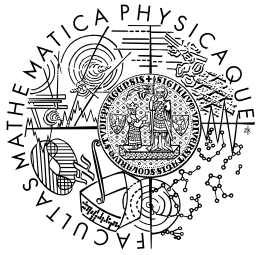


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Tereza Perlácová

Computer modeling of the inner ear

Institute of Theoretical Physics

Advisor of the master thesis:	prof. Mgr. Pavel Jungwirth, DSc.
Study programme:	Physics
Study branch:	Mathematical and computational modelling in Physics

Prague 2017

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In Prague July 10, 2017 signature of the author

Title: Computer modeling of the inner ear

Author: Tereza Perlácová

Department: Institute of Theoretical Physics

Advisor: prof. Mgr. Pavel Jungwirth, DSc.

Abstract: We introduce the implicit numerical approach to the existing mechanical model of the cochlea. We test four implicit methods: implicit Euler, Crank-Nicolson, backward difference formula of the second and of the third order for the linear and the nonlinear version of the model. The nonlinear model involves a function with a saturation property. Implicit methods applied to the nonlinear model lead to a system of nonlinear equations. We propose two ways how to solve this system. The first one includes transferring the nonlinearity to the right hand side of the newly formed linear system. The other one performs the linearization of the nonlinear function. We performed a detailed comparison of these versions from the point of view of the efficiency and the convergence to the reference solution. For the tolerance values of the numerical convergence that we employ, the first version shows to be more effective. The second version fails to converge at the strictly nonlinear regime. We conclude that the Crank-Nicolson method with the first version described above is the most suitable to our problem. Finally we use this method to build an arbitrarily precise connection between the mechanical and the electrical cochlear models, respecting the human physiology.

Keywords: cochlea, implicit numerical methods, mechanical model, partial differential equations

Acknowledgements

Huge thanks go to prof. Pavel Jungwirth who gave me the chance to work on the cochlear modelling project in his research group and who supervised me patiently. Another quantum of huge thanks goes to doc. Václav Kučera from the Department of Numerical Mathematics for his willingness and patient guidance in the questions of the numerical mathematics. Big thanks go to my colleague Ondřej Ticháček for helpful advice in various problems. I thank especially to my fiancé Peter Matvija for his help all along my work and for multiple useful discussions about the concerned topic. Thanks go also to Adriána Nikolovová for a persistent maintaining of a good ambience at my working place. I thank particularly to Dobrá farma that created an enjoyable environment where I could reset myself between individual stages of my work. Lastly, I absolutely thank to my parents who have always believed that I would finish this work one day...

Contents

Introduction

Why do we model hearing?

The (mammalian) perceiving of sound is the result of multiple processes occuring in the ear and in the auditory cortex of the brain. In a normal healthy person, these processes are set such that the system transforms uniquely an external physical stimulus to a specific sound perception. This ensures a clear understanding of the incoming information. However, not all of the persons have the auditory system with this normal functioning. Currently we register various types of partial or total hearing losses. These are caused by the damage of an auditory component situated at various levels of the auditory system. Persons with such damages may hear the sound very weakly or they may have severe problems in the perception of different sounds. To fix these incapacities, engineers develop and produce a variety of substitutes for the damaged auditory components like hearing aids or cochlear implants. These substitutes have as an objective the simulation of the normal functioning of a particular hearing component. When implemented to a living organism, they may therefore partially or totally improve its hearing abilities.

The development of cochlear implants brings us to one of the main reasons why we are attempting to model the hearing processes. To develop a device that could reliably substitute a part of the hearing apparatus, engineers need to use suitable mathematical models. A natural requirement for these models is to reproduce the hearing process as well as possible. Another requirement is to be compatible with the currently used technologies. The development of cochlear implants is therefore a common project of scientists and engineers, namely the scientists who develop the models and the engineers who turn the models into the material form.

Our model - model of cochlea

We utilize a model of an inner ear component, the cochlea, that is the crucial part in the transformation of physical external stimuli to nerve stimulations during the hearing process. This model has been created as a collection of several (originally mostly independent) parts, from which each one models a particular process in the cochlear behaviour. These parts have been developed by different researchers and the current work on this model consists primarily in the improvement of the linkings between them. In the development of our model we collaborate with the researchers from the MED-EL company in Innsbruck[1] that focuses on the development and creation of cochlear implants. The current version of our model is supposed to be published and presented to the MED-EL company in the near future. The model will then be further refined according to the requirements of MED-EL engineers.

[1] http://www.medel.com/

Goal

Our model is a collection of three main parts: the mechanical part, the electrical part and the auditory nerve part. While my colleague Ondřej Ticháček (see Ticháček (2014)) works mainly on the linking between the electrical model and the auditory nerve, my work focuses primarily on the mechanical model and partially on its linking to the electrical model.

The mechanical model that we use originates in the model developed by Fabio Mammano and Renato Nobili in the 90th years (see Mammano and Nobili (1993)). Since then this model has been upgraded several times but the current version still holds the principles integrated by Mammano and Nobili. The electrical part has been developped by Pavel Mistrík, at present a researcher in the MED-EL company, (see Mistrík et al. (2009)) in the first decade of the 21th century. Mistrík et al. (2009) managed to model the ion flow in the cochlea and he proposed also the dependence of the ion flow on the cochlear mechanics. This dependence now serves as a good approximation of the linking between the mechanical and the electrical parts.

The mechanical model is based on two differential equations that govern the motion of cochlear parts. These equations have been implemented into the computer model using various numerical approaches, each of them ensuring a particular goal for the user. Since the usual aim has been to obtain the results quickly, the model contains a very restricted domain of numerical methods. Now, as we try to make the model closer to the real hearing process, we extend our requirements. Our main current aim is to improve the linking between the mechanical and the electrical model so that this respects better the physiological aspects of hearing. This is related to the demands on the wider range of particular input model parameters. To make the model robust with respect to the choice of input parameters we need to consider a much broader domain of numerical approaches than we have done up to now. This brings us to the idea of the implementation of new numerical methods.

The goal of our work is thus to implement a suitable numerical approach to the mechanical model respecting the physiological connection of the mechanical and the electrical parts.

Plan of the work

The work is divided into two parts. The first part is the theoretical background. Here we present the basic mechanisms of the hearing processes in mammals and the principles of our cochlear model. We include here also selected chapters from numerical mathematics. The second part presents our work on the mechanical model. It summarizes eight experiments with the model that led us to the choice of the suitable numerical method.

Part I

Theory

1. Mammalian ear physiology

Ear is the organ that enables hearing. Ear collects sound waves from the surrounding environment and converts them to electrical signals that are subsequently sent to the brain. Transforming sound waves to the information decodable by the brain is generally not a trivial task. The way how ear reaches this aim has been largely unknown until recently. However, thanks to multiple experiments and research studies we can nowadays understand most of the ear's functioning. One of the first basic findings is that ear consists of multiple parts connected in series. Each part passes the signal to the following part after processing it in a specific way. The partitioning of the ear to entities with a separate function makes it a complex organ worthy to describe in a separate chapter.

According to the standard description, a mammalian ear consists of three main parts: the outer, middle and inner ears. In this chapter we provide the essential characteristics of each of these components and we specify their primary function in the process of hearing. Before doing this, we give a brief description of the decibel scale generally used to quantify the intensity of the sound waves. The information about the decibel scale is adopted from the section 1.2 in Pickles (2012) and the sections about the ear physiology follow chapters 2 and 3 in Pickles (2012).

The decibel scale

Sound wave is a longitudinal mechanical wave of pressure in a transmission medium. To express the measure of its intensity, we generally use the decibel scale expressed in Pickles (2012) as

$$\text{Number of dB} = 10 \log_{10} \left(\frac{\text{sound intensity}}{\text{reference intensity}} \right). \tag{1.1}$$

Since the intensity of the pressure wave is related to the pressure by the square root, (1.1) can be rewritten using the sound pressure as follows

$$\text{Number of dB} = 20 \log_{10} \left(\frac{\text{sound pressure}}{\text{reference pressure}} \right). \tag{1.2}$$

Furthermore, we express the *sound pressure* in (1.2) as the root mean square (RMS) value of the pressure wave and we set the *reference pressure* as the lowest sound pressure that is detectable by a human ear, i.e. $2 \times 10^{-5}\,\mathrm{N/m^2}$ RMS. Then the sound intensity levels in (1.2) are expressed in the units called dB SPL (Decibel Sound Pressure Level):

$$\text{Intensity level in dB SPL} = 20 \log_{10} \left(\frac{\text{RMS sound pressure}}{2 \times 10^{-5}\,\mathrm{N/m^2}} \right). \tag{1.3}$$

1.1 Outer ear

Outer ear is composed of two components: pinna and ear canal. Pinna constitutes the external part of the ear and ear canal is the tube connecting pinna to the tympanic

membrane (the eardrum) of the middle ear. The outer ear has various roles in the process of hearing. Its primary role is to collect the incoming sound waves and to funnel them to the middle ear. Its more delicate role consists in aiding sound localization in mammals.

The later function is enabled especially by the corrugated shape of the pinna. This makes the pinna to be a complex acoustic cavity. Incoming pressure waves may therefore exhibit multiple reflections on the rims of the pinna and may then interfere with the primary (non-reflected) wave while travelling through the ear canal. Depending on the wavelength and on the position of the ear with respect to the sound source, the primary and the reflected pressure waves interfere constructively or destructively. While the constructive interference gives rise to a peak in the sound intensity, the destructive interference induces its drop. The dependence of the sound intensity on the frequency of the incoming sound wave is therefore unique for an orientation of the ear with respect to the source. This enables a mammal to localize sound wave sources.

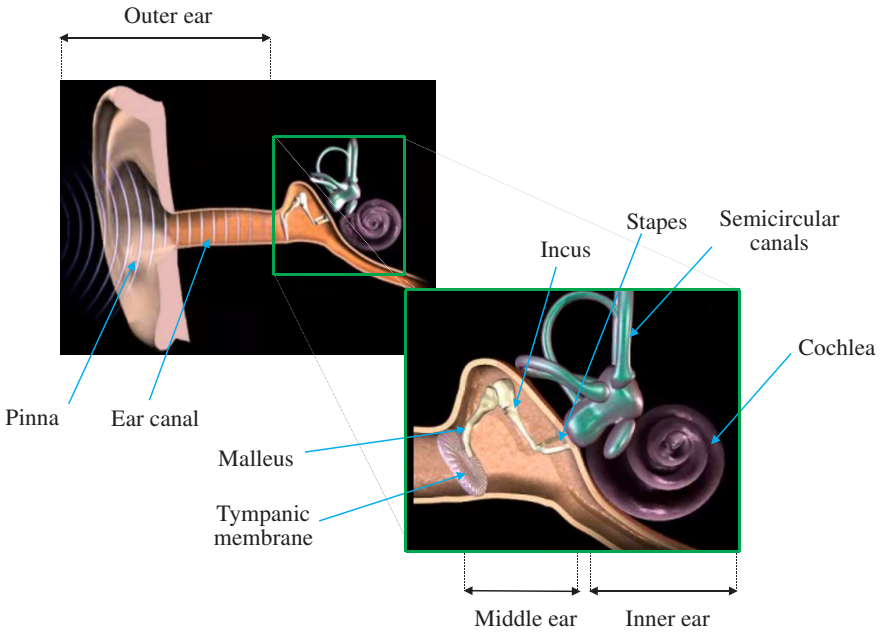


Figure 1.1: **Ear.** Images in the figure are reproduced from the video presentation [1].

1.2 Middle ear

Middle ear transfers sound waves from the outer ear apparatus to the cochlea. A sound wave that travels in the ear canal (in the direction from the exterior) reaches first the tympanic membrane and makes it vibrate. The vibration of the tympanic membrane is then communicated by three small inter-connected bones of the middle ear – malleus, incus and stapes – to the oval window of the cochlea. Why do we need this mediator of transfer between the outer ear and the cochlea?

Firstly, the difference between the impedance of the air (outer ear medium) and the impedance of cochlear fluids (the cochlear medium) is so high that without any mediator, most of the incoming sound hitting the oval window would be reflected back to the exterior. Thanks to specific connections of the middle ear bones and to the high ratio of the tympanic membrane area to the oval window area, the middle ear represents a highly efficient impedance transformer. Therefore, even though the cochlear impedance measured at the oval window is very high, the impedance of the ensemble of the middle ear and of the cochlea measured at the tympanic membrane is much lower. This impedance is therefore much closer to the impedance of the air, hence most of the energy of the sound vibrations coming from the air to the cochlea is efficiently transmitted.

Secondly, middle ear permits to transfer the pressure to only one of the two elastic windows of the cochlea. If there was no such concentrator of the incoming force, the pressure wave travelling through the cochlear fluid would be strongly attenuated and the perceived sound would be very weak.

1.3 Inner ear

Inner ear is formed by the part responsible for the balance, consisting primarily of semicircular canals, and by the part that is responsible for hearing, formed by the cochlea. Here we focus specifically to the hearing part.

Cochlea is the crucial element of the hearing apparatus: it transforms mechanical vibrations coming from the middle ear into a sequence of electrical stimulations that are sent to the brain. Cochlea is embedded in the the lateral part of the skull and it is formed by a coiled tube of the length of about 35 mm. The width of the cochlea in its natural coiled state is about 1 cm. The cochlear tube is divided into three longitudinal scalae, each of them containing a particular cochlear fluid and hearing processing parts. Two of the scalae, scala vestibuli and scala tympani are joined at the cochlear apex[1] by the aperture called helicotrema. The two scalae are filled with perilymph, a cochlear fluid resembling to extracellular fluids in its ionic composition. At the base of the cochlear tube, the scala vestibuli is connected to the oval window of the cochlea and the scala tympani to the round window of the cochlea. These two windows are elastic membranes embedded in the cochlear surface and they are responsible for the effective transmission of pressure vibrations from the stapes to the perilymph as well as for the effective release of the pressure from the perilymph to the surrounding medium. The third scala, scala media, forms a separate cochlear compartment that is not directly joined with the two other scalae. It is separated from the scala vestibuli by the Reissner's membrane and from the scala tympani by the basilar membrane. The cochlear

[1] By *cochlear apex* we mean the edge of the cochlear tube that is distant from the middle ear apparatus. The edge of the cochlear tube that is situated close to the middle ear is called *cochlear base*.

fluid that fills the scala media is the endolymph and its particularity consists in the fact that its ionic composition is similar to most of intracellular fluids.

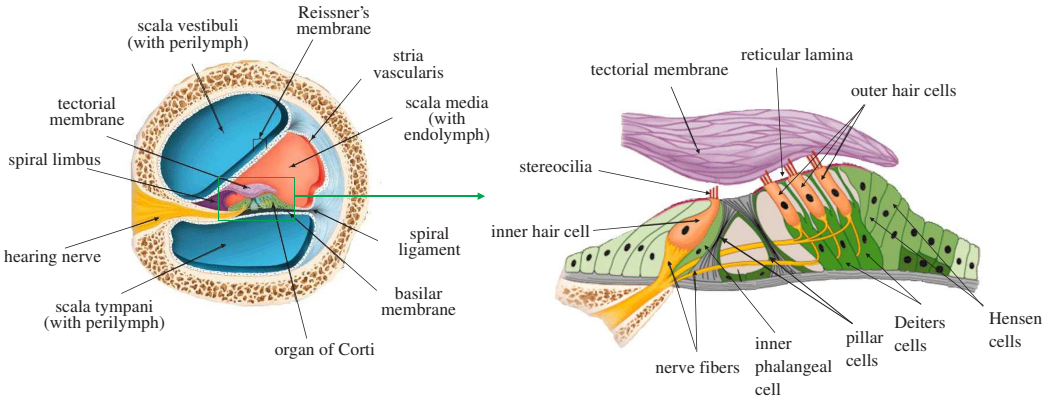


Figure 1.2: **Transversal section of cochlea (left) and zoom to organ of Corti with tectorial memebrane (right).** Figure is reproduced from figures in [2] and [3].

The part of the cochlea that executes the auditory transduction process[2] is the organ of Corti. The organ of Corti lies on the basilar membrane from the side of the scala media. It is separated from the endolymph by a thin layer that adheres to its upper surface, called reticular lamina. The key component structures of the organ of Corti are the transduction receptor cells, the hair cells. Human cochlea contains one longitudinal row of inner hair cells and three to five longitudinal rows of outer hair cells; one row being formed by approximately three thousand of the cells. The key part of a hair cell is the bundle of small protrusions that emerge from the apical part of the cell to the scala media. These are called stereocilia. They play an essential role in capturing of mechanical stimuli transmitted by cochlear fluids. Hair cells are innervated mostly by afferent nerves, sending the information from the cochlea to the central nervous system. Most of the afferent nerves ($90-95\,\%$) are connected to inner hair cells and the rest of them adheres to outer hair cells. Less numerous are efferent nerves that direct the information from the central nervous system to the cochlea.

The organ of Corti contains multiple cells with a secondary, mostly supporting role. The hair cells are surrounded by inner and outer phalangeal cells: inner phalangeal cells surround entirely inner hair cells and outer phalangeal cells (Deiters cells) hold the basal end of outer hair cells. Another type of supporting cells are Hensen cells residing close to outer hair cells. The rigidity of the organ of Corti is given by pillar cells.

[2]Sensory transduction is the conversion of a physical stimulus to nerve stimuli, particularly to a sequence of action potentials. The conversion is executed by receptor cells that capture the stimulus and transform it to the change in the intracellular electrical potential. This change in potential triggers the release of the neurotransmitter by receptor cells and the neurotransmitter activates action potentials in the nerve cells.

Another structure intruding into the scala media and playing a significant role in the hearing transduction is the tectorial membrane. Tectorial membrane is a fibrous oblong structure that is by one edge firmly attached to the spiral limbus and by the other edge attached to the organ of Corti. The principal connection of the tectorial membrane to the organ of Corti is performed by the longest row of outer hair cells' stereocilia. An important role in the transduction process plays also the stria vascularis, situated in the lateral part of the cochlear duct and being in direct contact with the endolymph.

2. Mechanism of hearing in cochlea

We distinguish four stages of the hearing process arising in the cochlea: mechanical processes, electrical processes, the release of neurotransmitter and the communication of the information to the brain via auditory nerve. In this chapter we will present particularly the two most specific parts for hearing: the mechanical and the electrical one.

2.1 Mechanical part

It has been shown that cochlear responses to external stimuli change if we consider the cochlea of a cadaver versus the cochlea of a living organism. This proves the presence of an active element in the cochlea and gives us an incentive to study the two behaviours separately.

2.1.1 Passive cochlea

Pilot experiments and observations

First direct observations of cochlear responses to mechanical stimuli were made by a Hungarian biophysicist Georg von Békésy. Von Békésy performed a series of experiments on cochleae of human and animal cadavers in order to see how the basilar membrane (BM) responds to a mechanical vibration applied to the oval window. His observations are summarized mainly in Von Békésy and Wever (1960), from which the most significant points can be resumed as follows:

A mechanical vibration of a fixed frequency applied to the cochlear oval window gives rise to a travelling wave of the BM displacement (see Figure 2.1). The wave has several particularities. Firstly, it travels always in the direction from the base to the apex of the cochlea. Secondly, the wave velocity decreases as it travels to the apex. Thirdly, the wave envelope has a specific shape: from base to apex it first grows shallowly, then reaches the maximum at a specific point and then decreases rapidly in the region close to the apex. Furthermore, the BM behaviour has a particular dependence on the input vibration frequency. At low input frequencies the maximum of the travelling wave envelope localizes near the cochlear apex and a broad region of the BM becomes activated. Raising the input frequency induces that the maximum of the travelling wave moves towards the cochlear base and that the activated BM region becomes narrower. In each case each point of the activated region of the BM vibrates with the frequency of the input stimulus. The described behaviour proves the *cochlear frequency selectivity.*

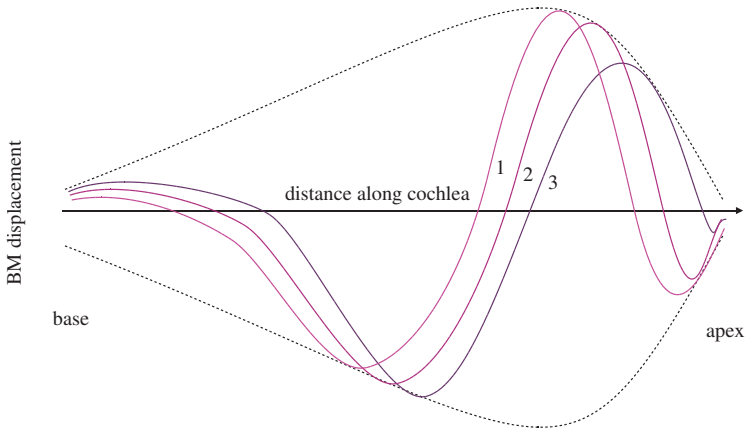


Figure 2.1: **Travelling waves as they were seen by von Békésy.** Solid lines represent the BM displacement at successive instants denoted by numbers. Dashed line is the envelope of the travelling wave. Input frequency is 200 Hz. Figure is a reproduction of Figure 22 from v. Békésy (1953).

Interpretation of the passive cochlear mechanics

Experiments made by von Békésy have provided a solid base to the interpretation of the passive cochlear mechanics. Here we present the principal ideas as they are summarized mainly by Pickles (2012, section 3.2.3.1) and Patuzzi (1996, section 5.1).

A mechanical vibration applied to the cochlear oval window initiates a pressure wave travelling through the perilymph. The pressure wave, as it propagates first by the scala vestibuli and then by the scala tympani, generates a pressure difference between these two scalae. The pressure is finally released by the round window. The mechanical effect of the pressure wave on the cochlear partition[1] is its displacement in the transversal plane. This displacement is depending on the position in the cochlear duct.

To quantify the effect of the pressure difference on the motion of the CP we introduce the notion of the *CP admittance*. It is defined as the velocity of the CP at a particular position divided by the pressure difference imposed at this point. The larger is the admittance of a CP point, the more significant is the response of this point to the pressure difference. The CP admittance is a function of the position in the cochlear duct and of the frequency of the input vibration.

The cochlea as a unified organ presents also its admittance to the incoming pressure wave. This is formed by admittance of individual parts: the stapes with the oval window with stapes, the round window and the cochlear interior. Following the reasoning of Patuzzi (1996, page 244), when a mechanical vibration of a very low frequency is imposed to the oval window, the perilymph just flows forth and back through the helicotrema and displaces the round window. The admittance of the cochlear interior is here given purely by the admittance of the helicotrema, see Figure 2.2C. At a little

[1] By *cochlear partition* (CP) we mean the set of the following cochlear components: the organ of Corti, the basilar membrane and the tectorial membrane.

higher input frequencies the CP is initiated into a movement in phase along its whole length, the base end moving less than the apex end. In this case the admittance of the helicotrema is decreased because of the inertia of the fluid and it's the admittance of the CP that comes into play. The admittance of the cochlear interior is therefore given by the admittance of the helicotrema with the admittance of the CP branched in parallel, see Figure 2.2D. The increase of the input frequency causes the enforcement of the CP motion and the lowering of the perilymph flow through the helicotrema. This can be interpreted as an increase of the CP's admittance and lowering of the helicotrema's admittance. This trend continues up to the point when the helicotrema's admittance becomes very low and the flow of the fluid through it ceases. The cochlear admittance is in this case purely given by the CP admittance, see Figure 2.2E. Increasing the input frequency further causes that the CP is no longer displaced in phase along its length. The behaviour of the CP has the form observed by Von Békésy and Wever (1960): a travelling wave that has the envelope form dependent on the input frequency. In this case the motion of the CP is influenced by multiple parameters: mass, damping and stiffness of the CP and the inertia of the perilymph, see Figure 2.2F. The CP admittance is dependent on the listed quantities in a complicated way and its derivation can be found for example in Elliott and Shera (2012).

An illustrative view on the CP admittance is provided by Pickles (2012), see Figure

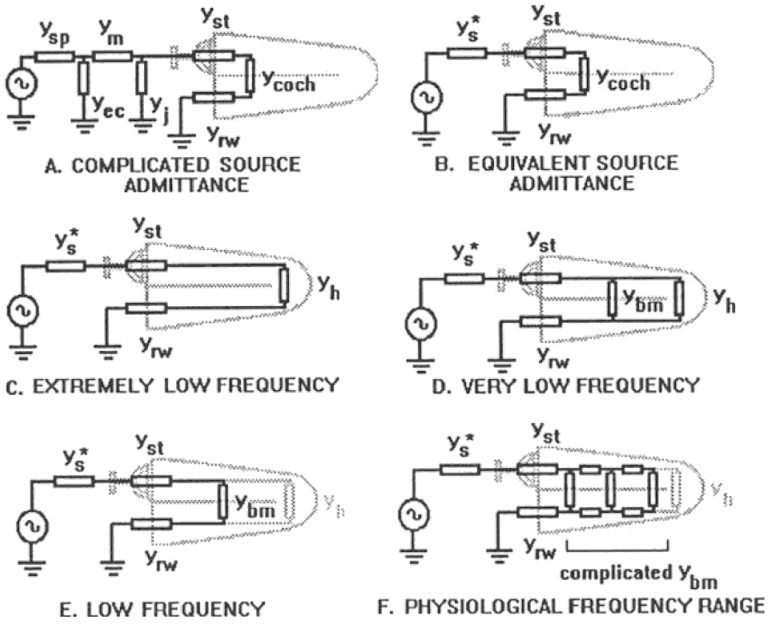


Figure 2.2: **Cochlear admittance at various input signal frequencies. (A)** the admittances of the sound source (y_{sp}), ear canal (y_{ec}), eardrum (y_m) and middle ear (y_j) are replaced by an equivalent source admittance $y_s^\star$ at **(B)**. **(C)-(F)** represent the changes of the admittance of various cochlear components with varying input signal frequency. y_{rw} represents the admittance of the round window, y_h of the helicotrema, y_{bm} of the cochlear partition and y_{st} of the stapes. Figure reprinted from Patuzzi (1996, Figure 4.21, page 244).

2.3. Pickles (2012) represents this quantity by a curve that grows from the base to a specific position depending on the input frequency, reaches the maximum at this position and then descends up to the apex. This form of the dependence results from the theory about the passive cochlear mechanics that he explains in his book (see section 3.2.3.1 in Pickles (2012)). The theory says that the formation of the travelling wave is due to the interaction of the CP stiffness with the inertia of the system CP-perilymph. The latter term embraces the mass of the CP and the inertia of the perilymph. The inertia of the perilymph is due to its longitudinal flow caused by the progressive gain of the phase of the CP. Experiments performed for example by Von Békésy and Wever (1960) and Emadi et al. (2004) show that the stiffness of the CP decreases and that the inertia of the system increases from base to apex. Hence close to the base, the stiffness part of the interaction is dominant and governs the displacement of the CP. Close to the apex it's the inertia that plays this role. The point where the stiffness dominated region and the inertia dominated region meet is the resonance point of the CP. This point is characterized by the cancellation of the effect of the two forces since they are equal in magnitude and exactly opposite in phase. The CP admittance is maximal at this point.

Another factor that affects the CP displacement is the damping force. The damping force originates in the viscous character of the perilymph and in the friction of neighbouring cochlear cross-sections. The damping causes a decrease of the admittance particularly at the resonant point. The final form of the admittance is shown in Figure 2.3A.

The Figure 2.3B represents the pressure difference across the CP. The pressure

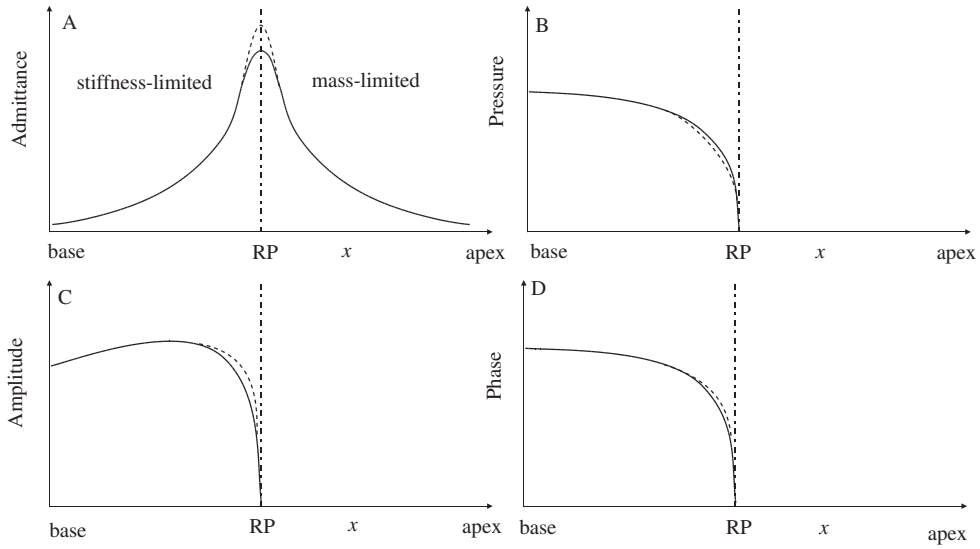


Figure 2.3: **Interpretation of the passive cochlear mechanics.** RP - resonance point, x - position in the cochlear duct. Solid lines correspons to the system with stiffness, mass and damping; dashed lines correspond to the system with decreased damping; dash-dot vertical line designates the position of the resonanace point. Figure is a reproduction of Figure 3.15 in Pickles (2012).

difference is high at the cochlear base. Here in the upper scala vestibuli the input vibration is initiated and the lower scala tympani is only waiting for this stimulus to arrive. The pressure difference then decreases and sharply drops to the zero value at the resonant point. The drop is caused by the high CP admittance that enables the pressure wave to pass through the basilar membrane into the scala tympani before reaching the helicotrema. In the remaining apical portion the pressure difference is zero.

From the derived dependences of the admittance and of the pressure difference along the cochlear duct we can derive the shape of the envelope of the BM displacement. This is plotted in Figure 2.3C. As can be seen in the figure, the amplitude first increases shallowly, then reaches the maximum at the basal proximity of the resonant point and then drops rapidly to zero at the resonant point. The phase change of the BM displacement in the cochlear duct is shown in Figure 2.3D. The phase of the BM displacement decreases from base to the resonant point and then holds a constant value up to the apex.

The reason why the maximum position of the BM displacement envelope changes with the input vibration frequency consists in the frequency-dependent inertia term. As an illustration, for higher input frequency the inertia forces are relatively larger but the stiffness remains unchanged. The resonant point therefore switches to a position closer to the base what modifies the pressure difference, the amplitude and the phase change curves in the same way. The BM is therefore activated mostly in the basal part.

The BM displacement corresponding to the above presented theory, i.e. possessing the amplitude plotted in Figure 2.3C and the phase change plotted in Figure 2.3D, corresponds well with the notion of the travelling wave that was observed experimentally.

2.1.2 Active cochlea

The von Békésy's experiments on cochleae of human and animal cadavers initiated a growing interest of the researchers on the cochlear behaviour. The researchers started to study the cochlear responses to various stimuli in various conditions and using various experimental settings. One of the settings in which the cochlea showed an exceptionally particular behaviour was the *in vivo* setting. Experiments performed on living organisms led to the observation of several new phenomena that had not been observed before.

One of such phenomena was observed by Kemp (1978). Kemp studied the responses of the ear canal of the outer ear to the acoustic stimuli near the lower threshold of audibility. Particularly, he measured the sound pressure at the ear canal in time intervals much larger than was the time interval of the input stimulus. He performed the experiment for both the dead and the living cochleae. The experiment revealed a secondary response to the input stimulus that appeared in a time delay and only in the living cochlea. Kemp identified this behaviour as the emission of a signal by the cochlea

itself and not as a passive consequence of the input stimulus. The living cochlea was therefore proved to be an active element.

Another phenomenon was observed by Russell and Nilsen (1997) who performed a series of experiments on anaesthetized (i.e. living) guinea pigs. Russell and Nilsen stimulated the cochlea with tones of a single frequency 15 Hz but with different intensities from 15 dB SPL to 100 dB SPL and they measured the amplitude of the BM displacement as a function of the position along the cochlear duct. The obtained results are represented in Fig5. The first observation that can be made in this figure is the change in the width and in the position of the main peak. For the lowest values of the signal intensity the peak is narrow and situated at the characteristic position (CPo) for the 15 Hz frequency. As the intensity increases the peak becomes broader, its slope on the basal side becomes shallower and its position shifts to the basal proximity of the CPo. The spatial selectivity of the frequency is therefore more significant at lower signal intensities than at the higher ones. The second observation concerns the nonlinear increasing trend of the BM displacement at the CPo position with the increase of the signal intensity. Figure 2.4 shows that the increase of the signal intensity by 20 dB causes a larger increase of the BM displacement for lower signal intensities than for higher signal intensities. This provides evidence of the nonlinearity of the active process.

