Ship traffic flow simulated by source panel method

by

Thomas Sjåstad

Thesis

for the degree of

Master of Science

Faculty of Mathematics and Natural Sciences
University of Oslo

August 2020

Abstract

Acknowledgements

WORK IN PROGRESS I would like to thank both my supervisors Morten- Hjorth Jensen and Thomas Mestl for their guidance and insight. This thesis would not have happened without you.

Contents

Chapter 1

Introduction

Before computers and computation, data was logged manually. While traveling across the worlds seas, logging wind and current information enabled ships to decrease their travel time (Lewis 1927).

1.1 Problem outline and motivation

Transportation on sea is still to this day one of the biggest businesses within the global trade. Maritime shipping amounts to 90 % of the world's trade transportation method (UN-Business Action Hub, United Nations 2020). Route planning is central to obtain the most favorable route i.e. shortest (fuel saving), fastest (increase profit) and safest (cost). Furthermore, raising awareness of potential problems and ensure the vessel's safe passage are crucial factors in route planning. Nowadays, determining a route relies in most cases on knowledge of past voyages or the environment of a route e.g. map data. With the implementation of the Automatic Identification System data (AIS data), the maritime industry are increasingly using these data to make predictions and estimates for vessels such as streamlining cargo transfers at harbors, helping companies take better decisions and predicting traffic flow of ships in given region and time period (Wang, Li, and Zhang 2019). AIS is installed on vessels in order to track vessels. This can be seen by other vessels in a vicinity or the data is stored by coastal authorities.

However, AIS data is not always reliable or available in every possible area wanted to traverse. The data could be limited by a malfunction on land or on the vessel. For instance; a vessel could be hiding its location, turning of their transmitter. The signal could be bad which can result in loss of data. The computation time required are often quite large as the amount of data needed to make a good prediction needs to be significantly large (Pallotta, Vespe, and Bryan 2013). In addition, there are often gaps between these data points. These gaps can be significant in distance leading to unknown whereabouts of the vessels trajectory. A method which does not require AIS - data is therefore needed to fill in the remaining trajectory of the route.

The method used in the thesis was inspired and done by a self-study searching in the literature for path planning methods. A method within the field of physics was sought after. Applying potential theory to build up an artificial motion planner came to light from (Pedersen and Fossen 2012) making the assumption that streamlines could represent real vessel routes.

The analytic solution are implemented and included for their simplicity and to add additional flow characteristics to the numerical approach (mentioned below). However, the solutions does not take into account the geometry of land of arbitrary shapes which in this case is crucial as we wanted to simulate the flow given map data. This approach resulted in a guessing game where it was attempted to represent the flow field by placing different potential in the different in order to fit the map profile. A new approach was necessary. The new approach would need to take the map characteristics as input in order to suffice for the model to generalize to other areas.

The source panel method was introduced in the late 1960s and is used in the aerodynamic industry and in research to this day (Anderson 1991). The source panel method is used to calculate flows over arbitrary bodies by approximating a body with panels (straight lines) and solve for the strength coefficients at each panel. This then produces the flow over a body. The source panel method seemed as an appropriate and more applicable approach. The idea here is to produce flow-fields given information about the map as input and calculate and thus solve for the velocity components needed to visualize a flow-field. There was not found any research regarding using real maps in the literature. The closest research was from (Pedersen and Fossen 2012) where an artificial map was created containing circular objects and thus solving for the flow around each object with a given target (end point).

1.1.1 Literature overview

An overview of the research done in the path-planning field is presented. The research area constituting route predicting include multiple disciplines such as, mathematics, physics and informatics. Also, in the past several years the application of artificial intelligence, more specifically, machine learning has become quite popular in maritime vessel tracking, because of it's suitability for data treatment in multiple fields within the industry (Pallotta, Vespe, and Bryan 2013), (Meijer 2017), (Århus and Salen 2018).

Depending on method some sort of input to model is required. Here, inputs will be of the type of a starting and ending point or AIS-data.

(Besse et al. 2015) analyzes taxi-drivers' patterns in San Francisco with GPS-data. Predicting the destination of the taxi's by their choice of route with a method derived from (Besse et al. 2015) a Symmetrized Segment-Path Distance (SSPD).

(Pallotta, Vespe, and Bryan 2013) implements a unsupervised learning scheme (Traffic Route Extraction and Anomaly Detection) to predict routes in different locations. By clustering AIS-data into waypoints for certain areas in a route. Vessels are then classified with state when entering a waypoint.

(Hvamb 2015) presents three approaches for route planners. These methods requires a map. The first one using a Voronoi diagram. Making connecting lines between vertices of polygons or points describing obstacles e.g islands. Then, perpendicular lines at their half distance are drawn and connected creating Voronoi lines. Each line not intersecting the obstacles are suggested routes.

The next method by (Hvamb 2015) is based on Rapidly-exploring Random Trees (RRT). Searching an area by sampling points a graph is produced. A bias toward to the goal makes the model possible for convergence.

The final method presented in (Hvamb 2015) is applies a Probabilistic roadmap and uses this for planning routes for marine vessels.

(Pedersen and Fossen 2012) makes a route planner by applying potential flow theory on marine vessels. Setting up potentials with different flow characteristics (sink, source, doublet and uniform) a flow pattern emerges. From this, following certain streamlines lead to the desired goal. In addition, a vessel guidance scheme is derived making sure the vessel gets to its destination.

1.2 Problem statement

Assuming that shipping routes can be simulated by fluid flows the objective of this thesis is to derive a methodology that could be used to generate a shipping route from any given A to B. The methodology will be based on principles and theory in the field of fluid mechanics, more specifically potential theory. Some map information is required to use as input when making this type of model. We will study how to the model behaves in a simulated environment compared to a realistic one, also looking if the model can be generalized, creating multiple routes.

The thesis will be split in two parts, a fluid mechanical and AIS-data part. The first part of the thesis, a fluid mechanical approach. The goal is to make a velocity vector field given the map characteristics and construct a route in which the vessels can base their travel on. The area of interest will be Trondheimsfjorden. From the findings of (Pedersen and Fossen 2012), the potential flow solution is a good approach for a vessel motion planner. We will take these findings one step further, testing this approach on a geographical area. We will attempt to implement both an analytic and numerical solver in this thesis. The reason for both solvers is that the analytic does not take into account the map characteristics, but are rather simple to produce, computational light and may assist in creating a complex fluid flow. Map characteristics are crucial when wanting to find the fluid velocities surrounding arbitrary geometries. The numerical method is called Source Panel Method (Anderson 1991). By using boundary conditions and setting constraints in specified areas of the landscape makes it possible to give the map as input to the model and thus computing the flow field in the area of interest.

Application of the analytic solutions will be simulated following a numerical solution for a more computational heavy simulation. The necessary representations of a map will be objects which needs to be avoided together with a end and/or start point of route.

The second part will be a data extraction of AIS data. Given a data set containing vessel information, how is it possible to construct a route. Analyzing the data will be the first step. Looking into how to structure the data in order to find patterns which the model can be based on. An attempt to classify routes into their direction of travel will be conducted. The routes will be divided into vessels travelling ”To”, ”From” or ”Through” the designated destination. This will help getting an overview of the data as the amount of data is often to large to plot up at the same, and not systematized.

1.3 Constraints

We will assume land is the only obstacle for each vessel. In reality there could be several vessels traversing the same area at the same time which could lead to a change in course.

1.4 Outline of thesis

The thesis is further structured as follows: A theory section including the equations which emerges from the Navier - Stokes equations assuming irrotational and incompressible fluids. A deduction of the source panel method will also be done. Next, we show how to implement the source panel method to a map using Python package Basemap to create the map. Finally, a verification of the generated source panel method will be done by comparing the flow fields with actual AIS-data.

Method	Author	Pros	Cons
Voronoi diagram	Hvamb 2015	- Based on map i.e need polygons to describe obstacles - No need for AIS data (may be useful if wanting to straighten out route)	- Initial route may not be realistic - Must apply additional algorithm for getting e.g shortest route - Computational heavy to generate
RRT	Hvamb 2015	- Able to provide several routes from same or different starting points - Several trees can be initiated at once - No need for AIS data - Possible to optimize w.r.t other metrics - A bias parameter can decrease number of calculations	- Struggles in finding paths in narrow passages - Exploration method is symmetric which causes unnecessary computations
PRM	Hvamb 2015	- Does not requires AIS data - Based on map - Can compute multiple routes - Can implement a straightening out technique for shorter routes	- Finding the shortest route, which is often the case, requires an additional algorithm - Computing multiple routes comes at a higher computational cost
Potential flow	Pedersen and Fossen 2012	- No need for AIS-data - Not computational heavy - Produces multiple routes by visualizing streamlines	- Must adjust strength-parameters for optimal map representation - Streamlines could be very close to objects and must therefore be selected carefully - If flow is non-uniform around a vessel leads to deviation from the streamline
SSPD	Besse et al. 2015	- Does not requires labeled data - Implementation able with cython for optimal computation speed	- Not easy to verify how much data is required for a good model
TREAD	Pallotta, Vespe, and Bryan 2013	- No prior information needed - Amount of data is huge - Model able to predict anomalies	- Not a good model for traffic with a low density of vessels - Certain amount of pre-processing is necessary for good and precise motion patterns to be made

Table 1.1: A table summarizing and categorizing the AIS-based models (brown) and map-based models (light blue) together with pros and cons of each model.

Part I

Theory

Chapter 2

Vectors and vectors fields

In the following section the mathematics used in the thesis is presented. The theory will mostly be from the field of fluid mechanics. A brief introduction to the equation governing the properties and motion of fluids will be given, namely Navier-Stokes equations. From the general Navier-Stokes equations, assumptions and simplifications about a fluid characteristics will be made, more specifically, irrotational and incompressible flow. This will yield a set of equations which will be the used to describe the flow of fluids. These equations are part of what constitute potential theory in fluid mechanics.

In addition to a analytical flow solver, we introduce a numerical method (Source Panel Method) for solving for the flow around objects with complex geometry. By approximating an arbitrary object with a set of panels and setting out a source type flow on each panel, we solve for the unknown source strength coefficient using boundary conditions leading to a system of equations which is solved.
A more detailed description can be found in (Gjevik 2009), (Acheson 2003) and (Anderson 1991).

2.0.1 Vector

A vector is used to describe a quantity with direction and magnitude. Vectors are useful in navigation for representing quantities such as positions and velocities. From a point of reference this can be represented in a coordinate system. Mathematically written as:

$$\vec{r} = [x, y, z]. \tag{2.1}$$

The arrow indicates that r is of the type vector. The first component moves along the x-axis, second along the y-axis and the third along the z-axis. The three components indicate that the vector $\vec{r}$ exists in a three-dimensional space. This can be visualized as seen in fig. 2.1 below:

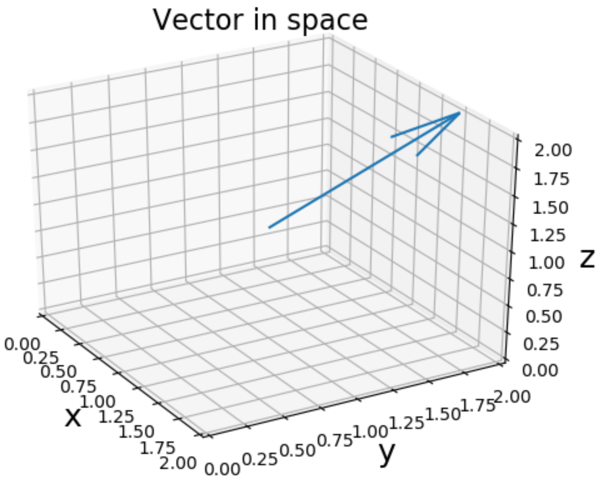


Figure 2.1: Above shows a typical vector representation in 3d-space. The arrow shows the vectors direction, its length represents its magnitude.

2.0.2 Vector field

As well as vectors, vector fields are useful when wanting to describe velocities, forces or accelerations within an area. A vector field $\vec{F}$ consists of a set of vectors assigned points within a space (see fig. 2.2). Describing motion of fluids, vector fields are commonly used when visualizing fluid motion. The vector quantities can vary, here they will represent velocities. The field are represented with vectors which are functions of spacial coordinates and time, denoted as:

$$\vec{F} = \vec{F}(x, y, z, t). \tag{2.2}$$

This can further be written as:

$$\vec{F}(x, y, z, t) = (F_x(x, y, z, t), F_y(x, y, z, t), F_z(x, y, z, t)), \tag{2.3}$$

where F_x, F_y, F_z are scalar fields. The field is called stationary if it is independent of time. See fig. (2.2) below for a stationary vector field.

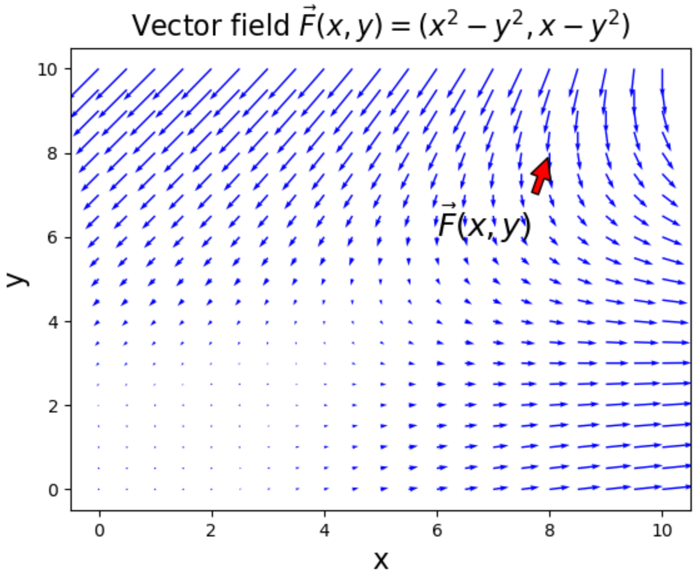


Figure 2.2: Here, a vector field is visualized $F(x, y)$ taking in two parameters x and y and calculates the field in a grid defined from [0-10] in both x and y-direction. The arrow length is the magnitude at that specific point, and also indicating the fields direction.

Chapter 3

Navier-Stokes equations

3.0.1 Navier-Stokes equations

The Navier-Stokes equations are the equations describing the motion of a fluid. The equations are derived using the laws of conservation of quantities such as mass, momentum and energy.
The continuity equation arise from the law of conservation of mass which states that mass can not be created or disappear. The general equation of continuity is stated below:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0, \tag{3.1}$$

where ρ is the fluid density, $\nabla = [\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}]$ the gradient and $\vec{u}$ the velocity vector. We will assume incompressible fluids i.e. the density ρ is constant. The continuity equation for incompressible fluid states:

$$\nabla \cdot \vec{u} = 0. \tag{3.2}$$

Both density terms are excluded as $\frac{\partial \rho}{\partial t} = 0$, and dividing by the density ρ.
This leads up to the next equation, the momentum equation for incompressible fluids. It states that a fluid's density times acceleration is proportional to the forces acting upon the fluid.

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \vec{u} + f_v. \tag{3.3}$$

The left hand side states the inertia term where $\vec{u}$ is the fluid velocity, $\frac{\partial \vec{u}}{\partial t}$ and $(\vec{u} \cdot \nabla)\vec{u}$ are the local and convective acceleration, respectively. The right hand side gives pressure gradient ∇p, the fluid density ρ, the kinematic viscosity ν. Viscous forces due to the stickiness of a fluid is given by the term $\nu \nabla^2 \vec{u}$ and external forces f_v. The "shipping" fluid is stationary $\frac{\partial \vec{u}}{\partial t} = 0$ and inviscous $\nu = 0$. These assumptions leads further to potential flow theory which will be covered next.

Chapter 4

Irrotational and incompressible flow

Navier Stokes' equations are nonlinear, integral equations which do not solve analytically. It is therefore commonly applied two approaches in order to solve Navier Stokes' equations. One, is to simplify the problem at hand to the extent that the equations become linear and therefore can be solved analytically. The second is, numerical methods used to solve the equations, allowing for a more realistic simulations and solutions. This is more commonly known as Computational Fluid Dynamics (CFD). The nonlinear integral terms are replaced by a discretized approach yielding solutions at specified points (or grid). We cover these methods in the following chapters.

4.0.1 Stream function

Given a vector field $\vec{F} = \vec{F}(x, y, z, t)$, it is possible to visualize the field $\vec{F}$ by field lines. Field lines can be described as tracing a line following the direction of a vector field $\vec{F}$. By looking a specific point P in time, field lines have the vector $\vec{F}$ as tangent. Suppose $d\vec{r} = (dx, dy, dz)$ is a small arc element on a field line at point P. The cross product between the vector $\vec{F}$ and $d\vec{r}$ is zero. If the vector $\vec{F}$ represents the velocity vector $\vec{u} = [u(x, y, z, t), v(x, y, z, t), w(x, y, z, t)]$ of a fluid, the cross product between the velocity $\vec{u}$ and $d\vec{r}$ can be written as:

$$\vec{u} \times d\vec{r} = 0. \tag{4.1}$$

Writing out the components from the cross product gives:

$$(vdz - wdy)\vec{i} = 0,$$

$$(wdx - udz)\vec{j} = 0,$$

$$(udy - vdx)\vec{k} = 0.$$

This gives the equations:

$$\frac{dx}{u} = \frac{dy}{v} = \frac{dz}{w}, \tag{4.2}$$

assuming $u, v, w \neq 0$. Now, $\vec{F}$ represents a velocity $\vec{u}$, the corresponding field lines are then called streamlines. As mentioned above, assuming a fluid's flow is stationary, the trajectory and streamlines are the same. Assuming a 2-dimensional space, streamlines give rise to differential equations with the velocity components u and v known. Rearranging eq.(4.2) and with u and v known the trajectory, thus streamlines can be found by integrating:

$$\frac{dy}{dx} = \frac{v}{u}. \tag{4.3}$$

This differential equation yields solutions on the form $y(x, C)$, where y represents streamlines with a given integration constant C. Next, we see how we can relate the velocities the stream function ψ.

