Blue Brain Project

EPFL Semester Project – Spring 2016

Explicit stabilized methods for numerical integration of Neuron's equations

Supervisors:
Casalegno Francesco
Rosilho de Souza Giacomo

Author:
Nicolò Ripamonti
Sciper n. 260853

Responsible:
Prof. Abdulle Assyr

May 18, 2016

Contents

Chapter 1

Introduction

Chapter 2

Biological model

Brief introduction ...

2.1 The Hodgkin–Huxley model

Say who introduced the model.. Talk about HH
Experimental results show that the process of depolarization of the surface membrane of a nerve fibre could be modeled as an equivalent electrical circuit (see Figure 2.1). It is important to underline that Hodgkin and Huxley derived their model not by describing the molecular events related to changes in permeability but they were driven by some questions regarding the general behaviour of cellular membrane[**main_paper_HH**]. Their starting assumption, verified later by data, was that the membrane permeability is influenced by the membrane potential and not by the membrane current: a direct consequence is that the main contribution to membrane permeability is the effect of electric fields on distribution of charged particles. This result shows the importance of including in the model the mechanism of the ions crossing the membrane (in particular sodium and potassium)

Unfortunately, at their time, they were only able to reject previously accepted assumptions and models but thanks to a exaustive study of nerve cells of squid giant axon, they got a more in-depth view of some properties of the membrane, like the steep relation between its potential and the ionic conductance.

Even small changes in the potential are followed by a e-fold increase in the ionic conductances of sodium and potassium: the effect of this phenomenon is of great importance since it affects the difference in ionic concentration between the inside of the cell and the extracellular liquid which in its turn generates a Nerst's potential that can be represented by a battery in the simplified circuit.

Another observed property is the behaviour of the membrane with respect to an imbalance between input and output current, i.e. more positive

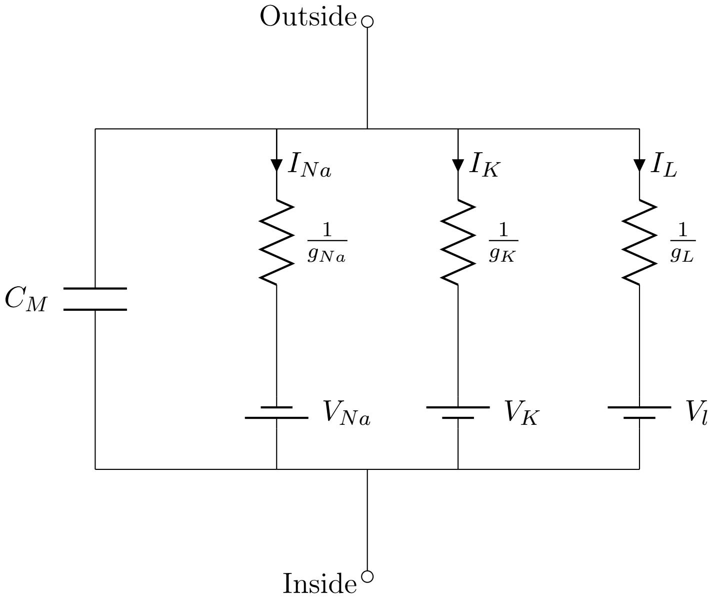

Figure 2.1: Equivalent electrical circuit

charges enter the cell than leave it or viceversa, which made them think that capacitive effects come into play to change membrane potential, which polarise or depolarise according to the change in the total charge. All these observations, combined with their previous hypothesis, were used to build a solid theoretical framework from which a set of equations could be developed.

2.2 Mathematical description

The total membrane current can be written as the sum of a capacity current (modeled as a perfect electrical capacitor without dielectric loss) and of an ionic current.

$$I = C_M \frac{\partial V}{\partial t} + I_i, \tag{2.1}$$

In (2.1) we used the following notation:

I is the total membrane current density [mA]

I_i is the ionic current density [mA]

V is the displacement of membrane potential from its resting value [mV]

C_M is the membrane capacity per unit area [...]

t is the time [...]

The choice of model the currents in parallel is due to the similarity between ionic current when $\frac{dV}{dt} = 0$ and the capacity current when $I = 0$. This model is enriched by identifying the different contributions for the ionic current, as follows:

$$I_i = I_{\mathrm{Na}} + I_{\mathrm{K}} + I_{\mathrm{L}},$$

where I_{Na} and I_{K} are the components related to different ions channels (Na=Sodium, K=Potassium) and I_{K} is the leak current. Using Ohm's law, expressed in term of conductance (i.e. $I = gV$), yields the equation,

$$I_i = g_{\mathrm{Na}}(V - V_{\mathrm{Na}}) + g_{\mathrm{K}}(V - V_{\mathrm{K}}) + g_{\mathrm{L}}(V - V_{\mathrm{L}}),$$

where V,V_{Na},V_{K} and V_{L} are the displacements from resting potentials. The sign of the individual ionic contribution can be positive or negative depending on the relation between the membrane voltage and the ionic potential: this can can lead to a problem of sign conventions, which is solved by setting positive a ionic current flowing out of the cell.

As said before, the conductance of sodium and potassium channel show a strong dependence on membrane potential, while g_{L} is usually taken as a constant: this is called leakage conductance and its value is smaller than the voltage-dependent conductances. Although Hodgkin and Huxley did not know the complex mechanism behind the rise and fall of ionic conductances, they were were able to reconstruct this relation by fitting their experimental results (see Figure (2.2)).

