1290

UNIVERSIDADE Đ
COIMBRA

Eva Carolina Santos Seiça

Maritime Drift Modelling Prediction

Dissertação no âmbito do Mestrado em Matemática, Ramo Análise Aplicada e Computação, orientada pela Professora Doutora Ercília Cristina da Costa e Sousa e pelo Engenheiro Carlos Matos Rodrigues (Critical Software) e apresentada ao Departamento de Matemática da Faculdade de Ciências e Tecnologia.

Outubro de 2020

Maritime Drift Modelling Prediction

Eva Carolina Santos Seiça

UNIVERSIDADE Ð
COIMBRA

Master in Mathematics

Mestrado em Matemática

MSc Dissertation | Dissertação de Mestrado

October 2020

Acknowledgements

Creio que esta é, e será sempre, a parte mais difícil de escrever. Se tudo o resto é escrito por, e com amor, esta parte ainda mais. Por me permitir expressar a minha gratidão (e haverá algo mais bonito que ter motivos para ser grata?) a pessoas (ou anjos?) que cruzaram em algum momento a minha vida e me fizeram chegar aqui.

Aos meus pais, porque sem eles, eu não estaria aqui, porque eles são os maiores exemplos da minha vida. Se eu puder um dia, ser metade do que eles são, então eu já serei feliz.

Ao meu irmão, porque foi um presente de Deus na minha vida, que veio ao mundo para me arrancar as risadas mais sinceras.

Aos meus avós (os 5!) pelo exemplo de superação que me deram e ainda dão. Porque me ensinaram que a vida é tudo menos fácil, que está sempre pronta a testar a nossa força e capacidade de seguir em frente. Que Deus me permita sentir sempre a presença deles, no coração (não há lugar mais bonito para guardar quem amamos), mesmo quando já não os tiver comigo.

Aos meus Professores, dotados de uma Excelência que palavras não podem descrever. Em especial à Professora Ercília por me exigir que desse sempre o meu melhor, mas também pelo carinho quando o desespero e a falta de confiança eram maiores que a coragem. Ainda, aos avaliadores desta dissertação, pelo tempo disponível para apreciar e contribuir para este meu projeto.

Aos amigos que neste trajeto da minha vida tive o privilégio de conhecer, em especial à Carina Tavares e ao Gonçalo Oliveira, que traçaram este caminho a meu lado. E àqueles cujo rumo mudou, mas que, pela amizade que nos une, a vida não tirou do meu caminho, em especial a Inês Costa.

Aos meus colegas da Critical Software, pelo espírito de trabalho em equipa que me transmitiram, por nunca deixarem que me sentisse sozinha e desamparada. É um orgulho para mim, poder contar no meu caminho com uma experiência tão enriquecedora, ao lado profissionais exímios. Em particular, aos Vulcan, à equipa do Oversee e ao Carlos Matos Rodrigues, por serem sempre uma fonte de motivação e me desafiarem a ser a melhor versão de mim própria. À Susana Boavida, por ser a minha "Scrum Mother" e pelo carinho e disponibilidade com que fez a revisão desta dissertação.
Ao feliz acaso de me ter cruzado, algures naqueles corredores, com a pessoa com o sorriso mais bonito do mundo e com mais paciência para me aturar e para me amar, mesmo nos dias difíceis.

Aos meus amigos de uma vida e, em particular, às mesmas de sempre. Que a vida me permita ter sempre tempo e sorrisos para partilhar com eles. Àqueles amigos que a vida teimou em roubar-me cedo demais (diz-se que é o que acontece com os anjos), que a memória dos sorrisos partilhados seja sempre a minha força e me lembre sempre de que a vida é efémera e devemos aproveitá-la para sermos e fazermos os outros felizes.

À minha segunda família, a Cruz Vermelha Portuguesa (Delegação de Coimbra em particular), e a todos os Voluntários, que são para mim exemplos de altruismo e de amor ao próximo. Peço todos os dias que a vida lhes retribua, sempre, todo este amor. Escolher fazer parte desta Ca(u)sa deu-me muito mais do que eu algum dia poderei retribuir: amigos que são família, sorrisos no meio das lágrimas, os obrigados mais sinceros.

A Deus, pela sorte de ter tudo o que referi até agora. Por permitir que as pedras no meu caminho nunca virem muros intransponíveis, por caminhar ao meu lado e nunca me deixar desanimar, nem nos momentos em que tudo parecia correr mal. Por ser, sempre, ao longo do caminho da vida, a minha força e sustento.

"Confiarei, ainda que o dia escureça..."

I believe that this is, and will always be, the most difficult part of writing this dissertation. If everything else is written by, and with love, this part even more. For allowing me to express my gratitude (and is there anything more beautiful than having reasons to be grateful?) to people (or angels?) who crossed my life at some point and made me get here.

To my parents, because without them, I wouldn't be here. They are the greatest examples of my life. If I can ever be half of what they are, then I'll be happy.

To my brother, because he was a gift from God in my life, who came into the world to get the most sincere laughs out of me.

To my grandparents (the 5!) for the examples of overcoming that they gave and still give to me. Because they taught me that life is nothing but hard, that life is always ready to test our strength and ability to move forward. May God always allow me to feel their presence, in my heart (there is no more beautiful place to keep those we love), even when I no longer have them with me.

To my Professors, endowed with an excellence that words cannot describe. Especially to Dr. Ercília for demanding that I always give my best, but also for the affection when despair and lack of confidence were greater than courage. Also to the judges of this dissertation for the time taken to enhance my work.

To the University of Coimbra, in particular to all who have contributed to making me proud to be here today to concluding what will always be one of the greatest achievements of my life.

To the friends that Mathematics brought into my life, especially Carina Tavares and Gonçalo Oliveira, who traced this path beside me, and to those whose course has changed, but life has not got them out of my way, in particular, Inês Costa.

To Critical Software and all my colleagues (that I now can call friends) at there, for the spirit of teamwork that they transmitted to me, for never letting me feel alone and helpless. It is a pride for me, to be able to count on my path with such an enriching experience, alongside excellent professionals. Especially to Vulcan, Oversee team and Carlos Rodrigues who followed my path closely and always gave me the necessary support. To Susana Boavida, for being my "Scrum Mother" and for the care and availability with which she revised this dissertation.
To the happy chance of having crossed, somewhere in those corridors, with the person with the most beautiful smile of the world, with more patience to me and to love me, even in the hard days.

To my friends for life and, in particular, to "the same as always". May life allow me to always have time and smiles to share with them. To the ones that life took away from me too soon (they say that this is what happen with angels), may the memory of our shared smiles be always my strength and always remember me that life is ephemeral and we must use it to be and make others happy.

To my second family, the Portuguese Red Cross family (especially Coimbra's Delegation), and to all the Volunteers, who are examples of altruism and love for their neighbor. I pray every day that life will always repay you for all this love. Choosing to be part of this HOME gave me much more than I will ever be able to repay: friends who are family, smiles in the midst of tears and the most sincere thanks.

To God, for the lucky to have everything I've mentioned so far. For allowing the stones in my path to never become insurmountable walls, for walking beside me and never letting me get discouraged, even when things seemed to go wrong. For being, always, along the path of life, my strength and sustenance.

"I will trust, even if the day gets dark ..."

Abstract

This dissertation started with a challenge related to a Critical Software's Project - Oversee - developed in partnership with the Portuguese Navy, Irish Coast Guard and currently being deployed to Papua New Guinea. It supports maritime agencies' ability to rapidly and effectively respond to incidents through a user experience that minimizes cognitive load whilst maximizing the value of critical information. With an intuitive graphical information system (GIS) it allows agency operators to instantly locate, respond to and manage emergencies with a common operational picture. Capabilities include real-time analysis of sea vessel movements and the ability to task available resources directly through the system. Apart from the Portuguese Navy, Oversee is used by maritime agencies like the Irish Coast Guard (IRCG), supporting their efforts to tackle 2,600 emergencies per year. Oversee improved the IRCG's ability to rapidly respond to maritime incidents and, in the first year after deployment, it helped to save 405 lives - an increase of 33% on the previous year. Oversee also allows to cut down on wasted resources by identifying some incidents as false alarms. The main goal of this study is to be able to predict the drift of an object by the deduction of a model that describes its trajectory based on some known data. Such data consists on conditions of the environment in which the object moves (water, in this case), and of the surrounding environment (weather conditions). In this dissertation, two models will be considered: one, related to an object of reduced dimensions where the influence of wind and different currents are neglected; another consists of an object with considerable dimensions where the influence of wind and Coriolis' force is considered. Gravity force produces different reactions in each one of these cases. A study will be carried out to describe the existence and uniqueness of the system's solutions. The consistency and global error bound are studied for the Runge-Kutta method, in particular, the 4th order Runge-Kutta method. Two types of stability are also analysed, A-stability and B-stability. For the two scenarios, numerical simulations will be carried out using the Runge-Kutta method to visualize the ocean drift under different assumptions.

Resumo

Esta dissertação teve início num desafio relacionado com um Projeto da Critical Software - o Oversee - em parceria com a Marinha Portuguesa, a Guarda Costeira Irlandesa e que está a ser atualmente implantado na Papua Nova Guiné. Este sistema nasceu para oferecer suporte à capacidade das agências marítimas de responder rápida e eficazmente a incidentes. Com um sistema de informação gráfica intuitivo (GIS), os operadores podem localizar, responder e gerir emergências instantaneamente com uma imagem operacional comum. Os recursos incluem análise em tempo real dos movimentos dos navios marítimos e a capacidade de verificar os recursos disponíveis diretamente através do sistema. Na Guarda Costeira Irlandesa (IRCG), o Oversee apoia cerca de 2600 emergências por ano. Com esta supervisão, a capacidade da Guarda Costeira Irlandesa foi melhorada no que diz respeito à resposta rápida a incidentes marítimos - no primeiro ano em que foi implementado ajudou a salvar 405 vidas, um aumento de 33% em relação ao ano anterior. O Oversee permite também reduzir o desperdício de recursos, identificando alguns incidentes como falsos alarmes.

O objetivo principal desta dissertação é prever a deriva de um objeto através da dedução de um modelo que descreva a sua trajetória com base em alguns dados conhecidos. Esses dados consistem nas condições do ambiente em que o objeto se move (a água, neste caso), e do ambiente circundante (condições meteorológicas). Para tal, serão considerados dois modelos: um, relativo a um objeto de dimensões reduzidas onde a influência do vento e de diferentes correntes é desprezada; o outro consiste num objeto com dimensões consideráveis em que a influência do vento e da força de Coriolis é considerada. A força da gravidade produz reações diferentes em cada um desses casos. Será realizado um estudo para descrever a existência e unicidade de soluções do sistema. A consistência e o erro global são estudados através do método Runge-Kutta, em particular, do Runge-Kutta de 4ª ordem. São ainda analisados, para o método Runge-Kutta, dois tipos de estabilidade, A-stability e B-stability. Para os dois cenários, as simulações serão feitas com o método Runge-Kutta de 4ª ordem.

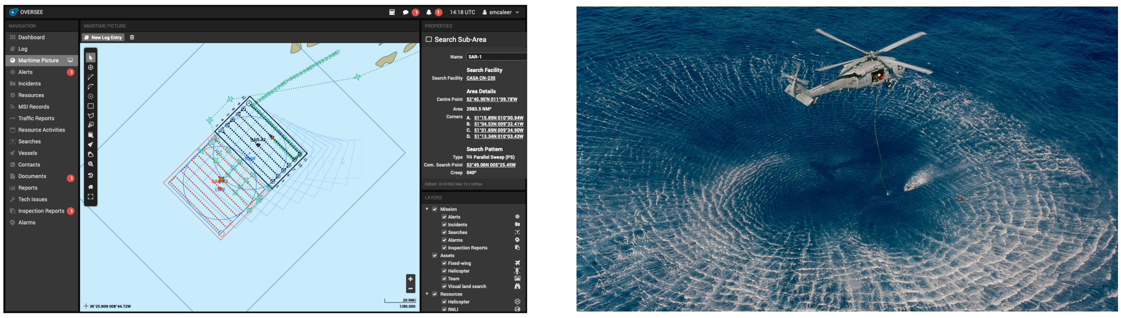
Table of contents

List of figures

Chapter 1

Introduction

In April 2019, one shipwreck occurred around the islands of São Tomé e Príncipe and another one around Peniche [13]. Thousands of kilometers separate these locations, but both incidents and respective SAR (Search and Rescue) operations have been monitored and coordinated from Portugal, more specifically from the Maritime Search and Rescue Coordination Center located at the Naval Base in Lisbon, using human resources (military of the Portuguese Navy) and national technological means, in particular, the OVERSEE software, developed by Critical Software (Portuguese company). Currently being implemented in Papua New Guinea, the motto is "Save on land with the knowledge of the Sea". OVERSEE is a maritime operations support information system used by customers in its daily search and rescue, environmental protection and sea law enforcement activities. It contains a drift simulation tool that provides an estimate of the different positions of an object adrift at sea over a specified period of time, based on its initial position and environmental factors acting on it such as wind and currents. It calculates the current in water generated by wind blowing over oceans' surface and estimates its probable error. Other currents acting on water are also considered such as tidal current and sea current. In coastal waters, tidal currents are usually important. To compute the tidal current, search planners should consult published tidal current tables, if available, for the vicinity of the datum position. Local knowledge is also often of great value in dealing with drift due to tidal currents. If valid local or computed data on short-term coastal surface currents are available, these values should be used. Sea currents derived from long-term seasonal averages should not be used when computing total water current in coastal waters.



☐ Search Sub-Area

Name SAR-1

Search Facility
Search Facility CASA CN-235

Area Details
Centre Point 52°45.95'N 011°09.78'W
Area 2983.5 NM²
Corners A. 51°15.89N 010°00.94W
B. 51°04.53N 009°32.41W
C. 51°01.69N 009°34.90W
D. 51°13.34N 010°03.43W

Search Pattern
Type Parallel Sweep (PS)
Com. Search Point 53°49.06N 005°25.45W
Creep 040°

Edited: 10 0106Z Mar 19 / refrias

36°25.80N 008°44.72W

(a) Drift simulation with Oversee system

(b) SAR operation monitored by Oversee

Fig. 1.1 Oversee system

Other works, related to maritime drift predicition, have appeared along the time. Recently, a team of researchers has developed a new algorithm to anticipate the location of drifting objects during the first three hours of a search by identifying zones in the water called transient attracting profiles (TRAPs), where currents and waves conspire to pull in nearby objects. In the early hours of future search-and-rescue operations, factoring in these previously hidden "attractors" could prove crucial in saving lives. The movement of ocean water can be represented mathematically as a velocity field, which in this case describes the speed and direction of water at each point on the surface. This new algorithm, described in May 2020 in Nature Communications [26] , uses ocean wave data and forecast models of this field to find zones with the strongest pull. Though invisible in the water, each TRAP can be drawn on a map as a curve of about 100 to 1000 meters long. As surface conditions change, TRAPs move slowly enough to drag objects along with them - similar to how a magnet moving under a table can pull an iron item along the tabletop. These researchers field-tested the method by throwing GPS-tagged manikins in the turbulent waters just south of Martha's Vineyard in Massachusetts. Each manikin followed a different trajectory, but "they all clustered on the same TRAP," just as the algorithm predicted. These initial tests were carried out close to shore, but the same mathematics predicts the presence of TRAPs in the open ocean. These will supplement, and not replace, existing models, by incorporating TRAP curves into actual prediction maps. More studies about effects of wind and buoyancy are being done in order to increase the predictions' accuracy.

This dissertation consists of conducting a maritime drift prediction study using mathematical models that can help on SAR operations. Two scenarios will be studied. In one case, an object of reduced dimensions is considered and influence of wind and different currents is neglected. The other case considers an object of larger dimensions where both the influence of wind and the force of Coriolis are considered.

The aim of the problems based on differential equations is to understand, as much as possible, what is expected to happen to a quantity satisfying a differential equation, i.e., predicting the value this quantity will have at some future time.