For a 2-dimensional description the equation for an incompressible flow yields:

$$\nabla \cdot \vec{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{4.4}$$

where the velocity vector $\vec{u} = [u, v]$. $\nabla = [\frac{\partial}{\partial x}, \frac{\partial}{\partial y}]$ is the differential operator. We can relate the velocity vector to the stream function by defining the stream function as:

$$u = \frac{\partial \psi}{\partial y}, \tag{4.5}$$

$$v = -\frac{\partial \psi}{\partial x}, \tag{4.6}$$

where ψ is the stream function. By putting eq's. (4.5) and 4.6) into eq. (4.4) we find:

$$\nabla \cdot \vec{u} = \frac{\partial^2 \psi}{\partial x \partial y} - \frac{\partial^2 \psi}{\partial x \partial y} = 0, \tag{4.7}$$

satisfying the continuity equation. Assuming an irrotational velocity field, $\nabla \times \vec{u} = 0$, the Laplace equation is obtained (see next section for more details):

$$\nabla \times \vec{u} = \frac{\partial^2 \psi}{\partial y^2} + \frac{\partial^2 \psi}{\partial x^2} = \nabla^2 \psi = 0, \tag{4.8}$$

where eq. (4.5) and eq. (4.6) are substituted in as expressions for the velocity components.

Furthermore, eq. (4.1) gives us the conditions on what the stream function must look like in order for this to be true. Putting eq. (4.5) and eq. (4.6) into eq. (4.1) and constraining it to a 2-dimensional flow the stream function yields:

$$udy - vdx = \frac{\partial \psi}{\partial y} dy + \frac{\partial \psi}{\partial x} dx - = 0, \tag{4.9}$$

$$\partial \psi = 0, \tag{4.10}$$

$$\psi = Constant. \tag{4.11}$$

The stream function ψ is constant along a streamline. All stream functions ψ satisfying eq (4.7) are valid choices for ψ. Different stream functions ψ can solve the Laplace equation yielding various flow characteristics. *Sinks*, *sources* and *doublets* are fundamental solutions. These will be derived in the next section. The velocity components v_r and v_θ given a stream function can be written in polar coordinates as:

$$v_r = \frac{1}{r}\frac{\partial \psi}{\partial \theta}, \tag{4.12}$$

$$v_\theta = -\frac{\partial \psi}{\partial r}, \tag{4.13}$$

and the Laplacian in polar coordinates:

$$\nabla^2 \psi = \frac{\partial^2 \psi}{\partial r^2} + \frac{1}{r}\frac{\partial \psi}{\partial r} + \frac{1}{r^2}\frac{\partial^2 \psi}{\partial \theta^2} = 0. \tag{4.14}$$

The radius r is related to a Cartesian coordinate system as $r = \sqrt{x^2 + y^2}$. The angle $\theta = \arctan(\frac{y}{x})$ is the angle from origo to r.

4.0.2 Velocity potential

Another way of relating a scalar function to the Laplace equation is what is known as the velocity potential. In the general case, the velocity potential is a scalar function denoted by $\Phi(x, y, z, t)$. In the following, we will consider a 2-dimensional scalar function $\Phi(x, y)$ in a steady state. We will use the same equation as for the stream function, but with a slightly different approach. Instead of starting with the continuity eq. (4.4), we assume an irrotational flow $\nabla \times \vec{u} = 0$ leading to the existence of a scalar function $\Phi(x, y)$ satisfying the irrotational property. As for the stream functions the velocity is said to be derived from a velocity potential and can be written as:

$$\vec{u} = \nabla\Phi, \tag{4.15}$$

where $\vec{u}$ is the fluid velocity, $\nabla = [\frac{\partial}{\partial x}, \frac{\partial}{\partial y}]$ is the differential operator and $\Phi(x, y)$ a scalar function. Inserting the above equation into eq. (4.4) we find:

$$\nabla \cdot \vec{u} = \nabla \cdot (\nabla\Phi) = \nabla^2\Phi = 0. \tag{4.16}$$

We have now related a scalar function Φ to the Laplace equation (4.16) as we did with the stream function ψ. The velocity components for the vector $\vec{u} = [u, v]$ are then given the velocity potential Φ:

$$u = \frac{\partial\Phi}{\partial x}, \tag{4.17}$$

$$v = \frac{\partial\Phi}{\partial y}. \tag{4.18}$$

The equations derived in this section are the fundamental equations needed describing fluid flows. The velocity potential and stream function theory are what is known as potential theory in fluid mechanics. Next, we look at the Laplace equation and derive some fundamental flow characteristics.

Chapter 5

Fundamental flows in two dimensions

5.0.1 Solutions of the Laplace equation

Here, analytical solutions of the Laplace equation are derived. In the literature, articles from (Pedersen and Fossen 2012) and ("Real-Time Obstacle Avoidance Using Harmonic Potential Functions") use the Laplace equation for motion planning and obstacle avoidance by building an artificial potential field creating flow characteristics based on the different solutions to the Laplace equation. Common flow characteristics and solutions to the Laplace equation are: *source*, *sink* and *doublet* flow. The same will be done in this thesis. A *source* representing a starting point for vessels, *sink* representing an end point and a *doublet* representing circular objects. These flow characteristics can be combined by applying the superposition principle, thus simulating basic flow patterns. The superposition states that if the velocity potential ϕ_1 is a solution of eq. (4.16) and ϕ_2 is a solution of eq. (4.16) then the sum of ϕ_1 and ϕ_2 is also a solution of eq. (4.16).

Uniform flow

A uniform flow has constant velocity components u and v in x and y direction, respectively. By the relation from eq (4.5) and eq (4.6) this satisfies eq (4.7) and is therefore a solution.

Sink and source flow

Consider a source flowing radially outwards from a point with a steady rate. By finding the amount flowing outwards, the velocity components of this flow can be found by integrating the flow over a circle with radius r this yields:

$$Q = \int_0^{2\pi} v_r r d\theta = 2\pi r v_r, \tag{5.1}$$

where Q is the flow strength. Substituting eq. (4.12) into eq. (5.2) for the radial velocity and integrating. This type of flow has the characteristics of a source and has the stream function:

$$\psi_{source} = \frac{Q}{2\pi}\theta \tag{5.2}$$

This is also a solution of the Laplace equation (4.8). The corresponding velocity components take the form:

$$u_{source} = \frac{Qx}{2\pi(x^2+y^2)}, \tag{5.3}$$

$$v_{source} = \frac{Qy}{2\pi(x^2+y^2)}, \tag{5.4}$$

where eq. (4.12) is equated with eq. (5.1) for the radial velocity and substituting $r = \sqrt{x^2+y^2}$. The stream function satisfies the Laplace equation. A flow which absorbs fluid isotropically is called a sink and has the same stream function as a source but with a negative sign:

$$\psi_{sink} = -\frac{Q}{2\pi}\theta, \tag{5.5}$$

sharing the same terms as the source but with opposite sign.

$$u_{sink} = -\frac{Qx}{2\pi(x^2+y^2)}, \tag{5.6}$$

$$v_{sink} = -\frac{Qy}{2\pi(x^2+y^2)}. \tag{5.7}$$

Doublet flow

A sink and source flow coinciding at the same point is called a doublet flow. By placing a sink in a point, $(\epsilon, 0)$ with angle θ_2 and a source in $(-\epsilon, 0)$ with angle θ_1. ϵ is small distance. This gives the stream function using the superposition principle:

$$\psi = -\frac{Q}{2\pi}(\theta_1 - \theta_2), \tag{5.8}$$

taking the tangent on each side gives:

$$\tan\Big(-\frac{2\pi\psi}{Q}\Big) = \tan(\theta_1 - \theta_2) = \frac{\tan(\theta_1) - \tan(\theta_2)}{1 + \tan(\theta_1)\tan(\theta_2)}. \tag{5.9}$$

Further we rewrite $\tan(\theta_1)$ as $\frac{y}{x-\epsilon}$ together with $\tan(\theta_1)$ as $\frac{y}{x+\epsilon}$ and insert this into eq (5.9).

$$\tan\Big(-\frac{2\pi\psi}{Q}\Big) = \frac{\frac{y}{x-\epsilon} - \frac{y}{x+\epsilon}}{1 + \frac{y^2}{(x^2-\epsilon^2)}}. \tag{5.10}$$

After cleaning up the equation reads:

$$\tan\Big(-\frac{2\pi\psi}{Q}\Big) = \frac{2y\epsilon}{x^2 + dy^2 - \epsilon^2} \tag{5.11}$$

Taking the inverse tangent on both sides of and remembering that ϵ is small a distance gives small values in the argument of arctan. For small values of a gives:

$$-\frac{2\pi\psi}{Q} = \arctan\Big(\frac{2ya}{x^2 + y^2 - \epsilon^2}\Big) \approx \frac{2y\epsilon}{x^2 + y^2 - \epsilon^2}, \tag{5.12}$$

and solving for the stream function yields:

$$\psi = -\frac{Q\epsilon y}{\pi(x^2 + y^2 - \epsilon^2)}. \tag{5.13}$$

As ϵ tends toward 0, Q will tend toward infinity. In this case, let $\epsilon \to 0$ and $Q \to \infty$ so that there product is constant. This can be written as:

$$\lim_{\substack{\epsilon\to 0\\ Q\to\infty}} -\frac{Q\epsilon y}{\pi(x^2 + y^2 - \epsilon^2)} = -\frac{ky}{\pi(x^2 + y^2)}, \tag{5.14}$$

where k is the strength of the doublet. The stream function equation for a doublet type flow reads:

$$\psi_{doublet} = -\frac{ky}{\pi(x^2 + y^2)}. \tag{5.15}$$

The velocity components u and v for a doublet is found from eq. (4.5) and eq. (4.6), respectively:

$$u = \frac{k(y^2 - x^2)}{(x^2 + y^2)^2}, \tag{5.16}$$

$$v = -\frac{kyx}{(x^2 + y^2)^2}. \tag{5.17}$$

Fig's. (5.1 - 5.3) visualize fundamental flows derived from the Laplace equation. Fig. 5.4 shows how the Laplace equation is able to create more complex flow with the superposition principle.

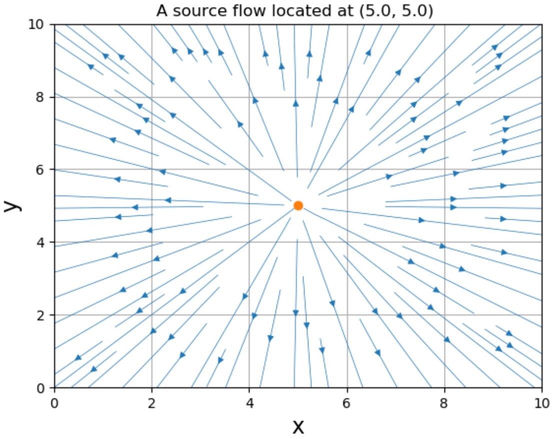


Figure 5.1: This flow characteristic is called a source flow. The velocity is points outward from it's origin (orange point). This flow characteristic emits fluid making it a representation for initial positions and circular objects.

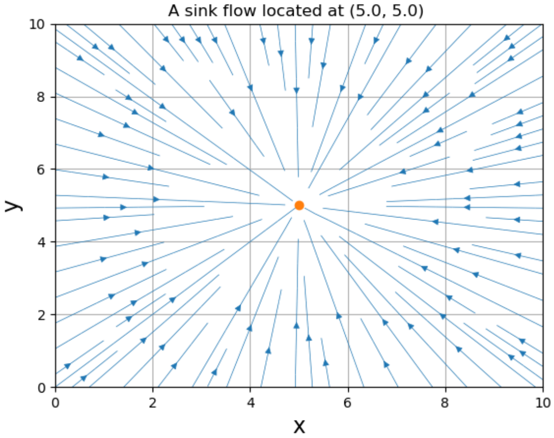


Figure 5.2: A solution of the Laplace equation produces streamlines directed radially inward to it's origin (orange point). The velocity decreases inversely proportional from it's distance from the origin. This flow characteristic is called a sink flow. This flow is useful when representing end point of routes as the sink absorbs the fluid.

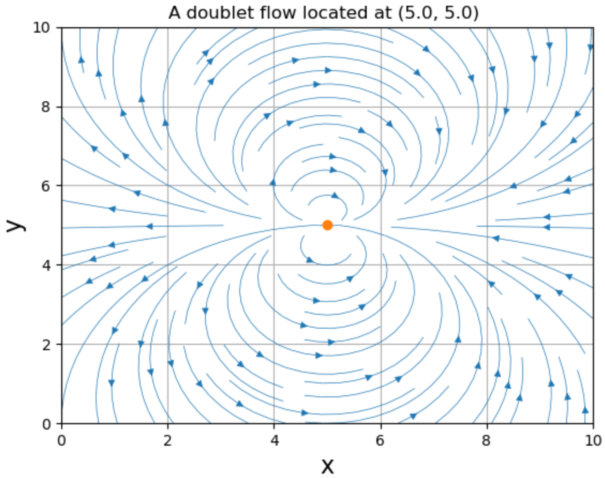


Figure 5.3: A doublet emerges from a sink and source coinciding in the point. A doublet represents objects i.e islands and land. From the figure we see the streamlines flow around the defined point of origin (orange).

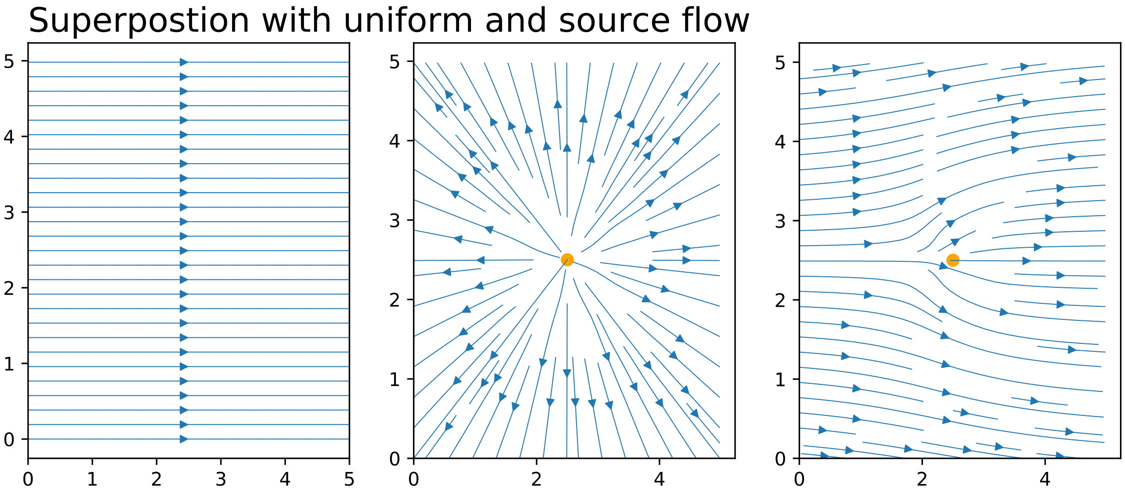


Figure 5.4: left a uniform flow $\vec{u} = (const, 0)$ is shown , middle shows a source flow located at $(2.5, 2.5)$ and right, combinations of both using the superposition principle

Chapter 6

Source Panel Method (SPM)

Whereas the situation gets more complex and the geometry can no longer be described by fundamental flows, numerical methods needs to be applied to solve for velocities in such areas. The source panel method (SPM) approximates arbitrary geometric shapes with a set of panels. By using boundary conditions at the surface and placing a small source potential a system of equations is set up and solved. The idea is to distribute sources around a body so that it deflects the incoming flow, creating a flow around or along a body. The following section explains in detail how the source panel method is set up with boundary conditions and geometry, how to calculate the source strengths and the velocity flow at every grid cell in the defined area leading us to a flow characteristic.

6.0.1 Boundary conditions

From the previous section we have different solutions leading to various flow characteristics such as uniform, source, sink and doublet flow. It was also found that the superposition principle could be applied to the Laplace equation because of its linearity property. The superposition principle allows for more realistic flow types than elementary solutions of the Laplace equation. One can simulate the effect from objects placed in the flow-field by restricting the flow at certain points in the region i.e. using boundary conditions (see fig. (6.1)). Boundary conditions replaces islands and coastlines. This will provide us with an estimate of a shipping route.

Islands and coastlines are impenetrable. Mathematically, the normal velocity must be zero at the surface. In addition, for inviscid fluids the tangential velocity component at the surface of the body is finite. The equations which arise from these boundary conditions can be written mathematically as:

$$\vec{V} \cdot \vec{n} = 0, \tag{6.1}$$

where $\vec{V}$ is the flow velocity and $\vec{n}$ is the normal vector pointing out of the body. The tangential boundary condition is denoted:

$$\vec{V} \cdot \vec{t} = c, \tag{6.2}$$

where $\vec{V}$ is the flow velocity, $\vec{t}$ is the tangential vector and c constant. For flow very far away from the body we assume the velocity to be uniform:

$$u = V_\infty, \tag{6.3}$$

where u is the velocity component in x-direction and V_∞ is a constant value resulting in a uniform flow in the x-direction, whereas the y component of the velocity vector is zero:

$$v = 0. \tag{6.4}$$

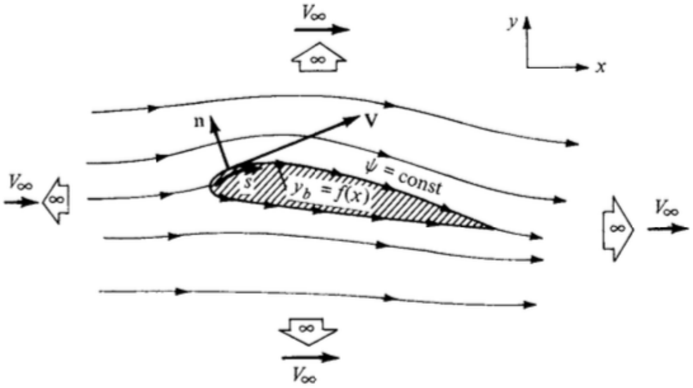


Figure 6.1: The figure shows flow around an airfoil. The uniform flow is the same as the equation above i.e. only contribution in the x-direction. Image gathered from (Anderson, 2007)

6.0.2 Approximating complex geometries with panels

When solving for the flow around a solid body i.e. the flow around a circular cylinder, the body's geometric shape is given. What about shapes that are not circular with arbitrary shape? The same principles of finding the velocity and strengths also apply to those objects, but it would be complicated to guess what combinations of flows constituting that specific shape. It would be beneficial to only need one kind of shape. The source panel method (SPM) is based on dividing a solid body into sets of panels. Each panel are assigned a source type flow. These panel source strengths make up a source sheet and can be represented by a function $\lambda(s)$ where s is the distance along the sheet and λ its corresponding strength value. For a point P located a distance r away from the small piece of the panel segment ds, an infinitesimal small potential $d\phi$ occurs at P. This can be written as:

$$d\phi = \frac{\lambda(s)ds}{2\pi}\ln r, \tag{6.5}$$

where $r = \sqrt{x^2 + y^2}$ is the distance from ds inducing a dϕ at point P. λ is the source strength. The full source sheet induced at P is found by integrating over the source sheet from a to b. The velocity potential can further be written as:

$$\phi = \int_a^b \frac{\lambda(s)ds}{2\pi}\ln r. \tag{6.6}$$

Earlier, it was stated that the source strength from a panel was $\lambda(s)$ which gave the equation stated above. Now, assuming that the source strength over a single panel is constant, we can allocate each panel with an unknown source strength coefficient $\lambda_1, \lambda_2, ..., \lambda_n$ yielding the possibility to approximate the source strengths at each panel. Each panel may have different source strengths λ and area located at the center of each panel (also called control point). For n panels, n source strengths occur.