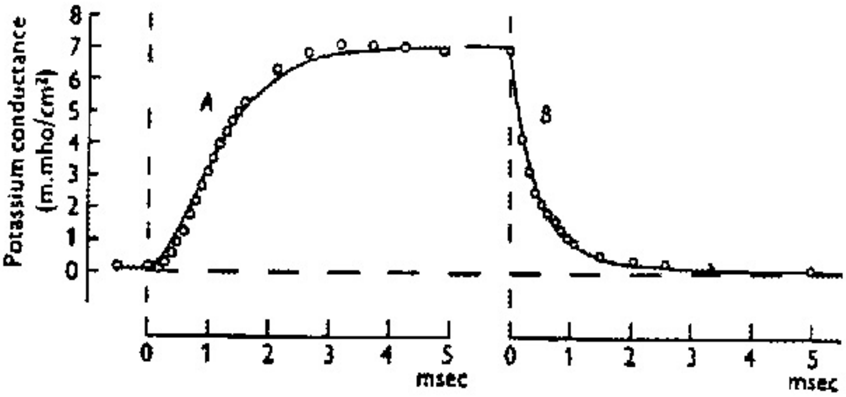

Figure 2.2: Rise and fall of potassium conductance associated with a depolarization and a polarization of 25 mV

Without any first principles avalaible they observed that the potassium conductance is proportional to the fourth power of the solution of a first order equation in the following sense

$$\begin{aligned} g_{\mathrm{K}} &= \bar{g}_{\mathrm{K}} n^4, \\ \frac{dn}{dt} &= \alpha_n(1-n) - \beta_n n, \end{aligned} \tag{2.2}$$

where $\bar{g}_{\mathrm{K}}$ has the dimension of a conductance/cm^2, α_n and β_n are constant in time but depend on membrane potential V and n is a dimensionless variable constrained between 0 and 1. In the resting state V = 0, n has the value

$$n_0 = \frac{\alpha_n|_{V=0}}{\alpha_n|_{V=0} + \beta_n|_{V=0}}.$$

In order to obtain an explicit formulation for α_n and β_n, many measuraments were taken at different temperatures, ionic concentrations and membrane potential configurations and then plotted against the resting potential V to give the following result

$$\alpha_n = 0.01\frac{V+10}{e^{\frac{V+10}{10}} - 1},$$
$$\beta_n = 0.125e^{\frac{V}{80}},$$

where the functions α_n and β_n are measured in ms and V is measured in mV. They noticed that during the depolarisation phase the variation in g_{K} has a sigmmoid shape, while the polarisation phase behaves as a decaying exponential: this situations are captured very well by α_n and β_n respectively. The fourth power that relates n to g_{K} is related to the number of first-order processes that have to occur in order to have the desired curvature of the sigmoid function: the more numerous they are, the more steep the shape will be. This is compatible with their educated guess (verified several years later) that each ionic channel contains several gates, and all of them have to be open in order to have an open channel (and so a sigmoid depolarisation) while it is sufficient that one of them is closed to shut the channel (hence the exponentially decaying polarisation). After this physical explanation, n can be interpreted as the probability for a given gate to be open: if we extend this to the more realistic case of a large number of channels, it becomes the fraction of the total number of gates that are in the permessive state (N.B. The definition of permissive/non-permissive state of the gate is a little bit different from the definition of open/close and it was given later than [**main_paper_HH**], but still today α and β represents the transition between this two states).

Regarding the sodium conductance, the procedure followed is only slightly different from the one adopted for the potassium. As before the idea is that the conductance is related to a certain number of first order process, but while for the potassium these processes are all of the same kind, described by the gate state n, now we have two different variables that come into play.

This led them to the following assumptions:

$$
\begin{aligned}
g_{\mathrm{Na}} &= m^3 h \bar{g}_{\mathrm{Na}}, \\
\frac{dm}{dt} &= \alpha_m(1-m) - \beta_m m, && (2.3) \\
\frac{dh}{dt} &= \alpha_h(1-h) - \beta_h h, && (2.4)
\end{aligned}
$$

where $\bar{g}_{\mathrm{Na}}$ is a constant and the α's and β's depend on V but not on time. Here h plays an opposite role of what n did for the potassium channels: in this case we consider it to be means when the channel is inactive and we have the polarisation phase. On the other hand, m behaves as n, with the only difference that for sodium ion channels the steepness of the relation between gate conductance and membrane potential is given by only 3 of these gates instead of 4 as before. However, even if all these "activating" gates m are open, but the "inactivating" gate h is open, then the channel remains closed. The initial states for the first-order equations describing the transition rates are again

$$
\begin{aligned}
m_0 &= \frac{\alpha_m|_{V_0}}{\alpha_m|_{V_0} + \beta_m|_{V_0}}, \\
h_0 &= \frac{\alpha_h|_{V_0}}{\alpha_h|_{V_0} + \beta_h|_{V_0}}.
\end{aligned}
$$

Now that we have all the coefficients, we can rewrite (2.1) as

$$
I = C_M \frac{\partial V}{\partial t} + \bar{g}_{\mathrm{K}} n^4 (V - V_{\mathrm{K}}) + \bar{g}_{\mathrm{Na}} m^3 h (V - V_{\mathrm{Na}}) + \bar{g}_{\mathrm{L}} (V - V_{\mathrm{L}}). \qquad (2.5)
$$