The observed phenomena revealed that the living cochlea must contain an active

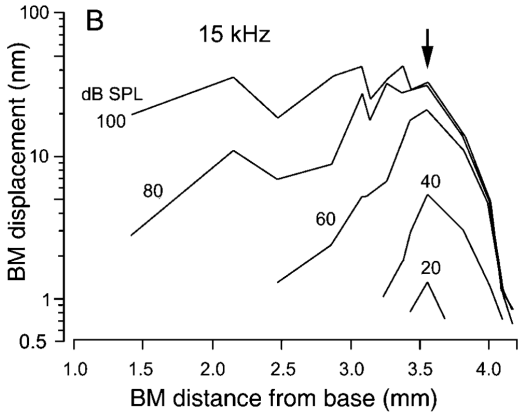


Figure 2.4: **Amplitude of the BM displacement at various intensities of the input signal and at 15 Hz.** Arrow designates the CPo. Figure is the reprinted figure 3.10B from Pickles (2012); the data are originally adopted from Russell and Nilsen (1997), Figure 1D.

element and that the active response is nonlinear and saturating for high signal intensities. This assumption was confirmed by theoretical models. In essence, models that included only passive elements were not able reproduce the observed shape of the travelling wave envelope: they could not capture the height and the width of the peak at the characteristic position. Introducing an active element in a limited region around the peak led to a better reproduction of the observed results. The active re-

gion is supposed to be activated by the travelling wave that passes through it and to subsequently feed the mechanical energy back to the travelling wave (see Pickles (2012, section 3.2.3.2)). This manifests as the amplification of the travelling wave amplitude at positions of the active region and the related sharp tuning of the travelling wave. The leading idea presented for example in Pickles (2012) is that the cochlear mechanics has two components: the passive one (with small amplitude and broadly tuned) and the active one (with greater amplitude and sharply tuned). At low intensities, the active component makes a relatively large contribution and causes the sharp tuning of the BM behaviour. Increasing the signal intensity the active mechanics loses its importance and it is the passive mechanics that becomes predominant, causing the broad tuning of the BM behaviour. Experiments show that the elements responsible for the active behaviour of the cochlea are the outer hair cells (OHCs) present in the organ of Corti. The protein prestin incorporated in their membrane enables the OHCs to contract and to dilate and to modulate in this way the displacement of the BM. More on the OHC's activity will be presented in the following section.

2.2 Electrical part

In the previous section we presented the mechanical processes that arise in the cochlea as a response to an external stimulus. Here we go further and present the mechanisms that are triggered by these mechanical processes. The triggered mechanisms happen at the level of cochlear hair cells and are related to a current flow, hence they belong to the electrical part of the response. Before explaining the mechanisms of the electrical part we present some additional information about the affected cochlear elements.

2.2.1 Additional information about cochlear components

The information in this section are adopted primarily from the chapters 3 and 5 in Pickles (2012).

Cochlear fluids

The principal cochlear fluids are the perilymph and the endolymph. As was already mentioned in the chapter 1, the perilymph fills the scala vestibuli and the scala tympani, two compartments that communicate directly via helicotrema. The endolymph fills the central cochlear scala, the scala media. Perilymph and endolymph play a significant role in the transduction process, notably thanks to their specific ionic composition and to the level of the electric potential.

Endolymph is characterized by possessing a high concentration of K^+ cations ($\approx 150\,\mathrm{mM}$) and a low concentration[2] of Na^+ cations ($\approx 1.3\,\mathrm{mM}$). This cation concen-

[2]The terms *high* and *low* are taken with regard to the concentration of any usual extracellular fluid

trations make the endolymph similar to an intracellular fluid (the similarity is mainly in the K^+ concentration). However, endolymph is also characterized by a high electrical potential that makes it completely unique in the set of human body fluids. While the usual intracellular potential varies between $-45\,\text{mV}$ and $-60\,\text{mV}$ and a usual extracellular potential takes values between 0 and $6\,\text{mV}$, the potential of the endolymph reaches from $87\,\text{mV}$ up to $100\,\text{mV}$ (the potential varies depending on the position in the cochlear duct, being higher near the base and lower near the apex, see Pickles (2012, section 3.3.2)) . The origin of this potential has been the object of many studies. Firstly it was shown that the endolymphatic potential is not the K^+ nor Na^+ diffusion potential: not the first one because it would be of the opposite sign, not the second one as it was shown experimentally by Johnstone and Sellick (1972). Secondly, many studies revealed that the key role in the maintaining of the endolymphatic potential is played by the stria vascularis (see Hibino et al. (2010)). A so-called two-cell model has been developed to explain mechanisms occuring in the stria vascularis when generating this potential (see Pickles (2012, section 3.3.2.2)). According to this model, the endolymphatic potential is generated as the K^+ diffusion potential across the membrane of the intrastrial cells. The diffusion potential in the stria is due to the very low concentration of K^+ cations in the intrastrial fluid in comparison with the high K^+ concentration in the intrastrial cells. The low K^+ concentration in the intrastrial fluid is maintained by ion-pumping processes[3] (see Pickles (2012, section 3.3.2.2) and Marcus et al. (2002)).

Perilymph is in its ionic composition similar to most extracellular fluids since it has the concentration of $K^+ \approx 5\,\text{mM}$ and of $Na^+ \approx 140\,\text{mM}$ (see Delprat (2016)). Its electrical potential is $\approx +6\,\text{mV}$, $+7\,\text{mV}$ in the scala tympani and $+5\,\text{mV}$ in the scala vestibuli (see Pickles (2012, section 3.3.3)). The difference in potentials of the endolymph and the perilymph, i.e. $\approx 80\,\text{mV}$, is referred to as the *endocochlear potential*.

Other cochlear extracellular fluids that the basal part of hair cells are in contact with have the same ionic composition as the perilymph but have a little lower electric potential of approximately $0\,\text{mV}$.

Hair cells

Hair cells are the receptor cells of the auditory transduction process. They are able to capture a mechanical stimulus by their apical end, then to process this information inside and to transform it into a sequence of neurotransmitter quanta that are finally released by their basal end to the neurons.

The apical end of a hair cell is formed by the cuticular plate and by three to five rows of small projections called stereocilia that rise from the plate (the ensemble of the stereocilia of one hair cell will be referred to as a *hair cell bundle*). Each stereocilium is formed by longitudinally superposed actin filaments that give it the significant rigidity.

[3] *Ion pumps* transfer ions in the diffusion gradient direction, i.e. from the low concentrated site to the more concentrated site, consuming the cellular energy stocked in ATPs

The flexibility of motion of a stereocilium is given only to the point where it emerges from the cuticular plate. A stereocilium can deflect as an entity but not bend at any point of its length (Pickles (2012, section 5.2.1)).

Stereocilia of a hair cell bundle are cross-linked in numerous ways. First, the stereocilia of different rows and the stereocilia of the same row are connected by side links. These couple the sides of the stereocilia almost parallel to the cuticular plate (see Figure 2.5). Second, the tip of the shorter stereocilium is connected to the side of the taller stereocilium by a tip link. Tip links connect only the stereocilia of different rows. While the side links have purely mechanical function, the tip links intervene actively in the transduction mechanism. The mechanical function of the side links consists in the mechanical coupling of the stereocilia of a bundle. Therefore, if a mechanical action is applied to one part of the bundle, this is immediately transferred to the whole bundle. The deflection of the stereocilia of a bundle is therefore always uniform. The transduction function of the tip links results from the fact that they are attached by their one end to mechanotransducer channels (MCs) of a hair cell. MCs are small gaps in the hair cell membrane situated in the apical end of the stereocilia. If opened, they let the cations pass directly into the hair cell interior(see Pickles (2012, section 5.2.2.1)).

The way in which the tip links are attached to MCs is not known. The model

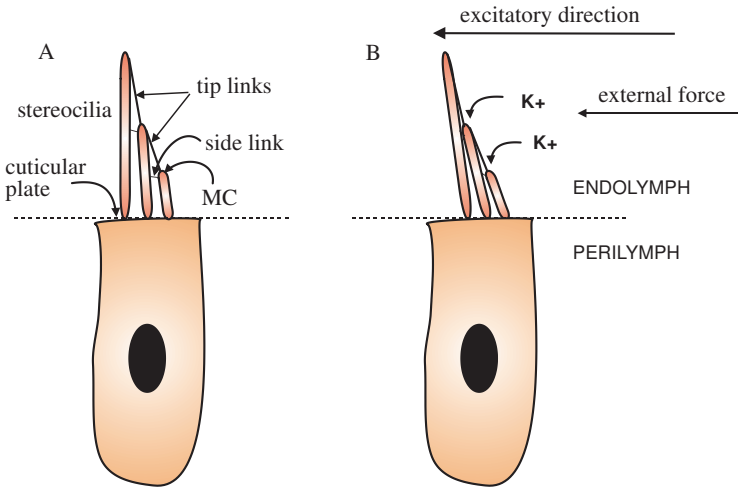


Figure 2.5: **Hair cell. (A)** hair cell at rest, **(B)** hair cell with applied external force in the excitatory direction - stereocilia deflect and the MCs open, allowing the transfer of ions between the cell interior and the endolymph.

that researchers suggest presents the action of the tip link to the channel as a direct mechanical action: the channel is opened by a tip link pull. When no tension from the tip link is applied, MCs are found to fluctuate between the opened and closed state by the influence of the thermal energy. The probability of finding the channel in one or another state is therefore given by Boltzmann distribution. When a tension is applied, a redistribution of states occurs and the probabilities change. To describe the behaviour of MCs in response to a mechanical action the researchers propose the

so-called *gating-spring theory*. The theory assumes that channels are pulled open by an elastic link, the gating spring, and it determines the probabilities of the opened and closed state under the effect of this spring. Both the side and the tip cross-linking of the stereocilia have a decisive effect on the passage of cations through MCs. If some stereocilia of the hair cell bundle are deflected in the direction towards the tallest stereocilia (the excitatory direction), all stereocilia of the bundle deflect in the same way, the tip links are stretched and the probability of finding the MCs opened rises. In reverse, the deflection of the stereocilia in the direction towards the smallest stereocilia (the inhibitory direction) slackens the tip links and the probability of finding the channels opened falls (see (Pickles, 2012, section 5.3.2)).

The apical end of hair cells is entirely surrounded by the endolymph. The hair cell interior fluid and the endolymph have approximately the same concentration of K^+ while the hair cell interior fluid has a higher concentration of Na^+ than the endolymph. The essential difference between these two fluids is related to their electrical potential. While the endolymph has the potential +80 mV, the hair cell fluid has −45 mV in the case of inner hair cells and −75 mV in outer hair cells. There exists therefore a potential difference of 125 mV or even 155 mV between these two media. The basal end of hair cells baths in the extracellular fluid that is similar to the perilymph in its ionic composition and has the electrical potential ≈ 0 mV.

The basal end of hair cells is in direct contact with afferent and efferent nerve fibres. The majority of the afferent auditory nerve fibres make the contact with inner hair cells (IHCs). IHCs are therefore the ones that play the main role in the transduction process. The outer hair cells (OHCs) serve mainly to amplify the movement of the basilar membrane.

2.2.2 Cochlear activity induced by the BM displacement

Mechanotransduction

A mechanical vibration applied to the oval window of the cochlea induces a travelling wave of the BM displacement along the cochlear duct. The travelling wave initiates a movement of the endolymph in the scala media and of the attached tectorial membrane. The relative movement of the basilar membrane with respect to the tectorial membrane causes the deflection of hair cell bundles of OHCs. The viscous drag of the endolymph on the stereocilia causes deflection of the hair cell bundles of IHCs. When stereocilia are deflected in the excitatory direction, the probability of the opening of MCs rises. When an MC is opened, the potential difference across the apical membrane of the hair cell induces a flux of K^+ cations directing from the endolymph to the intracellular fluid of the hair cell. The entry of K^+ cations into the hair cell causes the change of the intracellular potential to a less negative value, i.e. it causes the hair cell *depolarization*. The depolarization of the hair cell in turn activates opening of voltage-gated Ca^{2+}

and K^+ channels situated at the basal end of the hair cell. The opening of Ca^{2+} channels induces Ca^{2+} entry that stimulates the release of neurotransmitter by the basal part of the hair cell as well as the opening of Ca^{2+}-dependent K^+ channels. Since the equilibrium potential[4] for K^+ is more negative than the hair cell's resting potential, the difference in the concentration of K^+ across the basal membrane of the hair cell drives the efflux of the K^+ cations from the intracellular medium through the K^+ channels towards the intercellular medium. The efflux of K^+ causes the hair cell *hyperpolarization*, i.e. the change of the hair cell potential to more negative values. The K^+ is then recirculated by various types of supporting cells and other components back to the endolymph.

The transduction process is therefore associated to the K^+ recirculation through IHCs and OHCs. A hair cell acts in this process as a two-part component, each of them characterized by its own equilibrium potential for K^+. For more information about the mechanotransduction processes see Pickles (2012), Purves D (2001) and Mistrík et al. (2009).

Mechanical amplification of the travelling wave

Multiple experiments showed that the cochlear components that are responsible for the active modulation of travelling waves are the OHCs. As an example, Liberman and Dodds (1984) proved that when OHCs are destructed, the cochlear response to a mechanical stimulus is not sharply tuned. The question about how OHCs actively influence the cochlear behaviour is up to now not definitively answered. We consider two active OHCs motions to be the candidates for causing this behaviour. The first type of the OHC's active motion is the change in their length that occurs as a response to potential changes inside the cell. Experiments performed by Brownell et al. (1985) on isolated OHCs showed that when an OHC is depolarized, it reduces its length while when it is hyperpolarized, it increases it. Both length changes happen by relatively large amounts: they can be seen by light microscope (Brownell et al. (1985)) and may take up to $4-5\,\%$ of the cell length (Ashmore (1987)). As it was shown later, the length changes are primarily associated with the presence of motor proteins in the OHC plasma membrane (see for example the review article Nobili et al. (1998)). These have the ability of modifying their conformation when the membrane potential is changed. The deformation provoked by the motor proteins is transmitted along the longitudinal axis of the cell through the medium of the cytoskeleton[5]. The described mechanism is currently considered as the primary factor responsible for the active cochlear processes. The way how the OHC's length changes modulate the displacement of the BM is

[4] The *equilibrium potential* for an ion is the membrane potential where the net flow through any open channel is 0 (definition taken from Fitzakerley (2014))

[5] *Cytoskeleton* is a highly elastic structure bordering the inner face of the plasma membrane (Nobili et al. (1998))

however not exactly determined. A small review on the considered possibilities can be found in Pickles (2012, section 5.4.5.1).

The second type of the OHC's active movement is related to that of the OHC's hair bundle. It was shown that stereocilia may exhibit spontaneous active movements and that these movements may potentially generate a force magnifying the BM displacements. The real effect of the hair cell bundle motility on the cochlear active processes remains however uncertain. A discussion on this topic can be found in Pickles (2012, section 5.4.5.2) or Fettiplace (2006).

3. Mathematical model of the hearing process in the cochlea

3.1 Mechanical part

The mechanical part of our mathematical model simulates the BM response to input stimuli. We employ the model that is by the great majority based on the model created by F. Mammano and R. Nobili. Their ideas of the mathematical description of hearing mechanics are summarized in Mammano and Nobili (1993) and Nobili and Mammano (1996). Here we present the most essential elements of these ideas. For detailed information see the cited publications.
The essential part of the cochlea that needs to be mathematically described is the organ of Corti. The organ of Corti is a set of ≈ 3000 adjacent transversal segments of a unicellular width, each of them containing three OHCs and generally one IHC. The segments are lying on the basilar membrane and are in contact with the underlying tectorial membrane.

3.1.1 Passive cochlea

Principles

In our model we consider the combination of an organ of Corti transversal segment with the underlying basilar membrane segment as an element (referred to as a *BM element*) possessing a given mass and stiffness and oscillating in the transversal plane. The transversal motion of a BM element is driven by the force of the stapes that is transmitted by cochlear fluids from the oval window to that element. The motion is damped by viscous forces of the surrounding fluid and by the shearing motion between adjacent elements. The motion is also affected by the BM-BM coupling force. Mammano and Nobili (1993) characterize this force as the one that is exerted by one BM element to another BM element by the effect of their interaction with the surrounding fluid. The model neglects elastic longitudinal couplings of BM elements.

Mathematical model

In order to make the problem mathematically solvable i.e. to be able to establish explicit mathematical equations of the BM motion, we consider the BM with the underlying components as a unidimensional continuum in the longitudinal direction. We introduce the continuous position coordinate x from the interval [0,1] that represents the position along the unrolled cochlear duct. The $x = 0$ point lies at the basal end of the cochlea where the external force of the stapes is applied. The $x = 1$ point is situated

at the apical end of the cochlea close to the cochlear helicotrema. We denote by $u(x,t)$ the BM displacement in the transversal direction at the position x and at the time t. Then we can establish the motion equation for $u(x,t)$ as follows:

$$m(x)\frac{\partial^2 u(x,t)}{\partial t^2} + h(x)\frac{\partial u(x,t)}{\partial t} - \left[\frac{\partial}{\partial x}s(x)\frac{\partial}{\partial x}\right]\frac{\partial u(x,t)}{\partial t} + \\ + k(x)u(x,t) = F_S(x,t) + F_{BM}(x,t). \quad (3.1)$$

Following Mammano and Nobili (1993), the term-by-term analysis of the equation (3.1) is expressed in the following overview.

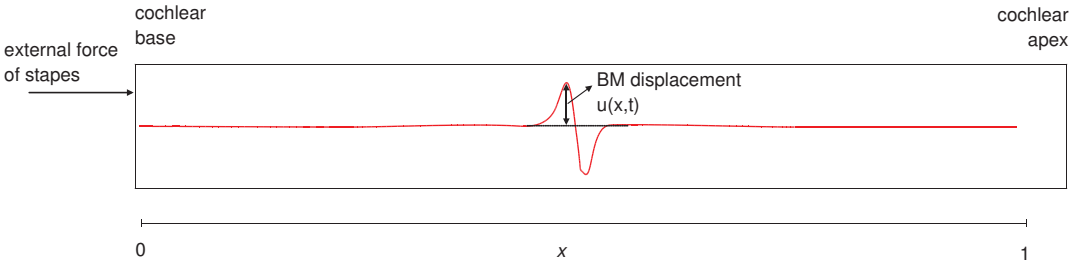


Figure 3.1: **Mechanical model setting, as designed by Mammano and Nobili (1993).** x is the position in the unrolled cochlea.

$m(x)\frac{\partial^2 u(x,t)}{\partial t^2}$	**mechanical inertial term**; $m(x)$ is the mass per unit length of the organ of Corti
$h(x)\frac{\partial u(x,t)}{\partial t}$	**fluid viscosity term**; $h(x)$ is the endolymph viscosity multiplied by the factor related to the geometry of the TM-reticular lamina (RL) cleft
$\left[\frac{\partial}{\partial x}s(x)\frac{\partial}{\partial x}\right]\frac{\partial u(x,t)}{\partial t}$	**shearing resistance term**; the term represents the resistance due to the shearing motion between adjacent segments of the organ of Corti; $s(x)$ is the average shearing viscosity coefficient of the organ of Corti multiplied by the effective area of the section at the position x
$k(x)u(x,t)$	**fiber stiffness term**; $k(x)$ is the elastic constant of the cochlear partition, it receives contribution from the BM fibers and the limbus-TM-stereocilia-RL system
$F_S(x,t) = -G_S(x)\frac{\partial^2 \sigma(t)}{\partial t^2}$	**stapes force term**; $\sigma(t)$ is the stapes displacement at time t; $G_S(x)$ is the Green's function representing the stapes-BM hydrodynamic coupling, $G_S(x)$ gives the force per unit length caused by the unit stapes acceleration and acting on the BM element at x
$F_{BM}(x,t) = -\int_0^1 G(x,y)\frac{\partial^2 u(y,t)}{\partial t^2}dy$	**BM force term**; $G(x,y)dy$ is the Green's function of the BM-BM hydrodynamic coupling, $G(x,y)dy$ gives the force contribution per unit BM length at site x caused by the unit acceleration of the dy-long BM segment at y

3.1.2 Active cochlea

Experiments have shown that models of the passive cochlea are not adapted to reproduce the behaviour of the BM displacement in living organisms. The reason is that they do not contain a sufficient number of degrees of freedom to reproduce the sharp peak in the BM displacement envelope produced at small ($0 - 40$ dB SPL) input signal intensities. In particular, they cannot fit the width and the height of the peak

at the same time. This finding led researchers to conclude that cochlea must provide a particular force contribution that had not been included into the existing model. As we have already stated in the section 2.2.2, subsequent experiments showed that the missing element in the interpretation of the cochlear mechanics is the force generated by OHCs via their capacity of electromotility (ie the active movements invoked by an electrical impulse). The problem that naturally arose concerned the way how this active element should be included into the models.

In our model we operate with the mathematical model of the active cochlear force proposed by F. Mammano and R. Nobili. Here we present only a brief overview of how they proceeded in the derivation of the active term and how they included it into the existing passive model. The full derivation and further comments are provided by Mammano and Nobili (1993) and Nobili and Mammano (1996).

Principles

The electromotility of OHCs has a significant effect on the BM displacement in the cochlear response to external stimuli. The interpretation of this mechanism by Mammano and Nobili (can be found in Mammano and Nobili (1993, section IV.A. and the related Figure 5)) is as follows: When an external stimulus induces the BM to move upwards, i.e. towards the scala media, the hair cell bundle of OHCs deflects in the excitatory direction. To understand why this happens we need to know two things. Firstly, there exists a shearing motion of the RL with respect to TM when the organ of Corti is rotated due to the BM displacement. Secondly, the tallest row of OHC stereocilia is firmly attached to the undersurface of the TM and the rest of the hair cell bundle is linked to the tallest row by side links (see chapter 2.2.1). The deflection of the stereocilia gives rise to a current flow to the OHC that depolarizes the cell and the cell contracts in its length. The contraction causes the organ of Corti to be pulled back downwards, i.e. towards the scala tympani. The RL is therefore pulled toward the BM, see Figure 3.2. An opposite situation, i.e. when the BM is first displaced downwards by an external force, results in the upwards final movement of the RL. The OHC active force is therefore always such that it acts in the opposite direction to the impulse displacement.

The observed properties of an OHC leads to the suggestion that the OHC works as a spring possessing a stiffness in the longitudinal direction of the cell. Mammano and Nobili (1993) based the derivation of the active force (also referred to as *undamping term*) on this idea, stating that "undamping term (...) is a pseudoelastic force term, i.e. a term proportional to the elastic reaction (...) of the BM, to be added to the lhs of Eq. (3.1)) with a positive sign". Nobili and Mammano (1996) provide an extension to this suggestion stating that "In Part I (...) we assumed that the OHC force is simply proportional to the local BM stiffness. This is legitimate only provided that the internal viscosity of the OHCs and Deiter's cells (DC) is negligible." Since the research

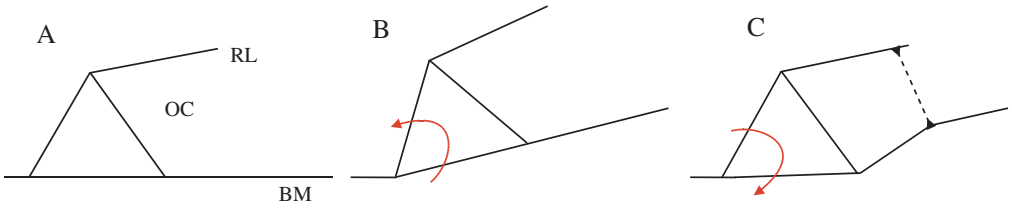


Figure 3.2: **Scheme of the active dynamics of the basilar membrane.** The scheme represents the transversal section of the cochlea from Figure 1.2 (left); the base line represents the BM, the triangle represents the tunnel of Corti formed by pillar cells and the upper line represents the reticular lamina. **(A)** no external force is applied; **(B)** external force on BM is applied and forces the BM to move upwards; **(C)** - contraction of OHCs pushes the organ of Corti to move downward. Figure inspired by Figure 2B in Nobili et al. (1998).

provided by Mammano and Ashmore (1993) shows that the damping elicited by OHCs and DCs plays an important role in the RL-BM coupling, Mammano and Nobili corrected their initial model by incorporating this term. Further experiments show that the RL-BM coupling is dominated by the damping of the DCs, hence it is characterized as a viscous interaction. The RL-BM coupling is in the model of Nobili and Mammano (1996) finally simulated as two spring-dashpot systems branched in series, see Figure 3.3.

Another structure that intervenes in the undamping mechanism is the tectorial membrane. Mammano and Nobili (1993) consider two viscoelastic connections of a segment of the TM: one to the spiral limbus, the other one to a triplet of OHC stereocilia. Measurements performed by Zwislocki et al. (1988) show that the stiffness of the TM attachement to stereocilia is approximately seven-times greater than the stiffness of its attachement to the limbus. The attachement to the limbus is therefore considered as mainly a viscous one and the TM-stereocilia system is viewed as a set of resonant mechanical systems. Each mechanical system is formed by one transversal segment of the system TM-stereocilia with the unicellular width. This implies that the system contains three OHC hair bundles and the corresponding portion of the TM. The mass of the resonant system is formed by the TM segment mass, the elasticity is provided by the three bundles of OHC stereocilia and the damping is furnished by the fluid in the TM-RL cleft. The harmonic motion of this system is performed in parallel with the reticular lamina and triggers the opening/closing of the mechanotransducer channels.

Mathematical model

The goal of this section is to present the derivation of the undamping force term $\mathrm{U}[x,t;u]$ that Nobili and Mammano (1996) add to the motion equation for the BM displacement (3.1). $\mathrm{U}[x,t;u]$ is a functional of $u(x,t)$ and it is linked to the condition $\mathrm{U}[x,t;0] = 0$. Its addition to the equation (3.1) is supposed to ensure the nonlinear undamping behaviour of the solution. The full derivation with additional comments

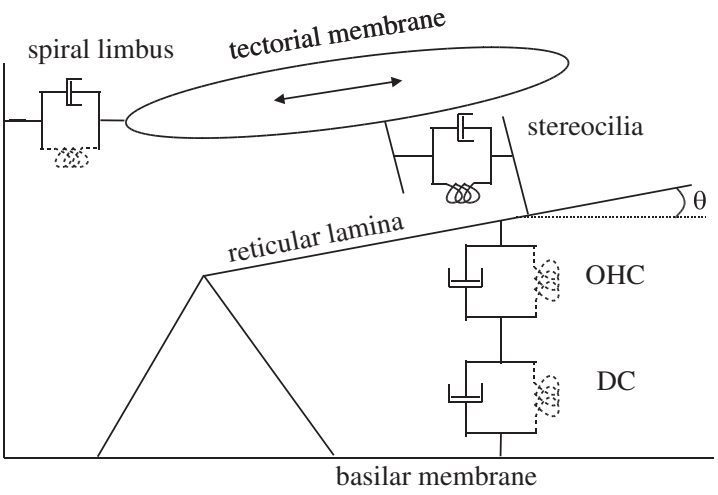


Figure 3.3: **Model of the active dynamics of the organ of Corti designed by Nobili and Mammano (1996).** Dashed lines represent the elements that Nobili and Mammano (1996) consider in their model as very small in comparison with other elements. Figure is inspired by figure 2C from Nobili et al. (1998).

can be found in Nobili and Mammano (1996).

Nobili and Mammano start with introducing the Fourier transform of $\mathrm{U}[x,t;u]$ with respect to time, denoted by $U_\omega[x;u]$. They propose the relation between $\mathrm{U}_\omega[x;u]$ and the Fourier transform of the OHC intracellular (receptor) potential $\mathrm{V}_\omega[x;u]$ as a linear one of the form

$$U_\omega[x;u] = \frac{\mathrm{d}F_\omega(x)}{\mathrm{d}V} V_\omega[x;u], \tag{3.2}$$

where $\mathrm{d}F_\omega(x)/\mathrm{d}V$ is the rate of change of the OHC force with the variation of the receptor potential. The term $\mathrm{d}F_\omega(x)/\mathrm{d}V$ combines the stiffness and the damping coefficient of OHCs and DCs and the OHC cell length. $\mathrm{V}_\omega[x;u]$ is further expressed in terms of the Fourier transform of the transducer current $\mathrm{I}_\omega[x,t;u]$ via the linear relation

$$V_\omega[x;u] = \frac{1}{b_\omega(x)} I_\omega[x;u], \tag{3.3}$$

where $b_\omega(x)$ represents the resistance-capacitance (RC) filtering of the cell membrane. Combining (3.2), (3.3) and the assumption that the viscoelastic RL-BM coupling cancels the RC filtering of the OHC membrane, if we return to the time domain we get the relation between the OHC force and the transducer current in the linear form

$$U[x,t;u] = c(x) I[x,t;u], \tag{3.4}$$

where $c(x) = \mathrm{d}F_\omega(x)/\mathrm{d}V \times 1/b(x)$.

As we have already explained in the section 2.2.1, the intensity of the transducer current depends on the probability of the opening of mechanotransducer channels (MCs). This probability is dependent on the stereocilia deflection and the deflection is caused by the shearing motion between TM and RL. The shearing motion is finally due to the BM displacement. Let's now describe this process mathematically.

Nobili and Mammano (1996) propose the relation

$$I[x,t;u] = I_S(x)S[x,t;u], \tag{3.5}$$

where $\mathrm{I}_S(x)$ is the saturation value of the transducer current at the position x and $\mathrm{S}[x,t;u]$ represents the transducer current through MCs corresponding to the BM displacement $u(x,t)$ and scaled as: $\sup(\mathrm{S}) - \inf(\mathrm{S}) = 1$. To build the dependence of S on the BM displacement u, Nobili and Mammano (1996) introduce an auxiliary functional representing the hair bundle deflection denoted by $y[x,t;u]$. The dependence of S on y is expressed using a three-level Boltzmann's partition function:

$$\mathrm{S(w)} = \frac{1}{1 + \mathrm{c_1 exp(-w/w_1)} + \mathrm{c_2 exp(-w/w_2)}} - \mathrm{b}, \tag{3.6}$$

where

$$w := w[x,t;u] = a(x)y[x,t;u] \tag{3.7}$$

and c_1, w_1, c_2, w_2, b and $a(x)$ are suitable coefficients. This expression incorporates the thermodynamical as well as the mechanical origin of the channels opening.

The dependence of the hair cell bundle deflection y on the BM displacement u is a result of the cochlear micromechanics. The driving force for the hair cell bundle deflection (that we consider to be performed in the radial direction[1]) is the radial component of the RL displacement. The radial component of the RL displacement is assumed to be the BM displacement $u(x,t)$ multiplied by the sinus of the angle θ formed by the RL and BM surfaces (this varies from $\approx 30°$ at the apex to $\approx 0°$ at the base). Hence for the time Fourier transform of $y[x,t;u]$, $y_\omega[x;u]$, it holds

$$y_\omega[x;u] = Z_\omega(x)u_\omega(x)\sin(\theta(\mathrm{x})), \tag{3.8}$$

where $Z_\omega(x)$ accounts for the viscoelastic properties of the TM attachements to the limbus and to the stereocilia. With the assumption that the TM-limbus attachment is primarily viscous, $Z_\omega(x)$ takes the form

$$Z_\omega(x) = \frac{i\omega}{\gamma_{TM} - i\left[\omega_{TM}^2(x) - \omega^2\right]/\omega}, \tag{3.9}$$

where ω_{TM} is the TM resonant frequency and γ_{TM} is the TM damping coefficient per TM mass.

Let's now suppose that $\omega_{TM}(x) \approx \omega_{CF}(x)$ for all x, where $\omega_{CF}(x)$ is the characteristic frequency (CF) of the position x. Then for ω from a small interval around the CF, the imaginary part of the denominator of the expression (3.9) is close to zero. Hence we can neglect it provided that $\gamma_{TM}(x)$ is large enough. Under these conditions the magnitude of $Z_\omega(x)$ is a linear function of ω that causes that also the magnitude of $y_\omega[x;u]$ is a linear function of ω.

[1] By *radial direction* in the cochlea we mean the direction orthogonal to the coiling cochlear axis and parallel to the BM fibers.

Let's now consider small BM displacements. This induces small deflections of the hair bundle y and the relation (3.5) can be reduced to a linear one with the slope equal to 1:

$$I[x,t;u] \approx I_S(x)a(x)y[x,t;u]. \tag{3.10}$$

Hence for small BM displacement and for $\omega \approx \omega_{TM} \approx \omega_{CF}$, the combination of (3.8), (3.9), (3.10) and (3.4) gives

$$U_\omega(x) = i\omega\beta(x)u_\omega(x) \tag{3.11}$$

with

$$\beta(x) = \frac{c(x)I_S(x)a(x)\sin\theta(x)}{\gamma_{TM}}. \tag{3.12}$$

We obtained therefore $U_\omega(x)$ such that as the magnitude of $U_\omega(x)$ is a linear function of ω. This is exactly the result we wanted to obtain since it is adequate to oppose the damping term $i\omega h(x)u_\omega(x)$ present in the Fourier transformation of the motion equation (3.1) for the passive cochlea. By setting $\beta(x)$ so that

$$\beta(x) \approx -h(x) \tag{3.13}$$

and by adding the active force U to the left-hand side of the equation (3.1), the damping term is entirely compensated by U and the solution exhibits a sharply tuned peak near the CF site. At higher input amplitudes the transducer current becomes saturated hence the undamping force term U looses its importance with respect to the damping term. In this case the damping causes that the solution exhibits the broadly tuned peak characteristic for the passive cochlea.

The inverse Fourier transform of (3.8) with the recalculated $Z_\omega(x)$ using coefficients following from (3.13) gives the motion equation for the hair bundle displacement in the radial direction, denoted by $y(x,t)$, as follows

$$\frac{\partial^2 y(x,t)}{\partial t^2} + \gamma_{TM}\frac{\partial y(x,t)}{\partial t} + \omega_{TM}^2 y(x,t) = -\sin(\theta(x))\frac{\partial^2 u(x,t)}{\partial t^2}. \tag{3.14}$$

Finally, the motion equation for the BM displacement of the active cochlea is

$$m(x)\frac{\partial^2 u(x,t)}{\partial t^2} + h(x)\frac{\partial u(x,t)}{\partial t} - \left[\frac{\partial}{\partial x}s(x)\frac{\partial}{\partial x}\right]\frac{\partial u(x,t)}{\partial t} + \\ + k(x)u(x,t) + U(x,t) = F_S(x,t) + F_{BM}(x,t). \tag{3.15}$$

It is worthy to notice in this derivation that the assumption $\omega \approx \omega_{TM} \approx \omega_{CF}$ has an particular physical meaning. The assumption indeed says that there is a limited region around the CF site where the undamping force is active. In the region more distant from the CF site the assumption is not valid and the undamping force has the form that may not directly compensate the damping force.