The objective is to solve each individual panel strength λ_i, $i = 1, 2, 3, ..., n$. The boundary condition for the normal component states that the velocity is zero in that direction. Furthermore, the boundary condition at each *panel point* allows for a set of equations to be expressed. Panel j giving rise to a potential $\phi(P)$ from eq. (6.5) can be written as:

$$\Delta\phi_j = \frac{\lambda_j}{2\pi}\int_j \ln r_{pj} ds_j, \tag{6.7}$$

where the distance from point P to panel j is $r_{pj} = \sqrt{(x_p - x_j)^2 + (y_p - y_j)^2}$. Point $P = (x_p, y_p)$. λ_j is constant over the panel j and moved outside the integral. The total potential $\phi(P)$ induced at point P is found by summing over each panel:

$$\phi(P) = \sum_{j=1}^{n} \Delta\phi_j = \sum_{j=1}^{n} \frac{\lambda_j}{2\pi}\int_j \ln r_{pj} ds_j, \tag{6.8}$$

giving the expression for the potential $\phi(P)$ at a point P.

Now, move the point P to a panel i and define the point (x_i, y_i) to be the control point of panel i we can express the influence of every panel to the ith panel as:

$$\phi(x_i, y_i) = \sum_{j=1}^{n} \frac{\lambda_j}{2\pi} \int_j \ln r_{ij} ds_j, \tag{6.9}$$

with the distance $r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. Next, we derive the expression for the normal velocities at each panel using eq. (6.9) and the freestream velocity vector.

Normal velocity calculation

The normal component boundary condition is applied at the control point (x_c, y_c) which is located at the center of each panel (see fig. 6.3). The freestream velocity is needed to generate a flow around the object is needed to evaluate the correct orientation of the vectors. Let, $\vec{n_i}$ be the normal vector located the control point of panel i. The freestream vector V_∞ can be generated at an angle α relative to the x-axis, angle α is commonly known as the angle of attack. The vector normal to the freestream velocity vector at the control point at the ith panel can be expressed as:

$$V_{\infty,n} = \vec{V_\infty} \cdot \vec{n_i} = V_\infty \cos(\beta_i), \tag{6.10}$$

where β_i is the angle between $\vec{V_\infty}$ and $\vec{n_i}$. The normal velocity component V_n at panel i produced by every panel can be found by:

$$V_n = \frac{\partial}{\partial n_i}[\phi(x_i, y_i)]. \tag{6.11}$$

The derivative $\frac{\partial}{\partial n_i}$ acts on the distance variable r_{ij}. Note, when $j = i$ the calculation for the same panel is done on itself and the distance variable becomes zero. The term is now located in the denominator leading to a singularity. When $j = i$ the contribution has been shown to give $\frac{\lambda_i}{2}$. This gives an update of the potential equation being expressed as:

$$V_n = \frac{\lambda_i}{2} + \sum_{j=1,(j \neq i)}^{n} \frac{\lambda_j}{2\pi} \int_j \frac{\partial}{\partial n_i} \ln r_{ij} ds_j. \tag{6.12}$$

This the normal vector component of the source panel contribution at panel i. The boundary states:

$$V_{\infty,n} + V_n = 0, \tag{6.13}$$

as the flow cannot penetrate the walls of a solid body, the latter equation can be expressed as:

$$V_\infty \cos(\beta_i) + \frac{\lambda_i}{2} + \sum_{j=1,(j\neq i)}^{n} \frac{\lambda_j}{2\pi} \int_j \frac{\partial}{\partial n_i} \ln r_{ij} ds_j = 0, \tag{6.14}$$

substituting eq's. (6.10) and (6.12) into eq. (6.13). We get an equation with n unknowns. The task now is to gather n equations to solve for the source strengths. Solving the source strength coefficients gives rise to the streamlines over the set of panels together with the freestream velocity. This equation can be expressed in matrix form solving the system of equation using numerical methods. This approach makes it possible to control the amount of accuracy needed for the model as the number of panels is used as input to the model. Next, we define the geometry used for SPM and tackle the integral term in eq. 6.14.

Thorough deduction of source panel method

The previous section provided an overview of the equations for calculating the flow around an arbitrary body. In this section, a more detailed derivation will be done. The integral terms will be calculated together with the geometry of the problem. This will be set up into a matrix form $Ax = b$ which can be solved thus leading to a solution.

6.0.3 Defining geometry

Each panel can be of different length and therefore each panel needs to be calculated separately. The starting point needs to be defined and thereafter calculated each panels property in an orderly fashion. This is necessary because the normal vector orientation will be affected by this choice. The suggested method for calculating panels are clockwise. Below, fig. 6.2 shows 8 panels approximating a circle with boundary points.

$$dx = (x_2 - x_1), \tag{6.15}$$

$$dy = (y_2 - y_1), \tag{6.16}$$

$$s_1 = \sqrt{dx^2 + dy^2}. \tag{6.17}$$

Eq's. (6.15, 6.16, 6.17) for the ith panels have the notation:

$$dx_i = x_{i+1} - x_i, \tag{6.18}$$

$$dy_i = y_{i+1} - y_i, \tag{6.19}$$

$$s_i = \sqrt{dx_i^2 + dy_i^2}, \tag{6.20}$$

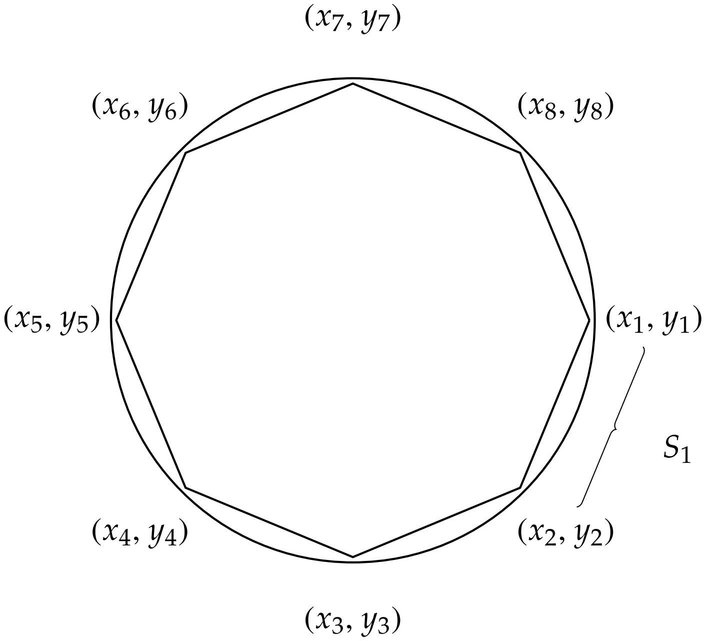


Figure 6.2: Approximating a circle with panels. n panels are created with boundary points representing the end points of each panel (x_b, y_b). From the figure, 8 panels s are created with 9 boundary points ((x_1, y_1) are counted twice for the first and last boundary point).

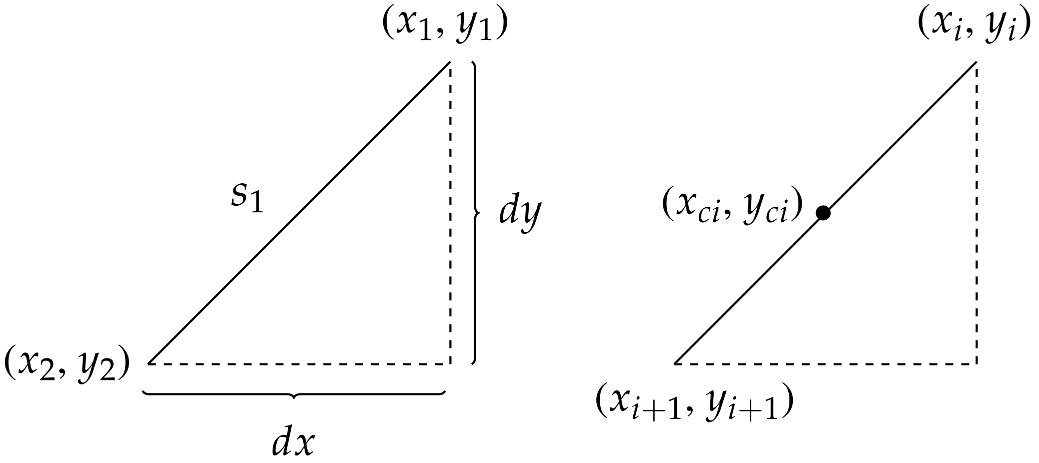


Figure 6.3: (left) a closer view of panel S_1 from fig. 6.2. The equations needed for the panel calculation can be seen above. (right) The general case for the ith panel with corresponding boundary points and control points.With the boundary points defined, the control points can be calculated by eq's. (6.21) and (6.22).

where $i = 1, 2, 3..., n$ goes up to the number of panels n. Furthermore, from fig. (6.3) the control points for each panels is needed. This is where the source strength is set up. This point has the relation:

$$x_{ci} = \frac{x_i + x_{i+1}}{2}, \tag{6.21}$$

$$y_{ci} = \frac{y_i + y_{i+1}}{2}. \tag{6.22}$$

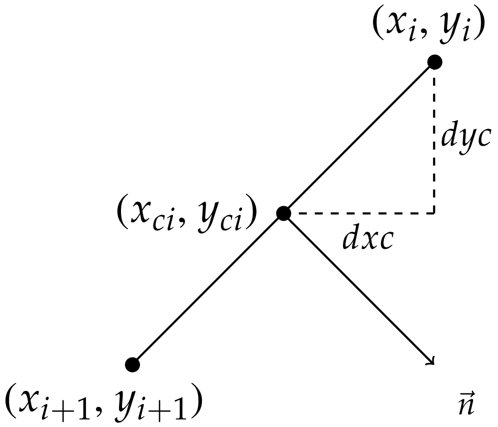


Figure 6.4: In this figure a normal vector $\vec{n}$ is added at the control point together with the change in x and y direction from the control point (dxc and dyc) which is necessary when calculating angles between panels and normal vectors.

The angle ϕ between the panel and x-axis can be found by:

$$\tan \phi_i = \frac{dy}{dx}. \tag{6.23}$$

Note, the angle ϕ_i does not change for a different placement of the x-axis. The angles δ_i and β_i are the angles from the x-axis to the normal vector, and from the freestream vector to the normal vector, respectively. They are expressed as:

$$\delta_i = \phi_i + \frac{\pi}{2}, \tag{6.24}$$

$$\beta_i = \delta_i - \alpha, \tag{6.25}$$

where α is the angle of attack from the freestream vectors. The angles are visualized in fig. 6.5. The normal vector components can be expressed as:

$$n_{xi} = x_{ci} + s_i \cos \delta_i, \tag{6.26}$$

$$n_{yi} = y_{ci} + s_i \sin \delta_i, \tag{6.27}$$

$$\vec{n} = n_{xi}\vec{i} + n_{yi}\vec{j}. \tag{6.28}$$

$\vec{i}$ and $\vec{j}$ are unit vectors in x and y - direction, respectively.

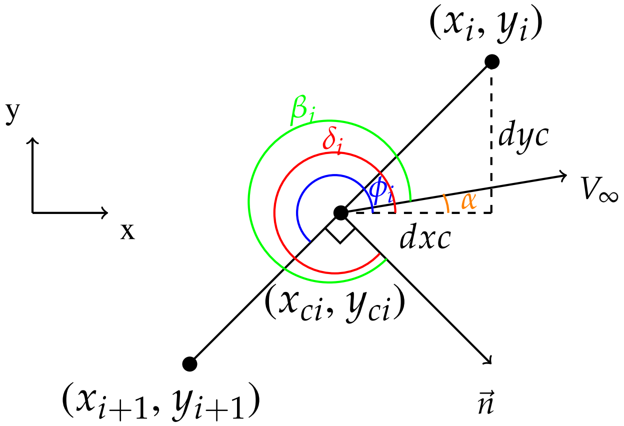


Figure 6.5: This figure shows the angles needed to compute the normal vectors of each panel derived from the boundary and control points. Going in clockwise direction with boundary points defined at (x_i, y_i) and (x_{i+1}, y_{i+1}). The control point (x_{ci}, y_{ci}) on the ith panel is located at the panels center.

Building complex flows

Suppose now that a curve with infinite sources located on it's arc. The expression for calculating the velocity potential at a point is given by the integral:

$$\phi_p = \int_a^b \frac{\lambda(s)}{2\pi} \ln r_p(s) ds \tag{6.29}$$

The integral assumes the source strength over the body' periphery (see figure 6.6) depends on the distance s along the surface S. By breaking the the surface up into straight lines the integral above can be manipulated into the equation:

$$\phi_p = \int_a^b \frac{\lambda(s_{ab})}{2\pi} \ln r_p(s_{ab}) ds_{ab} + \int_b^c \frac{\lambda(s_{bc})}{2\pi} \ln r_p(s_{bc}) ds_{bc} \tag{6.30}$$

Figure 6.7 depicts this integral case. The curve now consists of two lines where each line contains it's own source strength $\lambda(s_{ab})$ and $\lambda(s_{bc})$. The number of panels can be adjusted. For n panels the integral reads:

$$\phi_p = \sum_{j=1}^{n} \int_j \frac{\lambda(s_j)}{2\pi} \ln r_{pj}(s_j) ds_j \tag{6.31}$$

Now assume the source strength $\lambda(s)$ is constant along each panel. This leads to the source strength at panel j being denoted as $\lambda(s_j) = \lambda_j$ and the distance $r_{pj}(s_j) = r_{pj}$. We then write the integral equation above as:

$$\phi_p = \sum_{j=1}^{n} \lambda_j \int_j \frac{\ln r_{pj}}{2\pi} ds_j \tag{6.32}$$

Fig's. (6.6 - 6.8) show a step by step visualization of the approximations done above.

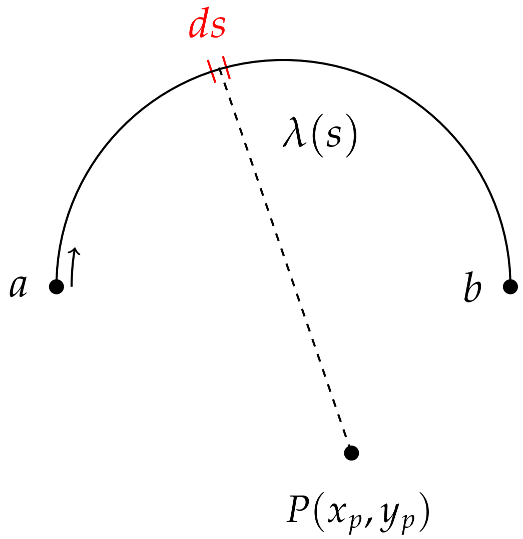


Figure 6.6: A part of a solid body (black line) with a source strength $\lambda(s)$ is shown here. The body induces a small potential $d\phi$ from ds at point P. Integrating from a to b grants the total contribution of the potential ϕ along the periphery of the surface. The arrow indicates the integration direction.

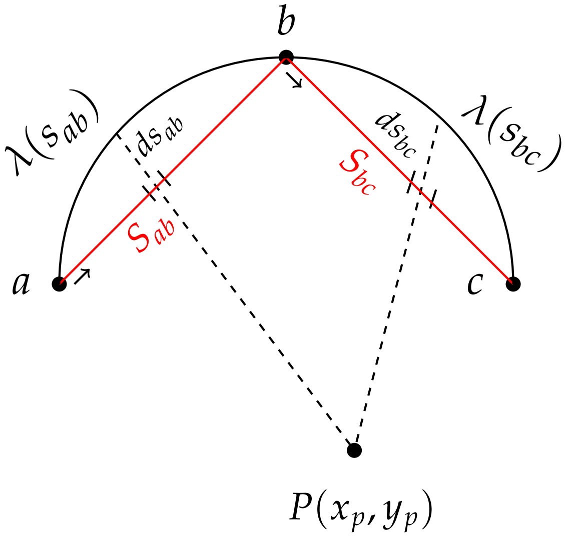


Figure 6.7: Here, we show the next step of approximating the geometry a body. By splitting the body into panels (red) gives two source strengths from $a - b$ and $b - c$ on the body. S_{ab} and S_{bc} are the panel lengths. A panel set of approximating the body is shown. More points, leading to more panels can be added for a better approximation.

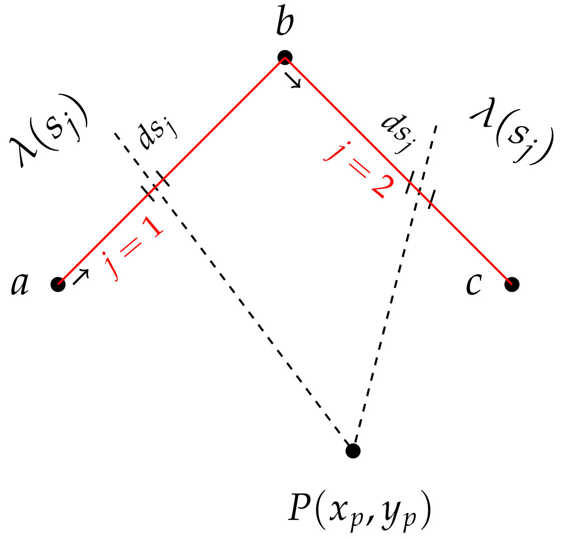


Figure 6.8: Here, we have assigned each panel (red) a constant source strength $\lambda(S_j)$ for $j = 1, 2, 3, ..., n$, leading to n source strengths for n panels. S_j is the jth panel length.