For the cable approximation of the axon, which is surrounded by a large amount of conducting fluid, the membrane current, in the case of a propagated action potential, is related to the potential V by the following electrostatical relation

$$
I = \frac{1}{2aR} \frac{\partial}{\partial x} \left(a^2 \frac{\partial V}{\partial x} \right),
$$

where R is the cytoplasmatic resistivity and a the axon radius, which may vary along the cable. Hence, after rearranging equation (2.5) to take into account the unit of measuraments used for the experimental setting, we obtain

$$
\begin{aligned}
10^{-3} C_M \frac{\partial V}{\partial t} = & \frac{10^4}{2aR} \frac{\partial}{\partial x} \left(a^2 \frac{\partial V}{\partial x} \right) \\
& - \bar{g}_{\mathrm{K}} n^4 (V - V_{\mathrm{K}}) - \bar{g}_{\mathrm{Na}} m^3 h (V - V_{\mathrm{Na}}) - \bar{g}_{\mathrm{L}} (V - V_{\mathrm{L}}) \qquad (2.6) \\
& - \frac{10^2 g_{syn} y}{2\pi a} \delta_{syn}(x)(V - V_{syn}),
\end{aligned}
$$

in which the most simple modelization of synaptic interaction is considered ($g_{syn}(y)$), depending only on the synaptic receptor state y evolving as

$$y(t) = \frac{t - t_{\text{off}}}{\tau} e^{\left(-\frac{t - t_{\text{off}} - \tau}{\tau}\right)}$$

This is a simple, yet effective, description of an incoming spike in the potential, able to capture the nature of the synaptic behaviour, but also to allow fast and efficient computation, required in the case of large neural networks.

2.3 Domain and boundary conditions

The model described in the previous section is applied on a domain which can be schematized on a domain as the one represented in Figure (2.3). The unbranched sections of the neuron are represented as 1D segments, an

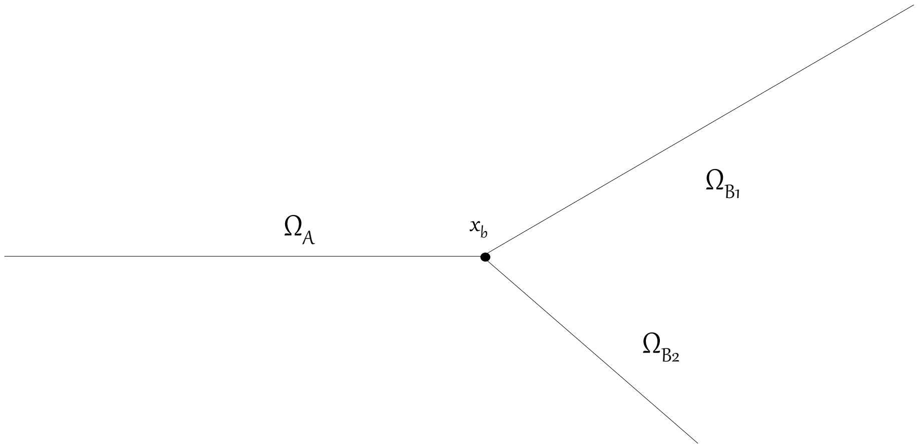

Figure 2.3: Dendritic tree with one branching node.

approximation due to the difference between length and thickness of each branch: this entails that all the variables considered are function of the axial abscissa along the cable but not of the radial distance from the center. This model is acceptable if the real radius, even if variable, show only relative deviations with respect to its mean value: some problem may occur if, for instance, we consider a tapered dendrite.

To set the boundary conditions and then complete the statement of the problem, electrostatic consideration are needed: the current between two points x and $x + \Delta x$ is proportional to the voltage difference and the inverse of membrane resistance, thus

$$I(x) = \frac{\pi a^2}{R} \frac{V(x - \Delta x) - V(x)}{\Delta x},$$

and by taking the limit $\Delta x \to 0$ we have

$$I(x) = \frac{\pi a^2}{R}\frac{\partial V}{\partial x}.$$

In the branching node, Kirchoff's law for the conservation of electric charge that

$$\left(a^2\frac{\partial V}{\partial x}\right)\Bigg|_{x=x_b}^{A} - \left(a^2\frac{\partial V}{\partial x}\right)\Bigg|_{x=x_b}^{B_1} - \left(a^2\frac{\partial V}{\partial x}\right)\Bigg|_{x=x_b}^{B_2} = 0,$$

which is chosen as boundary condition in the branching node x_b.
On the terminal points it is assumed that no current flows outside or inside, giving homogeneous Neumann conditions

$$\frac{\partial V}{\partial x} = 0.$$

Chapter 3

Spatial discretization

3.1 Finite differences

The partial differential equations (PDEs) (2.6),(2.2),(2.3) and (2.4) are replaced by a system of ordinary differential equations (ODEs), using a technique known as *method of lines*. The idea is to discretize all the dimensions of the problem but one (time in our case) and then solving the resulting equation (ODEs). First we need to define a mesh: on the internal regions of the dendrite a uniform mesh with spacing h is used, while for the elements near the boundary for which is reduced to $\frac{h}{2}$ (TODO: spiega perchè è più fine sul bordo, motivi di convergenza e chiedi a Francesco il motivo bio).