The main goal is to present an implementation and consequent application of the system of equations that models the speed and the trajectory described by a body, under different circumstances and under the influence of waves. The drift models will be deducted and studied in chapter 2. In chapter 3 a theoretical study is done for the system of differential equations related to the models previous presented in chapter 2. This study focuses on determining and analyzing the conditions for the existence and uniqueness of solutions of this initial value problems. The contents of chapter 4 focuses on the study of the Runge-Kutta numerical method, namely the study of consistency, global error bound and stability of the method. Finally, in chapter 5, numerical simulations are carried out, considering different variations of the variables involved in the model.

Chapter 2

Model problems

In this chapter two different drift models will be introduced. The movement of a body can be fully characterized by the forces that are exerted on it, i.e., it can be described by Newton's laws. Such laws determine that the resultant of the forces applied to a body produces an acceleration with its direction. In the first section, a model for small objects, where the influence of wind and different currents are neglected is presented. In the second one, another model considers an object with larger dimensions where both the influence of wind and Coriolis' force are considered. Finally, in the third section, a possible model for wave motion is described.

2.1 Model for small objects

To deduce the model that describes the drift of a floating object, it is necessary to understand which forces are acting on the body. In this case, the equation of the horizontal movement of a rigid body due to waves is deducted, assuming that the object is much smaller than the wavelength. This investigation is restricted to monochromatic waves. The movement equation is considered in a mobile frame and is given by [20]

$$m\mathbf{a} = \mathbf{F_d} + \mathbf{F_g}, \tag{2.1}$$

where $\mathbf{F_g}$ represents the sliding force due to gravity acting on the body, and $\mathbf{F_d}$ the drag force between the body and the water (see Figure 2.1). Still, $\mathbf{a}$ denotes acceleration and m the mass of the floating object. According to Figure (2.1), θ represents the slope of the sliding surface.

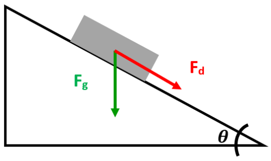


Fig. 2.1 Forces on a sliding surface slope model.

To better understand the acceleration, it is important to deduce its decomposition into two directional components at each point on the path. Such components, relative to the water surface, are called *normal* and *tangential* (**n** and τ, respectively) [29]. Next, the acceleration vector (2.1) is studied in detail.

In a mobile framework, $\mathbf{v} = v\tau$ represents the vector velocity, in the direction τ, where v represents the scalar speed and τ the versor in the tangential direction of the trajectory. If $\mathbf{r}$ designates the position vector, $\frac{d\mathbf{r}}{dt}$ consequently represents the displacement vector. Let s be the position of the body on the path.

Then,

$$\frac{d\mathbf{r}}{dt} = \frac{ds}{dt}\tau. \tag{2.2}$$

That is, the velocity vector in any movement, in order of time, is given by [31]

$$\mathbf{v} = \frac{ds}{dt}\,\tau. \tag{2.3}$$

The vectorial acceleration can be written, by deriving (2.3),

$$\mathbf{a} = \frac{d\mathbf{v}}{dt} = \frac{d^2s}{dt^2}\,\tau + \frac{ds}{dt}\,\frac{d\tau}{dt}. \tag{2.4}$$

In Figure 2.2 A and B are two points of a trajectory. A represents the location of the body at the time t_0 and B at the time $t_0 + \Delta t$. At the limit, $\Delta t \to 0$ and so, the directions of the two versers intersect at a common point, C. At this limit, the distances from C to each point, A and B, R_A and R_B respectively, will be the same. The distance between the centre of curvature at an instant t_i and the trajectory point at this same instant is de **radius of curvature**.

It is possible to say that the distance travelled by the body is then given by $s(t) = R\theta(t)$ in which R represents the radius of curvature and θ represents the angle relative to the radius of curvature.

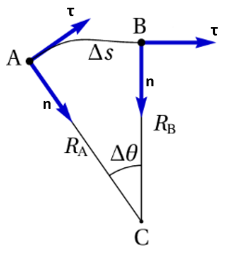


Fig. 2.2 Radius of curvature

We have that

$$\frac{d\tau}{dt} = \frac{d\theta}{dt}\,\mathbf{n} \tag{2.5}$$

in which $\mathbf{n}$ and τ are the normal and tangential versors, respectively. Replacing (2.5) in (2.4),

$$\mathbf{a} = \frac{d^2s}{dt^2}\,\tau + \frac{ds}{dt}\,\frac{d\theta}{dt}\,\mathbf{n} \tag{2.6}$$

and the acceleration decomposition is then obtained in its tangential and normal components.

The angular velocity value at each point of the trajectory is given by the velocity value divided by the radius of curvature, both at the point in study. It is now possible to write the angular velocity, $\frac{d\theta}{dt}$, according to these considerations. That is

$$\frac{d\theta}{dt} = \frac{1}{R}\frac{ds}{dt} \; . \tag{2.7}$$

The acceleration is then given by

$$\mathbf{a} = \frac{d^2s}{dt^2}\,\tau + \left(\frac{ds}{dt}\right)^2 \frac{1}{R}\,\mathbf{n} \; . \tag{2.8}$$

This is,

$$\mathbf{a} = \frac{dv}{dt}\,\tau + \frac{v^2}{R}\,\mathbf{n}. \tag{2.9}$$

Definition 2.1.1. *For a curve described by $y = f(x)$, the radius of curvature, R, is given by*

$$R = \frac{\left[1 + \left(\frac{dy}{dx}\right)^2\right]^{\frac{3}{2}}}{\left|\frac{d^2y}{dx^2}\right|} \; . \tag{2.10}$$

Let $\eta(x,t)$ be the deviation of the water surface (see Section 2.3) and $x(t)$ the horizontal location of the body (Figure 2.3).

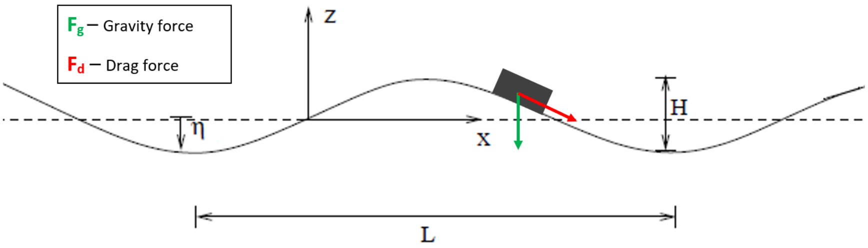


Fig. 2.3 Wave behaviour

It is now possible to identify normal and tangential versors in order to η given, respectively, by [20]

$$\mathbf{n} = \frac{\left(-\frac{\partial\eta}{\partial x}\,,\;1\right)}{\sqrt{1+\left(\frac{\partial\eta}{\partial x}\right)^2}}\,, \qquad \tau = \frac{\left(1\,,\;\frac{\partial\eta}{\partial x}\right)}{\sqrt{1+\left(\frac{\partial\eta}{\partial x}\right)^2}} \tag{2.11}$$

From Definition 2.1.1,

$$R = \frac{\left[1+\left(\frac{\partial \eta}{\partial x}\right)^2\right]^{\frac{3}{2}}}{\left|\frac{\partial^2 \eta}{\partial x^2}\right|} \tag{2.12}$$

and finally,

$$a_n = \frac{v^2}{R} = \frac{v^2\left|\frac{\partial^2 \eta}{\partial x^2}\right|}{\left[1+\left(\frac{\partial \eta}{\partial x}\right)^2\right]^{\frac{3}{2}}}. \tag{2.13}$$

Having this, the acceleration vector can now be written as

$$\mathbf{a} = \frac{dv}{dt}\,\tau + \frac{v^2\left|\frac{\partial^2 \eta}{\partial x^2}\right|}{\left[1+\left(\frac{\partial \eta}{\partial x}\right)^2\right]^{\frac{3}{2}}}\,\mathbf{n}\,. \tag{2.14}$$

Until now,

$$\mathbf{F_g} + \mathbf{F_d} = m\left[\frac{dv}{dt}\tau + \frac{v^2\,\frac{d^2\eta}{dx^2}}{\left(1+\left(\frac{d\eta}{dx}\right)^2\right)^{\frac{3}{2}}}\mathbf{n}\right]. \tag{2.15}$$

For the sliding force due to gravity [21], it follows that

$$\mathbf{F_g}\cdot\tau = -\frac{mg\,\frac{\partial \eta}{\partial x}}{\sqrt{1+\left(\frac{\partial \eta}{\partial x}\right)^2}}, \tag{2.16}$$

where $g \approx 9.81ms^{-2}$ and $\eta(x,t)$ represents the height of the body in the direction of gravity, that is, it represents the vertical deviation of the surface.

Performing the scalar product of (2.15) by τ, the following is obtained

$$m\,\frac{dv}{dt} = -\frac{mg\,\frac{\partial^2 \eta}{\partial x^2}}{\sqrt{1+\left(\frac{\partial \eta}{\partial x}\right)^2}} + F_{d_\tau} \tag{2.17}$$

where F_{d_τ} represents the drag force produced by the water on the body. Also, the horizontal component of velocity for the body in a dynamic system, is given by [20]

$$\frac{dx}{dt} = \mathbf{v} \cdot \tau = \frac{v}{\sqrt{1 + \left(\frac{\partial \eta}{\partial x}\right)^2}} \,. \tag{2.18}$$

This represents the velocity of the body tangential to the water surface.

The drag force is the force that resists to the movement of the solid body through the fluid [22]. It is a force characterized both by frictional and pressure forces that act in parallel and perpendicular to the surface of the body, respectively. This force is described by

$$F_{d_\tau} = \rho_w C_d A |V_w - v| (V_w - v) \tag{2.19}$$

with ρ_w the density of the fluid and C_d the drag coefficient. Also, A represents the contact area of the body with the fluid that it moves in and V_w and v tangential velocities of the fluid and the body, respectively.

This force can also be decomposed in its normal and tangential directions. Performing the scalar product of (2.15) by $\mathbf{n}$, it comes that

$$F_{d_\mathbf{n}} = \frac{mg}{\sqrt{1 + \left(\frac{\partial \eta}{\partial x}\right)^2}} + \frac{m\, v^2 \,\frac{\partial^2 \eta}{\partial x^2}}{\left(1 + \left(\frac{\partial \eta}{\partial x}\right)^2\right)^{\frac{3}{2}}} \,. \tag{2.20}$$

This force is, generally, represented by its tangential direction, since its normal direction is, nearly, balanced by the buoyancy force and so only the tangential component is able to produce movement in the body.

It is also necessary to consider that the body transfers part of its momentum to water when accelerating. This means that because the body moves in a fluid, it is subject to some inertia that leads to a difference between the actual mass of the body and its effective mass in the fluid. For this reason, appears the so-called added mass coefficient, C_m.

According to *Grotmaack* [14], the body's own inertia, $\mathbf{F_i}$, is balanced by the forces acting upon it, and so, the effective mass in water is greater than the real mass, which can be modelled by introducing a so-called added mass term (C_m represents the added mass coefficient) in the following way

$$\mathbf{F_i} = m(1 + C_m)\mathbf{a}.$$

Having this,

$$m(1 + C_m)\, \mathbf{a} = \mathbf{F_g} + \mathbf{F_d}. \tag{2.21}$$

Let m_i, inertial mass, be $m_i = m(1+C_m)$. Replacing (2.14) in (2.21), the following is obtained

$$m_i \left(\frac{dv}{dt}\tau + \frac{v^2 \left| \frac{\partial^2 \eta}{\partial x^2} \right|}{\left[1 + \left(\frac{\partial \eta}{\partial x} \right)^2 \right]^{\frac{3}{2}}} \mathbf{n} \right) = \mathbf{F_g} + \mathbf{F_d}. \tag{2.22}$$

The system of differential equations for the tangential velocity of the body in a coordinate system moving with the wave is given by

$$\begin{cases} m_i \dfrac{dv}{dt} = -gm_i \dfrac{\frac{\partial \eta}{\partial x}}{\sqrt{1+\left(\frac{\partial \eta}{\partial x}\right)^2}} + \dfrac{1}{2}\rho_w C_d A |V_w - v|(V_w - v) \\ \dfrac{dx}{dt} = \dfrac{v}{\sqrt{1+\left(\frac{\partial \eta}{\partial x}\right)^2}} \end{cases} . \tag{2.23}$$

2.2 Model for cargo containers

The immersion of a container is variable. It may remain constant or be a function of time. The main goal is to obtain a dynamic model that hindcasts drift trajectories under the influence of wind, current and Coriolis forcing. The wind action is considered on the emerged surface. To introduce the model, the Sherbro accident is now narrated [23].

On 8 December 1993, the French container ship Sherbro was en route from Cherbourg to Montoir (near Nantes) when it was caught in a heavy storm. Several rows of containers broke loose and fell overboard while others, thrown off balance, collapsed on deck.

In total, 88 containers fell overboard, 10 of which contained dangerous substances: two types of pesticides (12.2 tonnes), nitrocellulose (21.6 tonnes), sulphur (1 tonne), phenol (200 kg), methylketone (35 kg) and a flammable product (3.6 tonnes). Among these substances, the pesticides presented the greatest threat for the environment due to their persistence and therefore their bioaccumulation potential in marine organisms. On December 9, at 13:50, an overflight spotted 20 containers within a large area. On December 10, seven containers were drifting fourteen miles south west of Casquets area with a same kind of immersion ratio than the previous observations. On December 11, wind was still strong and one container full of cigarettes was found beached near Flamanville city coast. More, three observed containers were drifting between Guernesey and the Cotentin coastline. On December 12, at 13:30, one pesticides container was recovered. As 17:30, an air overflight succeeded in locating other seven containers. Three of them beached on the western coast near Flamanville. Two containers were drifting in the surroundings of La Hague Cape and one of those was damaged. The other ones were drifting four miles North West of Carteret. The mean position of the containers given by the numerical simulation [23] is consistent with the observations. Nonetheless, for an immersion ratio close to the extreme percent ranges, simulated trajectories were too fast or too slow. Containers 80% and 90% immersed were beached on the North coast of Guernsey Island and that was not observed. With an immersion of 10%, the container drifted too far in the Northeast direction. The hypothesis of 10% immersed container is unrealistic according to the observations. It appears that the best agreement between the simulated positions and the observed positions were given by the containers having a 60% immersion.

The model deduction will now be explained. The basic movement conservation can be written as [23]

$$m\frac{\partial \mathbf{v}}{\partial t} + mf\mathbf{k}\Lambda\mathbf{v} = \mathbf{F_a} + \mathbf{F_d} + \mathbf{F_r} \tag{2.24}$$

where m represents the mass of the object, $\mathbf{F_a}$ the wind drag, $\mathbf{v}$ the horizontal velocity, f the Coriolis parameter, k a unit vertical vector, Λ is the wavelength, $\mathbf{F_d}$ the water drag and $\mathbf{F_r}$ the radiation force (see Figure 2.4). Notice that both, $\mathbf{F_a}$ and $\mathbf{F_r}$ were not presented at the last model.

Also, the gravity force previously presented, $\mathbf{F_g}$, is now replaced by the radiation force, $\mathbf{F_r}$, in which the gravitational acceleration also appears. The inertial motion will be picked up by the Coriolis force and the object will be driven into an inertia circle which is then carried forward by the basic ocean current.

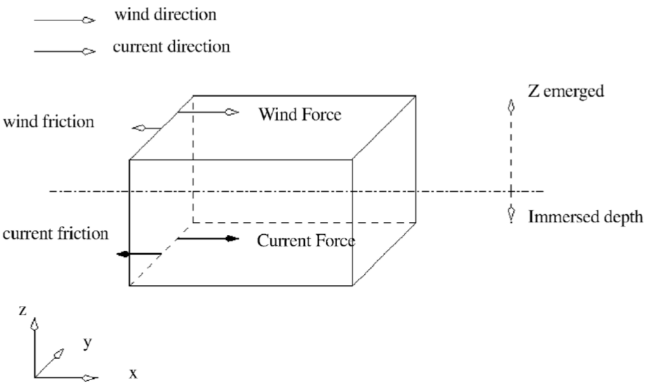


Fig. 2.4 Drift forces and parameters

Ekman [11] formulated the classical solution to the problem of upper ocean response to wind in a rotating reference frame. Sudden onset of wind over a viscous ocean initially at rest leads to damped inertial oscillations with a period $T = \frac{2\pi}{f}$ where f is the Coriolis parameter or inertial frequency. The Coriolis-Stokes force is perhaps the least disputed wave-mean flow interaction term in ocean models, and is an important part of coupled modeling systems [6].