3.2 Electrical part

The electrical part of our model simulates the K^+ circulation in the cochlea. The K^+ circulation reacts in a particular way to the displacements of the basilar membrane, it is therefore directly connected to the mechanical model. On the other side, the K^+ circulation triggers the hair cell neurotransmitter release hence it must be connected to the model of the hearing nerve.

The model that we are operating with originates in the work of P. Mistrík, C. Mullaley, F. Mammano and J. Ashmore, see Mistrík et al. (2009). It has been further developed and refined by P. Mistrík and J. Ashmore (see Mistrík and Ashmore (2010)) and O. Tichácek (see Tichácek (2014)). In the following overview we present only the basic ideas of the model; for detailed information see the cited publications.

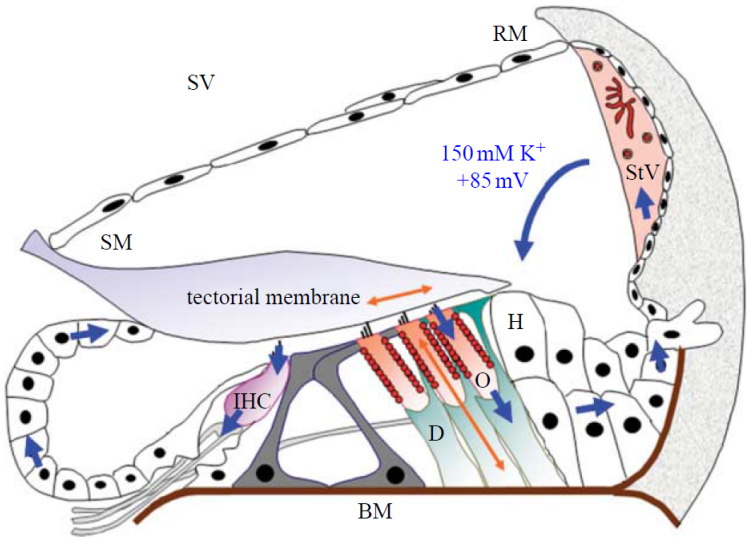
Cochlea as a system of electrical circuits

Let us first specify the details concerning K^+ circulation in the cochlea, as it is seen by Mistrík et al. (2009). Mistrík et al. (2009) consider a cochlear cross-section in the transversal direction with the unicellular width (as defined in the mechanical part) and they account for two parallel K^+ passages in this cross-section: the first one passing through IHC and the second one passing through OHCs. The K^+ flow through OHCs has been experimentally well explored: K^+ passes from the scala media through mechanotransducer channels into the OHC, then it is released into the intercellular space of the organ of Corti through K^+ channels in the basal membrane of the OHC, then it passes to the perilymph, then it is actively pumped into the spiral ligament, then to the stria vascularis and finally back to the scala media. We have less information about the passage of the K^+ flow through the IHC, however we suppose that it is happening independently from the OHC passage and that it passes through the spiral limbus. Both of the passages are represented in Figure 3.4a. The K^+ flow is driven predominantly by the endocochlear potential. This is maintained at a high value due to the active processes in the stria vascularis (see section 2.2). Furthermore, the model assumes the K^+ flow also in the longitudinal direction. This is realized specifically in the stria vascularis, organ of Corti and spiral limbus.

Mistrík et al. (2009) dissects the cochlear duct into a set of 300 transversal segments (referred to as cross-sections), each of them possessing the width of multiple cells. Each cross-section is then translated to an electrical circuit where the cochlear structures are represented by passive or active circuit elements. Passive elements are the resistances and capacitances and active elements are the batteries imposing a certain potential difference. The current in the electrical circuit is naturally carried by K^+ cations. The scheme of the circuit is represented in Figure 3.4c.

The electrical circuit is primarily formed by two subcircuits, one of them in-

volving IHCs and the other involving OHCs. IHCs and OHCs are represented by two-component circuit elements, see Figure 3.4b. One component substitutes the apical part of the hair cell membrane and it is composed of the capacitance and variable resistance branched in parallel. This element is branched in series with the other one that represents the basal part of the hair cell membrane. This second component is composed of two branches connected in parallel: one is formed by the capacitance and the other is formed by the resistance in series with the battery that imposes the voltage of $-100\,\mathrm{mV}$. This voltage represents the Nernstian driving force for K^+ ions, i.e. the force that drives the K^+ ions from the hair cell interior through the cell membrane to the exterior due to the fact that the membrane equilibrium potential for K^+ is more negative than the cell resting potential. The branches including IHC and OHCs contain also resistances ascribed to other cochlear structures, mainly to the organ of Corti for the OHCs branch and to the spiral limbus for the IHC branch. Other elements of the circuit are the battery, the resistance and the capacitance that represent the generated endocochlear potential and the properties of the stria vascularis. The individual electrical circuits are interconnected by branches with resistances of the organ of Corti, of the spiral limbus and of the stria vascularis, see Figure 3.4d.



(a)

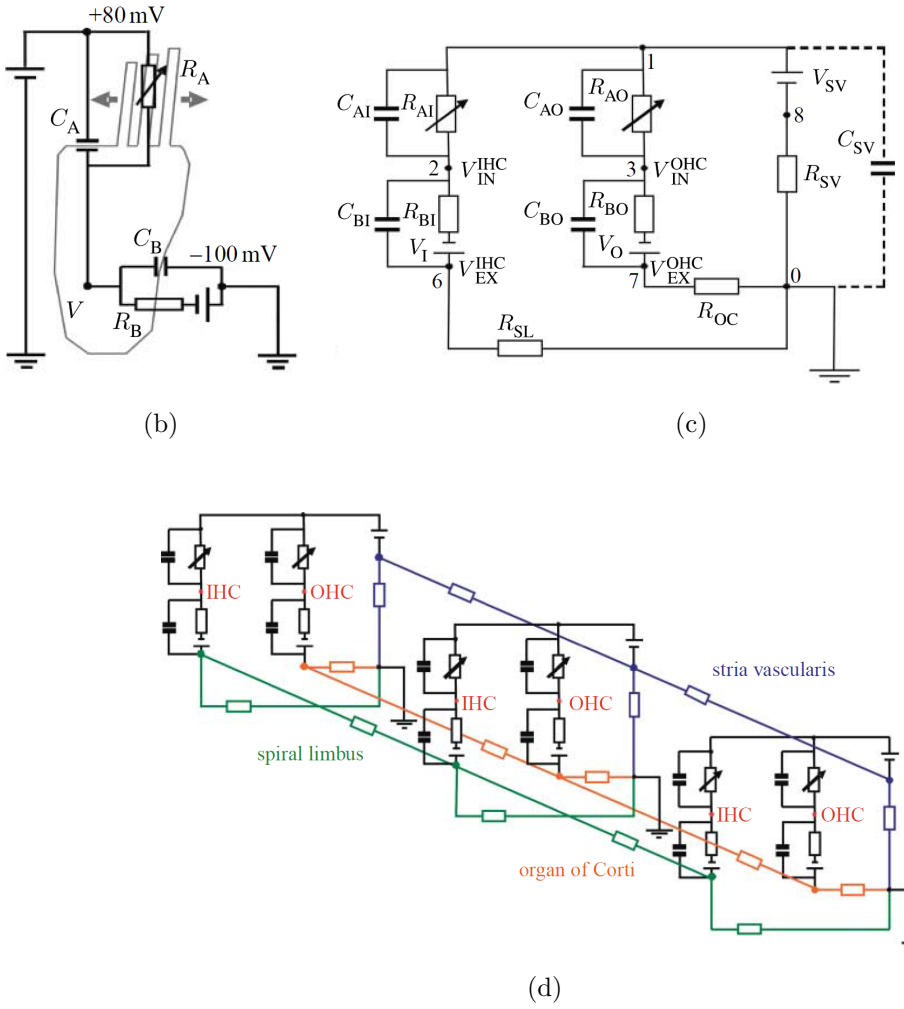


Figure 3.4: **Recirculation of K^+ in cochlea designed by Mistrík et al. (2009). (A)** K^+ recirculation in the cochlea; (StV) - stria vascularis, (RM) - Reissner's membrane, (SV) - scala vestibuli, (SM) - scala media, (D) - Deiters' cells, (H) - Hensen's cells, (O) - outer hair cell, (IHC) - inner hair cell; **(B)** schema of the electrical circuit of a hair cell, **(C)** electrical circuit corresponding to one cochlear transversal section; the most left branch corresponds to an IHC, the second left branch corresponds to an OHC, the most right two branches represent the stria vascularis; IN refers to intracellular, EX refers to extracellular; R_{SL} denotes the resistance of the spiaral limbus, R_{OC} refers to the resistance of organ of Corti; **(D)** illustrates the longitudinal electrical coupling of cochlear transversal sections. Figure reprinted from Mistrík et al. (2009).

Variable resistances of hair cells

The resistances R_{AI} and R_{AO} are variable in such a way that their magnitude can be expressed as a function of the BM displacement, yielding therefore the connection between the mechanical and the electrical models. R_{AI} and R_{AO} are indeed linked to the probability of opening/closing of mechanotransducer channels (MCs) and this is in turn related to the magnitude of the BM displacement.

Let's first see how the deflection of the stereocilia affects the probability of opening

of an MC. There have been multiple ideas how to describe this effect. A significant progress in the description has been brought by experiments of Corey and Hudspeth (1979). Corey and Hudspeth (1979) revealed that the opening/closing response of MCs to an external stimulus is very fast and furthermore it is temperature dependent. The fact that the latency of MCs is very small favors the simple, mechanics-based mechanisms. This idea is supported by the argument that if the contrary was true, i.e. if the mechanism involved some biochemical reactions, these would have to progress at unphysiologically high rates. The successful theories are therefore based on the idea that MCs are opened/closed by a mechanical action. The experimentally detected temperature dependence implies that this mechanical action must be triggered at least partially by thermal energy.

The probability of the MC opening is in our model given by the gating spring theory, presented for example in Markin and Hudspeth (1995). We use in particular a two-state gating-spring model which means that we account for two complementary states of an MC: opened or closed. The theory assumes that the opening/closing of the MC is in correspondence with the position of the gate that is attached to it. The gate is in turn attached to an elastic spring with the stiffness κ. Let us first consider the case of a single MC placed out of the cochlea. When the MC is opened, the spring is stretched a little, i.e. it bears a certain tension. In the situation when the channel passes from the opened state to the closed state (by the effect of the thermal energy), the gating spring stretches more and gains an additional amount of the tension. When it in turn passes to the original open state, the additional tension is released and the spring returns to the original state. This situation gives rise to a certain probability of the opened or closed state that is determined by the Boltzmann's distribution law. The situation is considerably changed when we introduce an external force applied to the spring. This force is considered to act by stretching or compressing the spring in the original state, i.e. by respectively increasing or decreasing the initial tension when the MC is opened. The effect of the additional force is the redefinition of the open/closed state probabilities (see Pickles (2012, section 5.3.2.6, Figure 5.11)). An analogical situation occurs when the MC is put into the cochlear environment. Here, the external force is represented by the mechanical action of stereociliar tip links on MCs, induced by the deflection of stereocilia. Particularly, when the hair bundle is in the resting position, the situation corresponds to the first described case when the external force is not applied. The gating spring bears some tension, hence the MC has a finite probability of being open. When the hair bundle is deflected in the excitatory direction, the spring stretches more and it pushes the opening probability of the MC to higher values. Deflecting the hair bundle in the inhibitory direction brings the compression of the spring and consequently the lowering of the MC opening probability. Following the reasoning of Markin and Hudspeth (1995) the dependence of the MC opening probability (p_o) on the hair bundle

displacement X is given by the dependence

$$p_o = \frac{1}{1 + \exp\left(-\frac{Z(X-X_0)}{kT}\right)}, \tag{3.16}$$

where Z is the gating sensitivity, X is the hair bundle's displacement from its resting position, X_0 is the midpoint of the opening transition, i.e. the point where the probability of the MC opening is 1/2, k is the Boltzmann constant and T is the temperature.

Mistrík et al. (2009) use the result (3.16) to develop the dependence of the MC conductivity g_{MC} on the BM displacement u in the form

$$g_{MC}(u) = g_{MC,max} \frac{1}{1 + \exp\left(-\frac{u-u_0}{d}\right)} \tag{3.17}$$

where $g_{MC,max}$ is the limit value of g_{MC} as $u \to \infty$, u_0 is the BM displacement at which MCs are half-activated and d is the width of the transition range; all the three variables are different for IHCs and OHCs. Mistrík et al. (2009) use the relation (3.17) it their model.

3.3 Connection of the electrical model to the hearing nerve

The previous section provided an overview of the model that simulates the K^+ recirculation in the cochlea. The output of this model is formed by voltages at different cochlear sites and by currents passing through the voltage sources. The next step in the modelling of the hearing process is the simulation of the connection between the electrical model and the model of the auditory nerve. Processes that are related to this connection occur at the level of IHCs since these are the receptor cells responsible for the transduction process. The input to this part of the model is the intracellular voltage of an IHC, denoted by $V(t)$. The output is expected to be the amount of the neurotransmitter that is released by the IHC into the synaptic cleft, i.e. into the space between the IHC and the nerve axon. The mathematical model that we employ of originates in the model presented by Sumner et al. (2002). The Sumner's model is extended by new results mainly by Eguia et al. (2010) and Zampini et al. (2014). The final model has multiple parts that are subsequently presented in the following overview, sticking to the notation of Sumner et al. (2002).

A. Depolarization of an IHC causes that the calcium channels open and that the Ca^{2+} flow into the IHC increases

Sumner et al. (2002) suggest the following relation for the calcium current I_{Ca} through the IHC membrane

$$I_{Ca}(t) = G_{Ca}^{max} m_{I_{Ca}}^3(t)(V(t) - E_{Ca}), \tag{3.18}$$

where G_{Ca}^{max} is the calcium conductance in the vicinity of the synapse when all the calcium channels are open, $m_{I_{Ca}}(t)$ is the fraction of the calcium channels that are open and E_{Ca} is equilibrium potential for Ca^{2+}. Recent researches performed by Goutman and Glowatzki (2007) show that the dependence of I_{Ca} on $m_{I_{Ca}}$ is linear, hence they suggest the expression of $I_{Ca}(t)$ in the form

$$I_{Ca}(t) = G_{Ca}^{max} m_{I_{Ca}}(t)(V(t) - E_{Ca}). \tag{3.19}$$

We opted finally for the relation (3.19) to be included into our model.

Further, $m_{I_{Ca}}(t)$ admits a steady state value $m_{I_{Ca},\infty}$ that is in Sumner et al. (2002) expressed as

$$m_{I_{Ca},\infty}(V) = [1 + \beta_{Ca}^{-1}\exp(\gamma_{Ca}V)]^{-1}, \tag{3.20}$$

where γ_{Ca} and β_{Ca} are constants chosen such that they fit the experimental data. An updated form of the relation (3.20) comes from Zampini et al. (2014) who state

$$m_{I_{Ca},\infty}(V) = \left[1 + exp\left(\frac{V_{1/2} - V}{S}\right)\right]^{-1}. \tag{3.21}$$

where $V_{1/2}$ and S are suitable constants. We use the expression (3.21) in our model.

Moreover, $m_{I_{Ca}}(t)$ can be expressed as a low-pass-filtered function of $m_{I_{Ca},\infty}$ hence it satisfies the equation

$$\tau_{I_{Ca}} \frac{\mathrm{d}m_{I_{Ca}}(t)}{\mathrm{d}t} + m_{I_{Ca}}(t) = m_{I_{Ca},\infty}, \tag{3.22}$$

where $\tau_{I_{Ca}}$ is a time constant.

B. The stimulated calcium ion flow into the IHC causes an increase in the calcium concentration in the vicinity of the synapse

The relation for the calcium concentration $[\mathrm{Ca}^{2+}]$ is

$$\tau_{[\mathrm{Ca}]} \frac{\mathrm{d}[\mathrm{Ca}^{2+}](t)}{\mathrm{d}t} + [\mathrm{Ca}^{2+}](t) = I_{Ca}(t), \tag{3.23}$$

with $\tau_{[Ca]}$ being a time constant and expressing that $[Ca^{2+}]$ is a low-pass-filtered function of $I_{Ca}(t)$.

C. The calcium concentration in the vicinity of the synapse affects the probability of the neurotransmitter release

The probability of the neurotransmitter release $k(t)$ by the IHC is in Sumner et al. (2002) given by the relation

$$k([\mathrm{Ca}^{2+}]) = \max(([\mathrm{Ca}^{2+}]^3 - [\mathrm{Ca}^{2+}]_{thr}^3)z, 0), \tag{3.24}$$

where $[\mathrm{Ca}^{2+}]_{thr}$ is the constant representing the threshold $[\mathrm{Ca}^{2+}]$ value and z is a scalar that converts the cube of the calcium concentration into the release rate. Another relation is proposed by Eguia et al. (2010) stating that

$$k([\mathrm{Ca}^{2+}]) = \frac{[\mathrm{Ca}^{2+}]k_{max}}{A + [\mathrm{Ca}^{2+}]}, \tag{3.25}$$

where k_{max} is the maximum release rate and A is a parameter. In our model we operate with the relation (3.24).

D. The released neurotransmitter into the synaptic cleft is lost or recycled

The basic idea of the neurotransmitter (NT) processing that we have adopted in our model has been proposed by Meddis (1986), see Figure 3.5. Meddis (1986) suggests that there is a free transmitter pool in the basal part of an IHC and that the NT leaks from the cell into the synaptic cleft with the release rate $k(t)$. Denoting the quantity of the free NT in the pool by $q(t)$, the amount of the released NT into the cleft in dt can be expressed as $k(t)q(t)dt$. Meddis (1986) suggests further that the cell is equipped with a NT factory that replenishes the pool with the NT at the rate $y(1-q(t))$. Here, 1 was set to be the maximal amount of NT present in the pool i.e. the maximal value that $q(t)$ can take. The amount of the NT present in the synaptic cleft is denoted by $c(t)$. From the cleft, $lc(t)dt$ of the NT is definitively lost and $rc(t)dt$ of the NT is retaken back to the cell. The retaken NT is stored in the reprocessing store from which it is released into the transmitter pool. The quantity of the NT in the reprocessing store is denoted by $w(t)$ and the release rate into the pool is x. Finally, the equations governing the time changes of the NT quantity in the transmitter pool $q(t)$, in the synaptic cleft $c(t)$ and in the reprocessing store $w(t)$ are formulated as follows

$$\frac{\mathrm{d}q}{\mathrm{d}t} = y(1-q(t)) + xw(t) - k(t)q(t), \tag{3.26a}$$

$$\frac{\mathrm{d}c}{\mathrm{d}t} = k(t)q(t) - lc(t) - rc(t), \tag{3.26b}$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = rc(t) - xw(t), \tag{3.26c}$$

where individual terms with the appropriate sign determine the inflow/outflow of the NT to/from the appropriate site.

The model that we have adopted to implement into our work is presented by Sumner et al. (2002) accounts for two modifications in the above presented ideas. The first one is that the NT is present in the transmitter pool in the form of quanta called NT vesicles. The transmitter pool will be newly called the immediate store, in order to not to be confused with a pool of freely dispersed NT. The release of the NT from the immediate store into the cleft is therefore realised by finite quanta. In the cleft, the vesicles are opened and the NT is dispersed. Some NT is returned back into the IHC and is repackaged into vesicles in the reprocessing store. NT is sent to the immediate store already in the form of vesicles. The immediate store has the capacity of maximum M NT vesicles and is refilled by the vesicles coming from the factory. The second modification is the stochastic aspect imported to the model. This new approach assumes that the NT enters and leaves the immediate store stochastically. Sumner et al. (2002) hence introduce the function representing the stochastic transport, $N(n,\rho)$, in

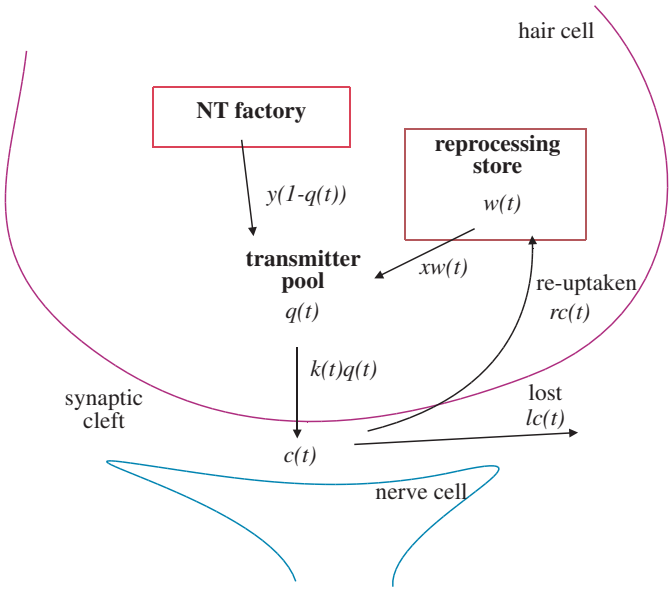


Figure 3.5: **Schema of the model of the neurotransmitter (NT) processing by Meddis (1986).**

which each of the n quanta has an equal probability of release, ρdt, in a single simulation interval. Applying the stochastic approach and assuming the quantal character of NT in the immediate store, the governing equations take the form

$$\frac{\mathrm{dq}}{\mathrm{dt}} = N((M - q(t)), y) + N(w(t), x) - N(q(t), k(t)), \tag{3.27a}$$

$$\frac{\mathrm{dc}}{\mathrm{dt}} = N(q(t), k(t)) - lc(t) - rc(t), \tag{3.27b}$$

$$\frac{\mathrm{dw}}{\mathrm{dt}} = rc(t) - N(w(t), x). \tag{3.27c}$$

3.4 Model of the hearing nerve

The previous section provided a model of the neurotransmitter release by an IHC into the synaptic space. The neurotransmitter in the synaptic space plays the primary role in the triggering of sudden depolarizing and hyperpolarizing actions of auditory nerve cells, forming *action potentials*. These are communicated to the central nervous system which retransforms them to the initial input information. In this section we will provide the mathematical model of the responses of the hearing nerve to external stimuli. Our model is in its major part formed by the deterministic Hodgkin-Huxley model introduced by A. L. Hodgkin and A. F. Huxley in 1950s (see Hodgkin and Huxley (1952b)) and is extended by various stochastic approaches.

3.4.1 The Hodgkin-Huxley model

Hodgkin and Huxley (1952b) introduced the model of the cell membrane as represented in Figure 3.6. Following Figure 3.6, the current I through the cell membrane can be processed in different ways. Either it may charge the capacitor C_M that is ascribed

to the cell membrane (the capacity current is denoted by I_C) or it may be carried by movement of ions through the resistors that are branched in parallel with the capacitors. Hodgkin and Huxley assume three types of ionic current: the currents I_{Na} and I_K carried respectively by sodium and potassium ions and a small 'leakage current' I_l carried by chloride and other ions. In the scheme of the Figure 3.6 each of the currents passes by a separate branch equipped with a resistance and a battery. The resistances are related to specific ion channels embedded in the cell membrane. While the sodium and potassium resistances R_{Na} and R_K vary according to the opened/closed channel state, the resistance of the leakage current R_L is assumed to be constant. The battery potentials E_{Na}, E_K and E_l in the circuit represent the equilibrium potential specific for sodium, potassium and chloride and other ions. We denote by U the membrane potential. Furthermore, for practical uses, Hodgkin and Huxley (1952b) introduce the potentials V, V_{Na}, V_K and V_l as:

$$V = U - E_r, \tag{3.28a}$$

$$V_{Na} = E_{Na} - E_r, \tag{3.28b}$$

$$V_K = E_K - E_r, \tag{3.28c}$$

$$V_l = E_l - E_r, \tag{3.28d}$$

where E_r is the absolute value of the cell resting potential. Variables V, V_{Na}, V_K and V_l designate therefore the displacement from the resting potential.

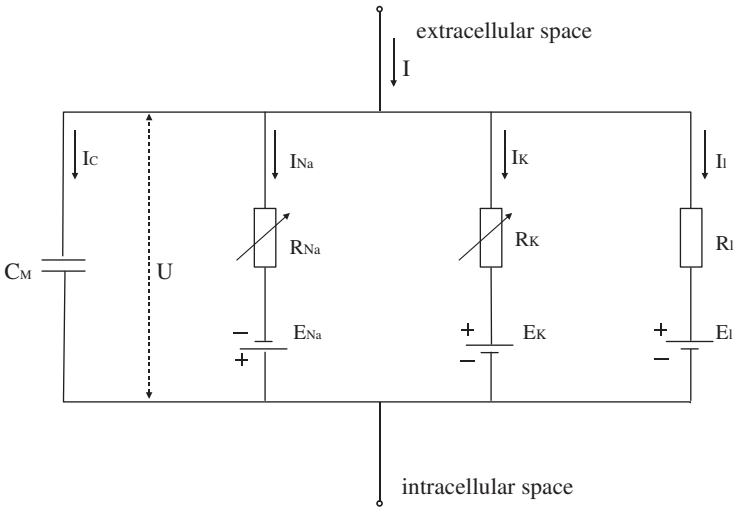


Figure 3.6: **Electrical circuit representing cellular membrane as seen by Hodgkin and Huxley (1952b).** The most left branch represents the charging/discharging of the membrane capacitance, other branches represent the passage of various ions through the membrane. Character l designates chloride and other ions. E_x represent equilibrium potentials of the membrane for the ion x. The resistances R_{Na} and R_K are variable. Figure is the reproduction of Figure 1 from Hodgkin and Huxley (1952b).

Following the scheme of the membrane electric circuit in Figure 3.6 we can write

the equation relating the above introduced currents as

$$I = I_C + I_{Na} + I_K + I_l. \tag{3.29}$$

Expressing the capacitive current as $I_C = C_M \mathrm{d}V/\mathrm{d}t$, introducing the ion channel conductance as $g_i = 1/R_i$ with i meaning the three current types, and reorganizing the terms we can rewrite the equation (3.29) as

$$C_M \frac{\mathrm{d}V}{\mathrm{d}t} = -g_{Na}(V - V_{Na}) - g_K(V - V_K) - \overline{g}_l(V - V_l) + I, \tag{3.30}$$

where the overbar of the g_l designates that this variable is considered as being constant.

Experiments performed by Hodgkin and Huxley (1952a) showed that the sodium and potassium conductances are functions of the intracellular voltage V and of time. The Figure 3.7 presents the data from the experiment where they observed the change

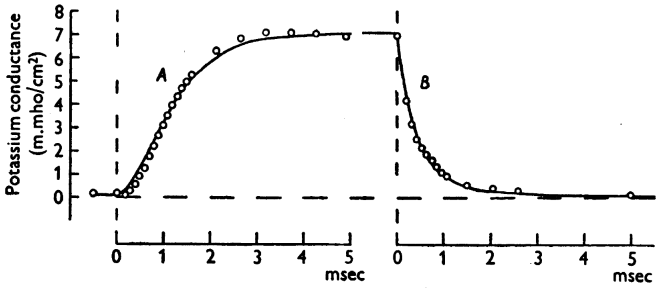


Figure 3.7: **Potassium conductance responding to a 25 mV depolarization and a subsequent repolarization to the resting potential as measured by Hodgkin and Huxley (1952a).** Circles are the experimental data; solid curves are the result of the fitting by a relation of the type $(A - B\exp(-Ct))^4$ with A, B and C constants. Figure reprinted from Hodgkin and Huxley (1952b, figure 2).

of the potassium conductance in time when depolarizing the cell by 25 mV and then subsequent repolarizing back to the cell resting potential. As is reasoned in Hodgkin and Huxley (1952b), to fit the data of the depolarization experiment for higher times, one needs a first-order equation since the data have the growing trend as $(1 - \exp(-t))$. However to fit the data at the beginning of the record, one needs a third- or fourth-order equation because the curve grows first very shallowly and then meets an inflection point. For the data of the repolarization the situation is simpler in that the conductance falls exponentially. Hodgkin and Huxley propose a solution that simplifies the problem of the data fitting: they suppose that g_K is proportional to the fourth power of a variable denoted by n which obeys a first-order equation. The conductance then growths as $(1 - \exp(-t))^4$ in the depolarization case and falls as $\exp(-4t)$ in the repolarization case. The reasoning for the dependence of the sodium conductance on time is made similarly however it applies to a different type of the data curve. In fact, in the depolarization experiment the sodium conductance first growths very similarly to the case of the potassium conductance but then it reaches a maximum and decreases to the initial value. To capture this type of the dependence, Hodgkin and Huxley introduce two

variables m and h obeying the first-order equation and they multiply the third power of m by the first power of h. While the rise of the sodium conductance in the beginning is well captured by the third power of m, the fall of the conductance for longer times is well described by h. Hodgkin and Huxley (1952b) propose finally the following set of equations

$$g_K = \overline{g}_K n^4, \tag{3.31a}$$

$$g_N a = \overline{g}_{Na} m^3 h, \tag{3.31b}$$

where n, m and h take values between 0 and 1 and obey the equations

$$\frac{\mathrm{d}n}{\mathrm{d}t} = \alpha_n(1-n) - \beta_n n, \tag{3.32a}$$

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \alpha_m(1-m) - \beta_m m, \tag{3.32b}$$

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \alpha_h(1-h) - \beta_h h, \tag{3.32c}$$

where α_i and β_i are functions of V. The variables n and m have in the presented form the meaning of respectively the potassium and the sodium current activation while the variable h represents the sodium current inactivation. Each of the equations (3.32a), (3.32b) and (3.32c) can be further reformulated in the form

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{x_\infty - x}{\tau_x} \tag{3.33}$$

with x representing the variables m, n or h and with

$$x_\infty = \frac{\alpha_x}{\alpha_x + \beta_x} \tag{3.34}$$

and

$$\tau_x = \frac{1}{\alpha_x + \beta_x}, \tag{3.35}$$

where x_∞ and τ_x are again functions of V. In this formulation ant at the fixed value of V, the x_∞ represents the value that x tends to in time with the time constant τ_x. The voltage dependences of α_i and β_i or alternatively of x_∞ and τ_x have been found by fitting experimental data and can be found in Hodgkin and Huxley (1952b). Summarizing, Hodgkin and Huxley (1952b) proposed the system of equations (3.30), (3.31), (3.32), and the dependences of α_i and β_i on V. This system involves the parameter I and can be solved for V. Depending on the input current I, the system either exhibits or does not exhibit an action potential, see Figure 3.8. Action potential represents a sudden depolarization, a subsequent repolarization up to values lower than the resting potential and a final slow rise up to the resting potential. The final phase when the potential is below the resting potential is called the *refractory period*. During this period, other action potentials cannot be activated. There exists a threshold value of the input current for the action potential to be initiated. If I has a lower value, the potential first rises a little and it is very quickly returned back to the resting value,

without exhibiting the behaviour of an action potential. The situation is illustrated in Figure.

Several simplifications of the system of equations of Hodgkin and Huxley have

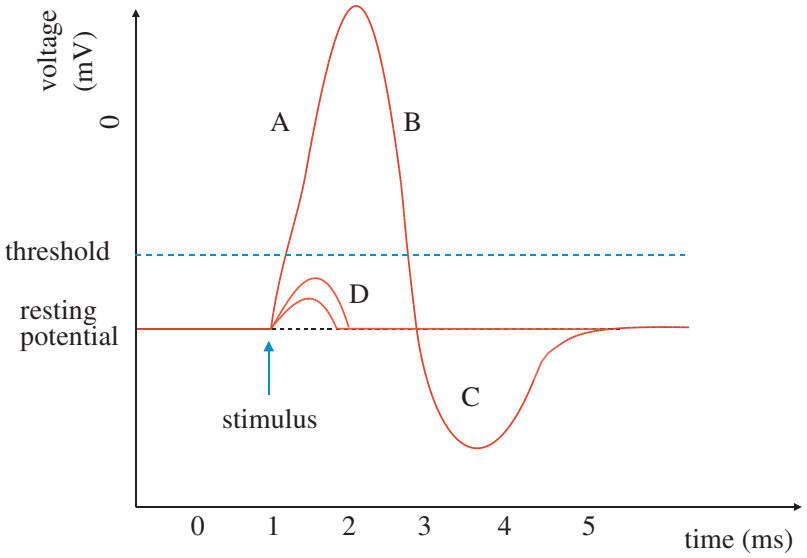


Figure 3.8: **Action potential.** A - depolarization, B - repolarization, C - refractory period, D - voltage variations stimulated by subthreshold stimuli.

been introduced to enable us to see how action potentials can represent solutions of the system. One of the simplified models is the FitzHugh-Nagumo model that works with two variables (v and w) instead of the four (V, n, m and h) and the solutions can therefore be easily seen in the phase portrait. For more information, see for example Georgescu et al. (2012).

3.4.2 The stochastic approach and implementation

To model the excitation of action potentials, Hodgkin and Huxley (1952b) used the assumption that ion channels embedded in the cell membrane behave following deterministic principles. Later, it was shown that this assumption is not justified. It was suggested that the model has to be furnished by a stochastic feature in order to describe the phenomena like random excitation of action potentials at subthreshold injected currents (see Strassberg and DeFelice (1993)). It was shown that ion channels are probabilistic devices and that they fluctuate randomly between two discrete states, the opened and the closed one. To introduce the stochastic approach into the deterministic model, various methods have been developed. The common feature of these methods is that the stochastic nature of ion channels is described by Markov processes. The methods differ in the particular introduction of these processes into the model. While Strassberg and DeFelice (1993), Rubinstein (1995) and Chow and White (1996) use exact methods, Fox (1997) opt for an approximation by stochastic differential equations. The four above cited models are compared in Mino et al. (2002).

