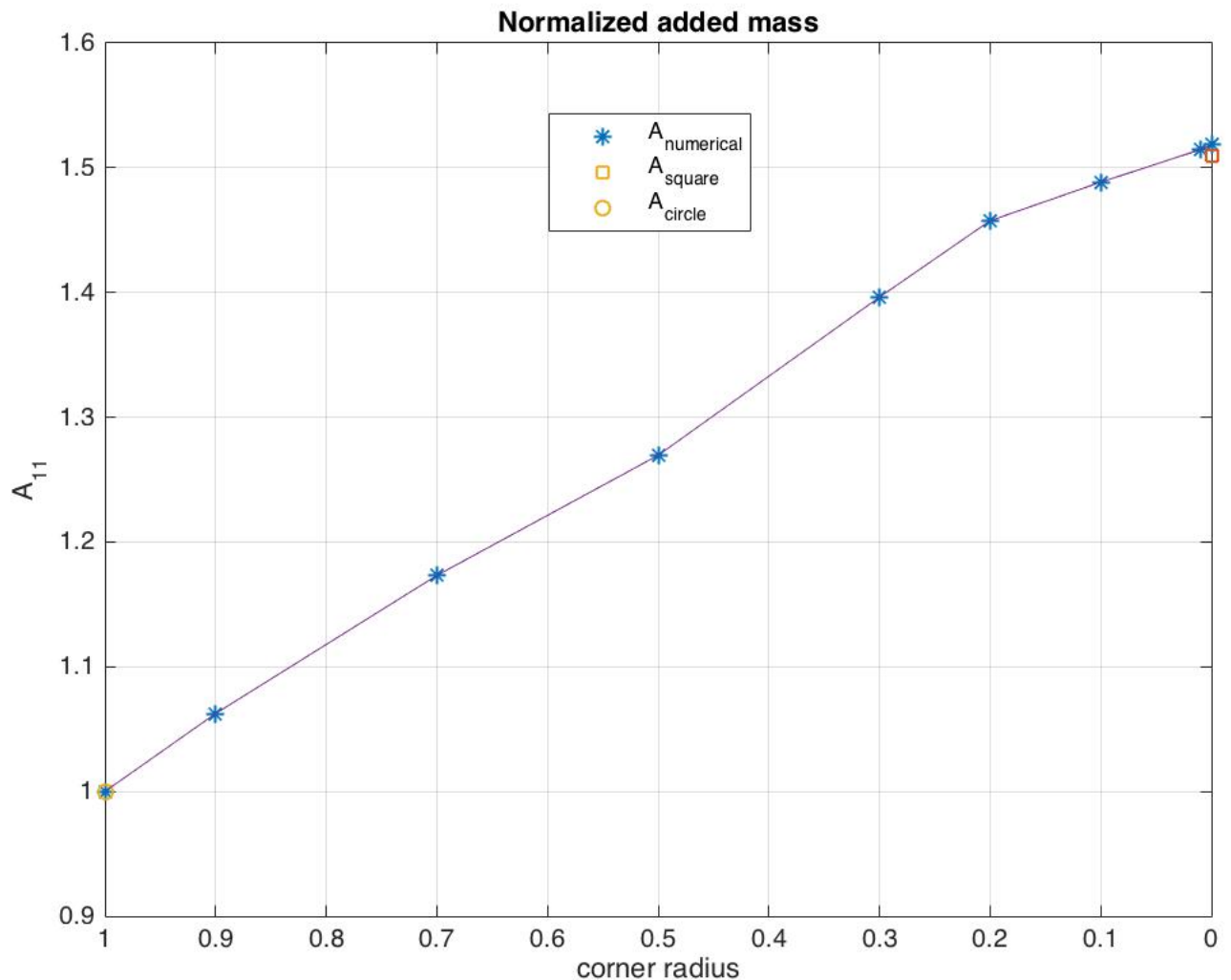


Figure 5.5: Normalized added mass of a rounded square for different radii of curvature, circle and square with sharp corners

From [figure 5.5](#), we see that the added mass increases as the radii of curvature decreases. For comparison, the numerical values of the added mass found for $r=1$, $r=0.9$, $r=0.01$ and $r=0$, are given in [table 5.3](#) with the Lewis form added mass for circle and square. As can be seen, the

numerical value matches the analytical exactly for the circle, and is slightly overestimated for the square. It is also observed very little deviation between the rounded square with $r=0.01$ and the square with sharp corners.

Table 5.3: Added mass for different radii of curvature

| $A_{Circle-Lewis}$ | $A_{Square-Lewis}$ | $A_{r=1}$ | $A_{r=0.9}$ | $A_{r=0}$ | $A_{r=0.01}$ |
|--------------------|--------------------|-----------|-------------|-----------|--------------|
| 1 | 1.51 | 1.0000 | 1.082 | 1.5185 | 1.5145 |

5.3 Square, direct matching

In the following are the results obtained from analyzing the flow around a square cylinder, using the analytical corner model, given in equation 2.26 around the corners. The matching between the global and local velocity potential is done directly. A sensitivity study to the size of the local domain is conducted, and the effect of using different numbers of matching nodes at different locations, is looked into.

5.3.1 Sensitivity and convergence

Prior to conducting analysis, the size of the local domain, the number of constants in the local solution and the location of the matching nodes have to be set. Using direct matching it is emphasized that the number of constants and number of matching node have to be the same. To find the optimal combination of these, a sensitivity study is conducted. All analysis are done using 640 elements uniformly distributed over the body surface, except for when a is varied. The reason for this, is that the 10 nodes being checked should be at the same distance from the apex. i.e the increment h , between the global nodes should be the same for all the global nodes. By setting h , such that 640 elements are used for $a = 0.1$ the appropriate number of nodes are found, and the STD are calculated for all cases.

To find the combination of a , N_{out} , and N_B , which gives the best solution, the criterion defined in chapter 3.2 is used. Firstly the number of outer matching nodes are considered. Here a and N_B are kept constant at 0.05 and 2 respectively. N_{out} is then varied from a minimum of 0 to a

maximum of 16. The standard deviation at the 10 global nodes closest to the corner is calculated for each combination, given in table 5.4.

Table 5.4: Standard deviation for different numbers of outer domain matching nodes

| | $N_{out} = 0$ | $N_{out} = 2$ | $N_{out} = 4$ | $N_{out} = 8$ | $N_{out} = 16$ |
|-------|---------------|---------------|---------------|---------------|--------------------|
| STD | 0.2736 | 47.82 | 161.7 | 1605.0 | $2.939 \cdot 10^5$ |

From table 5.4 one sees that STD increases, as more terms are added. This is assumed to be true for all combinations of N_B and a . Looking for the best fit solution, N_{out} is hence set to zero. For a and N_B the covariance is not so clear. Because of this, all possible combinations between a and N_B are tested, presented in table 5.5.

Table 5.5: Standard deviation for different numbers of boundary matching nodes, and local domain size.

| N | a | $N_B = 2$ | $N_B = 4$ | $N_B = 6$ |
|-------|------|---------------|-----------|-----------|
| 708.0 | 0.01 | 2.009 | 8.568 | 27.16 |
| 700.0 | 0.02 | 0.9693 | 2.615 | 6.629 |
| 692.0 | 0.03 | 0.5872 | 1.195 | 2.721 |
| 688.0 | 0.04 | 0.3773 | 0.652 | 1.451 |
| 680.0 | 0.05 | 0.2454 | 0.4134 | 0.9432 |
| 672.0 | 0.06 | 0.1851 | 0.294 | 0.6958 |
| 664.0 | 0.07 | 0.1758 | 0.2345 | 0.5605 |
| 660.0 | 0.08 | 0.184 | 0.2058 | 0.4746 |
| 652.0 | 0.09 | 0.2015 | 0.1972 | 0.4241 |
| 644.0 | 0.1 | 0.2196 | 0.1989 | 0.3901 |

Emphasized in table 5.5, it is found that the solution that gives the smallest standard deviation, is one where $a = 0.07$ and $N_B = 2$. In figure 5.6 the local solution at 10 the global nodes is plotted with the global solution

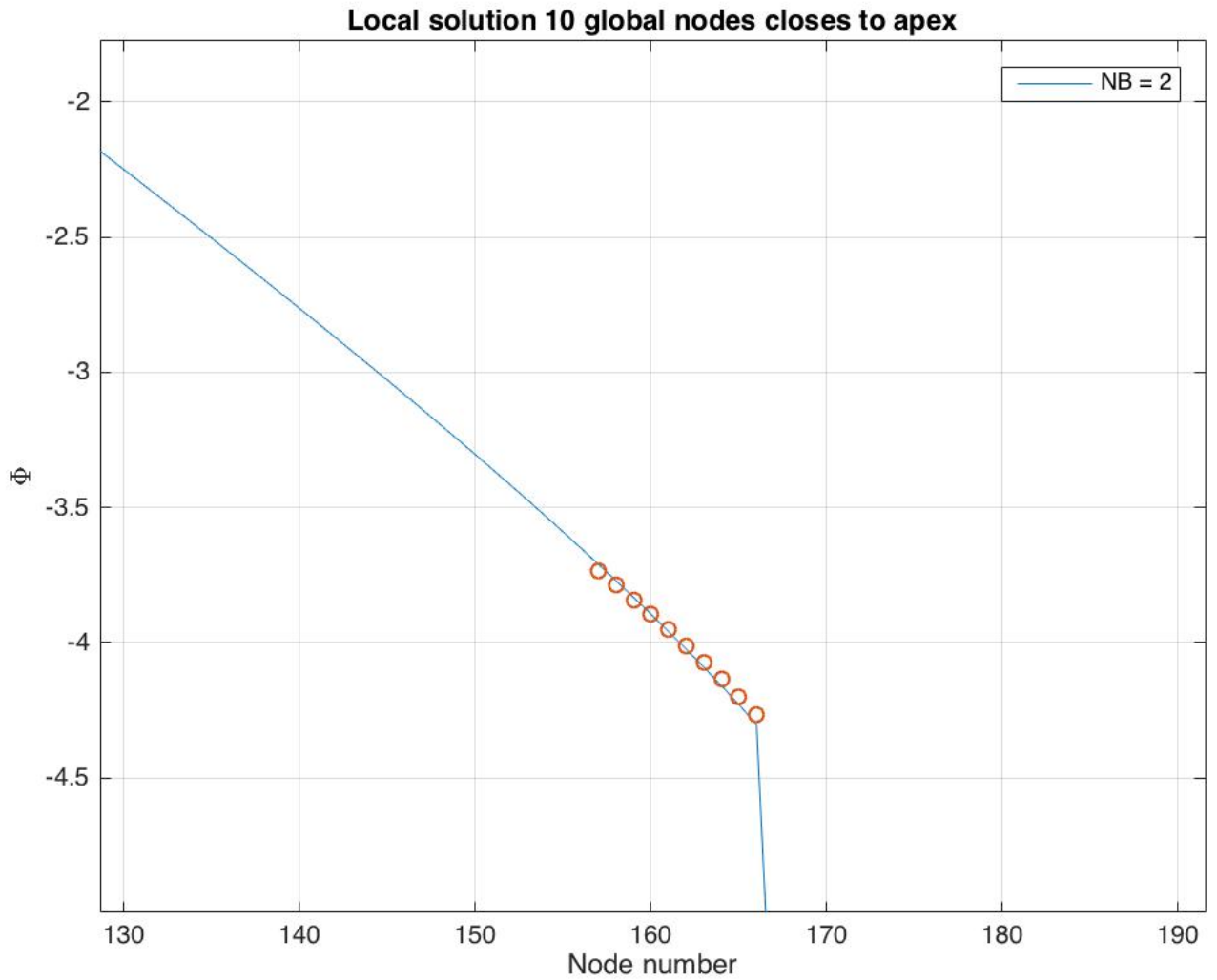


Figure 5.6: Φ_{loc} at 10 global nodes closest to the local domain of the top left corner, plotted with global velocity potential

Before any results can be presented, it has to be made sure that the solution is within the asymptotic range of convergence. To this end, p , GCI and F are calculated at the 5 control nodes illustrated in figure 5.7. Solutions are computed for 3 different grids, using a refinement factor $r=2$. In the coarsest grid, 160 nodes are used and in the finest $160 \cdot 2^2 = 640$ nodes are used. The numerical values are given in table 5.1. In all calculations $a = 0.07$, $N_B = 2$ and $N_{out} = 0$.

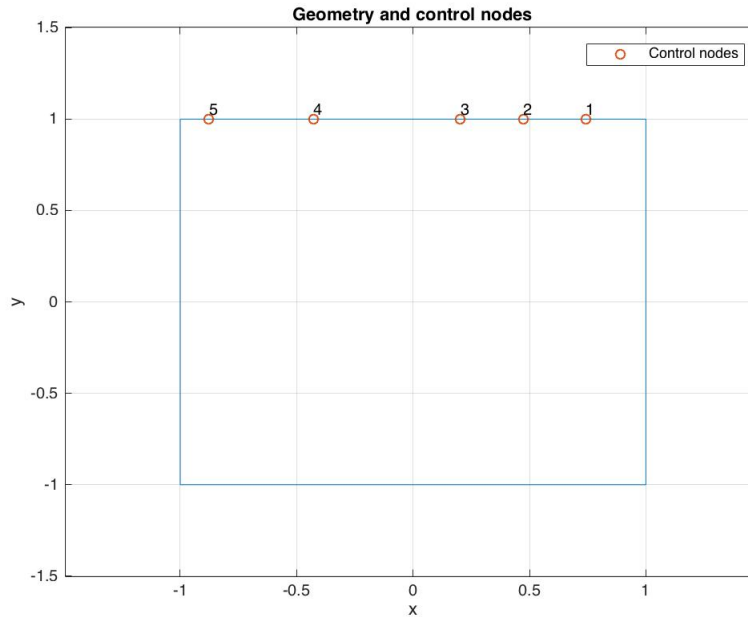


Figure 5.7: Control nodes for convergence study

Table 5.6: Observed order of accuracy, GCI and approximated true solution for 5 nodes on the surface on the cylinder

| Node | p | GCI_{32} | GCI_{21} | F |
|------|-------|------------|------------|--------|
| 1.0 | 1.143 | 0.0006817 | 0.0003085 | 1.0004 |
| 2.0 | 1.168 | 0.000778 | 0.0003461 | 1.0004 |
| 3.0 | 1.185 | 0.0008407 | 0.0003695 | 1.0005 |
| 4.0 | 1.162 | 0.0007539 | 0.0003369 | 1.0004 |
| 5.0 | 1.146 | 0.001267 | 0.000438 | 1.0005 |

From table 5.11, we see that the error bands are satisfactory small for the control nodes, and that the largest value of GCI is approximately 0.04%, for the finest grid. All values of F are close to unity, ensuring that the solution is in fact within the asymptotic range of convergence. In figure 5.8, is the convergence plot for control node 1. Included in the figure is a straight line, which represents Φ_{h0} for this node.

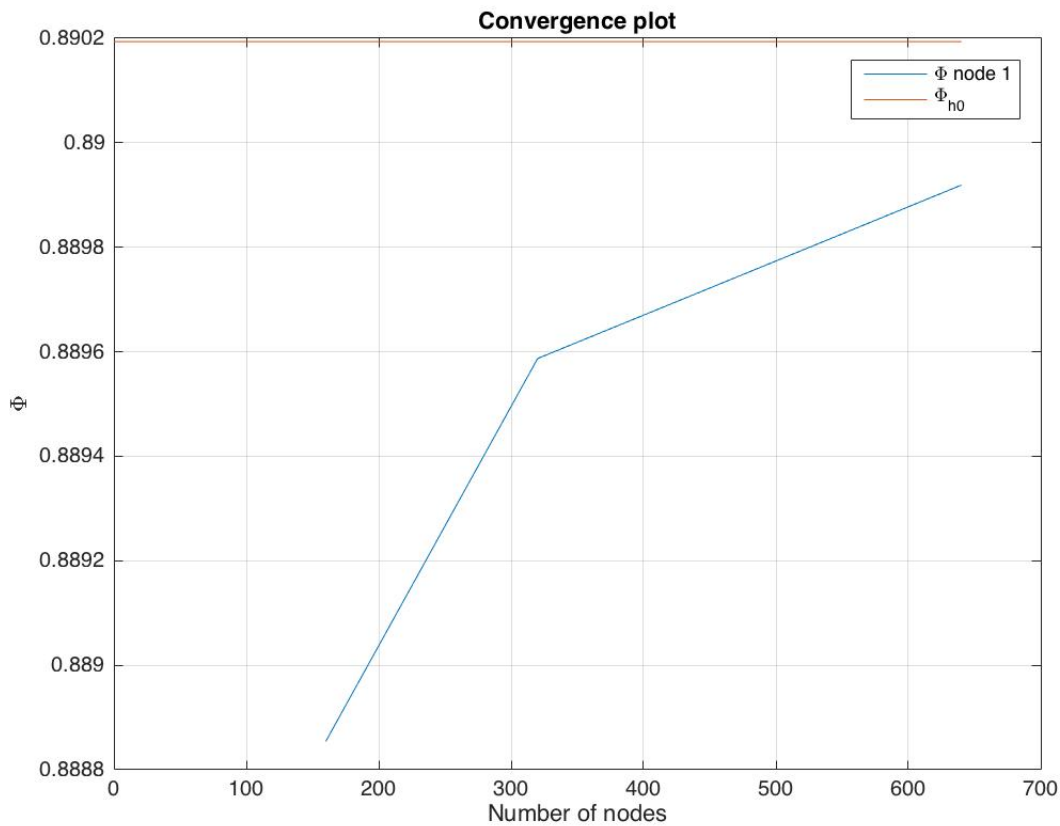


Figure 5.8: Convergence plot for control node 1, with approximate true solution

Even though it is found that $a=0.07$ and $N_B = 2$ gives the least STD, a solution with $N_B = 4$ is chosen. The local solution with the smallest STD is then one with $a=0.09$. The reason for this, will soon be justified, but has to do with the fact that more than one constant is needed in the derivative of the local solution, to match velocities that are not equal on both sides of a corner.

5.3.2 Results

Using the findings from the convergence and sensitivity study, the results from the analysis are presented here. All analysis are conducted using 640 elements, $a=0.09$, $N_{out} = 0$ and $N_B = 4$. Several of the plots are plotted with "Node number" as x-label. The numbering of the elements are as indicated in figure 5.9 where node 1 is marked, going counter-clock wise. The end of the global domain, and start of the local domain is also marked in the figure. It is important to note

that even though the global results from all the nodes are plotted as a continuous line, the results between node $\frac{N}{4}$ and $\frac{N}{4} + 1$, $\frac{N}{2}$ and $\frac{N}{2} + 1$ and so on, are not as described in the plot, but as described by the local potential.

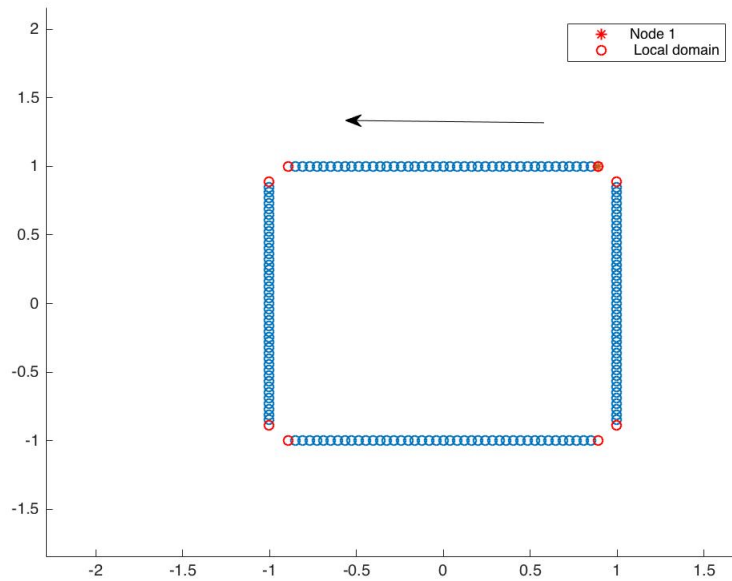


Figure 5.9: Node numbering, direction of integration and intersection between local and global domain

In figure 5.10 is a plot of the global potential. By global potential it is meant the potential over the body, not including the corner potential.

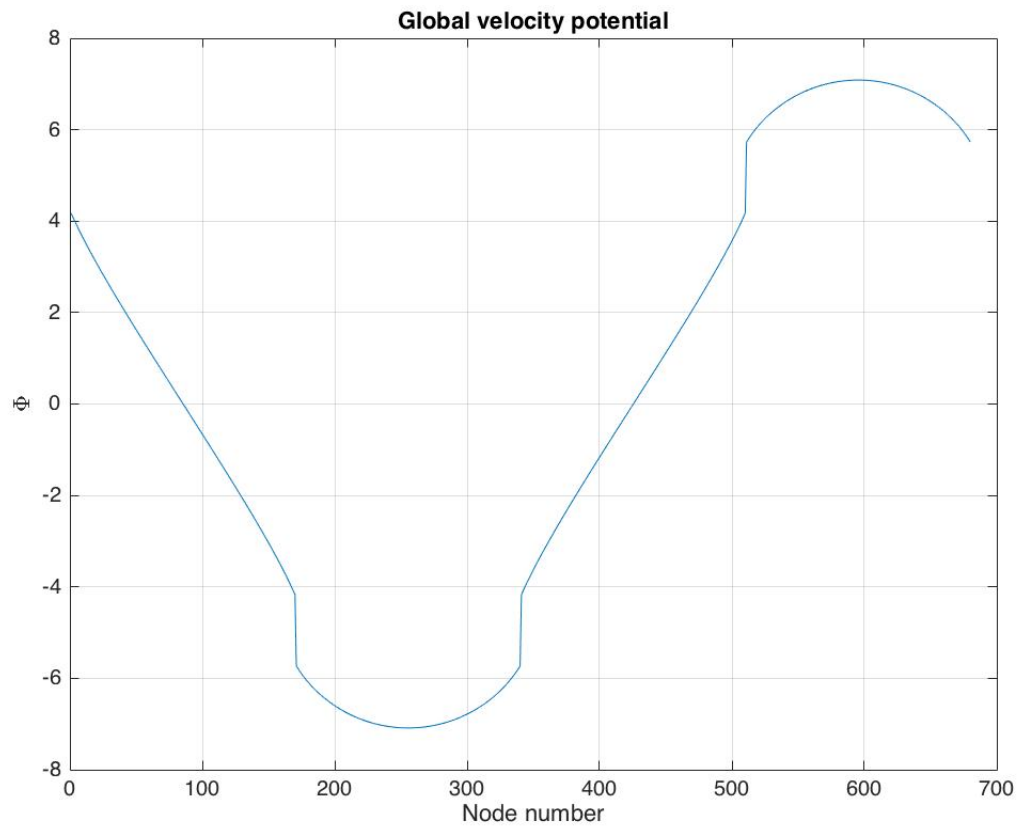


Figure 5.10: Global velocity potential, $N_B = 4$, $a = 0.09$

The local potential for the four corners in the square is presented in figure 5.11, here corner 1 is the top left corner, counting counter-clock wise.