The drag produced by the wind into the emerged section of the object is given by

$$\mathbf{F_a} = \frac{1}{2}\rho_a C_a S_a |\mathbf{V_a} - \mathbf{v}|(\mathbf{V_a} - \mathbf{v}) \tag{2.25}$$

where C_a is the drag coefficient, S_a is the cross-sectional area affected by the wind and V_a is the wind velocity at $10m$. C_a is defined as 1 due to the fact that air drag coefficient of bluff objects at high Reynolds number is typically about 1 [27]. As deducted so far, the tangential version of this force can be deducted such as

$$F_{a_\tau} = \frac{1}{2}\rho_a C_a S_a |V_a - v|(V_a - v) \tag{2.26}$$

The immersed section of the object is subjected to the water drag, a force that can be written as

$$\mathbf{F_d} = \frac{1}{2}\rho_w C_d A|\mathbf{V_w} - \mathbf{v}|(\mathbf{V_w} - \mathbf{v}) \tag{2.27}$$

where C_d is the drag coefficient, A is the cross-sectional area affected by the water and V_w is the water velocity. C_d has been defined as $0,8 < C_d < 1,2$ [5].

The tangential component of the force is then given by

$$F_{d_\tau} = \frac{1}{2}\rho_w C_d A|V_w - v|(V_w - v) \tag{2.28}$$

The force due to reflection of waves on a container wall is now defined:

$$\mathbf{F_r} = \frac{1}{4}\rho g a^2 L \tag{2.29}$$

where L is the length of the container normal to incident waves of amplitude a.

2.3 Mathematical formulation of the wave motion

Incompressible inviscid fluid flow usually gives a sufficiently accurate description for water wave motion [10]. Since water is an incompressible fluid, this means that its density is constant through time. Water viscosity is considered negligible [14] and any rotational movement is always ignored. So, potential theory can be used to describe the flow. Consider the case in which the Cartesian frame (x,y,z) is taken. In this section, $\eta(x,t)$, which represents the deviation from the water surface, relative to its horizontal balance, is introduced.

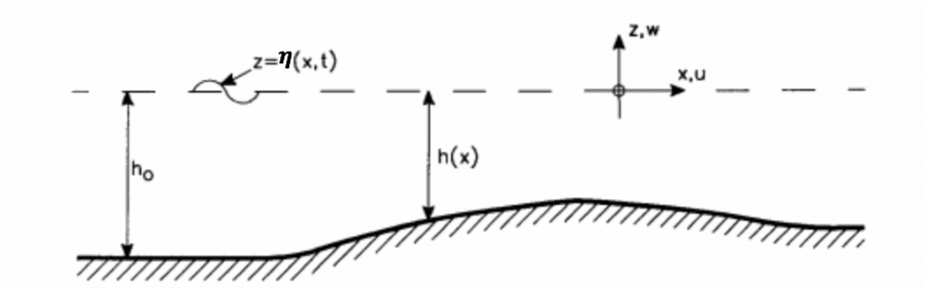


Fig. 2.5 Geometry in a vertical wave section

Assume that waves propagate in the horizontal direction, x, and a fluid domain boundary above, $z = \eta(x,t)$ for each instant of time t. Let the underneath impermeable surface of the fluid be at $z = -h(x,y)$. The potential velocity is given by $\Phi(x,z;t)$ related to the flow velocity components u_x and u_z in the horizontal and vertical directions respectively:

$$u_x = \frac{\partial \Phi}{\partial x} \qquad u_z = \frac{\partial \Phi}{\partial z} \; .$$

It is now time to apply boundary conditions in order to close the system of equations. Consider the following set of equations:

$$\frac{\partial^2\Phi}{\partial x^2}+\frac{\partial^2\Phi}{\partial z^2}=0, \quad -h\leq z\leq \eta(x,t) \tag{2.30a}$$

$$\frac{\partial\Phi}{\partial t}+g\eta=0, \quad z=\eta(x,t) \tag{2.30b}$$

$$\frac{\partial\eta}{\partial t}=\frac{\partial\Phi}{\partial z}, \quad z=\eta(x,t) \tag{2.30c}$$

$$\frac{\partial\Phi}{\partial z}=0, \quad z=-h \tag{2.30d}$$

Equations representing, respectively, Laplace Equation (2.30a), dynamic free-surface boundary condition (2.30b), kinematic free-surface boundary-condition (2.30c) and kinematic bed boundary-condition (2.30d).

Laplace equation (2.30a) can be solved using the method of separation of variables such that

$$\Phi(x,z;t)=X(x)Z(z)T(t). \tag{2.31}$$

Replacing into (2.30a), and supposing that $X(x),\ Z(z),\ T(t)$ are not identically zero,

$$\frac{1}{X}\frac{d^2X}{dx^2}=-\frac{1}{Z}\frac{d^2Z}{dz^2}. \tag{2.32}$$

Since the left-hand side is a function of x alone and the right-hand side is a function of z alone, both sides are equal to a constant. Choosing this constant $-k^2$, traveling waves are obtained:

$$\tfrac{d^2X}{dx^2}+k^2X=0 \quad , \quad \tfrac{d^2Z}{dz^2}+k^2Z=0 \tag{2.33}$$

with general solutions

$$X(x)=A\cos(kx)+B\sin(kx) \quad , \quad Z(z)=Ce^{kz}+De^{-kz}. \tag{2.34}$$

From (2.30d), comes

$$\frac{\partial\phi}{\partial z}=Z'(z)X(x)T(t)=0, \ \ z=-h. \tag{2.35}$$

This implies that $Z'(z)=0$ at $z=-h$, and so, $Z'(-h)=0$. Now, at $z=-h$,

$$Z'(z)=kCe^{kh}-kDe^{-kh}=0. \tag{2.36}$$

Then, $D = Ce^{-2kh}$, and finally comes that

$$\begin{aligned} Z(z) &= Ce^{kz} + De^{-kz} = \frac{C}{e^{kh}}e^{kh+kz} + \frac{Ce^{-2kh}}{e^{kh}}e^{kh}e^{-kz} \\ &= \frac{C}{e^{kh}}\left[e^{k(h+z)} + e^{-2kh}e^{kh}e^{kz}\right] \\ &= \frac{C}{e^{kh}}\left[e^{k(h+z)} + e^{-kh-kz}\right] \\ &= \frac{C}{e^{kh}}\left[e^{k(h+z)} + e^{-k(h+z)}\right] \end{aligned} . \tag{2.37}$$

It is now possible to write $Z(z)$ as follows

$$Z(z) = \frac{2C}{e^{kh}}\left[\frac{e^{k(z+h)} + e^{-k(z+h)}}{2}\right] = \frac{2C}{e^{kh}}\cosh[k(z+h)] = \tau\cosh[k(z+h)], \tag{2.38}$$

where τ is a constant to be determined.

From (2.34) and (2.38), the solution for Φ can be written as

$$\Phi(x,z;t) = \tau\cosh[k(z+h)][A\cos(kx) + B\sin(kx)]T(t), \tag{2.39}$$

where k is the wave number. From the kinematic free surface condition presented on (2.30c) and kinematic free boundary condition (2.30c), we have that

$$\frac{\partial\Phi}{\partial t} = -g\eta, \tag{2.40a}$$

$$\frac{\partial^2\Phi}{\partial t^2} = -g\frac{\partial\eta}{\partial t}, \tag{2.40b}$$

$$\frac{\partial\Phi}{\partial t} = -g\frac{\partial\Phi}{\partial z}, \tag{2.40c}$$

$$T''(t)X(x)Z(z) = -g\, Z'(z)X(x)T(t), \tag{2.40d}$$

$$T''(t)Z(z) = -g\, Z'(z)T(t), \tag{2.40e}$$

$$T''(t) = -g\frac{Z'(z)}{Z(z)}T(t)\ ,\ \text{at}\ \ z = \eta(x,t). \tag{2.40f}$$

Using (2.38),

$$T''(t) = -g\tau\frac{\sinh(k(z+h))}{\cosh(k(z+h))}T(t). \tag{2.41}$$

Finally, at $z = \eta(x,t)$,

$$T''(t) + g\tau \tanh(k(z+h))T(t) = 0 \tag{2.42}$$

under the condition that $X(x) \neq 0$. Let, for $h > 0, \ \forall k$

$$\omega^2 = g\tau \tanh(k(z+h)) > 0. \tag{2.43}$$

The general solution for $T(t)$ is then

$$T(t) = E\cos(\omega t) + F\sin(\omega t). \tag{2.44}$$

It is now possible to write the general expression for Φ for a single value of the separation constant k, from (2.42) and (2.44),

$$\Phi(x,z;t) = \tau\cosh[k(z+h)][A\cos(kx) + B\sin(kx)][E\cos(\omega t) + F\sin(\omega t)]. \tag{2.45}$$

Notice that ω can be identified as the wave frequency. Because k can be known as the wave number, (2.43) is the dispersion relationship (relation between the wave frequency and the wave number). A progressive wave solution can be obtained as

$$\Phi(x,z;t) = A\cosh(k(z+h))\sin(kx - \omega t). \tag{2.46}$$

From (2.30b), the corresponding expression for the free surface elevation is

$$\eta(x,t) = \frac{A\omega}{g}\cosh(kh)\sin(kx - \omega t). \tag{2.47}$$

The amplitude a of the free surface elevation can be introduced as

$$a = \frac{A\omega}{g}\cosh(kh) \tag{2.48}$$

and so,

$$\eta(x,t) = a\cos(kx - \omega t). \tag{2.49}$$

Chapter 3

Initial value problem

A fundamental question that arises in scientific modelling is whether a given differential equation, with its respective initial condition, can be reliably used to predict behaviour of the trajectory at later times. The three attributes of an initial value problem that have to be considered are: whether there actually exists a solution; whether the solution, if it exists, is unique; and, thirdly, how sensitive the solution is to small perturbations to initial conditions. In this chapter, questions about existence and uniqueness of solutions for initial value problems are discussed. Models discussed in previously chapters are particular cases of the general initial value problems presented here.

3.1 Introduction

Let the initial value problem be considered as

$$y'(t) = f(t, y(t)), \quad t \in I \ . \tag{3.1}$$

Given t_0 and y_0 it is intended to find a solution to the Equation (3.1) for $t > t_0$ such that

$$y(t_0) = y_0. \tag{3.2}$$

The system of equations formed by (3.1) and (3.2) define the initial value problem. To guarantee the unicity of the solution, it is necessary to impose conditions on the function $f(t, y)$.

To study the models described in Chapter 2, existence and unicity of solutions of a system of ordinary differential equation will be analyzed [16]. The models are consistent with an initial value problem, that is, consists on two ordinary differential equations plus two initial value conditions. These differential equations describe the system evolution over time.

Models presented can be translated into a system of ordinary differential equations (ODEs) such that

$$\begin{cases} v'(t) = f_1 \\ x'(t) = f_2 \end{cases} , \tag{3.3}$$

where, for the model for small objects, $f_1 = \dfrac{F_\tau + F_{d_\tau}}{m(1+C_m)}$ and $f_2 = \dfrac{v}{\sqrt{1+\left(\frac{\partial\eta}{\partial x}\right)^2}}$. For the cargo containers model, $f_1 = \dfrac{F_{a_\tau} + F_{d_\tau} + F_r}{m^2 fk\Lambda v}$ and $f_2 = v$. This system is completed with the following initial conditions

$$x(t_0) = x_0 \qquad v(t_0) = v_0 \quad . \tag{3.4}$$

3.2 Existence and unicity of solutions

In this section it is intended to make a study a general problem, this is, a problem that can be written as

$$\begin{cases} y'(t) = f(t,y) \\ y(t_0) = y_0 \end{cases} \tag{3.5}$$

where $f = (f_1,\dots,f_n)^T : D \longrightarrow \mathbb{R}^n$ with $D \subset \mathbb{R}\times\mathbb{R}^n$ open and $(t_0,y_0) \in D$.

In order to establish results of existence and unicity of solutions to initial value problems some results and definitions are considered [16].

Definition 3.2.1. *A function $\phi(t)$ is a solution of the initial value problem (3.5) if $\phi(t)$ is differentiable in an interval J that contains t_0 as an interior point such that $(t,\ \phi(t)) \in D$, $\phi'(t) \equiv f(t,\ \phi(t))$ in J, and still $\phi(t_0) = y_0$.*

Lemma 3.2.1. *Let f be such that $f \in C(D,\mathbb{R}^n)$. So, $y(t)$ is a solution of the initial value problem (3.5) if and only if $v(t)$ is solution of the integral*

$$y(t) = y_0 + \int_{t_0}^{t} f(s,y(s))ds \tag{3.6}$$

where

$$\int_{t_0}^{t} f\ ds = \left(\int_{t_0}^{t} f_1\ ds,\dots,\int_{t_0}^{t} f_n\ ds\right)^T. \tag{3.7}$$

Proof. Let $y(t)$ be a solution of (3.5). Then,

$$y'(t) \equiv f(t,\ y(t)), \quad y(t_0) = y_0.$$

It comes

$$y(t) - y_0 = \int_{t_0}^{t} f(s,\ y(s))ds$$

So, $y(t)$ is a solution of the equation (3.6). Supposing then that $y(t)$ is a solution of the equation (3.6), $f(s,\ y(s))$ can be integrated from t_0 to t for any t in the domain. Also, $\int_{t_0}^{t} f(s,\ y(s))ds$ is continuous at t. From (3.6), $y(t)$ is continuous. Besides that, since $f \in C(D,\mathbb{R}^n)$, $f(t,\ y(t))$ is continuous, the right member of (3.6) is differentiable and $y(t)$ too. Differentiating both members of (3.6), becomes

$y'(t) \equiv f(t,\ y(t))$. Notice that $y(t_0) = y_0$.
Then, $y(t)$ is a solution of (3.5). □

Definition 3.2.2 (Lipschitz condition). *Let f such that $f : D \subset \mathbb{R} \times \mathbb{R}^n \longrightarrow \mathbb{R}^n$ and f is said Lipschitz in $v \in D$ if $\exists\ k > 0$, with k constant, such that*

$$|f(t,y_1) - f(t,y_2)| \le k\ |y_1 - y_2| \tag{3.8}$$

for any $(t,y_1),(t,y_2) \in D$. Also, f is locally Lipschitz in $x \in D$ if $\forall (t^*,y^*) \in D$, there ir a neighborhood $\mathscr{N}^*$ of (t^*,y^*) in D and a constant $k^* > 0$ such that

$$|f(t,y_1) - f(t,y_2)| \le k^* |y_1 - y_2| \tag{3.9}$$

for any $(t,y_1),(t,y_2) \in \mathscr{N}^*$.

Lemma 3.2.2 (Gronwall inequality). *Let $M \in \mathbb{R}$, $h \in C([t_0,\infty),\mathbb{R}_+)$ for some $t_0 \in \mathbb{R}$ and $u(t)$ continuous solution of the following inequality*

$$u(t) \le M\ +\ \int_{t_0}^{t} h(s)u(s)ds, \quad t \ge t_0. \tag{3.10}$$

So,

$$u(t) \le M\ e^{\int_{t_0}^{t}\ h(s)ds}, \quad t \ge t_0. \tag{3.11}$$

Proof. Let $r(t) = M + \displaystyle\int_{t_0}^{t} h(s)u(s)ds$.
Then, for $t \ge t_0$, $r'(t) = h(t)u(t) \le h(t)r(t)$, that is $r'(t) - h(t)r(t) \le 0$. Multiplying both sides by $e^{-\int_{t_0}^{t} h(s)ds}$, results that

$$\left(r(t)\ e^{-\int_{t_0}^{t} h(s)ds} \right)' \le 0$$

and so, $r(t)\ e^{-\int_{t_0}^{t} h(s)ds}$ is decreasing. Also, for $t \ge t_0$,

$$r(t)e^{-\int_{t_0}^{t} h(s)ds} \le r(t_0) = M.$$

It turns out that

$$r(t) \le M\ e^{-\int_{t_0}^{t} h(s)ds}.$$

We conclude what is intended since $u(t) \le r(t)$. □

Corollary 3.2.1. *Let $h \in C([t_0,\infty),\mathbb{R}_+)$ for some $t_0 \in \mathbb{R}$ and $u(t) \ge 0$ continuous solution of the following inequality*

$$u(t) \le \int_{t_0}^{t} h(s)u(s)ds\ , \quad t \ge t_0. \tag{3.12}$$

Then, $u(t) \equiv 0,\ \ t \ge t_0$.