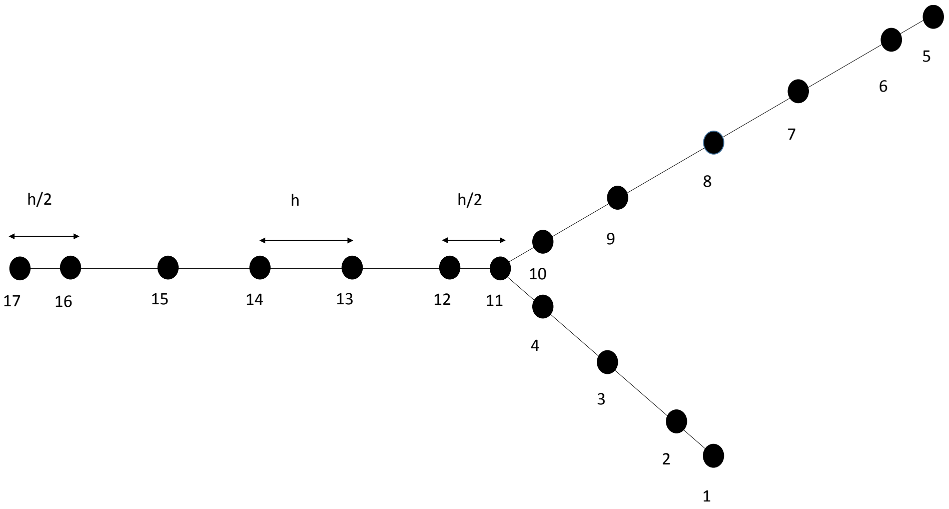

Figure 3.1: Example of mesh used

A possible starting point for the discretization is to look at finite differences centered at each point x_j of the mesh: for the points far from boundaries the following standard results for uniform grid can be used

$$\frac{\partial f}{\partial x} = \frac{f(x + \frac{\Delta x}{2}) - f(x - \frac{\Delta x}{2})}{\Delta x} + \mathcal{O}((\Delta x)^2),$$

which is a second order centered approximation of the first order derivative of a function. By applying this form twice in the diffusive term of (2.6) we get to

$$\frac{\partial}{\partial x}\left(a^2 \frac{\partial V}{\partial x}\right) \approx a^2 \frac{V(x + \Delta x) - 2V(x) + V(x - \Delta x)}{(\Delta x)^2},$$

in the case of a constant.
Near the terminal and ending nodes the situation is more subtle because of the nonuniformity of the grid and so we need to go more in depth with Taylor series: we report, as example, the case of the last node of the branch Ω_A.

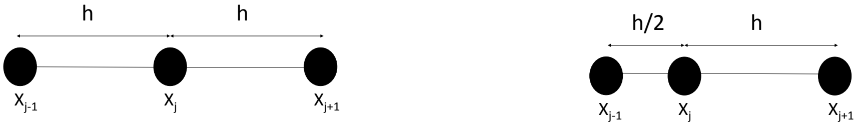

Figure 3.2: Local region in the middle of the mesh (left) and near the terminal point of the branch Ω_A (right)

Consider

$$f(x_{j+\frac{1}{2}}) = f(x_j) + \frac{\partial f}{\partial x}(x)\bigg|_{x=x_j} \frac{\Delta x}{2} + \frac{\partial^2 f}{\partial x^2}(x)\bigg|_{x=x_j} \left(\frac{\Delta x}{2}\right)^2 + \mathcal{O}((\Delta x)^3),$$

$$f(x_{j-\frac{1}{2}}) = f(x_j) - \frac{\partial f}{\partial x}(x)\bigg|_{x=x_j} \frac{\Delta x}{4} + \frac{\partial^2 f}{\partial x^2}(x)\bigg|_{x=x_j} \left(\frac{\Delta x}{4}\right)^2 + \mathcal{O}((\Delta x)^3),$$

and by combining them you get

$$\frac{\partial f}{\partial x}(x)\bigg|_{x=x_j} = \frac{f(x_{j+\frac{1}{2}}) + 3f(x_j) - 4f(x_{j-\frac{1}{2}})}{\frac{3}{2}\Delta x} + \mathcal{O}((\Delta x)^2)$$

3.2 Node enumeration

Chapter 4

Time integration

4.1 Stability of numerical integrators

4.1.1 Examples

4.1.2 Linear stability analysis

The more general case we can face is the non-linear non-automous ODEs

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y} = \mathbf{f}(t, \mathbf{y}), & \mathbf{y} \in \mathbb{R}^n, t \in [t_1, t_2] \\ \mathbf{y}(t_0) = \mathbf{y}_0. \end{cases} \tag{4.1}$$

To study its stability behaviour we need to approximate this form to a case which is more manageable, by means of Taylor's series applied to the force term in (4.1). Suppose we have a smooth solution $\boldsymbol{\phi}(t)$, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y} = \mathbf{f}(t, \boldsymbol{\phi}(t)) + \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(t, \boldsymbol{\phi}(t))(\mathbf{y}(t) - \boldsymbol{\phi}(t)) + \ldots, \tag{4.2}$$

and then, by discarding all the terms of order bigger than 1 and setting $\mathbf{w}(t) = \mathbf{y}(t) - \boldsymbol{\phi}(t)$ in (4.2), we remain with