In our model, we use the implementation of the stochastic approach into the computer model made by Bruce (2007). Bruce (2007) implemented both the model of Chow and White and the model of Fox. We currently make use predominantly of the more accurate but slower model of Chow and White.

4. Selected parts from numerical mathematics

This section presents selected chapters from numerical mathematics that we use in this work. First, we present several methods to solve numerically ordinary differential equations and then we describe numerical methods to solve the system of linear equations.

4.1 Numerical methods for solving Ordinary Differential Equations (ODEs)

This section follows primarily the chapter 5 from Dolejší (2015) and reproduces some ideas from the chapter 3 of Dolejší (2016).

4.1.1 Numerical methods: general approach

We are looking for the function $y : [a,b] \to \mathbb{R}^m$, $m \in \mathbb{N}$ that satisfies

$$y' = f(x,y) \tag{4.1a}$$

with the initial condition

$$y(a) = \eta, \tag{4.1b}$$

and with $f = (f_1,...,f_m) : [a,b] \times \mathbb{R}^m \to \mathbb{R}^m$ and $\eta \in \mathbb{R}^m$ given; f is supposed to be such that there exists a unique global solution $y : [a,b] \to \mathbb{R}^m$.

We aim to solve the problem (4.1) numerically. We first define a partition of the interval $[a,b]$. For simplicity we choose the equidistant partition i.e. we divide the interval $[a,b]$ to N intervals of the length $h_x = (b-a)/N$ by the nodes $\{x_i\}_{i=0}^N$ defined as $x_i = a + ih_x, i = 0,...,N$. We try to find the approximate values of the solution at nodes $\{x_i\}_{i=0}^N$, denoted by y_i, $y_i \in \mathbb{R}^m, i = 0,...,N$. The approximate values $\{y_i\}_{i=0}^N$ are generally given by an *l-step* numerical method:

$$y_{k+1} = F_k(x_{k+1},x_k,x_{k-1},...,x_{k-l+1}; y_{k+1},y_k,y_{k-1},...,y_{k-l+1}), \quad k = l-1,..,N-1, \tag{4.2}$$

where F_k are m appropriately chosen functions to fit the problem (4.1). If F_k do not depend explicitly on y_{k+1}, the numerical method defined by (4.2) is referred to as *explicit*, otherwise it is characterized as *implicit*.

We consider two types of error that are related to the adopted numerical method.

- **global error** G_k evaluated in x_k is defined as the difference between the approximate value y_k obtained by the numerical method (4.2) and the exact solution y of the problem (4.1) evaluated in x_k. Particularly,

$$G_k = y_k - y(x_k). \tag{4.3}$$

- **local error** L_k evaluated in x_k is defined as the difference between the approximate value y_k obtained by the numerical method (4.2) and the exact solution u_{k-1} of (4.1a), satisfying the condition $u_{k-1}(x_{k-1}) = y_{k-1}$ instead of the initial condition (4.1b), evaluated in x_k, namely

$$L_k = y_k - u_{k-1}(x_k). \tag{4.4}$$

Let us mention several properties of numerical methods of the type (4.2):

- we say that the numerical method (4.2) has **the order** p if

$$L_k = O(h_x^{p+1}). \tag{4.5}$$

  The order of numerical method gives us an information about the rate of convergence of the solution to the exact solution when the step h_x diminishes. Particularly, the approximate solution of a method of order p converges to the exact solution as h_x^p when $h_x \to 0$.

- we say that the numerical method (4.2) is **stable** if the global error G_k propagates with attenuation in the following steps, i.e. its contribution to the global error at nodes $x_i, i = k+1,...,N$ diminishes. If a numerical method is unstable, the opposite happens, ie the global error propagates with an enforcing tendency. One of the direct effects of the stability is that rounding errors occuring in finite precision arithmetic computations are attenuated and do not disrupt the solution. Oppositely, an unstable method induces the amplification of rounding errors which destroys the numerical solution. Generally, there exist many notions of stability in the numerical mathematics. The one that fits the best to our definition is the *A-stability.*

4.1.2 Explicit methods

In the case of explicit methods, F_k in the expression (4.2) is not a function of y_{k+1}. The approximate solution at the node x_{k+1} can therefore by directly calculated using the approximate solutions at nodes $x_{k-l+1},...,x_k$. Explicit methods are therefore generally very practical to use due to their speed. An inconvenient property of most explicit methods is the conditional stability. The conditional stability means that the method is stable for certain small values of the step h_x and unstable for the complementary values. Here we present a class of explicit one-step conditionally stable methods.

Explicit Runge-Kutta methods

An *s-stage* explicit Runge-Kutta method is given by the formula

$$y_{k+1} = y_k + h_x \sum_{i=1}^{s} b_i q_i, \tag{4.6}$$

where

$$q_1 = f(x_k, y_k), \tag{4.7a}$$

$$q_2 = f(x_k + c_2 h_x, y_k + h_x(a_{21} q_1)), \tag{4.7b}$$

$$q_3 = f(x_k + c_3 h_x, y_k + h_x(a_{31} q_1 + a_{32} q_2)), \tag{4.7c}$$

$$\vdots \tag{4.7d}$$

$$q_s = f(x_k + c_s h_x, y_k + h_x(a_{s1} q_1 + a_{s2} q_2 + \cdots + a_{s,s-1} q_{s-1})). \tag{4.7e}$$

The coefficients c_i, a_{ij} and b_i are grouped in the so-called *Butcher's tableau*:

$$\begin{array}{c|cccc} 0 & & & & \\ c_2 & a_{21} & & & \\ \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & \dots & a_{s,s-1} & \\ \hline & b_1 & \dots & b_{s-1} & b_s \end{array}$$

The relation between the number of stages s of a Runge-Kutta method having the order p and the order p is expressed as follows: in general, it holds $s \geq p$ and if $p \geq 5$, then $s > p$.

In our work, we will use particularly two explicit Runge-Kutta methods:

explicit Euler method

$\mathbf{s = 1}$, $p = 1$ and it is conditionally stable

$$y_{k+1} = y_k + h_x f(x_k, y_k), \tag{4.8}$$

6-stage Runge-Kutta method

$\mathbf{s = 6}$, $p = 5$, it is conditionally stable and is characterized by the Butcher's tableau 4.1.

0						
1/5	1/5					
3/10	3/40	9/40				
4/5	44/45	−56/15	32/9			
8/9	19372/6561	−25360/2187	64448/6561	−212/729		
1	9017/3168	−355/33	46732/5247	49/176	−5103/18656	
	35/384	0	500/1113	125/192	−2187/6784	11/84

Table 4.1: **Butcher's tableau for 6-stage Runge-Kutta method**

4.1.3 Implicit methods

Implicit methods are characterized by explicit dependence of F_k from the expression (4.2) on y_{k+1}. This property brings difficulties to the computation of y_{k+1} since it cannot be computed directly as it was in the case of explicit methods. The expression (4.2)

leads in the case of implicit methods to a system of linear equations (if F_k is linear with respect to y_{k+1}) or nonlinear equations (if F_k is nonlinear with respect to y_{k+1}) with the unknown y_{k+1}. To solve the system, we use appropriate techniques known from numerical linear algebra. The computation of y_{k+1} is therefore generally slower than in the case of explicit method. On the other hand, their important advantage is the unconditional stability. This means that we can choose an arbitrarily large step h_x to reach the stability of the method. Here we present several one-, two- or three- step implicit methods that are unconditionally stable.

implicit Euler method

- method of the 1st order

$$y_{k+1} = y_k + h_x f(x_{k+1}, y_{k+1}), \tag{4.9}$$

Crank-Nicolson method

- method of the 2nd order, it combines the implicit and explicit Euler methods

$$y_{k+1} = y_k + \frac{h_x}{2}\left[f(x_k, y_k) + f(x_{k+1}, y_{k+1})\right], \tag{4.10}$$

Backward Difference Formula of the 2^{nd} order

$$\frac{3}{2}y_{k+1} - 2y_k + \frac{1}{2}y_{k-1} = h_x f(x_{k+1}, y_{k+1}), \tag{4.11}$$

Backward Difference Formula of the 3^{rd} order

$$\frac{11}{6}y_{k+1} - 3y_k + \frac{3}{2}y_{k-1} - \frac{1}{3}y_{k-2} = h_x f(x_{k+1}, y_{k+1}). \tag{4.12}$$

4.1.4 Approaches to solve nonlinear ODEs with implicit numerical methods

As we stated in the section 4.1.3, an implicit method applied to the original problem (4.1) leads to a system of linear or nonlinear equations that we need to solve at each step. The ways how to solve a system of linear equations are multiple and will be presented in the part 4.2. For the moment we assume that we can handle the linear problem satisfactorily and we focus on solving systems of nonlinear equations. In particular, we consider the following nonlinear form of the problem (4.1a):

$$y' = \mathbf{J}y + g(x) + h(y), \tag{4.13}$$

where $\mathbf{J}$ is a constant matrix $m \times m$, g is an arbitrary function of x: $g = (g_1,...,g_m) : [a,b] \to \mathbb{R}^m$ and h is a nonlinear function of y: $h = (h_1,...,h_m) : \mathbb{R}^m \to \mathbb{R}^m$.

Application of any of the four implicit methods considered in the section 4.1.3 to the problem (4.13) leads to

$$\mathbf{A}y_{k+1} - ch(y_{k+1}) = b_k, \tag{4.14}$$

where A is a matrix $m \times m$, b_k is a vector of the length m and c is a scalar constant that involves the step h_x. We present two approaches of how to attack the nonlinear term $ch(y_{k+1})$ in (4.14).

Explicitly treated h

One way how to deal with the nonlinear problem (4.14) is to treat the nonlinearity explicitly. This means that we evaluate the function h at a specific point and we include it to the right hand side. The newly formed linear problem is then easily solvable. This method belongs to the group of the so-called *Implicit-Explicit Methods (IMEX)*, see Ascher et al. (1997) for more information and other examples.

We proceed by iterations at each step and we evaluate the nonlinear function h at the point that forms a suitable combination of solutions from previous iterations. In this way we transform the nonlinear problem (4.14) at each step k to an iterating sequence of linear problems in the form

$$\mathbf{A}y_{k+1}^i = d_k^i \tag{4.15}$$

where y_{k+1}^i is the solution in the iteration i of the step k and d_k^i is the right hand side that may take various forms; we consider two of them:

$$d_k^i = b_k + ch(y_{k+1}^{i-1}), \quad i = 1,2,... \tag{4.16}$$

and

$$d_k^i = b_k + ch(2y_{k+1}^{i-1} - y_{k+1}^{i-2}), \quad i = 2,3,... \quad . \tag{4.17}$$

The initial value y_{k+1}^0 for the calculation of d_k^0 in (4.16) and additionally the value y_{k+1}^1 for the calculation of d_k^1 in (4.17) are chosen arbitrarily. The solution y_{k+1} of the nonlinear problem (4.14) at the step k is given by the value to which the approximate solutions y_{k+1}^i converge:

$$y_{k+1} = \lim_{i \to \infty} y_{k+1}^i. \tag{4.18}$$

In practical use, we solve the sequence of problems (4.15) at each step k until the moment given by a stopping criterion.

Newton method - linearization

The obstacle in the form of the nonlinearity in (4.14) can be surmounted by linearization of the nonlinear function h.

At each step k, we proceed by iterations and at each iteration i we look for the approximate solution y_{k+1}^i. At each iteration we approximate the nonlinear term $h(y_{k+1}^i)$

by its linear expansion developed in y_{k+1}^{i-1} and evaluated in y_{k+1}^{i}:

$$h(y_{k+1}^{i}) \approx h(y_{k+1}^{i-1}) + \frac{\mathbf{Dh}}{\mathbf{Dy}}(y_{k+1}^{i-1})\left(y_{k+1}^{i} - y_{k+1}^{i-1}\right), \quad i = 1,2,... \tag{4.19}$$

Hence at each iteration we have the linear problem

$$\mathbf{B_k^i} y_{k+1}^{i} = d_k^i \tag{4.20}$$

that we solve for y_{k+1}^{i}, with

$$\mathbf{B_k^i} = \mathbf{A} - c\frac{\mathbf{Dh}}{\mathbf{Dy}}(y_{k+1}^{i-1}) \tag{4.21}$$

and

$$d_k^i = b_k + ch(y_{k+1}^{i-1}) - c\frac{\mathbf{Dh}}{\mathbf{Dy}}(y_{k+1}^{i-1})y_{k+1}^{i-1}. \tag{4.22}$$

The principal disadvantage of this method in practical applications is that we are obliged to recalculate the matrix $\mathbf{B_k^i}$ at each iteration. In some cases this procedure may be rather time-consuming or difficult. Often in these situations, a more effective method turns out to be the so-called *fixed Newton method*. The fixed Newton method is also based on the linearization but uses the constant slope of the linear function in all iterations, namely

$$h(y_{k+1}^{i}) \approx h(y_{k+1}^{i-1}) + \frac{\mathbf{Dh}}{\mathbf{Dy}}(y_{k+1}^{0})\left(y_{k+1}^{i} - y_{k+1}^{i-1}\right), \quad i = 1,2,.... \tag{4.23}$$

The nonlinear problem therefore reduces to the solution of

$$\mathbf{B_k} y_{k+1}^{i} = d_k^i \tag{4.24}$$

for y_{k+1}^{i} with

$$\mathbf{B_k} = \mathbf{A} - c\frac{\mathbf{Dh}}{\mathbf{Dy}}(y_{k+1}^{0}) \tag{4.25}$$

and

$$d_k^i = b_k + ch(y_{k+1}^{i-1}) - c\frac{\mathbf{Dh}}{\mathbf{Dy}}(y_{k+1}^{0})y_{k+1}^{i-1}. \tag{4.26}$$

The sought solution y_{k+1} of the problem (4.14) is then given by

$$y_{k+1} = \lim_{i\to\infty} y_{k+1}^{i}. \tag{4.27}$$

Similarly to the previous case, in practical uses we define a stopping criterion for the iteration process to get the solution.

The advantage of the fixed Newton method is that we do not have to calculate the matrix $\mathbf{B_k}$ at each iteration but only once in a step. The inconvenient is that since it does not take into account the shape of the nonlinearity at each iteration, it may converge more slowly than the classical alternative.

4.2 Numerical methods for solving systems of linear equations

Let's consider the system of N linear equations for N unknowns written in the form

$$\mathbf{A}x = b, \tag{4.28}$$

where $\mathbf{A}$ is a regular matrix $N \times N$ and b is a vector of the length N. To solve the system (4.28) numerically, we can use various direct or iterative numerical methods. Direct methods are advantageous in finding the solution in only one step, however this step may be rather time or capacity demanding. Iterative methods find the solution in multiple steps, however these each of them may be very fast since it involves only multiplication of a vector by matrix. If, in particular, the matrix is sparse, this option may be rather time and space saving. It therefore depends on the particular problem whether we opt for a direct or an iterative method. Here we present particularly one of each. The information about both methods are taken from the book Tebbens et al. (2012); the direct method follows Chapter 4 and the iterative method follows Chapter 9.

4.2.1 Direct method: method using LU decomposition of the matrix A

If we decompose the matrix A in such a way that

$$\mathbf{A} = \mathbf{L}\mathbf{U}, \tag{4.29}$$

where $\mathbf{L}$ is the lower triangular matrix and $\mathbf{U}$ is the upper triangular matrix, then the problem (4.28) can be rewritten in the form

$$\mathbf{U}x = y, \tag{4.30a}$$

$$\mathbf{L}y = b \tag{4.30b}$$

and can be easily solved by the forward substitution of (4.30b) getting y and by the subsequent backward substitution of (4.30a) getting x. The two substitutions are generally performed very quickly and especially MATLAB performs it very fast by the operator of a backward slash. The only expensive operation is the computation of the matrix decomposition.

The decomposition of a regular square matrix A in the form (4.29) with an additional property that the diagonal of $\mathbf{L}$ is unit is called the *LU decomposition* of the matrix $\mathbf{A}$. The algorithm to calculate the LU decomposition of a matrix is given by the Gauss elimination process. In fact, the Gauss elimination transforms a given matrix $\mathbf{A}$ to the upper triangular matrix $\mathbf{U}$ by successive multiplications by lower triangular matrices from the left. Each of the lower triangular matrices have a unit diagonal and one non-zero subdiagonal column that is formed by a certain combination of the

elements of **A**. These matrices have such form that their multiplication results in a lower triangular matrix that is exactly the matrix **L** from the decomposition (4.29). The elements of **L** and **U** can therefore be calculated from elements of the matrix **A** in a way that is given by the Gauss elimination process.

According to the presented scheme, the LU decomposition seems to be a practical tool for solving linear systems of the form (4.28). Further analyses show however that this method may not be stable in some cases that means that it may lead to unjustifiably large values destroying the computation. To improve the stability of the method, an additional operation, the pivoting, is included into the computation. The pivoting in the Gaussian elimination means that before each elimination of subdiagonal elements in a column, we rearrange the rows of the matrix (partial pivoting) or the rows and the columns of the matrix (full pivoting) in order to get the maximal element in the diagonal position. This ensures that the ratio of an arbitrary element of the column subdiagonal to the diagonal element is less than 1. Since these ratios form the elements of the matrix **L**, each element of the matrix **L** is then less then 1. In Chapter 4, Tebbens et al. (2012) use this result to introduce the *growth factor* that determines the backward stability of the method. The LU decomposition method with pivoting is therefore conditionally stable.

Let's now focus on the partial pivoting and let's see what differences brigs the partial pivoting element to the expression (4.29). The partial pivoting introduces a permutation matrix **P** into (4.29) such that

$$\mathbf{PA} = \mathbf{LU} \tag{4.31}$$

hence the LU decomposition of the matrix **A** with pivoting is in fact the LU decomposition of the matrix **PA** without pivoting. The system (4.30) is therefore rewritten in the form

$$\mathbf{U}x = y, \tag{4.32a}$$

$$\mathbf{L}y = \mathbf{P}b \tag{4.32b}$$

Matlab implementation

To perform the LU decomposition of a matrix, MATLAB uses the `lu` built-in function, see [4]. Here we present several input and output parameters that `lu` accepts in its algorithm.

```
[L,U,P] = lu(A)
```

`L, U, P, A` — `L`, `U` and `P` designate respectively the lower triangular, the upper triangular and the permutation matrices as the result of LU factorization of the matrix `A` with pivoting; the relation between the matrices is `LU=PA`

4.2.2 Iterative method: Generalized Minimal Residual Method (GMRES)

GMRES is an iterative method for solving (4.28) that at the k-th iteration searches an approximation of the solution named x_k while the initial approximation x_0 is given. The x_k is searched in the space $x_0 + K_k(\mathbf{A},r_0)$ where $K_k(\mathbf{A},r_0)$ is k-th Krylov subspace defined as

$$K_k(\mathbf{A},r_0) = \text{span}\{r_0,\mathbf{A}r_0,\mathbf{A}^2r_0,...,\mathbf{A}^{k-1}r_0\} \tag{4.33}$$

with $r_0 = b - \mathbf{A}x_0$ being the residual of the initial approximation x_0. The approximate solution $x_k \in x_0 + K_k(\mathbf{A},r_0)$ is determined at each iteration by the condition of obtaining minimal norm of the residual r_k, $r_k = b - \mathbf{A}x_k$. As was derived in Chapter 9 of Tebbens et al. (2012), this requirement is equivalent to the condition that the residual is orthogonal to the space $\mathbf{A}K_k(\mathbf{A},r_0)$. Formally the method can be summarized as

$$x_k \in x_0 + K_k(\mathbf{A},r_0), \qquad r_k \perp \mathbf{A}K_k(\mathbf{A},r_0). \tag{4.34}$$

As well as other methods based on Krylov subspaces, GMRES finds the solution in a finite number of iterations (thought in the precise arithmetic), particularly in N iterations if $A \in \mathbb{C}^{N\times N}$. However in practical uses, we generally do not need the exact solution of the problem (4.28) and we look only for the approximation of this solution. We get the approximation of the solution by stopping the GMRES algorithm in a specified moment and by taking the lastly calculated x_k. The stopping criterion is generally given by the condition that the relative residual $\|b - \mathbf{A}x_k\| / \|b\|$ is lower than an initially given tolerance value. The fact that we need to stop the iteration process to obtain the desired solution implies a natural requirement on the high convergence rate of approximate solutions x_k. At high rate of convergence of the method we only need a small number of iterations to satisfy the stopping criterion, thus making the whole GMRES computation fast. In our work we will use the *preconditioning* technique to improve the convergence properties of GMRES. Some GMRES general convergence results can be found in Saad (2003, section 6.11.4).

Preconditioning of GMRES

In our computations we will make use of a generally used and efficient method to improve the convergence rate called *preconditioning*. The preconditioning of the system (4.28) consists in its modification to the form

$$\left(\mathbf{P_L}^{-1}\mathbf{A}\mathbf{P_R}^{-1}\right)(\mathbf{P_R}x) = \mathbf{P_L}^{-1}b, \tag{4.35}$$

where $\mathbf{P_L}$ and $\mathbf{P_R}$ are suitably chosen matrices. The algorithm of GMRES is applied to the newly defined system (4.35) and the solution is recalculated back using the matrix $\mathbf{P_R}$. The crucial task for users is to choose the matrices $\mathbf{P_L}$ and $\mathbf{P_R}$ so that

the convergence properties of (4.35) be better than that of (4.28). Following Tebbens et al. (2012), the generally used rule to reach this goal is to choose the matrices so that

$$\mathbf{P_L}^{-1}\mathbf{A}\mathbf{P_R}^{-1} \approx \mathbf{I}. \tag{4.36}$$

If $\mathbf{P_L} = \mathbf{I}$ and $\mathbf{P_R} \neq \mathbf{I}$ we say that it is the preconditioning from the right, if $\mathbf{P_R} = \mathbf{I}$ and $\mathbf{P_L} \neq \mathbf{I}$, it is the preconditioning from the left and if $\mathbf{P_L} \neq \mathbf{I}$ and $\mathbf{P_L} \neq \mathbf{I}$, we talk about the split preconditioning. Following (4.36), we require $\mathbf{P_R} \approx \mathbf{A}$ in the preconditioning from the right, $\mathbf{P_L} \approx \mathbf{A}$ in the preconditioning from the left or

$$\mathbf{P_L}\mathbf{P_R} \approx \mathbf{A} \tag{4.37}$$

in the split preconditioning. Some results from the comparison of these three preconditioning types for GMRES are summarized in Saad (2003).

One of the most used ways to determine the matrices $\mathbf{P_L}$ and $\mathbf{P_R}$ satisfying (4.37) in the split preconditioning is the *incomplete LU factorization* of the matrix $\mathbf{A}$. The incomplete LU factorization of $\mathbf{A}$ consists in the decomposition of $\mathbf{A}$ in the form $\mathbf{A} = \mathbf{LU} - \mathbf{R}$ where $\mathbf{L}$ and $\mathbf{U}$ are the lower and upper triangular matrices respectively. The specificity of this decomposition is that $\mathbf{L}$ and $\mathbf{U}$ have zeros at initially predefined positions and $\mathbf{R}$ is the residual matrix of this decomposition. The basic type of incomplete lu factorization denoted by ILU(0) involves the same nonzero pattern of $\mathbf{L}$ and $\mathbf{U}$ as the nonzero patterns of the lower and upper parts of $\mathbf{A}$. A general algorithm to obtain this decomposition is the Gaussian elimination with dropping elements at the zero positions of $\mathbf{A}$. This algorithm is rather inexpensive and fast to perform but may lead only to a crude approximation that may not improve the GMRES convergence properties. To fix this problem several alternatives of ILU(0) have been developed. These are designed so that they allow more fill-in of $\mathbf{L}$ and $\mathbf{U}$ according to a specified criterion. One way is to specify the level of fill-in that leads to ILU(p) methods described in detail in Saad (2003, section 10.3.3). Another way involves the dropping of $\mathbf{L}$ and $\mathbf{U}$ elements depending on the relation of these elements with a specified tolerance value, leading for example to the ILUTP algorithm (Saad (2003, section 10.4.4)) or to the Crout ILU algorithm (Saad (2003, section 10.4.6)). These alternatives lead to an increase of accuracy of the decomposition however they have a greater computational cost. To reduce the effect of dropping elements in the `ilu` process we may opt for the *modified* strategy. This technique permits to compensate the discarted entries so that the row (or column) sums of $\mathbf{A}$ be equal of those of $\mathbf{LU}$, see Saad (2003, section 10.3.5) for more information. It depends on the particular situation what `ilu` version we will opt for.

Matlab implementation

MATLAB disposes of a built-in function `gmres` that executes the GMRES algorithm. The function accepts multiple input parameters from which several are specified in the

following overview. More on the `gmres` function can be found in [5]

`[x, flag, relres, iter, resvec] = gmres(A, b, restart, tol, maxit, M1, M2, x0)`

`x, A, b`	`x` is the desired approximation of the solution of the linear problem `Ax=b`
`restart`	denotes the number of GMRES iterations after which the algorithm, takes the lastly computed approximation x_{res} and uses it as the initial approximation to another subsequent GMRES algorithm; we say that `gmres` accepts `restart` of *inner* iterations
`tol`	specifies the tolerance value; the computation stops if $\|b - \mathbf{A}x_k\| / \|b\|$ <`tol`; the default value is 10^{-6}
`maxit`	denotes the maximum number of repeating the GMRES algorithm with `restart` inner iterations; the maximum total number of iteration is therefore `restart*maxit`; if `restart` is not specified, then GMRES algorithm is executed only once and `maxit` denotes the maximum number of iterations
`M1, M2`	designate the preconditioning matrices: `M1` represents $\mathbf{P_L}$ and `M2` represents $\mathbf{P_R}$ in (4.35)
`x0`	is the initial guess to the GMRES algorithm; the default is a zero vector
`flag`	gives information about the reliability of the result
`relres`	returns the relative residual norm of the final approximation
`iter`	returns a two-element vector: the first element designates the number of outer iterations and the second designates the number of inner iterations
`resvec`	returns a vector of relative residual norms (for the preconditioned system) at each inner iteration

To build the split preconditioning matrices `M1` and `M2` different from the identity matrix we use the built-in function `ilu` that performs the incomplete LU factorization. Here we present the `ilu` MATLAB function with the selected input and output parameters.

`[L,U,P] = ilu(A, setup)`

`L, U, P, A`	denote respectively the lower triangular, the upper triangular and the permutation matrices as the result of the incomplete factorization algorithm of the matrix `A`

`setup` specifies the type of the incomplete LU decomposition; it may contain up to five

setup options from which the selected ones are specified in the following overview

`type` may take various arguments

- 'nofill'(default) - performs ILU(0)
- 'ilutp' - performs incomplete LU decomposition with specified threshold value `droptol` and with pivoting;
- 'crout' - performs the crout version of the incomplete LU factorization with specified threshold value `droptol`

`droptol` specifies the dropping tolerance of the incomplete LU factorization; the nonzero extra-diagonal elements of $\mathbf{U}$ satisfy $|\mathbf{U}(i,j)| \geq$ `droptol`$\|\mathbf{A}(:,j)\|$ and the nonzero extra-diagonal elements of $\mathbf{L}$ satisfy $|\mathbf{L}(i,j)| \geq$ `droptol`$\|\mathbf{A}(:,j)\|/\mathbf{U}(j,j)$

`milu` may take various arguments

- 'row' - row-sum modified incomplete LU factorization
- 'col' - column-sum modified incomplete LU factorization

5. Mechanical part of the model: discretization and implementation

In the section 3 we provide a detailed description of how we model mechanical processes that arise in the cochlear duct in response to external stimuli. We explain that the crucial active parts in the mechanical processes are the displacement of the basilar membrane and the deflection of hair cell bundles. The description ends up by writing the equations (3.15) and (3.14) governing these motions. In this chapter we will work with these equations, particularly with the equation (3.15) in its original form assuming $F_{ohc}(y(x,t)) = U(x,t)$ and with the equation (3.14) in a modified form, namely:

Equation for the BM displacement $\boldsymbol{u}(x,t)$

$$m(x)\frac{\partial^2 u(x,t)}{\partial t^2} + h(x)\frac{\partial u(x,t)}{\partial t} - \left[\frac{\partial}{\partial x}s(x)\frac{\partial}{\partial x}\right]\frac{\partial u(x,t)}{\partial t} + \\ + k(x)u(x,t) + F_{ohc}(y(x,t)) = F_S(x,t) + F_{BM}(x,t), \quad (5.1)$$

Equation for the hair cell bundle displacement $\boldsymbol{y}(x,t)$

$$M(x)\frac{\partial^2 y(x,t)}{\partial t^2} + H(x)\frac{\partial y(x,t)}{\partial t} + K(x)y(x,t) = -C(x)\frac{\partial^2 u(x,t)}{\partial t^2} \quad (5.2)$$

with C a (for instant) arbitrary function of x and

$$F_S(x,t) = -G_S(x)\frac{\partial^2 \sigma(t)}{\partial t^2} \quad (5.3)$$

$$F_{BM}(x,t) = -\int_0^1 G(x,z)\frac{\partial^2 u(z,t)}{\partial t^2}dz, \quad (5.4)$$

and we add that we have: $x \in [0,1]$, $t \geq 0$, $\frac{\partial u(x,0)}{\partial t} = u(x,0) = 0$, $\frac{\partial y(x,0)}{\partial t} = y(x,0) = 0$, $\forall x \in [0,1]$ and that the boundary conditions will follow from the discretization of the term $\left[\frac{\partial}{\partial x}s(x)\frac{\partial}{\partial x}\right]\frac{\partial u(x,t)}{\partial t}$.

(5.1) and (5.2) are equations of the second order with respect to the time derivative. They can be rewritten in the form of a system of four equations of the first order:

$$\begin{aligned}
\frac{\partial u(x,t)}{\partial t} &= v(x,t) \\
m(x)\frac{\partial v(x,t)}{\partial t} &= F_S(x,t) + F_{BM}(x,t) - F_{ohc}(y(x,t)) - k(x)u(x,t) + \left[\frac{\partial}{\partial x}s(x)\frac{\partial}{\partial x} - h(x)\right]v(x,t) \\
\frac{\partial y(x,t)}{\partial t} &= w(x,t) \\
M(x)\frac{\partial w(x,t)}{\partial t} &= -C(x)\frac{\partial v(x,t)}{\partial t} - H(x)w(x,t) - K(x)y(x,t).
\end{aligned} \quad (5.5)$$

5.1 Discretization of the system of motion equations in space

In this section we will treat the problem of the spatial discretization of the system (5.5).

5.1.1 Discretization of the position coordinate x

We introduce N discrete nonequidistant points with the coordinate x_j covering the interval [0,1] and this is done in the following way:

```
MiddlePoint=0.35;           % point of change in [0,1]
sig1=2*(1.2)^2;             % coefficient of the 1st dependence
sig2=2*(0.46)^2;            % coefficient of the 2nd dependence

xo=(1:N)/N;         % initial equidistant division of the interval [0,1]
ind=ceil(MiddlePoint/xo(1))        % index corresponding to the MiddlePoint
dx(1:ind)=exp(((xo(1:ind)-xo(ind)).^2)./sig1);
dx(ind+1:N)=exp(((xo(ind+1:N)-xo(ind)).^2)./sig2);
        % distance between neigbor discrete points
dx=dx/sum(dx);                          % normalization so that sum(dx)=1

x(1)=dx(1);
for i=2:N,
        x(i)=x(i-1)+dx(i);        % allocation of new position coordinates
end
```

As can be easily understood from the MATLAB code, `MiddlePoint` determines a unique point index `ind` that, after the allocation of new position coordinates, determines the point x_{ind} with minimal distance to the neighbour points. With the change of the position coordinate from x_{ind-1} to 0 and from x_{ind+1} to 1, the relative distance dx between points is increased and this increase is more rapid in the second interval (see Figure 5.1). The point density is therefore maximal in the proximity of x_{ind} and is diminished more rapidly when passing from x_{ind+1} to 1 than from x_{ind-1} to 0.

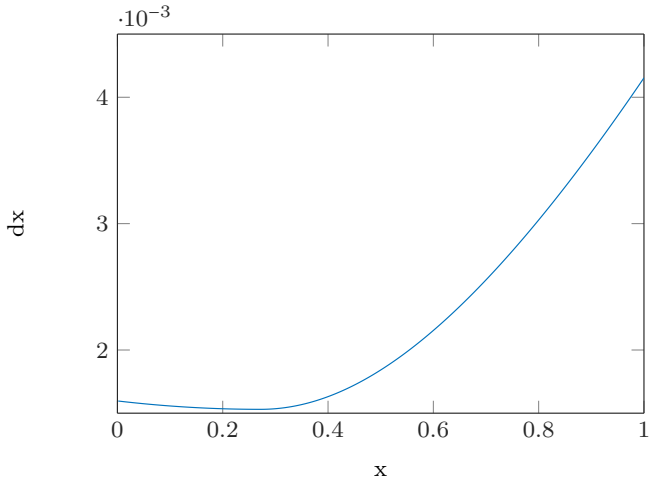


Figure 5.1: **Interpoint distances dx as a function of x for** $N = 1000$**.** Vector dx is defined as $dx_i = x_i - x_{i-1}$

A nonequidistant division of the interval [0,1] was introduced already in the first version of the program by Mammano and Nobili (see the related article Mammano and Nobili (1993)). This one has been characterized by the fixed position of the maximal and minimal point densities, namely maximal at x_1 and minimal at $x_N = 1$. In the region between x_1 and x_N the distances dx obeyed an exponential law. The improvement of the code to the up-to-date form was made by Vetešník and Gummer (2012). The improvement consists in the correction of the position of the maximal point density: it is shifted from 0 to the value given by `MiddlePoint`, i.e. approximately by 0,27 (after the new allocation of coordinates in the code). The reason why this form of the dependence was selected is the observed reaction of the BM to the input signals of different frequencies. While for higher frequencies, the BM oscillates primarily in the region close to the stapes (i.e. in the proximity of $x = 0$) and the activated region is narrow, for lower frequencies the BM is oscillates close to the cochlear apex with a larger activated region. The narrowest region that is activated is placed around 0,27.