Lemma 3.2.3. *1. Let $f_m \in C([a,b],\mathbb{R}^n)$ for $m = 1,2,\dots$ and the set of functions $\{f_m(t)\}_{m=1}^{\infty}$ uniformly convergent for $f(t)$ in $[a,b]$. Then, $f \in C([a,b],\mathbb{R}^n)$, and*

$$\lim_{m\longrightarrow\infty}\int_a^b f_m(t)dt = \int_a^b f(t)dt. \tag{3.13}$$

2. Let $f_m \in C([a,b],G)$ for $m = 1,2,\dots$, where G is a compact subset of $\mathbb{R}^n$, and the subset of functions $\{f_m(t)\}_{m=1}^{\infty}$ is uniformly convergent for $f(t)$ in $[a,b]$. So, $\forall g \in C(G,\mathbb{R}^n)$, the subset of functions $\{g(f_m(t))\}_{m=1}^{\infty}$ is uniformly convergent for $g(f(t))$ in $[a,b]$.

Observation 3.2.1. *Similarly, point 2 of the Lemma (3.2.3) can be generalized as follows:*

For $g \in C([a,b]\times G,\mathbb{R}^n)$, the set of functions $\{g(t,f_m(t))\}_{m=1}^{\infty}$ is uniformly convergent for $g(t,f(t))$ in $[a,b]$.

Lemma 3.2.4 (Weierstrass M-test). *Consider the constants M_m, $m = 1,2,\dots$ such that*

$$\begin{aligned} &|g_m(t)| \le M_m,\ \ \forall t \in [a,b]\ ,\ \ m = 1,2,\dots \\ &\textstyle\sum_{m=1}^{\infty} M_m \ \ convergente \end{aligned} \quad . \tag{3.14}$$

So, $\displaystyle\sum_{m=1}^{\infty} g_m(t)$ is uniformly convergent on $[a,b]$.

Lemma 3.2.5. *Let*

$$G = \{(t,y) : |t-t_0| \le a,\ |y-y_0| \le b\} \tag{3.15}$$

and $f \in C(G,\mathbb{R}^n)$ from Lipschitz in y on the domain G. The initial value problem (3.5) has a unique solution for $|t-t_0| \le \gamma$ where $\gamma = \min\{a\ ,\ \dfrac{b}{M}\}$ and $M = \max_{(t,y)\in G}|f(t,y)|$. The result can be translated by the following figure:

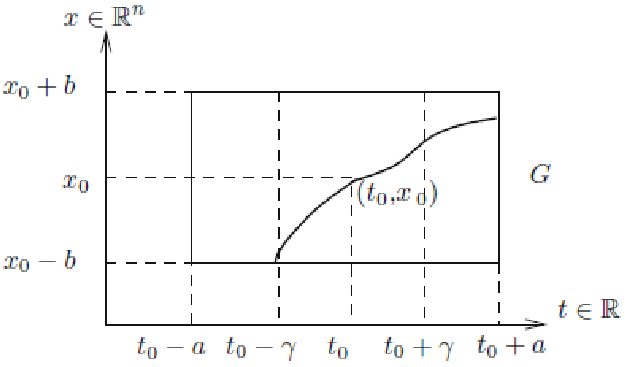


Fig. 3.1 Rectangular domain

Proof. Notice that (3.5) is equivalent to the integral (3.6).

- Unicity is proven first:
  Let (3.5) have two solutions $y_1(t)$ and $y_2(t)$. From (3.2.1), these solutions are also solutions of

the integral (3.6), that is

$$y_i(t) = y_0 + \int_{t_0}^{t} f(s, y_i(s))ds, \quad i = 1,2. \tag{3.16}$$

Now, subtracting the equations for $i = 1,2$ and applying the Lipschitz condition (3.2.2), comes, for $t \geq t_0$

$$\begin{aligned} |y_1(t) - y_2(t)| &= \left| \int_{t_0}^{t} (f(s, y_1(s))) - (f(s, y_2(s)))ds \right| \\ &\leq \int_{t_0}^{t} |(f(s, y_1(s))) - (f(s, y_2(s)))|ds \\ &\leq \int_{t_0}^{t} k|y_1(s) - y_2(s)|ds, \end{aligned} \tag{3.17}$$

where k is the Lipschitz constant. Still, let $u(t) = |y_1(t) - y_2(t)|$. Results that $u(t) \leq \int_{t_0}^{t} ku(s)ds$ for $t \geq t_0$. From Gronwall's inequality (3.2.1), $u(t) \equiv 0$ for $t \geq t_0$. Similarly, it can be proved that $t < t_0$, for the left side of Gronwall's inequality (3.2.2). Becomes proved that $y_1(t) \equiv y_2(t)$, and so, if existing, the solution is unique.

- Existence is now proven:
  A sequence of functions $\{y_m(t)\}_{m=0}^{\infty}$ is defined by recurrence as follows

$$\begin{aligned} &y_0(t) = y_0 \\ &y_{m+1}(t) = y_0 + \int_{t_0}^{t} f(s, y_m(s))ds, \quad m \in \mathbb{N}_0 := 0,1,2,\ldots \end{aligned} \tag{3.18}$$

Let $y_m(t)$ be defined in the interval $J := [t_0 - \gamma, t_0 + \gamma]$ and still

$$|y_m(t) - y_0| \leq b \, , \quad m \in \mathbb{N}_0 \tag{3.19}$$

and $(t, y_m(t)) \in G$ for $t \in J$. This is trivially verified for $m = 0$. Suppose true for some $m \in \mathbb{N}_0$. Then, $(t, y_m(t)) \in G$ for $t \in J$. From (3.18), $y_{m+1}(t)$ exists in J and

$$|y_{m+1}(t) - y_0| = \left| \int_{t_0}^{t} f(s, y_m(s))ds \right| \leq M|t - t_0| \leq M\gamma \leq b. \tag{3.20}$$

This is, $(t, y_{m+1}(t)) \in G$ for $t \in J$.
It is now shown that $y_m(t)_{m=0}^{\infty}$ is uniformly convergent in J. For that case, rewrite $y_m(t)$ as following

$$y_m(t) = y_0 + \sum_{i=0}^{m-1} (y_{i+1}(t) - y_i(t)), \quad m = 1,2,\ldots \tag{3.21}$$

By induction, we have $t \in J$ and $i = 0, \dots, m-1$

$$|y_{i+1}(t) - y_i(t)| \leq \frac{M\,k^i |t-t_0|^{i+1}}{(i+1)!} \leq \frac{M\,k^i\,\gamma^{i+1}}{(i+1)!}. \tag{3.22}$$

Clearly, (3.22) is true for $i = 0$. Assume (3.22) valid for some $i \in \mathbb{N}$. So, from (3.18) and using Lipschitz condition, for $t \in J : t \geq t_0$,

$$\begin{aligned} &|y_{i+2}(t) - y_{i+1}(t)| \leq \int_{t_0}^{t} |f(s, y_{i+1}(s)) - f(s, y_i(s))| ds \\ &\leq k \int_{t_0}^{t} |y_{i+1}(s) - y_i(s)| ds \leq k \int_{t_0}^{t} \frac{M\,k^i (s-t_0)^{i+1}}{(i+1)!}\, ds \\ &= \frac{M\,k^{i+1}(t-t_0)^{i+2}}{(i+2)!} \leq \frac{M\,k^{i+1}\gamma^{i+2}}{(i+2)!}. \end{aligned} \tag{3.23}$$

Similarly, for $t \in J$ and $t < t_0$. So, (3.22) is verified by $i+1$. Since the series $\sum_{i=1}^{\infty}(y_i(t) - y_{i-1}(t))$ is convergent, using the Weierstrass M-test (3.2.4), the series of functions $\sum_{i=1}^{\infty}(y_i(t) - y_{i-1}(t))$ is uniformly convergent in J. Let $y(t)$ be the uniform limit of $\{y_m(t)\}_{m=0}^{\infty}$. By the Lemma (3.2.3) and the Observation (3.2.1), $y(t)$ s continuous in J and $f(t, y(t))$ uniformly convergent for $f(t, y(t))$ in J. So, $f(t, y(t))$ is continuous in J. Taking limits in $m \longrightarrow \infty$, we obtain

$$y(t) = y_0 + \int_{t_0}^{t} f(s, y(s)) ds, \qquad t \in J \tag{3.24}$$

ithat is, $y(t)$ is solution from (3.6) and consequently, solution of (3.5).

□

Theorem 3.2.1 (Picard's existence and unicity)**.** *Let D be an open subset $\mathbb{R}^n \times \mathbb{R}^n$ and $(t_0, y_0) \in D$. Suppose that $f \in C(D, \mathbb{R}^n)$ is locally Lipschitz (3.2.2) in $y \in D$. Then, there is $\gamma > 0$ such that (3.5) has a unique solution that exists for $|t - t_0| \leq \gamma$.*

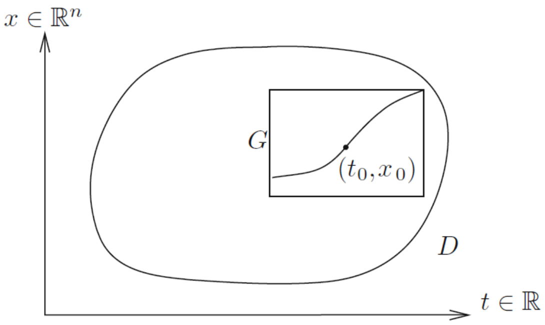


Fig. 3.2 Open domain

Proof. Since D is an open set and $(t_0, y_0) \in D$, then there are $a, b > 0$ such that G, defined by (3.15), is contained in D (see Figure (3.2)). As f is locally Lipschitz in $x \in D$, a and b can be chosen small enough that the Lipschitz condition (3.2.2) with a fixed constant $k > 0$ is verified in G. By the Lemma 3.2.5, the intended conclusion is concluded. □

Chapter 4

Numerical methods

Suppose that a particle is moving in such a way that, at time t_0, its position is equal to y_0 and that, at this time, the velocity is known to be v_0. The simple principle of Euler's method is that, in a short period of time (so short that there is no time for velocity to change significantly from v_0), the change in position will be approximately equal to the change in time multiplied by v_0. If the motion of the particle is governed by a differential equation, the value of v_0 will be known as a function of x_0 and y_0 [4]. In this chapter, Runge-Kutta method is presented and consistency, error bounds and stability are studied for this method.

4.1 Runge-Kutta method

The Runge-Kutta method was born to allow a number of evaluations of the derivative to take place in a step, i.e., to be a generalization of Euler's method. Consider that the initial value problem, as described before,

$$y'(t) = f(t,y), \qquad y(t_0) = y_0 \ , \tag{4.1}$$

has solution

$$y(t) = y(t_0) + \int_{t_0}^{t} f(s,y(s))ds. \tag{4.2}$$

To obtain a numerical solution to this problem, defined in $[t_0,t_M]$, the interval is divided in equidistant points such that

$$t_k = t_0 + kh, \;\; k = 0,1,...N, \;\; h = \frac{t_M - t_0}{N}. \tag{4.3}$$

For each t_k it is intended to find a numerical approximation for $y(t_k)$. Such approximations are calculated successively.

Let y_n be an approximation of the exact solution $y(t_k)$, with $k = 0,\ldots,M$ and so, $t_M = T$. The numerical methods of one-step can be obtained approximating the integral in the right-hand-side of (4.2).

The explicit one-step numerical methods can be obtained using the approximation y_n such that

$$y_0 = y(t_0) \tag{4.4a}$$

$$y_{k+1} = y_n + h\phi(t_k, y_n, h), \;\; k = 0, \dots, M-1 \tag{4.4b}$$

where $\phi : [t_0, T] \times \mathbb{R}^n \times (0,\ h_0] \longrightarrow \mathbb{R}^n$.

To use the implicit, the method can be written as

$$y_0 = y(t_0) \tag{4.5a}$$

$$y_{k+1} = y_n + h\phi(t_k, y_n, y_{k+1}, h), \;\; k = 0, \dots, M-1 \tag{4.5b}$$

where $\phi : [t_0, T] \times \mathbb{R}^n \times (0,\ h_0] \longrightarrow \mathbb{R}^n$.

The numerical method that is going to be used is the Runge-Kutta method.

Definition 4.1.1. *Let s be an integer (that represents the number of stages) and $a_{21}, a_{31}, a_{32}, \dots, a_{s1}$, $a_{s2}, \dots, a_s, a_{s-1}, b_1, \dots, b_s, c_2, \dots, c_s$ real coefficients. Then, the method*

$$\begin{aligned}
k_1 &= f(t_0,\ y_0) \\
k_2 &= f(t_0 + c_2 h,\ y_0 + h a_{21} k_1) \\
k_3 &= f(t_0 + c_3 h,\ y_0 + h(a_{31} k_1 + a_{32} k_2)) \\
&\dots \\
k_s &= f(t_0 + c_s h,\ y_0 + h(a_{s1} k_1 + \dots + a_{s,s-1} k_{s-1})) \\
y_1 &= y_0 + h(b_1 k_1 + \dots + b_s k_s)
\end{aligned} \tag{4.6}$$

is called an s-stage of the explicit Runge-Kutta method for (4.1).

According to the previous definition,

$$\phi(t_k, y_n, y_{k+1}, h) = \sum_{j=1}^{s} b_j k_j \tag{4.7}$$

where $k_j = f(t_j + c_j h, y_j + h\sum_{i=1}^{s} a_{ji} k_i)$, with $j = 1, \dots, s$.

Next, the way to obtaind the family of Runge-Kutta methods is described. This description is done based on 2nd order Runge-Kutta method.

Taylor's Method at order 2 is based on the following expression

$$y(t+h) = y(t) + hy'(t) + \frac{h^2}{2} y''(t) + T_2(h) \tag{4.8}$$

where $T_2(h)$ represents the truncature error, that is studied on the next section.

This expression can be rewritten as

$$y(t+h) \approx y + hf + \frac{h^2}{2}(f_t + f_y f) \tag{4.9}$$

or, also,

$$y(t+h) \approx y + \frac{h}{2}f + \frac{h}{2}(f + hf_t + (hf)f_y). \tag{4.10}$$

The expression in curved brackets corresponds to a Taylor expansion of f, centred on (t,y) and, along the (h,hf) which allows to circumvent the use of partial derivatives f_t, f_y.

Considering the Taylor's formula of order 1 with Lagrange Remainder [19], comes that

$$f(t+h,y+hf) = f(t,y) + hf_t(t,y) + (hf)f_y(x,y) + T_1(h) = f + hf_t + (hf)f_y + T_1(h) \tag{4.11}$$

and so, the expression can be written as

$$y(t+h) = y + \frac{h}{2}f + \frac{h}{2}f + (t+h,y+hf). \tag{4.12}$$

Runge-Kutta method family of order 2 can be obtained introducing parameters in the expression (4.10) such that

$$y(t+h) \approx y + \omega_1 h_f + \omega_2 h(f + \alpha h f_t + \beta h f f_y). \tag{4.13}$$

Such parameters must satisfy

$$\omega_1 + \omega_2 = 1 \tag{4.14a}$$

$$\omega_2 \alpha = \frac{1}{2} \tag{4.14b}$$

$$\omega_2 \beta = \frac{1}{2}. \tag{4.14c}$$

Heun method can so be obtained doing $\omega_1 = \omega_2 = \frac{1}{2}$ and $\alpha = \beta = 1$.