To sum up, the goal is to build a flow around arbitrary objects. The geometry was defined using the boundary and control points which are known values. The angles are needed when calculating the velocities and solving the integral term from eq. (6.14). This will be done in the next section. We found that a point object induces a small potential ϕ at a point P in space. Further, we found an expression of the total potential $\phi(P)$ at point P. Assuming the source strength parameter λ is constant along each panel, allowed us to brake up an arbitrary shape into n panels consisting of straight lines. The source strengths λ are located at the control points of each panel. Combining this with a freestream velocity, we have all the contributions needed in order to calculate the flow around an object of arbitrary shape. In the next section we dive further into the details how to set up the equations needed for solving the panel source strengths and the velocities.

6.0.4 Calculating the velocities around an arbitrary shaped object

At the surface of an object the normal velocity component of a flow is zero. This condition will be used to set up a system of equations. Further, to calculate the velocities the tangential component will be used. Finally the potential can be solved at every grid cell for the defined area of interest.

$$\phi_p = V_\infty \cos(\alpha)x + V_\infty \sin(\alpha)y + \sum_{j=1}^{n} \lambda_j \int_j \frac{\ln r_{pj}}{2\pi} ds_j. \tag{6.33}$$

The two first terms are the freestream velocity vectors in x and y direction respectively. Taking the derivative of the potential ϕ_p in the x and y direction the velocities can be found:

$$V_{x,p} = \frac{\partial \phi_p}{\partial x}, \tag{6.34}$$

$$V_{y,p} = \frac{\partial \phi_p}{\partial y}. \tag{6.35}$$

The normal and tangential velocity deal with the physical part of the solution. They can be expressed as:

$$\vec{V} \cdot \vec{n} = 0. \tag{6.36}$$

No fluid can pass through objects

With the source strengths λ placed at the control points of each panel the normal velocity component will be used to solve for the source strengths. The velocity potential at panel i can be written using eq (6.33):

$$\phi_p = V_\infty \cos(\alpha)x + V_\infty \sin(\alpha)y + \sum_{j=1}^{n} \lambda_j \int_j \frac{\ln r_{ij}}{2\pi} ds_j, \tag{6.37}$$

where we sum up the contribution of every source strength λ_j induced at panel i. $i = 1, 2, 3, ..., n$ and $j = 1, 2, 3, ..., n$ run up to the number of panels. The normal and tangential velocities at panel i can be expressed from the relation between the velocity potential as:

$$V_{n,i}^{Total} = \frac{\partial \phi_i}{\partial n_i} = V_\infty \cos(\alpha) \frac{\partial x_i}{\partial n_i} + V_\infty \sin(\alpha) \frac{\partial y_i}{\partial n_i} + \sum_{j=1}^{n} \frac{\lambda_j}{2\pi} \int_j \frac{\partial}{\partial n_i} \ln r_{ij} ds_j = 0. \quad (6.38)$$

By looking at the first two terms in the normal direction. The freestream velocity components are denoted:

$$V_{n,i}^{freestream} = \frac{\partial \phi_i}{\partial n_i} = V_\infty \cos(\alpha) \frac{\partial x_i}{\partial n_i} + V_\infty \sin(\alpha) \frac{\partial y_i}{\partial n_i}. \quad (6.39)$$

The change in x-direction with respect to the normal derivatives can be written as:

$$\frac{\partial x_i}{\partial n_i} = \cos(\delta_i) \quad , \quad \frac{\partial y_i}{\partial n_i} = \sin(\delta_i). \quad (6.40)$$

The setup can seen in figure 6.9. Inserting eq (6.39) into eq (6.40) and using the trig identity $\cos(a \pm b) = \cos(a)\cos(b) \mp \sin(a)\sin(b)$ we end up with simpler term for the freestream velocity:

$$V_{n,i}^{freestream} = V_\infty \cos(\beta_i). \quad (6.41)$$

Recall, β_i is the angle from angle of attack α to δ_i $\beta_i = \delta_i - \alpha$. See figure 6.5. Putting eq (6.41) into eq (6.38):

$$V_{n,i}^{Total} = \frac{\partial \phi_i}{\partial n_i} = V_\infty \cos(\beta_i) + \sum_{j=1}^{n} \frac{\lambda_j}{2\pi} \int_j \frac{\partial}{\partial n_i} \ln r_{ij} ds_j = 0. \quad (6.42)$$

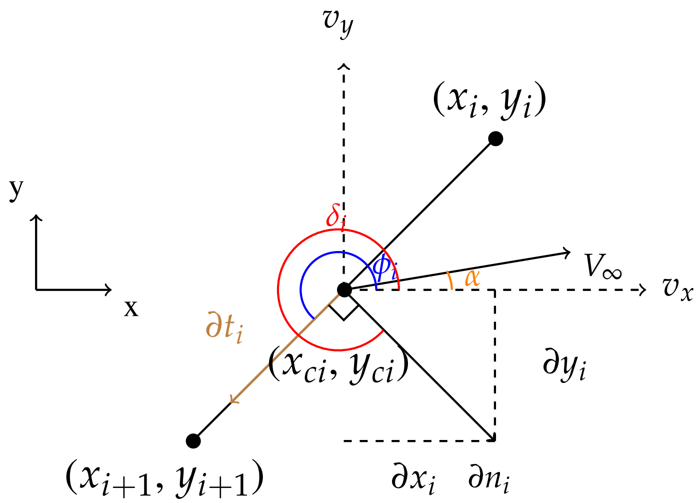


Figure 6.9: Panel i has both tangential and normal components. These two quantities can be expressed with trigonometric functions.

6.0.5 Normal vector geometry integral

In this section we look at the integral part of the total normal velocity induced at panel i. Recall:

$$V_{n,i}^{Total} = \frac{\partial \phi_i}{\partial n_i} = V_\infty \cos(\beta_i) + \sum_{j=1}^{n} \frac{\lambda_j}{2\pi} \int_j \frac{\partial}{\partial n_i} \ln r_{ij} ds_j = 0. \tag{6.43}$$

Let's put our attention at the integral that needs to be solved and express the terms with the use of variables mentioned above. Let the integral be denoted by I_{ij}:

$$I_{ij} = \int_j \frac{\partial}{\partial n_i} \ln r_{ij} ds_j. \tag{6.44}$$

The logarithmic term can be manipulated by the chain rule $\frac{d \ln(f(x))}{dx} = \frac{1}{f(x)} \frac{df(x)}{dx}$. Applying this to eq (6.44) gives:

$$I_{ij} = \int_j \frac{1}{r_{ij}} \frac{\partial r_{ij}}{\partial n_i} ds_j. \tag{6.45}$$

Also, recall the distance r_{ij} which is the distance between panel j to panel i:

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{6.46}$$

Taking the derivative of the distance w.r.t to normal component yields the equation:

$$\frac{\partial r_{ij}}{\partial n_i} = \frac{(x_i - x_j)\frac{\partial x_i}{\partial n_i} + (y_i - y_j)\frac{\partial y_i}{\partial n_i}}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}. \tag{6.47}$$

From fig. (6.9) and the eq's. in (6.40) we recognize the derivatives in the latter equation as $\cos{(\delta_i)} = \frac{\partial x_i}{\partial n_i}$ and $\sin{(\delta_i)} = \frac{\partial y_i}{\partial n_i}$. Substituting eq's. (6.47):

$$\frac{\partial r_{ij}}{\partial n_i} = \frac{(x_i - x_j)\cos(\delta_i) + (y_i - y_j)\sin(\delta_i)}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}. \tag{6.48}$$

Gathering eq. (6.48), eq. (6.46) and inserting this into eq. (6.45) the equation now reads:

$$I_{ij} = \int_j \frac{(x_i - x_j)\cos(\delta_i) + (y_i - y_j)\sin(\delta_i)}{(x_i - x_j)^2 + (y_i - y_j)^2} ds_j. \tag{6.49}$$

Further, we want to express the integrand in terms of s_j as this is the integration variable. We find that x_j and y_j

$$x_j = X_j + s_j\cos(\phi_j)\ , \quad y_j = Y_j + s_j\sin(\phi_j). \tag{6.50}$$

Here, we represent the point on the jth panel as a parameter representation. X_j and Y_j are starting points for the panel, and s_j is the length of the jth panel. ϕ_j is angle between the x-axis and the panel j. The angle ϕ_j can be derived from the identity $\cos(\delta_i) = -\sin(\phi_i)$ and $\sin(\delta_i) = \cos(\phi_i)$ where the relation of δ_i and ϕ_i are $\delta_i = \phi_i + \pi/2$ as seen in figure 11. Inserting this into I_{ij} the integral now reads:

$$I_{ij} = \int_j \frac{(x_i - x_j)(-\sin(\phi_i)) + (y_i - y_j)\cos(\phi_i)}{(x_i - x_j)^2 + (y_i - y_j)^2} ds_j. \tag{6.51}$$

For the sake tidyness, the numerator will be considered first. Thereafter, the denominator will be considered. From the above equation the numerator reads:

$$(x_i - X_j - s_j\cos(\phi_j)(-\sin(\phi_i)) + (y_i - Y_j - s_j\sin(\phi_i)\cos(\phi_i)), \tag{6.52}$$

after inserting eq (6.50). Further we arrange the terms such that:

$$(X_j - x_i)\sin{(\phi_i)} + (y_i - Y_j)\cos{(\phi_i)} + s_j(\cos{(\phi_j)}\sin{(\phi_i)} - \sin{(\phi_j)}\sin{(\phi_i)}), \tag{6.53}$$

and use the trig identity $\sin a - b = \sin a \cos b - \cos a \sin b$ on the last term:

$$(X_j - x_i)\sin(\phi_i) + (y_i - Y_j)\cos(\phi_i) + s_j \sin(\phi_i - \phi_j). \tag{6.54}$$

At last we write the numerator in terms of s_j as this is the variable which will be integrated over:

$$s_j A + D, \tag{6.55}$$

where $A = \sin(\phi_i - \phi_j)$ $D = (X_j - x_i)\sin(\phi_i) + (y_i - Y_j)\cos(\phi_i)$. The numerator is now simplified and can be integrated.

Moving on the denominator. Similar approach will used as with the numerator. From eq (6.49) write out the terms and insert eq's (6.50):

$$x_i^2 - 2x_i(X_j + s_j\cos\phi_j) + (X_j + s_j\cos\phi_j)^2 + y_i^2 - 2y_i(Y_j + s_j\sin\phi_j) + (Y_j + s_j\sin(\phi_j)^2. \tag{6.56}$$

Now gather all s_j terms with same polynomial degree. This leads to:

$$s_j^2 + 2s_jB + C, \tag{6.57}$$

where $B = X_j\cos\phi_j + Y_j\sin\phi_j - x_i\cos\phi_j - y_i\sin\phi_j$ and $C = (x_i - X_j)^2 + (y_i - Y_j)^2$. The end term with the numerator and denominator put back into eq(6.44) we finally end up with:

$$I_{ij} = \int_0^{S_j} \frac{s_jA + D}{s_j^2 + 2s_jB + C} ds_j. \tag{6.58}$$

Before integrating we need to complete the square in the denominator.

$$s_j^2 = 2Bs_j + C = s_j^2 + 2Bs_j + B^2 + C - B^2 = (s_j + B)^2 + E^2, \tag{6.59}$$

where $E = \sqrt{C - B^2}$. Now the integral reads:

$$I_{ij} = \int_0^{S_j} \frac{s_jA + D}{(s_j + B)^2 + E^2} ds_j, \tag{6.60}$$

which is now on the form that can be integrated. Setting $u = s_j + B$ leads to the expression:

$$I_{ij} = \int_0^{S_j} \frac{(u - B)A + D}{u^2 + E^2} du. \tag{6.61}$$

The integral will be split up into two integrals by separating the numerator.

$$I_{ij} = A\int_0^{S_j} \frac{u}{u^2 + E^2} du + (D - AB)\int_0^{S_j} \frac{1}{u^2 + E^2} du. \tag{6.62}$$

Using substitution $\gamma = u^2 + E^2$ on the first integral again we obtain the solution as:

$$A \int_0^{S_j} \frac{u}{u^2 + E^2} du = \frac{A}{2} \int_0^{S_j} \frac{d\gamma}{\gamma} = \frac{A}{2} [\ln \gamma]_0^{s_j}. \tag{6.63}$$

Now substitute u and γ

$$\frac{A}{2} \Big[\ln (S_j + B^2 + E^2) - \ln (B^2 + E^2) \Big], \tag{6.64}$$

$$\frac{A}{2} \ln \left[\frac{(S_j + B)^2 + E^2}{B^2 + E^2} \right] = \frac{A}{2} \ln \left[\frac{S_j^2 + 2S_jB + B^2 + E^2}{B^2 + E^2} \right]. \tag{6.65}$$

With the final touch of using $E^2 = C - B^2$ we end up with:

$$\frac{A}{2} \ln \left[\frac{S_j + 2S_jB + C}{C} \right]. \tag{6.66}$$

The second integral in eq (6.62) has the solution:

$$(D - AB) \int_0^{S_j} \frac{1}{u^2 + E^2} du = \frac{(D - AB)}{E} \arctan \left[\frac{u}{E} \right], \tag{6.67}$$

as with the first integral we substitute back and end up with:

$$\frac{D - AB}{E} \left(\arctan \left[\frac{S_j + B}{E} \right] - \arctan \left[\frac{B}{E} \right] \right). \tag{6.68}$$

Let's sum up the most important parts of this integration below. The integration gave the solution:

$$I_{ij} = \frac{A}{2} \ln \left[\frac{S_j + 2S_jB + C}{C} \right] + \frac{D - AB}{E} \left(\arctan \left[\frac{S_j + B}{E} \right] - \arctan \left[\frac{B}{E} \right] \right). \tag{6.69}$$

The terms A, B, C, D and E are:

$$\begin{gathered}
A = \sin (\phi_i - \phi_j), \\
B = (X_j - x_i) \cos \phi_j + (Y_j - y_i) \sin \phi_j, \\
C = (x_i - X_j)^2 + (y_i - Y_j)^2, \\
D = (X_j - x_i) \sin \phi_i + (y_i - Y_j) \cos \phi_i, \\
E = \sqrt{C - B^2}.
\end{gathered}$$

Finally after an amount of mathematical operations the solution I_{ij} is found which means the normal velocity component can be computed by:

$$V_{n,i} = V_\infty \cos\beta_i + \sum_{j=1}^{n} \frac{\lambda_j}{2\pi} I_{ij} \tag{6.70}$$

This sums up the normal vector computations.

The normal and tangential geometry integral have been expressed in terms of panel s_j thus making it possible to integrate and finding the contributions of every panel λ_j induced onto panel i. Next we express the equations in a matrix.

6.0.6 Solving the system of equations

With the I_{ij} integral solved, the next part is to solve for source strength at each panel. Recall the normal component being equal to zero at every control point (x_{ci}, y_{ci}), $i = 1, 2, 3..., n$.

$$V_{n,i}^{Total} = V_\infty \cos(\beta_i) + \sum_{j=1}^{n} \frac{\lambda_j}{2\pi} I_{ij} = 0 \tag{6.71}$$

$$\sum_{j=1}^{n} \frac{\lambda_j}{2\pi} I_{ij} = -V_\infty \cos(\beta_i) \tag{6.72}$$

When $j = i$ the I_{ij} has been found to be $\frac{\lambda}{2}$. The equations above now reads:

$$\frac{\lambda_i}{2} + \sum_{\substack{j \neq i \\ j=1}}^{n} \frac{\lambda_j}{2\pi} I_{ij} = -V_\infty \cos(\beta_i) \tag{6.73}$$

Eq (6.71) can further be inserting into a matrix representing a set of linear equations.

$$\begin{bmatrix} \pi & I_{12} & I_{13} & \dots & I_{1n} \\ I_{21} & \pi & \dots & \dots & I_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I_{n1} & I_{n2} & \dots & \dots & \pi \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \vdots \\ \lambda_n \end{bmatrix} = -2\pi V_\infty \begin{bmatrix} \cos\beta_1 \\ \cos\beta_2 \\ \cos\beta_3 \\ \vdots \\ \cos\beta_n \end{bmatrix} \tag{6.74}$$

Multiplying the matrix equation with 2π simplifies the equation. The set of equation in matrix form grants the source strengths of all n panels which is solved numerically. This approximation can be adjusted by increasing the amounts of panels making the result more accurate. The source strength distribution which is now solved makes it possible to solve for the remaining flow in the defined area.

6.0.7 Streamline geometry integral

The geometry integral part of the normal, tangetial and streamline components have a lot in common, but the main idea is to make it clear which equations are needed and in what order to obtain streamlines around a body. Here, with the sources strengths solved for, the velocities at every grid cell can be calculated. Using eq (6.33) which was the velocity potential induced at a point P and take the derivatives in x and y direction the velocity component are found eq (6.34) and eq (6.35), respectively. The integrand is similar to the ones above and is carried out in the same manner. Note, the only change is the variable in which to take the derivative. The integral has the solution:

$$V_{x,p} = V_\infty \cos(\alpha) + \sum_{j=1}^{n} \frac{\lambda_j}{2\pi} \left[\frac{-\cos\phi_j}{2} \ln\left[\frac{S_j + 2S_jB + C}{C} \right] + \frac{A_{xp}}{E} \left(\text{arctan}\left[\frac{S_j + B}{E} \right] - \text{arctan}\left[\frac{B}{E} \right] \right) \tag{6.75}$$

where $A_{xp} = x_p - X_j + \cos\phi_j B$

$$V_{y,p} = V_\infty \sin(\alpha) + \sum_{j=1}^{n} \frac{\lambda_j}{2\pi} \left[\frac{-\sin\phi_j}{2} \ln\left[\frac{S_j + 2S_jB + C}{C} \right] + \frac{A_{yp}}{E} \left(\text{arctan}\left[\frac{S_j + B}{E} \right] - \text{arctan}\left[\frac{B}{E} \right] \right) \right] \tag{6.76}$$

where $A_{yp} = y_p - Y_j + \sin\phi_j B$.