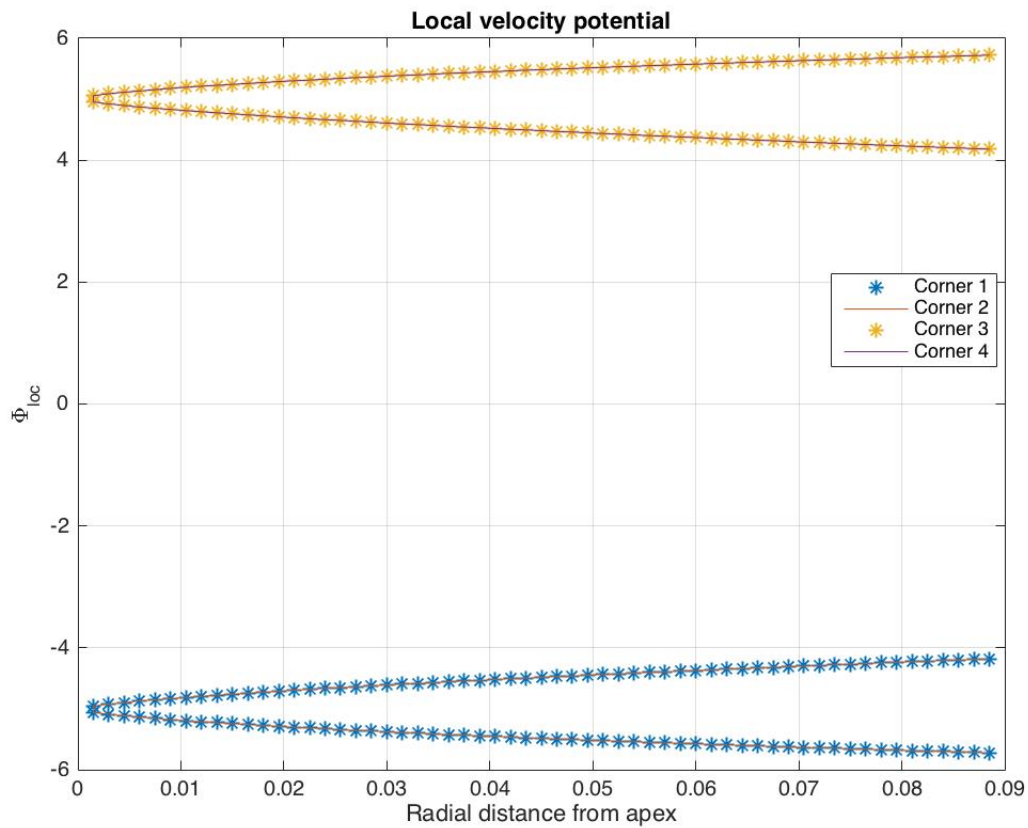
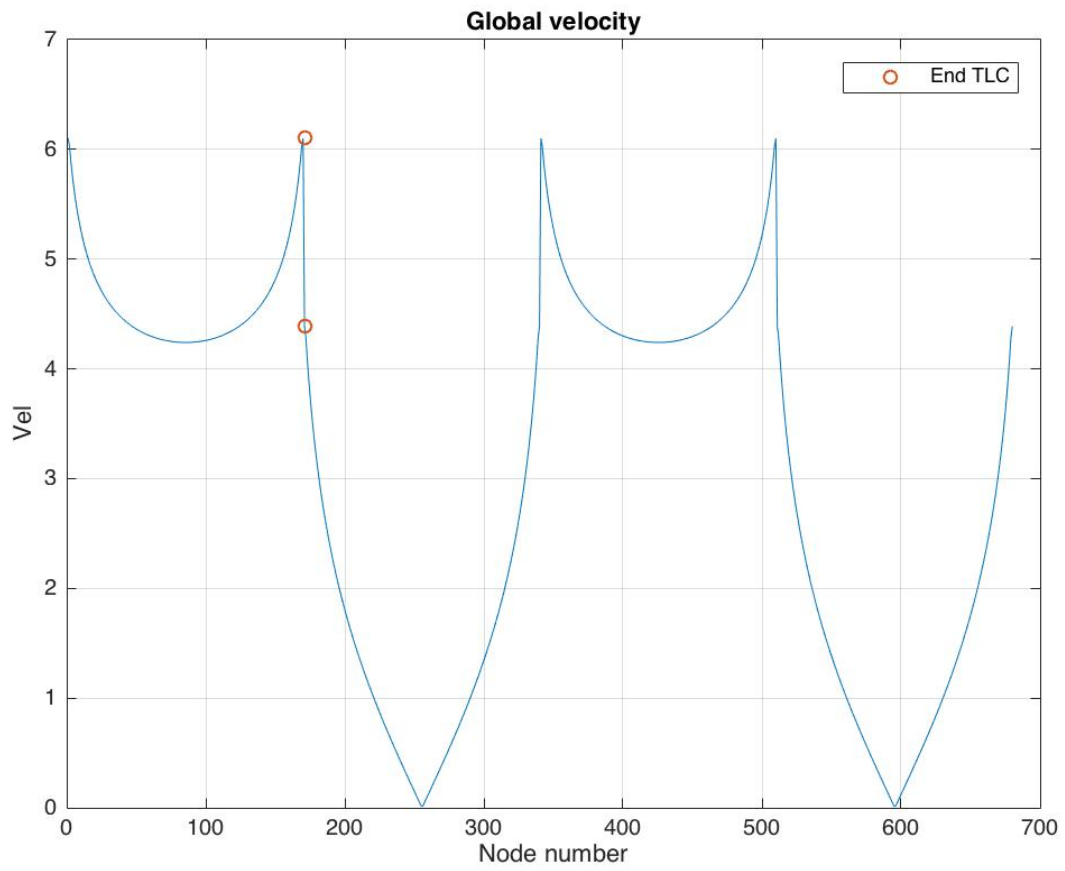
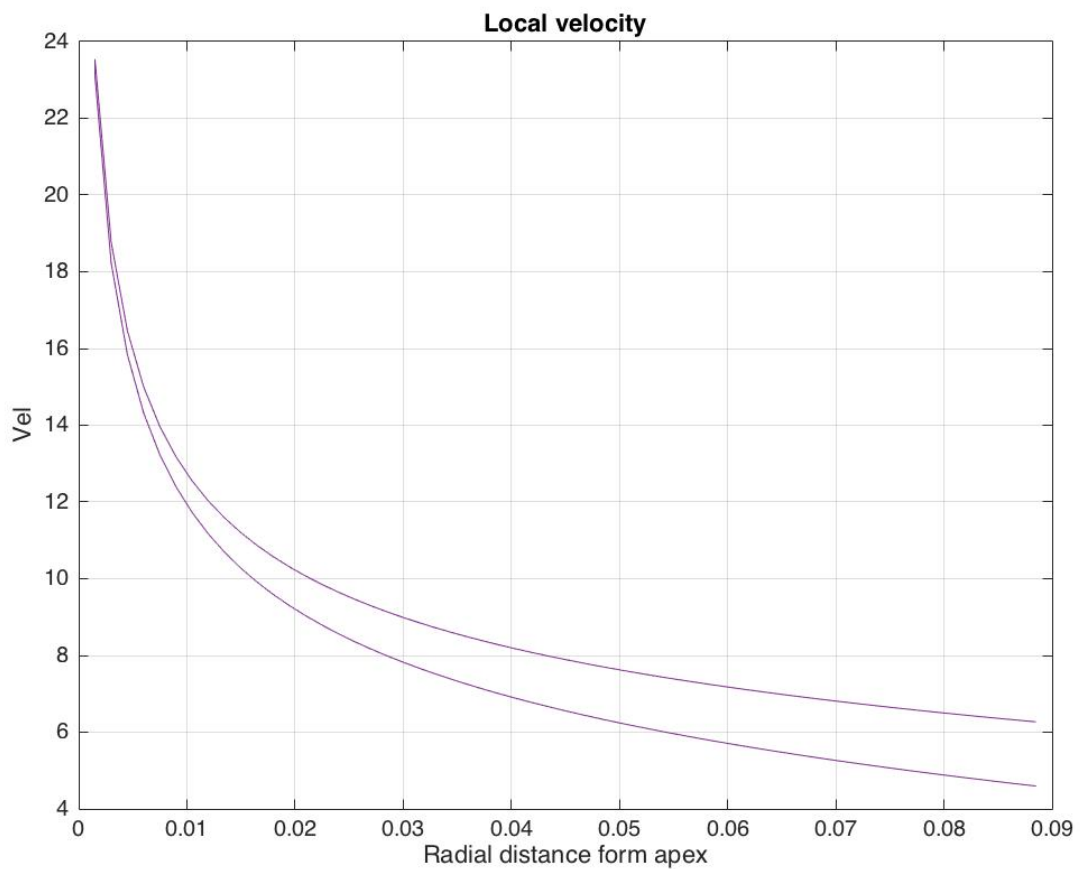


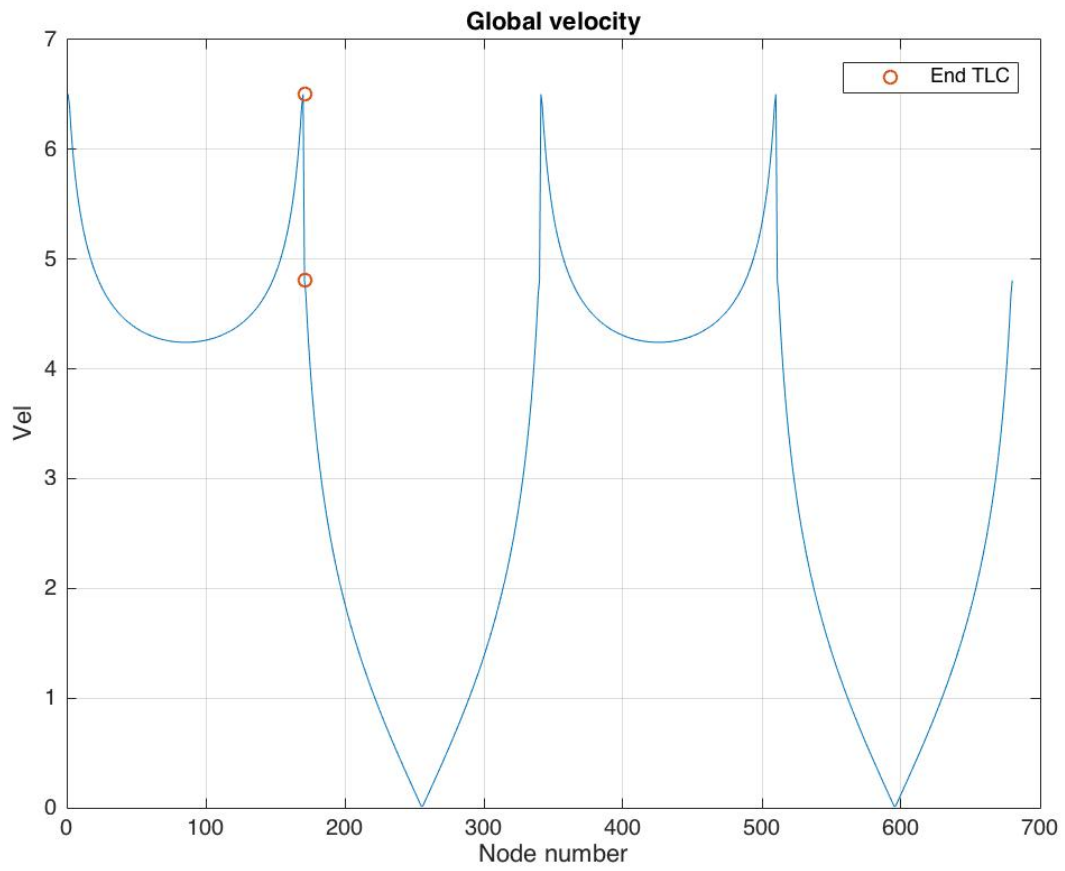
Figure 5.11: Local velocity potential, $N_B = 4$, $a = 0.09$

To find the velocity, central difference as defined in appendix D is used for global velocity. The local velocity is found by differentiating equation 2.26 analytically. For comparison, the global and local velocity for both the case of $N_B = 4$ is given in figure 5.12 and for $N_B = 2$ in figure 5.13. In the global plots, the end of the global domain in the top left corner, are marked for comparison of the two solutions.

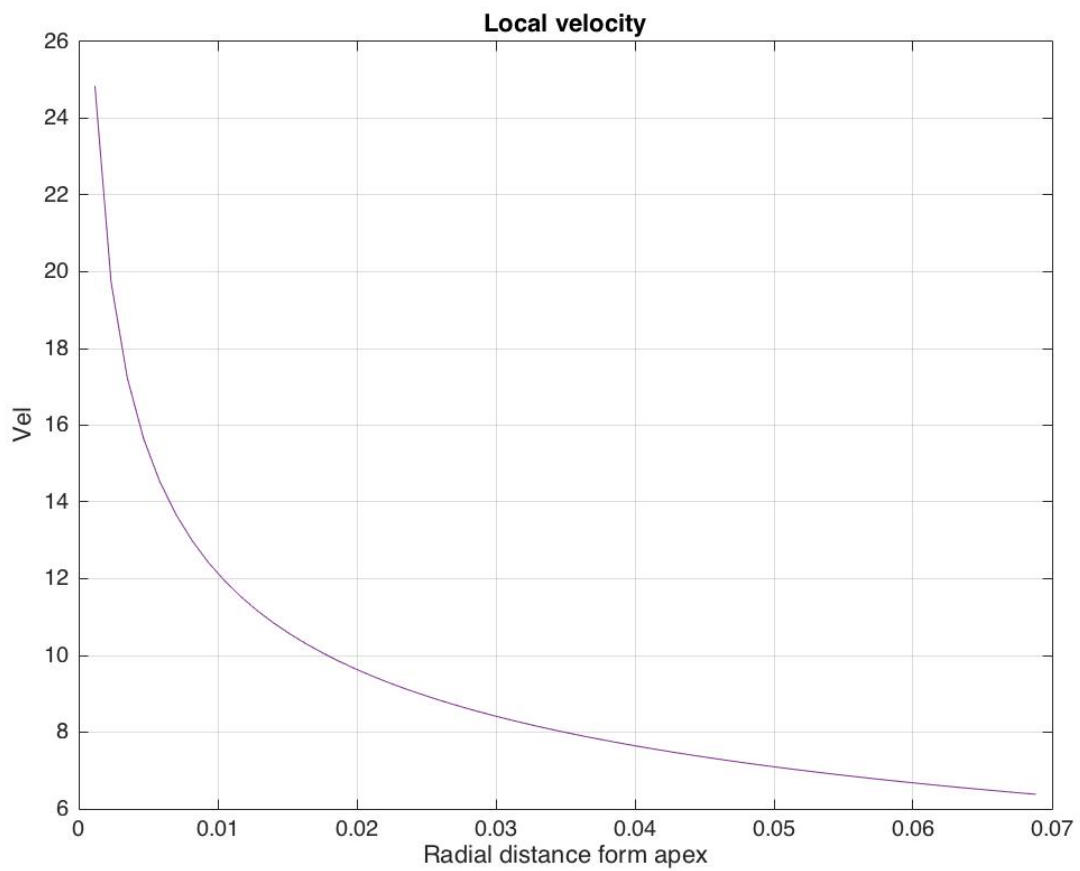


(a) Global velocity, $N_B = 4$, $a = 0.09$





(a) Global velocity, $N_B = 2$, $a = 0.07$



As can be seen from the figures, using 4 constants in the local potential, gives a local velocity that much better matches the global velocities, than using 2 constants. The numerical values at the intersection between the domains are given in table 5.7.

Table 5.7: Global and local velocity in intersection between local and global domain for top left corner

| N_B | Side 1 | | Side 2 | |
|-------|------------|-----------|------------|-----------|
| | V_{glob} | V_{loc} | V_{glob} | V_{loc} |
| 2 | 6.5027 | 6.3811 | 4.8052 | 6.3811 |
| 4 | 6.1012 | 6.2695 | 4.3863 | 4.5966 |

A solution for the full domain is produced by utilizing equation 2.11 for a uniform grid in the outer domain. In figure 5.14 and 5.15 are the velocity potential and velocity respectively. For the velocities, a coarser resolution is used than in the velocity potential. The velocities close to the corners succeed the maximum of 6 displayed by the color bar, this is used as a limit to also be able to visualize the lower velocities.

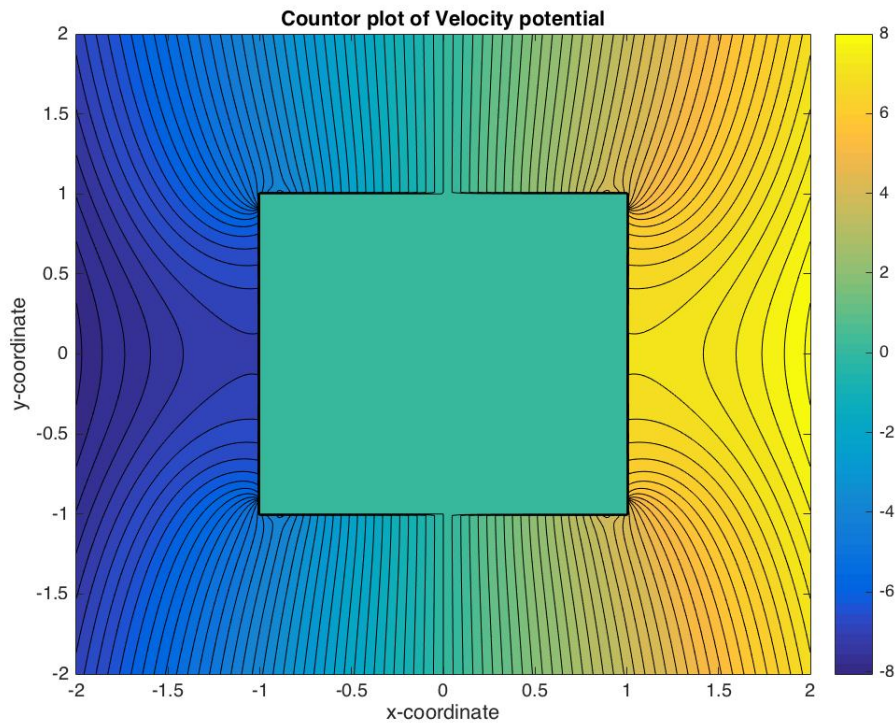


Figure 5.14: Velocity potential in surrounding fluid

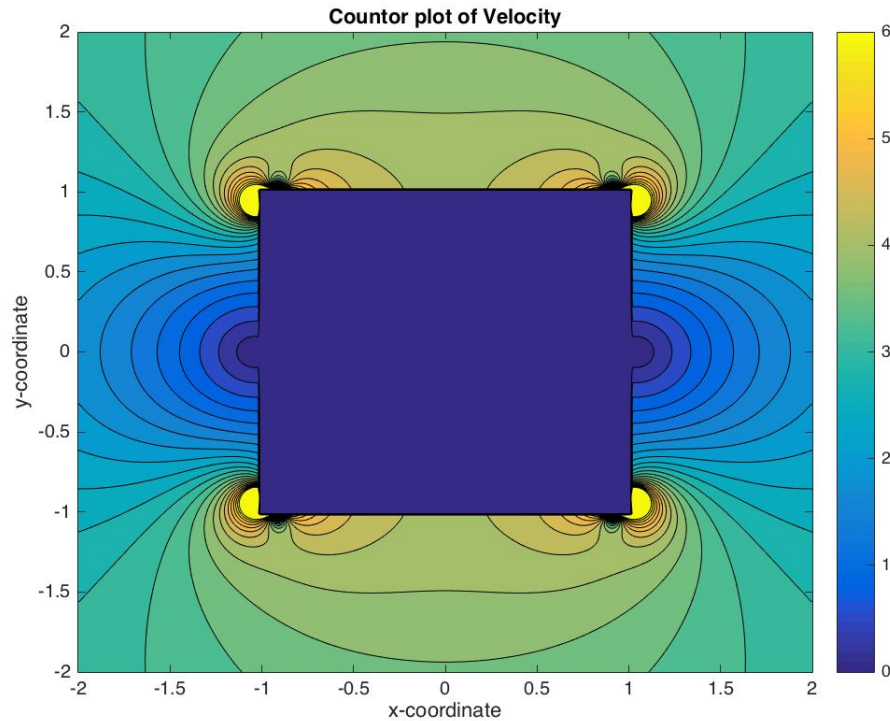


Figure 5.15: Velocity of surrounding fluid

For comparison with earlier results, the added mass for the square cylinder oscillating with infinite frequency, is calculated. The results are obtained using $a=0.09$ $N_B = 4$ and $N_{out} = 0$, The resulting added mass is given in table 5.8, along with the Lewis form value and the calculated value for the square without a local solution. The deviation is between the this model, and the Lewis form value.

Table 5.8: Added mass of oscillating square cylinder in infinite fluid

| A_{Lewis} | A_{num} | $A_{r=0}$ | Deviation |
|-------------|-----------|-----------|-----------|
| 1.51 | 1.4788 | 1.5185 | 0.0207 |

5.4 Square, Least square matching

For comparison and validation, the same BVP, i.e the flow around a 2D square cylinder, where the corner flow is described by the corner potential is analyzed again. The only difference is that

here, least square method is used in the continuity equations.

5.4.1 Sensitivity and convergence

As in the direct matching case, the size of the local domain, the number of constants, and the location of the matching nodes have to be set a priori. In the case of least square matching, the number of constants and matching nodes does not have to be the same. Hence, the effect of using fewer constants than matching nodes is considered. A sensitivity study is firstly carried out, to find the best fit solution. All analysis where the local domain size is not varied, are conducted using 640 elements.

Firstly, the effect of different combinations of N_B , N_{out} and N_C are investigated. Here, a is kept constant at 0.1 and the results are assumed to be valid for all a . All the numerical details are not presented here, but it is found that the best combination is one where $N_C = 2$. Results for the different combinations of N_B and N_{out} for this case are given in table 5.9.

Table 5.9: Standard deviation for different combinations of N_B and N_{out} , using $N_C = 2$

| N_{out} | $N_B = 2$ | $N_B = 4$ | $N_B = 6$ |
|-----------|-----------|-----------|-----------|
| 0 | 0.2207 | 0.2232 | 0.2282 |
| 2.0 | 5.34 | 3.488 | 2.614 |
| 4.0 | 7.27 | 5.393 | 4.313 |
| 8.0 | 8.963 | 7.442 | 6.389 |
| 16.0 | 10.15 | 9.146 | 8.354 |
| 32.0 | 10.87 | 10.29 | 9.805 |

From table 5.9, it is clear that the solution which has the least STD, is one with $N_{out} = 0$. Further it is observed that the lowest STD is obtained by $N_B = 2$. Next the effect of different combination of a and N_B and N_C are tested. a is varied from 0.01 to 0.1. Seeking to keep the grid density constant, h is set to the value it has for $a=0.1$, $N=640$, which leads to more elements being used in for smaller a . N_B is equal to 2,4 or 6, and N_{out} is 2,4 or 6. In figure 5.16 are the obtained STD for all different combinations. The minimum values are marked in the figure, and given in table 5.10.