The second order methods can be written as following

$$y_{k+1} = y_n + h(\omega_1 k_1 + \omega_2 k_2); \tag{4.15a}$$

$$k_1 = f(t_k, y_n); \tag{4.15b}$$

$$k_2 = f(t_k + h\alpha, y_n + h\beta k_1). \tag{4.15c}$$

Regarding the 4th order Runge-Kutta method, it is deducted in the same way but has two extra terms, i.e.,

$$y_{k+1} = y_n + h(\omega_1 k_1 + \omega_2 k_2 + \omega_3 k_3 + \omega_4 k_4) \tag{4.16}$$

that are forced to agree with the first 5 terms of the Taylor method. The resulting formula is as follows

$$y_{k+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k4); \tag{4.17a}$$

$$k_1 = f(t_k, y_n) \tag{4.17b}$$

$$k_2 = f(t_k + \frac{h}{2}, y_n + \frac{h}{2}k_1) \tag{4.17c}$$

$$k_3 = f(t_k + \frac{h}{2}, y_n + \frac{h}{2}k_2) \tag{4.17d}$$

$$k_4 = f(t_k + h, y_n + hk_3). \tag{4.17e}$$

4.2 Consistency and global error bound

In this section, consistency and global error bound are studied. Errors generated after many steps of the method will accumulate and reinforce each other. It is important to understand how this happens and bound the global error using a difference inequality. To solve the initial value problem presented in (4.1) having a one-step method, its explicit numerical method is given by

$$y_{k+1} = y_n + h\Phi(t_k, y_n; h), \quad 0 \leq k \leq N-1 \tag{4.18}$$

where $y_0 = y_0$ and Φ is called the increment function.

The global error of this method is then given by

$$E_k = y(t_k) - y_n. \tag{4.19}$$

Still, the truncation error, $T_k(h)$ is given by

$$T_k(h) = y(t_{k+1}) - y(t_k) - h\Phi(t_k, y(t_k); h), \quad 0 \leq k \leq N-1. \tag{4.20}$$

So,

$$\frac{y(t_{k+1}) - y(t_k)}{h} = \Phi(t_k, y(t_k); h) + \frac{T_k(h)}{h}, \tag{4.21}$$

and, the global truncation error is given by

$$T(h) = \max_{0 \leq k \leq N-1} |T_k(h)|. \tag{4.22}$$

A requirement for our method to be convergent is that $\lim_{h\to 0} \frac{T(h)}{h} = 0$ with

$$\lim_{h\to 0} \frac{y(t_{k+1}) - y(t_k)}{h} = y'(t_k) \tag{4.23a}$$

$$\lim_{h\to 0} \Phi(t_k, y(t_k); h) = f(t_k, y(t_k)). \tag{4.23b}$$

In this case, the method is said consistent with the initial value problem [28]. Still, the method is said consistent of order p if $\frac{T(h)}{h} \leq Ch^p$ where C is a real constant and independent from h.

Theorem 4.2.1. *A one-step explicit method is consistent if, and only if,*

$$\lim_{h\to 0} \max_{0\leq k\leq N-1} |\Phi(t_k, y_n; h) - f(t_k, y_n)| = 0. \tag{4.24}$$

Theorem 4.2.2. *Let the initial value problem be approximated by (4.18) with $y_0 = y_0$.*

- *Let Φ be a Lipschitz continuous function, that is, there is a constant L such that for $0 \leq h \leq h_0$ and for any (t_k, y_n) and (t_k, z_k),*

$$\|\Phi(t_k, y_n; h) - \Phi(t_k, z_k; h)\| \leq L\|y_n - z_k\|. \tag{4.25}$$

- *The method is consistent of order p, i.e.,*

$$\frac{T(h)}{h} \leq Ch^p.$$

If both conditions are verified, then the global error is limited by

$$\|E_k\| \leq \|y(t_k) - y_n\| \leq \left(\frac{e^{L(t_k - t_0)} - 1}{L}\right) Ch^p, \quad k = 0, 1, \ldots, M, \quad L > 0. \tag{4.26}$$

When $L = 0$, $E_k \leq (t_k - t_0)Ch^p$.

Proof. Suppose that both items of the Theorem are verified and also $E_0 = 0$ since $y_0 = y_0$. Let y_{k+1} be

$$y_{k+1} = y_n + h\Phi(t_k, y_n; h). \tag{4.27}$$

Replacing the exact solution,

$$y(t_{k+1}) = y(t_k) + h\Phi(t_k, y(t_k); h) + T_k(h). \tag{4.28}$$

Subtracting both equalities,

$$y(t_{k+1}) - y_{k+1} = y(t_k) - y_n + h[\Phi(t_k, y(t_k); h) - \Phi(t_k, y_n; h)] + T_k(h). \tag{4.29}$$

Taking norms,

$$\|E_{k+1}\| \leq \|E_k\| + h\|\Phi(t_k, y(t_k); h) - \Phi(t_k, y_n; h)\| + \|T_k(h)\|. \tag{4.30}$$

By the hypothesis,

$$\|E_{k+1}\| \leq \|E_k\| + hL\|y(t_k) - y_n\| + Ch^{p+1}. \tag{4.31}$$

Recursively applying this equality and adding the resultant geometric sum,

$$\|E_{k+1}\| \leq \frac{(1+hL)^{k+1} - 1}{L} Ch^p. \tag{4.32}$$

For any $t \geq 0$, $e^t \geq 1 + t$, and so, $1 + hl \leq e^{hL}$. Also, $(k+1)h = t_{k+1} - t_0$. From this,

$$\|E_{k+1}\| \leq \frac{e^{L(t_{k+1}-t_0)} - 1}{L} Ch^p. \tag{4.33}$$

Finally,

$$\|E_k\| \leq \frac{e^{L(t_k-t_0)} - 1}{L} Ch^p. \tag{4.34}$$

□

4.3 Stability of Runge-Kutta methods

A lot can be said about the qualitative behaviour of dynamical systems, looking at the local solution behaviour in the neighbourhood of some equilibrium points. In this chapter the stability of these points and the related stability of Runge-Kutta methods will be studied. The majority of definitions and results presented in this section are based in [9] and [12].

4.3.1 A-Stability

It is crucial to define, in advance, the stability of an equilibrium point. In general, the stability concerns the behaviour of solutions near an equilibrium point in the long term. A system of differential equations, for which $f(t,y)$ can be written as a function of y, is only an autonomous differential equation. Equations of this type are invariant under a translation of time, i.e., if $\tau = t - t_0$, then, $\frac{d\tau}{dt} = 1$, and also

$$\frac{dy}{d\tau} = \frac{dy}{dt}\frac{dt}{d\tau} = \frac{dy}{dt}. \tag{4.35}$$

Without loss of generality the initial value of time can be assumed to be $t = 0$, as long as the system is autonomous, i.e., the initial value problem can be replaced by

$$\frac{dy}{d\tau} = f(y), \qquad y(0) = y_0 \ . \tag{4.36}$$

Considering the autonomous system of differential equations (4.35), with $y(t, y_0)$ the solution of the system with the initial condition $y(t_0) = y_0$, some definitions are now given.

Definition 4.3.1 (Stability in the sense of Lyapunov)**.** *An equilibrium point, say y^*, is stable in the sense of Lyapunov (or just stable) for the given differential equation if,* $\forall \varepsilon > 0$*,* $\exists \delta > 0$ *and* $T > 0$

such that, if $\|y_0 - y^*\| < \delta$*, then, for* $t > T$*,*

$$\|y(t,y_0) - y*\| < \varepsilon. \tag{4.37}$$

Definition 4.3.2 (Asymptotic stability)**.** *An equilibrium point,* y^**, is asymptotically stable if there exists* $\gamma > 0$ *such that, for any initial condition* y_0*, such that* $\|y_0 - y^*\| < \gamma$

$$\lim_{t\to\infty} \|y(t,y_0) - y*\| = 0. \tag{4.38}$$

Definition 4.3.3. *An equilibrium point,* y^**, is unstable if it is not stable (in the sense of Lyapunov).*

Lemma 4.3.1. *Any asymptotically stable equilibrium point is also stable. The opposite is not necessarily true.*

The next result is the Linearization Theorem.

Theorem 4.3.1. *[9][12][24] Consider*

$$\frac{dy}{dt} = Ay + F(y) \tag{4.39}$$

in $\mathbb{R}^d$ *such that* $y(0) = y_0$ *states the initial condition to which the equation is subjected. Also,* A *is a constant* $d \times d$ *matrix with* λ_i *(eigenvalues) having all negative real parts and* F *is* C^1 *in a neighbourhood of* $y = 0$*, with* $F'(0) = F(0) = 0$*, where* $F'(y)$ *is the Jacobian matrix of* F*. Then,* $y = 0$ *is an asymptotically stable equilibrium point and if* A *has any eigenvalues with positive real part, then* $y = 0$ *is an unstable equilibrium point.*

Despite the restrictions of A and F, this theorem allows to take some conclusions.

Having y^* an equilibrium point which is not at the origin, it is always possible to shift it to the origin by introducing a translation. Defining $\tilde{y} = y - y^*$, the following comes

$$\frac{d\tilde{y}}{dt} = \frac{dy}{dt} = f(\tilde{y} + y^*) \equiv \tilde{f}(\tilde{y}). \tag{4.40}$$

This has the same form as the initial equation but has an equilibrium point at $\tilde{y} = 0$.

Considering the differential equation (4.36) with C^2 vector field and an equilibrium point at the origin, Theorem 4.3.1 is allowed to be used following the steps:

1. Apply Taylor series to f around 0: $f(y) = f(0) + f'(0)y + R(y)$. Because 0 is an equilibrium point, $f(0) = 0$;

2. Set $A = f'(0)$;

3. Define $F(y) = R(y)$.

The equation can now be written in the form

$$\frac{dy}{dt} = Ay + F(y). \tag{4.41}$$

It is trivial to conclude that $F(0) = F'(0) = 0$.
If the eigenvalues of A have negative real parts, then we can conclude that the origin is asymptotically stable. This leads to a reformulation of Theorem 4.3.1 [12].

Theorem 4.3.2. *[24] Suppose that f in (4.36) is C^2 and has an equilibrium point, y^*.*

If the eigenvalues of the Jacobian matrix, $J = f'(y^*)$, all lie strictly in the left complex half-plane, then the equilibrium point, y^*, is asymptotically stable.

If J has any eigenvalue in the right complex half-plane, then y^* is an unstable point.

The stability theory is developed based on test problems and so, the linear case can be studied to introduce this theory.

For a linear system of ordinary differential equations,

$$\frac{dy}{dt} = Ay, \tag{4.42}$$

where A is a $d \times d$ matrix with a basis of eigenvectors. The general solution can be written in the compact form

$$y(t) = \sum_{i=1}^{d} C_i e^{\lambda_i t} y_i, \tag{4.43}$$

with λ_i are the eigenvalues, y_i the corresponding eigenvectors and C_i are coefficients. It is possible, from this, to conclude that stability is determined by the eigenvalues.

Consider the problem (4.42). Applying Euler's method,

$$y_{n+1} = y_n + hAy_n = (I + hA)y_n.$$

If y_n is expanded in the eigenvalues y_i, $i = 1, \ldots, d$,

$$y_n = \alpha_1^{(n)} y_1 + \cdots + \alpha_d^{(n)} y_d.$$

Then,

$$y_{n+1} = (I + hA)\left(\alpha_1^{(n)} y_1 + \cdots + \alpha_d^{(n)} y_d\right).$$

Therefore,

$$y_{n+1} = \alpha_1^{(n)}(I + hA)y_1 + \cdots + \alpha_d^{(n)}(I + hA)y_d.$$

If y_i is an eigenvector of A, then

$$(I + hA)y_i = y_i + hAy_i + hAy_i = y_i + h\lambda_i y_i = (1 + h\lambda_i)y_i.$$

This implies

$$y_{n+1} = \sum_{i=1}^{d} \alpha_i^{(n)} (1 + h\lambda_i) y_i.$$

It is also possible to write

$$y_{n+1} = \sum_{i=1}^{s} \alpha_i^{(n+1)} y_i.$$

Comparing the last two equations and using the uniqueness of the basis representation [25],

$$\alpha_i^{(n+1)} = (1 + h\lambda_i)\,\alpha_i^{(n)}.$$

Origin is a stable point if $|1+\lambda_i| \leq 1$ and is asymptotically stable if $|1+\lambda_i| < 1$.

The stability condition for linear case of Euler method is given by [1]: For every eigenvalue λ of A, $h\lambda$ must lie inside a disk of radius 1 centred at $z = -1$ in the complex plan (see Figure 4.1).

Given the set of eigenvalues of A, the above condition implies a restriction on the maximum step size h.

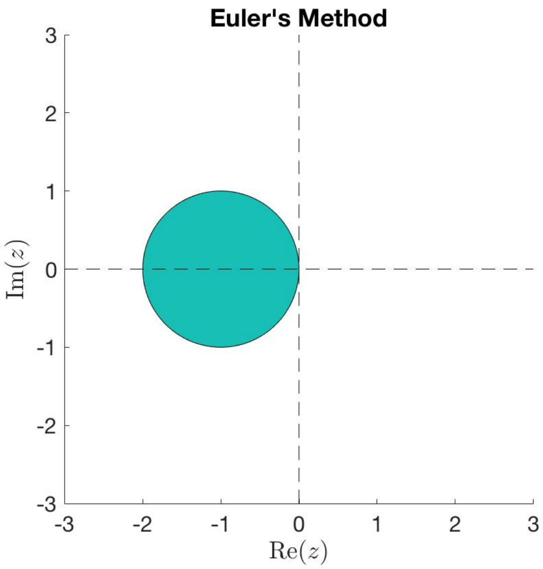


Fig. 4.1 Region of absolute stability of Euler's method

In order to develop a theory applicable to most cases, the scalar case is introduced. It is possible to see that Runge-Kutta methods, when applied to $y' = \lambda y$ can be written in the form

$$y_{n+1} = R(h\lambda)y_n, \tag{4.44}$$

where $R(\mu)$ is a rational function of μ.

Considering the Runge-Kutta method applied to a linear test problem $y' = \lambda y$, for the internal stages, with $i = 1, \ldots, s$, the following relations are obtained

$$Y_i = y_n + \mu \sum_{j=1}^{s} a_{ij} Y_j. \tag{4.45}$$

This system can be casted into a matrix form with $Y = (Y_1, \ldots, Y_s)^T$, $\mathbf{1} = (1, \ldots, 1)^T$ and A the matrix with entries a_{ij}, i.e,

$$Y = y_n\mathbf{1} - \mu AY. \tag{4.46}$$

This means that

$$Y = y_n(I - \mu A)^{-1}\mathbf{1} \tag{4.47}$$

and similarly,

$$y_{n+1} = y_n + \mu\sum_{j=1}^{s} b_j Y_j = y_n + \mu b^T Y. \tag{4.48}$$

The stability function R can be finally obtained from the combination of these three expressions:

$$y_{n+1} = R(\mu)y_n \tag{4.49a}$$

$$R(\mu) = 1 + \mu b^T(I - \mu A)^{-1}\mathbf{1} \tag{4.49b}$$

From Cramer's rule [17] computation of the inverse of a matrix, R is a rational function in μ.

Applying now the general Runge-Kutta method to $y' = Ay$,

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i A Y_i, \tag{4.50}$$

where

$$Y_i = y_n + h\sum_{j=1}^{s} a_{ij} Y_j. \tag{4.51}$$

The expansion of y_n, y_{n+1} and Y_i in the linearly independent set of d eigenvectors of A (eigenbasis) is now applied.

Let U be a matrix composed with the eigenvectors as its columns so that $AU = U\Lambda$ where Λ is the diagonal matrix of eigenvectors. This means that $U^{-1}AU = \Lambda$ is the diagonalization of A.