Solving the integrals leads to the streamlines around a solid body with arbitrary shape. To sum up, a numerical method will have to be relied on. Analytic solutions of the Laplace equation around arbitrary bodies can in general not be done and the requirements for the source panel method is knowing the shape of the body which is approximated by a set of panels (piecewise straight line segments) which can be varied for more accurate results. At the center of each panel a small source strength flow is placed. The velocity components at the center of the panel are equipped with boundary conditions on the normal and tangential velocity. The boundary conditions gives rise to two geometric integrals where the source strength values are needed. The normal geometric is first solved finding the source strength values at the center of each panel. The tangential can then be used to find the flows at the center. At last, the contribution of panel give rise to the remaining velocities to be solved in the area.

Chapter 7

Map projections

The knowledge of the earth's shape has been known by cartographers for the past 2000 years (Snyder 1987). When traveling far it was necessary to know where and distance of travel. How to measure distances on the globe has been needed for many years when people needed to navigate from one place to the other.

This is mentioned here, as the data we will be working with are registered as longitude and latitude coordinates. A short motivation why these quantities are used on maps will be mentioned in the coming sections. Additionally, how to convert a spherical surface onto a rectangular surface will be the focus in the following section.

7.0.1 Geographic coordinate system

A coordinate system is useful tool when working with positional data. This gives insight distances in multiple dimensions and also from the use of mathematics grants the possibility of calculating positions from a given reference point or relative to given points.

When determining positions on a spherical shaped object e.g. the earth, one must take into account the curvature and how distances are measured. The measures of longitude and latitude have been implemented and defined in such a way to make a structured way to split up the globe. These are angular measurements. There are several ways of defining latitude and longitude: geocentric , astronomical and geographic. The latter is most widely used.

Latitude, also known parallels are measured from north to south direction with an interval of 180 °. Equator is located at 0 °. The earth is not a perfect sphere. This affects the relative lengths of latitudes. 1 °of latitude amounts to 110.567km at Equator and 111.699

km at the poles (Encyclopedia britannica 2020).

Longitude, also known as meridians are measured from west to east with an interval of 180 °. The prime meridian, which is where 0 °is defined, runs through Greenwich, London The relative distances are equal for 1 °of longitude and equals 111.32 km (Encyclopedia britannica 2020). Longitude and latitude coordinates make up a geographic coordinate system. Positional information can be determined in terms of the prime meridian and the Equator. See figure below:

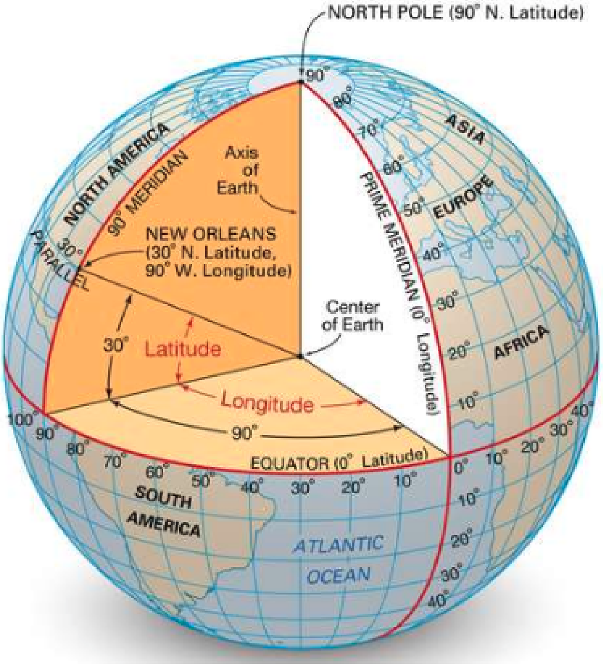


Figure 7.1: Here we see how each angle of longitude and latitude is defined from the earth's center. Figure gathered from (Encyclopedia britannica 2020).

7.0.2 Converting geographical coordinates to a Cartesian coordinate system

The longitude and latitude values are collected from the earth's surface. The standard reference system for geospatial information is World Geodetic System (WGS84). The Global Positioning System uses this system.

Since the SPM is defined in a two dimensional coordinate system, a conversion from longitude- latitude to Cartesian coordinates is necessary.

$$x = 111412.84\cos(\phi) - 93.5\cos(3\phi) + 0.118\cos(5\phi) \tag{7.1}$$

$$y = 111132.92 - 559.82\cos(2\phi) + 1.175\cos(4\phi) - 0.0023\cos(6\phi) \tag{7.2}$$

x and y are measured in meters m, ϕ are latitude values represented in radians. Important that the latitude values are in radians which is common for most programming languages.

When calculating lengths on a sphere, specifically the earths sphere the haversine formula can be applied. The Haversine formula calculates the distance the between two points. The points have longitude and latitude values.

$$a = \sin^2\left(\frac{lat_1 - lat_0}{2}\right) + \cos\left(lat0\right)\cos\left(lat1\right)\sin^2\left(\frac{lon_1 - lon_0}{2}\right) \tag{7.3}$$

$$c = 2\arcsin\left(\sqrt{a}\right) \tag{7.4}$$

$$Haverdist = cr \tag{7.5}$$

The points have coordinates (lon_0, lat_0) and (lon_1, lat_1), r is the earth's radius.

7.0.3 Map selection

The data used in the thesis are logged from the surface of a sphere i.e. the earth. When it comes time to visualize the data, this is done on a flat surface. Some background is needed in order to find the appropriate representation. This section presents an introduction to maps and the different ways to represent maps in two dimensions. As we know the earth has a spherical shape, but in geography class the earth was usually shown to us on a two dimensional map. This is known as a map projection. If you tried to flatten the earth onto a rectangular plane also trying to maintain properties such as geometry, area size and distance you would not succeed (see figure 7.2). An example is trying to peel an orange and make it's shape rectangular. Converting from three to two dimensions leads to distortion (Snyder 1987).

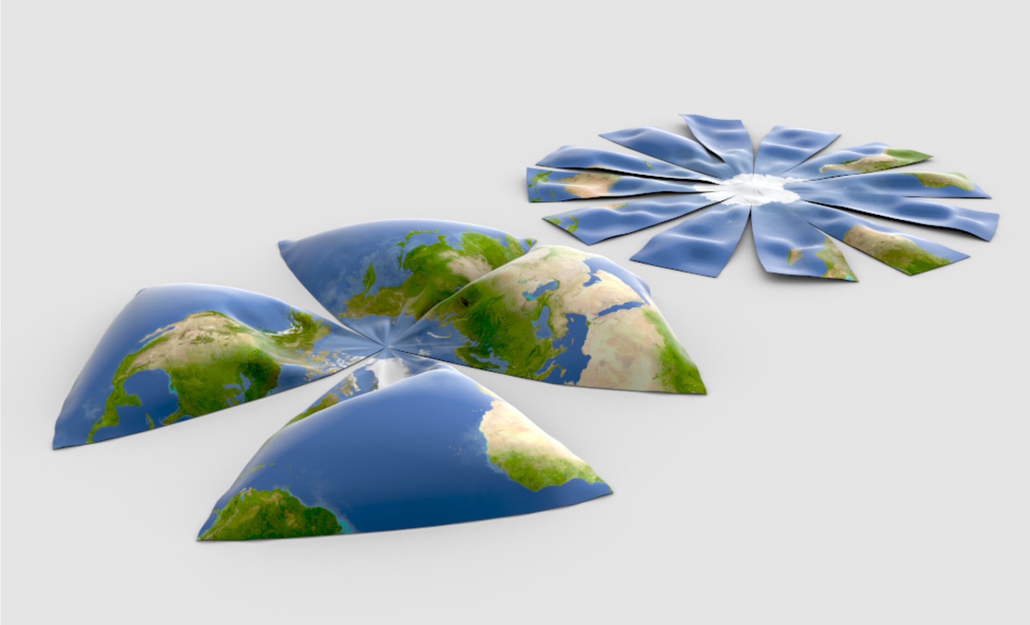


Figure 7.2: Here, the earth flattened out without any distortion. Linking together the pieces comes at the price of distortion. Figure gathered from (Šavrič and Kennedy, Melita 2020).

7.0.4 Projections

When deciding which projection to use, it is useful to consider what it is most important quantity to preserve. Below we state some projection properties which are commonly considered:

Equal-area preserves the area-size of objects. This is useful when wanting to see the relative size of multiple objects.

Conformal preserves the angle between points. This is important when navigating at sea and when using large- scaled maps.

Equidistant maintains the distance between points. Preserving this quantity comes in handy when wanting to compare several distances.

Compromise considers some distortion from all properties, but attempting to balancing it.

There are three types of methods commonly used when it comes to map projection: cylinder, cone and plane.

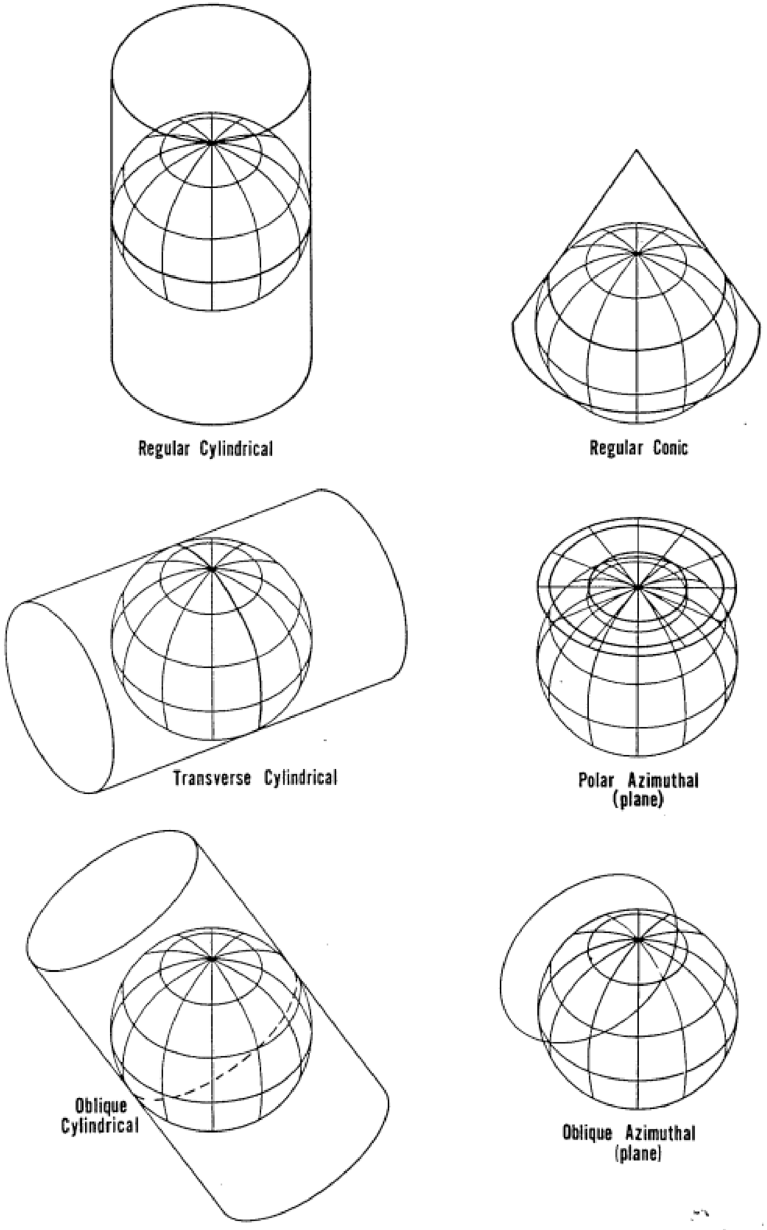


Figure 7.3: Here we see how the different projections are made from a spherical surface to a flat surface. Figure gathered from (Snyder 1987).

Chapter 8

AIS data

A concise introduction of the data used in the thesis is presented. We list up what is contained within an AIS message. A motivation as to why AIS data was introduced in the first place is mentioned. However, its limitations needs to be specified in order to gain insight in the work needed to process and use the data. Autonomous Identification System data, for short AIS-data, is a mandatory self-reporting system installed on maritime vessels originally developed to avoid collisions. It sends out information about a vessels characteristics in real time. The transmitter the AIS system uses to communicate is based on Very High Frequency (VHF) bands. The receiver on the other end, typically other ships within a vicinity or vessel traffic services (VTS), use the information for safety measures i.e. avoiding collisions. Additional receivers include; buoys and satellites (S-AIS). The data is divided in two types: static and dynamic. Static remain the same after being registered for the first time, dynamic can change and is updated. AIS characteristics are listed in the table below.

AIS characteristics		
Variables:	**Specifications:**	**Data type:**
IMO number	International Maritime Organization identification system	Static
ShipType	The vessels classified ship type	Static
SOG	Speed over ground	Dynamic
Longitude	Geographic coordinate horizontal	Dynamic
Latitude	Geographic coordinate vertical	Dynamic
COG	Course over ground/heading	Dynamic
Time Stamp Position	Time of logged data	Dynamic

The development of AIS-data come after the environmental catastrophe of the Exxon Valdez oil spillage in 1989 [1]. In 2004 the International Maritime Organization (IMO) made it mandatory for all vessels with a weight over 300 gross tons to install a AIS transceiver [2]. The huge amounts of data being logged has made analyzing AIS-data a useful tool within the maritime industry. Being able to predict arrival and waiting times at ports, turnaround times and make accurate decisions in advance regarding risk assessment is a benefit for the parties involved.

The transmitter or receiver could cause errors leading to an AIS message or several not being registered. This causes the data set to contain gaps in the time and spatial parameters between the AIS messages. Moreover, a satellite is able to cover a larger area than other AIS receivers which can lead to interference between ships (Smestad, Asbjørnslett, and Rødseth 2017).

In this thesis, an AIS- data set limited to a time window and restricted to an area around the Trondheimsfjord was provided by DNVGL extraction within a specified area will be made of the AIS-data. The AIS-data will help gain insight and perspective on the route

[1] https://www.marineinsight.com/maritime-history/the-complete-story-of-the-exxon-valdez-oil-spill/

[2] http://www.imo.org/en/OurWork/safety/navigation/pages/ais.aspx

done by vessels as this is actual route done by vessels. As AIS-data is installed by many different vessel types, an interesting view is to view what vessel types take which route depending on the environment. This could help with significance of the fluid mechanical model comparing routes.

8.0.1 Preprocessing AIS-data

AIS-data provides several variables at a given timestamp. These variables are:

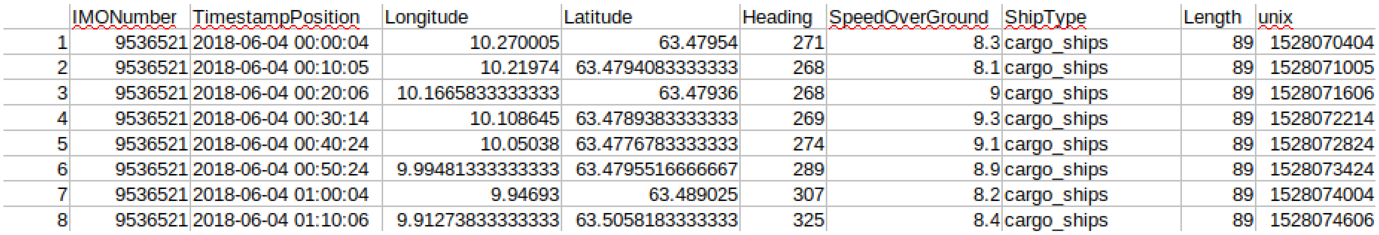
- **International Maritime Organization (IMO) number:** Introduced in 1987 to enhance ”maritime safety, and pollution prevention to facilitate the prevention of maritime fraud” [3]. Each vessel gets assigned a unique number used for identifying vessels. Vessels over 100 gross tonns are required to carry an IMO number with the exceptions to vessels engaged solely in fishing, vessels without me- chanical means of propulsion, pleasure yachts, vessels engaged on special service, hipper barges, hydrofoils and hovercrafts, floating decks, vessels of war and wooden vessels (Aarsnes 2018).
- **Longitude and latitude:** make up vessel positions in a geographic coordinate system. Their units are angles and make up unique points on a sphere. In a cartesian coordinate system the longitude and latitude needs to be converted in a x and y coordinates for further computations.
- **Heading:** is based on directions on a compass. Used for finding vessels' course measured in angles where 0°and 360 °is true north, 90 °true east, 180 °true south and 270 °true west.
- **Speed Over Ground (SOG):** is a vessel speed in magnitude relative to the earth's surface. SOG is measure in knots or(1.852 km/h).
- **Ship type:** specifies the ships characteristic e.g. cargo ship, passenger ship, tug boat, tank ships, coast guard ship, pleasure craft, high-speed craft, towing vessel, fishing boats, sailing vessels, ekranoplans. If the vessel is not of this ”other” is registered.
- **Length:** the vessels maximum length.
- **Time stamp position:** is the time when data was registered. Measured with an accuracy up to seconds.

[3] http://www.imo.org/en/ourwork/msas/pages/imo-identification-number-scheme.aspx

Data structure

The key elements in a tidy data set is being consistent, easy to access and work with and exploit. Another element to take into consideration is how the computer will be able to extract values. This will help when it comes to analyzing the data (Wickham 2014).

Each row represent a observation and each column represent a variable. According to(Wickham 2014) this is the preferred method in organizing data. Below is an image of how the AIS-data is organized from file:



	IMONumber	TimestampPosition	Longitude	Latitude	Heading	SpeedOverGround	ShipType	Length	unix
1	9536521	2018-06-04 00:00:04	10.270005	63.47954	271	8.3	cargo_ships	89	1528070404
2	9536521	2018-06-04 00:10:05	10.21974	63.4794083333333	268	8.1	cargo_ships	89	1528071005
3	9536521	2018-06-04 00:20:06	10.1665833333333	63.47936	268	9	cargo_ships	89	1528071606
4	9536521	2018-06-04 00:30:14	10.108645	63.4789383333333	269	9.3	cargo_ships	89	1528072214
5	9536521	2018-06-04 00:40:24	10.05038	63.4776783333333	274	9.1	cargo_ships	89	1528072824
6	9536521	2018-06-04 00:50:24	9.99481333333333	63.4795516666667	289	8.9	cargo_ships	89	1528073424
7	9536521	2018-06-04 01:00:04	9.94693	63.489025	307	8.2	cargo_ships	89	1528074004
8	9536521	2018-06-04 01:10:06	9.91273833333333	63.5058183333333	325	8.4	cargo_ships	89	1528074606

Figure 8.1: The first 8 rows of the data set is shown. Each observation is a row number with columns representing the vessel' characteristic/information at a given time.