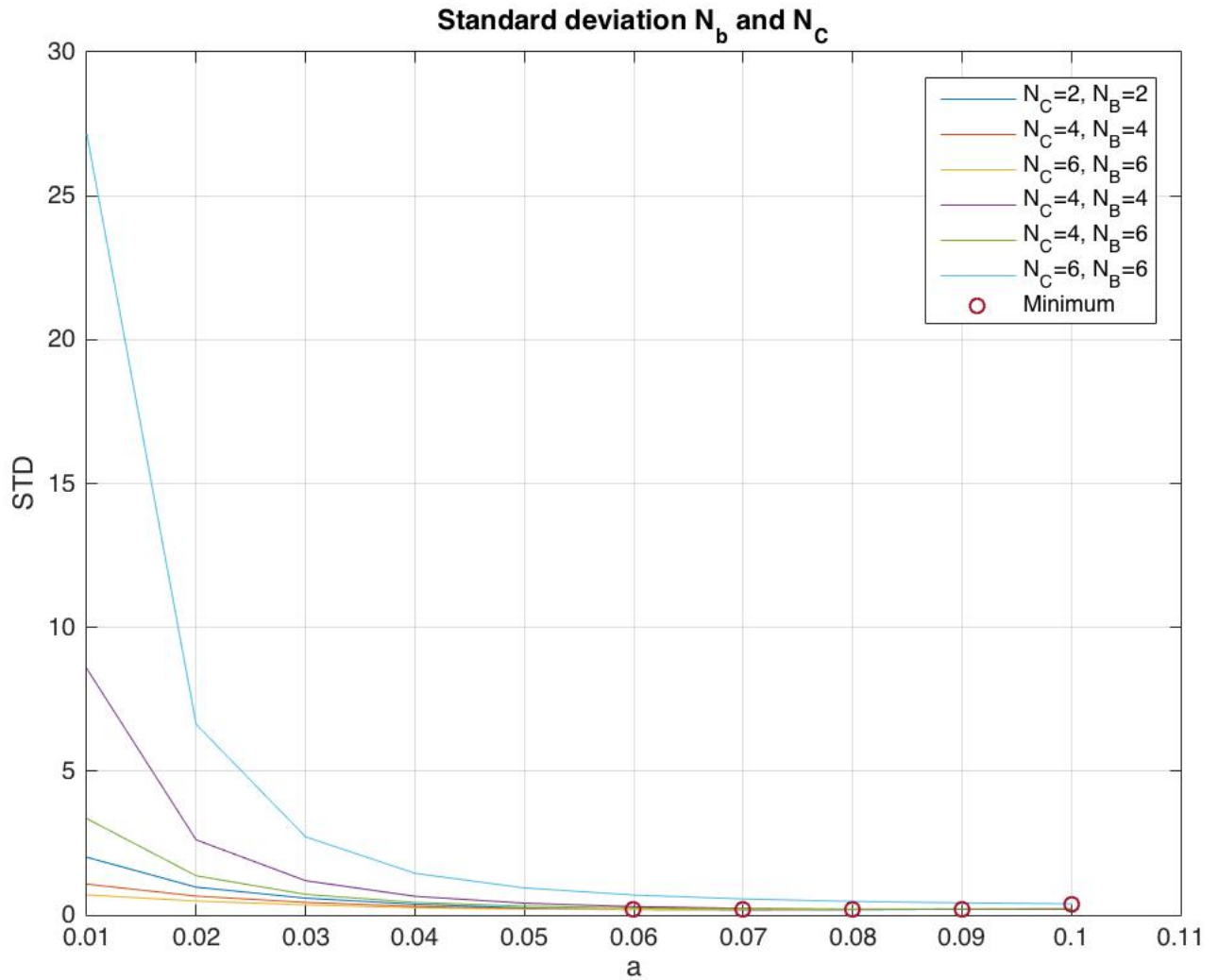


Figure 5.16: Standard deviation for all combination of N_B and N_C

Table 5.10: Minimum standard deviation for different combinations of N_C , N_B and a .

| | $N_C = 2$ | | | $N_C = 4$ | | $N_C = 6$ |
|---------|---------------|-----------|-----------|-----------|-----------|-----------|
| | $N_B = 2$ | $N_B = 4$ | $N_B = 6$ | $N_B = 4$ | $N_B = 6$ | $N_B = 6$ |
| a | 0.07 | 0.07 | 0.06 | 0.09 | 0.08 | 0.1 |
| min std | 0.1758 | 0.1844 | 0.1915 | 0.1972 | 0.2087 | 0.3901 |

From table 5.10 it is observed that the minimum STD is 0.1758, which is the result from using $N_B = 2$ and $N_C = 2$. In figure 5.17 is the local potential plotted at the 10 global nodes closest to the top left corner for this case.

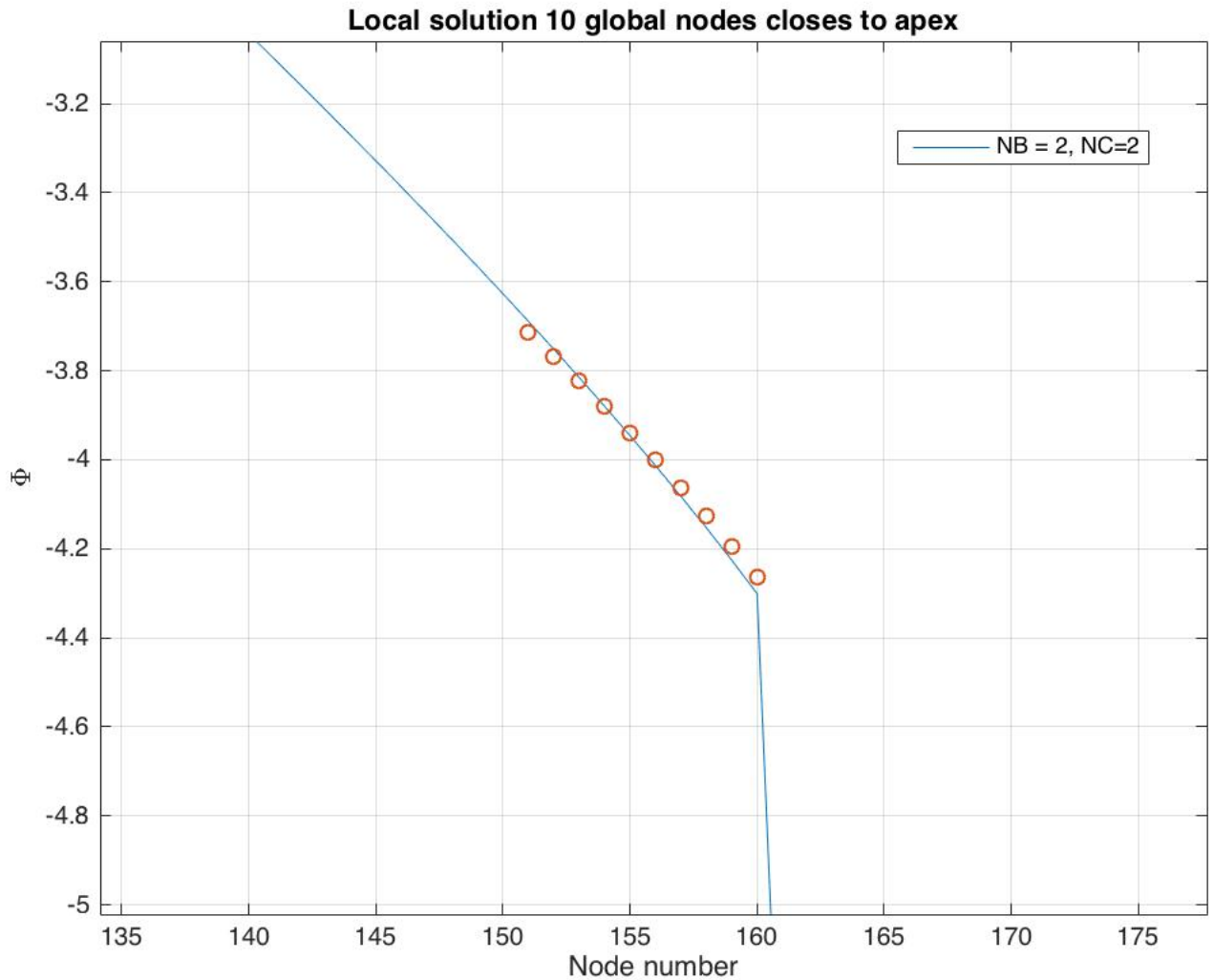


Figure 5.17: Local solution at 10 global nodes closest to the local domain of the top left corner, using $N_B = 2$, $N_{out} = 0$, $N_C = 2$ $a=0.07$.

Before presenting any results, it must be ensured that the solution is within the asymptotic range of convergence. For the same control nodes given in figure 5.7, p, GCI and F are calculated using 3 different grids, all refined by a factor of 2. The coarsest having $N=160$ elements, and the finest $N = 160 \cdot 2^2 = 640$ elements. $a = 0.07$, $N_B = 2$, $N_{out} = 0$ and $N_C = 2$.

Table 5.11: Observed order of accuracy, GCI and fraction for 5 nodes on the surface on the cylinder

| Node | p | GCI_{32} | GCI_{21} | F |
|------|-------|------------|------------|--------|
| 1.0 | 1.143 | 0.0008521 | 0.0003857 | 1.0004 |
| 2.0 | 1.168 | 0.0009725 | 0.0004326 | 1.0004 |
| 3.0 | 1.185 | 0.001051 | 0.0004619 | 1.0005 |
| 4.0 | 1.162 | 0.0009424 | 0.0004211 | 1.0004 |
| 5.0 | 1.137 | 0.001583 | 0.0005476 | 1.0008 |

From table 5.11, it is taken that the error-band is satisfactory small, with the largest value being 0.055%, for the finest grid. Further the values of the F, ensures us that the solution is indeed within the asymptotic range of convergence. In figure 5.18 is the convergence plot for the node 1, with Φ_{h0} included.

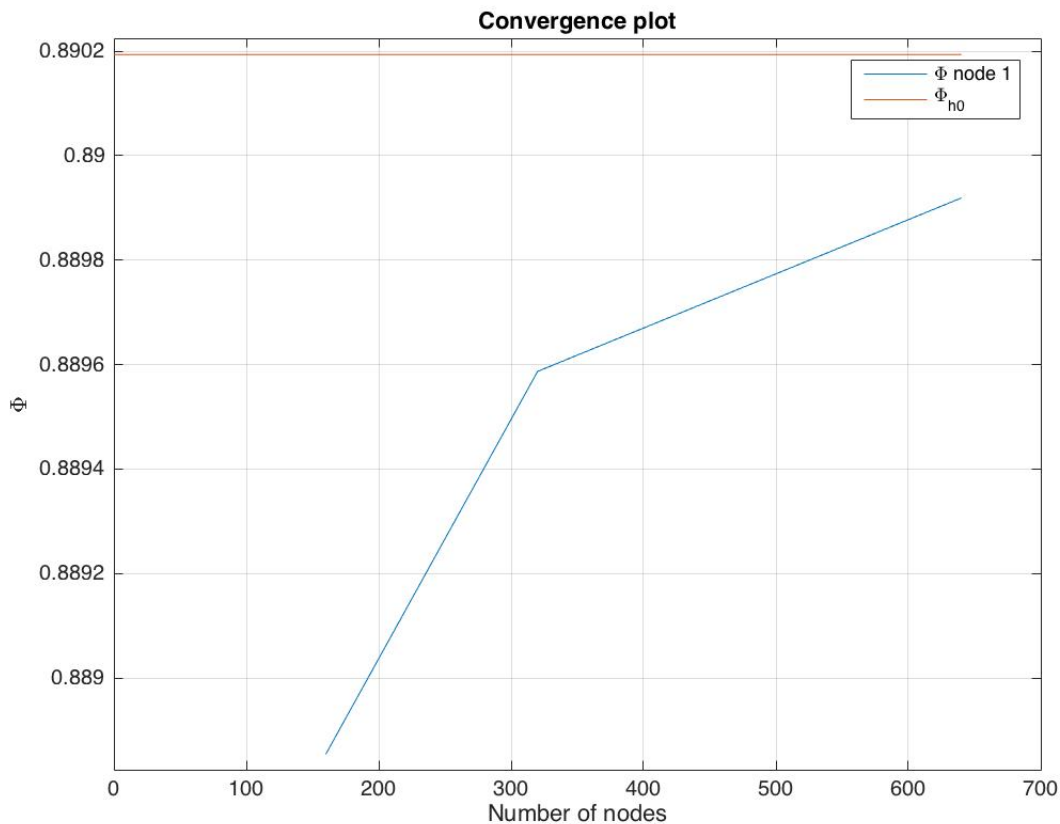


Figure 5.18: Convergence plot for control node 1, including approximated exact solution

The best match is found be one where only two constant are used in the local solution. However, also in this case the velocity on the two sides of a corner are not the same, and to better match the different velocities, more than one constant is needed in the local velocity. Therefor the results for the solution with smallest STD using 4 constants are presented in the following. The parameters used in the local solution is therefor: $a=0.09$, $N_B = 4$, $N_C = 4$ and $N_{out} = 0$.

5.4.2 Results

Using the results form the convergence and sensitivity study, the resulting velocity potentials and velocities, on the boundary of the body and in the domain is presented. The global velocity potentials and velocities are plotted against node numbers, which are numbered the same way as in figure 5.9. All results presented are obtained by using 640 elements on the body surface.

Given in figure 5.19 is the global potential over the surface of the cylinder.

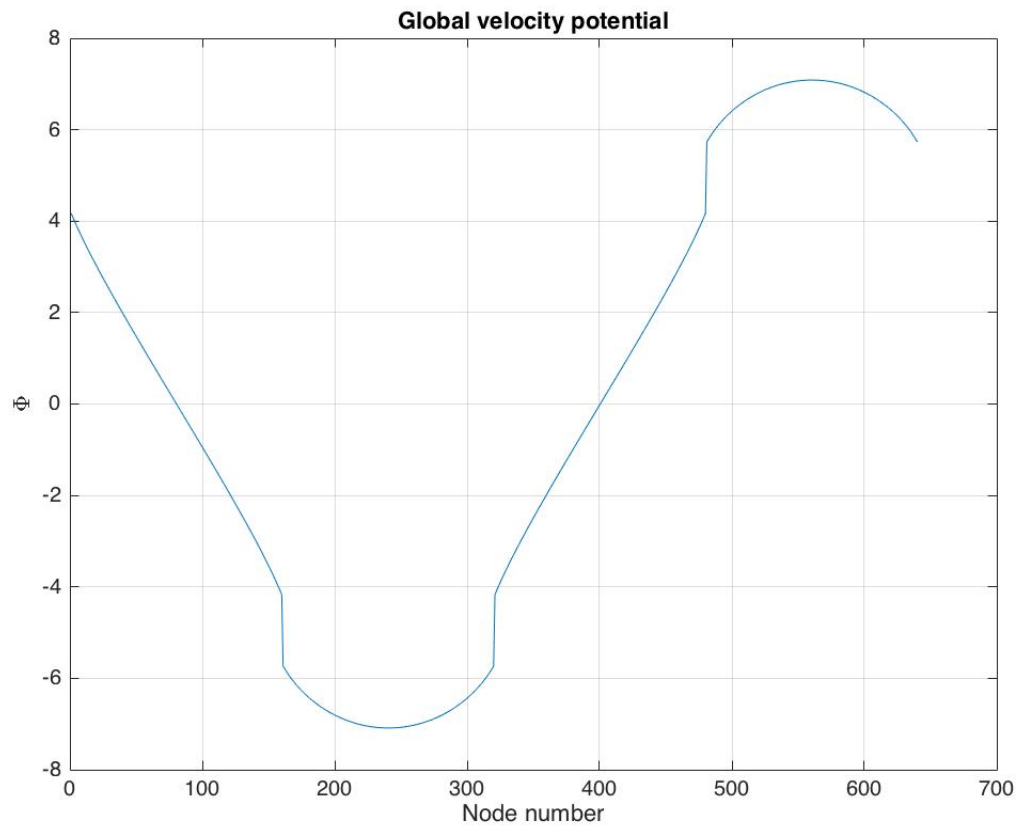


Figure 5.19: Global velocity potential, $N_B = N_C = 4$, $a = 0.09$

The local potential for the four corners in the square are given in figure 5.11. Corner 1 is the top left corner, TLC, counting counter-clock wise.

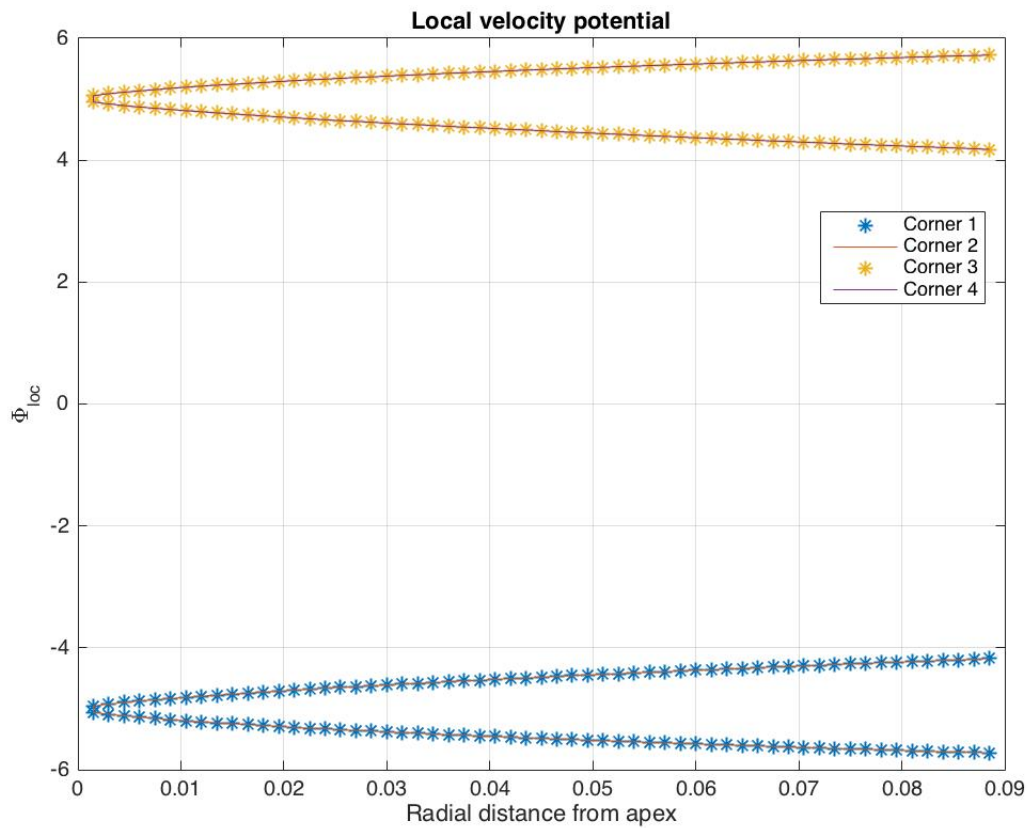
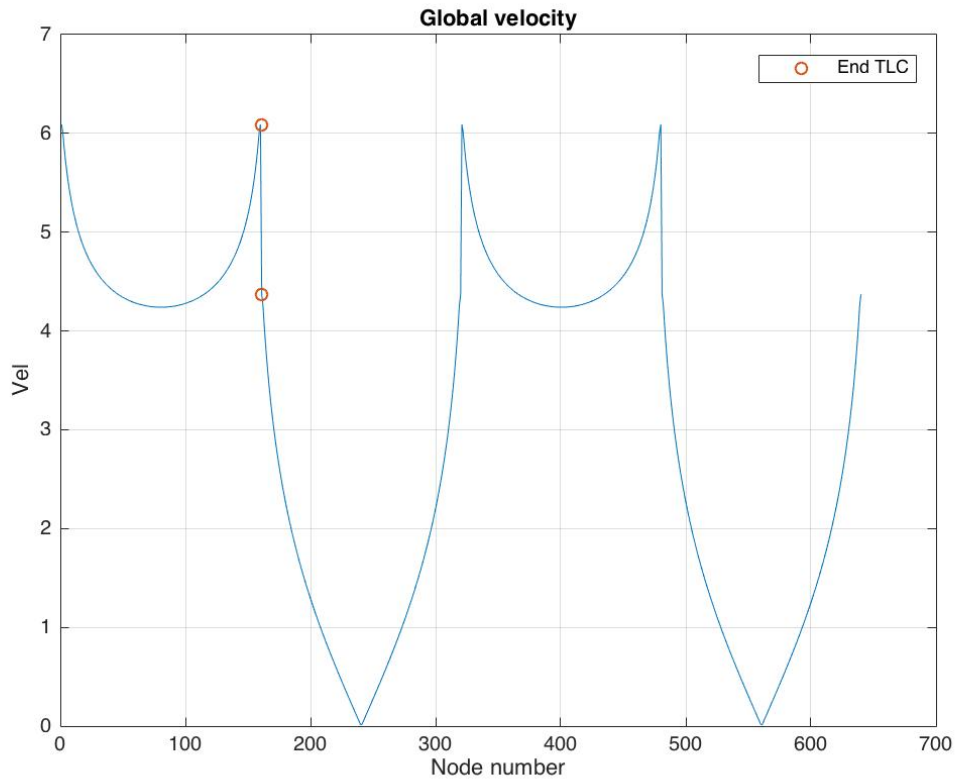
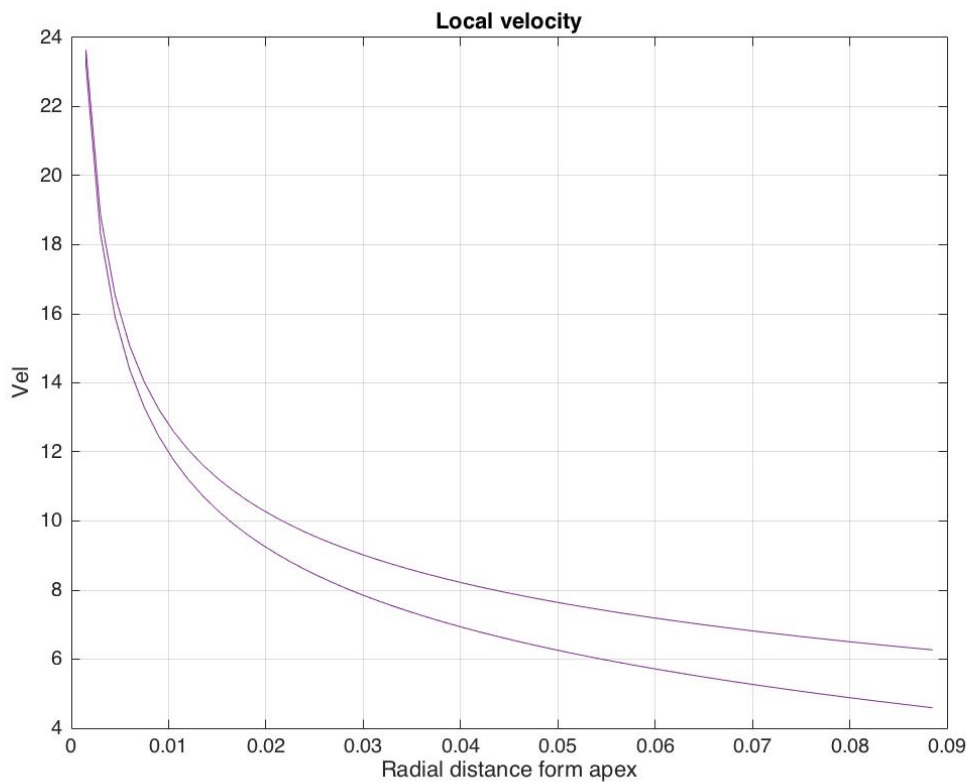


Figure 5.20: Local velocity potential, $N_B = N_C = 4$, $a = 0.09$

For comparison, the global and local velocity for both the case of $N_B = N_C = 4$ are presented in figure 5.21 and for $N_B = N_C = 2$ in 5.22. In the global plots the end of the global domain in the TLC are marked, for comparison of the two solutions.