Also, let Z_i and z_n be defined as follows

$$y_n = Uz_n, \quad Y_i = UZ_i, \quad i = 1, \dots, s. \tag{4.52}$$

The Runge-Kutta method (4.17)can now be rewritten as

$$Uz_{n+1} = Uz_n + h\sum_{i=1}^{s} b_i AUZ_i \tag{4.53}$$

or,

$$z_{n+1} = z_n + h\sum_{i=1}^{s} b_i \Lambda Z_i. \tag{4.54}$$

Similarly,

$$UZ_i = Uz_n + h\sum_{i=1}^{s} a_{ij} AUZ_j \tag{4.55}$$

or,

$$Z_i = z_n + h\sum_{i=1}^{s} a_{ij}\Lambda Z_j. \tag{4.56}$$

Since Λ is a diagonal matrix, the system can be completely decoupled into d independent scalar iterations. This can be translated as the application of the same Runge-Kutta method to the d scalar, and complex, differential equations

$$\frac{dz^{(i)}}{dt} = \lambda_i z^{(i)}$$

with $y = Uz$ and $z^T = (z^{(1)}, \ldots, z^{(d)})$.

After all these considerations, it is now possible to see that stability of the origin can be analysed for a given numerical method by just considering the scalar problem.

Theorem 4.3.3. *[12] Given the differential equation $y' = Ay$ where it is assumed that the $d \times d$ matrix A has a basis of eigenvectors and the corresponding eigenvalues λ_i, consider applying a Runge-Kutta method. The Runge-Kutta method has a stable (asymptotically stable) fixed point at the origin when applied to the differential equation if, and only if, the same method has a stable (asymptotically stable) equilibrium at the origin when applied to each of the scalar differential equations $\frac{dy}{dt} = \lambda y$, with λ an eigenvalue of A.*

A Runge-Kutta method applied to the scalar differential equations $\frac{dy}{dt} = \lambda y$ can be written as

$$y_{n+1} = R(h\lambda)y_n. \tag{4.57}$$

The following corollary is obtained.

Corollary 4.3.1. *Consider a linear differential equation $y' = Ay$ with diagonalizable matrix A. Let a Runge-Kutta method be given with stability function R. The origin is stable for the numerical method applied to the differential equation (at step size h) if and only if*

$$|R(\mu)| \leq 1,\ \forall \mu = h\lambda \tag{4.58}$$

with λ an eigenvalue of A.

Definition 4.3.4. $\{\mu : |R(\mu)| \leq\}$ *is the stability region of the Runge-Kutta method.*

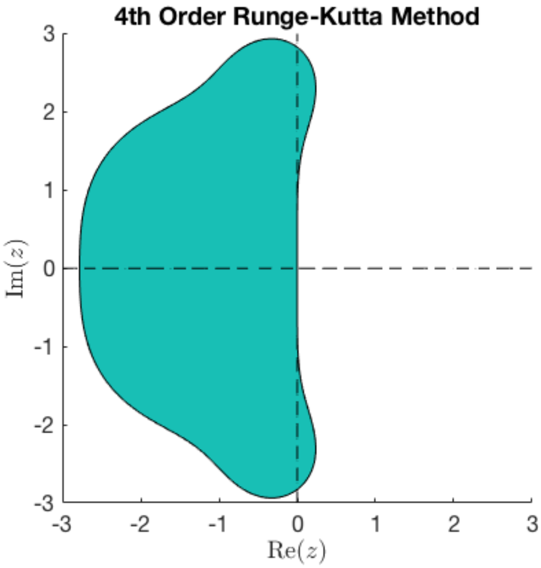


Fig. 4.2 Region of absolute stability of 4th order Runge-Kutta method

Through fourth order, $R(h\lambda)$ agrees with the Taylor series expansion of $\exp(h\lambda)$. The stability function in this case is given by

$$R(\mu) = 1 + \mu + \frac{1}{2}\mu^2 + \frac{1}{6}\mu^3 + \frac{1}{24}\mu^4. \tag{4.59}$$

The concept of absolute stability [18][15] needs to be considered. As said before, the construction of stability is related with the following equation

$$y'(t) = \lambda y(t). \tag{4.60}$$

This very simple equation is used as a model to predict the stability behaviour of a numerical methods for general nonlinear systems [9].

Any Runge-Kutta method applied to (4.60) is reduced to the representation of (4.57) in which $R : \mathbb{C} \to \mathbb{C}$ is a polynomial or rational function with real coefficients.

Definition 4.3.5. *A Runge-Kutta method is called absolutely stable at $\mu \in \mathbb{C}$ if, for this μ, $|R(z)| \leq 1$.*

This implies that $|y_{n+1}| \leq |y_n|$ for any (h,λ) pair such that $h\lambda = \mu$.

Let the following implicit relation represent the one-step Runge-Kutta method for the numerical integration of the initial value problem (4.1)

$$y_{k+1} = y_n + h\Phi(t_k, y_n; h) \tag{4.61}$$

where y_k is the approximation of $y(t_k)$ (see Section 4.1).

In addition to the Runge-Kutta approximations, y_n and y_{k+1}, a second pair satisfying the same Runge-Kutta relation should be considered, say u^*k and u^*_{k+1}. The step sizes are always the same.

So,

$$u^*_{k+1} = u^*_k + h\Phi(t_k, u^*_k; h) \tag{4.62}$$

Definition 4.3.6. *The method (4.61) is called contractive if there exists real numbers $h_0 > 0$ and $0 \leq \varepsilon \leq 1$ such that*

$$\|u^*_{k+1} - y_{k+1}\| \leq \varepsilon \|u^*_k - y_n\|, \quad \text{for all } h \in (0, h_0]. \tag{4.63}$$

Definition 4.3.7. *The method represented at (4.61) is called unconditionally contractive if a real number, $0 \leq \varepsilon \leq 1$, exists such that*

$$\|u^*_{k+1} - y_{k+1}\| \leq \varepsilon \|u^*_k - y_n\|, \quad \text{for all } h > 0. \tag{4.64}$$

Absolute stability is, in fact, a contractivity property. Absolute stability of Runge-Kutta method is completely determined by the properties of the stability function R.

Any Runge-Kutta method applied to

$$y'(t) = Ay(t), \tag{4.65}$$

where A is a constant matix, yields to

$$y_{k+1} = R(hA)y_n. \tag{4.66}$$

$R(hA)$ is the result of applying the stability function R to the matrix hA. This matrix valued function exists if the complex valued function $R(\mu)$ is defined on the set of eigenvalues of hA. If $h\lambda_i$ are the eigenvalues of hA, then, $R(h\lambda_i)$ are the eigenvalues of $R(hA)$.

Suppose that A is a normal matrix. Then, $R(hA)$ is normal and

$$\|R(hA)\|_2 = \max_i |R(h\lambda_i)|. \tag{4.67}$$

Theorem 4.3.4. *[8] Let λ_i be the eigenvalues of A. Also, if A is a normal matrix,*

$$|R(h\lambda_i)| \leq 1, \quad i = 1, \dots, s \tag{4.68}$$

is necessary and sufficient for contractivity in $\|\cdot\|_2$.

Still, let $\lambda \in \mathbb{C}$ be given and introduce the normal matrix

$$\begin{bmatrix} Re(\lambda) & -Im(\lambda) \\ Im(\lambda) & Re(\lambda) \end{bmatrix}$$

with eigenvalues λ and $\overline{\lambda}$. Since A is normal, $R(hA)$ is also normal with eigenvalues $R(h\lambda)$ and $R(h\overline{\lambda})$. This implies that

$$\|R(hA)\|_2 = |R(\mu)|. \tag{4.69}$$

Lemma 4.3.2. *A Runge-Kutta method is A-stable if and only if the method is unconditionally contractive (4.3.7) in $\|\cdot\|_2$ on the class of dissipative, constant coefficient problems (4.60) for $y(0) = y_0$ and with normal matrix A.*

A more restrictive condition, somewhat, is given by

$$|R(h\lambda_i)| < 1 \tag{4.70}$$

for arbitrary matrices A and with λ_i, $i = 1,\ldots,m$, being an eigenvalue of A. Equation (4.66) can be rewritten as

$$y_{k+1} = R^{n+1}(hA)y_0. \tag{4.71}$$

Also, if (4.70) is satisfied, the spectral radius (σ) satisfies $\sigma[R(hA)] < 1$.

In this case, matrix $R^n(hA)$ converges to the null matrix as $n \to \infty$ [30]. However, this convergence behaviour gives no indication as to the magnitude of y_{n+1} for finite n. If $Re(\lambda_i) < 0$, a similar situation arises for the exact solution $y(t)$ of (4.65) because this solution is always a linear combination of polynomials in t multiplied by exponential functions $\exp(\lambda_i t)$ [9].

Lemma 4.3.3. *Condition* (4.70) *merely implies that the numerical solution* y_{n+1} *defined by* (4.66) *cannot have unbounded growth in n (stability). If A is normal, condition* (4.70) *implies strict contractivity in* $\|\cdot\|_2$.

The stability function $R(\mu)$ determines the behaviour of a one-step method when this method is applied to the model problem (4.60).

Application of the Runge-Kutta method to the initial value problem (4.1) reveals that

$$R(z) = 1 + b^T z(I - Az)^{-1}e, \;\; z = h\lambda. \tag{4.72}$$

Notice that R is a polynomial in z if the method is explicit. In all other cases, is a rational function.

Let y_n, y_{n+1} and Y_i be numerical solutions defined by

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i f(t_n + c_i h, Y_i) \tag{4.73a}$$

where

$$Y_i = y_n + h\sum_{i=1}^{s} a_{ij} f(t_n + c_j h, Y_j), \;\; 1 \le i \le s. \tag{4.73b}$$

Theorem 4.3.5. *The stability function of the Runge-Kutta method (4.73) is given by*

$$R(z) = \frac{det(I - zA + zeb^T)}{det(I - zA)}. \tag{4.74}$$

Proof. Application of the Runge-Kutta method (4.73) to the model equation (4.60) yields the linear system

$$\begin{bmatrix} 1-h\lambda a_{11} & -h\lambda a_{12} & \ldots & -h\lambda a_{1s} & 0 \\ -h\lambda a_{21} & 1-h\lambda a_{22} & \ldots & -h\lambda a_{2s} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -h\lambda a_{s1} & -h\lambda a_{s2} & \ldots & 1-h\lambda a_{ss} & 0 \\ -h\lambda b_1 & -h\lambda b_2 & \ldots & -h\lambda b_s & 1 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_s \\ Y_{k+1} \end{bmatrix} = \begin{bmatrix} y_k \\ y_k \\ \vdots \\ y_k \\ y_k \end{bmatrix}$$

with Y and y defined as in (4.73).

Cramer's rule expresses y_{n+1} as the quotient of two determinants. The denominator is given by the determinant of the matrix presented, which is clearly equal to $det(I - zA)$. The numerator is the determinant of the matrix

$$\begin{bmatrix} 1-h\lambda a_{11} & -h\lambda a_{12} & \dots & -h\lambda a_{1s} & y_n \\ -h\lambda a_{21} & 1-h\lambda a_{22} & \dots & -h\lambda a_{2s} & y_n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -h\lambda a_{s1} & -h\lambda a_{s2} & \dots & 1-h\lambda a_{ss} & y_n \\ -h\lambda b_1 & -h\lambda b_2 & \dots & -h\lambda b_s & y_n \end{bmatrix}.$$

Subtraction of the last row from the first s rows leaves this determinant invariant, and so,

$$y_{k+1} = \frac{det(I - zA + zeb^T)}{det(I - zA)} y_n. \tag{4.75}$$

□

4.3.2 B-Stability

In this Section, a more general contractivity theory is studied and some nonlinear stability concepts are also presented. This behaviour analysis will concentrate most on dissipative equations applications, i.e., equations satisfying for some real inner product $< \cdot, \cdot >$,

$$< f(t,\varepsilon_1) - f(t,\varepsilon_2), \varepsilon_1 - \varepsilon_2 > \leq 0 \tag{4.76}$$

for all $\varepsilon_1, \varepsilon_2 \in \mathbb{R}^m$ with $t \geq 0$.

From now on, is considered that the differential equation is dissipative on the whole space $\mathbb{R}^m$.

Suppose that $\tilde{y}_n$, $\tilde{y}_{n+1}$ and $\tilde{Y}_i$ are solutions obtained by perturbations or different starring values satisfying

$$\tilde{y}_{n+1} = \tilde{y}_n + h \sum_{i=1}^{s} b_i f(t_n + c_i h, \tilde{Y}_i) \tag{4.77a}$$

$$\tilde{Y}_i = \tilde{y}_n + h \sum_{i=1}^{s} a_{ij} f(t_n + c_j h, \tilde{Y}_j), \quad 1 \leq i \leq s. \tag{4.77b}$$

This following definition has been introduced by Butcher [3] and considers the concept of B-stability for Runge-Kutta methods applied to stiff nonlinear initial value problems with autonomous differential equations (f independent of t).

Definition 4.3.8. *A Runge-Kutta method (4.73) is called B-stable if, for all autonomous problems satisfying (4.76), all y_n, $\tilde{y}_n$ and all $h > 0$,*

$$\|y_{n+1} - \tilde{y}_{n+1}\| \leq \|y_n - \tilde{y}_n\|. \tag{4.78}$$

Later [2][7], a slightly different concept was defined, applied to non-autonomous equations satisfying (4.76).

Definition 4.3.9. *A Runge-Kutta method (4.73) is called BN-stable if, for all f satisfying (4.76), all y_n, $\tilde{y}_n$ and all $h > 0$, inequality (4.78) holds.*

Both stability definitions mean unconditional numerical contractivity for problems dissipative in some inner product norm.