As seen from figure (8.1) the structure of the data is up to standard with the guide lines from (Wickham 2014). Next, we calculate the velocity components of each observation by using 2 columns from the data set.

Calculating velocities

Velocities are useful units to calculate. This grants the possibility to visualize a vessels direction and heading. Here, we will see which vessels travel to and from Trondheim. Vessels may take detour, traveling through Trondheim to then converge to Trondheim after a while. From figure (8.1) the velocities can be found by using columns: heading and speed over ground. The velocities can be found with these two equations:

$$v_{lon} = SOG \cdot \sin(heading) \tag{8.1}$$

$$v_{lat} = SOG \cdot \cos(heading) \tag{8.2}$$

v_{lon} and v_{lat} are the velocity components in the corresponding longitude and latitude direction, respectively. SOG (Speed Over Ground) is the vessels magnitude of velocity and heading is the vessels angle in degrees.

Route extraction

Visualizing the data can be done in many ways. Often as the data set is big, plotting every data point can be messy and misleading. Here, some information about how one route emerges and what classifies a route is of interest. We want to see where each vessel travels and which path each individual vessel takes. By looking at the timestamp column in the data set, there is often a trend of logged data from a certain time interval. This could indicate the start and end of a route. Keeping in mind that it could also mean some else e.g. turning of the tracking system, malfunction in the power etc. More details on this in the implementation chapter.

A check can be done with the velocities v_{lon}, v_{lat}. If the direction for each time series has the same direction they follow the same route.

Part II

Implementation

This section presents the methods and tools needed for visualizing and calculating the flow field in a given domain. A route classification based on raw AIS-data will also be carried out dividing this chapter into two parts.

The reason for this approach was that the solution relied on objects places in the area and thereafter finding the flow-field. This was also an important property because if it was possible to separate and set a constraint on the boundaries of land and sea then the boundary conditions and panels could then represent a coastlines on the selected area. The method would now need the coastlines of the selected map. Basemap with its built-in attributes, can collect the coastlines as line segments. For more details see below. The necessary components for computing and implemented the source panel method was in place.

In order to study the data and the environment around, a visualisation tool is necessary. This grants insight into how the vessel traverse and how much data is necessary for extracting wanted routes in area of interest. This will be done with a map present. In addition, the flow fields need to be plotted in order to understand it's motion. This section presents a framework in how to create a flow field using the source panel method applied for a real life scenario. From the works of (Pedersen and Fossen 2012) the method showed promising results for vessels. Traversing through islands we attempt to take it one step further by implementing it on a real map containing coastlines. Thereafter, a map can be created showing the most interesting areas by its input arguments saving computation time. This is done with a package from the matplotlib toolkit called Basemap (explained in more detail below).

With the map created with boundaries the next step will be to plot up some of the AIS-data. Reading the data into a python program is done with pandas.

Chapter 9

Analytic solutions

9.1 Elementary flows

The solutions of Laplace equation gives rise to some elementary flow types e.g sink, source and doublet. From the superposition principle this grants many different solution leading many different flow types. For the analytic solution it was chosen to create a class instance for each flow characteristics initializing the class with a flow strength value and location of origin in space $(x0, y0)$. For each method the defined grid is inputted and the corresponding velocities are returned. The velocity contributions are then added together for the final flow-field to visualized. Below is a flow chart for the class instance:

```
flow = Flow.VelocityPotential(source_str, x0, y0)
u_source, v_source = flow.source(X, Y)
```

Parameters needed to create a flow is it strength and location $(x0, y0)$, respectively. Next, the flow characteristic is specified: source, sink or doublet type flow giving out its velocity components u,v.

An example of how a flow type is created with the class *VelocityPotential()* giving out the velocity components after specifying flow type. When every elementary flow are been created, the velocities are added together resulting in the final flow field.

9.2 Representing obstacles with flow characteristics

When calculating routes from A to B there are obstacles present. It is necessary to be able to represent these obstacles in the same the manner as the start and end points. The are some possibilities when it is comes to choosing the correct representation. The logical choice will be a source -or doublet flow. Recall, the source flow has a flow characteristic of moving radially away from its center point. In addition to placing source and sink representing the start and end point respectively, a source flow can also serve as a representation of obstacles which in this case would be islands with circular shape. The same goes for a doublet flow characteristic as this has been done in the literature (Pedersen and Fossen 2012) and (Palm and Driankov 2014).

9.3 Setting the individual strength coefficients

When placing out the individual flow types, the flow strengths or strengths coefficients needs to be set. The strength coefficients can be regulated and adjusted. This will affect the streamline paths. This is the case for analytic solution. One could argue that together with the flow characteristics the flow strengths determines the streamlines generated because of its. Also the obstacles size must taken into account when setting the strengths coefficients as the source and doublets flow characteristics flatten out according to eq (5.2) and eq(5.15).

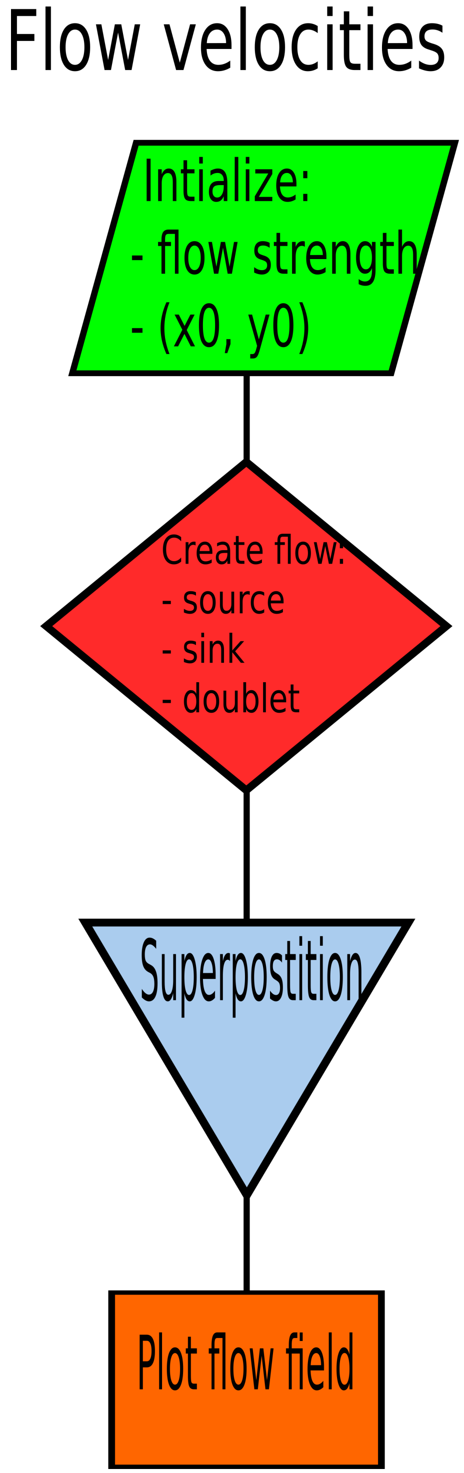


Figure 9.1: Flow chart using the class instance for calculating the velocities at every grid point for the analytic solutions of the Laplace equation.

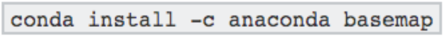
Chapter 10

Basemap

Creating the map in python is done with the matplotlib basemap toolkit. Together with matplotlib Basemap can represent geospatial data onto different map projections. The array containing vessel information can be used to read the longitude and latitude values needed for setting up the boundaries of the map. The necessary requirements for Basemap [1] are:

- Matplotlib 1.0.0 (or later, download)
- Python 2.6 (or later, including Python 3)
- Matplotlib 2.2 LTS requires Python 2.7 or later
- Matplotlib 3.0 requires Python 3.5 or later
- NumPy 1.2.1 (or later)
- Array support for Python
- PROJ4 Cartographic Projections Library

If using anaconda [2] environment Basemap can be installed by the following command in the terminal:



```
conda install -c anaconda basemap
```

When creating a map with Basemap an class object instance is made. The object as attributes and methods which are treated equally as any other class object in python.

[1] https://matplotlib.org/basemap/users/installing.html

[2] https://anaconda.org/anaconda/basemap

The arguments passed into the Basemap object the boundary points, projection type and resolution. Plotting the map is done using matplotlib's standard plotting commands. An example of a Basemap class instance is created with some formatting:

```python
from mpl_toolkits.basemap import Basemap
m = Basemap(llcrnrlon = minlon, llcrnrlat = minlat , urcrnrlon = maxlon , urcrnrlat = maxlat,
            resolution = 'i', projection = 'cyl', lat_0 = lat0, lon_0 = lon0)
m.drawmapboundary(fill_color = 'white')
m.fillcontinents(color = 'lightgrey', lake_color = 'white', zorder=1)
```

Listing 10.1: Here we show some the attributes Basemap has for its map objects together with the input needed for creating a map.

An object m is created with parameters lower left corner longitude ($llcrnrlon$), lower left corner latitude ($llcrnrlat$), upper right corner longitude ($urcrnrlon$) and upper right corner latitude ($urcrnrlat$). Basemap allows to choose 4 different resolutions: c (crude), l (low), i (intermediate), h (high), f (full) or None. The projection type . lon_0 and lat_0 sets the center of the map.

Creating map

Given the AIS-data we now can limit our map to the maximum and minimum longitude and latitude values of the data. The range of longitude and latitude degrees are (-180, 180) and (-90, 90) respectively. 0 degrees of longitude is the Prime Meridian which goes through Greenwich, England. negative values represent the western hemisphere and positive values represent the eastern hemisphere. For degrees of latitude, the 0 degree is placed at equator. Negative values represent the southern hemisphere and positive values the northern hemisphere.

The maximum and minimum values can found by:

```python
minlon = max(-180, min(df['Longitude']))
minlat = max(-90, min(df['Latitude']))
maxlon = min(180, max(df['Longitude']))
maxlat = min(90, max(df['Latitude']))
```

Listing 10.2: Boundary points for the map can be found from max and min function in the standard python library

These values are used as arguments for the Basemap class instance see listing 5.3. The projection chosen is cylindrical making the representation rectangular preserving the distance in the projection. See figure 10.1 below.

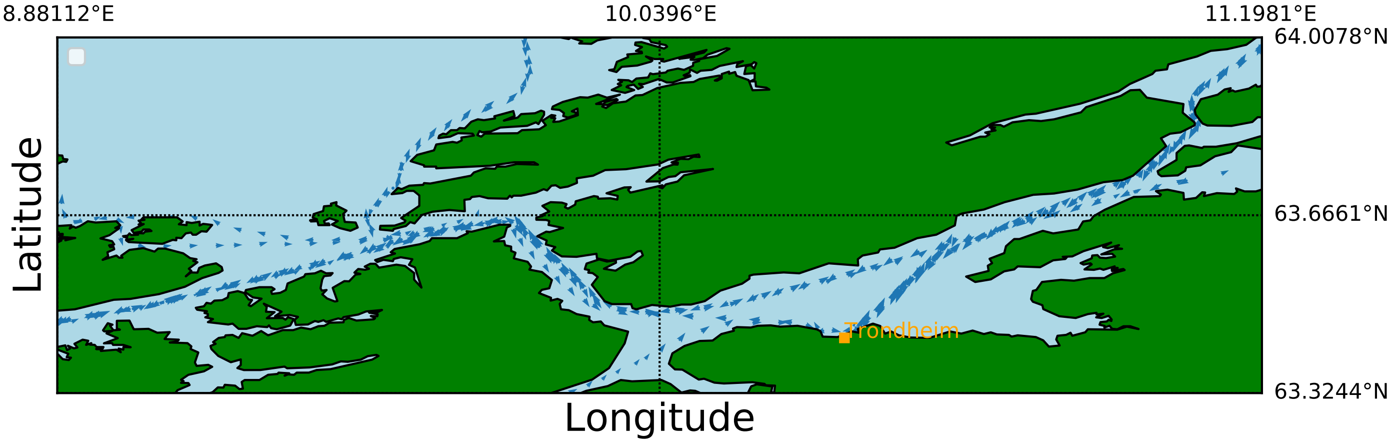


Figure 10.1: An overview of all tugboats in AIS-data. logged from 4/6/2018 to 13/10/2019 in the fjord of Trondheim. The blue arrows indicate the vessels direction of travel. The resolution in Basemap is set high. Format: grey is land white is ocean.

The map instance has several attributes. After creating a map instance, the coastlines are found using the method:

```
coast = m.drawcoastlines()
```

The data is gathered from a geography data set[3]. This object holds information about the coastlines which can be retrieved further by the class attribute:

```
coordinates = coast.get_segments()
```

coordinates holds a list of arrays where each array contains vertices making up the polygon representation of each connected land area. Lists can be converted into numpy arrays making it more convenient to work when computing mathematical operations.

[3] GSHHG(https://www.soest.hawaii.edu/pwessel/gshhg/)

Chapter 11

Application of SPM in Trondheimsfjorden

11.1 Computing the source strengths

With the use of Basemap' attributes, the necessary information needed to compute the source strengths λ is obtained. The analytic solution grant many different flow solution, but we do not necessary know the combination to a describe a certain flow. The idea is to place a set of unknown source strengths λ at the center of each segment of coastlines i.e. control points. Note that we assumed that each λ are constant, but can vary in strength. The vertices are collected from the attribute $get\_segments()$ and make up the boundary points (x_b, y_b) which is the only input to the model (see fig 6.3). Furthermore, given the boundary points the control points (x_{ci}, y_{ci}) can be calculated from eq (6.21) (6.22), respectively. The angles ϕ_i, δ_i and β_i are acquired using first eq (6.23), then eq's (6.24) and (6.25). The angles are stored in arrays with dimension $nx1$ where n is the number of panels.

From the boundary points, we have calculated the necessary components of what is needed for computing the geometric integral. Thereafter, solving the system of equations for the source strengths λ is what we will go through next.

For solving the geometric integrals for the normal and tangential component eq (6.44) and eq(**??**) two empty matrices, each with dimension nxn where n is the number of panels, are set up. As we saw from the derivation of the normal and tangential geometric integral, they share very much the same variables. Therefore, these are calculated at the same time. The I_{ij} geometric integral is calculated by the terms from eq (6.69) where A,B,C,D and

E is listed below . S_j is the length of panel j. For all $j \neq i$ eq (6.74) fills every element of the matrix, leaving us with the diagonal elements. As stated before it was found that when $j = i$ the I_{ij} has the value of $\frac{\lambda_i}{2}$. The tangential part is computed in the same manner using eq (**??**) for computing each element in the matrix J_{ij}. We are now ready to solve the system of equations, which means finding the source strengths λ at each control point (XC, YC).

We use the numpy module np.linalg.solve which solves a set of linear equations. In order to solve the matrix equations, the matrix must be square, have full rank i.e the columns or rows must linearly independent [1]. First, the diagonal elements of the matrix needs to be filled with π from eq(6.74) for the case when $j = i$ which means physically when we calculate the total contribution of potential induced at itself. With the matrix produced from the python program $sourcepanel.py$ where the function $solveGeometricIntegrals()$ is implemented, we use the array of β values found from $flowboundary.py$ and thus solve $I\lambda = b$ where I is the geometric integral values, b are the array β multiplied with $-2\pi V_\infty$. The full matrix equation can be seen in eq (6.74).

[1] (G. Strang, Linear Algebra and Its Applications, 2nd Ed., Orlando, FL, Academic Press, Inc., 1980, pg. 22)

```
I, J = sourcepanel.solveGeometricIntegrals(XC, YC, panels, phi, beta, XB, YB)
np.fill_diagonal(I, np.pi)
b = -2*np.pi*Vinf*np.cos(beta)
lamda = np.linalg.solve(I, b)
```

Computing the geometric integrals I, J and furthermore solving the system of equations from the matrix equation $I\lambda = -b$

11.1.1 Computing velocities

After computing the source strengths, the job is now to set up an area in which we can compute and see how the source strengths will affect this area and thereafter the velocities vectors. First we define a number of grid cells (nx, ny), more accurate results require more grid cells which leads to a higher computational load.

In order to help the streamlines converge to the wanted location, we have here decided to place a sink potential at the desired point, namely Trondheim in this case. After calculating the velocities with the defined grid points we just use the superposition principle adding this contribution to every grid cell.