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{w} = \mathbf{A}(t)\mathbf{w}(t), \tag{4.3}$$

where $\mathbf{A}(t) = \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(t, \boldsymbol{\phi}(t))$. Now we have to kill the time dependency in $\mathbf{A}(t)$, by freezing $t = t_0$, in order to obtain a constant matrix $\mathbf{A}_0$. By doing this, the following analysis will be valid only near the starting time and to extend it further we have to change the istant in which the time is frozen. For the sake of clarity now we suppose the matrix A_0 to be fully diagonalizable, i.e. there exists an orthogonal matrix $\mathbf{P}$ s.t.

$$\mathbf{P}^{-1}\mathbf{A}_0\mathbf{P} = \mathbf{D}, \qquad D = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}. \tag{4.4}$$

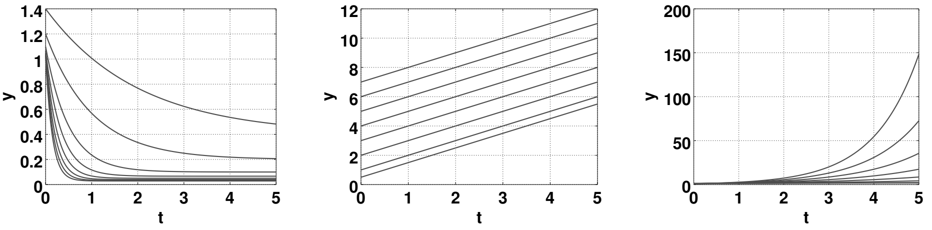

Figure 4.1: Solution of Dahlquist's equation for Re(λ)<0 (stable), Re(λ)$=0$ (asymptotically stable) and Re(λ)>0 (unstable)

This is not always the case, but even in more complex scenarios where the diagonalization is not possible, the stability can be investigated requiring different conditions to A. Using (4.4) in (4.3) we are left with a set of n decoupled equations

$$\frac{\mathrm{d}}{\mathrm{d}t}v_i = \lambda_i v_i, \qquad i = 1 \ldots n, \tag{4.5}$$

where λ_i-s are the eigenvalues of A_0. Each of them represents an example of Dalqhuist test equation

$$\frac{\mathrm{d}}{\mathrm{d}t}y = \lambda y, \qquad \lambda \in \mathbb{C}. \tag{4.6}$$

The solution for this famous problem is an exponential which slope depends on the sign of λ (Figure 4.1). Hence, the meaning of (4.5) is that even if only one of the eigenvalues of the linearized system has a real part greater than zero, then the solution is unstable [**Dahlquist**]. A possible definition of stability was given by Lyapunov in [**Lyapunov**]

Definition 1. A solution to (4.1) that exists for all $t \geq t_0$ is said to be stable (asyntotically stable) in the sense of Lyapunov if $\forall \epsilon > 0\ \exists \delta > 0$ s.t. if $\|\Delta \mathbf{y}\| < \delta$ then

$$\left(\lim_{t=\infty}\right) \quad \|\mathbf{y}(t, t_0, \mathbf{y}_0) - \mathbf{y}(t, t_0, \mathbf{y}_0 + \Delta \mathbf{y})\| < \epsilon.$$

4.1.3 Stability of Runge–Kutta methods

The Runge–Kutta (RK) methods are the most common way to solve an ODE: they are a family of explicit and implicit iterative numerical schemes. Another important group of ODEs solver is the Linear multistep methods [**Hairer_1**], which we will not deal with in this report. RK methods are based on the idea of computing the numerical solution after a time step Δt through a weighted sum of intermediate steps, which approximate the slope of the solution at specific points $\in [t, t + \Delta t]$. Any Runge Kutta method can be presented in the following general form

Definition 2. Let $s \in \mathbb{N}^+$ and let b_i, a_{ij} and c_i, $1 \leq i, j \leq s$ be real numbers. A s-stage Runge-Kutta method for the integration of

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t} y = f(t, y), \\ y(t_0) = y_0, \end{cases} \tag{4.7}$$

is given by

$$\mathrm{K}_i = f\left(t_0 + \mathrm{c}_i \Delta t, y_0 + \Delta t \sum_{j=1}^{s} \mathrm{a}_{ij} \mathrm{K}_j\right),$$
$$y_1 = y_0 + \Delta t \sum_{i=1}^{s} \mathrm{b}_i \mathrm{K}_i.$$

A RK method applied to (4.7) gives approximations at different times $\{y_n(\Delta t, \lambda)\}_{n \geq 0}$. Also for the numerical solution is possible to define a stability domain based on bounded values.

Definition 3. The stability domain of a RK method is

$$S_n = \{z = \Delta t \lambda \in \mathbb{C} : \{y_n(\Delta t, \lambda)\}_{n \geq 0} \quad \text{remains bounded}\}$$

Usually this definition is reinterpreted in terms of stability function, which is introduced in the following statement

Definition 4. Following the same notation of Definition 2, using a RK method to solve Dahlquist's test problem, we obtain after one step $y_1 = R(\Delta t \lambda) y_0$, where

$$R(z) = 1 + \mathbf{b}^T z (I - z\mathbb{A})^{-1} \mathbb{I}, \qquad z = \Delta t \lambda$$
$$\mathbf{b} = \begin{pmatrix} \mathrm{b}_1 \\ \vdots \\ \vdots \\ \mathrm{b}_s \end{pmatrix}, \quad \mathbb{A} = \begin{pmatrix} \mathrm{a}_{1,1} & \mathrm{a}_{1,2} & \dots & \\ \mathrm{a}_{2,1} & \ddots & & \vdots \\ & & \ddots & \mathrm{a}_{n-1,n} \\ & \dots & \mathrm{a}_{n,n-1} & \mathrm{a}_{n,n} \end{pmatrix} \quad \mathbb{I} = \begin{pmatrix} 1 \\ \vdots \\ \vdots \\ 1 \end{pmatrix}$$

is called stability function.