Corollary 4.3.2. *BN-Stability implies B-stability.*

Suppose that the matrix A of Runge-Kutta parameters is non-singular and let A^{-T} the be the transpose of A^{-1}. The symmetric matrix is then given by

$$Q = BA^{-1} + A^{-T}B - A^{-T}bb^TA^{-1}. \tag{4.79}$$

Theorem 4.3.6. *Let B and Q be nonnegative definite. Then the Runge-Kutta method is B-stable.*

Proof. Let $\varepsilon_1, \dots, \varepsilon_s$ be arbitrary vectors from $\mathbb{R}^m$. Then the nonnegativity definite matrix of Q implies that

$$\sum_{i,j=1}^{s} \left(b_i a_{ij}^{(-1)} + b_j a_{ji}^{(-1)} - \sum_{k=1}^{s} b_k a_{ki}^{(-1)} b_k a_{kj}^{(-1)} \right) < \varepsilon_i, \varepsilon_j > = \sum_{i,j=1}^{s} q_{ij} < \varepsilon_i, \varepsilon_j > \ \geq 0, \tag{4.80}$$

where $a_{ij}^{(-1)}$ and q_{ij} denote the (i,j) element of A^{-1} and Q, respectively. Let $v_0 = y_n - \tilde{y}_n$, $v = y_{n+1} - \tilde{y}_{n+1}$, $v_i = Y_i - \tilde{Y}_i$ and $w_i = hf(t_n + c_ih, Y_i) - hf(t_n + c_ih, \tilde{Y}_i)$. From (4.77b) and (4.73), using the non-singularity of A,

$$w_i = \sum_{j=1}^{s} a_{ij}^{(-1)} \{(Y_j - y_n) - (\tilde{Y}_j - \tilde{y}_n)\} = \sum_{j=1}^{s} a_{ij}^{(-1)} (v_j - v_0). \tag{4.81}$$

In order to compute the norm of v, consider the difference between (4.77a) and (4.73).

$$\begin{aligned}
&\|v\|^2 = \|v_0 + \textstyle\sum_{i=1}^{s} b_i w_i\|^2 \\
&= \|v_0\|^2 + 2 < v_0, \textstyle\sum_{i=1}^{s} b_i w_i > + < \textstyle\sum_{k=1}^{s} b_k w_k, \textstyle\sum_{k=1}^{s} b_k w_k > \\
&= \|v_0\|^2 + 2\textstyle\sum_{i=1}^{s} b_i < v_i, w_i > + 2\textstyle\sum_{i=1}^{s} b_i < v_0 - v_i, \textstyle\sum_{j=1}^{s} a_{ij}^{(-1)}(v_j - v_0) > \\
&+ \left\langle \textstyle\sum_{i,k=1}^{s} b_k a_{ki}^{(-1)}(v_i - v_0), \textstyle\sum_{i,k=1}^{s} b_k a_{kj}^{(-1)}(v_j - v_0) \right\rangle \qquad (4.82) \\
&= \|v\|^2 + 2\textstyle\sum_{i=1}^{s} b_i < v_i, w_i > \\
&- \textstyle\sum_{i,j=1}^{s} \left(b_i a_{ij}^{(-1)} + b_j a_{ji}^{(-1)} - \textstyle\sum_{k=1}^{s} b_k a_{ki}^{(-1)} b_k a_{kj}^{(-1)} \right) < v_i - v_0, v_j - v_0 > \\
&= \|v\|^2 + 2\textstyle\sum_{i=1}^{s} b_i < v_i, w_i > - \textstyle\sum_{i,j=1}^{s} q_{ij} < v_i - v_0, v_j - v_0 > .
\end{aligned}$$

Since $b_i \geq 0$, the second term on the right-hand side is nonpositive and $< v_i, w_i > \leq 0$ according to (4.76) and the definition of v_i and w_i. The last term is nonnegative by (4.80). It is possible to conclude that $\|v\|^2 \leq \|v_0\|^2$. □

For one dimensional problems, Theorem 4.3.6 can be proved as following. Quantities Y_i and $\tilde{Y}_i$, for $i = 1, \ldots, s$, are scalars. The following vectors can be defined

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_s \end{bmatrix} \quad \text{and} \quad \tilde{Y} = \begin{bmatrix} \tilde{Y}_1 \\ \tilde{Y}_2 \\ \vdots \\ \tilde{Y}_s \end{bmatrix}, \quad \text{with } Y,\ \tilde{Y} \in \mathbb{R}^s. \tag{4.83}$$

Also,

$$F(t_n, Y) = \begin{bmatrix} f(t_n + c_1 h,\ Y_1) \\ \vdots \\ f(t_n + c_s h,\ Y_s) \end{bmatrix} \quad \in \mathbb{R} \tag{4.84}$$

and

$$V = Y - \tilde{Y}, \ \ W = hF(t_n, Y) - h(t_n, \tilde{Y}). \tag{4.85}$$

The subtraction of (4.77b) from (4.73b) leads to

$$W = A^{-1}(V - v_0 e), \tag{4.86}$$

and equation (4.82) can be formulated as

$$\begin{aligned} |v|^2 &= |v_0 + b^T W|^2 = |v_0|^2 + 2v_0 b^T W + |b^T W|^2 \\ &= |v_0|^2 + 2V^T BW - 2V^T BW + 2v_0 e^T BW + W^T bb^T W \\ &= |v_0|^2 + 2V^T BW - 2\left(V - v_0 e\right)^T BA^{-1}\left(V - v_0 e\right) + \left(V - v_0 e\right)^T A^{-T} bb^T A^{-1}\left(V - v_0 e\right) \\ &= |v_0|^2 + 2V^T BW - \left(V - v_0 e\right)^T Q\left(V - v_0 e\right). \end{aligned} \tag{4.87}$$

Theorem 4.3.7. *If an implicit Runge-Kutta method is such that*

1. *A is non-singular;*
2. *B is nonnegative;*
3. *The abscissas $c_1, \ldots, c_s$ are distinct;*
4. *$0 \leq e^T BA^{-1} e \leq 2$;*
5. *$B(s)$: $\sum_{i=1}^{s} b_i c_i^{k-1} = \frac{1}{k}$, for $1 \leq k \leq s$;*

6. $C(s-1)$: $\sum_{i=1}^{s} a_{ij} c_j^{k-1} = \frac{1}{k} c_i^k$ *, for* $1 \leq k \leq s-1$;

7. $D(s-1)$: $\sum_{i=1}^{s} b_i c_i^{l-1} a_{ij} = \frac{1}{l} b_j (1 - c_j^l)$ *, for* $1 \leq j \leq$ *and* $1 \leq l \leq s-1$;

then, the method is B-stable.

Proof. Notice that the condition 3 of the Theorem 4.3.7 implies the regularity of the Vandermonde matrix V defined by

$$V = \begin{bmatrix} 1 & c_1 & \dots & c_1^{s-1} \\ 1 & c_2 & \dots & c_2^{s-1} \\ \vdots & \vdots & & \vdots \\ 1 & c_s & \dots & c_s^{s-1} \end{bmatrix}. \tag{4.88}$$

Also, hypothesis 5, 6, and 7 are equivalent, respectively, to

- $b^T V I_s = e^T S I_s$;
- $A V I_{s-1} = C V S I_{s-1}$;
- $I_{s-1} V^T B A = I_{s-1} S (e e^T - V^T C) B$;

where $B = diag(b_1, \dots, b_s)$, $C = diag(c_1, \dots, c_s)$ and $S = diag\left(1, \frac{1}{2}, \dots, \frac{1}{s}\right)$. Multiplying Q with matrix $CVSI_{s-1}$ and applying the previous conditions,

$$\begin{aligned} QCVSI_{s-1} &= A^{-T} BCVSI_{s-1} + BA^{-1} CVSI_{s-1} - A^{-T} bb^T A^{-1} CVSI_{s-1} \\ &= A^{-T} BCVSI_{s-1} + BVI_{s-1} - A^{-T} bb^T VI_{s-1} \\ &= A^{-T} BCVSI_{s-1} + A^{-T} B \left(ee^T - CV\right) SI_{s-1} - A^{-T} Bee^T SI_{s-1} = \mathbf{0}. \end{aligned} \tag{4.89}$$

Notice that $CVSI_{s-1}$ has rank $s-1$. Thus, the dimension of the null space of Q is at least equal to $s-1$. If this dimension is equal to s, then Q is the zero matrix, hence nonnegative. Supposing the dimension equal to $s-1$, the nullspace of Q is spanned by the first $s-1$ columns of CVS, according to (4.89). These are just the last $s-1$ columns of VS and because VS is non-singular, the first column of this matrix, e, does not lie in the null space of Q.

$$\begin{aligned} e^T Q e &= e^T BA^{-1} e + e^T A^{-T} Be - e^t A^{-T} bb^T A^{-1} e \\ &= 2\left(e^T BA^{-1} e\right) - \left(e^T A^{-T} Be\right)\left(e^T BA^{-1} e\right) \\ &\left(e^T BA^{-1} e\right)\left(2 - e^T BA^{-1} e\right) \geq 0 \end{aligned} \tag{4.90}$$

according to the hypothesis 4 of the Theorem 4.3.7.Thus, Q in nonnegative. Moreover, B in nonnegative and Theorem 4.3.6 can be applied and so, the method is B-stable.

□

Chapter 5

Numerical Simulations

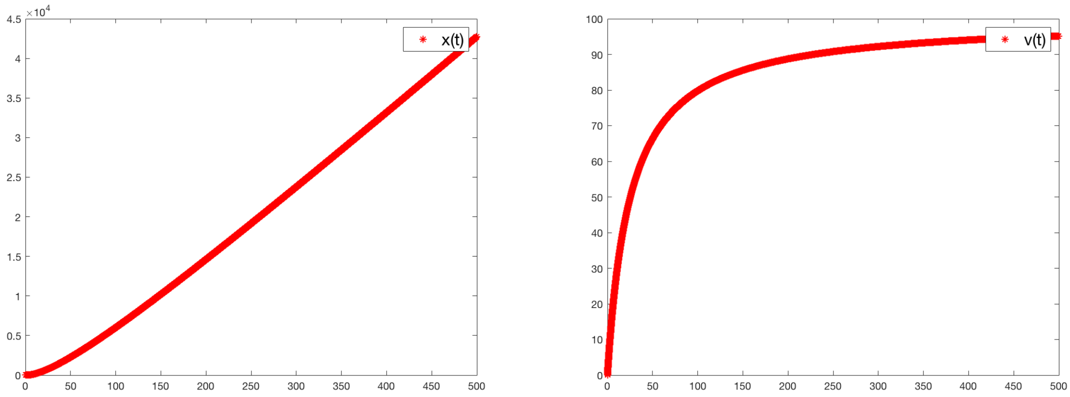
This chapter presents numerical solutions of the systems discussed previously which models the speed and trajectory of a moving body. The numerical method used to solve the systems is the 4th order Runge-Kutta method and the codes are implemented in Matlab.

To implement the model in Matlab, four functions were created. One implements the Runge-Kutta method for $x(t)$ and $v(t)$ (rksolver1). The other two contain information about the equations which build the systems f_1 and f_2 (3.3) (derivsrk_dy and derivsrk_dx, respectively). Last, but not least, a runner function has been created to run all the information digested for the other ones.

To understand the influence of body mass and water/wind velocities in the trajectories of objects adrift at the ocean, results for different values of the parameters involved in the system (body mass, water and wind velocities) are presented.

5.1 Small objects model

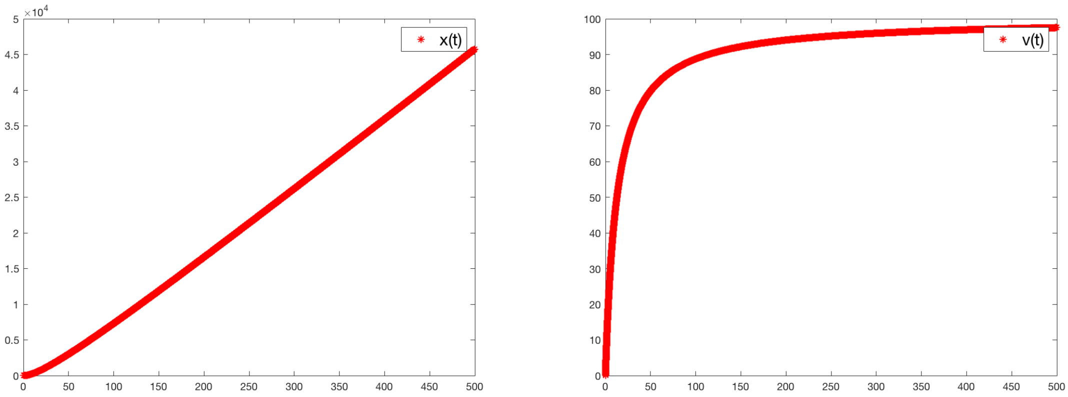
In this first simulation for the small objects model, the only parameter that changes is the mass of the drifting object.



(a) Position along time

(b) Velocity along time

Fig. 5.1 Simulation of small objects model, $m = 100$, $Vw = 100$.



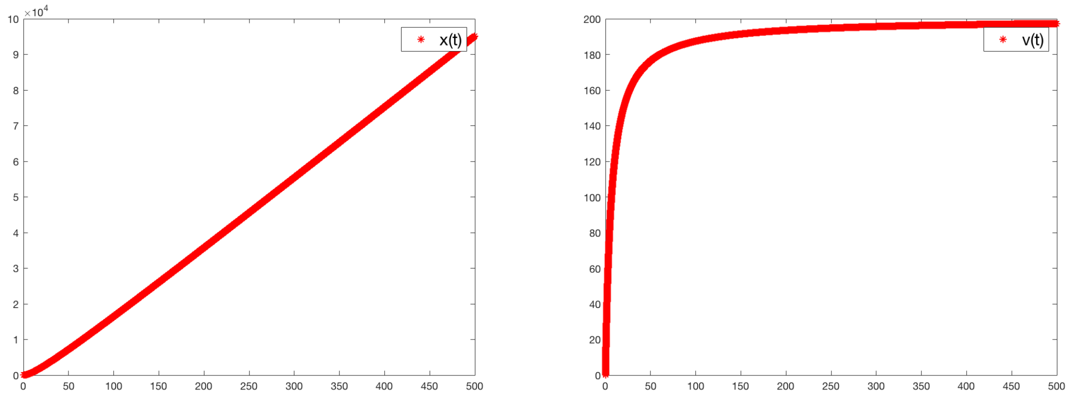
(a) Position along time

(b) Velocity along time

Fig. 5.2 Simulation of small objects model, $m = 50$, $Vw = 100$.

Notice that, by observing Fig. 5.1 and Fig. 5.2, the object with less mass reaches faster a higher speed and travels a greater distance, as expected. The speed of the object adrift tends to the speed of the water since this is the main factor that triggers movement in this model.

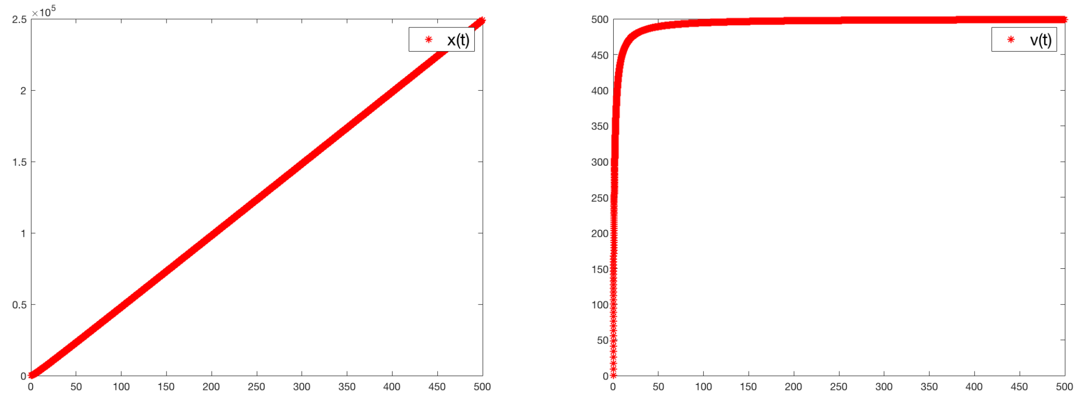
In the second simulation of the small objects model, shown by Fig. 5.3 and Fig. 5.4, the only parameter that changes is the water velocity.



(a) Position along time

(b) Velocity along time

Fig. 5.3 Simulation of small objects model, $m = 100$, $Vw = 200$.



(a) Position along time

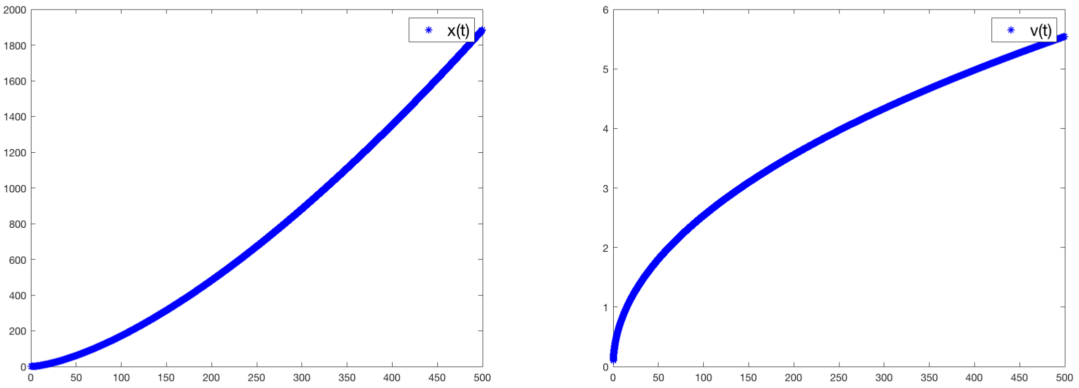
(b) Velocity along time

Fig. 5.4 Simulation of small objects model, $m = 100$, $Vw = 500$.

As expected, higher water speed causes the object to have a higher drift speed. As before, the speed of the adrif object tends to the speed of the water.

5.2 Cargo containers model

In the first simulation for the cargo containers model, the parameter that varies is the wind velocity. In this simulation, Va is considered equal to 0, and can be compared with the simulation displayed in Fig. 5.1.