5.1.2 Discretization of functions of the spatial coordinate f(x,...)

First, we introduce the vector $U(t)$ of the length N that approximates the BM displacement $u(x,t)$ at discrete positions x_j:

$$U(t) := \begin{pmatrix} U_1(t) \\ U_2(t) \\ \vdots \\ U_N(t) \end{pmatrix} \approx \begin{pmatrix} u(x_1,t) \\ u(x_2,t) \\ \vdots \\ u(x_N,t) \end{pmatrix}. \tag{5.6}$$

Analogically, we introduce vectors $V(t)$, $Y(t)$ and $W(t)$ of the BM velocity $v(x,t)$, of the hair cell bundle displacement $y(x,t)$ and of the hair cell bundle velocity $w(x,t)$

approximated at discrete positions x_j. The time derivative of a time dependent vector is taken by elements, that is for example for $U(t)$:

$$\frac{dU(t)}{dt} = \begin{pmatrix} \frac{dU_1(t)}{dt} \\ \frac{dU_2(t)}{dt} \\ \vdots \\ \frac{dU_N(t)}{dt} \end{pmatrix} \tag{5.7}$$

Then we introduce the N-long vector m of the functional values of $m(x)$ at discrete positions x_j:

$$m := \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_N \end{pmatrix} = \begin{pmatrix} m(x_1) \\ m(x_2) \\ \vdots \\ m(x_N) \end{pmatrix} \tag{5.8}$$

and we define the $N \times N$ diagonal matrix $\mathbf{m}$ with m on the diagonal:

$$\mathbf{m} := \mathrm{diag}\,(m) = \begin{pmatrix} m_1 & 0 & \dots & 0 \\ 0 & m_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & m_N \end{pmatrix}. \tag{5.9}$$

Analogically, we define N-long vectors k, s, h, G_S, M, H, K, C from $k(x)$, $s(x)$, $h(x)$, $G_\mathrm{S}(x)$, $M(x)$, $H(x)$, $K(x)$ and $C(x)$ and the corresponding diagonal matrices $\mathbf{k}$, $\mathbf{s}$, $\mathbf{h}$, $\mathbf{G_S}$, $\mathbf{M}$, $\mathbf{H}$, $\mathbf{K}$ and $\mathbf{C}$.

5.1.3 Discretization of $\mathbf{F_{BM}}$, $\left[\frac{\partial}{\partial x}\mathbf{s}(x)\frac{\partial}{\partial x}\right]$ and $\mathbf{F_{ohc}}$, further notations

If we define $N \times N$ Green's function matrix $\mathbf{G}$ as

$$\boldsymbol{G}_{ij} = G(x_i, z_j),$$

for $i,j \in \{1,...,N\}$, we can write F_{BM} in the discretized form:

$$F_{BM}(t) = -\boldsymbol{G}\frac{dV(t)}{dt}.$$

To discretize the operator $\left[\frac{\partial}{\partial x}s(x)\frac{\partial}{\partial x}\right]$ we use the finite difference method, i.e.

$$\frac{\partial}{\partial x}s(x)\frac{\partial}{\partial x} \approx \begin{pmatrix} -\frac{1}{dx_1} & \frac{1}{dx_1} & 0 & \\ & \ddots & \ddots & \\ & 0 & -\frac{1}{dx_{N-1}} & \frac{1}{dx_{N-1}} \\ & & 0 & -\frac{1}{dx_N} \end{pmatrix} \mathbf{s} \begin{pmatrix} \frac{1}{dx_1} & 0 & & \\ -\frac{1}{dx_2} & \frac{1}{dx_2} & 0 & \\ & \ddots & \ddots & \\ & 0 & -\frac{1}{dx_N} & \frac{1}{dx_N} \end{pmatrix} =: \mathbf{Sh}. \tag{5.10}$$

Finally, let's denote the discrete form of F_{ohc} by the N-long vector holding the same name F_{ohc} and let's make two further definitions

$$A(t) := \frac{dV(t)}{dt} \tag{5.11}$$

$$S(t) := \frac{\partial^2\sigma(t)}{\partial t^2}. \tag{5.12}$$

Using the introduced notation we can finally write the space-discrete form of the equations (5.5):

$$\frac{dU(t)}{dt} = V(t) \tag{5.13a}$$

$$(\mathbf{G}+\mathbf{m})\,A(t) = -G_S S(t) - \mathbf{k}U(t) - (\mathbf{h}-\mathbf{Sh})V(t) - F_{ohc}(Y(t)) \tag{5.13b}$$

$$\frac{dY(t)}{dt} = W(t) \tag{5.13c}$$

$$\mathbf{M}\frac{dW(t)}{dt} = -\mathbf{C}A(t) - \mathbf{K}Y(t) - \mathbf{H}W(t). \tag{5.13d}$$

If we multiply the equation (5.13d) by $(\mathbf{G}+\mathbf{m})\mathbf{C^{-1}}$ from the left side we get

$$(\mathbf{G}+\mathbf{m})\mathbf{C}^{-1}\mathbf{M}\frac{dW(t)}{dt} = -(\mathbf{G}+\mathbf{m})A(t) - (\mathbf{G}+\mathbf{m})\mathbf{C}^{-1}\mathbf{K}Y(t) - (\mathbf{G}+\mathbf{m})\mathbf{C}^{-1}\mathbf{H}W(t). \tag{5.13e}$$

Then, substituting the term $(\mathbf{G}+\mathbf{m})A(t)$ in (5.13e) by the right hand side of the equation (5.13b) and using (5.11) we get a new version of the system (5.13):

$$\frac{dU(t)}{dt} = V(t) \tag{5.14a}$$

$$(\mathbf{G}+\mathbf{m})\,\frac{dV(t)}{dt} = -G_S S(t) - \mathbf{k}U(t) - (\mathbf{h}-\mathbf{Sh})V(t) - F_{ohc}(Y(t)) \tag{5.14b}$$

$$\frac{dY(t)}{dt} = W(t) \tag{5.14c}$$

$$\begin{aligned}(\mathbf{G}+\mathbf{m})\mathbf{C}^{-1}\mathbf{M}\frac{dW(t)}{dt} = {} & G_S S(t) + \mathbf{k}U(t) + (\mathbf{h}-\mathbf{Sh})V(t) + F_{ohc}(Y(t)) - \\ & - (\mathbf{G}+\mathbf{m})\mathbf{C}^{-1}\mathbf{K}Y(t) - (\mathbf{G}+\mathbf{m})\mathbf{C}^{-1}\mathbf{H}W(t)\end{aligned} \tag{5.14d}$$

The new system (5.14) is a system of $4N$ equations with $4N$ unknowns: $U(t)$, $V(t)$, $Y(t)$ and $W(t)$.

5.2 Discretization of the system of motion equations in time

In this part we treat the problem of the time discretization of the space-discrete system (5.14).

5.2.1 Linear case

Definition of the system to solve

If we consider

$$F_{ohc}(Y(t)) = \mathbf{F}Y(t) \tag{5.15}$$

where $\mathbf{F}$ is an $N \times N$ matrix, then the system of equations (5.14) is linear with respect to the 4N unknown functions.

Let's now define the vector function $Z(t)$ of the length $4N$:

$$Z(t) := \begin{pmatrix} U(t) \\ V(t) \\ Y(t) \\ W(t) \end{pmatrix} \tag{5.16}$$

and the $4N \times 4N$ matrices $\mathbf{M_t}$ and $\mathbf{J}$:

$$\mathbf{M_t} := \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (\mathbf{G}+\mathbf{m}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & (\mathbf{G}+\mathbf{m})\mathbf{C^{-1}M} \end{pmatrix} \tag{5.17}$$

$$\mathbf{J} = \begin{pmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ -\mathbf{k} & -(\mathbf{h}-\mathbf{Sh}) & -\mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{k} & (\mathbf{h}-\mathbf{Sh}) & \mathbf{F}-(\mathbf{G}+\mathbf{m})\mathbf{C^{-1}K} & -(\mathbf{G}+\mathbf{m})\mathbf{C^{-1}H} \end{pmatrix} \tag{5.18}$$

where $\mathbf{I}$ is the N×N identity matrix and $\mathbf{0}$ is the N×N zero matrix and let's finally introduce the $4N$–long vector function $g(t)$:

$$g(t) := \begin{pmatrix} 0 \\ -G_S S(t) \\ 0 \\ G_S S(t) \end{pmatrix}. \tag{5.19}$$

where 0 is the N-long vector of zeros.

Then we can rewrite the linear version of the system (5.14) in the form:

$$\mathbf{M_t}\frac{dZ(t)}{dt} = \mathbf{J}Z(t) + g(t). \tag{5.20}$$

If we multiply the relation (5.20) by the matrix $\mathbf{M_t}^{-1}$ from the left, we get the equation in the form

$$\frac{dZ(t)}{dt} = \mathbf{J_M}Z(t) + g_M(t) \tag{5.21}$$

where $\mathbf{J_M} = \mathbf{M_t}^{-1}\mathbf{J}$ and $g_M = \mathbf{M_t}^{-1}g$.

Time discretization

We denote each time level by $k, k = 0,1,...$, the time step by h_t, the approximate solution at the level k by Z_k and we define $g_{M,k} := g_M(kh_t)$.

Our model disposes of two implemented explicit numerical methods: the explicit Euler method implemented by Ticháček (2014) and the 6-stage Runge-Kutta method implemented by Vetešník and Gummer (2012). Both of the methods are applied to the linear system in the form (5.21) and lead to a direct calculation of Z_{k+1}.

- **Explicit Euler method**

  Following (4.8) we get

$$Z_{k+1} = Z_k + h_t(\mathbf{J_M} Z_k + g_{M,k}). \tag{5.22}$$

- **6-stage Runge Kutta method**

  Following (4.6) we get

$$Z_{k+1} = Z_k + h_t \sum_{i=1}^{s} b_i q_i \tag{5.23}$$

  where $q_i, i = 1,...,s$ are defined by (4.7) with $f(t,Z) = \mathbf{J_M} Z + g_M(t)$. The coefficients c_i, a_{ij} and b_i are given by the Butcher's tableau 4.1.

5.2.2 Nonlinear case

Definition of the system to solve

Let's now consider F_{ohc} as the nonlinear function of Y:

$$(F_{ohc}(Y))_i = 0.1\,\mathbf{F}_{ii}\left(\frac{1}{1 + c_1\exp(-\frac{\mathrm{Y_i}}{\mathrm{y_1}}) + \mathrm{c_2}\exp(-\frac{\mathrm{Y_i}}{\mathrm{y_2}})} - b\right) \tag{5.24}$$

with $y_1 = 0.01139$, $y_2 = 0.03736$, $c_1 = 0.7293$, $c_2 = 1.4974$, $b = 0.30991$ and the same matrix $\mathbf{F}$ as in the linear case.

Let's now proceed analogically to the linear case. We keep the definition (5.16) of the vector function $Z(t)$, the definition (5.17) of the matrix $\mathbf{M_t}$ as well as the definition (5.19) of the vector $g(t)$. Let's now define the matrix $\tilde{\mathbf{J}}$ in the following way:

$$\tilde{\mathbf{J}} = \begin{pmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ -\mathbf{k} & -(\mathbf{h}-\mathbf{Sh}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{k} & (\mathbf{h}-\mathbf{Sh}) & -(\mathbf{G}+\mathbf{m})\mathbf{C^{-1}K} & -(\mathbf{G}+\mathbf{m})\mathbf{C^{-1}H} \end{pmatrix} \tag{5.25}$$

where $\mathbf{I}$ is the N×N identity matrix and $\mathbf{0}$ is the N×N zero matrix and finally let's introduce the vector function $h(Z)$ of the length 4N:

$$h(Z) := \begin{pmatrix} 0 \\ -F_{ohc}(Z(2N+1:3N)) \\ 0 \\ F_{ohc}(Z(2N+1:3N)) \end{pmatrix}. \tag{5.26}$$

where 0 is the N-long vector of zeros.

Then we can rewrite the nonlinear version of the system (5.14) in the following simplified form:

$$\mathbf{M_t}\frac{dZ(t)}{dt} = \tilde{\mathbf{J}}Z(t) + g(t) + h(Z(t)) \tag{5.27}$$

or, by multiplying (5.27) by $\mathbf{M_t}^{-1}$ from the left, in the form

$$\frac{dZ(t)}{dt} = \tilde{\mathbf{J}}_M Z(t) + g_M(t) + h_M(Z(t)), \tag{5.28}$$

where $\tilde{\mathbf{J}}_\mathbf{M} = \mathbf{M_t}^{-1}\tilde{\mathbf{J}}$, $g_M = \mathbf{M_t}^{-1}g$ and $h_M = \mathbf{M_t}^{-1}h$.

Time discretization

We adopt here the notation of h_t, Z_k and g_k from the linear case.

Applied to the nonlinear equation (5.28), explicit methods lead to a direct expression of Z_{k+1} just as it was in the linear case. The nonlinearity therefore brings no problem to computations that follow from the application of explicit methods. Here we present the explicit Euler method implemented by Ticháček (2014) and the 6-stage Runge-Kutta method implemented by Vetešník and Gummer (2012).

- **Explicit Euler method**
  Following (4.8) we get

$$Z_{k+1} = Z_k + h_t(\tilde{\mathbf{J}}_{\mathbf{M}} Z_k + g_{M,k} + h_M(Z_k)). \tag{5.29}$$

- **6-stage Runge Kutta method**
  Following (4.6) we get

$$Z_{k+1} = Z_k + h_t \sum_{i=1}^{s} b_i q_i \tag{5.30}$$

  where $q_i, i = 1...s$ are defined by (4.7) with $f(t,Z) = \tilde{\mathbf{J}}_{\mathbf{M}} Z + g_M(t) + h_M(Z)$. The coefficients c_i, a_{ij} and b_i are given by the Butcher's tableau 4.1.

5.3 Computer program description

We use a MATLAB program that computes the numerical solution $Z_k, k = 0,1,...$ of the linear or nonlinear system in both forms: original ((5.20) or (5.27)) or multiplied by the $\mathbf{M_t}^{-1}$ matrix ((5.21) or (5.28)). It accepts the input in the form of a signal and creates the output in the form of the vector Z_k for specified range of k. The particular sections of the program are presented in the following overview.

Input signal and parameters

It is generally possible to set the input signal $\sigma(t)$ to an arbitrary form. In this work we will consider for simplicity only cosine waves of a particular frequency `f` and amplitude `A`. To get the term $S(t)$ that is present in the motion equations, we compute the second time derivative of $\sigma(t)$ and we multiply it by the *envelope* function $1/2\tanh(a(t-t0)+1)$ with $a = 400$ and $t0 = 0.012$. The objective of this multiplication is to obtain a very slow increase of the signal amplitude at the beginning of the simulation. This causes that no secondary oscillations of the solution are produced. The envelope function asymptotically approaches 1 as $t \to \infty$ hence the original signal remains practically not modified by the multiplitcation for higher time values. An example of the final $S(t)$ is represented in Figure 5.2.

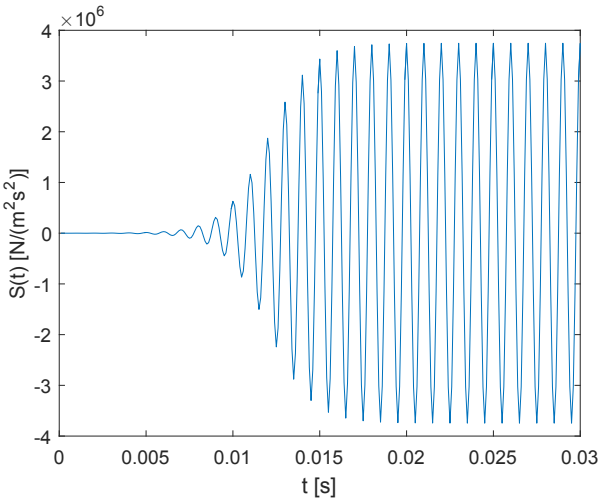


Figure 5.2: **Second time derivative of the input signal multiplied by the envelope function. Input signal has frequency 1000 Hz and amplitude 90 dB.**

The main input parameters of the program are the following:

`A`	[dB] input signal amplitude
`f`	[Hz] input signal frequency
`h_t`	[ms] time step
`T`	[s] duration of the simulation
`N`	number of discretization points in space
`algorithm`	numerical method

We make an assumption that the unit of the input signal amplitude `A` that is denoted by dB in the model means the unit dB SPL.

Program core

The core of the program is formed by the time loop. This, at each step, calls the *a priori* specified algorithm to solve the system of equations.

Output data

The result of the simulation are: the approximate BM displacement and velocity and the approximate hair cell bundle displacement and velocity in time (forming the vector Z_k). The part that interests us the most is the BM displacement since it enters to the electrical part of the model. An example of the output BM displacement is represented in Figure 5.3.

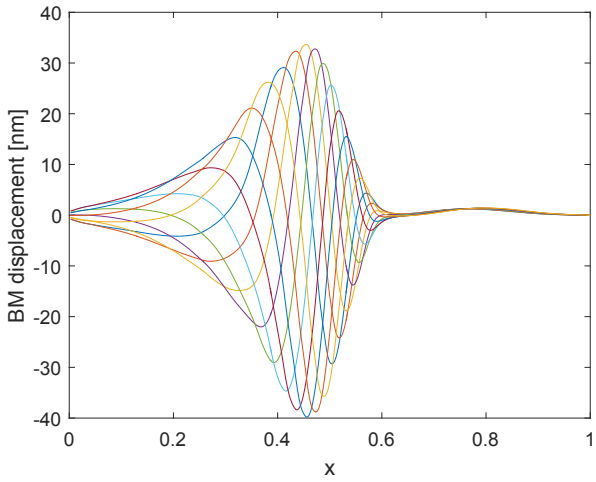


Figure 5.3: **10 consecutive BM displacements in time taken after 20 ms of the simulation. Input parameters: h_t = 0.1 ms, f = 1000 Hz, A = 90 dB.**

Part II

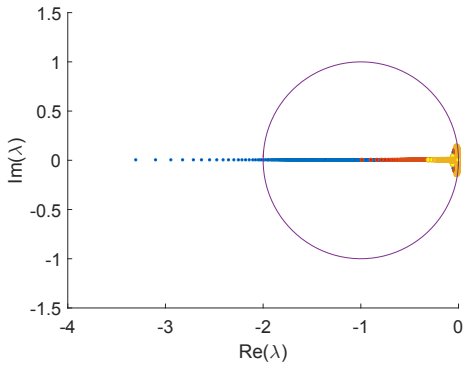
Applications

Goals of our work

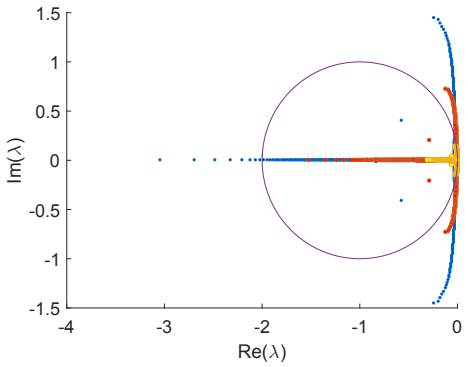
Up to now we have used only explicit numerical methods to solve the governing equations of the mechanical model. Explicit methods turn out to be efficient in cases where we work with a small number of discretization points in space (around 300). They ensure a fast computation and the stability in the range of time steps that we usually need. The number of discretization points around 300 is however insufficient for our current purposes. Why?

As we mentioned in the Introduction, the aim of our current work is to introduce the physiological quantities into the connection between the mechanical and the electrical models. Particularly, we want that the connection be realized individually at each of the 3000 cochlear transversal sections with a unicellular width. These sections are placed equidistantly in the cochlear duct. We explained in the Section 3 that the electrical model represents the cochlea as a set of longitudinally coupled electrical circuits. The number of transversal cochlear sections that are represented by electrical circuits can be chosen essentially arbitrarily hence there is no problem to set the number of 3000. A more complicated situation occurs with the mechanical model. Here we work with a nonequidistant division of the position along the cochlear duct. To get 3000 equidistant positions we will need firstly to make use of interpolation techniques and secondly to study how the solution at these points changes with the varying number of discretization points. Due to their level of nonequidistancy, we will make the study in the range of approximately [300,4000] of the discretization points. We need therefore to ensure that the mechanical model is adapted to this range of discretization points.

Let's return back to our original model with only explicit methods. If we increase the number of discretization points, holding the time step constant at a reasonable value, the stability of the system becomes violated. We are forced to choose much smaller time steps to make the simulation work properly (see the related Figure 5.4). A more efficient way to treat this problem in these cases appears to be the use of implicit numerical methods. These enable us to choose rather big time steps and the stability of the computation persists. Our goal in this work is to propose and test suitable implicit numerical methods on the mechanical model and to choose the one that is the most suitable to our requirements.



(a) **Fixed time step $\mathbf{h = 0.001}$ ms.** $N = 300$ (yellow), $N = 1000$ (red), $N = 2000$ (blue)



(b) **Fixed number of spatial discretization points $\mathbf{N = 300}$.** $h = 0.001$ ms (yellow), $h = 0.005$ ms (red), $h = 0.01$ ms (blue)

Figure 5.4: **Eigenvalues λ of $(\mathbf{h_t J})$ in the complex plane, with J the Jacobian matrix of the linear system .** N denotes the number of discretization points. The circle in both plots denotes the stability region of the explicit Euler method. **(a)** shows that increasing N the absolute value of the real part of eigenvalues increases and the eigenvalues go out of the stability region of the explicit Euler method at some point, **(b)** shows that to put all the eigenvalues to the explicit Euler stability region we need to diminish the time step appropriately.

6. Linear model

Implementation

We opt for four implicit numerical methods: IE, CN, BDF2 and BDF3 and we implement them into our model. All the four methods, when applied to the system in the form (5.20) or (5.21), lead to the system of linear equations

$$\mathbf{A} Z_{k+1} = b_k \tag{6.1}$$

that we solve for Z_{k+1} at each time step k. In the case of implicit methods it is more convenient to work with the linear system in the form (5.20) than (5.21). In this way we omit the computation of the inverse of the matrix $\mathbf{M_t}$ and moreover we avoid its multiplication by elements of the right hand side of (5.20) that would otherwise destruct their sparsity.

In practice, since the matrix $\mathbf{A}$ is constant in time, it is efficient to use the LU decomposition of $\mathbf{A}$ to solve (6.1). In fact, the constancy of $\mathbf{A}$ allows us to compute the most expensive operation, i.e. the LU decomposition (see the part 4.2.1), only once and this is done before the time loop. At each time step, we compute therefore only the value of b_k and we perform one forward and one backward substitution (4.30). These substitutions are already very fast, particularly in MATLAB with the function accorded to the sign \ and sparse nature of $\mathbf{L}$ and $\mathbf{U}$ matrices. Particularly in our model we use the LU decomposition with the added pivoting to improve the stability properties of the computation. The backward and forward substitutions used to get the solution Z_{k+1} of the system (6.1) follow therefore the modified formulas (4.32)

In the following overview we present the implemented implicit methods written in the form (6.1).

- **Implicit Euler method**
  Following (4.9) we get

$$(\mathbf{M_t} - h_t \mathbf{J})\, Z_{k+1} = \mathbf{M_t} Z_k + h_t g_{k+1} \tag{6.2}$$

- **Crank-Nicolson method**
  Following (4.10) we get

$$\left(\mathbf{M_t} - \frac{h_t}{2}\mathbf{J}\right) Z_{k+1} = \mathbf{M_t} Z_k + \frac{h_t}{2}\left(g_{k+1} + g_k + \mathbf{J} Z_k\right) \tag{6.3}$$

- **Backward difference formula of the 2^{nd} order**
  Following (4.11) we get

$$\left(\frac{3}{2}\mathbf{M_t} - h_t \mathbf{J}\right) Z_{k+1} = 2\mathbf{M_t} Z_k - \frac{1}{2}\mathbf{M_t} Z_{k-1} + h_t g_{k+1} \tag{6.4}$$

- **Backward difference formula of the 3^{rd} order**
  Following (4.12) we get

$$\left(\frac{11}{6}\mathbf{M_t} - h_t\mathbf{J}\right) Z_{k+1} = 3\mathbf{M_t} Z_k - \frac{3}{2}\mathbf{M_t} Z_{k-1} + \frac{1}{3}\mathbf{M_t} Z_{k-2} + h_t g_{k+1} \tag{6.5}$$

MATLAB implementation

Each of the methods is written in a separate MATLAB script, see functions `cochleaSolverIE_lin`, `cochleaSolverCN_lin`, `cochleaSolverBDF_2_lin` and `cochleaSolverBDF_3_lin` in Appendix. All of them have a similar structure so here we explain only the basic case of the implicit Euler method.

To the `cochlea.m` home script before the time loop we add:

- the calculation of the $\mathbf{J}$ and $\mathbf{M_t}$ matrices
- the calculation of the $\mathbf{A}$ matrix

```
A = M_t-h_t*J;
```

- LU decomposition with pivoting of `A`

```
[L,U,P] = lu(A).
```

To the `cochleaSolverIE_lin` script we transfer the matrices `L`, `U` and `P` and we insert there

```
RHS = P*(gf*h_t+y);
y = U\ (L\ RHS);
```

where `gf` means g_{k+1} and `y` represents $\mathbf{M_t} Z_k$ and Z_{k+1} from the first and from the second line of the code respectively.

6.1 Experiment 1: convergence of the BM displacement computed by implicit numerical methods with decreasing time step

The first experiment aims at showing the behaviour of the solution that corresponds to the BM displacement, calculated by the four implicit numerical methods with a varying time step. The results should reveal which of the calculated BM displacements converges the most rapidly to the exact solution and which of them exhibits the smallest deviation from the exact solution for all of the values of the time step.

Initialization

We run a series of simulations with different time steps, different numerical methods and different frequencies of the input signal. The input parameters for this experiment are:

```
N = 1000;                          % number of discretization points in space
A = 20;                                % [dB] amplitude of the input signal
f = [200,2000,15000];                % [Hz] frequency of the input signal
h_t_200Hz = [0.5,0.1,0.05,0.01];                %[ms] time steps for 200Hz
h_t_2000Hz = [0.05,0.01,0.005,0.001];           %[ms] time steps for 2000Hz
h_t_15000Hz = [0.005,0.001,0.0005,0.0001];  %[ms] time steps for 15000Hz
t_save = 0.025;                  % [s] time of the results saving
```

The number of discretization points in space and the amplitude are kept unchanged in the simulations since we do not expect that they have a significant influence on the results. The frequencies are taken from different parts of the range of human hearing. They are chosen so that different regions of the BM are activated by the input signal (200 Hz at the cochlear base, 2000 Hz in the middle and 15000 Hz at the cochlear apex). The time steps are chosen for each frequency separately and are selected as four numbers from approximately 1/10 of the input signal time period to approximately 1/500 of the input signal time period. The variable `t_save` designates the time in milliseconds at which we save the instantaneous BM displacement. Its value is chosen such that the BM displacement be equilibrated after the 20 ms of continuous increase in the input signal amplitude. We denote the computed BM displacement at the time `t_save` by U.

In addition to the tested simulations we generate a so-called "reference solution" for each frequency of the input signal. The input parameters to this simulation are the following:

```
N = 1000;                % number of discretization points in space
A = 20;                 % [dB] amplitude of the input signal
h_t = 0.0001                 % [ms] time step
f = [200,2000,15000]         % [Hz] input signal frequencies
method = RK6;                % used numerical method
t_save = 25;   % [ms] time of results saving
```

In this simulation we use the explicit RK6 method and we used very small time step of $h_t = 0.00001$ ms. We denote the reference solution at time `t_save` by U_{ref}.

Results

We plot the relative error $||U - U_{ref}||/||U_{ref}||$ for each frequency from `f` interval. The resulting curves are plotted in the figure 6.1.

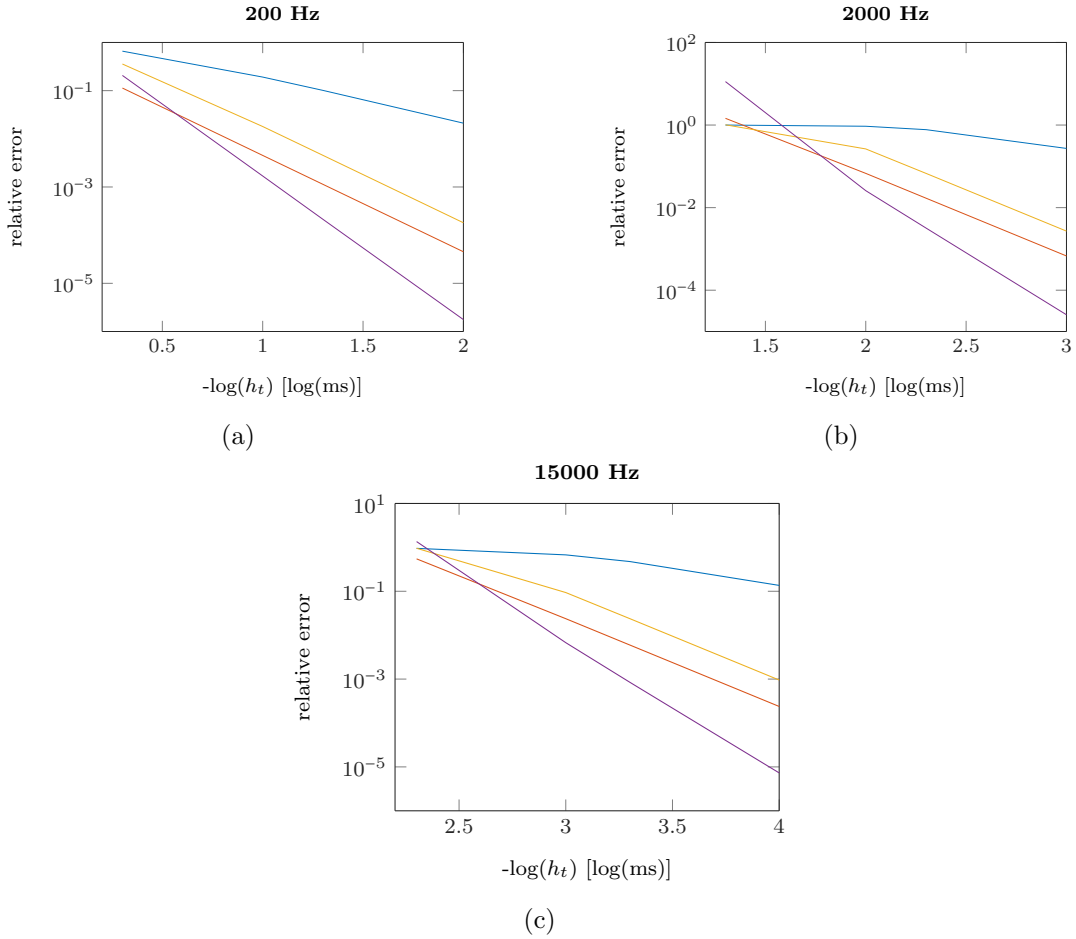


Figure 6.1: **Convergence of the calculated BM displacement to the reference solution in the linear model; IE (blue), CN (red), BDF2 (yellow), BDF3 (violet).** At each figure we plot the relative error $\|U - U_{ref}\|/\|U_{ref}\|$ of the calculated BM displacement at 25 ms with relation to the reference solution for different methods and for different time steps.

Discussion and conclusions

The figure 6.1 reveals firstly that the BM displacement calculated by the four implicit methods converges to the reference solution, each of them with the rate that corresponds to the order of the method. Secondly, it shows that the most convenient method for our purpose is the CN method. This combines a high rate of convergence and a low sensibility of the solution to the change of the time step. Hence if we run the simulation with the time step between 1/10 and 1/50 of the input signal time period, we get a result very close to the reference solution. Moreover, to get a solution closer to the reference solution, we do not need to decrease the time step too much. For the values of 1/50 of the time period and lower, the method gives already a sufficiently good approximation.

7. Nonlinear model

A significantly different situation from the previous one happens with the nonlinear system. The four implicit methods IE, CN, BDF2 and BDF3, if applied to the linear system (5.20), lead to a system of linear equations (6.1) easy to solve using LU decomposition. However, their application to the nonlinear problem (5.27) leads to the nonlinear system in the form

$$\mathbf{A}Z_{k+1} - ch(Z_{k+1}) = b_k \tag{7.1}$$

with c a constant depending on the method and on the time step h_t, that is much more difficult to solve. Note that $\mathbf{A}$ is in the nonlinear case different from $\mathbf{A}$ in the linear case.

As in the linear case, we use the version 5.27 of the system to solve since it keeps the original sparsity of the problem.

Nonlinearity

We remind that the nonlinearity $h(Z)$ has the form

$$h(Z) = \begin{pmatrix} 0 \\ -F_{ohc}(Z(2N+1:3N)) \\ 0 \\ F_{ohc}(Z(2N+1:3N)) \end{pmatrix} \tag{7.2}$$

with

$$(F_{ohc}(Y))_i = 0.1\,\mathbf{F}_{ii}\left(\frac{1}{1 + c_1\exp(-\frac{Y_i}{y_1}) + c_2\exp(-\frac{Y_i}{y_2})} - b\right) \tag{7.3}$$

where $\mathbf{F}_{ii}$, y_1, y_2, c_1, c_2 and b are specific coefficients related to the problem, particularly $y_1 = 0.01139$, $y_2 = 0.03736$, $c_1 = 0.7293$, $c_2 = 1.4974$ and $b = 0.30991$. Let's denote by $f(Y_i)$ the function

$$f(Y_i) = \frac{1}{1 + c_1\exp(-\frac{Y_i}{y_1}) + c_2\exp(-\frac{Y_i}{y_2})} - b. \tag{7.4}$$

Then f has the form specified in the figure 7.1.