Finally, we compute the velocity components at every defined grid cell in the area of interest. This is done with eq's (6.75) and (6.76). Note that from the latter equations the sink potential located at the end point is not included. Therefore, the final equation for computing the velocities at a point P due the potential induced at that point reads:

```
XGeom, YGeom = sourcepoint.solvePointGeometry(XP, YP, panels, phi, XB, YB)
```

```
vx[i, j] = Vinf*np.cos(AoA) + np.dot(lamda, XGeom.T)/(2*np.pi) + vx_sink[i, j]
vy[i, j] = Vinf*np.sin(AoA) + np.dot(lamda, YGeom.T)/(2*np.pi) + vy_sink[i, j]
```

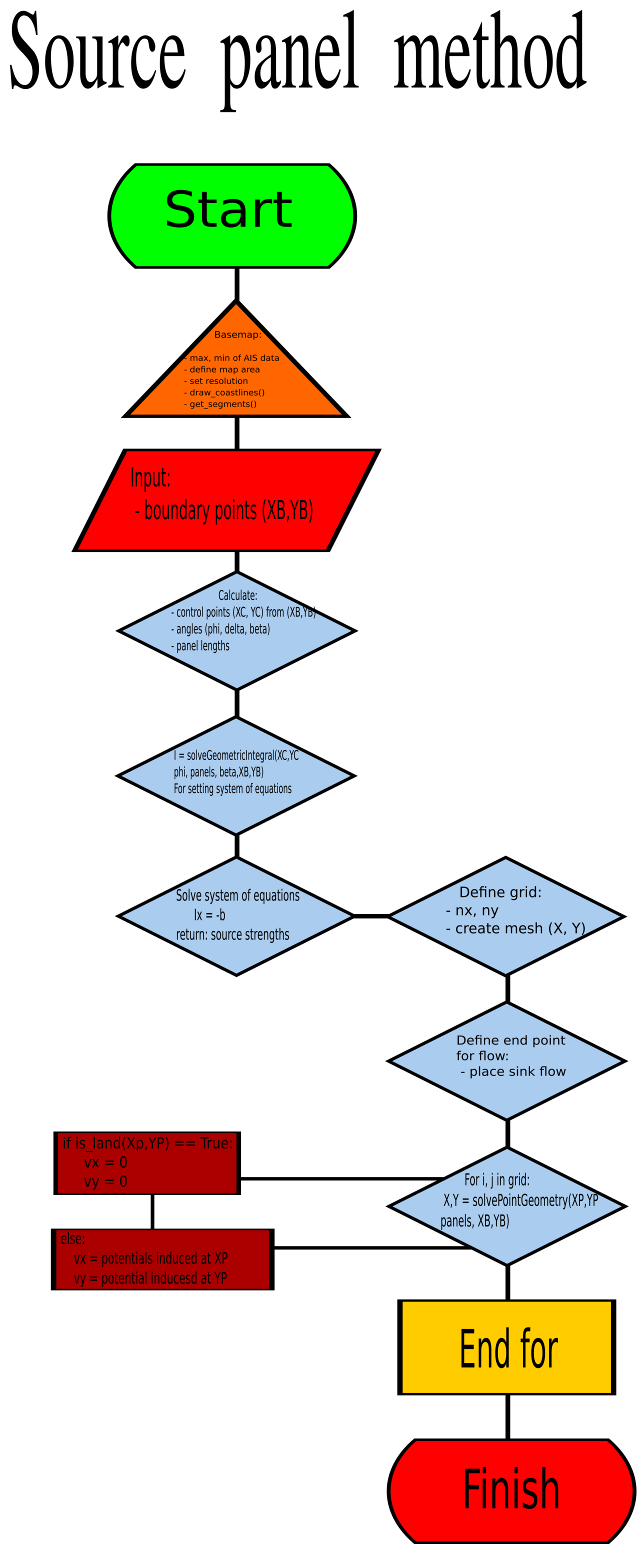


Figure 11.1: Flow chart for the source panel method

To summarize: a map was created from the max and min longitude and latitude values of the data set setting boundary points of the map area. We obtained the coastlines making it possible to set out the control points at mid point of each coastline segment. Furthermore, at the control points source strengths are placed giving n equations for n panels. The source strengths were then solved for by representing the n equation in matrix form eq (6.74). An intermediate step was to set a sink flow in the destination making flow more likely to converge toward the goal. With source strengths computed and the sink set out, the final task was finding the velocity at a point P induced by the source strengths yielding the velocity components of the flow at every point in the defined grid.

Chapter 12

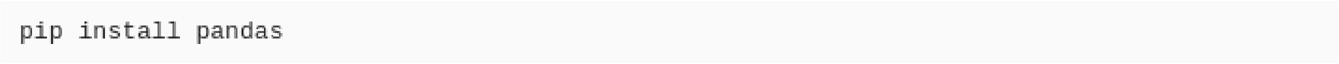
Application of AIS-data

12.0.1 Creating routes with Pandas

Pandas is a Python library intended for data analysis. A Dataframe (df) type object let the user manipulate a dataset by indexing, slicing, subsetting, merging, accessing rows and columns to only name a few operations. Pandas requires Python 3.6.1 or higher. Pandas can be installed with anaconda[1]:

```
conda install -c anaconda pandas
```

or if using Python Package Index (PyPI) [2]:



```
pip install pandas
```

Pandas supports different types of files format. Here, the file type will be comma-separated-value, more commonly known as a csv file. The CSV files are read with pandas as:

```
import pandas as pd
df = pd.read_csv('Routing/aisToTrondheim.csv', index_col = '
    TimestampPosition', parse_dates = True)
```

Listing 12.1: The data lies in the folder ”Routing”. Pandas allows for several ways to index the data. We choose to index on the column ”TimestampPosition” column. This is read in as an argument when reading the file

[1] https://anaconda.org/anaconda/pandas

[2] https://pandas.pydata.org/pandas-docs/stable/getting_started/install.html

Calculating distances on a sphere is done with the Haversine formula: We create a new column in the df and calculate the distance between the wanted end point and every registered positional coordinate. This is calculated by: x_i and y_i are the ith longitude and latitude values in the df, respectively. x_{final} and y_{final} are the end points longitude and latitude values, respectively. We can calculate velocities and store the values and preserve the index in new columns in the df:

NOTE: not correct computations!

```
df[’Heading’] = np.radians(df[’Heading’])
df[’VelocityLongitude’] = df[’SpeedOverGround’]*np.sin(df[’Heading’])
df[’VelocityLatitude’] = df[’SpeedOverGround’]*np.cos(df[’Heading’])
df[’DistTrondheimHaversine’] = convert.distsphere(df[’Longitude’],
    trond_lon, df[’Latitude’], trond_lat)
```

Create and add new columns *Velocity Longitude*, *Velocity Latitude* and *DistTrondheimHaversine* and use the existing ones to convert degrees to radians.

The next step is to take a look at the data. As mentioned above (AIS-data Extracting routes) the timestamp column can be used to sample routes from the data set. The data set has dimension $71694x12$. Each column represent a variable for specific timestamp (see figure of AIS-data example plot above, 8.1). Working with all that data becomes too complex and chaotic making it difficult to extract information when visualizing. In addition, since their are many different vessels types e.g. with different size this will affect how each vessel type traverses. To get a clearer view, a subset of the data is drawn out. Here we it is based on the vessel type. We create a new df consisting only of one single vessel type:

```
df_subset = df[df[’ShipType’].isin([’tugboats’])].copy()
```

Listing 12.2: All tugboat type vessels are stored as new df which can be analyzed further

The subset now consists of only one vessel type (in this case tugboats) from the main data set. The objective now is to extract routes from the data subset as the only order in the dataset is their respective timestamp. From the timestamp column it can be seen that on the same day the data contains a fixed interval of logged AIS-data varying from seconds to minutes. One could argue that this is the same vessel being tracked. By checking the IMO number one can be sure that the data come from the same vessel.

A variable holding all unique dates in the subset is needed for separating the data even more. This is based on the assumption that every data point holding the same date constitute one unique route. The number of data points vary from date to date. We need to keep track of the number of data points in each route. At last, each set of routes made, we want to find vessels travelling to some end point. Each defined route does not necessarily share the same end point. There are various reasons for this some of which include: vessels turn off their AIS-sender, there could be a malfunction, vessels are not necessarily bound to travel to the same destination. We choose to separate each route into three categorise: vessels travelling to, through and from the destination. This classification is of interest when using the flow field we want every vessel to converge to the same point.

For each route we check if the distance is decreasing or increasing and make the assumption: If the last absolute distance value is larger than the first, we conclude that the vessel is travelling from the destination.

Vessels travelling through is measured by counting the number of times the distance increase occurs. If the count is larger then some number, we conclude the route is travelling through. And vice versa for route travelling to. By making this classification, we can easily discard routes not of interest. Below we show the implementation:

```python
for i in range(unique_dates.shape[0]):
    route = df_subset.loc[str(unique_dates[i])]
    last_dist = route['DistTrondheim'][-1]
    first_dist = route['DistTrondheim'][0]

    if last_dist < first_dist:
        df_subset.loc[route.index, 'Direction'] = 'To'
        count = 0

        # check route going to if starting to deviate from dest.
        for j in range(route.shape[0] - 1):
            next_dist = route['DistTrondheim'][j + 1]
            prev_dist = route['DistTrondheim'][j]
            if next_dist < prev_dist:
                #dist decreasing
                count = 0
            else:
                #dist increasing
                count += 1
            if count == 3:
                #Ship could be travelling through
                df_subset.loc[route.index, 'Direction'] = 'Through'
                break
    else:
        df_subset.loc[route.index, 'Direction'] = 'From'
```

To access the classified routes, we create df and collect all vessel classified as ’To’ from the ’Direction’ column and gather the corresponding variables needed to plot and visualize:

```python
df_ToTrondheim = df_subset.loc[df_subset['Direction'] == 'To',
                              ['Longitude', 'Latitude', '
   VelocityLongitude', 'VelocityLatitude', 'DistTrondheim']]
```

With the use of Pandas we have been able to extract subsets of data which in this case makes it less chaotic when visualizing the data. We have been able categorize and produce routes from the data. In addition, a basic classification determining the vessels’ direction and end points is made.

Part III

Results

Chapter 13

Results

Our main assumption and approach for creating a ship route simulator is based on the assumption that the path a vessel will follow is representing vessel's track as streamlines in the flow of a fluid. The route extraction for AIS-data sets up the vessel's trajectories toward Trondheim which is used to verify and compare the fluid flow. The goal is to obtain streamlines which converge in Trondheim from different areas on the map.

We presented the analytic and numerical solution for velocity potential and stream functions which lead to the representation of nonuniform flow profiles in a domain. Here, we will attempt to create a flow around circular objects using the analytic solutions of the Laplace equations. The cylinder representing an object which cannot be traversed through. To check the validity we add another cylinder and see if the solution still holds and can be generalized to an increase of objects.

13.1 Analytical flows around circular objects

For given circular objects and geometries we found that doublet was a good candidate for creating flows around objects. Note, not every choice for a streamline is optimal. Some streamlines lead too close to the object in front and could in worst case lead to collision with the object. This becomes an instance of optimizing w.r.t safety or economy. Travelling closer objects leads to a shorter path to goal, but the risk of collision is larger. The analytic solutions do have some important properties. Analytic solution are have a physical aspect to them, meaning they yield a good representation of the real world. Sharing some flexibility in where the flows are placed, the flow will nearly almost lead to the end point. The geometry though needs to easy to represent with the analytic solutions. Increasing the complexity of the flow, it might not be possible to use the elementary flow types. If it was possible the combinations of potentials or stream functions from Laplace equations needed to be tested a many time in order to fit the stream line profile.

From fig 13.1 and **??** they are chosen in a way that there exist path between the start and end point.

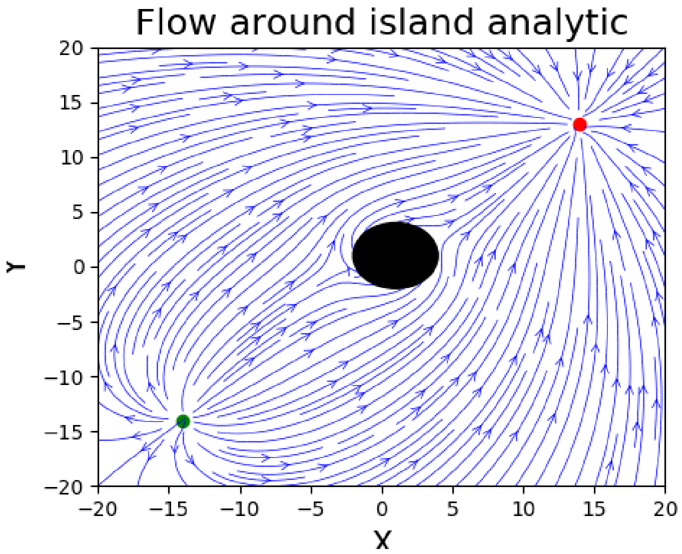


Figure 13.1: Here, streamlines (blue) shows a fluid flowing from Green (starting points), past the a circular object, converging at red (end point). The strengths chosen here are: 1, 4, 0.5

This solution is very computational light. The computational complexity can be increased by the amount of grid points defined in the domain. Note that the streamline does begin or end at the exact location of the source and sink. The reason being that a singularity occurs in those points see eq(5.3), (5.4), (5.6), (5.7).

Next we will see how to the flow appears when representing obstacles with a source flow characteristic.

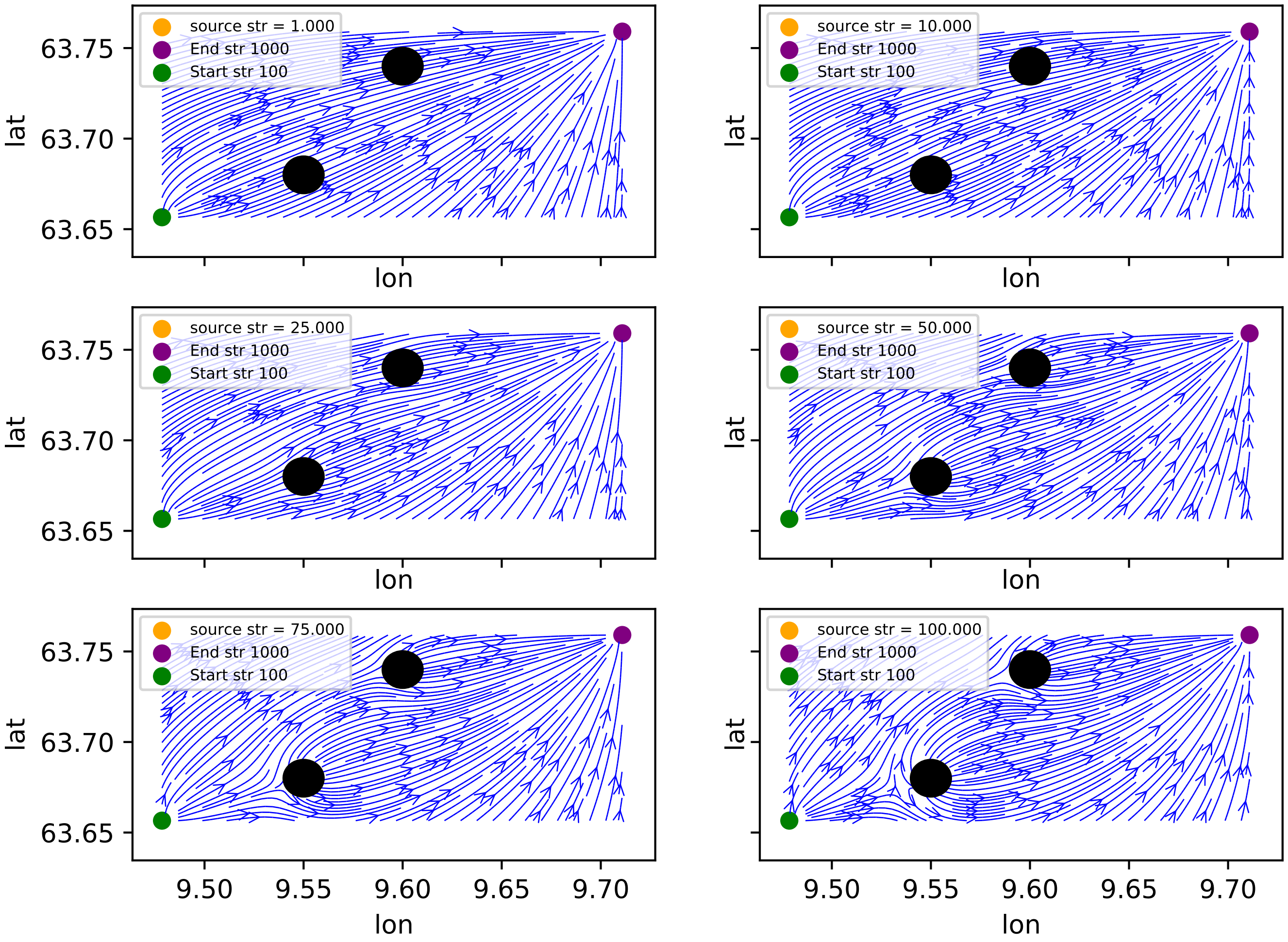


Figure 13.2: Different source strengths for islands (black) can be seen here. The flow changes together with the strengths. The source type flow (orange) is placed in the center of each island.

For circular objects the source type flow can be used, but the strength coefficient is an important parameter for changing the flow characteristic. As seen the figure above 13.2, the goal is to make the streamlines bend away from the islands creating a potential route from a starting point A (green) to end B (purple). In the upper left, the island does not have a strong enough coefficient relative to the start and end strength coefficients. Increasing the strength coefficients we see that in middle-right and lower left figures that the flow around the islands tend to curve away from the islands. By choosing one streamline to follow and assuming the two islands present, a vessel could traverse through.

Verifying analytical solution on real islands

Here, we see how the model handles the flow around real islands located outside the coast of Norway. A starting and end point will be defined at each side of the islands. As we will see, the islands will not have a complete circular geometry, but we will assume that

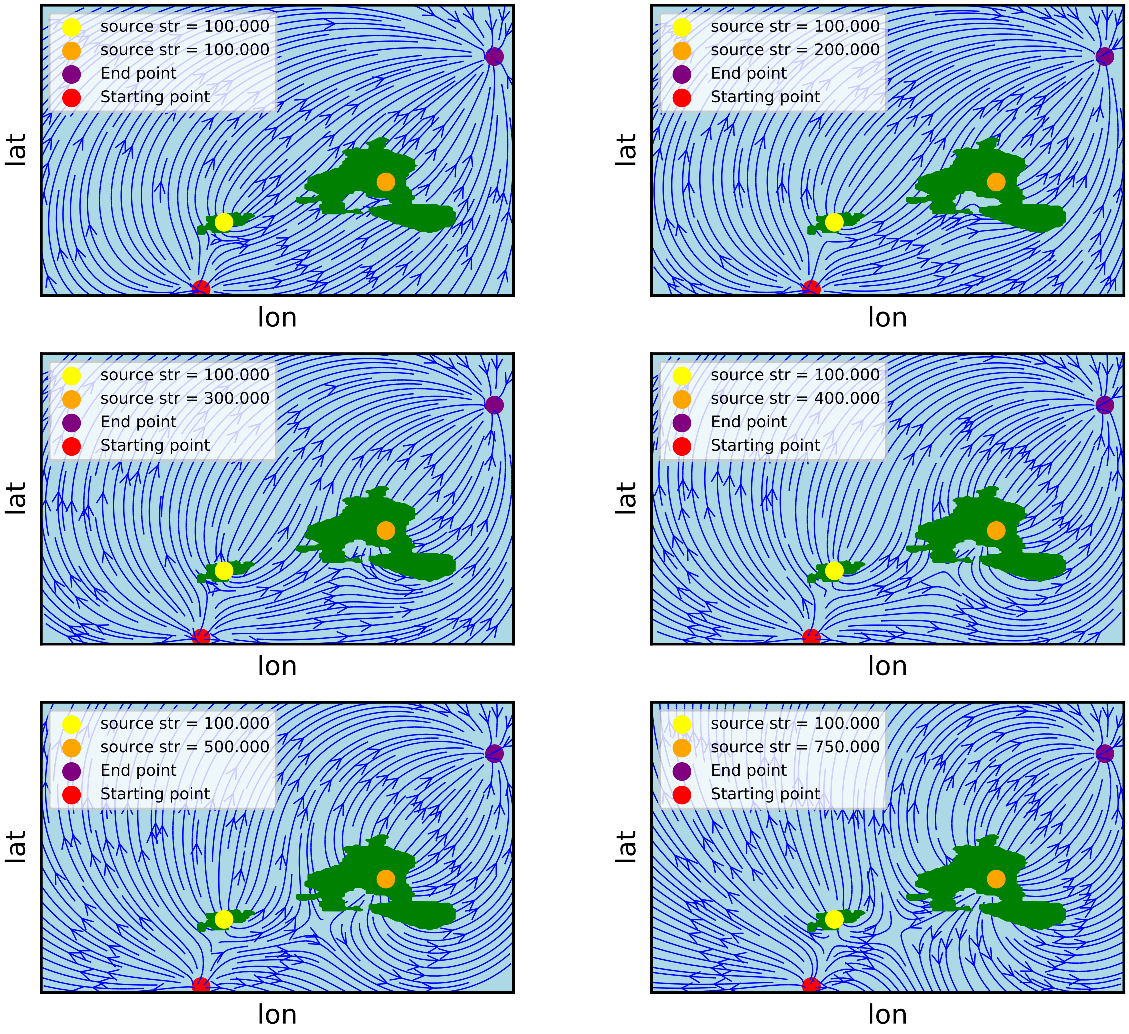


Figure 13.3: Sources located at (yellow and orange)

With this approach limitations occur. The complexity of the domain is not very applicable to real world map or environments. The adjusting the flow strength could be never ending if the goal is to replicate one specific vessel route with a streamline. In addition, when the objective comes to solving for distinct flow-field in a domain, information about the domain is needed. However, using the source panel method the complexity can increase and additionally we can rely on using map as an input to the model.