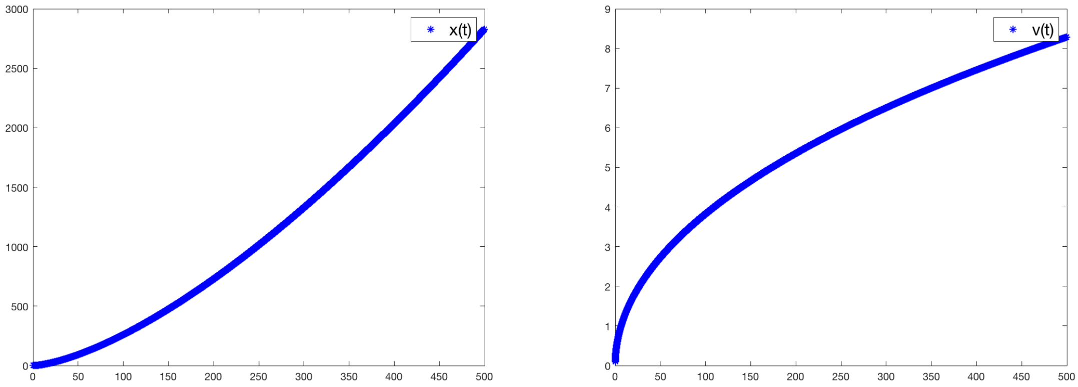


(a) Position along time

(b) Velocity along time

Fig. 5.5 Simulation of cargo containers model, $m = 100$, $Vw = 100$, $Va = 0$.

Even neglecting the wind velocity, the two models produce different results for objects with the same mass and subject to the same speed of water. In the next simulation, an analysis is done about the wind speed on the total speed of the object under the same conditions as in the previous example (except for the wind speed parameter).



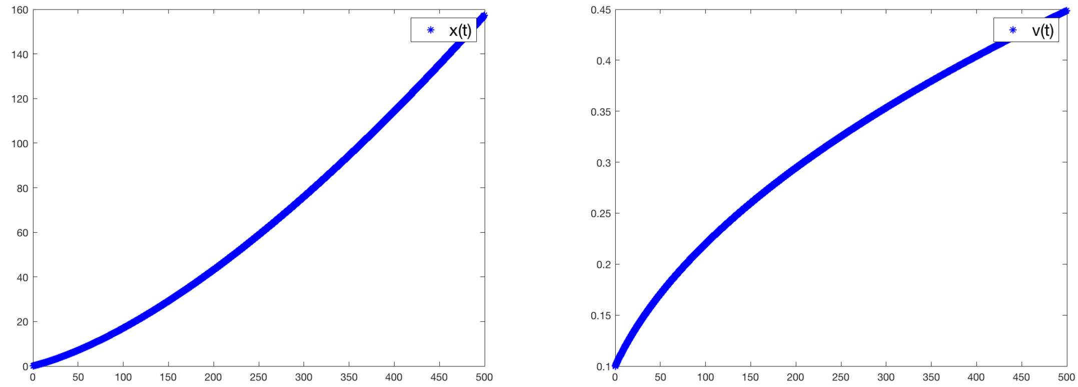
(a) Position along time

(b) Velocity along time

Fig. 5.6 Simulation of cargo containers model, $m = 100$, $Vw = 100$, $Va = 100$.

It is noticed that with wind contribution, the object travels a great distance and reaches a high velocity.

Now, a simulation to study the influence of the mass is presented. The mass is the only parameter that changes and the other parameters are the same as in (Fig. 5.6).



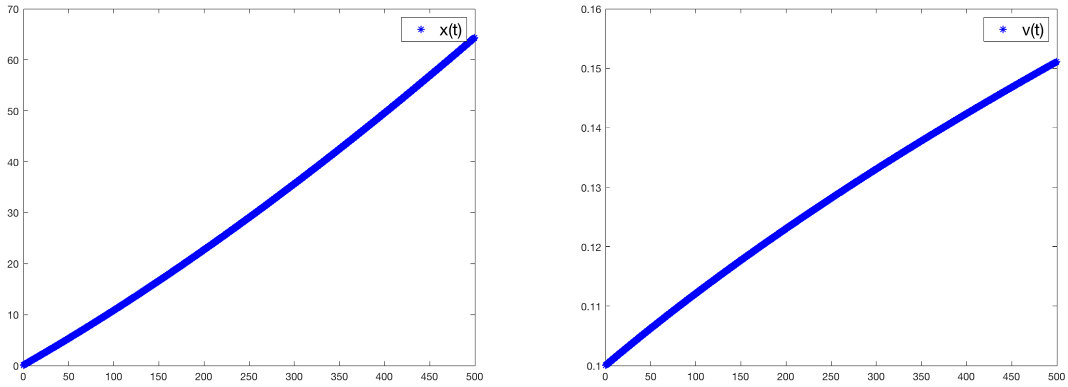
(a) Position along time

(b) Velocity along time

Fig. 5.7 Simulation of cargo containers model, $m = 2000$, $Vw = 100$, $Va = 100$.

Notice that the object with greater mass reaches lower velocity, and travels a smaller distance, as expected.

Neglecting the water velocity, let us study how the wind velocity influences the movement and velocity of an object with high mass as in the previous simulation (Fig. 5.7).



(a) Position along time

(b) Velocity along time

Fig. 5.8 Simulation of cargo containers model, $m = 2000$, $Vw = 0$, $Va = 100$.

As expected, total velocity has decreased when the water velocity was neglected. The object, with the same mass, was only subject to wind velocity and therefore, the total velocity decreased, impacting the distance travelled by the object.

It is possible to conclude that, objects with higher mass are less influenced by the forces to which they are subjected.

Chapter 6

Conclusion

Drifting objects in a marine environment are subject to different circumstances and are influenced in different ways. It is crucial to adapt the model according to the characteristics of the objects and also of the environment in which the object is inserted.

Objects such as cargo containers, compared to smaller objects such as a human body, need more force applied to them in order to produce movement. Still, there is always a two-way application of force on the object. When the object moves, it means that one of the forces is superior and it is in the direction of that same force that the object moves. If they are the same, they are balanced and there is no movement.

The wind produces more force on the object. This force can occur in the direction of the water force or in the opposite direction. In the first case, without loss of generality, the speed of the object is increased and in the second it is decreased.

Numerical simulations carried out with the 4th order Runge-Kutta method corroborate these conclusions, allowing to conduct a maritime drift prediction study using mathematical models that can help on SAR operations. The aim of using differential equations problems was to be able to predict the value of some quantity, satisfying a differential equation, at some future time.

References

[1] Atkinson, K., W., H., and Stewart, D. (2009). *Numerical Solution of Ordinary Differential Equations*. John Wiley & Sons, Inc.

[2] Burrage, K. and Butcher, J. C. (1979). *Stability criteria for implicit Runge-Kutta methods*. SIAM J.

[3] Butcher, J. C. (1975). *A stability property of implicit Runge-Kutta methods*. BIT.

[4] Butcher, J. C. (2003). *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, Inc.

[5] Cabioc'h, F. and Aoustin, Y. (1997). Criteria for decision making regarding response to accidentallty spilled chemicals in packaged form: hydrodynamic asoects. *Spill Science and Technology Bulletin*, 4:7–15.

[6] Christensen, K. H., Breivik, Dagestad, K., Röhrs, J., and Ward, B. (2018). Short-term predictions of ocean drift. *Oceanography*, 31:59–65.

[7] Crouzeix, M. (1979). *Sur la B-stabilité des méthodes de Runge-Kutta*. Numer.

[8] Dahlquist, G. (1975). *Error analysis for a class of methods for stiff nonlinear initial value problems*. Springer-Verlag.

[9] Dekker, K. and Verwer, J. (1984). *Stability of Runge-Kutta methods for stiff nonlinear differential equations*. North-Holland.

[10] Dingemans, M. (1994). Water wave propagation over uneven bottoms.

[11] Ekman, V. W. (1905). *On the Influence of Earth's Rotation on Ocean-Currents*. UNIVERSITY MICROFILMS, INC.

[12] Frank, J. (2008). *Numerical Modelling of Dynamical Systems*. Centrum Wiskunde & Informatica.

[13] Gomes, P. Z. (2019). Salvar em terra com o saber do mar. *Exame*, 428:44–47.

[14] Grotmaack, R. (2003). Small rigid floating bodies under the influence of water waves. *Research Letters in the Information and Mathematical Sciences*, 5:143–157.

[15] Henrici, P. (1962). *Discrete variable methods in ordinary differential equations*. John Wiley and Sons, Inc.

[16] Kong, Q. (2014). *A Short Course in Ordinary Differential Equations*. Springer.

[17] Kosinski, A. A. (2001). Cramer's rule is due to cramer. *Mathematics Magazine*, 74:310–312.

[18] Lambert, J. D. (1973). *Computational methods in ordinary differential equations*. John Wiley and Sons, Inc.

[19] Liu, Y. and Khim, J. (September, 20, 2020). Taylor's theorem (with lagrange remainder). https://brilliant.org/wiki/taylors-theorem-with-lagrange-remainder/.

[20] Marchenko, A. V. (1999). The floating behaviour of a small body acted upon by a surface wave. *J. AppL Maths Mechs*, 63:471–478.

[21] Meylan, M. H., Yiew, L. J., Bennets, L. G., French, B., and Thomas, G. A. (2015a). Surge motion of an ice floe in waves: comparison of theoretical and experimental models. *Annals of Glaciology*, 56:155–159.

[22] Meylan, M. H., Yiew, L. J., Bennetts, L. G., French, B. J., and Thomas, G. A. (2015b). Surge motion of an ice floe in waves: comparison of a theoretical and an experimental mode. *Annals of Glaciology*, 56:69.

[23] Pierre, D., Gwénaële, J., Cabioc'h, F., Landau, Y., and Erwann, L. (2002). Drift model of cargo containers. *Spill Science and Technology Bulletin*, 7:279–288.

[24] Rauch, J. (2014). Advanced ordinary differential equations and dynamical systems. http://www.math.lsa.umich.edu/ rauch/558/nonlinearsinks.pdf.

[25] Santana, A. P. and Queiró, J. F. (2010). *Introdução à Álgebra Linear*. gradiva.

[26] Sierra, M. (2020). Catching your drift. *Scientific American*, 323:14–15.

[27] Smith, S. (1993). Hindcasting iceberg drift using current profiles and winds. *Cold Regions Sciences and Technology*, 22:33–45.

[28] Sousa, E. (2016). *Apontamentos de Matemática Numérica II*. Departamento de Matemática, Faculdade de Ciências e Tecnologia, Universidade de Coimbra.

[29] Strang, G. and Edwin, H. (2016). *Calculus - Volume 1*. Openstax.

[30] Varga, R. S. (1962). *Matrix iterative analysis*. Prentice Hall, Inc.

[31] Villate, J. E. (2019). *Dinâmica e Sistemas Dinâmicos*. Universidade do Porto.

Appendix A

Matlab Codes

A.1 Absolute stability regions

```
%% Function to plot Euler absolute stability region

function theplot = Absolute_Stability_Euler()
x = -3:0.01:1;
y = -3:0.01:3;
[X,Y] = meshgrid(x,y);
Z = X+1i*Y;
y0 = 0.0*x;
x0 = 0.0*y;
W1 = 1+Z;
Q1 = W1.*conj(W1);
figure(1)
hold on;
theplot = contourf(X,Y,-Q1,[-1 -1]);
plot(x,y0,'--k',x0,y,'--k');
hold off;
axis equal;
set(gca,'fontsize',16);
title('Euler''s Method','fontsize',18);
xlabel('$\mathrm{Re}(z)$','interpreter','latex','fontsize',18);
ylabel('$\mathrm{Im}(z)$','interpreter','latex','fontsize',18);
end
```

```
%% Function to plot 4th order Runge-Kutta absolute stability region

function theplot = Absolute_Stability_4RungeKutta ()
x = -3:0.01:3;
```

```
y = -3:0.01:3;
[X,Y] = meshgrid(x,y);
Mu = X+i*Y;
y0 = 0.0*x;
x0 = 0.0*y;
W1 = 1 + Mu + .5*Mu.^2 + (1/6)*Mu.^3 + (1/24)*Mu.^4;
Q1 = W1.*conj(W1);
figure(1)
hold on;
theplot = contourf(X,Y,-Q1,[-1 -1]);
plot(x,y0,'--k',x0,y,'--k');
hold off;
axis equal;
set(gca,'fontsize',16);
title('4th Order Runge-Kutta Method','fontsize',18);
xlabel('$\mathrm{Re}(z)$','interpreter','latex','fontsize',18);
ylabel('$\mathrm{Im}(z)$','interpreter','latex','fontsize',18);
end
```

A.2 Drift models

```
%% Function to define v'(t)

function dy = derivsrk_dy(x,time,y)
%% Uncomment in conformity

rhow = 0.9; %water density
m = 100; %object's mass
Cm = 0.2; %added mass coefficient
Vw = 200; %water velocity
Cd = (0.8+0.065*abs(Vw))*0.001; %drag coefficient
Sw = 10; %cross-sectional area affected by the water
g= 0.98; %gravitical acelaration

rhoa = 1.2; %air density
Sa = 10; %cross-sectional area affected by the wind
Va = 5000; % wind velocity
a = 1.25; %wave amplitude
L = 10; %the lenght of the container normal to incident waves of amplitude a
f = 1; %coriolis force
D = 1; %wave lenght
```

```
K = 1; %vertical vector
k = 0.01; %wave number
w = 0.01; %wave frequency

%Small objects model
eta_x = k*sin(k*x-w*time);
dy = -g*eta_x/sqrt(1+eta_x^2) + (0.5* rhow * Cd * Sw * abs(Vw-y)*(Vw-y)) / m*(1+Cm); % drift equation

%Cargo containers model
dy = (0.5*rhoa*Cd*Sa*abs(Va-y)*(Va-y) + 0.5*rhow*Cd*Sw*abs(Vw-y)*(Vw-y) + 0.25*rhow*g*a^2*L)/(m^2*f*K*D*y); %cargo containers
```

```
%% Function to define x'(t)

function dx = derivsrk_dx(x,time,y)
%% Uncomment in conformity

%Small objects model
k = 0.01; %wave number
w = 0.01; %wave frequency
eta_x = k*sin(k*x-w*time);
dx = y/sqrt(1+eta_x^2);

%Cargo containers model
dx = y;
```

```
%% Function to run 4th order Runge-Kutta

function [t,data,data2] = rksolver1(y,x,dt,t_final, derivsrk_dy, derivsrk_dx)
time = 0;
Nsteps = round(t_final/dt); %number of steps to take
t = zeros(Nsteps,1);
data = zeros(Nsteps,1);
data2 = zeros(Nsteps,1);
t(1) = time; %store initial condition for t
data(1) = y;
data2(1) = x;
    for i=1:Nsteps

    k1 = dt * feval(derivsrk_dy,x,time, y);
    l1 = dt * feval(derivsrk_dx,x,time, y);
```

```

        k2 = dt * feval(derivsrk_dy,x, time + dt / 2, y + k1 / 2);
        l2 = dt * feval(derivsrk_dx,x, time + dt / 2, y + l1 / 2);

        k3 = dt * feval(derivsrk_dy,x, time + dt / 2, y + k2 / 2);
        l3 = dt * feval(derivsrk_dx,x, time + dt / 2, y + l2 / 2);

        k4 = dt * feval(derivsrk_dy,x, time + dt, y + k3);
        l4 = dt * feval(derivsrk_dx,x, time + dt, y + l3);

        y = y + k1/6 + k2/3 + k3/3 + k4/6;
        x = x + l1/6 + l2/3 + l3/3 + l4/6;

        time = time + dt;
        t(i+1) = time;
        data(i+1) = y;
        data2(i+1) = x;
        end
end
```

```
%% Simulations runner

[t,data,data2] = rksolver1(0.1, 0.1, 0.02, 500, @derivsrk_dy,
    @derivsrk_dx);

figure(1)
plot(t,data, 'r*')
legend('v(t)','FontSize',16);


figure(2)
plot(t,data2,'r*')
legend('x(t)','FontSize',16);
```