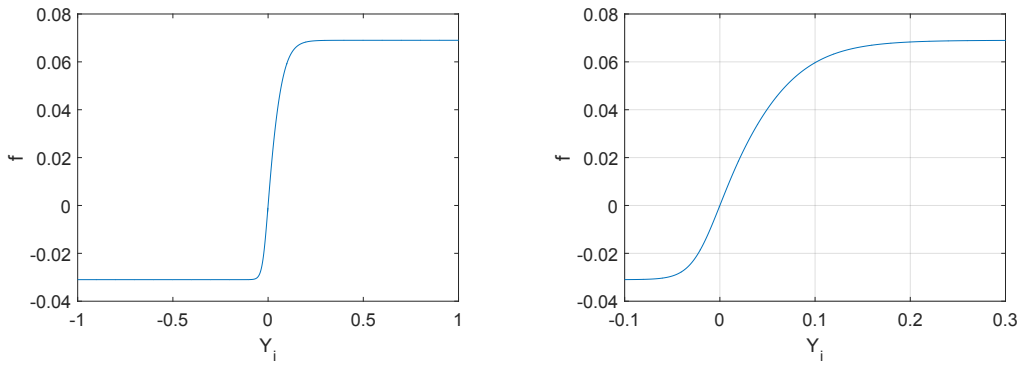
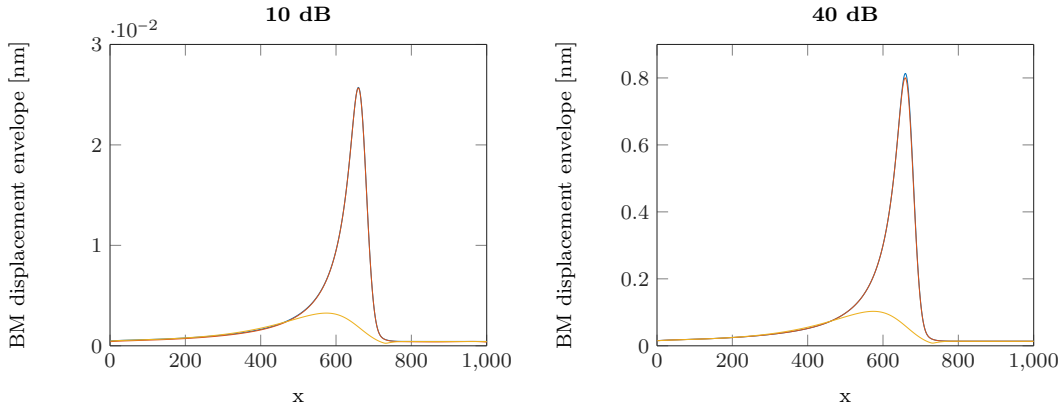


Figure 7.1: **The shape of the nonlinearity.** (a) on the interval [-1,1], (b) zoomed to the interval [-0.1,0.3]

The function f takes the value 0 at $Y_i = 0$ and moreover it has the derivative of 1 at this point. As $Y_i \to \infty$, f tends to the value $1 - b$ and as $Y_i \to -\infty$ f tends to $-b$.

To see what is the influence of the nonlinear term on the solution, we performed an experiment where we ran successively the simulations with various values of the input signal amplitude and with different types of the model: linear, nonlinear and passive. The last term means that we put F_{ohc} to be the identically zero function. The ways how these versions of model behave at various amplitudes are shown in Figure 7.2.



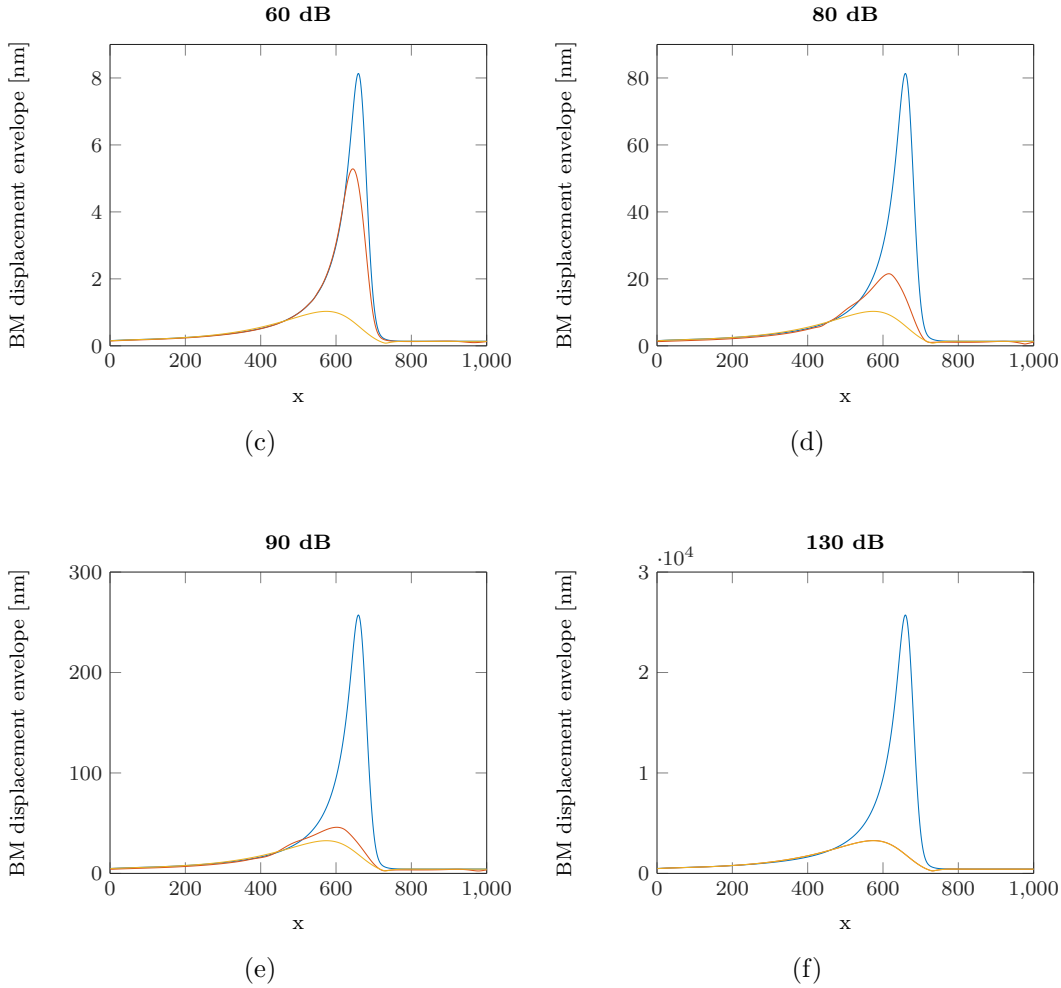


Figure 7.2: **Envelope of the BM displacement with the input signal of 1000 Hz. Linear model (blue), nonlinear model (red), passive model (yellow).** (a) shows that for 10 dB, envelopes for linear and nonlinear models coincide, hence the nonlinear model is in the linear regime. The nonlinear model starts to perceive the nonlinearity at amplitudes above 40 dB, as can be seen in (b) - there is a tiny shift of the red and blue curves at their maxima. (c), (d), (e), (f) show that the behaviour of the nonlinear model approaches to the behaviour of the passive model as the amplitude increases up to the maximal value of 130 dB.

Implementation

Here we present two approaches that we use to fix the nonlinearity when applying implicit numerical methods. The approaches follow the reasoning in the part 4.1.4

- **Method 1: explicitly treated nonlinearity and the LU decomposition**
  The first way how to deal with the nonlinear term $ch(Z_{k+1})$ in (7.1) is to include it into the right hand side of (7.1) and to proceed iteratively at each time step until the convergence of approximate solutions[1] occurs. This method is explained

[1] By *approximate solution* we mean in this chapter the solution of the system obtained at each iteration.

in detail in the part *Explicitly treated h* in the section 4.1.4. Using this method we solve a system of linear equations (4.15) at each iteration, involving the matrix $\mathbf{A}$ that is constant for each iteration and each time step and involving the right hand side in the form (4.16) or (4.17) that needs to be regularly recalculated. To solve the linear system we use the method related to the LU decomposition with pivoting of the matrix $\mathbf{A}$, see part 4.2.1 for description of this method. Thanks to the constancy of $\mathbf{A}$, the LU decomposition is performed only once before the time loop and we perform solely the backward and the forward substitutions (4.32) at each iteration to get the approximate solution. The final solution Z_{k+1} is then given by making the limit (4.18) of approximate solutions Z^i_{k+1}.

- **Method 2: Newton method and GMRES**
  The second method to treat the nonlinear term is to perform its linearization as it is done in the expression (4.19), see *Newton method-linearization* in the section 4.1.4). This method leads to a system of linear equations (4.20) at each iteration, involving a different matrix $\mathbf{B}^i_k$ and different right hand side d^i_k each time. To solve the linear system we opt for an iterative method that is expected to compute quickly the approximate solution. Particularly, we use the GMRES method that is a commonly used method to solve linear systems with a general matrix (for the description of the method see the section 4.2.2). To improve the convergence properties of GMRES we test the application of various preconditioners. We use particularly the split preconditioners and opt for the incomplete LU factorization, see the section *Preconditioning* in the section 4.2.2 for further details.

Both of these methods are iterative, i.e. at each time step we get a sequence of approximate solutions that is supposed to converge to the final solution. Method 1 computes each iteration very quickly but may converge slowly since it does not take into account the particular shape of the nonlinearity. Method 2 computes each iteration slowlier but may converge quickly since it involves the form of the nonlinear function into the iteration process. It is not possible to estimate a priori which of the methods is more effective.

To test both methods and their effectiveness we apply them to the CN algorithm and we perform several experiments. The observations of our experiments are summarized in the following section.

7.1 Method 1 optimization

We dispose of the Method 1 applied to the CN algorithm (see `cochleaSolverCN_nonlin_M1.m`) and we seek for the version of this method that ensures a fast computation of each time step and leads moreover to a good approximation of the solution. The criterion of a good approximation of the solution is given by an a

priori selected tolerance value *tol*. Particularly, at each iteration i at each time step k we find the approximate solution Z_{k+1}^i and we check if this satisfies the criterion

$$\frac{\|Z_{k+1}^i - Z_{k+1}^{i-1}\|}{\|Z_{k+1}^i\|} < \text{tol}. \tag{7.5}$$

If the criterion is satisfied then we accept Z_{k+1}^i as the solution at the time step k, otherwise we continue in the iteration process. The ratio in the left hand side of 7.5 will be referred to as the *relative error*.

7.1.1 Experiment 2: convergence test of direct and extrapolated versions

We have two versions of the Method 1 at our disposal: 4.16 that will be denoted by Version A and 4.17 denoted by Version B. In this experiment we test these versions to see which of them ensures faster achievement of the criterion 7.5. Since the computational time of each iteration is approximately the same for both versions, we will compare the number of iterations at each time step.

Initialization

We run a series of simulations with the parameters

```
N = 1000;
f = 1000;                % [Hz]
h_t = 0.1;                % [ms]
T = 0.025;                        % [s] simulation length
A = [20,70,130];        %[dB]
tol = 1e-4;                % tolerance value as the stopping criterion
method = ['A','B'];        % Version A and B of the Method 1
```

and we save the total number of iterations at each time step for Version A and B separately. We fix the variables `N`, `f` and `h_t` since we do not expect that they would influence the conclusion of the experiment. The amplitudes are chosen from different parts of the interval [0,140] dB that we take into account in this work.

Results

The results are summarized in Figure 7.3.

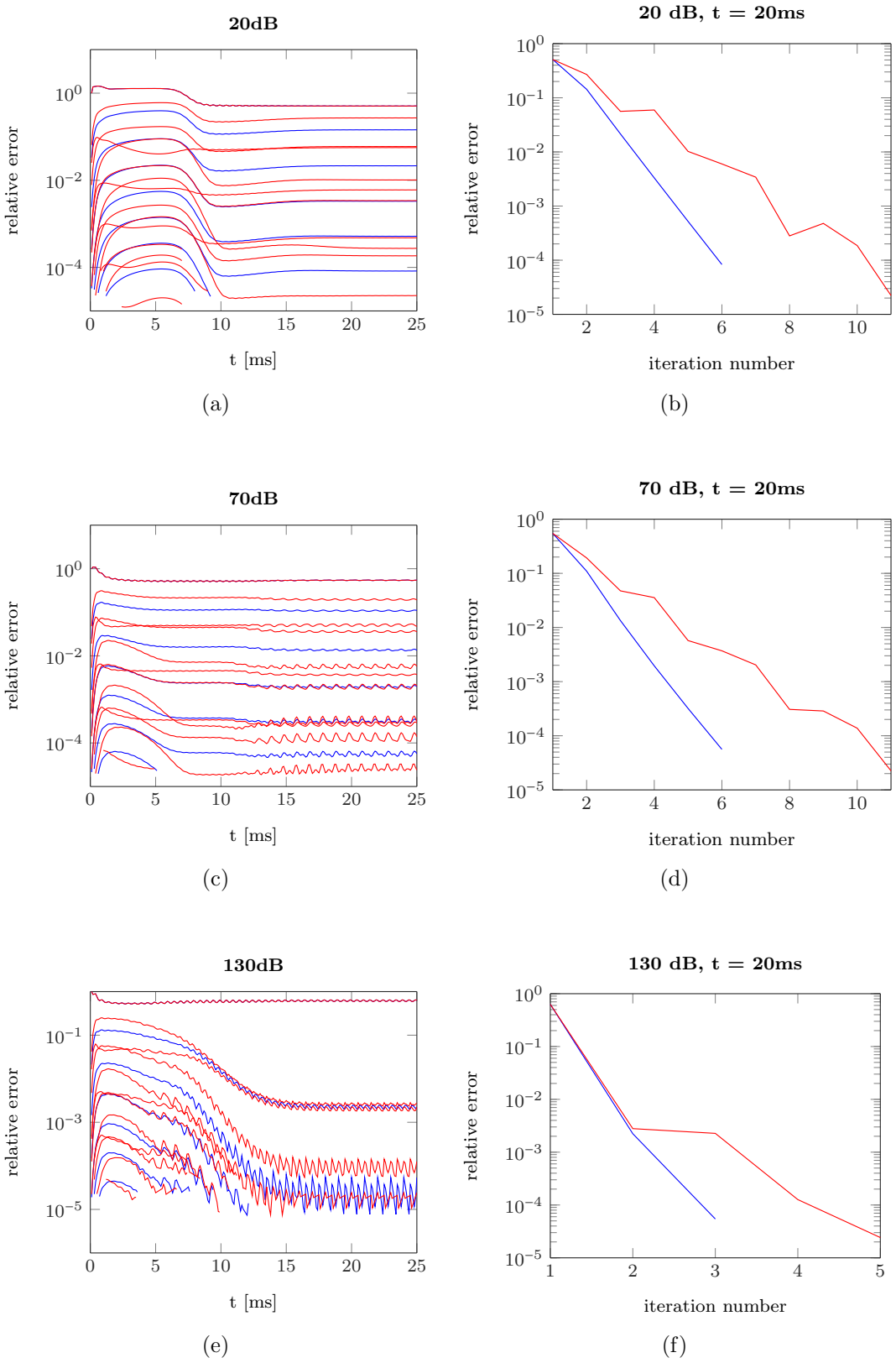


Figure 7.3: **Convergence of Version A (blue) and B (red) of the Method 1.** (a), (c), (e) show the relative error $\|Z_{k+1}^i - Z_{k+1}^{i-1}\|/\|Z_{k+1}^i\|$ at each iteration i at each time step k from 0 to 25 ms. (b), (d), (f) show the relative error as a function of the iteration number i at the fixed time of 20 ms.

Discussion and conclusions

Figure 7.3 shows that for each of the tested input signal amplitudes and for each time step Version A converges more rapidly to the desired tolerance than Version B. The results show therefore that to estimate the next approximate solution it is better to use the previous approximate solution rather than a linear extrapolation of the two previous ones. This result is related to the value of `tol` and could be different if we chose a much smaller value. Since the value of 10^{-4} is satisfactory for our purposes, we keep this result and will use Version A in the following computations.

Another observation can be concluded from Figure 7.3. When we are in the strictly nonlinear regime (130 dB from 15 ms), the convergence of both versions is much faster than in the linear regime (20 dB) - in the nonlinear regime we need $3-5$ iterations and in the linear one we need $6-11$ iterations. This is a direct consequence of the form of the nonlinearity: it has a very small slope from a certain point. The remaining question is if the approximate solutions converge correctly, i.e. if the final solution is close to the exact solution of the equation. This question will be answered later in the Experiment 6.

7.2 Method 2 optimization

As in the case of Method 1, we aim to optimize Method 2 applied to the CN algorithm (see the script `cochleaSolverCN_nonlin_M2.m`) so that it calculates the solution effectively. The good approximation of the solution is imposed by a given tolerance value *tol* and the related stopping criterion 7.5. The time in which the approximate solutions attain this tolerance can be adjusted by various factors. Particularly, we will adjust the GMRES preconditioner and the type of the Newton method.

7.2.1 Experiment 3: choice of the GMRES preconditioner

As we stated in the section 4.2.2, a commonly used way to improve the convergence properties of the GMRES method is the use of the preconditioning technique. We stated also that efficient preconditioners fulfill the criterion 4.36, that is 4.37 for the split preconditioners.

In this experiment we will focus on the split preconditioning with the use of the incomplete LU factorization technique. We will consider three situations: 1. ilu($\mathbf{B_k^i}$) at each Newton iteration; 2. ilu($\mathbf{B_k^1}$) at each time step and only in the first Newton iteration; 3. ilu($\mathbf{A}$) before the time loop - only once in the simulation. These situations will be referred to as *positions*.

In the first part we will test various *ilu types* and we will observe their influence on the number of GMRES iterations and on the GMRES duration. We will choose the best three of them. Then we will test various *positions* of the `ilu` process and we will

choose the most efficient of them. Finally we will investigate in detail the *ilu types* in the chosen position.

Eperiment 3A - ilu type test

We make a general test of *ilu types* to see which of them could be good candidates to generate effective GMRES preconditioning matrices. We test particularly 11 types.

Initialization

We run a series of simulations with the parameters

```
A = 90;         % [dB]
f = 1000;         % [Hz]
h_t = 0.1;         % [ms]
N = 300;
T = 0.0005;         %[s]
t_save = 0.5         % [ms]
position = [1,3];
tol_Newton = 1e-4;
tol_gmres = 1e-6;
ilu_type = 1:11;

%------ ILU TYPE--------
% 1  [L,U] = ilu(sparse(Matrix),struct('type','ilutp','droptol',1e-6));
% 2  [L,U] = ilu(sparse(Matrix),struct('type','ilutp','droptol',1e-4));
% 3  [L,U] = ilu(sparse(Matrix),struct('type','ilutp','droptol',1e-2));
% 4  [L,U] = ilu(sparse(Matrix),struct('type','nofill'));
% 5  [L,U] = ilu(sparse(Matrix),struct('type','nofill','milu','row'));
% 6  [L,U] = ilu(sparse(Matrix),struct('type','nofill','milu','col'));
% 7  [L,U] = ilu(sparse(Matrix),struct('type','crout','droptol',1e-6));
% 8  [L,U] = ilu(sparse(Matrix),struct('type','crout','droptol',1e-4));
% 9  [L,U] = ilu(sparse(Matrix),struct('type','crout','droptol',1e-2));
% 10 [L,U] = ilu(sparse(Matrix),struct('type','crout','milu','row',...
%        'droptol',1e-6));
% 11 [L,U] = ilu(sparse(Matrix),struct('type','crout','milu','col',...
%        'droptol',1e-6));

%----GMRES comand in cochleaSolverCN_nonlin_M2.m--
[y,~] = gmres(B,d,[],tol_gmres,200,L,U,z1);
```

hence for 11 *ilu types* and for the *position* 1 and 3. Since we make only a quick general test, we stop the simulation already at the beginning (after 5 time steps) and we save the GMRES duration (in seconds denoted by *timeGmres*), the number of GMRES iterations (*iter*) and the `ilu` duration (in seconds denoted by *timeIlu*), all in the first Newton iteration.

Results

The results are summarized in Table 7.1. Figures 7.4 and 7.5 represent selected data from Table 7.1. In Figure 7.4 we plot *timeGmres* for the two positions and *timeGmres*+*timeIlu* for the position 1. The latter should represent the approximate duration of one Newton iteration for the position 1 while *timeGmres* represents itself the duration of a Newton iteration for the position 3. Figure 7.5 shows the comparison of *iter* for each method and position.

	timeGmres		*timeIlu*		*iter*	
ilu type	*position* 1	*position* 3	*position* 1	*position* 3	*position* 1	*position* 3
1	0,047	0,083	0,274	0,185	2	5
2	0,076	0,122	0,223	0,181	5	6
3	0,409	0,733	0,038	0,043	67	106
4	0,038	0,055	0,099	0,109	10	14
5	1,044	0,159	0,113	0,136	164	40
6	0,726	0,991	0,099	0,113	130	161
7	0,313	0,247	0,188	0,141	65	59
8	1,317	1,368	0,048	0,059	200	200
9	0,974	1,216	0,031	0,041	166	200
10	0,278	0,243	0,194	0,150	61	57
11	0,184	0,162	0,184	0,137	43	41

Table 7.1: **General test of *ilu types* at *position* 1 and 3.** *timeGmres* is the GMRES duration in seconds, *timeIlu* is the ilu factorization duration in seconds and *iter* is the number of GMRES iterations.

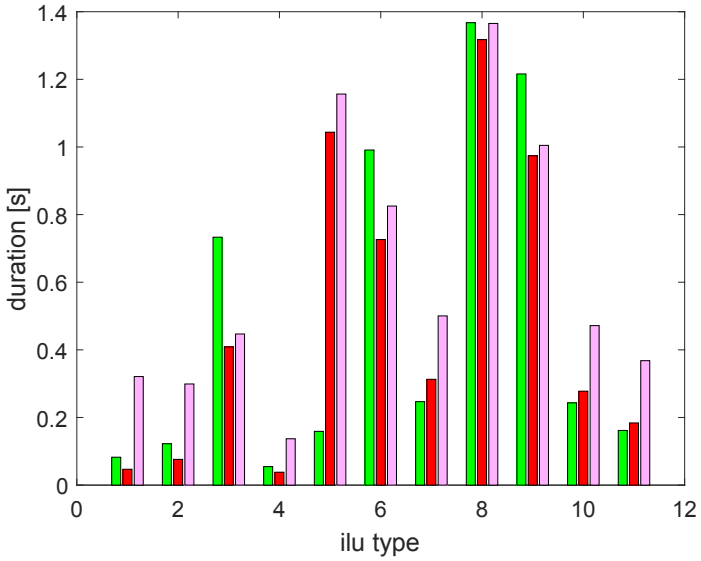


Figure 7.4: **GMRES and one Newton iteration duration comparison for 11 *ilu types* and *positions* 3 and 1; *timeGmres* for the position 3 (green), *timeGmres* for the position 1 (red), *timeGmres*+*timeIlu* for the position 1 (pink).** The green and pink bars represent approximately the Newton iteration duration for respectively the *position* 3 and 1.

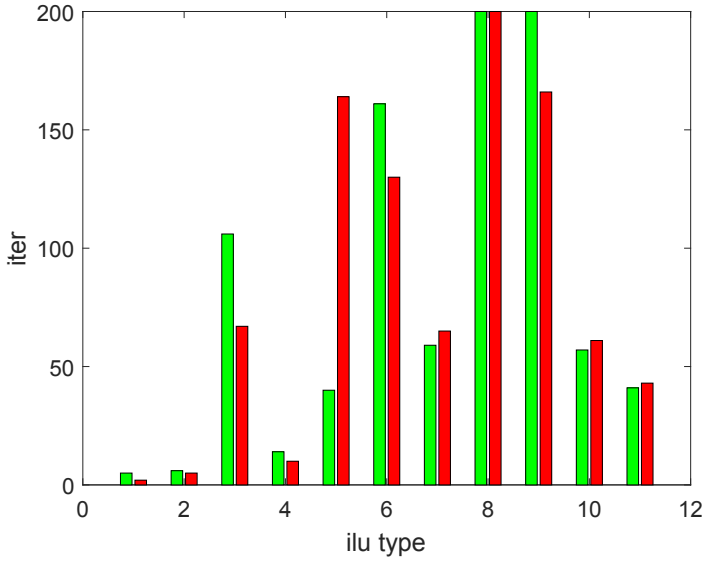


Figure 7.5: **Comparison of the number of GMRES iterations for 11 *ilu types* and *positions* 3 and 1; *position* 3 (green), *position* 1 (red).**

Discussion and conclusions

For both positions 1 and 3, the methods 1, 2 and 4 turn out to be the most efficient. We therefore stick to these three methods and we study them further in the following experiments.

Experiment 3B - 1, 2, 3 *position* and 1, 2, 4 *ilu type* test

We test the *positions* 1, 2 and 3 and the *ilu types* 1, 2 and 4 from the previous experiment to see which combination will be the most efficient[2] to precondition our system. The *position* 1 ensures a fast convergence of the GMRES process at each iteration because the criterion (4.37) is well satisfied in this case. This *position* however requires the `ilu` computation at each Newton iteration that is generally rather time consuming. The *position* 2 reduces the computation of `ilu` to only once at each time step but it also causes that the condition (4.37) is satisfied less strictly from the second Newton iteration. Hence it needs more GMRES iterations to converge in the second and the following Newton iterations. The *position* 3 requires only one computation of `ilu` in the whole simulation but may lead to very small GMRES convergence rate since the matrix $\mathbf{A}$ may be very different from the matrix $\mathbf{B}_{\mathbf{k}}^{\mathbf{i}}$ in some cases. Except from the *position*, *ilu type* may also more or less influence the effectiveness of the preconditioner. While the *ilu type* 1 leads to reliable[3] preconditioning matrices, these may have a

[2]By *efficient* preconditioner we mean the preconditioner that causes a small GMRES duration. This may be related to a small number of GMRES iterations or to a small duration of particular GMRES iterations.

[3]By *reliable* split preconditioning matrices we mean the matrices that satisfy well the condition (4.37).

big number of nonzero elements. GMRES converges therefore in a small number of iterations but these may be rather time-consuming. The *ilu type* 2 is a softer version of the *ilu type* 1 - it produces sparser but less reliable preconditioning matrices. The *ilu type* 4 produces sparse matrices that ensure a fast computation of each GMRES iteration but may not ensure its fast convergence. It is therefore difficult to conclude a priori which of the combinations will be the most effective to use in our system. We make a series of tests to see the result.

Initialization

We run a series of simulation with the parameters

```
A = 90;         % [dB]
f = 1000;         % [Hz]
h_t = 0.1;         % [ms]
N = [300,1000,2000];
T = 0.025;         % [s]
tol_Newton = 1e-4;
tol_gmres = 1e-6;
position = [1,2,3];
        % 1 - at each iteration
        % 2 - at each time step
        % 3 - once before the time loop
ilu_type = [1,2,4];
% 1  [L,U] = ilu(sparse(Matrix),struct('type','ilutp','droptol',1e-6));
% 2  [L,U] = ilu(sparse(Matrix),struct('type','ilutp','droptol',1e-4));
% 4  [L,U] = ilu(sparse(Matrix),struct('type','nofill'));

%----GMRES comand in cochleaSolverCN_nonlin_M2.m--
[y,~] = gmres(B,d,[],tol_gmres,200,L,U,z1);
```

The value of `A` was chosen such that during the 25 ms of the simulation we could observe the behaviour at the linear and the nonlinear regime, with the emphasis to the nonlinear regime. The linear regime is present at the time before 5 ms. We make the test of the *position* and the *ilu type* for three different numbers of N, ie for three different sizes of the problem.

Results

The results are summarized in Figure 7.6.

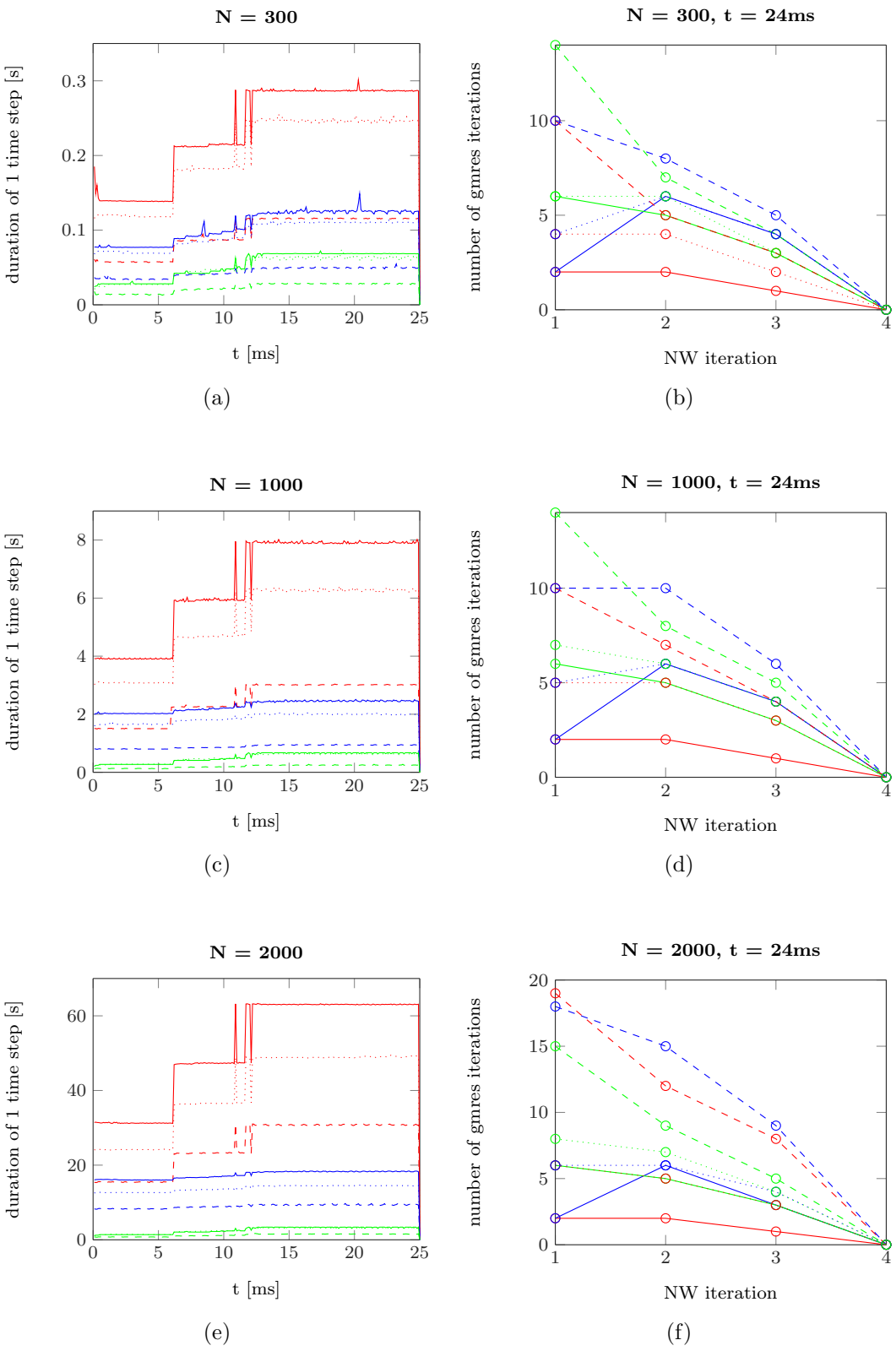


Figure 7.6: **Comparison of *positions* 1, 2, 3 and *ilu types* 1, 2, 4 of the ilu factorization from the point of view of the GMRES duration; *position* 1 (red), *position* 2 (blue), *position* 3 (green); *ilu type* 1 (solid line), *ilu type* 2 (dotted line), *ilu type* 4 (dashed line).** (a), (c), (e) show the duration of each time step from 0 to 25 ms, (b), (d), (f) show the number of GMRES iterations for each Newton iteration at the fixed time $t = 24$ ms.

Discussion and conclusions

Let us first look at different *ilu types* (line styles) for a given *position* (color) in Figure 7.6. Figures 7.6b, 7.6d and 7.6f show that the *ilu type* 4 induces generally the largest number of iterations, while the *ilu type* 2 less and the *ilu type* 1 the least number of GMRES iterations. This result has been expected, as we state in the introductory part. Another result comes when observing the duration of a time step in Figures 7.6a, 7.6c and 7.6e. Data corresponding to the *position* 3 - where the duration of a time step does not include the duration of an `ilu` process - show that even though the *ilu type* 4 induces the most of GMRES iterations, these are much less time-consuming than in the case of the other two *ilu types*. This confirms our initial hypothesis that the *ilu type* 4 produces sparse but not too reliable preconditioning matrices. The new information is that at the conditions that are set in this experiment, the *ilu type* 4 induces such number of GMRES iterations that it still holds the post of the most rapid method. Another information can be extracted from the data plotted in 7.6a, 7.6c and 7.6e. The data corresponding to the *position* 1 reveal the approximate duration of the `ilu` computation since this arises at each Newton iteration. The jumps in the curves represent therefore the occurence of a new Newton iteration with the `ilu` computation. It shows that `ilu` in the *ilu type* 4 is the least, in the *ilu type* 2 more and in the *ilu type* 1 the most time-consuming operation. This result also confirms our expectations. Let us now look at different *positions* for a fixed *ilu type*. Figures 7.6b, 7.6d and 7.6f show that in the first Newton iteration the *positions* 1 and 2 induce the same number of GMRES iterations and this is less than the number of iterations induced by the *position* 3 (except for the *ilu type* 4 in N = 2000). This result has been well expected as well as the behaviour of the curves in the remaining Newton iterations. The most remarkable point is related to the behaviour of the *position* 3 since the number of GMRES iterations is in this case not very different from the number of GMRES iterations corresponding to other two *positions* (particularly at the second Newton iteration, the green data pass below the blue data for a fixed line type). This confirms that in the conditions set in our experiment (particularly the `h_t` value) the `ilu` of the `Matrix` at the *position* 3 gives a relatively reliable preconditioning matrices for the GMRES. Figures 7.6a, 7.6c and 7.6e show primarily that in the range of the GMRES iterations numbers that occur in our experiment, the `ilu` computation is a more time-consuming operation than the GMRES process. This fact is more significant for bigger N where we see a splitting of curves to the red above the blue above the green ones. The most effective *position* therefore shows to be the *position* 3 that induces more GMRES iterations than the other ones but this number is in the current setting so low that the GMRES process is less time-consuming than the `ilu` computation.