This function does not identify a scheme, since two different numerical methods can have the same stability function. Since we want our method to be stable, the solution of the test problem need to be analyzed. Using Definition 4 we have

$$y_n = R(\Delta t \lambda) y_{n-1} = \dots = (R(\Delta t \lambda))^n y_0 < \infty$$

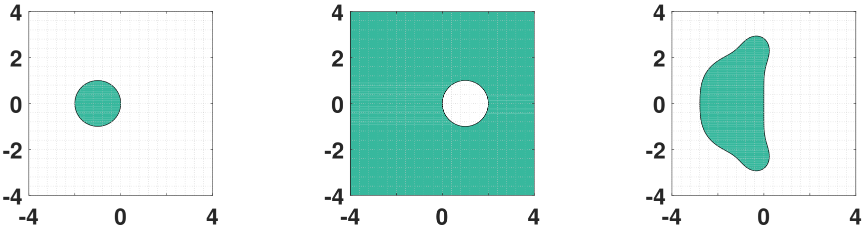

Figure 4.2: Stability regions of explicit Euler, implicit Euler and classical Runge-Kutta

for every y_0 if and only if $|R(\Delta t\lambda)| < 1$, which is an equivalent characterization of the stability domain.

As we have seen before, the test equation has a stable solution only if $\lambda \in \mathbb{C}^-$. A good property is that the numerical methods conserve the stability in the same case.

Definition 5. (Dahlquist 1963) A method, whose stability domain satisfies $S_n \in \mathbb{C}^-$ is called *A-stable*.

The stability function of a RK method can be expressed as the ratio between two polynomials of degree $\leq s$ for the implicit case and as a polynomial of the degree $\leq s$ for explicit method. Hence, by using the maximum principle theorem, it is possible to give sufficient and necessary conditions for the A-stability of a scheme.

Definition 6. A RK method is A-stable if and only if its stability function satisfies:

- $|\mathrm{R}(iy)| \leq 1 \quad \forall y \in \mathbb{R}$ (also called *I-stability* condition)
- $\mathrm{R}(z)$ is analitic in $\mathbb{C}^-$

The *A-stability* property ensures that our numerical solution does not explode, but there still could be some bounded oscillations. A stronger requirement is sufficient to guarante a better qualitative representation of the exact solution.

Definition 7. A RK method is *L-stable* if it is *A-stable* and $\lim_{z\to\infty} \mathrm{R}(z) = 0$

The limit for $z \to \infty$ can be replaced by $z \to -\infty$, since we are dealing with rational function, so we can focus on $\mathbb{C}^-$.

As can be seen in Figure 4.3, an *L-stable* implicit method could behave better than an only *A-stable* implicit method like the implicit midpoint, even if the former has a lower order of accuracy than the latter. This is due to the additional property $\mathrm{R}(\Delta t\lambda) \approx 0$ when we have big time steps ($\Delta t \gg \lambda^{-1}$). Since we are working with rational functions

$$\lim_{z\to-\infty} R(z) = \lim_{z=iy, y\to\infty} R(z), \tag{4.8}$$

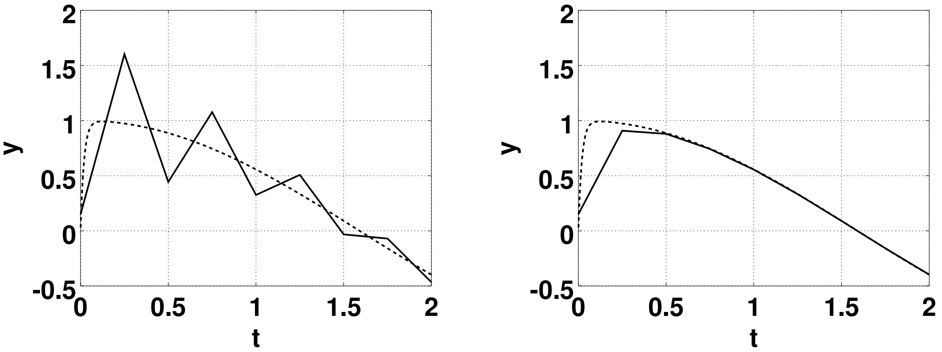

Figure 4.3: Solution of Curtis-Hirschfelder equation using implicit mid-point(left) and Euler implicit(right) methods. The former method is only *A-stable* while the latter is also *L-stable*. The dotted line represents in both cases the exact solution.

If the stability domain of the numerical method contains only $\mathbb{C}^-$ then $|R(iy)| = 1 \ \forall y \in \mathbb{R}$. But from (4.8) we have that, in case of z with a large negative part, $R(z)$ is almost equal to 1 (but smaller). Considering the role of $R(z)$ in Definition 4, the obscillations are only slghtly damped.