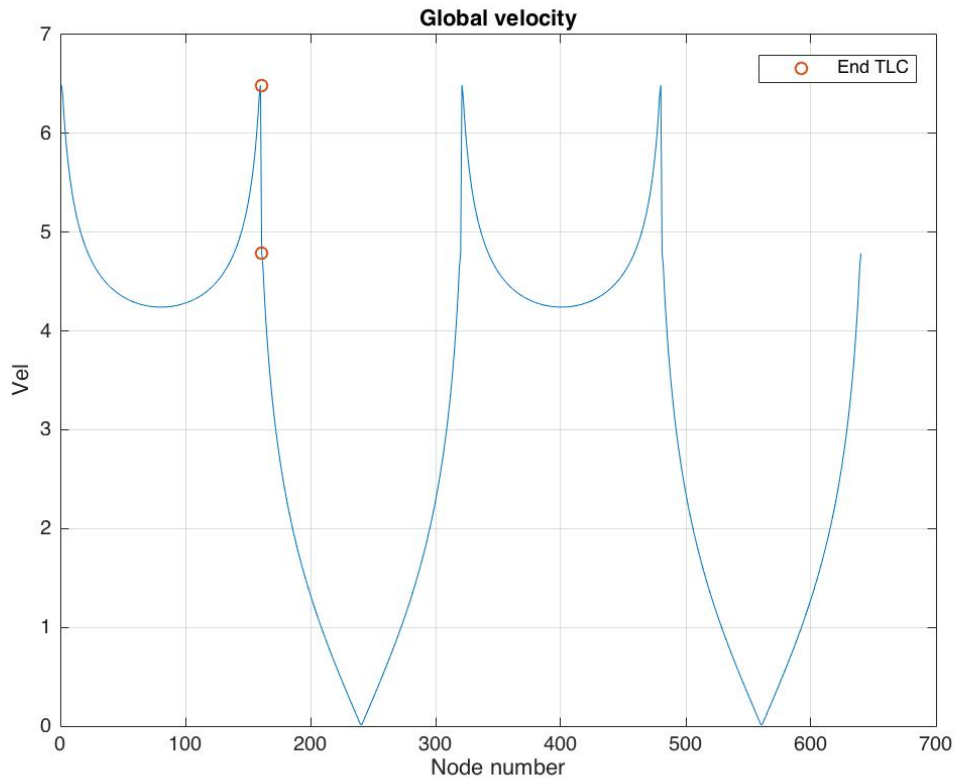


(a) Global velocity, $N_B = N_C = 4$, $a = 0.09$

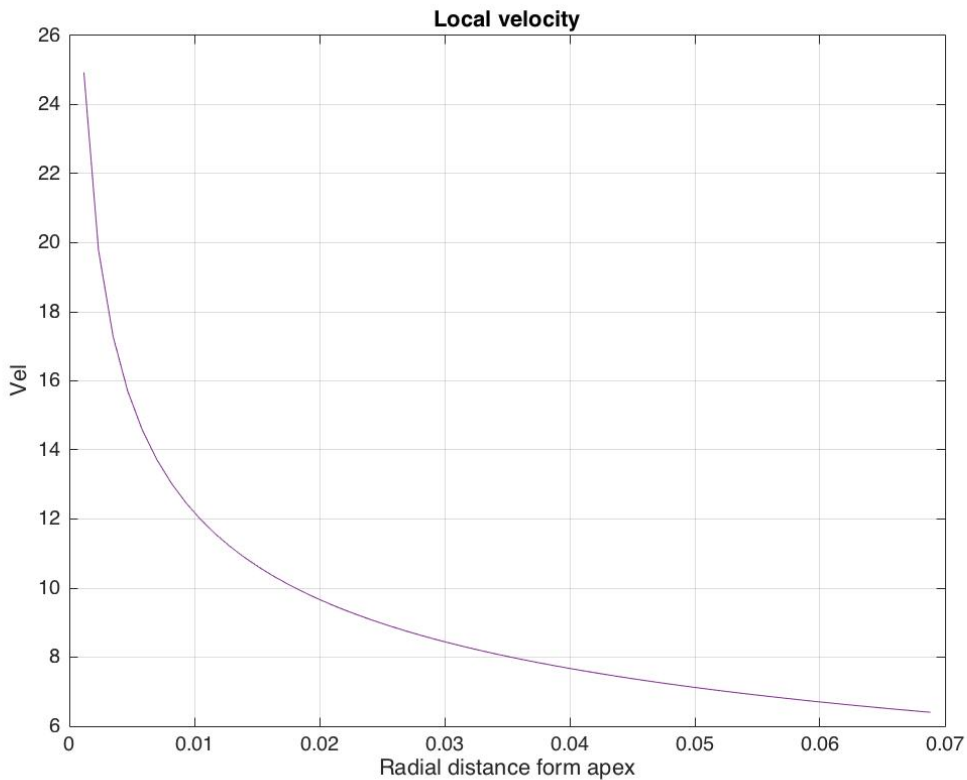


(b) Local velocity, $N_B = N_C = 4$, $a = 0.09$

Figure 5.21: Global and local velocity for $N_B = N_C = 4$



(a) Global velocity, $N_B = 2 = N_C = 2$, $a = 0.07$



(b) Local velocity, $N_B = N_C = 2$, $a = 0.07$

Figure 5.22: Local and global velocity for $N_B = N_C = 2$

As can be seen from the figures, the local velocity calculated using 4 constants, much better matches the the global velocity, than for a solution using only 2 constants. The numerical values at the intersection between the domains are presented in table 5.12.

Table 5.12: Global and local velocity in intersection between local and global domain for top left corner

| N_B | Side 1 | | Side 2 | |
|-------|------------|-----------|------------|-----------|
| | V_{glob} | V_{loc} | V_{glob} | V_{loc} |
| 2 | 6.4870 | 6.4037 | 4.7855 | 6.4037 |
| 4 | 6.0898 | 6.2689 | 4.3706 | 4.5936 |

A solution for the full domain is produced by utilizing equation 2.11 for a uniform grid in the outer domain. In figure 5.23 and 5.24 are the velocity potential and velocity respectively. A coarser resolution is used than in the velocity, to better capture the details at lower velocities.

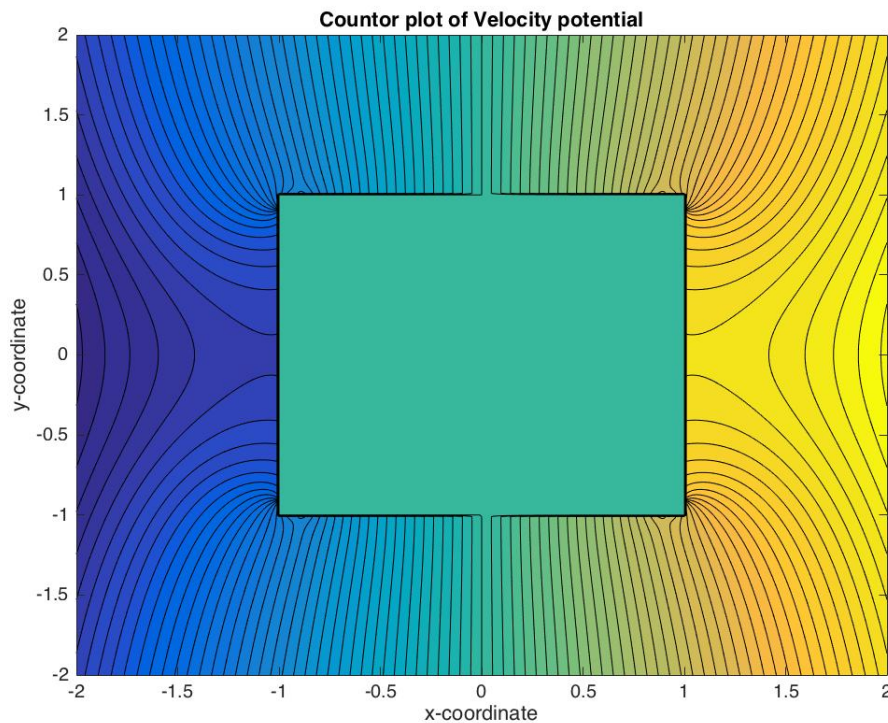


Figure 5.23: Velocity potential in surrounding in fluid

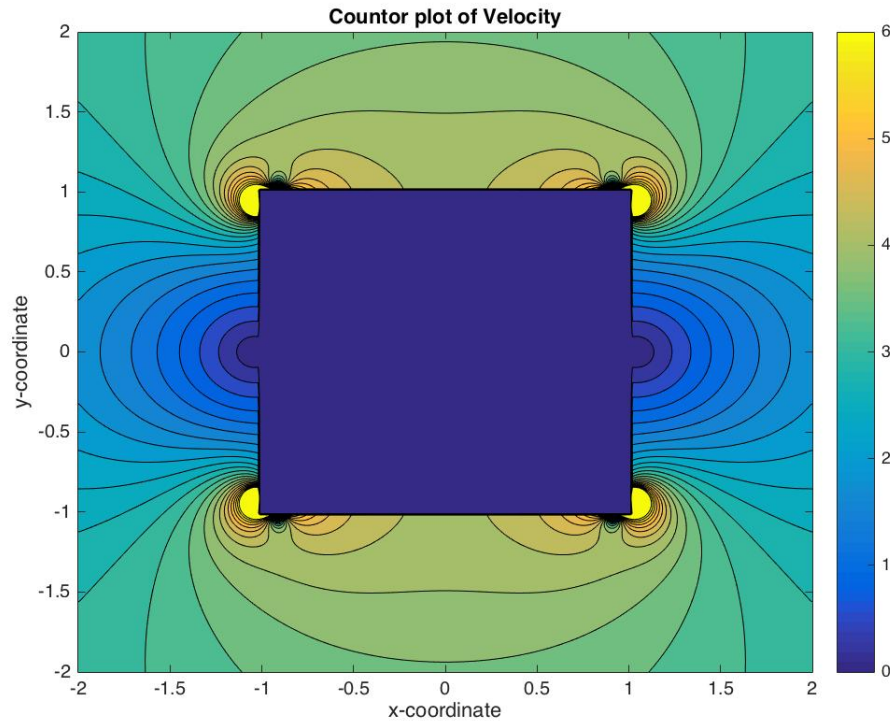


Figure 5.24: Velocity of surrounding fluid

For comparison, the added mass for the square cylinder, using least square matching is calculated. The resulting added mass in surge, using the the results from the convergence and sensitivity study is given in the table 5.13

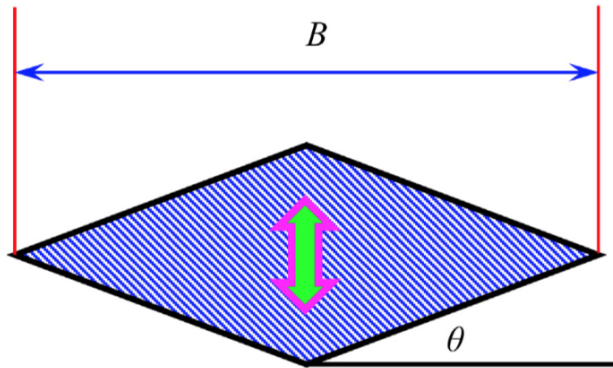
Table 5.13: Added mass of oscillating square cylinder in infinite fluid, analyzed with $N_B = N_C = 4$ using least square matching

| A_{Lewis} | A_{num} | $A_{r=0}$ | Deviation |
|-------------|-----------|-----------|-----------|
| 1.51 | 1.4790 | 1.5185 | 0.0205 |

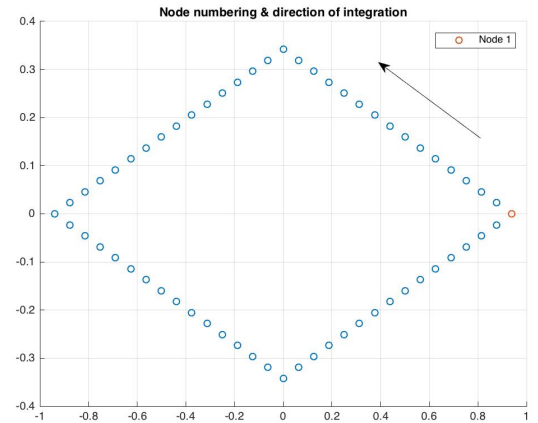
5.5 Oscillating double

The last problem analyzed in infinite fluid, is the case of a wedge, oscillating in heave with infinite frequency. The geometry is defined in figure 5.25 a). The added mass in heave is in this case is analytical known, and given in Faltinsen (2005) as equation 5.3. Because of this, no convergence study is conducted, and the results are merely verified by comparison to the analytical

solution. In cases where the velocity potential over the body surface is presented, it is plotted against node number. The numbering of the nodes and direction of integration is as given in figure 5.25 b)



(a) Double Wedge



(b)

Figure 5.25: Geometry of wedge Liang et al. (2015), and node numbering and direction of integration

$$A_{33} = \frac{2\rho(0.5B \tan(\theta))}{\tan(\theta)} \left[\frac{\pi}{\sin(\theta)} \frac{\Gamma(1.5 - \theta/\pi)}{\Gamma^2(1.0 - \theta/\pi) \cdot \Gamma(0.5 + \theta/\pi)} - 1 \right] \quad (5.3)$$

In figure 5.26 are the resulting added mass in heave for different dead rise angles, θ . The results are normalized by $\rho\pi B^2/8$, and plotted with the normalized analytical solution for the same θ . The results are computed using 160 elements on each side of the wedge, 640 in total. Numerical data on the largest deviation between the analytical and numerical solution is given in table 5.14

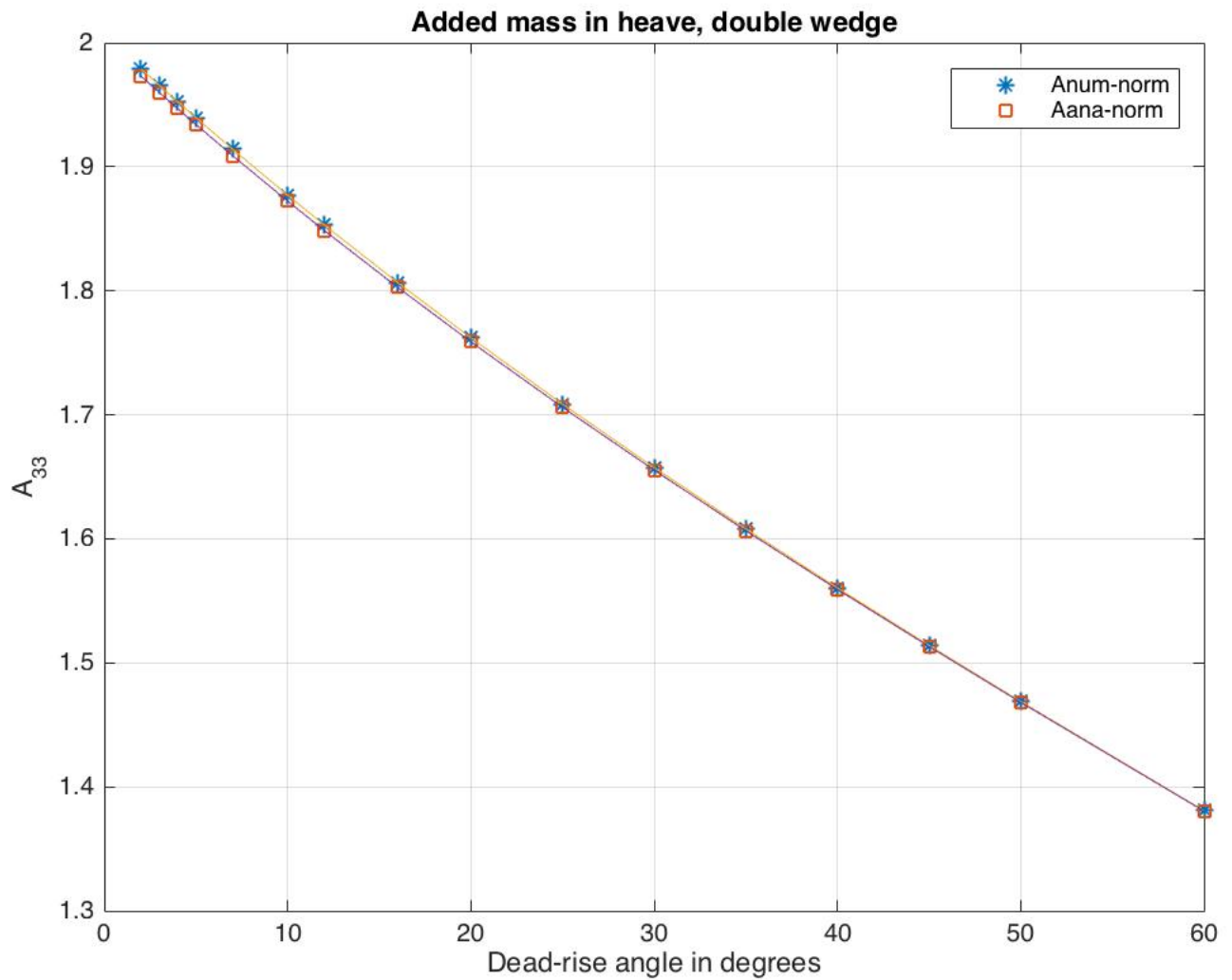


Figure 5.26: Added mass in heave for double wedge with 20 ° dead rise angle, oscillating in infinite fluid

Table 5.14: Largest deviation between analytical and numerical added mass in heave for double wedge

| Dead rise angle | $A_{ana} - A_{num}$ | $\frac{A_{ana} - A_{num}}{A_{ana}}$ |
|-----------------|---------------------|-------------------------------------|
| 2 ° | 0.006 | 0.003 |

As can be seen from the figure, the results obtained with the BEM solver, agree well with the analytical solution. For comparison with later solutions, the resulting velocity potential for θ 20 ° is given in figure 5.27.

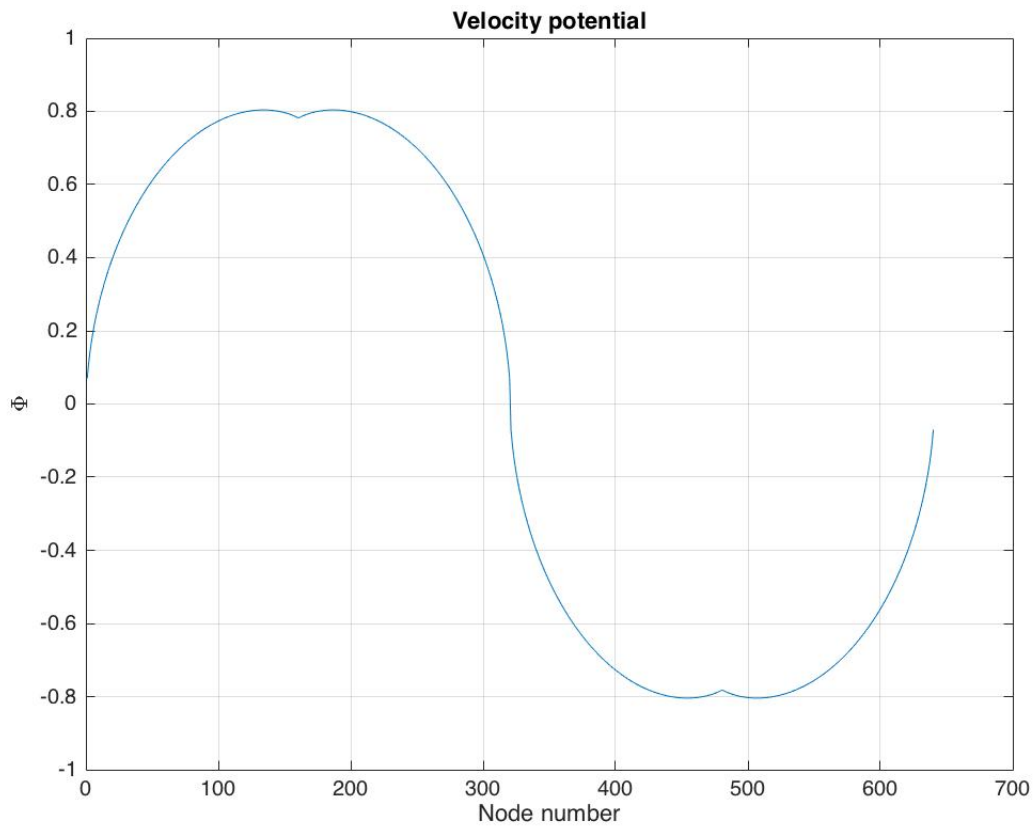


Figure 5.27: Velocity potential for double wedge with 20° dead rise angle, oscillating in infinite fluid

Chapter 6

Free surface problems

In the last part of this thesis, are results from problems where the free surface is present. Both the case of wall sided, and non-wall sided boundary conditions are looked into, and their convergence properties are presented. A local corner model is proposed to deal with convergence issues in the free surface.

6.1 Boundary Value Problem

The BVP for the problems analyzed in this chapter consists of a 2D sectional geometry, oscillating with infinite frequency in the free surface. With the fluid being ideal, the governing equation is that of Laplace. The BVP is illustrated in figure 6.1. Here the free surface has Dirichlet condition and the body is impermeable, yielding a Neumann condition. To compare the results with the results from the previous chapter, where the case was a constant current in x-direction or oscillation in surge, the problems are modeled as illustrated below, with the free surface parallel to the y-axis.

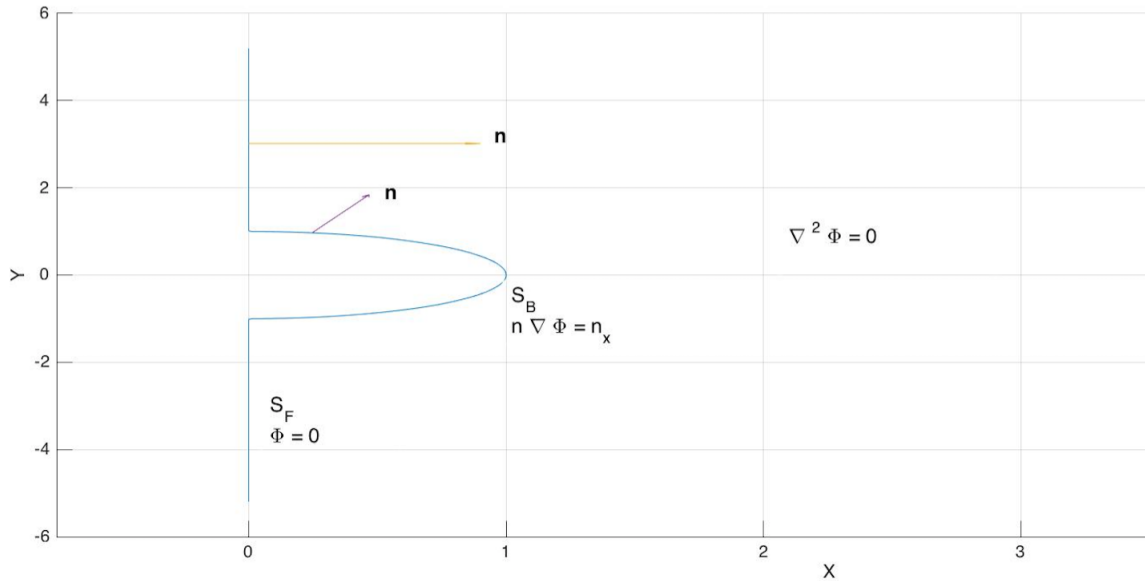


Figure 6.1: Geometrical description of boundary value problem

6.2 Cylinder with draft equal to radius

In this section are results obtained from analyzing a cylinder, oscillating with infinite frequency in the free surface. The radius of the cylinder is $R=1$, and the normal vector is positive into the fluid domain. The analytical solution to this problem is known, and the normalized velocity potential is given as equation 6.1. The added mass is given in equation 6.2 Faltinsen (1990). Because of this, no convergence study of the solution is conducted, and the solution is only verified by comparing with the analytical solution. Both results from the double body problem and using Rankine sources are presented, and compared.

$$\Phi = \frac{R^2}{r} n_x \quad (6.1)$$

$$A_{33} = 0.5\rho\pi R^2 \quad (6.2)$$

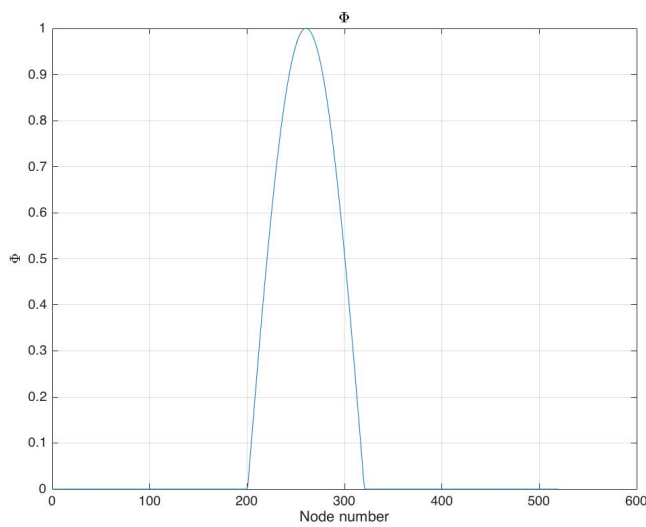
For the method where Rankine sources are used, it is important that the free surface is sufficiently long. To this end, grid stretching is used. In table 6.1 are the element data used to solve

this problem.