Experiment 3C - 1, 2 and 4 *ilu type* test for the *position* 3

In the previous experiment we showed that the option where the `ilu` is computed from `A` only once before the time loop and the resulting matrices are used as split preconditioners for GMRES at each Newton iteration is a very efficient method to use at least for the initial parameters that we have set. In this experiment we fix the *position* 3 of `ilu` and we test the three *ilu types* 1, 2 and 4 that we used also previously. Now, we will extend the initial parameters, namely the range of N from 300 up to 5000. For N = 300, 1000 and 2000 the data are plotted in Figure 7.6. Figure 7.6 shows that the *ilu type* 3, even though it induces more GMRES iterations than the other two *ilu types*, causes the fastest GMRES computation. This is caused by the sparsity of the preconditioning matrices that are generated by `ilu`. The *ilu type* 2 induces less GMRES iterations and a greater GMRES duration since the produced preconditioning matrices are less sparse and more reliable. The *ilu type* 1 induces the smallest number of GMRES iterations and causes approximately the same GMRES duration as the method 2. This method produces the less sparse but the most reliable matrices. The described results are valid for the three chosen values of N. Let us make an experiment to see if this situation remains the same also for a grater N.

Initialization

We run a series of simulations with the initial parameters:

```
A = 90;         % [dB]
f = 1000;         % [Hz]
h_t = 0.1;         % [ms]
N = [300,1000,1500,2000,2500,3000,3500,4000,4500,5000];
T = 0.025;         % [s]
t_save = 24  % [ms]
tol_Newton = 1e-4;
tol_gmres = 1e-6;
position = 3;
        % 3 - once before the time loop
method = [1,2,4];
% 1  [L,U] = ilu(sparse(Matrix),struct('type','ilutp','droptol',1e-6));
% 2  [L,U] = ilu(sparse(Matrix),struct('type','ilutp','droptol',1e-4));
% 4  [L,U] = ilu(sparse(Matrix),struct('type','nofill'));

%----GMRES comand in cochleaSolverCN_nonlin_M2.m--
[y,~] = gmres(B,d,[],tol_gmres,200,L,U,z1);
```

and we save the time step duration, the GMRES duration and the number of GMRES iterations at each Newton iteration at the time `t_save`.

Results

The results are summarized in Figure 7.7 and 7.8.

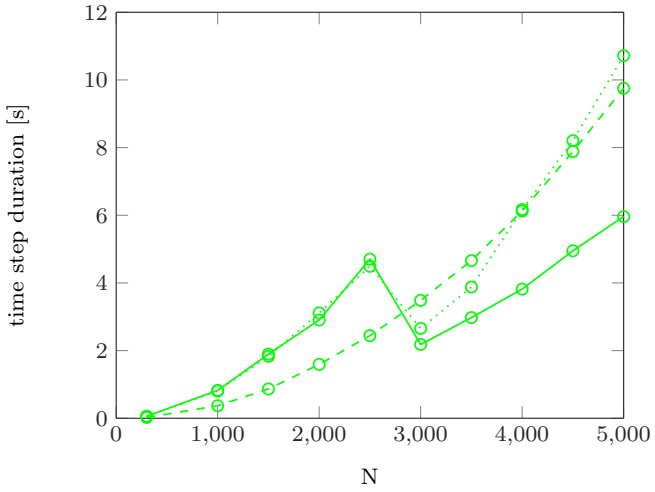
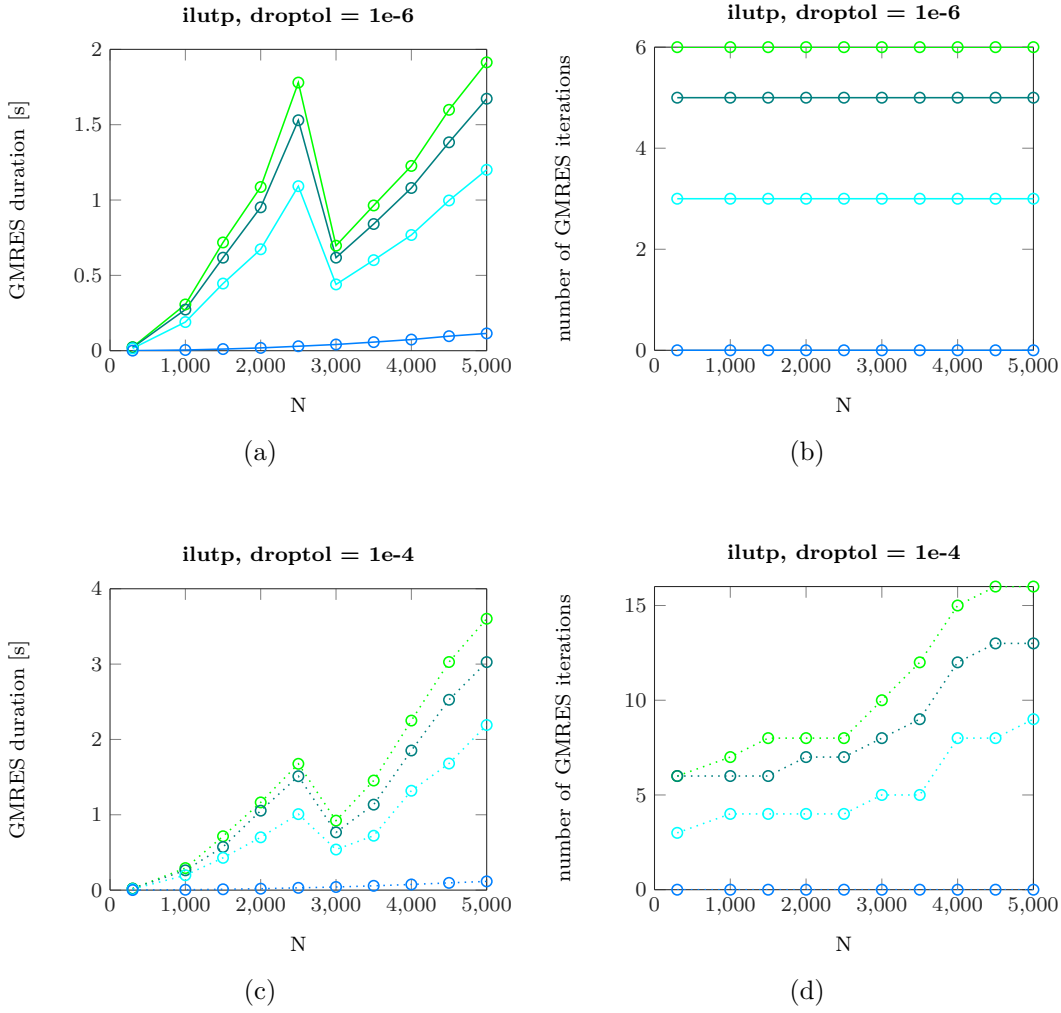
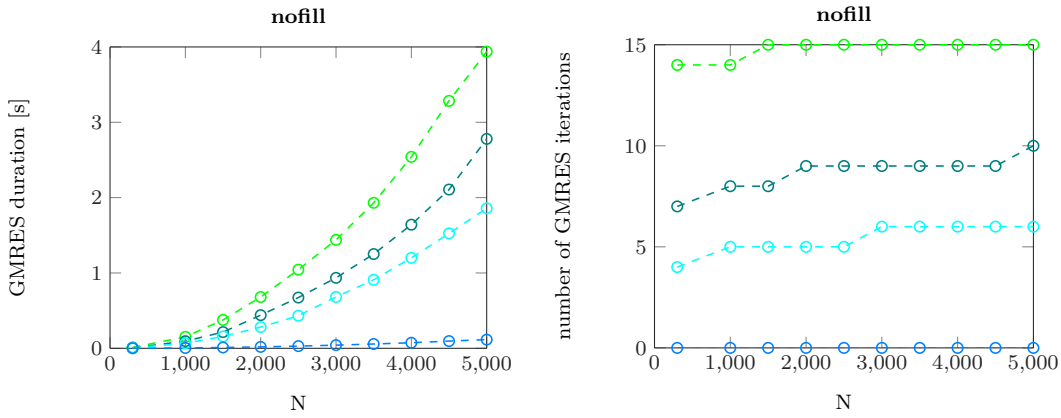


Figure 7.7: **The time step duration at 24 ms for different N, the *position* 3 and the *ilu types* 1 (solid line), 2 (dotted line), 4 (dashed line).**





(e) (f)

Figure 7.8: **GMRES duration and number of GMRES iterations as a function of N, for each Newton iteration: 1 (light green), 2 (dark green), 3 (cyan), 4 (blue) and for different *ilu types*: 1 (a)(b); 2 (c)(d); 4 (e)(f).**

Discussion and conclusions

Figure 7.7 shows that in the range of N from 0 to 2500 the *ilu types* 1 and 2 induce similar time step duration while the *ilu type* 4 induces a lower time step duration. The method 4 is therefore the fastest method in this range of N. Starting from $N = 3000$ the situation changes rapidly. The time step duration becomes the lowest for the *ilu type* 1 and the biggest for the *ilu type* 3. The latter one switches its role with the *ilu type* 2 from $N = 4000$. The *ilu type* 1 remains the fastest method up to $N = 5000$. The abrupt change in the behaviour of the curves for the *ilu types* 1 and 2 between $N = 2500$ and $N = 3000$ is unexpected. To see better the behaviour of the *ilu types*, we extracted from the time step duration only the GMRES durations at each Newton iteration and we plotted them in Figure 7.8. Figure 7.8b shows that the number of GMRES iterations induced by the *ilu type* 1 remains the same for each number of N and at each Newton iteration. We suppose therefore that the drop between $N = 2500$ and $N = 3000$ is caused by the change in the structure of the preconditioning matrices. We suppose that increasing N from 2500 to 3000, a set of matrix elements becomes so small that they are excluded by the dropping criterion of the method. A similar situation arises in the case of the *ilu type* 2. The principal difference from the *ilu type* 1 consists in the continuously increasing number of the GMRES iterations with the increasing N. This causes that the drop in the GMRES duration between $N = 2500$ and $N = 3000$ is lower than in the case of the *ilu type* 1. Figure 7.8d shows moreover that the increase of the number of GMRES iterations is sharper from $N = 3000$ than to this value. This is supposed to be the reason of the sharp increase of the GMRES duration from $N = 3000$. The behaviour of the *ilu type* 4 is much different from the previous two. The GMRES duration increases continuously from $N = 300$ to $N = 5000$ while the number of GMRES iterations changes only a little. This can be well explained by the

increase in the size of $\mathbf{A}$ with the growing N.

Figure 7.8 shows that the fourth Newton iteration for all of the three *ilu types* exhibits no GMRES iteration but a nonzero GMRES duration. This is related to the situation where the error of the Newton method is greater than the predefined tolerance but the initial relative residual of the GMRES method is lower than the GMRES predefined tolerance. The GMRES duration is therefore given only by the duration of an initial computation of `gmres` before the iterations start.

The reason of the behaviour of *ilu types* 1 and 2 with the change of N from 2500 to 3000 remains unclear and should be studied in detail. For future uses we will opt for the *ilu type* 4.

7.2.2 Experiment 4: classical vs. fixed Newton

In the Chapter 4 we stated that there exists another version of the Newton method, the fixed Newton method, that omits the recalculation of the matrix $\mathbf{B_k^i}$ at each iteration. The matrix $\mathbf{B_k}$ is calculated only once in the time step and this is before the Newton iteration loop. We will test this method to see if this could be a good candidate to an effective method solving our equations. In the classical Newton method we calculate the matrix $\mathbf{B_k^i}$ at each iteration and then we use the GMRES iterative method to find the approximate solution of the newly formed linear problem. The implementation of the fixed Newton is realized differently. Since we compute the solution of the linear problem with the same matrix at each iteration of a time step, we do not opt for an iterative solver. Instead, we calculate the LU decomposition of $\mathbf{B_k}$ once in a time step and then we compute the approximate solution at each iteration using one forward and one backward substitution. The corresponding script `cochleaSolverCN_nonlin_M2_fixed.m` with the implemented fixed Newton method can be found in Appendix.

Initialization

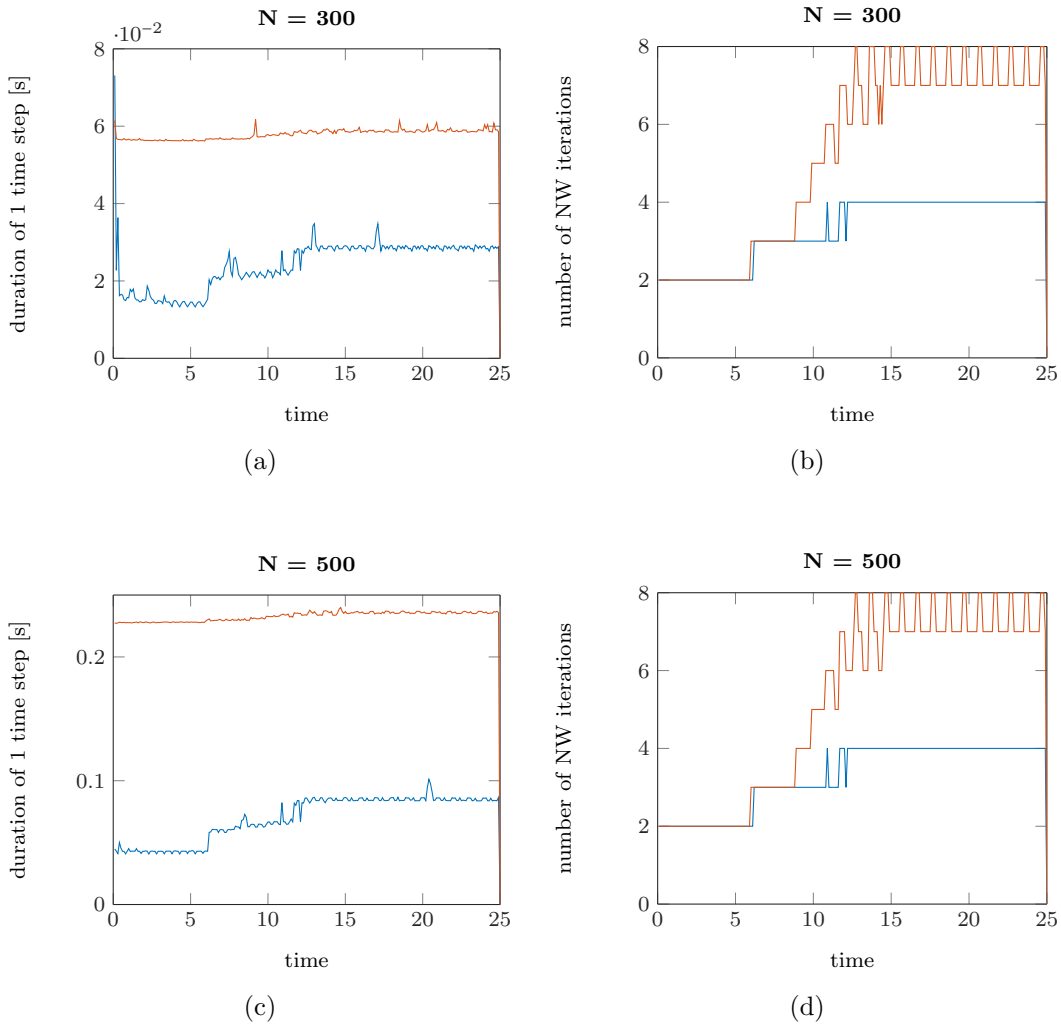
We run a series of simulations with the parameters

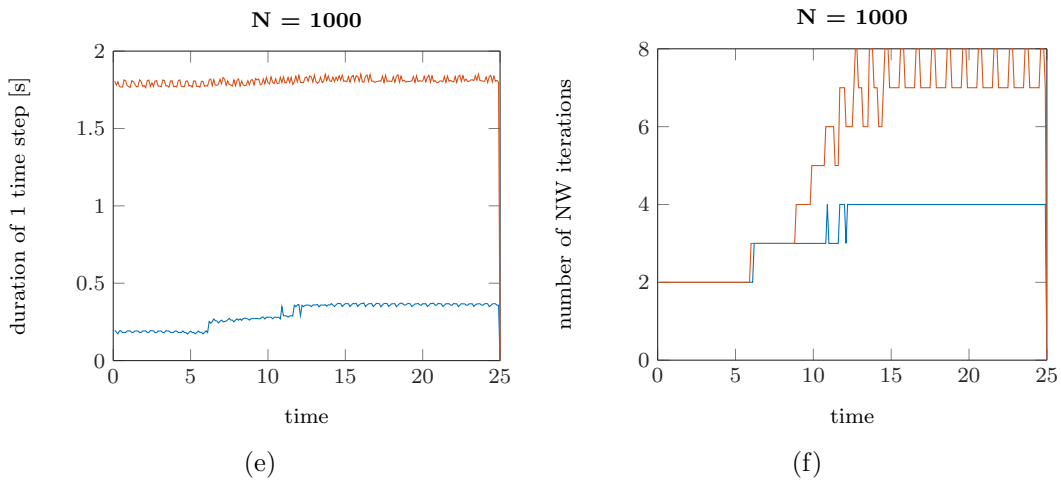
```
A = 90;        % [dB]
f = 1000;        % [Hz]
h_t = 0.1;        % [ms]
N = [300,500,1000];
T = 0.025;        %[s]
method = ['classical','fixed']
        % 'classical' - classical Newton
        % 'fixed' - fixed Newton
tol_Newton = 1e-4;        % tolerance for the classical Newton method
tol_gmres = 1e-6;        % gmres tolerance
tol_fixedNewton = 1e-4;        % tolerance for the fixed Newton method
```

and we save the time step duration and the number of Newton iterations at each time step.

Results

The results are plotted in Figure 7.9.





(e) (f)

Figure 7.9: **Classical (blue) versus fixed (red) Newton method.** NW iteration in (b), (d), (f) means Newton iteration.

Discussion and conclusions

Figures 7.9b, 7.9d and 7.9f show that the number of Newton iterations for the fixed Newton method is equal to (mainly in the linear regime) or greater than (in the nonlinear regime) for the classical Newton. We observe that this behaviour is the same for all of the three situations with different N and confirms our hypothesis about the slower convergence of the fixed Newton from the introductory part. Let us now extend our focus to Figures 7.9a, 7.9c and 7.9e. We see (mainly in Figure 7.9a) that while the blue curve exhibits visible jumps at the times when there is an increase in the number of iterations, there is no such behaviour of the red curve. In other words, while the dependence of the time step duration on the time for the classical Newton follows the shape of the number of iterations dependence, for the fixed Newton this does not occur. The dependence in this case exhibits only small variations that are much less significant than in the case of the classical method. At the same time, the time step duration for each value of N and for each time is smaller in the classical method than in the fixed method and the difference between them increases with the N increasing. Our interpretation of this behaviour is the following. The most expensive operation in the fixed Newton method is the `lu` decomposition. Each iteration is then very fast, hence the time step duration is practically equal to the duration of the `lu` process. This causes that the duration of the time step for this method is almost constant in Figures 7.9a, 7.9c and 7.9e. The fact that this dependence accepts bigger values when we increase N is due to the increasing cost of the `lu` process with the increasing size of the decomposed matrix. The classical Newton method behaves differently. The time step duration is almost entirely due to the GMRES process duration. Hence increasing the number of iterations we get an appropriately increased time step duration. Based on Figure 7.9 we conclude that for the given tolerance `tol` the classical Newton is

a faster method than the fixed Newton. The number of Newton iterations is indeed so small that their summed duration does not exceed the duration of the `lu` process performed in the fixed Newton. We showed that this result is valid for the number N of 300, 500 and 1000. The duration of the `lu` process is however generally more expensive for bigger values of N. As well, we do not expect that the number of iterations in the classical Newton would increase abruptly with increasing N. We suppose therefore that the classical Newton method is more effective also for larger values of N than the ones used in the current experiment.

7.3 Comparison of Method 1 and 2 with CN and final optimization

In the previous two sections we analysed various versions of Method 1 and 2 of the CN method from the point of view of their effectiveness. Here we will use these results to compose Method 1 and 2 so that they be effective for the currently set parameters and we will compare the final versions. First we will compare the number of iterations and the time step duration and then we will study their convergence with decreasing time step. The latter study will be extended to the four implicit methods that we take into account in this work: IE, CN, BDF2 and BDF3.

In this chapter we use the following versions of Method 1 and 2:

Method 1	Version A - without extrapolation
Method 2	preconditioner choice: *position* 3, *ilu type* 4 - before the time loop and the *nofill* version Newton method choice: classical

7.3.1 Experiment 5: time step duration and number of iterations comparison

We compare the number of iterations and the time step duration at particular times for Method 1 and 2 with the CN algorithm. Based on the previous reasoning we suppose that Method 1 involves more iterations than Method 2. The time step duration cannot be estimated a priori.

Initialization

We run nine simulations with the following parameters:

```
A = [20,70,130];        % [dB]
f = [200,2000,15000];        % [Hz]
h_t = [0.5,0.05,0.005];        % [ms] time step for various frequencies
N = 1000;
T = 0.030;                %[s]
t_save = [5,12,30]; % [ms] time of the data save
method  = [1,2]; % method 1 or 2 for the CN algorithm
```

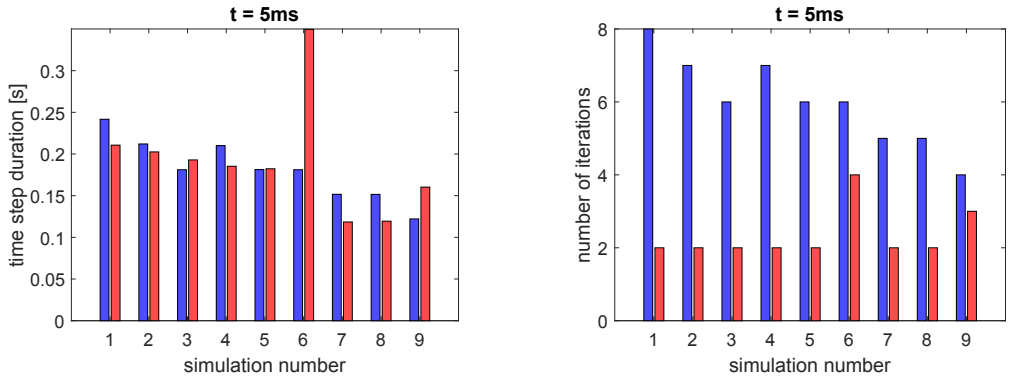
```
tol_method1 = 1e-4;        % tolerance for the method 1
tol_method2 = 1e-4;        % tolerance for the method 2
tol_gmres = 1e-6;

%--- we have 9 simulations -----
% 1: 200Hz, 20dB
% 2: 200Hz, 70dB
% 3: 200Hz, 130dB
% 4: 2000Hz, 20dB
% 5: 2000Hz, 70dB
% 6: 2000Hz, 130dB
% 7: 15000Hz, 20dB
% 8: 15000Hz, 70dB
% 9: 15000Hz, 130dB
%----------------------------------------
```

and we save the number of iterations N_{iter} and the time step duration t_{dT} at times specified by `t_save`. This is done for each of the nine simulations and for each of the two methods. The values of `t_save` are chosen such that we could observe the behaviour of the model in each experiment at various regimes. At 5 ms the 20 dB and 70 dB simulations are in the linear regime and the 130 dB simulation is in the nonlinear regime. At 12 ms, the 70 dB simulation as well as 130 dB are in the nonlinear regime while the 20 dB is still in the linear regime. At 30 ms the situation is the same as at 12 ms. We include this value into the `t_save` interval because the simulation is here already stabilized at the amplitude given by the decibel number.

Results

The results are summarized in Figure 7.10.



(a) (b)

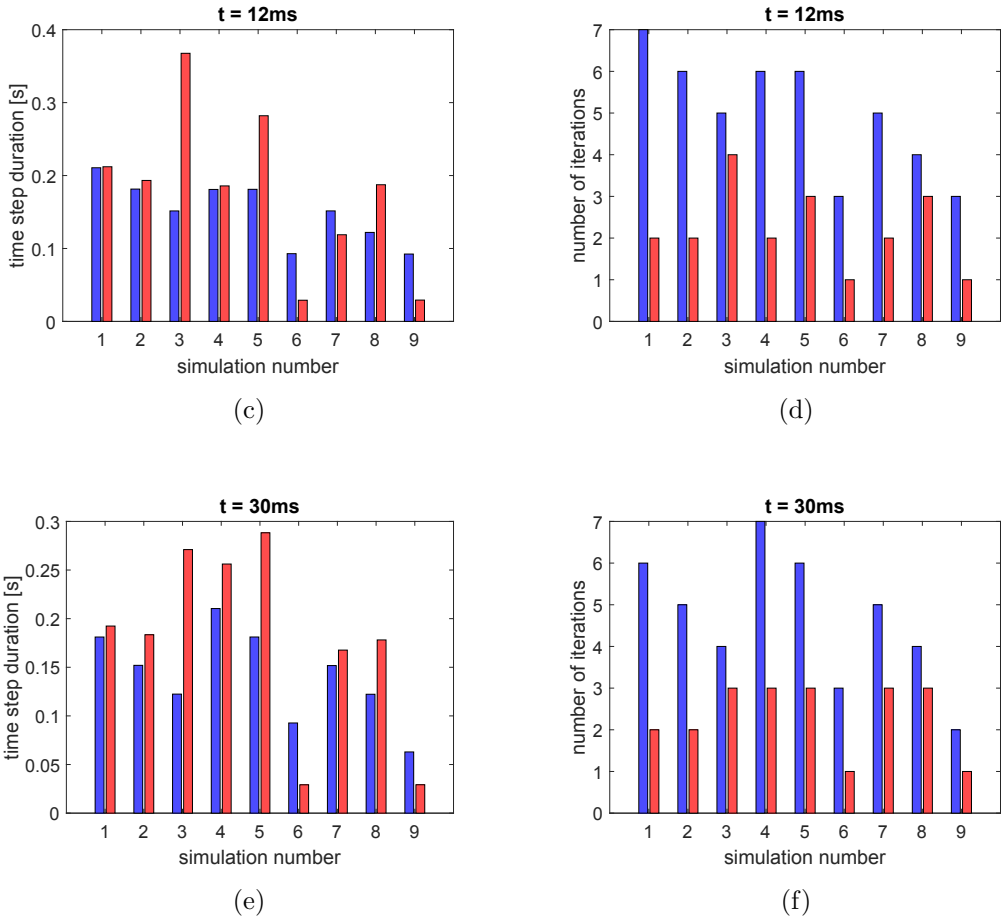


Figure 7.10: **Comparison of Method 1 (blue) and 2 (red) of CN algorithm at various times of the simulation.** Time step duration t_{dT} and number of iterations of the method N_{iter}. Simulation numbers: 1 - 200 Hz, 20 dB; 2 - 200 Hz, 70 dB; 3 - 200 Hz, 130 dB; 4 - 2000 Hz, 20 dB; 5 - 2000 Hz, 70 dB; 6 - 2000 Hz, 130 dB; 7 - 15000 Hz, 20 dB; 8 - 15000 Hz, 70 dB; 9 - 15000 Hz, 130 dB.

Discussion and conclusions

As we see in Figures 7.10b, 7.10d and 7.10f N_{iter} of Method 2 is in all cases smaller than N_{iter} of Method 1: while Method 2 involves $1-4$ iterations, Method 1 needs up to 8 iterations in some cases[4]. This observation approves our hypothesis: Method 1 converges slower than Method 2. Concerning t_{dT}, Figures 7.10a, 7.10c and 7.10e show that the two methods are in most cases very close each other. Especially the simulation 5 at 5 ms shows that six iterations of Method 1 and two iterations of the Method 2 have similar time length. The duration of a specific number of iterations of Method 2 may however differ from case to case. For example, the two iterations in the simulation 5 at 5 ms take approximately 0.06 s longer time than the two iterations in the simulation 8 at 5 ms. This is due to the GMRES process running at each iteration that may converge differently.

Let us now discuss the changes of N_{iter} in time (ie. for different values of `t_save`).

[4]By *case* we mean a particular simulation and a particular time value of 5 ms, 12 ms or 30 ms.

For 20 dB - ie. the simulations 1, 4, 7 - we do not see a particular trend with increasing time: N_{iter} takes values between 5 and 8 for Method 1 and between 2 and 3 for Method 2. For 70 dB and the frequencies 2000 Hz and 15000 Hz - ie. the simulations 5, 8 - we see an increase of N_{iter} for Method 2 when we pass from the linear regime of 5 ms to the nonlinear regime of 12 ms. The phenomenon of the increase of N_{iter} with increasing time of the simulation for the classical Newton method has been also observed in the Experiment 4, see Figure 7.9. Here we performed the experiment with the input amplitude of 90 dB that has the threshold time for the linear regime similar to our case of 70 dB. We can conclude therefore that there is an increase in N_{iter} for Method 2 with the passage from the linear to the nonlinear regime. This point is however not valid in the simulations 6 and 9 - with 130 dB, 2000 Hz and 130 dB, 15000 Hz. In these two cases, since we are in the nonlinear regime at all three values of `t_save`, we would expect a high number of iterations for Method 2. Figure 7.10 shows that this is true for 5 ms but not for 12 ms and 30 ms. Here N_{iter} falls to the number 1. To understand what happens in these cases, we need to look at the particular solutions of the simulations. This will be performed in the next experiment.

7.3.2 Experiment 6: convergence of the BM displacement computed by implicit numerical methods with decreasing time step

In the preceding experiments we studied the effectiveness, i.e. especially the time step duration of the two available methods Method 1 and Method 2. Up to now we did not however study the accuracy of these methods. In the following experiment we will compare the solutions obtained by the two methods with a *reference solution*. We will extend our study to all the four implicit methods that we use in our work: IE, CN, BDF2 and BDF3.

Initialization

We implement Method 1 and Method 2 to IE, BDF2 and BDF3 in the same manner as to the CN algorithm, see scripts `cochleaSolverIE_nonlin_M1.m`, `cochleaSolverIE_nonlin_M2.m`, `cochleaSolverBDF2_nonlin_M1.m`, `cochleaSolverBDF_nonlin_M2.m`, `cochleaSolverBDF3_nonlin_M1.m` and `cochleaSolverBDF3_nonlin_M2.m`. We run a series of simulations with the initial parameters:

```
A = [20,70,130];        % [dB]
f = [200,2000,15000];        % [Hz]
h_t_200Hz = [0.5,0.1,0.05,0.01];               %[ms] time steps for 200Hz
h_t_2000Hz = [0.05,0.01,0.005,0.001];          %[ms] time steps for 2000Hz
h_t_15000Hz = [0.005,0.001,0.0005,0.0001]; %[ms] time steps for 15000Hz
t_save = 25;                           % [ms] time of the results saving
N = 1000;
T = 0.030;        %[s]
```

```
method  = [1,2];          % Method 1 or 2
algorithm = ['IE','CN','BDF2','BDF3']; % implicit numerical method
tol_method1 = 1e-4;        % tolerance for the method 1
tol_method2 = 1e-4;        % tolerance for the method 2
tol_gmres = 1e-6;
```

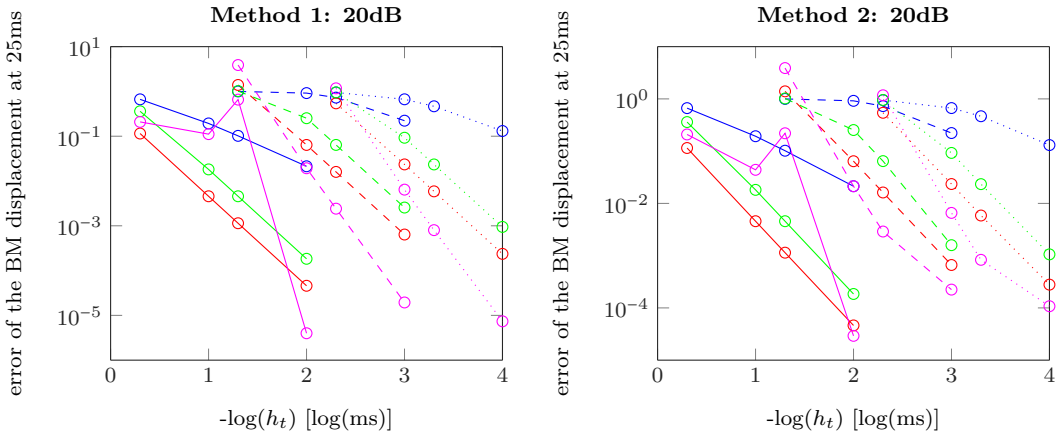
and we save the first N elements of the solution Z_{k+1} at the time given by `t_save`. These elements correspond to the BM displacement at the this instant and we denote this vector by U. Then we generate the *reference solution* with the use of the explicit RK6 method and with the following parameters:

```
N = 1000;                % number of discretization points in space
A = [20,70,130];                % [dB]
f = [200,2000,15000];        % [Hz]
h_t = 0.0001                 % [ms] time step
method = RK6;                 % used numerical method
t_save = 25;   % [ms] time of results saving
```

We save again the solution corresponding to the BM displacement at the time `t_save` and we denote it by U_{ref}.