4.1.4 Stiff problems

The concern about stability of a numerical solution is due to the resolution of *stiff* problems. There is not a universally accepted definition of *stiffness* of an ODE problem, but a possible tentative characterization is the following

Definition 8. A stable problem $\frac{\mathrm{d}}{\mathrm{d}t}y = f(t, y)$ is called *stiff* when $\max_i |\lambda_i|$ is large, where λ_i is an eigenvalue of $\frac{\partial f}{\partial y}$, with coefficients frozen at some (t_0, y_0). $L := \max_i |\Re(\lambda_i)|$ is sometimes called *stiffness* index and gives an hint about the magnitude of the *stiffness*, while usually the letter ρ is usually used to refere to $\max_i |\lambda_i|$.

For a numerical method to be stable $z = \Delta t \rho$ need to be included in its stability domain. Since ρ could be even of the scale of 10^6 for common problems, a particularly small time step must be chosen in order to not have a solution that literally explodes after few steps. The time step required is influenced by different parameters, firstly by the nature of the problem, but also by the stability domain of the scheme employed. If the stablity domain is restricted to a small portion of the complex plane near the origin, a very fine time step is used, even if we are not interested in having a so detailed picture of the solution in the that time interval. This issue should be taken into account every time we have to chose an ODE solver: sometimes the benefits of using an easy-to-implement and fast method are vanished by the insanely high number of steps required to "calm down" the solution.

4.2 Implicit methods: Solving a linear system

4.2.1 Implicit methods properties

For implicit method, to find the stages defined in Definition 2 a system of non-linear equations has to be solved in order to find the different K_i. For fully implicit methods the dimensions of the system is equal to the number of stages, while for some other methods part of the stages could be determined explicitely. Even worse, if we are the dealing with an n-dimensional problem, the dimension grow to $n \times$ (number o implicit stages). This makes implicit methods more costly to implement and to be solved and more subject to round-off errors. In [**Hairer_1**] some tricks to reduce this kind of errors and make the non-linear system easier to be solved iteratively are introduced: the authors also present good choices for the initial state and stopping criterion of the iterative procedure. One of the advices we want to report is the drawback of fixed point procedure to solve a non-linear system. Given a system of equation equation $\mathbf{x} = \mathbf{G}(\mathbf{x})$, similar to the one that we have to solve for the stages, and provided a reasonably good starting guess for the solution $\mathbf{x}^0$, this procedure requires to compute $\mathbf{x}^{k+1} = \mathbf{G}(\mathbf{x}^k)$ until the difference between two iterations satisfies the requested tolerance. To converge, we need to work with a contraction, i.e.

$$\|\mathbf{G}(\mathbf{x}) - \mathbf{G}(\mathbf{y})\| \leq \nu\|\mathbf{x} - \mathbf{y}\|, \qquad \nu < 1.$$

If we consider that

$$\mathbf{G}(\mathbf{x}) = \mathbf{G}(\mathbf{y}) + \mathbf{G}'(\mathbf{y})(\mathbf{x} - \mathbf{y}) + \dots,$$

and for the problem arising from implicit methods we have, if $\mathbf{x}^0$ is close to the solution then

$$\nu \sim \|\mathbf{G}'(\mathbf{y})\| \sim C\Delta t\rho.$$

But for stiff problems we know that ρ could be very large: this means that what we have gain by using an implicit method will be lost since a very fine time step is required for convergence. A better alternative to solve kind of problems is the Netwon-Raphson method, which relies on the computation

All these complexities are the price for a usually large stability domain. This is the reason for which they are the standard for solving really stiff problems. We must pay attention to not think that any time discrretization will give a stable numerical solution: while it is true that the stability region is bigger than the one of explicit methods, only a sub set of implicit methods are *A-stable* (see Figure **??**).

4.2.2 Strang splitting

As we have seen in the previous section the non-linearity of the problem could cause some issues regarding instabilty and difficulties in the implementation.

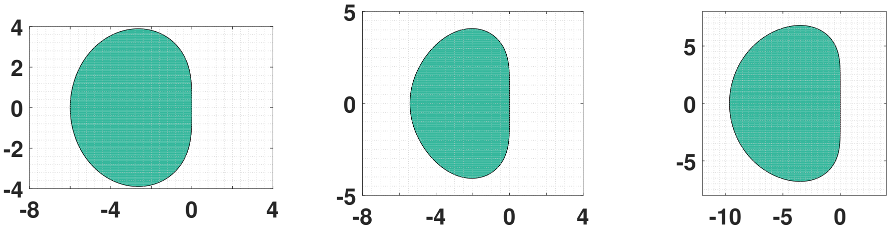

Figure 4.4: Stability regions of explicit Euler, implicit Euler and classical Runge-Kutta

For particular class of operators the non-linearity could be killed by splitting the problem in the sum of linear operators, much easier to deal with. Here we will present a possible technique to address this splitting, with the approach *divide et impera.*