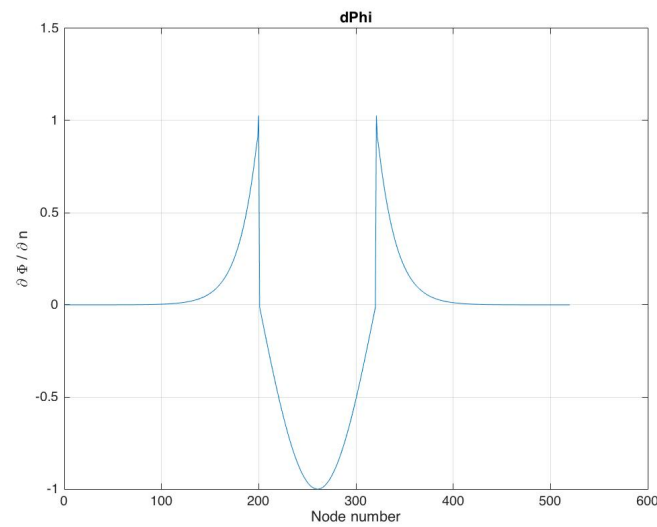
Table 6.1: Element data for problem with cylinder in oscillating in free surface, using free space Green's functions

| No. of Elements | Elements on S_F | Elements on N_{S_B} | Stretching factor | Length F_S |
|-----------------|-------------------|-----------------------|-------------------|--------------|
| 520 | 400 | 120 | 1.03 | 322.44R |

The resulting velocity potential and normal derivative are given in figure 6.2.



(a) Velocity potential



(b) Normal Velocity

Figure 6.2: Velocity potential and normal velocity of cylinder oscillating with infinite frequency in the free surface

Below, are the resulting velocity potential for the same problem analyzed using the double body method. Here only the wetted surface, S_B has to be integrated over, and hence only Φ on the body is found.

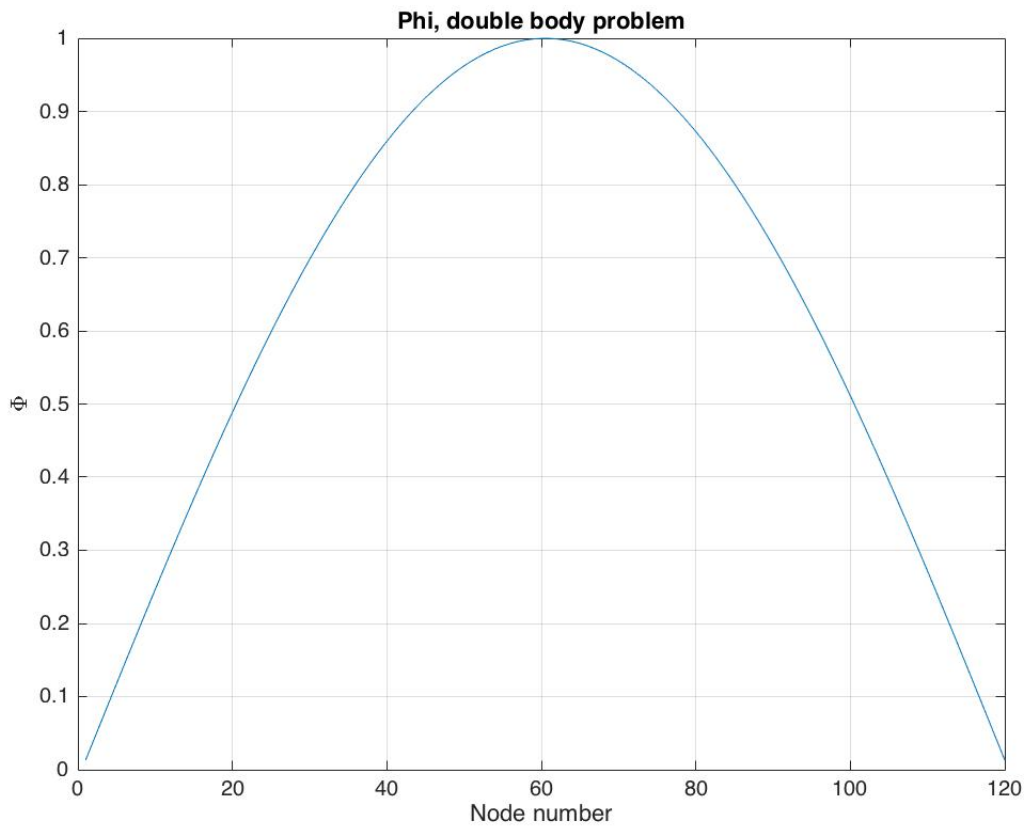


Figure 6.3: Velocity potential for half cylinder oscillating in free surface, found by double body considerations

To verify that the results are satisfactory close to the analytical value they are compared. This is done for the velocity potential, where the STD is calculated, given in table 6.2.

Table 6.2: Standard deviation between analytical solution and numerical values obtained using the double body method and Rankine source distribution

| STD $\Phi_{Doublebody}$ | STD $\Phi_{Rankine}$ |
|-------------------------|----------------------|
| 0.0001455 | 0.0001122 |

From table 6.2, we see that the numerical results agree well with the analytical solution. Lastly the added mass of the cylinder is calculated, given in table 6.3 with the analytical value. The results are normalized by $\pi \cdot R^2 \rho$.

Table 6.3: Added mass of half cylinder oscillating in free surface

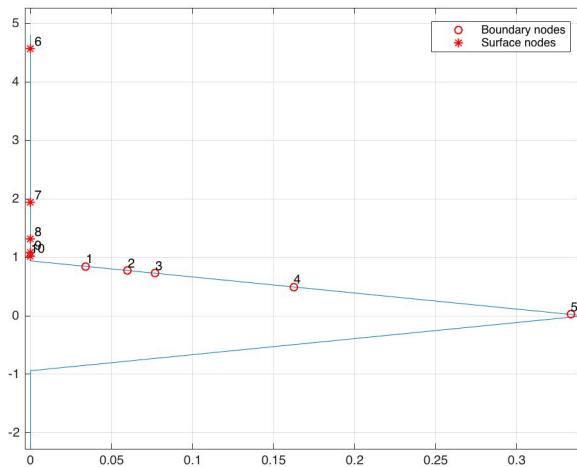
| Analytical | Double body | Free space |
|------------|-------------|------------|
| 0.5 | 0.5001 | 0.5000 |

6.3 Wedge oscillating in the free surface

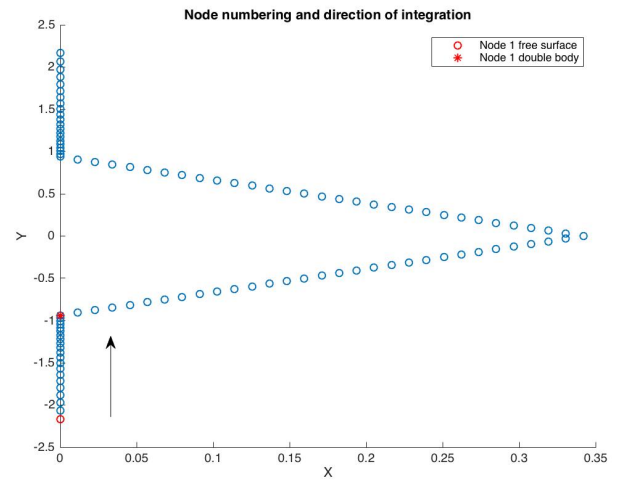
It is of interest to find the velocity potential and added mass of the wedge oscillating in a free surface. To this end, both the double body method and distributing Rankine sources, are considered. The wedge considered, is half the wedge in figure 5.25, cut by line through the two horizontal corners, and rotated 90° to the left. For the second method, which has a general Robin boundary, the boundary conditions are *not* wall sided, as for the cylinder case. This, as will be shown in the following, leads to convergence issues at the intersection between the surface and body. Because of this, p , GCI and F are calculated for both methods, and compared.

6.3.1 Convergence and Sensitivity

For both methods p , GCI and F are calculated at 5 control nodes, located at the surface of wedge, comparing values of Φ for the different grids. The numerical results for the double body problem are given in table 6.4. The wedge being analyzed has a dead rise angle of 20° , and 320, 640 and 1280 elements are used over the body surface. The location and numbering of the control nodes are given in figure 6.4 a)



(a) Control nodes



(b) Node numbering

Figure 6.4: Control nodes and node numbering

Table 6.4: p , GCI and F for wedge oscillating in the free surface, analyzed using double body

| Node | p | GCI_{32} | GCI_{21} | F |
|------|-------|------------|------------|--------|
| 1.0 | 1.138 | 0.001426 | 0.0006479 | 0.9994 |
| 2.0 | 1.134 | 0.001148 | 0.0005234 | 0.9995 |
| 3.0 | 1.125 | 0.0006263 | 0.0002872 | 0.9997 |
| 4.0 | 1.175 | 0.0004009 | 0.0001776 | 0.9998 |
| 5.0 | 1.152 | 0.002291 | 0.001032 | 0.9990 |

From table 6.4, it is observed that the GCI is satisfactory small for all control nodes, with the largest error-band being approximately 0.1%. The fraction F, also indicates that the solution is within the asymptotic range of convergence. It is therefor taken that the finest grid produces convergent results.

For the method where Rankine sources are used, p , GCI and F are in addition found at 5 control nodes on S_F , comparing values of Φ_n . In table 6.5 are the numerical values, obtained using 320, 640 and 1280 elements on the body surface. On S_F 200,400 and 800 elements are used, with a stretching factor of 1.03.

Table 6.5: p , GCI and F for wedge oscillating in the free surface, analyzed using Rankine source distribution.

| Node | p | GCI_{32} | GCI_{21} | F |
|------|---------|------------|------------|--------|
| 1.0 | 1.015 | 0.005782 | 0.002867 | 0.9977 |
| 2.0 | 1.017 | 0.004499 | 0.002227 | 0.9982 |
| 3.0 | 1.025 | 0.001909 | 0.0009391 | 0.9992 |
| 4.0 | 1.021 | 0.001197 | 0.0005902 | 0.9995 |
| 5.0 | 0.9747 | 0.006195 | 0.003145 | 1.002 |
| 6.0 | 0.9781 | 0.004726 | 0.002404 | 0.9981 |
| 7.0 | 1.031 | 0.01317 | 0.006476 | 0.9946 |
| 8.0 | 1.041 | 0.01525 | 0.00746 | 0.9937 |
| 9.0 | 1.063 | 0.02417 | 0.01168 | 0.9899 |
| 10.0 | -0.4297 | -1.435 | -1.382 | 1.398 |

From table 6.5 it can be seen that the values of GCI are satisfactory small for node 1-5, and the fraction indicates that the solution here is within the asymptotic range of convergence, even though results here are poorer than for the double body method. Further, for node 6-10 the values of GCI and F are less satisfactory moving closer to the the body surface, and the solution is not be expected to converge here.

6.3.2 Results

In the following, results form the analysis of the oscillating wedge is presented for both the double body method and Rankine source distribution. The global quantities, such as Φ and Φ_n are plotted using node number as x-label. Here the node numbering is a given in figure 6.4 b)

In figure 6.5, is the resulting velocity potential for the oscillating wedge, using the double body method. The potential is found to be the same as the first half of the double wedge potential, for the same dead rise angle.

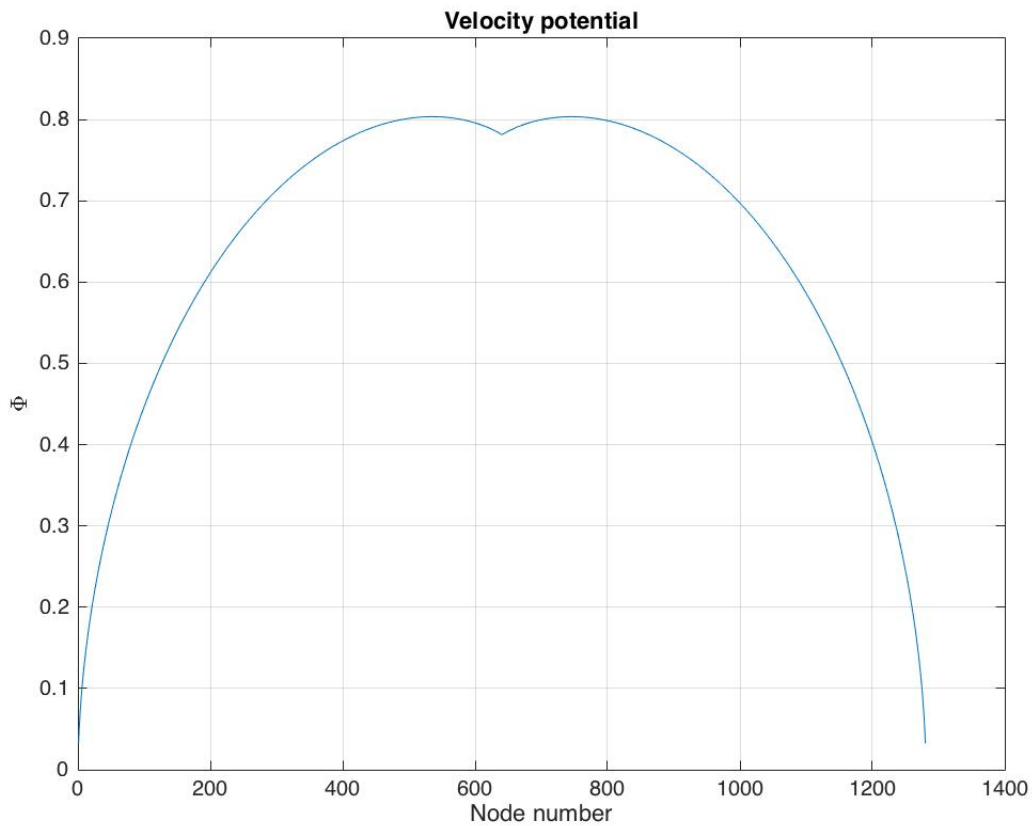
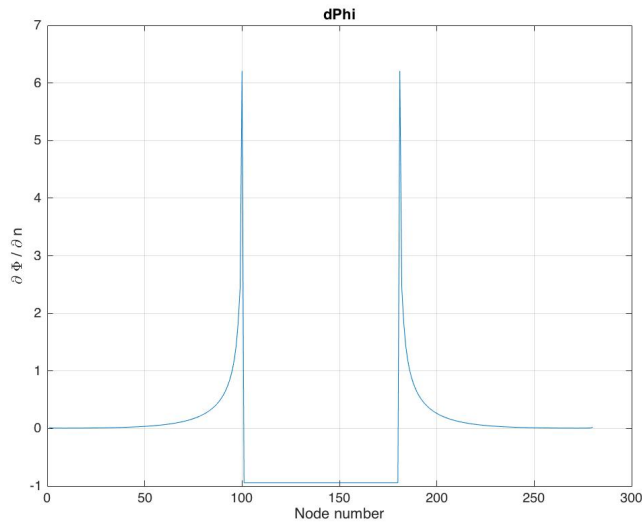
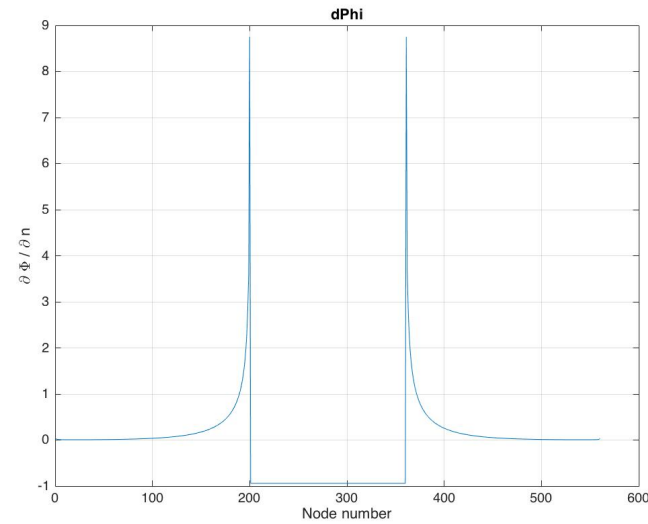


Figure 6.5: Velocity potential distribution over wedge surface, for oscillation in free surface, analyzed by double body method

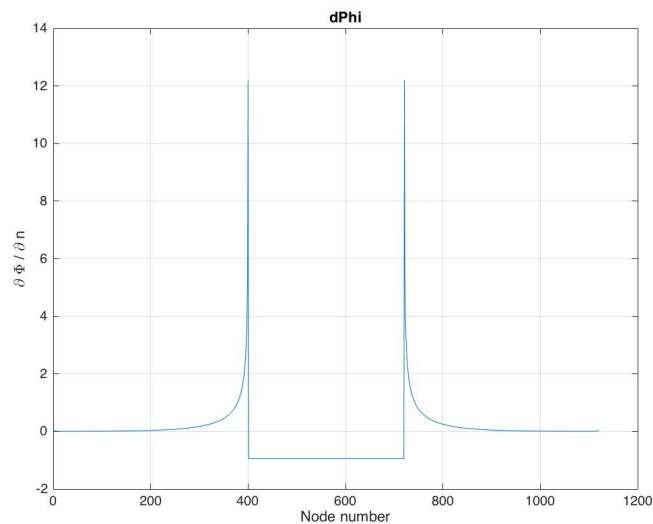
For the method where Rankine sources are used, the convergence study indicates that the solution will have convergence issues for Φ_n , in the free surface close to the body surface. In figure 6.6 are the resulting normal velocity on the body and free surface, for the three grids used in the convergence analysis.



(a) 520 elements



(b) 1040 elements



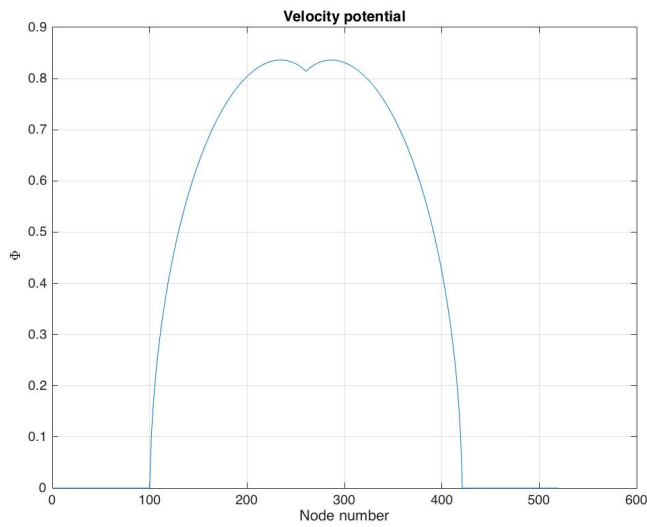
(c) 2080 elements

Figure 6.6: Normal velocity on S_F and S_B using Rankine source distribution

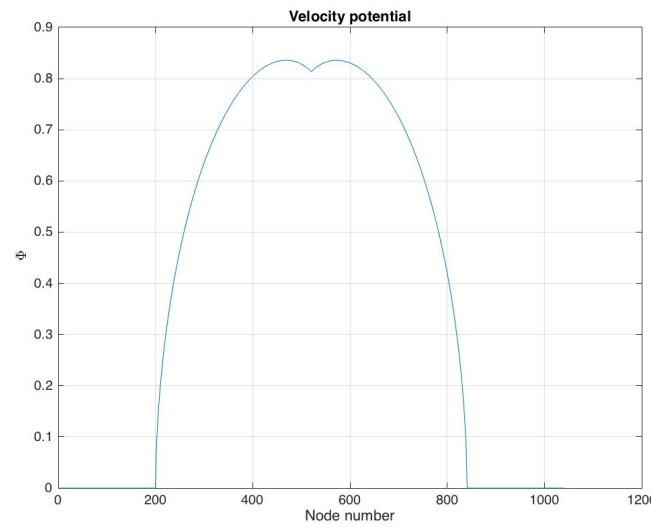
As can be seen from figure 6.6, there are two sharp peaks in three plots. These are on the free surface, increasing as we move closer to the wedge. Also, it is noted that as more elements are used, the peaks *does not* converge to towards a finite value, but continue to increase in value.

In figure 6.7, are the resulting velocity potentials for the the same grids. Here we observe that there is no convergence problem, and that the result match the results form the double body

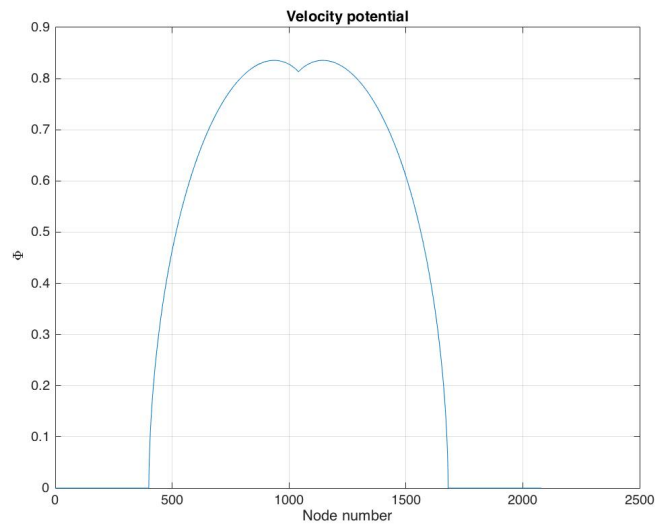
analysis.



(a) 540 elements



(b) 1040 elements



(c) 2080 elements

Figure 6.7: Velocity potential on the free surface and body of oscillating wedge

Given the fact that there is no convergence issue of the velocity potential, the the results can be used to find global quantities such as the added mass. For the same dead rise angles as in the infinite case, the added mass of the oscillating wedge is calculated. In figure 6.8 are the resulting added mass from the two methods, are plotted with the corresponding analytical solution,

which is simply the same as equation 5.3, multiplied by a half.