13.1.1 Verifying SPM on Trondheimsfjorden

We apply the source method to a real map, Trondheimsfjorden (see fig (13.5) - 13.5). How the model generalizes will be investigated. Furthermore, the resolution of Basemap plays an important as this determines the amount of boundary points. A too high resolution leaded to numerical errors when calculating panels lengths. The number of coastlines increase as the resolution increases giving a value of 0 in the E term in the geometric integral eq (6.69). This resulted further into the matrix I becoming singular. This was case for the two highest resolution choices $'h'$ (high) and $'f'$ (full). In addition, the attribute from Basemap $get\_segments()$ was not able produce correct coastline information on these two resolution settings. These resolutions had to be discarded. The resolution was then set to the $'i'$ (intermediate) being the best resolution able to draw correct boundary points and giving the correct representation of the map. Some adjustments were still necessary. The area around Trondheim does not have continuous coastline and islands in the fjord. Storing the coastal information which are vertices where the coastlines are represented as polygons. Going from different land areas additional lines were created which also created several boundary points and control points. This were needed to be located and removed or moved. Before calculating the source strengths additional control points were made at the map boundaries for flow not able to escape.

As with analytic solution some parameters are able to be adjusted. This included flow strengths such as sink strength at Trondheim and uniform flow which are not part of the coastline source strengths. This changes the streamlines and are as well as for the analytic solution very important to adjust correctly in order to get a good flow representations.

The grid for calculating the velocities are uniformly spaced. Some grid cells may be skipped as some grid cell are located on the land areas. This is taken care by calling the function $find\_land.arr()$ returning the value 0 if value is over sea and 1 if over land. These two values are switched making it convenient to make all values over land 0.

From fig (13.5 and 13.4) we see how the SPM produces streamlines. Let's examine how the SPM performs. Starting from the upper not all streamlines can be followed to Trondheim as some only move horizontally and ending up in coastal territory. It is the same case for the upper right corner. We would like the flow to run through every possible way and converge to Trondheim,but this is clearly not the case. However, there seem to be some potential in the model. We see that when approaching Trondheim, the streamlines seems to be more attracted to the sink located at Trondheim.

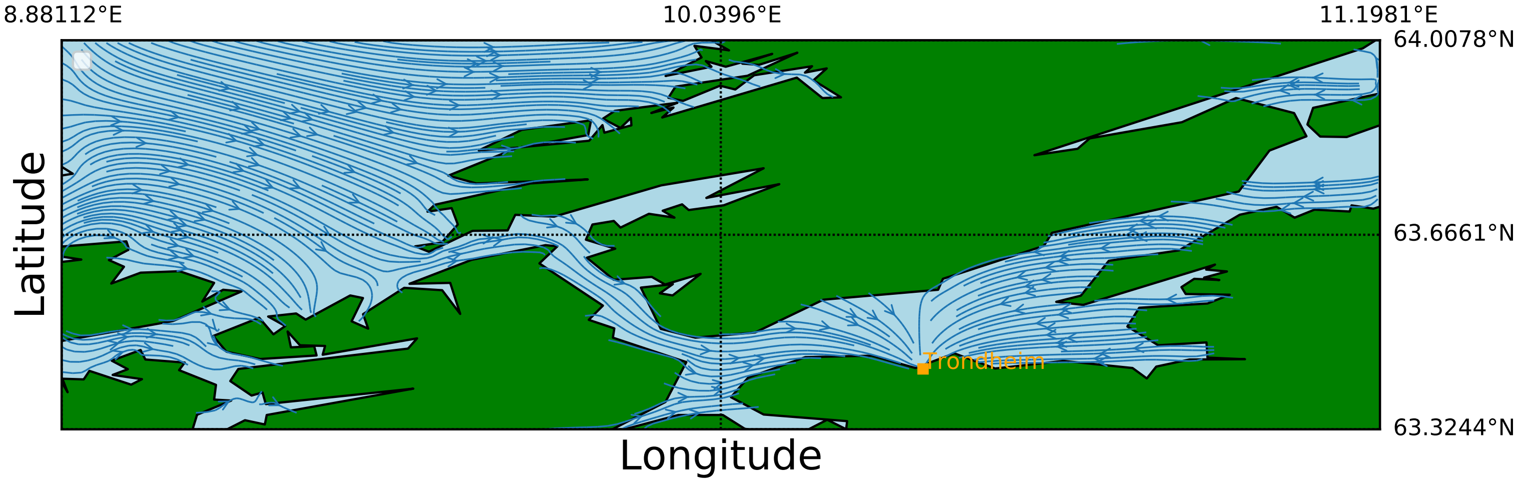


Figure 13.4: Streamlines produced from SPM together with a sink flow located at Trondheim and uniform flow. The uniform flow is set to be positive at $X < Trondheimlon$ and negative else. The $AoA = 0$ giving a uniform flow in the x-direction. The strength coefficient of the sink is set to $\lambda_{sink} = 1$.

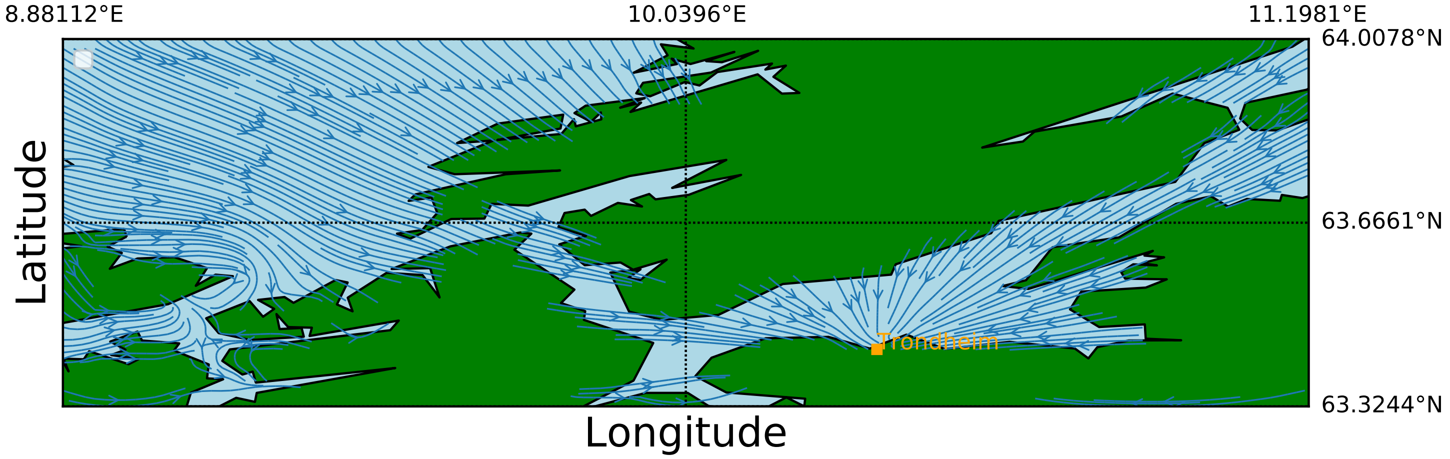


Figure 13.5: Here we can see change in the uniform contribution for SPM compared to 13.5. The parameters are set to: $AoA = -\frac{\pi}{2}$, $\lambda_{sink} = 0.1$ and $V_\infty = 0.01$

From figures (13.5) and (13.4) adjusting the parameters for V_∞, AoA and $_{sink}$ impacts the generated streamlines a significant amount. One might consider the map being to large as it look to be to difficult to get every streamline to converge to Trondheim.

13.2 Classifying routes

The raw AIS-data is not ordered based on any metric. Therefore, a classification is done making is more straightforwardly to collect routes. From 10.1 every data point is plotted. This is of course not an optimal setting and is excessive. The reason for this was to make it clear how chaotic the data looks like without any metric. For this case, we would also like to point out that not all classification done are with the equal amount of significance or validity. The reason being the lack of data which could be caused by malfunction at either sender or receiver, turning of sender or others.

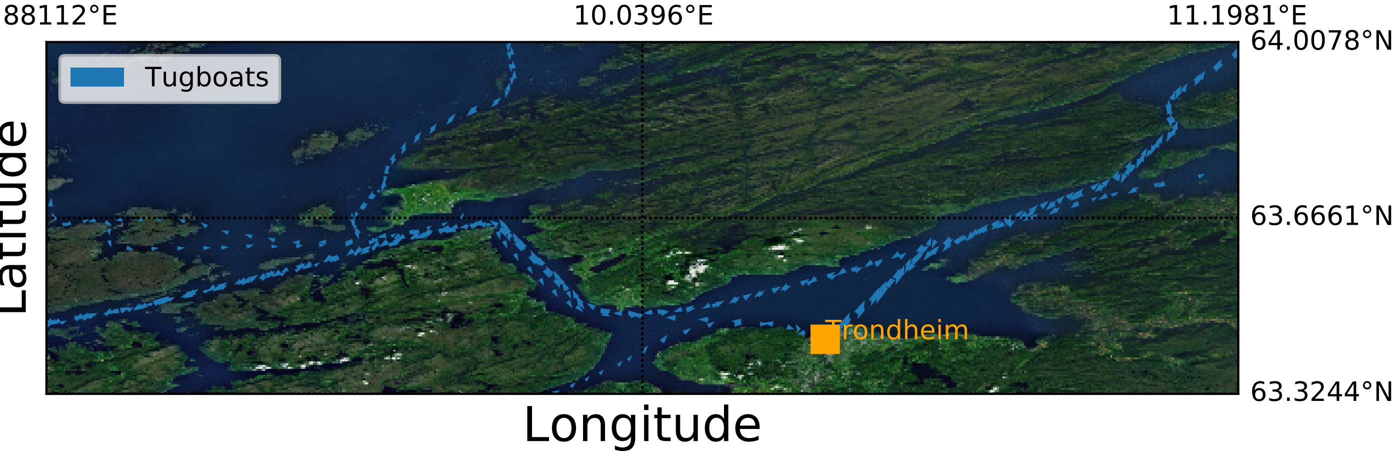


Figure 13.6: A subset of the data containing all tugboats are seen here. Some similar traversing is done going to and from Trondheim. Some tugboats are going straight past Trondheim which must be handled by route extraction.

The projection style chosen is equidistant cylindrical. This projection divides the earth up into equal rectangular pieces. Preserving the distance between points and thereafter comparing them is what is mainly focused on here. As seen in figure 13.7 at upper part we see only two AIS-data points. This underlines the mentioning of data not being logged for equal amount of timestamps and do not guarantee a convergence toward Trondheim only the direction.

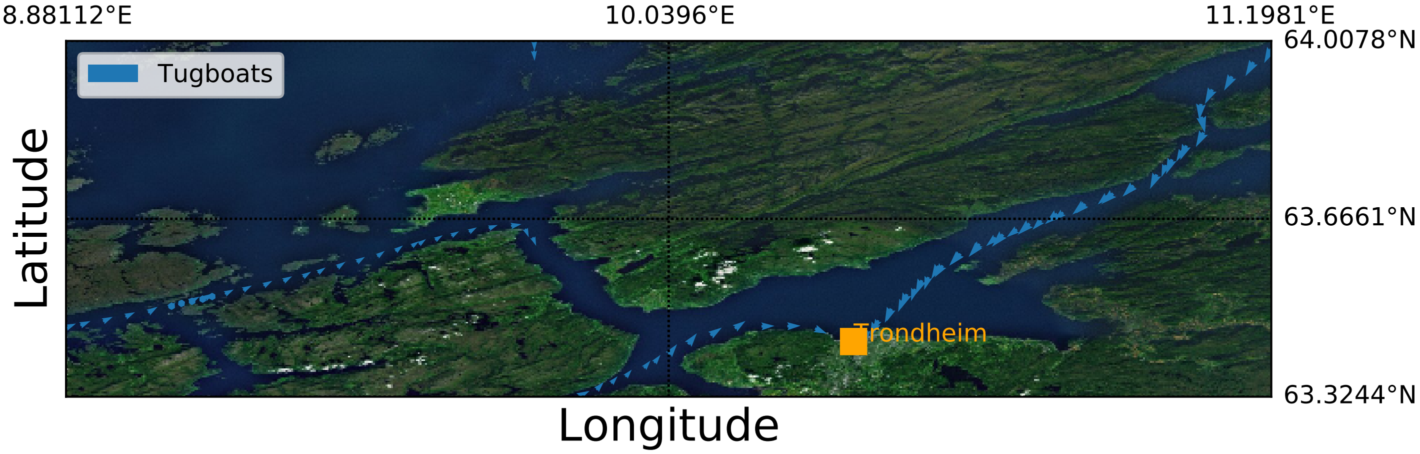


Figure 13.7: Here, all ”To” Trondheim vessel are plotted. As we see, the amount of AIS-data contained in the individual routes vary from route to route.

13.3 Comparing SPM with vessel route

Bibliography

Aarsnes, Marion (June 2018). “A Feasibility Study of Assessing Bunkering Operations Through AIS Data”. en. In: p. 197.

Acheson, D.J (2003). *Elementary Fluid Dynamics*. Oxford.

Anderson, John D. (1991). *Fundamentals of Aerodynamics*. en. 2nd ed. McGraw-Hill Series in Aeronautical and Aerospace Engineering. New York: McGraw-Hill. ISBN: 978-0-07-001679-8.

Arhus, Gisle Hoel and Stian Røyset Salen (2018). “Predicting Shipping Freight Rate Movements Using Recurrent Neural Networks and AIS Data”. en. In: p. 78.

Besse, Philippe et al. (Aug. 2015). “Review and Perspective for Distance Based Trajectory Clustering”. en. In: *arXiv:1508.04904 [cs, stat]*. arXiv: `1508.04904 [cs, stat]`.

Encyclopedia britannica (2020). *Latitude and Longitude*. https://www.britannica.com/science/latitude.

Gjevik, Bjørn (2009). *Innføring i Fluidmekanikk*.

Hvamb, Knut (2015). “Motion Planning Algorithms for Marine Vehicles”. en. In: p. 153.

Kim, Jin-Oh and Pradeep K Khosla. “Real-Time Obstacle Avoidance Using Harmonic Potential Functions”. en. In: (), p. 28.

Lewis, Charles Lee (1927). *Matthew Fontaine Maury, the Pathfinder of the Seas*. United States naval institute.

Meijer, Ricardo (2017). “ETA Prediction Predicting the ETA of a Container Vessel Based on Route Identification Using AIS Data”. PhD thesis.

Pallotta, Giuliana, Michele Vespe, and Karna Bryan (June 2013). “Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction”. en. In: *Entropy* 15.12, pp. 2218–2245. ISSN: 1099-4300. DOI: `10.3390/e15062218`.

Palm, Rainer and Dimiter Driankov (2014). “Fluid Mechanics for Path Planning and Obstacle Avoidance of Mobile Robots:” en. In: *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics*. Vienna, Austria: SCITEPRESS - Science and and Technology Publications, pp. 231–238. ISBN: 978-989-758-039-0 978-989-758-040-6. DOI: `10.5220/0004986902310238`.

Pedersen, Morten D. and Thor I. Fossen (2012). "Marine Vessel Path Planning & Guidance Using Potential Flow". en. In: *IFAC Proceedings Volumes* 45.27, pp. 188–193. ISSN: 14746670. DOI: 10.3182/20120919-3-IT-2046.00032.

Šavrič, Bojan and Kennedy, Melita (2020). *Map Projections*. https://www.google.com/url?sa=i&url=https G4borMyEPS8um7oLQP&ust=1582711305852000&source=images&cd=vfe&ved=0CAIQjRxqFwo

Smestad, Bjørnar Brende, Bjørn Egil Asbjørnslett, and Ørnulf Jan Rødseth (2017). "Expanding the Possibilities of AIS Data with Heuristics". en. In: *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation* 11.2, pp. 93–100. ISSN: 2083-6473. DOI: 10.12716/1001.11.02.10.

Snyder, John (1987). *Map Projections- a Working Manual*.

UN-Business Action Hub, United Nations (2020). *Business.Un.Org*. https://business.un.org/en/entities/13.

Wang, Xuantong, Jing Li, and Tong Zhang (Dec. 2019). "A Machine-Learning Model for Zonal Ship Flow Prediction Using AIS Data: A Case Study in the South Atlantic States Region". en. In: *Journal of Marine Science and Engineering* 7.12, p. 463. ISSN: 2077-1312. DOI: 10.3390/jmse7120463.

Wickham (2014). "Tidy Data". In: *Journal of statistical software*, p. 24.