Suppose we have the following differential equation that we want to integrate for a single time step Δt

$$\frac{\mathrm{d}}{\mathrm{d}t} y = f(y,t) = f_1(y,t) + f_2(y,t), \tag{4.9}$$

and that

$$\frac{\mathrm{d}}{\mathrm{d}t} y = f_1(y,t),$$
$$\frac{\mathrm{d}}{\mathrm{d}t} y = f_2(y,t)w,$$

happen to be easier problems than the original (no non-linearities, know exact solutions, ...). The idea for the construction of the new method is to solve first one of the two problems with a certain initial data

$$\frac{\mathrm{d}}{\mathrm{d}t} y = f_1(y,t), \qquad y(0) = y_n.$$

This take us to $y^* = y_1(\Delta t; y_s)$, where y_1 represents the solution of the differential equation on vector field f_1. As next step, we start from y^* and solve the second initial value problem

$$\frac{\mathrm{d}}{\mathrm{d}t} y = f_2(y,t), \qquad y(0) = y^*.$$

on the same interval Δt to obtain $y_{n+1} = y_2(\Delta t; y^*)$ It is possible to reinterpret this procedure in terms of *flows*[**Hairer_1**]

Definition 9. The *flow* of a system $\frac{\mathrm{d}}{\mathrm{d}t} = f(y,t)$ is the mapping which, to any point y_0 in the phase space, associates the value $y(t)$ of the solution with initial value $y(0) = y_0$. This map, denoted by ϕ_t, is thus defined by

$$\phi_t(y_0) = y(t), \quad if \quad y(0) = y_0.$$

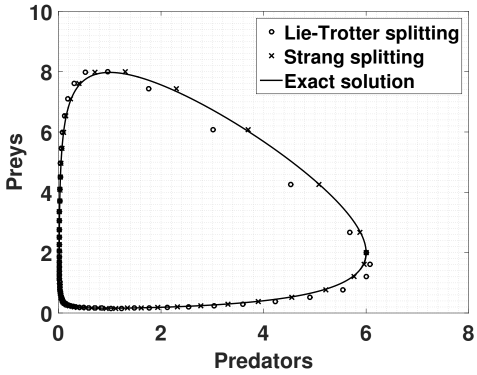

Figure 4.5: Solution of Lotka-Volterra equation [**Lotka**]. Comparison between exact solution and different splitting techniques.

Denoting the flow related to f_1 and f_2 as ϕ_{t,f_1} and ϕ_{t,f_2} respectively, what we have previously computed is $y_{n+1} = \phi_{\Delta t,f_2}(\phi_{\Delta t,f_1}(y_n)) = (\phi_{\Delta t,f_2} \circ \phi_{\Delta t,f_1})(y_n) = \phi^{T-L}_{\Delta t,f}(y_n)$. This is an example of a *splitting method*, the *Lie-Trotter splitting*. Since the order of the operators is arbitrary, also $y_{n+1} = (\phi_{\Delta t,f_1} \circ \phi_{\Delta t,f_2})(y_n) = (\phi^{T-L}_{\Delta t,f})^*(y_n)$ would have been a viable choice (and we obtain the adjoint of the previous operator). By Taylor's expansion we can prove that

$$(\phi_{\Delta t,f_1} \circ \phi_{\Delta t,f_2})(y_0) = \phi_{\Delta t,f}(y_0) + \mathcal{O}(h^2)$$
$$(\phi_{\Delta t,f_2} \circ \phi_{\Delta t,f_1})(y_0) = \phi_{\Delta t,f}(y_0) + \mathcal{O}(h^2)$$

and so both the methods gives only a 1^{st} order approximation of the solution to (4.9)

4.3 Explicit methods: Stabilizing with stages

Chapter 5

Setting of the problem

Cable equation:

$$\begin{aligned}10^{-3}C_M\frac{\partial V}{\partial t} =& \frac{10^4}{2aR}\frac{\partial}{\partial x}\left(a^2\frac{\partial V}{\partial x}\right)\\ &-\bar{g}_{\mathrm{K}}n^4(V-V_{\mathrm{K}})-\bar{g}_{\mathrm{Na}}m^3h(V-V_{\mathrm{Na}})-\bar{g}_{\mathrm{L}}(V-V_{\mathrm{L}})\\ &-\frac{10^2 g_{syn}y}{2\pi a}\delta_{syn}(x)(V-V_{syn}),\end{aligned}$$

Boundary conditions:

- Branching node

$$\left(a^2\frac{\partial V}{\partial x}\right)\Bigg|_{x=x_b}^{A} = \left(a^2\frac{\partial V}{\partial x}\right)\Bigg|_{x=x_b}^{B_1} + \left(a^2\frac{\partial V}{\partial x}\right)\Bigg|_{x=x_b}^{B_2}$$

- Terminal nodes

$$\frac{\partial V}{\partial x} = 0$$

Initial condition:

$$V\Bigg|_{t=0} = V_0$$

Gate states:

$$\begin{aligned}\frac{dn}{dt} &= \alpha_n(1-n)-\beta_n n\\ \frac{dm}{dt} &= \alpha_m(1-m)-\beta_m m\\ \frac{dh}{dt} &= \alpha_h(1-h)-\beta_h h\end{aligned}$$

Initial conditions

$$
\begin{aligned}
n_0 &= \frac{\alpha_n|_{V=0}}{\alpha_n|_{V=0} + \beta_n|_{V=0}} \\
m_0 &= \frac{\alpha_m|_{V=0}}{\alpha_m|_{V=0} + \beta_m|_{V=0}} \\
h_0 &= \frac{\alpha_h|_{V=0}}{\alpha_h|_{V=0} + \beta_h|_{V=0}}
\end{aligned}
$$