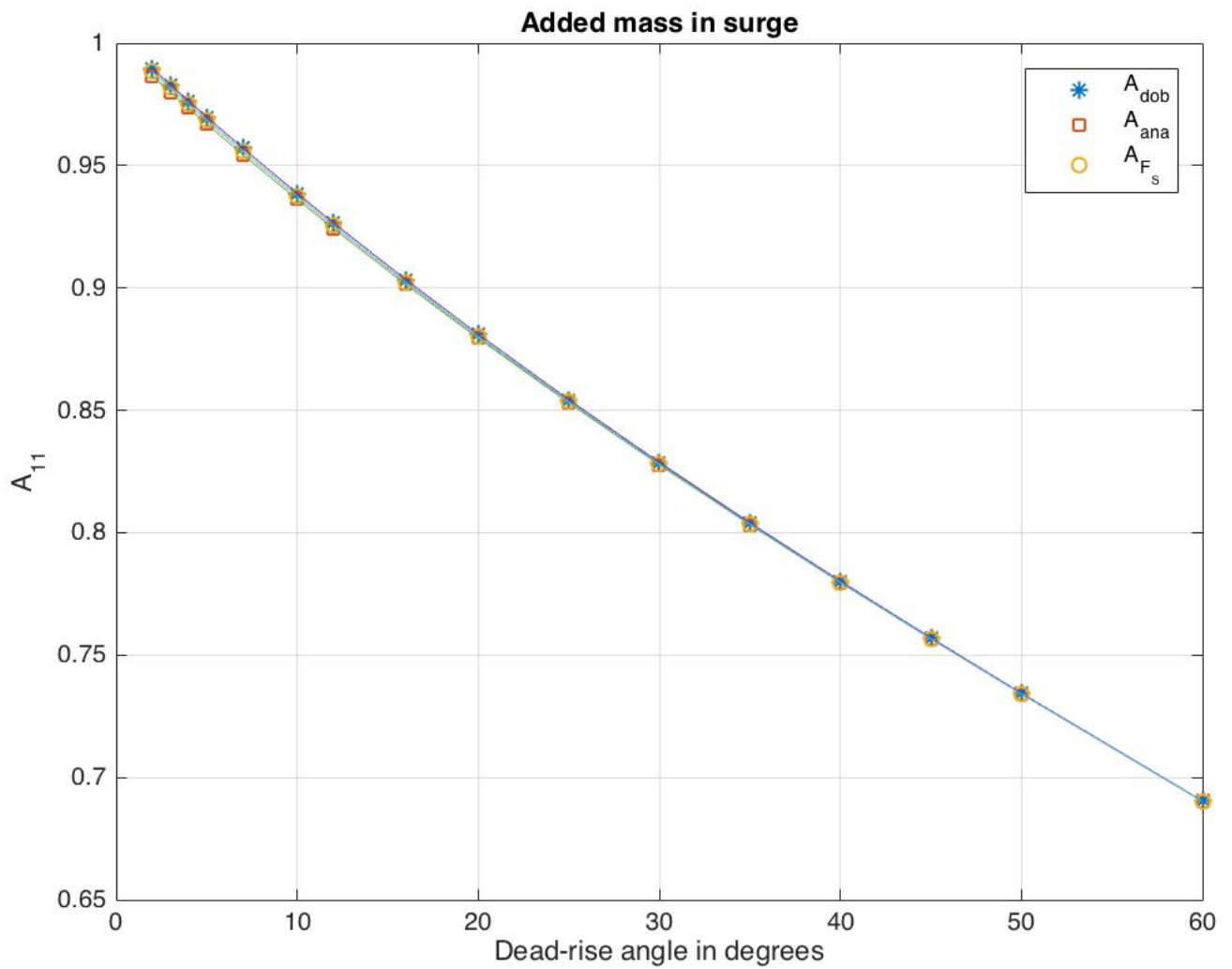


Figure 6.8: Added mass for oscillating wedge in free surface

From figure 6.8 we see that the resulting added mass for both methods, agree well with the analytical solution.

6.4 Local analysis

If a solution for both the normal velocity on the free surface and velocity potential on the surface of an oscillating body is wanted, the convergence issues in S_F have to be addressed. To this end, a local solution can be implemented in the intersection between the surface and the body. Using the same local coordinate system as defined in figure 2.4, a very similar local solution for the flow in the local domain is used. Taking inspiration from [Zhao and Faltinsen \(1996\)](#), the horizontal velocity (vertical in their case) can be written as:

$$W = -D \cdot \frac{r^{\frac{1}{2-\beta/\pi}-1}}{2-\beta/\pi} \quad (6.3)$$

Where $\beta = 2\theta$. Integrating this, a potential in the local domain is obtained as:

$$\Phi_{loc} = D_0 + D \cdot r^{\frac{1}{2-\beta/\pi}} \quad (6.4)$$

Coupling this velocity potential with the global equation as in equation 2.30, one can solve for the unknown potential and normal velocity. To determine the unknown coefficients, matching equations are used in the intersection between the local and global domain. They are:

$$\begin{aligned} W &= \frac{\partial \Phi}{\partial n} && \text{on } \Sigma_{S_F} \\ \Phi_{loc} &= \Phi && \text{on } \Sigma_{S_B} \end{aligned} \quad (6.5)$$

6.4.1 Convergence and sensitivity

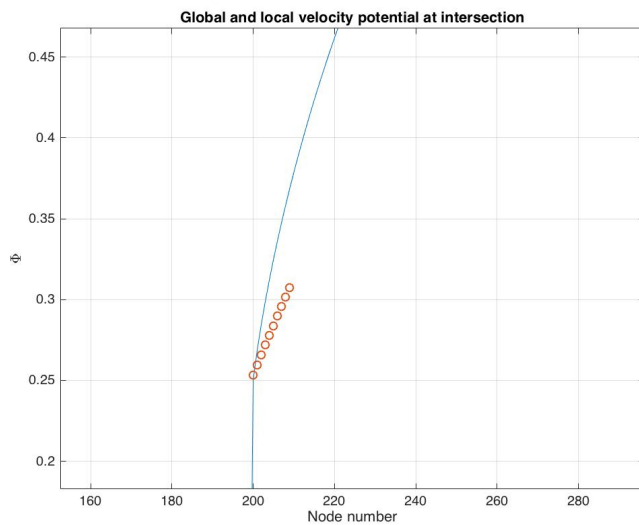
As previously, a wedge with $\theta = 20^\circ$, is analyzed. To find the size of the local domain, a sensitivity study is conducted. The criteria to decide a, is the solution where the local solution best matches the global on the 10 nodes closest to the intersection on *both* S_F and S_B . Hence the local potential and the local normal velocity is calculated on the 10 closest nodes on the body surface and free surface respectively. In the analysis, 160 elements are used on either side of the wedge, and 200 on the free surface on either side of the wedge, 720 elements in total. The STD

is then calculated for each domain size, given in table 6.6.

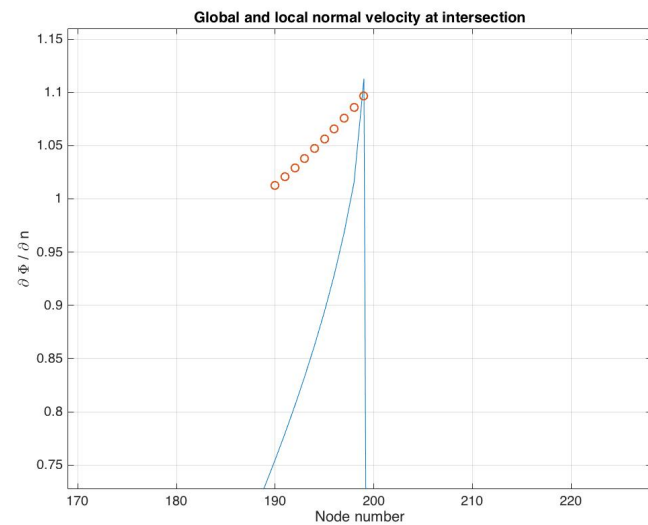
Table 6.6: Standard deviation between local and global solution, for velocity potential and normal velocity at 10 global nodes closest to intersection

| a | STD $\frac{\partial\Phi}{\partial n}$ | STD Φ | Sum |
|------|---------------------------------------|------------|-------|
| 0.01 | 0.07594 | 7.728 | 7.804 |
| 0.02 | 0.201 | 5.325 | 5.526 |
| 0.03 | 0.2635 | 4.082 | 4.346 |
| 0.04 | 0.299 | 3.327 | 3.626 |
| 0.05 | 0.3206 | 2.816 | 3.137 |
| 0.06 | 0.3342 | 2.446 | 2.78 |
| 0.07 | 0.3428 | 2.163 | 2.506 |
| 0.08 | 0.348 | 1.94 | 2.288 |
| 0.09 | 0.3509 | 1.757 | 2.108 |
| 0.1 | 0.3522 | 1.605 | 1.957 |

It is observed that the STD of the normal velocity increases as the local domain size increases and that the opposite is the case for the the velocity potential. The domain size is therefor chosen to be the one that has the least sum of STD, which is $a=0.1$. In figure 6.9 are the local solution at the 10 global nodes plotted with the global solution at both intersections



(a) Global and local velocity potential at S_B



(b) Global and local normal velocity at S_F

Figure 6.9: Global and local velocity potential and normal velocity

Further p , GCI and F are calculated on the same nodes as in figure 6.4. The analysis here is done using 320, 640 and 1280 elements on the body, and 200,400 and 800 nodes on the free surface. The obtained values are given in table 6.7

Table 6.7: F , GCI and p at surface boundary and free surface control nodes

| Node | p | GCI_{32} | GCI_{21} | F |
|------|--------|------------|------------|--------|
| 1.0 | 0.9645 | 0.02374 | 0.01228 | 0.9907 |
| 2.0 | 0.9664 | 0.01892 | 0.009754 | 0.9927 |
| 3.0 | 0.9579 | 0.007838 | 0.004047 | 0.9970 |
| 4.0 | 0.9645 | 0.003072 | 0.001689 | 0.9321 |
| 5.0 | 0.9558 | 0.003666 | 0.001917 | 0.9859 |
| 6.0 | 1.044 | 0.01569 | 0.007715 | 0.9863 |
| 7.0 | 1.045 | 0.02259 | 0.0142 | 0.7710 |
| 8.0 | 0.9236 | 0.01296 | 0.006866 | 0.9951 |
| 9.0 | 1.006 | 0.01493 | 0.007479 | 0.9940 |
| 10.0 | 1.131 | 0.006261 | 0.002852 | 1.0024 |

From table 6.7 it is found a satisfactory low band of error for all control nodes. However, F are not satisfactory close to unity for all the control nodes. It is also noted that p , is approximately half of the analytical value for constant element BEM. Despite the questionable numerical performance of the solver, the results from the analysis is presented in next section, using the finest grid

6.4.2 Results

In the following are the results obtained from analyzing the oscillating wedge, using the local solution in the intersection between the body and the free surface. The node numbering is given in figure 6.4 b).

In the figures below are the resulting velocity potential and normal velocity for both the global and normal solution. It is computed using 1280 elements on the body and 400 on the free surface.

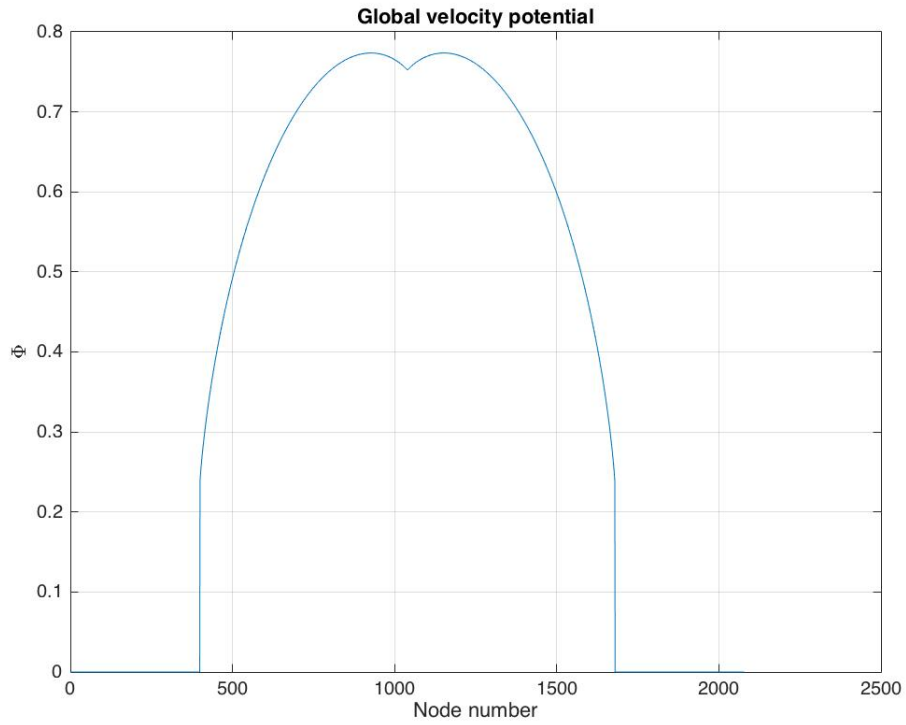


Figure 6.10: Global velocity potential of oscillating wedge, with local solution in the intersection

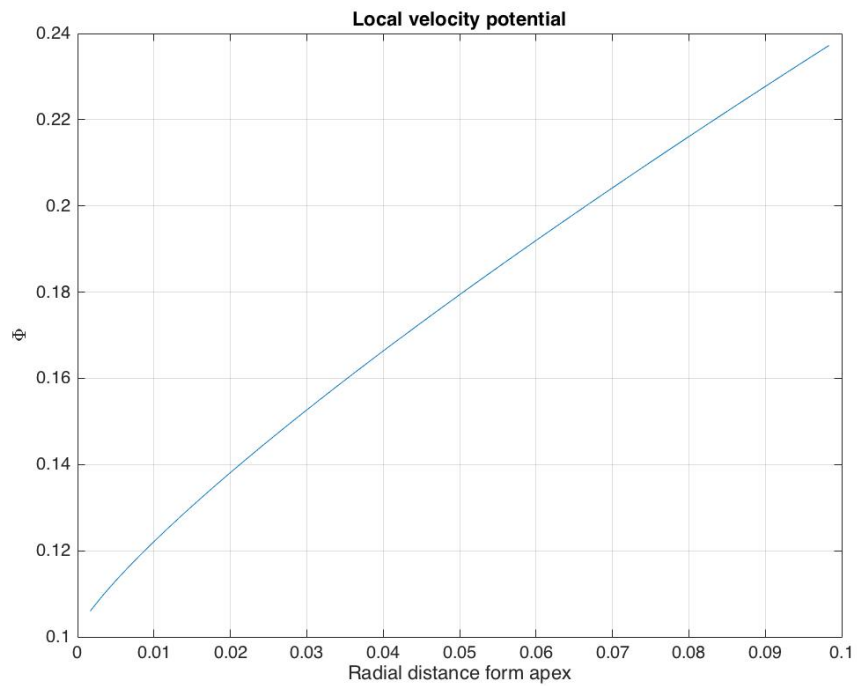


Figure 6.11: Local velocity potential of oscillating wedge, with local solution in the intersection

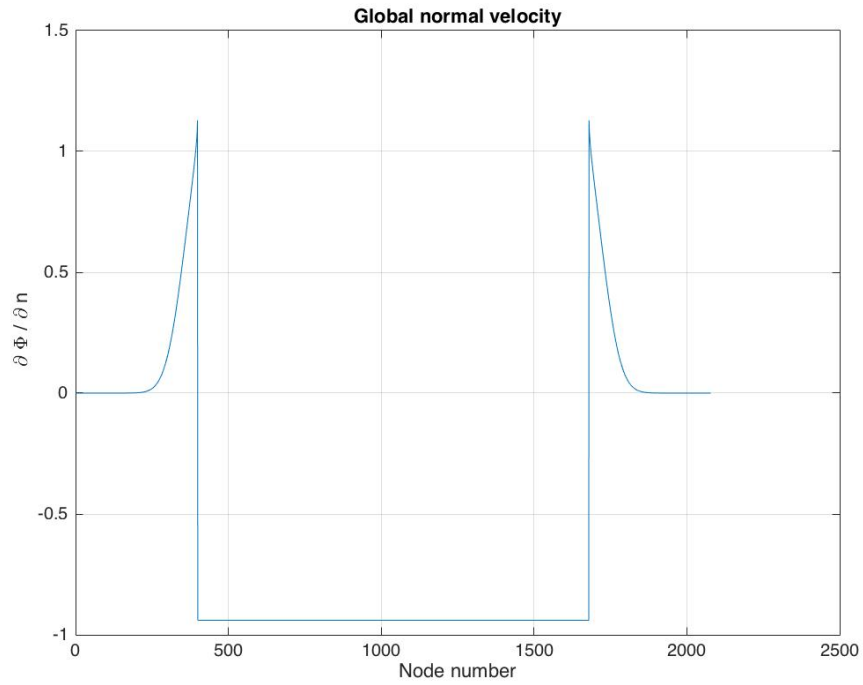


Figure 6.12: Global normal velocity of oscillating wedge, with local solution in the intersection

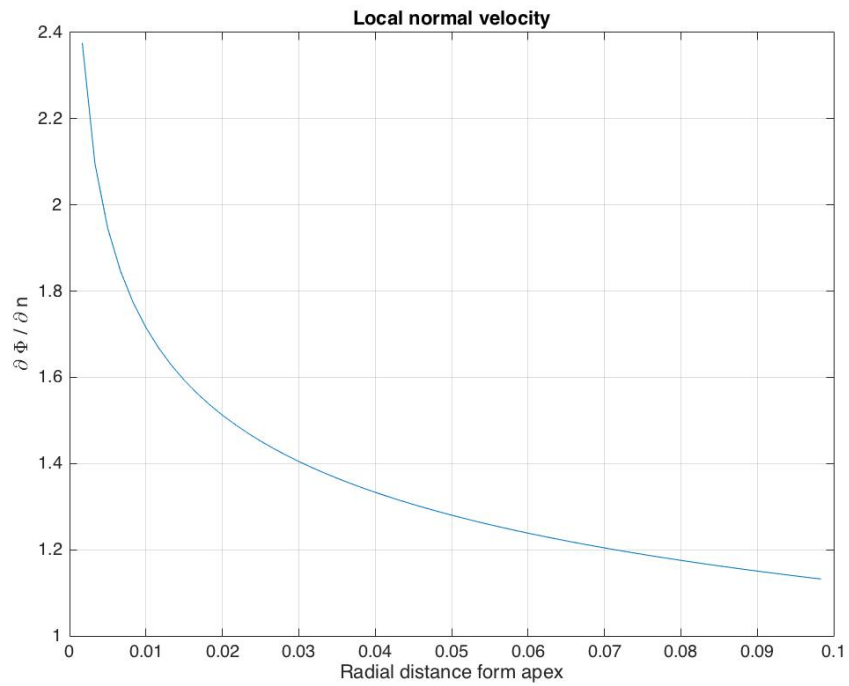


Figure 6.13: Local normal velocity of oscillating wedge, with local solution in the intersection

From the figures presented it is noted that the normal velocity in the intersection between the body and free surface, now is convergent. The matching equations used in the analysis, require continuity in the normal velocity in the intersection between the local and global domain on S_F , and in Φ in the intersection on S_B . As can be seen in table 6.8 these match well. However, the local and global solutions does not match at the opposite intersections.

Table 6.8: Global and local velocity potential and normal velocity at the intersections between the domains

| | Φ | Φ_{loc} | Φ_n | Φ_{loc_n} |
|--------------------|--------|--------------|----------|----------------|
| Intersection S_B | 0.2391 | 0.2391 | -0.9397 | 1.1287 |
| Intersection S_F | 0 | 0.2372 | 1.1287 | 1.1287 |

For comparison with the analysis done without the local model, the added mass is calculated for the the same dead rise angles as previous. The resulting added mass is normalized by $\rho\pi B^2/8$ given in figure 6.14, plotted with the analytical solution.

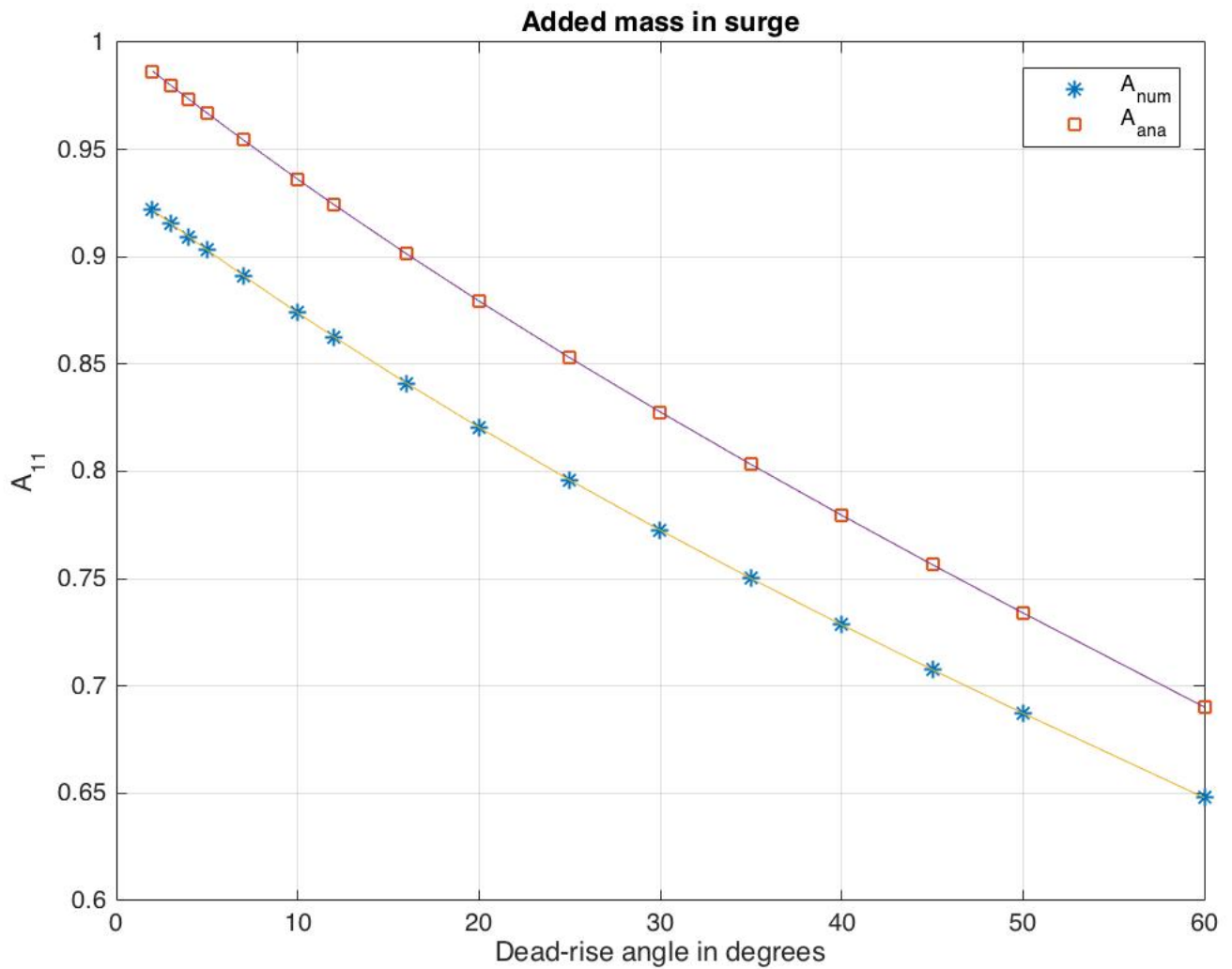


Figure 6.14: Added mass in heave of oscillating wedge, analyzed using with local flow model in intersection between the free surface and body surface

From figure 6.14 it is clear that the resulting added mass, computed with this model is underestimated. The deviation for all dead rise angles is in the range 6-9 %, which is considerable. The reason for this, and suggestions to how to deal with this problem, with more is discussed in the following chapter.

Chapter 7

Discussion and Concluding Remarks

In this chapter, the results presented in the previous chapters are discussed, and concluding arguments are made.

Several problems where the analytical solution is known, are analyzed in this thesis. Analyzing the rounded square in constant current, a clear trend is found, with increasing maximum and minimum values of Φ as the radius of curvature is decreased. The added mass is found to gradually go from the analytical value of a circle, towards the analytical value of a square, as the corners become sharper. GCI, F and p, are only calculated for $r=0.5$ for the rounded square. The results show satisfactory low GCI, and the observed order of accuracy is close to analytical value of 2. This is thought to give credibility to the BEM solver, when applied to other problems.

For the problems involving a square in infinite fluid, the local flow model is introduced. For both direct and least square matching, the sensitivity studies indicate that the best solution, is one where 2 constants and matching nodes are used. Despite this, solutions involving 4 constants are chosen. The reason for this, is that differentiating a local solution with only two constants, results in a local velocity with only one constant. Since the global velocity is not equal on both sides of the corners, the local velocity cannot match the global velocity on both sides, using only two constants. V and V_{loc} using 4 constant does not match the global velocity perfectly. The reason for this may be connected with how the differentiation is performed. For the intersection node, the global velocity is found by using forward or backward difference, which has a low order

of accuracy. Better results could likely also be achieved if the differentiation in the intersection had been done using both the local and global solution. In both the case of 2 and 4 constants, Φ_{loc} and Φ match well.

The sensitivity studies, repeatedly found that introducing matching nodes in the outer domain, gave worse matching between the solutions, while matching nodes on the boundary of the body, did not change the STD much. The reason for the outer nodes performing badly, is likely because of the location of the global nodes, where the STD is calculated. Their location, as indicated in figure 7.1, are on the body boundary, and by matching the solution on nodes which are off the boundary, the solution at these nodes have gradients which differ from the gradient at the boundary nodes, giving larger STD. Therefore using outer nodes in the matching equations, the STD should not only be calculated on the boundary nodes closest to the apex, but also at global nodes in the outer domain, close to the apex. Because of this, the solution might be said to be biased to give better results in the sensitivity study for matching nodes on the boundary. However, solutions that fits the 10 global nodes in the figure 7.1, are found in both cases, with STD in the range of 0.2-0.25, which is thought to be sufficiently small. The conclusion here is, that even though the solution is biased to perform better when boundary matching nodes are used, good local models to describe the corner flow are found.

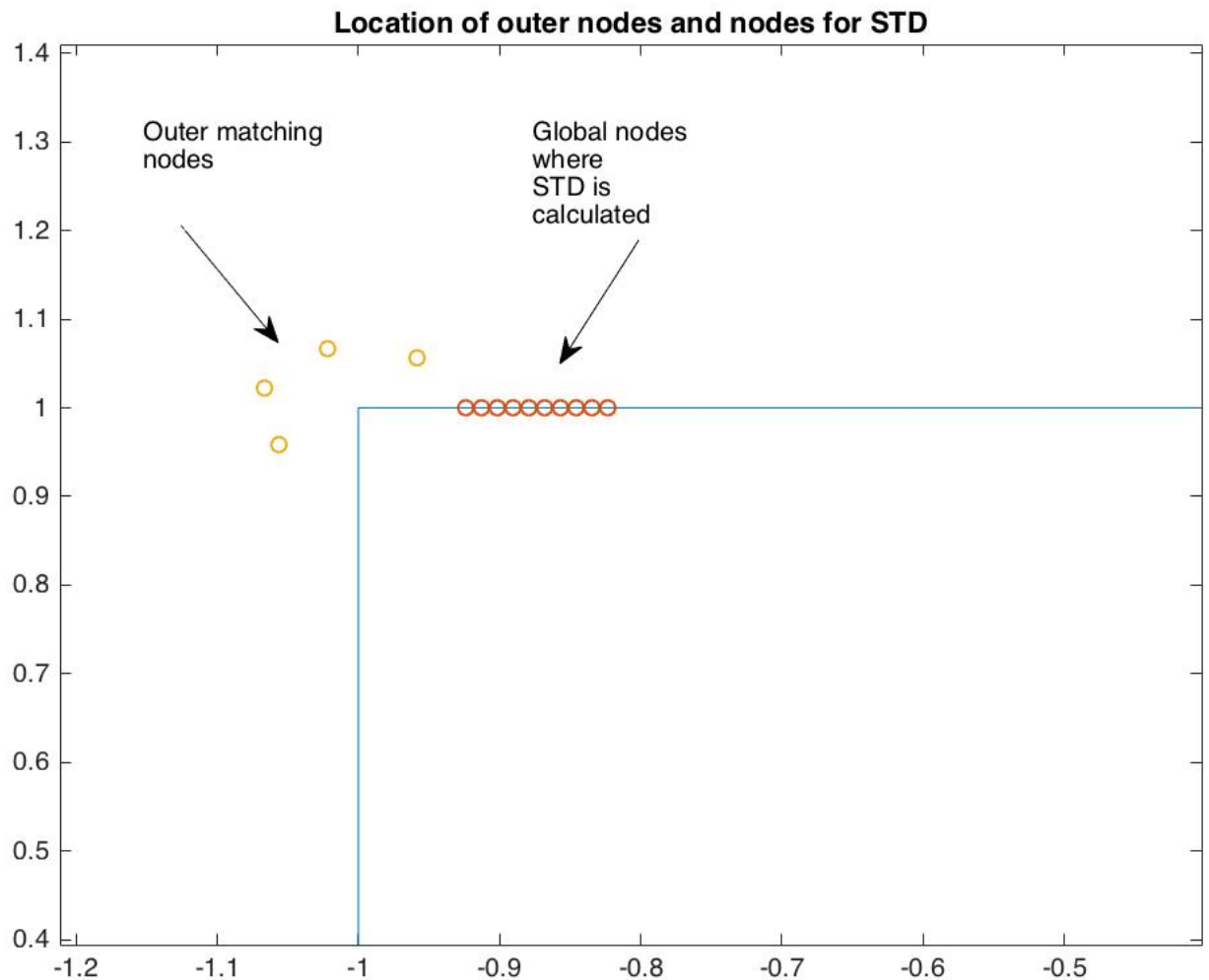


Figure 7.1: Location of outer matching nodes and global nodes where STD is calculated.

One of the great advantages with using least square matching over direct matching, is that the number of constants does not have to be the same as the number of matching nodes. This makes it possible to perform matching on a large number of nodes, while keeping the number of constants low. However, because of the biased criterion, zero outer nodes are used in the matching here as well. The number of boundary matching nodes are kept limited. The reason for this, is that they are located on the intersection or outside the local domain, and it is thought that this exceedance should be limited.

Introducing the local corner flow model, results in the velocity behaving asymptotically close to

corner, and thus better describing the velocity here, than without the local corner model. The resulting velocity potential for the current problem, found by both methods agree well and very little deviation is detected. The deviation would probably be larger, if the least square matching had been performed on a larger number of matching nodes. The results from both methods, are found to be very similar to the potential of the square without a local corner model. The added mass in both cases are close to the Lewis form added mass, deviating with about 2%.

Solving the free surface problems, both the double body method and distributing Rankine sources are considered. For the oscillating cylinder with wall sided boundary conditions, both methods produce near identical results, with negligible deviation. Both agree with the analytical solution for the half cylinder, in terms of added mass and velocity potential.

For the wedge oscillating in the free surface, the boundary conditions are not wall sided. For the double body problem, this does not matter, and the results agree well with the analytical solution. For the latter method, convergence issues are experienced in the free surface. GCI and p , for control node 10 (in S_F close to S_B), are here negative and the value of GCI , large. No convergence issues are experienced for Φ . The values here agree with those from the double body solution, and both methods produce added mass which are very close the analytical value.

Comparing the values of GCI , p and F for both methods, we find that the double body method performs better. Here the values of p are higher, GCI lower, and the the fraction F closer to unity, for control nodes 1-5. For the control nodes on S_F , it is observed that the GCI gradually becomes larger and F further from unity, moving closer to the body, indicating an increasing convergence problem.

Even though both methods produce good results for the added mass, its arguable that the double body method should be chosen if only a solution over the body surface is wanted. The first argument for this, is its better numerical performance. The second concerns the computational cost. Using Rankine sources over the body and free surface, leads to a larger set of equations to be solved, which is unnecessary if only a solution over the body is needed.

To address the convergence issue on S_F , a local flow model is introduced. Investigating the solutions sensitivity to the local domain size, the lowest STD is found to be 1.97, which is con-

siderably higher than for the square cylinders. It must however be noted that 20 nodes instead of 10 for squares, are used in the calculation. In terms of matching, the local solution *does not* match the global at the intersections, where continuity is not required, i.e $\Phi_{loc} \neq \Phi$ on S_F and $V_{loc} \neq \Phi_n$ on S_B . Because of this poor matching is achieved, and it might be fully or partly the reason for the results deviating from the analytical.

The results obtained with this model, does not agree well with the analytical results. The distribution of Φ over the wedge surface is found to be lower than for the models without the local solution, and the added mass for all dead-rise angles are considerable lower. The calculated values of p , GCI and F show that the numerical performance of this solver is poor, and might not meet the requirements to conclude that the solution is convergent. On the positive side is the fact that the model does deal with the convergence issue on S_F .

The concluding remark of this model is therefore that the results obtained with it are not reliable, and the model should be worked further on, to get more satisfactory results.

Throughout the analysis conducted in thesis, it is for several of the problems found very low estimates of the observed order of accuracy. Some of this deviation can probably be prescribed to factors such as grid stretching, coupled analysis and geometrical singularities. The effects of all these factors are not known to the author, and should be investigated further. The deviation can also be due to implementation errors either in the BEM codes or the calculation of p itself.

Chapter 8

Suggestions for further work

In this thesis, a BEM solver has been developed, and several different BVPs solved using it. Several of the results show good agreement with our expectations and analytically known solutions, while others deviate. The work done in this thesis is far from perfect, and there are several aspects which should be investigated further.

Firstly, the observed order of accuracy, p are in many cases found to be considerably lower than the analytical value of 2 for constant BEM. The reason for the low values of p , can be due to effects such as grid stretching, coupled solvers, geometrical singularities, or implementation errors. This is something that should be looked further into.

The local flow model, introduced in the intersection between S_B and S_F has poor matching with the global solution. The model should therefore be further modified, to better match the global solution at all intersections. Possible modifications are to introduce more constants in the solution, and hence enabling more matching equations to be used. It should further be investigated if the poor results obtained, are due to the limitations of the local model, or if there are other problems as well.

A very interesting expansion of the work done in this thesis, would be to use the BEM solvers developed, and implement them to solve initial boundary value problems, IBVPs. This can for instance be done for practical problems such as slamming, where the effects of the free surface are very important, and the case of geometrical singularities such as the keel of a hull, are

important to handle correctly. This is something that is left open for a future student to work with.

Appendix A

Divergence Theorem in the plane

Let A_c be an arbitrary control area in the xy -plane bounded by a closed contour boundary C , with normal vector \mathbf{n} pointing *inside* the the control area. Then the surface integral over A_c of the divergence of a two-dimensional differentiable vector function \mathbf{F} , can be written as the flow rate across C .

$$\int_{A_c} \nabla \cdot \mathbf{F} dA = - \int_C \mathbf{F} \cdot \mathbf{n} dl \quad (\text{A.1})$$

Where dl is the arc length of the contour

Appendix B

Method of Least squares

The Method of Least squares, is a method of minimizing the error committed, when fitting a curve to a discrete set of values. Given a set of data points $(x_1, y_1), \dots, (x_n, y_n)$, we can define the error of saying that the curve which best describes the trend, say $y = C_0 + \sum_{i=1}^{N_c} C_i x^{m_i}$, as:

$$E(C_0, \dots, C_{N_c}) = \sum_{i=1}^N (y_i - (C_0 + \sum_{j=1}^{N_c} C_j x_i^{m_j}))^2 \quad (\text{B.1})$$

Where N is the number of data points, and N_c is the number of constant in the approximation. Which is just N times the variance of the data set $(y_1 - y(x_1)), \dots, (y_N - y(x_N))$ To find the constants which minimizes this error, the following equation system is solved for each unknown constant C_i ?

$$\begin{aligned} \frac{\partial E}{\partial C_0} &= \sum_{i=1}^N (y_i - (C_0 + \sum_{j=1}^{N_c} C_j x_i^{m_j})) \cdot 1 = 0 \\ &\vdots \\ \frac{\partial E}{\partial C_i} &= \sum_{i=1}^N (y_i - (C_0 + \sum_{j=1}^{N_c} C_j x_i^{m_j})) \cdot (-x_i^{m_j}) = 0 \\ &\vdots \\ \frac{\partial E}{\partial C_n} &= \sum_{i=1}^N (y_i - (C_0 + \sum_{j=1}^{N_c} C_j x_i^{m_j})) \cdot (-x_n^{m_j}) = 0 \end{aligned} \quad (\text{B.2})$$

Where C_i is That is, for each constant, we differentiate with respect to it on solve the equation for the constant with minimizes the error, namely which is when the differentiated is zero.

Appendix C

Solution Details

In this thesis several different BVPs are solved using BEM. Even though all the problems are solved uniquely, several of the integrals appear multiple times. In the following, how the different integrals are evaluated is presented.

Firstly all geometries are discretized into a finite number of elements. In the simplest case, where no corners or geometrical singularities are present, there are only two integrals that needs to be evaluated at each panel, given in equation 2.11, repeated here for convince:

$$\int_{E_i} n_j \cdot \nabla \ln(r) dl(\mathbf{x}) \quad (\text{C.1})$$

$$\int_{E_j} \ln(r) dl(\mathbf{x}) \quad (\text{C.2})$$

To be able to integrate these along the boundary, a local coordinate system is defined on the element, with one axis being parallel to it, and the other perpendicular. The local angle of the element is found by geometrical considerations as, $\theta = \text{atan}((y_2 - y_1)/(x_2 - x_1))$. Further the location of the singularity is calculated relatively to the new coordinate system by:

$$\begin{aligned}
x0 &= (xs - xm) * \cos(\theta) + (ys - ym) * \sin(\theta) \\
y0 &= (xs - xm) * \sin(\theta) - (ys - ym) * \cos(\theta)
\end{aligned}
\tag{C.3}$$

where x_s , y_s , x_m and y_m are the global location of the singularities and element midpoint respectively. In the coordinate system, the new y-axis is perpendicular to the element, and integration must hence only be performed in the new x-direction. The limits of integration, (s_1, s_2) are $(-h/2, h/2)$, and the y can be evaluated analytically as:

$$\begin{aligned}
\int_{s_1}^{s_2} \ln(r) dl(\mathbf{x}) &= 1/2 (-s_1 + x0) \ln(s_1^2 - 2s_1x0 + x0^2 + y0^2) \\
&+ 1/2 (s_2 - x0) \ln(s_2^2 - 2s_2x0 + x0^2 + y0^2) - y0 \arctan\left(\frac{s_1 - x0}{y0}\right) + y0 \arctan\left(\frac{s_2 - x0}{y0}\right) + s_1 - s_2
\end{aligned}
\tag{C.4}$$

and

$$\int_{s_1}^{s_2} \frac{\partial}{\partial n} \ln(r) dl = \arctan\left(\frac{s_2 - x0}{y0}\right) - \arctan\left(\frac{s_1 - x0}{y0}\right)
\tag{C.5}$$

In the case of geometrical singularities, the local corner solution as defined in 2.26, is introduced. In the coupled method in the constant current problems, three new integrals have to be evaluated in the corner domain. They are:

$$\int_{C_{loc}} Ux \cdot \frac{\partial \ln(r)}{\partial n} ds
\tag{C.6}$$

$$\int_{C_{loc}} \left[C_0 + \sum_{i=1}^{N_c-1} C_i r^{m_i} \cos(m_i \theta) \right] \cdot \frac{\partial \ln(r)}{\partial n} ds
\tag{C.7}$$

and

$$\int_{C_{loc}} \ln(r) \cdot U \frac{\partial x}{\partial n} ds \quad (C.8)$$

The integrations are performed around the corners, but instead of using the local coordinate system as defined in figure.. the integrals are evaluated in the global domain. Two integrations are performed for each corner. For each corner, the start and end of the local domain is defined in global coordinates, and the integration is done from the start to the corner apex, and from the corner apex to the end of the domain.

The two first integrals have very complicated analytical solutions, and they are therefor integrated numerically, using 16 point adaptive Gauss integration. This is done by defining the x and y coordinates of the element as:

$$\begin{aligned} x &= 0.5(x_1 + x_2) + 0.5(x_2 - x_1) * s \\ y &= 0.5(y_1 + y_2) + 0.5(y_2 - y_1) * s \end{aligned} \quad (C.9)$$

Where x_1 , x_2 , y_1 and y_2 are the endpoints of the elements, and s the integration variable, ranging from -1 to 1. The radial distance to the corner apex in the local solution, is respected by defining r in equation C.7 as:

$$r = 0.5(a - \text{sgn} \cdot a \cdot s); \quad (C.10)$$

Where sgn is 1 in the first integral and -1 in the second.

Appendix D

Central Difference

Central difference is a numerical technique used to find the derivatives, by approximating the differential operators constituting the field equation locally [Heinzl \(2007\)](#). If the solution u , is known at node $i+1$ and $i-1$, the derivative can be approximated by using central difference defined as:

$$\frac{\partial u}{\partial x} = \frac{u_{i+1} - u_{i-1}}{2h} \quad (\text{D.1})$$

Further if only the u is known on i and $i+1$, or i and $i-1$, the differential operator can be approximated by forward and backward difference, defined respectively as:

$$\frac{\partial u}{\partial x} = \frac{u_{i+1} - u_i}{h} \quad (\text{D.2})$$

$$\frac{\partial u}{\partial x} = \frac{u_i - u_{i-1}}{h} \quad (\text{D.3})$$

The central difference schemes have following order of accuracy [Heinzl \(2007\)](#):

- Forward difference: $O(h)$
- Backward difference: $O(h)$

- Central difference: $O(h^2)$

It is hence desirable to use central difference, if possible.

Appendix E

Matlab code

E.1 Geometry

```
1 % This function discretizes the global part of the geometry, and defines
2 % end and midpoints of the elements, and normal and tangential vector.
3 % Further the location of the matching nodes on the boundary and in the
4 % outer domain is defined
5
6 function ...
    [nx,ny,tau_x,tau_y,X_e,Y_e,X_m,Y_m,B,X_out,Y_out,ang_out,ang,Sigma_M,R] ...
    = geometry(Nel,U,l,a,Nout,NB)
7
8 % Element length
9 dl = l / Nel*4;
10
11 % One side of square
12 line = l/2:-dl:-l/2;
13
14 L = length(line);
15 One = ones(1,L);
16
17 % End points of element
```

```

18 X_e = [line, -One(1:L),-line(1:L),One(1:L)]';
19 Y_e = [One, line(1:L),-One(1:L),-line(1:L)]';
20
21
22 X_m = zeros(4*L-4,1);
23 Y_m = X_m; nx=X_m; ny=nx; tau_x=nx; tau_y=ny;
24
25
26 %Loops throug all endpoints, calculating collocation points, normal and
27 %tangential vector in x and y direction
28 jj=0;
29 for j =1:4
30     for i = 1:L-1
31
32         %Index
33         h = i +(j-1)*(L-1);
34         I = h +jj;
35
36         %Collocation points
37         X_m(h) = (X_e(I) + X_e(I+1))/2;
38         Y_m(h) = (Y_e(I) + Y_e(I+1))/2;
39
40         %Normal
41         nx(h,1) = (Y_e(I+1) - Y_e(I))/dl;
42         ny(h,1) =-(X_e(I+1) - X_e(I))/dl;
43
44         %Tangent
45         tau_x(h,1) = (X_e(I+1) - X_e(I))/dl;
46         tau_y(h,1) = (Y_e(I+1) - Y_e(I))/dl;
47
48     end
49     jj = jj+1;
50 end
51
52 %Boundary condition
53 B = -U*nx;

```

```

54
55
56 % Matching nodes on boundary
57
58 %Angles of coordinate axis
59 angl = 0;
60 ang2 = 3*pi/2;
61 ang = [ang1,ang2; ang2,ang1; angl,ang2; ang2,ang1];
62
63 % Location of matching nodes on boundary
64 Sigma_M =zeros(1,4*Nb);
65 for i = 1:4
66     for j = 1:Nb
67         I = j + Nb*(i-1);
68         if i==4 && j > Nb/2
69             Sigma_M(I) = -(Nb/2-j);
70         else
71             Sigma_M(I) = Nel/4*i - (Nb/2-j);
72         end
73     end
74 end
75
76
77 %Boundary matching nodes distance form apex
78 D = abs(X_m(1) -X_m(2)); % Distance between to nodes
79 R = zeros(1,Nb); % Vector with radial distance to boundary ...
    matching nodes
80 for i = 1:Nb/2
81     R(i) = a + D*(Nb/2-i);
82 end
83 R(Nb/2+1:Nb) = fliplr(R(1:Nb/2));
84
85
86 % Defining outer matching nodes
87
88 if Nout ==0

```

```

89     X_out = 0;
90     Y_out = 0;
91     ang_out = 0;
92 else
93
94     %Pos of matching nodes outside boundary
95     X_pos = zeros(1,Nout);
96     Y_pos = X_pos;
97     angle = X_pos;
98
99     for i =1 : Nout
100         angle(i) = i * 3*pi/2 / (Nout+1);
101
102         % X and Y location of outer nodes around corner 1
103         X_pos(i) = a*cos(angle(i))-1;
104         Y_pos(i) = a*sin(angle(i))+1;
105     end
106
107     % Location of nodes, around corner 1-4
108     X_out = [X_pos,fliplr(X_pos),-X_pos,-fliplr(X_pos)]';
109     Y_out = [Y_pos,-fliplr(Y_pos),-Y_pos,fliplr(Y_pos)]';
110     ang_out=[angle,fliplr(angle),angle,fliplr(angle)];
111 end
112
113 end

```

E.2 Matcalc.m

```

1     % This script preformes the integration of the green function and ...
        it normal
2 % derivative in the local domain, and sorts the known and unknwon values
3
4 function [BB,b] = Pot_include_local(X_e,Y_e,X_m,Y_m,BCT,BCV,Nel,N_fs,N_w)

```

```
5
6
7
8 % Intgral values of normal derivative of Greens func
9 G = zeros(Nel,Nel);
10
11 % Intgral values of Greens func
12 H = zeros(Nel,Nel);
13
14 % Flips vectors to have right direction of integration
15 X_e = fliplr(X_e);
16 Y_e = fliplr(Y_e);
17 X_m = flipud(X_m);
18 Y_m = flipud(Y_m);
19
20
21 % Loops trthrough all elements
22 for j = 1:Nel
23
24     %Singular point
25     xs = X_m(j);
26     ys = Y_m(j);
27
28     jj=0;
29     for i =1:Nel
30
31         %Index
32         if i== N_fs
33             jj=jj+1;
34
35         elseif i==N_fs +N_w
36             jj=jj+1;
37         end
38
39         %Endpoints of element
40         x1 = X_e(i+jj);
```

```

41     x2 = X_e(i+1+jj);
42     y1 = Y_e(i+jj);
43     y2 = Y_e(i+1+jj);
44
45     %Element length
46     dx = x2-x1;
47     dy = y2-y1;
48     ds = sqrt(dx^2 +dy^2);
49
50     % Integration limits new coordinate system
51     s1 =-ds/2;
52     s2 = ds/2;
53
54     angle = atan2(dy,dx); %Of element
55
56     % Midpoints of panels
57     xm = X_m(i);
58     ym = Y_m(i);
59
60     %Distance to source in local coordinate system
61     x0 = (xs-xm)*cos(angle) + (ys-ym)*sin(angle);
62     y0 = (xs-xm)*sin(angle) - (ys-ym)*cos(angle);
63
64     % Integral values for G
65     if y0 ==0;
66         I2 = 1/(4*pi) * log((x0-s2)^2 + y0^2)*(s2-x0) - 1/(2*pi)*s2;
67         I1 = 1/(4*pi) * log((x0-s1)^2 + y0^2)*(s1-x0) - 1/(2*pi)*s1;
68     else
69         I2 = 1/(4*pi) * log((x0-s2)^2 + y0^2)*(s2-x0) - 1/(2*pi)*s2...
70             + 1/(2*pi)*atan((s2-x0)/(y0))*y0;
71
72         I1 = 1/(4*pi) * log((x0-s1)^2 + y0^2)*(s1-x0) - 1/(2*pi)*s1...
73             + 1/(2*pi)*atan((s1-x0)/(y0))*y0;
74     end
75
76     %Integral values of H

```

```
77     if abs(y0) ≤ 10^-10 %Rounding error
78         I = 0;
79     else
80         I = 1/(2*pi) * (atan((x0-s2)/y0) - atan((x0-s1)/y0));
81     end
82
83     % Calculated values are put into the Matrices at appropriate
84     % location
85     G(j,i) = I2-I1;
86     H(j,i) = I;
87
88     end
89 end
90
91 % Adding the loaded elements contribution
92 H = H + 0.5*eye(Nel,Nel);
93
94
95
96
97 %All known values in AA, RHS. Unknown in BB, LHS
98 AA = zeros(Nel,Nel);
99 BB = zeros(Nel,Nel);
100
101 %Loops through the elements
102 for j =1:Nel
103     for i=1:Nel
104         if BCT(i) ==0 % Phi is known
105             AA(j,i) = -H(j,i);
106             BB(j,i) = -G(j,i);
107
108         else % Phi_n is known
109             AA(j,i) = G(j,i);
110             BB(j,i) = H(j,i);
111         end
112     end
end
```

```
113 end
114
115 %Vector with known values
116 b = AA*BCV;
117
118
119 end
```

E.3 Corner

```
1 % This script preforms integration over the local corner domain of the
2 % cylinder. The terms connected with each constant in the local ...
3 % solution is
4 % stored in Coeff1 for matching nodes on the boundary, and the
5 % terms connected with outer nodes are stored in Coeff2. The ...
6 % integration is
7 % preformed in the global domain, respecting the the local potential
8 % variable ,r, distance to the corner apex. Further, direct matching
9 % between the global and local domain is preformed, where unknown elements
10 % are stored in the matrix L and known values in vector e.
11
12
13
14 function [m,Coeff1,Coeff2,L,e,b] = ...
15     Corner(X_m,Y_m,X_out,Y_out,Nel,a,Nout,NC,NB,ang_out,ang,Sigma_M,R,b,U,d)
16
17 % m values
18 m = zeros(NC/4 ,1);
19 for i = 1 : NC/4-1
20     m(i+1) = i*pi/(2*pi - pi/2);
21 end
```



```
21 % Endpoints of local domain in global coordinates
22 aa = [-1+a;-1;-1];
23 bb = [-1;-1;-1+a];
24 XC = [aa , bb , -aa , -bb];
25 YC = [-bb , aa , bb , -aa];
26
27 % Normal vector to elements, two for each corner
28 nx = [0,-1,-1,0,0,1,1,0];
29 ny = [1,0,0,-1,-1,0,0,1];
30
31 % For singularity on boundary
32 Coeff1 = zeros(Nel,NC);
33 for i =1:Nel
34
35     % Singular values
36     x0 = X_m(i);
37     y0 = Y_m(i);
38
39     Sigma =0; % Summation variable
40     jj=0;
41     for j = 1:4 %each corner
42         for k =1:2 %number of integrals
43
44             %Index
45             I = k + (j-1)*2;
46
47             % End coordinates
48             x1 = XC(k,j);
49             x2 = XC(k+1,j);
50             y1 = YC(k,j);
51             y2 = YC(k+1,j);
52
53             % Length of element
54             ds = 0.5*sqrt((x2-x1)^2 + (y1-y2)^2);
55
56             % x and y coordinate
```

```

57     x = @(s) 0.5*(x1+x2) + 0.5*(x2-x1).*s;
58     y = @(s) 0.5*(y1+y2) + 0.5*(y2-y1).*s;
59
60     % Local distance to corner apex
61     sgn = (-1)^(k+1); % Sign, integrating from a to 0 or 0 to a
62     r = @(s) 0.5.*(a - sgn*a.*s);
63
64     %Green's functions
65     g = @(s) ds.*(1/(2*pi)).* log(sqrt((x(s)-x0).^2 + ...
66         (y(s)-y0).^2));
67     g_x = @(s) nx(I).*ds.*(1/(2*pi)).* (x(s) - x0) ./ ((x(s) - ...
68         x0).^2 + (y(s) - y0).^2);
69     g_y = @(s) ny(I).*ds.*(1/(2*pi)).* (y(s) - y0) ./ ((x(s) - ...
70         x0).^2 + (y(s) - y0).^2);
71     g_n = @(s) g_x(s) + g_y(s);
72
73     % Integration terms of known values
74     Int_U = @(s) U.*x(s).*g_n(s);
75     Int_n = @(s) nx(I).*U.*g(s);
76
77     Sigma = Sigma + integral(Int_U,-1,1) - integral(Int_n,-1,1);
78
79     % Looping thourg each constant
80     for c = 1:NC/4 %For each constant in a corner
81         if c == 1
82             Int_C = @(s) g_n(s);
83         else
84             Int_C = @(s) r(s).^m(c).*cos(m(c).*ang(j,k)) .* g_n(s);
85         end
86
87         % Storing terms in matrix
88         Coeffl(i,c+jj) = Coeffl(i,c+jj) + integral(Int_C,-1,1);
89     end
90 end %for k
91
92 jj = jj + NC/4; %Index

```

```

90     end
91     b(i) = b(i) +Sigma; % Known values
92 end
93
94
95 % For singularities outside the boundary
96 Coeff2 = zeros(4*Nout,NC);
97 for i =1:4*Nout
98
99     %Singular element
100    x0 = X_out(i);
101    y0 = Y_out(i);
102
103    % Summation
104    Sigma =0;
105    jj=0;
106    for j = 1:4 % Each corner
107        for k =1:2 % Number of integrals
108
109            %Index
110            I = k + (j-1)*2;
111
112            % End points
113            x1 = XC(k,j);
114            x2 = XC(k+1,j);
115            y1 = YC(k,j);
116            y2 = YC(k+1,j);
117            ds = 0.5*sqrt((x2-x1)^2 + (y1-y2)^2);
118
119            % x and y variable
120            x = @(s) 0.5*(x1+x2) + 0.5*(x2-x1).*s;
121            y = @(s) 0.5*(y1+y2) + 0.5*(y2-y1).*s;
122
123            % Local distance to corner apex
124            sgn = (-1)^(k+1); % sign
125            r = @(s) 0.5.*(a - sgn*a.*s);

```

```

126
127     %Green's functions
128     g = @(s) ds.*1/(2*pi).* log(sqrt( (x(s)-x0).^2 + ...
129         (y(s)-y0).^2));
130     g_x = @(s) ds.*1/(2*pi).* (x(s) - x0) ./ ( (x(s) - x0).^2 + ...
131         (y(s) - y0).^2);
132     g_y = @(s) ds.*1/(2*pi).* (y(s) - y0) ./ ( (x(s) - x0).^2 + ...
133         (y(s) - y0).^2);
134     g_n = @(s) nx(I).*g_x(s) + ny(I).*g_y(s);
135
136     % Integral terms of known values, added to sigma
137     Int_U = @(s) U.*x(s).*g_n(s);
138     Int_n = @(s) nx(I).*U.*g(s);
139
140     Sigma = Sigma + integral(Int_U,-1,1) -integral(Int_n,-1,1) ;
141
142     % Coefficient terms
143     for c = 1:NC/4 %For each constant in a corner
144         if c == 1
145             Int_C = @(s) g_n(s);
146         else
147             Int_C = @(s) r(s).^m(c)*cos(m(c)*ang(j,k)) .* g_n(s);
148         end
149         Coeff2(i,c+jj) = Coeff2(i,c+jj) + integral(Int_C,-1,1);
150     end
151     end %for k
152
153     jj = jj + NC/4; % Index
154     end
155     % Known values are stores in d
156     d(i) = d(i) + Sigma + U*X_out(i);
157 end
158 %Matching equations for outer nodes
159 for i = 1:4 %For each corner

```

```

159     for j = 1:Nout % for each outer nodes in a corner
160
161         I = j + (i-1)*Nout; %row index
162
163         for c = 1:NC/4 % for each constant
164
165             CC = c + (i-1)*NC/4; %Col index
166             if c ==1
167                 Coeff2(I,CC) = Coeff2(I,CC)+1;
168             else
169                 Coeff2(I,CC) = Coeff2(I,CC)+a^(m(c))*cos(m(c)*ang_out(I));
170             end
171         end
172     end
173 end
174
175
176 %Matching equations for boundary nodes
177 L = zeros(4*Nb, Nel+NC);
178 for i =1:4*Nb
179     L(i, Sigma_M(i)) = -1;
180 end
181
182
183 for i = 1:4 % for each corner
184     for j =1:Nb % for each boundary matching node
185
186         % Row index
187         I = j + (i-1)*Nb;
188
189         % Angle of coordinate axis in local domain
190         if j ≤ Nb/2
191             theta = ang(i,1);
192         else
193             theta = ang(i,2);
194         end

```

```

195
196     for c = 1:NC/4 % for each constant
197         col = Nel + c + NC/4*(i-1);
198
199         % Local potential
200         if c==1
201             L(I,col) = 1;
202         else
203             L(I,col) = R(j)^m(c)*cos(m(c)*theta);
204         end
205
206     end %for c
207 end
208 end
209
210 % Known values
211 e = U*X_m(Sigma_M)';
212
213 end

```

E.4 Least Square

```

1 % Matches the solutions using least squares
2
3 function [LS,d] = ...
4     LeastSquare(a,ang,m,U,X_m,Nel,NB,Nout,NC,K,Coeff2,angle,f,X_out,R)
5 %Matching equations for outer nodes
6 LS =zeros(4*NC,Nel+4*NC);
7 d = zeros(4*NC,1);
8
9 % Location of matching nodes on boundary
10 Sigma_M =zeros(1,4*NB);

```

```

11 for i = 1:4 % For each corner
12     for j = 1:NB
13         I = j + NB*(i-1); %Index
14
15         if i==4 && j > NB/2
16             Sigma_M(I) = -(NB/2-j);
17         else
18             Sigma_M(I) = Nel/4*i - (NB/2-j);
19         end
20
21     end
22 end
23
24 % Adding global pot. values
25 for c =1:4 % for each corner
26     if c==2 ||c==4
27         Ang = fliplr(angle);
28     else
29         Ang = angle;
30     end
31
32     for i = 1:NC %For each equation
33         I = i + (c-1)*NC; % Row index
34
35         Start = 1 + (c-1)*Nout;
36         End = Nout + (c-1)*Nout;
37
38
39         %Adding the potential of the outer nodes
40         for j = 1:Nel
41             if i ==1
42                 LS(I, j) = sum(K(Start:End, j));
43             else
44                 Sigma = 0;
45                 for k =1:Nout
46                     KK = k +(c-1)*Nout;

```

```

47             Sigma = Sigma + a^(m(i))*cos(m(i)*Ang(k)) * K(KK,j);
48         end
49         LS(I,j) = Sigma;
50     end
51 end
52
53 %Adding potential to boundary nodes
54 for k=1:NB
55     KK = k + (c-1)*NB;
56     if i==1 %if eq 1
57         LS(I,Sigma_M(KK)) = LS(I,Sigma_M(KK)) + 1;
58     else %Not eq 1
59         if k ≤ NB/2
60             AAA = 1;
61         else
62             AAA = 2;
63         end
64         LS(I,Sigma_M(KK)) = LS(I,Sigma_M(KK)) + ...
65             R(k)^(m(i))*cos(m(i)*ang(c,AAA));
66     end
67 end
68 end %for each equation, i
69 end %for each corner, c
70
71
72 %Adding all terms connetet with local sol.
73 for c = 1:4 % for each corner
74     if c==2 || c==4
75         Ang = fliplr(angle);
76     else
77         Ang = angle;
78     end
79
80     for i = 1:NC %For eqaution
81

```



```

82     I =i + (c-1) * NC; %Index for row
83
84     for j =1:NC % for each constant in local solution
85
86         J = j + (c-1)*NC + Nel; %col index, LS
87         JJ =j + (c-1)*NC;      %col index Coeff2
88
89         Start = 1+(c-1)*Nout;
90         End    = Nout + (c-1)*Nout;
91
92         % Adding terms for outer nodes
93         if i == 1 %eq 1
94             if j ==1 % if C0 term
95                 LS(I,J)= -1*Nout + sum(Coeff2(Start:End , JJ));
96             else %not C0 term
97                 Sigma =0;
98                 for k=1:Nout %for each angle
99                     KK = k + (c-1)*Nout;
100                    Sigma = Sigma - a^(m(j))*cos(m(j)*Ang(k)) + ...
101                               Coeff2(KK, JJ) ;
102                end
103                LS(I,J) = Sigma;
104            end
105        else %if not eq 1
106
107            if j ==1 %if C0 term
108                Sigma =0;
109                for k=1:Nout %for each angle
110                    KK = k + (c-1)*Nout;
111                    Sigma = Sigma + a^m(i)*cos(m(i)*Ang(k)) * (-1 + ...
112                               Coeff2(KK, JJ) );
113                end
114                LS(I,J) = Sigma;
115            else %not C0 term
116                Sigma =0;

```

```

116         for k =1:Nout
117             KK = k + (c-1)*Nout;
118             Sigma = Sigma + a^(m(i))*cos(m(i)*Ang(k)) * ...
                    (-a^m(j)*cos(m(j)*Ang(k)) + Coeff2(KK,JJ) );
119         end
120         LS(I,J) = Sigma;
121     end
122 end
123
124 %Adding terms form the boundary nodes
125 if i==1 %if equation 1
126     if j==1 %if C0 term
127         LS(I,J) = LS(I,J) - 1*NB;
128     else %not C0 term
129         Sigma =0;
130         for k =1:NB
131             if k ≤ NB/2
132                 AAA = 1;
133             else
134                 AAA = 2;
135             end
136             Sigma = Sigma - R(k)^m(j)*cos(m(j)*ang(c,AAA));
137         end
138         LS(I,J) = LS(I,J) + Sigma;
139     end
140 else %not eq 1
141     if j==1 %if C0 term
142         Sigma =0;
143         for k =1:NB
144             if k ≤ NB/2
145                 AAA = 1;
146             else
147                 AAA = 2;
148             end
149             Sigma = Sigma - R(k)^m(i)*cos(m(i)*ang(c,AAA));
150         end

```

```

151             LS(I,J) = LS(I,J) + Sigma;
152         else % not C0 term
153             Sigma =0;
154             for k =1:NB
155                 if k ≤ NB/2
156                     AAA = 1;
157                 else
158                     AAA = 2;
159                 end
160                 Sigma = Sigma - ...
                    R(k)^(m(i))*cos(m(i)*ang(c,AAA)) * ...
                    R(k)^(m(j))*cos(m(j)*ang(c,AAA));
161             end
162             LS(I,J) = LS(I,J) + Sigma;
163         end
164     end %Adding boundary nodes
165
166     end
167 end
168 end %for each corner, c
169
170
171 %Vector d
172
173 %for all outer nodes
174 for c =1:4 %for each corner
175     if c==2 ||c==4
176         Ang = fliplr(angle);
177     else
178         Ang = angle;
179     end
180
181     for i=1:NC %for each equation
182         I = i + (c-1)*NC;
183
184         if i == 1 %if equation 1

```

```

185     Sigma =0;
186     for k = 1:Nout %for all outer nodes
187         KK = k + (c-1)*Nout;
188         Sigma = Sigma - f(KK) - U*(X_out(KK));
189     end
190     d(I) = Sigma;
191 else %Not equation 1
192     Sigma = 0;
193     for k = 1:Nout
194         KK = k + (c-1)*Nout;
195         Sigma = Sigma + a^m(i)*cos(m(i)*Ang(k)) * ( - ...
196             f(KK)-U*(X_out(KK)));
197     end
198     d(I) = Sigma;
199 end
200 end
201
202 %For the nodes on the boundary
203 for c=1:4 %for each corner
204     for i=1:NC %For each equation
205         I = i + (c-1)*NC;
206
207         if i==1 %if equation 1
208             Sigma =0;
209             for k=1:NB
210                 KK = k + (c-1)*NB;
211                 Sigma = Sigma + U*X_m(Sigma_M(KK));
212             end
213             d(I) = d(I) - Sigma;
214         else %not eq 1
215             Sigma = 0;
216             for k=1:NB
217                 KK = k + (c-1)*NB;
218                 if k ≤ NB/2
219                     AAA = 1;

```

```

220         else
221             AAA = 2;
222         end
223         Sigma = Sigma + R(k)^m(i)*cos(m(i)*ang(c,AAA)) * ...
                U*X_m(Sigma_M(KK));
224         end
225         d(I) = d(I) - Sigma;
226     end
227
228 end
229 end
230
231
232
233 end

```

E.5 Sysmat

```

1     %This program assembles and solves the the system matrix. Several ...
        places in
2 %the code, somewhat complex calculation are done, in order to put the
3 %values in the tight place in the matrix. The logic behind these will not
4 %be explained, just marked, Index.
5 %Firstly the location of the matching boundary is found and stored. Then
6 %the system matrix if filled up with contribution as follows:
7
8 function [M,C,Phi_s,Phi,G] = SysMat (Nel,Coeff1,H,U,X_m,b,d,NC,LS)
9
10 %System matrix and vector for solving the system
11 M = zeros (Nel + 4*NC,Nel + 4*NC);
12 G = zeros (Nel + 4*NC,1);
13
14 %The H matrix is put into the NxN first elements of M

```

```

15 M(1:Nel,1:Nel) = H;
16 M(1:Nel, Nel+1 : Nel+4*NC) = Coeff1;
17 G(1:Nel) = b;
18
19 % Continuity equations
20 M(Nel+1 : Nel+4*NC , 1 : Nel + 4*NC) = LS;
21 G(Nel+1 : Nel+4*NC) = d;
22
23 % The system matrix is solved for the unknown vector X.
24 X = M\G; % solving matrix equation
25
26 Phi_s = X(1:Nel); % Unknown part of the gloabal vel potential
27 C = X(Nel+1:Nel+4*NC); % Constants in the local potential solution
28
29
30 Phi = Phi_s+ U.*X_m(:,1); %Total velocity potential
31
32 end

```

E.6 Sensitivity

```

1 % This script calculates the global and local velocity potential at the 10
2 % global nodes closest to the top left corner on the sqaure cylinder
3
4 function [Phi_loc_TLC_s1] = Loc_Pot_TLC (C,m,NC,X_m,Y_m,Nel)
5
6
7 % Number of nodes
8 Nodes = 10;
9
10 % Vector where loc. pot is saved
11 Phi_loc_TLC_s1 = zeros(Nodes,1);
12

```

```

13 % Angle of local coordinate axis
14 Angle = 0;
15
16 %Global location of top left corner
17 cor = [-1,1];
18
19 %local coordinates
20 xloc = cor(1,1)-X_m(Nel/4 -9: Nel/4);
21 yloc = cor(1,2)-Y_m(Nel/4 -9: Nel/4);
22
23 %Radial distance to nodes from corner
24 r = sqrt(xloc.^2 + yloc.^2);
25 r = flipud(r);
26
27 % Calculates the local potential at the 10 nodes
28 for i = 1:Nodes %for each node
29     sigma =0;
30     for j =1:NC % for each constant in local sol
31         if j==1
32             sigma = sigma + C(j);
33         else
34             sigma = sigma + C(j)*r(i)^m(j)*cos(m(j)*Angle);
35         end
36     end
37     Phi_loc_TLC_sl(i) = sigma;
38 end
39
40
41 end

```

E.7 Convergence

```

1     % This script calculates the location of the control nods for each

```

```

2 % iteration
3
4 function [Phi_CP,P] = Controlnodes(Phi,factor,I,Nel)
5
6 %Control Nodes
7 P = [Nel/10, Nel/16, Nel/20 , Nel/8+Nel/16, Nel/4];
8
9 % Finds the correct indices of the nodes to be controlled
10 if I ==1
11     Phi_CP = Phi(P);
12 else
13     for i = 1:length(P)
14         idx1 = P(i)-factor + factor/2;
15         idx2 = idx1 + 1;
16
17         Phi_CP(i) = (Phi(idx1) + Phi(idx2))/2;
18     end
19 end
20
21 % This script calculates the observed order of accuracy, GCI and
22 % approximated true solution at the control nodes, using the save vales
23 % for Phi or dPhi
24
25 function [p,F_h0, GCI,Fraction] = OrderCheck(ContNodes,I)
26
27
28 r = 2; %Refinement factor
29 Fs =1.25; %Safety factor
30 L = length(ContNodes); %Number of Nodes checked
31
32 %Order of Accuracy
33 for i =1:L
34     for j=1:I-2
35         d1 = ContNodes(i,j) - ContNodes(i,j+1);
36         d2 = ContNodes(i,j+1) - ContNodes(i,j+2);
37

```



```

38         p(i,j) = log(d1/d2) / log(r);
39     end
40 end
41
42 %Approximated exact solution
43 for i =1:L %For each point
44     for j =1:I-2 % for each of the two finest solutions
45         F_h0(i,j) = ContNodes(i,j+2) + ...
                     (ContNodes(i,j+2)-ContNodes(i,j+1))/(r^p(i,j)-1);
46     end
47 end
48
49
50 %GCI
51 for i =1:L % for each point
52     for j=1:I-2 % for each p
53
54         for k =1:2 %for comparable GCI
55
56             col = k + (j-1)*2; % Index
57             J = j +(k-1);      % Index
58
59             eps = abs((ContNodes(i,J) - ...
                       ContNodes(i,J+1))/(ContNodes(i,J+1)));
60
61             GCI(i,col) = eps/(r^p(i,j)-1)*Fs;
62         end
63     end
64 end
65 end

```

Bibliography

- Dijkstra, W., Kakuba, G., and Mattheij, R. M. M. (2011). *CONDITION NUMBERS AND LOCAL ERRORS IN THE BOUNDARY ELEMENT METHOD*. World Scientific Publishing Co. Pte. Ltd.
- Faltinsen, O. M. (1990). *Sea loads on ships and offshore structures*. Cambridge ocean technology series. Cambridge University Press, Cambridge.
- Faltinsen, O. M. (2005). *Hydrodynamics of high-speed marine vehicles*. Cambridge University Press, Cambridge.
- Faltinsen, O. M. and Tymocha, O. (2009). *Sloshing*. Cambridge University Press, Cambridge.
- Ferziger, J. H. and Perić, M. (2002). *Computational methods for fluid dynamics*. Springer, Berlin, 3rd, rev. ed. edition.
- Heinzl, R. (2007). Concepts for scientific computing.
- Liang, H., Faltinsen, O. M., and Shao, Y.-L. (2015). Application of a 2d harmonic polynomial cell (hpc) method to singular flows and lifting problems. *Applied Ocean Research*, 53:75–90.
- Newman, J. N. (1977). Marine hydrodynamics.
- Pozrikidis, C. (2002). *A practical guide to boundary element methods with the software library BEMLIB*. Chapman and Hall/CRC, Boca Raton, Fla.
- Roache, P. J. (1998). *Verification and validation in computational science and engineering*. Hermosa, Albuquerque, N.M.
- Zhao, R. and Faltinsen, O. M. (1996). *Water entry of arbitrary two-dimensional sections with and without flow separation*. s.n., S.l.