Results

We plot the relative error of the BM displacement at the instant `t_save`, defined by $\|U - U_{ref}\|/\|U_{ref}\|$ against minus the decimal logarithm of the time step. We use the minus sign in the scale of the x axis because we want that the time step decreases from the left to the right. This plot is repeated for each value of `A`, `f`, `method` and `algorithm`. The results are represented in Figure 7.11.



(a) (b)

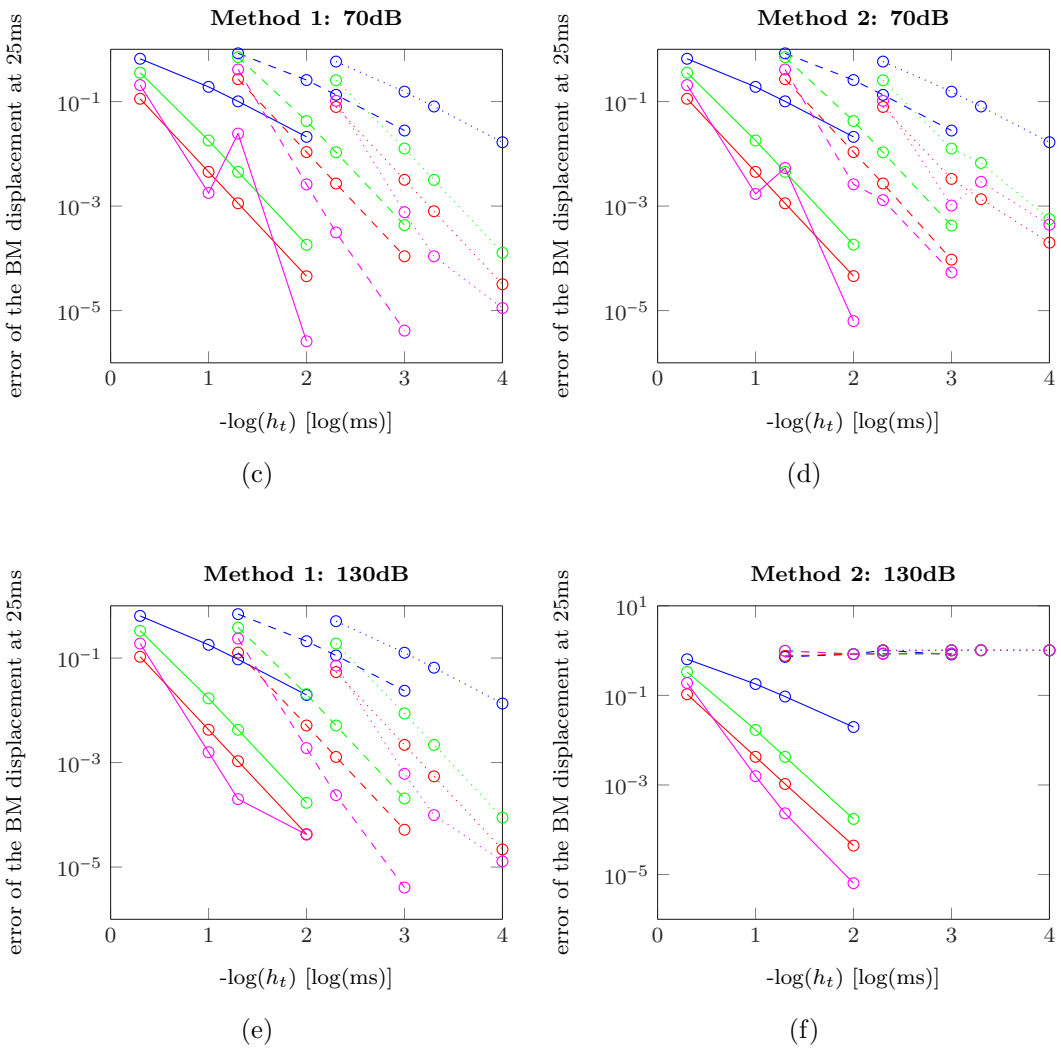


Figure 7.11: **Convergence of the BM displacement computed by IE (blue), CN (red), BDF2 (green) and BDF3 (magenta) with decreasing time step; input frequency of 200 Hz (solid line), 2000 Hz (dashed line), 15000 Hz (dotted line)**

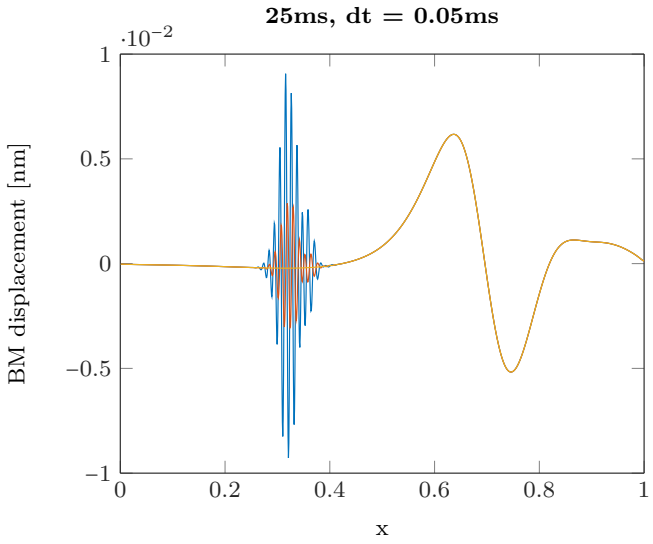


Figure 7.12: **oscillations of the BM displacement at 25 ms computed by BDF3 algorithm at 20dB and 200Hz. Method 1 (blue), Method 2 (red) and RK6 (yellow)**

Discussion and conclusions

Figure 7.11 shows that in most cases of amplitudes and frequencies the rate of convergence of the BM displacement computed by a numerical method to the *reference solution* is corresponding to the order of the method. The BDF3 converges in the fastest way, CN and BDF2 converge similarly but more slowly than BDF3 and IE converges the most slowly. Two significant discrepancies are visible in Figure 7.11. The first one concerns BDF3 mainly at 200 Hz. The convergence curve exhibits severe jumps to large values mainly at the time step 0.05 ms. This behaviour is related to the occurence of oscillations in the solution U and it is represented in Figure 7.12. This Figure shows the BM displacement at 25 ms. Further simulations showed that the oscillations arise already at the beginning of the computation and that they are damped completely after a period of time. The second discrepancy concerns the convergence of the BM displacement for Method 2 at 130 dB. Figure 7.11f shows that at 2000 Hz and 15000 Hz, methods do not converge to the reference solution. Further computations show that this is related to a specific behaviour: up to approximately 15 ms the computation proceeds in a standard way and then the solution becomes constant in time up to the end of the computation (the solution is frozen in time). This phenomenon explains the fall of the number of iterations for Method 2 at 130 dB to 1 in the Experiment 5. The Newton method indeed finds the solution at the specified tolerance already in the first iteration. Even after decreasing the GMRES tolerance to the value 10^{-20} the same behaviour occurs. The discrepancy points out that the Method 2 may fail at the nonlinear regime.

We conclude that the CN algorithm with Method 1 is the best for our uses since it

converges well to the reference solution in all cases and at the same time, it computes the BM displacement not very deviated from the reference solution also for big time steps.

7.3.3 Experiment 7: tolerance adjustment for CN with Method 1

In the previous experiment we concluded that the most reasonable choice of the implicit numerical method to implement into our program is CN with Method 1. Up to now we used the tolerance value in the stopping criterion of this method equal to 10^{-4}. This value may however be unnecessarily small and may therefore induce more iterations than is really needed. In the following experiment we will adjust this value and this will be done by watching the changes in approximate solutions obtained at each iteration.

Initialization

We run two simulations with the following initial parameters:

```
A = [20,130];         % [dB]
f = 2000;         % [Hz]
h_t = 0.05          %[ms]
t_save = 25;           % [ms] time of the results saving
N = 1000;
T = 0.030;         %[s]
method  = 1;          % method 1 or 2
algorithm = 'CN';   % implicit method
tol_method1 = 1e-4;           % tolerance for the method 1
```

and we save the elements `1:N` of the approximate solution Z_k^i, denoted by U^i, at each iteration i and at the time level k corresponding to the time `t_save`. These elements of the solution vector represent the calculated BM displacement. At the same time we save the value of the relative error of the approximate solution $||Z_k^i - Z_k^{i-1}||/||Z_k^i||$ at each iteration i.

Results

BM displacements U^i and relative errors of the approximate solutions for various iterations i for 20 dB and 130 dB are represented in Figure 7.13.

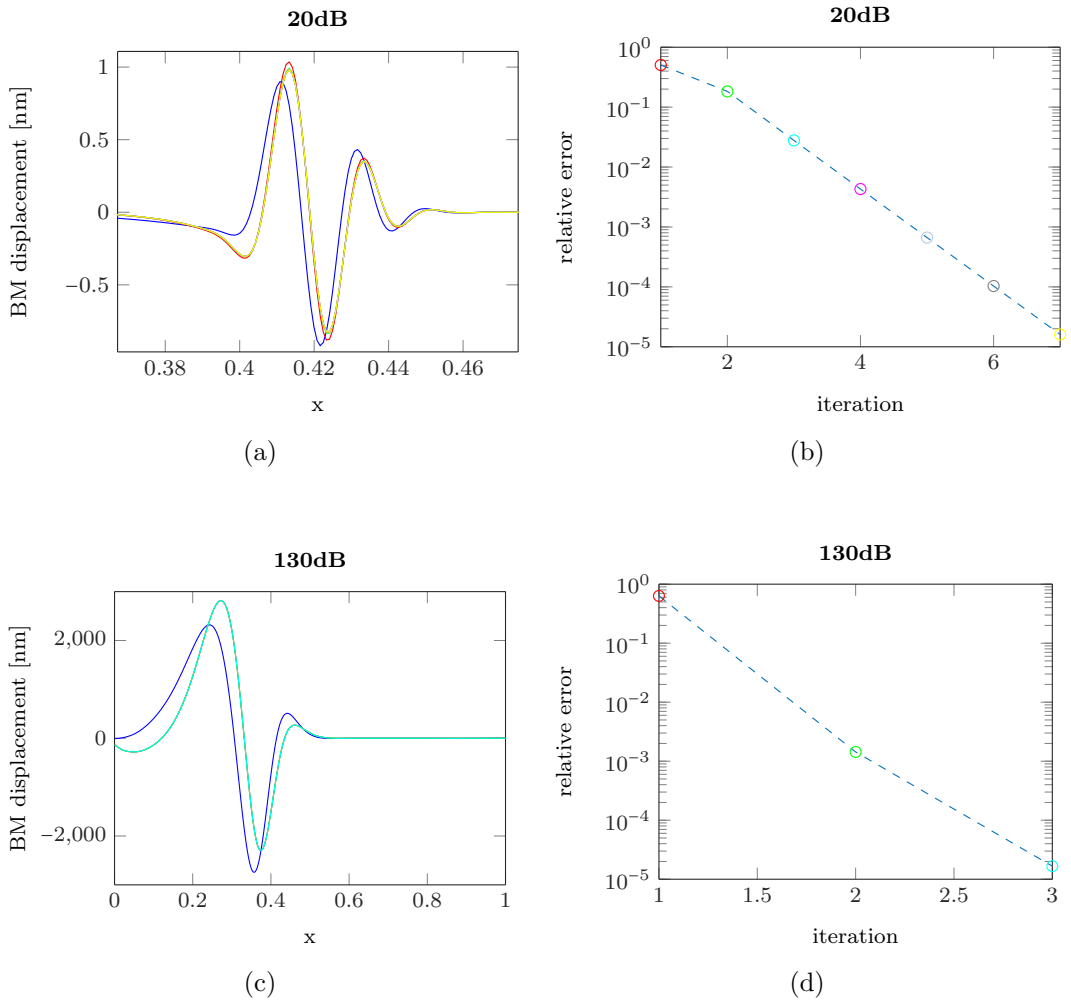


Figure 7.13: **Convergence of BM displacement approximate solutions at 25 ms for CN algorithm with Method 1.** (a) and (c) show the BM displacement U^i for each iteration i as the function of the spatial coordinate x; in (a) the plot is zoomed to the region of the curve that varies the most with increasing iteration number. The remaining portions of the curves are close to zero and vary only a little; (b) and (d) represent the relative error $\|Z_k^i - Z_k^{i-1}\|/\|Z_k^i\|$ of the approximate solution Z_k^i at each iteration i. Each color corresponds to a particular iteration. The zero-th iteration, i.e. the first guess of the solution, is marked by blue, the first iteration is marked by red and then green, cyan, magenta, grey, black and yellow.

Discussion and conclusions

Figures 7.13a and 7.13c show that starting from the first iteration, the BM displacement change by very small amounts and primarily in the extrema portions of the curves. While the system in the linear regime (i.e. at 20 dB) needs seven iterations to attain the desired tolerance, the system in the nonlinear regime (at 130 dB) needs only three. The tolerance of the method can be adjusted to a bigger value, for example the value of 10^{-2} should give a satisfactorily good result.

7.4 Interpolation to 3000 equidistant points

As we stated in the Goal of our work, we aim to express the solution of our system (particularly the BM displacement) at 3000 points equidistantly distributed in the interval [0,1]. In the program we work with the nonequidistant distribution of discretizing points and these are given by the `gaussgrid` function (see in Appendix). The method that we will use to get the solution in equidistant points is the interpolation by cubic spline, ie by a piecewise polynomial function where the polynomials are of the order three and the function has continuous first and second derivatives in all internal nodes. To see how the solution at 3000 equidistant points changes with the number of spatial discretizing points N we make the following experiment.

7.4.1 Experiment 8: Convergence of the BM displacement with decreasing spatial step

Initialization

We run a series of simulations with the parameters

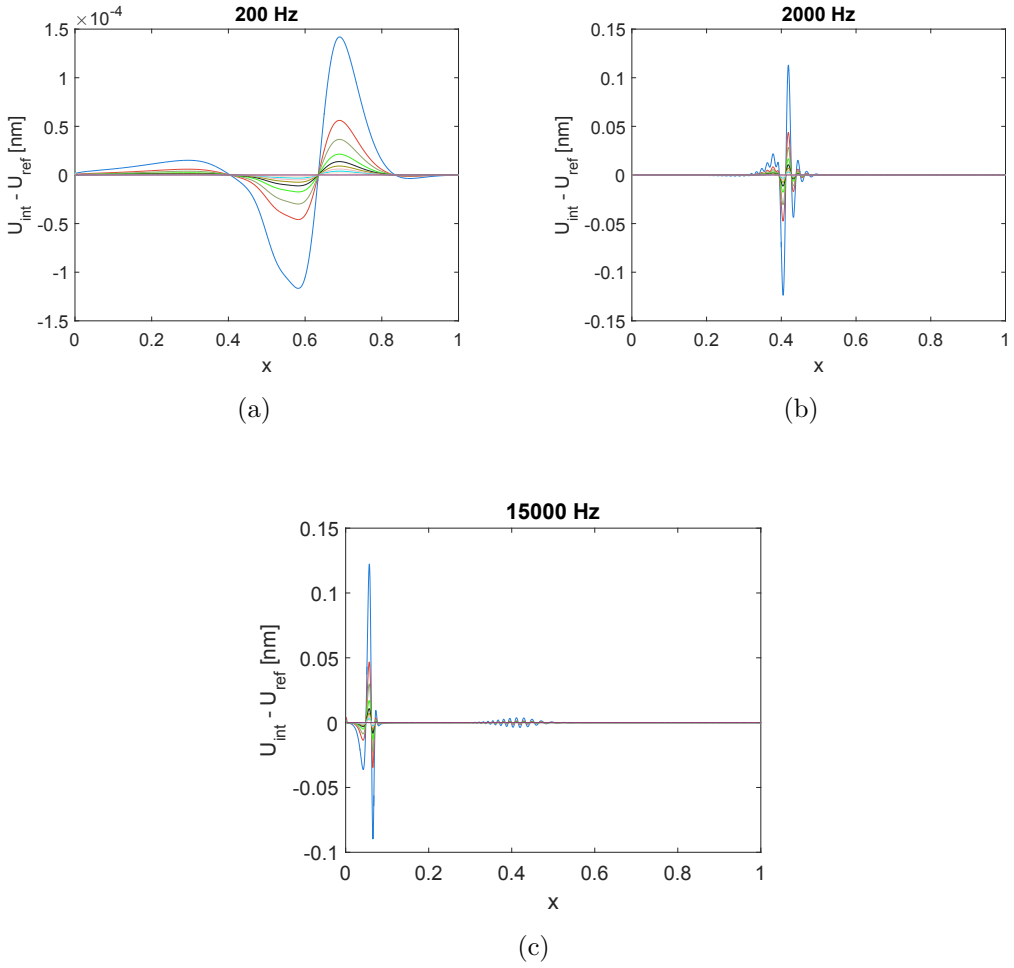
```
A = 90;                 % [dB]
f = [200,2000,15000];        % [Hz]
h_t_200Hz = 0.5;                %[ms]
h_t_2000Hz = 0.05;         %[ms]
h_t_15000Hz = 0.005;          %[ms]
N = [300,700,1000,1500,2000,2500,3000,3500,4000,5000];
t_save = [5,25];                   % [ms] time of the results saving
T = 0.030;        % [s]
method  = 1;
algorithm = 'CN'; % implicit method
tol_method1 = 1e-2;         % tolerance for the method 1
x_noneq = gaussgrid(N);         % nonequidistant discretization
x_eq = (1:3000)/3000;         % equidistant distribution in [0,1]
```

and we save the solution at the given by `t_save`. The values of `t_save` are chosen such that we could observe the behaviour of the system in the linear regime 5 ms and in the nonlinear regime 25 ms. We take the first N elements that correspond to the BM displacement and we interpolate them to 3000 equidistant points `x_eq` using `interp1` built-in MATLAB function with the option `'spline'` (for cubic spline) and `'extrap'` (to extrapolate at the end of the interval). The so-created solutions will be denoted by U_{int}. The solutions that have been generated with $N = 5000$ will be referred to as *reference solutions* and denoted by U_{ref}.

Results

We plot the differences $U_{int} - U_{ref}$ for each value of N in a separate figure for each `f` and `t_save`. The plots are displayed in Figures 7.14 and 7.15. Then we plot the relative error $\|U_{int} - U_{ref}\|/\|U_{ref}\|$ as a function of N for each frequency in Figure 7.16. Since

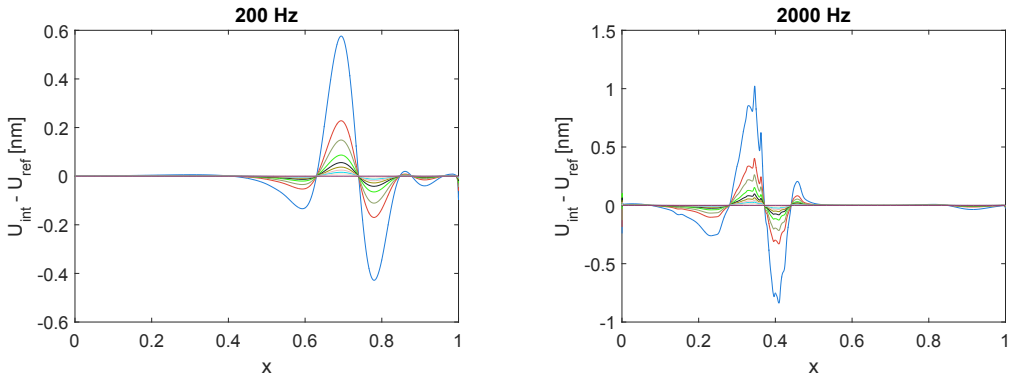
for $f = 15000\,\text{Hz}$ the U_{int} takes unreasonable large values at the apical end (mainly for 25 ms), we exclude the first 4 points from the calculation of the relative error at this frequency. As well, Figures 7.14c and 7.15c contain the plot only at `5:3000` equidistant points.



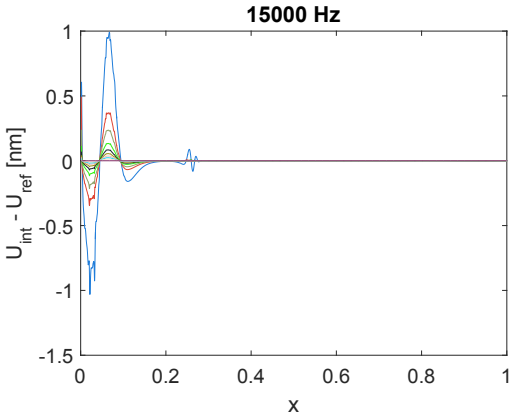
(a) (b)

(c)

Figure 7.14: **Difference of the interpolated solution at 3000 equidistant points in [0,1] from the reference solution at 5 ms (linear regime) and for various values of N.** Data corresponding to various values of N are plotted with different colour - see Figure 7.16a for the colour correspondance to N.

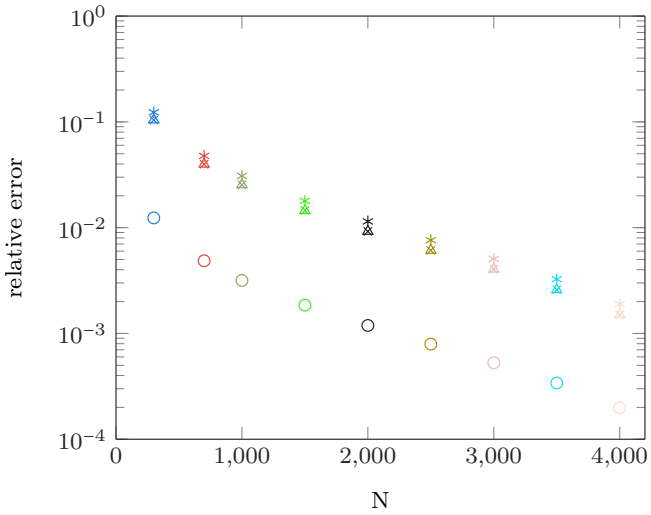


(a) (b)

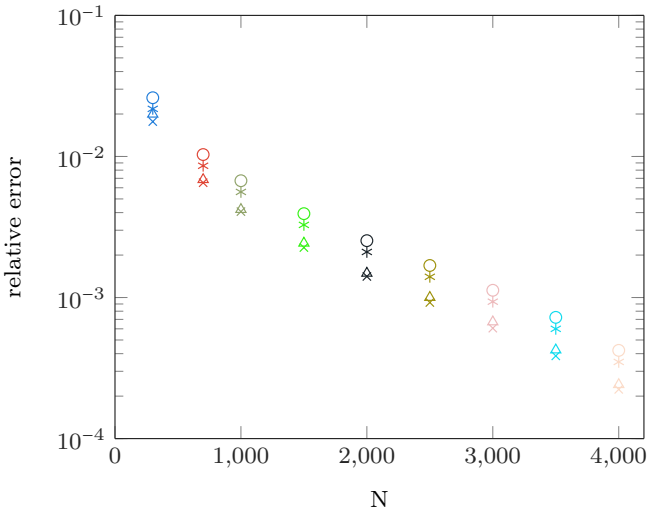


(c)

Figure 7.15: **Difference of the interpolated solution at 3000 equidistant points in [0,1] from the reference solution at 25 ms (nonlinear regime) and for various values of N.** Data corresponding to various values of N are plotted with different colour - see Figure 7.16b for the colour correspondance to N.



(a)



(b)

Figure 7.16: **Relative error of the interpolated solution at 5 ms (a) and at 25 ms (b) as a function of the number of spatial discretizing points N for various frequencies: 200 Hz (circles), 2000 Hz (stars), 15000 Hz (triangles); 15000 Hz without the most apical 4 points (crosses).**

Discussion and conclusions

Figure 7.16 shows that the interpolated BM displacement converges to the reference solution with increasing N. Comparing the two plots of 7.16, we see that the higher frequency simulations exhibit a larger relative error for 5 ms than for 25 ms. This may be partially caused by temporary oscillations arising around $x = 0.4$ at these frequencies. The main cause is that in the linear regime the activated region of the BM is narrower than in the nonlinear regime. The solution is here therefore more sensitive to the changes in the spatial discretization.

It depends on our objectives what value of N we should choose in future simulations. Big values of N assure the accuracy of the solution but cause that the simulation takes very long time. Simulations with low value of N run fast but not so accurately.

Conclusion

The main aim of our work was to implement a new numerical approach into the model, namely the implicit numerical methods. We selected four of the most commonly used implicit methods: implicit Euler (IE), Crank-Nicolson (CN), backward difference formula of the second order (BDF2) and of the third order (BDF3) and we tested them on the linear and the nonlinear versions of the model. In the linear case the implementation of these methods is straightforward. We conclude that the CN method convenes the most to our requirements since it combines the fast convergence rate for decreasing time step and the small sensitivity of the solution to the change in the time step (see Figure 6.1.) The use of implicit methods in the nonlinear case turned out to be much more difficult. This difficulty is enhanced by the specific shape of the nonlinear function: it exhibits a saturating property. This system leads to a system of nonlinear equations at each time step. We proposed two ways how to solve this system and we tested them from the the point of view of the efficiency and accuracy.

The two proposed versions of implicit methods are iterative and transform the nonlinear system to a linear one at each iteration. The first version simply hides the nonlinearity into the right hand side, hence each iteration is performed fast but the method converges slowly. The other version is more sophisticated: it uses the linear approximation of the nonlinear function at each iteration. This version computes each iteration slowly but needs very few iterations to satisfy the criterion of the numerical convergence. Our analysis showed that, for the initial parameters that we need in simulations, the first, simple version is more effective. Even though it involves more iterations, these are in sum less time-consuming than the few iterations of the second method. Moreover, the first version turned out to be more reliable in the convergence. While this one converged to the reference solution at each simulation, the second version failed to converge at high input amplitudes. This happened at each case of the two versions applied to the four implicit methods. Finally, we concluded that the first version of the CN method would be the best choice to solve our nonlinear problem.

With the winning method we performed the test of the sensitivity of the solution interpolated at 3000 equidistant point on the number of discretization points. We saw a clear convergence to the reference solution. It depends therefore on our requirement of the solution accuracy, what number of discretization points we opt for.

The result of our work is the implementation of an implicit numerical method that allows us to increase the number of discretization points to a practically arbitrary value, while preserving the stability of the computation. The interpolation of the solution to 3000 equidistant points gives us the values that serve as the direct input to the electrical model. The connection between the mechanical and the electrical models is therefore well established.

The work on the improvement of the connection between the mechanical and the electrical models is however not finished yet. There are several parts that need to be revised. As an example, the model involves an inconsistency concerning the probability of opening of mechanotransducer channels. This is in our model expressed as a function of the BM displacement (see section 3.2) and of the hair bundle displacement (see section 3.1). We aim to develop and to implement into the model a unified expression of this term. The removing of this inconsistency will thus be the next stage of the work on the model.

Bibliography

[1]. Hearing and the cochlea. Drugs.com; https://www.youtube.com/watch?v=P5gyQf2gVvA, lastly visited on 2/7/2017.

[2]. Cochlea. Encyclopædia Britannica, inc.; https://www.britannica.com/science/ear/Cochlea, lastly visited on 2/7/2017.

[3]. Organ of corti. Encyclopædia Britannica, inc.; https://www.britannica.com/science/organ-of-Corti, lastly visited on 2/7/2017.

[4]. lu. Matlab documentation; https://www.mathworks.com/help/matlab/ref/lu.html, lastly visited on 7/7/2017.

[5]. gmres. Matlab documentation; https://www.mathworks.com/help/matlab/ref/gmres.html, lastly visited on 7/7/2017.

Ascher, U. M., S. J. Ruuth, and R. J. Spiteri (1997). Implicit-explicit runge-kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics 25*(2-3), 151–167.

Ashmore, J. (1987). A fast motile response in guinea-pig outer hair cells: the cellular basis of the cochlear amplifier. *The Journal of Physiology 388*(1), 323–347.

Brownell, W. E., C. R. Bader, D. Bertrand, and Y. De Ribaupierre (1985). Evoked mechanical responses of isolated cochlear outer hair cells. *Science 227*, 194–197.

Bruce, I. C. (2007). Implementation issues in approximate methods for stochastic hodgkin–huxley models. *Annals of biomedical engineering 35*(2), 315–318.

Chow, C. C. and J. A. White (1996). Spontaneous action potentials due to channel fluctuations. *Biophysical journal 71*(6), 3013–3021.

Corey, D. and A. Hudspeth (1979). Response latency of vertebrate hair cells. *Biophysical journal 26*(3), 499–506.

Delprat, B. (2016). Composition of the cochlear fluids. http://www.cochlea.eu/en/cochlea/cochlear-fluids, lastly visited on 29/6/2017.

Dolejší, V. (2015). Numerical software. http://www.karlin.mff.cuni.cz/~dolejsi/Vyuka/NumSoft_15_11_16.pdf, lastly visited on 7/7/2017.

Dolejší, V. (2016). Fundamentals of numerical mathematics. http://www.karlin.mff.cuni.cz/~dolejsi/Vyuka/ZNM_16_06_22.pdf, lastly visited on 7/7/2017.

Eguia, M. C., G. C. Garcia, and S. A. Romano (2010). A biophysical model for modulation frequency encoding in the cochlear nucleus. *Journal of Physiology-Paris 104*(3), 118–127.

Elliott, S. J. and C. A. Shera (2012). The cochlea as a smart structure. *Smart Materials and Structures 21*(6), 064001.

Emadi, G., C.-P. Richter, and P. Dallos (2004). Stiffness of the gerbil basilar membrane: Radial and longitudinal variations. *Journal of Neurophysiology 91*(1), 474–488.

Fettiplace, R. (2006). Active hair bundle movements in auditory hair cells. *The Journal of physiology 576*(1), 29–36.

Fitzakerley, J. (2014). Equilibrium (or reversal) potentials. University of Minnesota Medical School Duluth, `http://www.d.umn.edu/~jfitzake/Lectures/DMED/IonChannelPhysiology/MembranePotentials/EquilibriumPotentials.html`, lastly visited on 29/6/2017.

Fox, R. F. (1997). Stochastic versions of the hodgkin-huxley equations. *Biophysical journal 72*(5), 2068–2074.

Georgescu, A., C. RocSoreanu, and N. Giurgiteanu (2012). Fitzhugh-nagumo model. *Bifurcation, Symmetry and Patterns*, 197.

Goutman, J. D. and E. Glowatzki (2007). Time course and calcium dependence of transmitter release at a single ribbon synapse. *Proceedings of the National Academy of Sciences 104*(41), 16341–16346.

Hibino, H., F. Nin, C. Tsuzuki, and Y. Kurachi (2010). How is the highly positive endocochlear potential formed? the specific architecture of the stria vascularis and the roles of the ion-transport apparatus. *Pflügers Archiv European Journal of Physiology 459*(4), 521–533.

Hodgkin, A. and A. Huxley (1952a). The components of membrane conductance in the giant axon of loligo. *The Journal of physiology 116*(4), 473–496.

Hodgkin, A. L. and A. F. Huxley (1952b). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology 117*(4), 500–544.

Johnstone, B. and P. Sellick (1972). The peripheral auditory apparatus. *Quarterly reviews of biophysics 5*(1), 1–57.

Kemp, D. T. (1978). Stimulated acoustic emissions from within the human auditory system. *The Journal of the Acoustical Society of America 64*(5), 1386–1391.

Liberman, M. C. and L. W. Dodds (1984). Single-neuron labeling and chronic cochlear pathology. iii. stereocilia damage and alterations of threshold tuning curves. *Hearing research 16*(1), 55–74.

Mammano, F. and J. F. Ashmore (1993). Reverse transduction measured in the isolated cochlea by laser michelson interferometry. *Nature 365*(6449), 838–841.

Mammano, F. and R. Nobili (1993). Biophysics of the cochlea: linear approximation. *The Journal of the Acoustical Society of America 93 6*, 3320–32.

Marcus, D. C., T. Wu, P. Wangemann, and P. Kofuji (2002). Kcnj10 (kir4. 1) potassium channel knockout abolishes endocochlear potential. *American Journal of Physiology-Cell Physiology 282*(2), C403–C407.

Markin, V. S. and A. Hudspeth (1995). Gating-spring models of mechanoelectrical transduction by hair cells of the internal ear. *Annual review of biophysics and biomolecular structure 24*(1), 59–83.

Meddis, R. (1986). Simulation of mechanical to neural transduction in the auditory receptor. *The Journal of the Acoustical Society of America 79*(3), 702–711.

Mino, H., J. T. Rubinstein, and J. A. White (2002). Comparison of algorithms for the simulation of action potentials with stochastic sodium channels. *Annals of biomedical engineering 30*(4), 578–587.

Mistrík, P. and J. F. Ashmore (2010). Reduced electromotility of outer hair cells associated with connexin-related forms of deafness: an in silico study of a cochlear network mechanism. *JARO-Journal of the Association for Research in Otolaryngology 11*(4), 559–571.

Mistrík, P., C. Mullaley, F. Mammano, and J. Ashmore (2009). Three-dimensional current flow in a large-scale model of the cochlea and the mechanism of amplification of sound. *Journal of The Royal Society Interface 6*(32), 279–291.

Nobili, R. and F. Mammano (1996). Biophysics of the cochlea ii: Stationary nonlinear phenomenology. *The Journal of the Acoustical Society of America 99*(4), 2244–2255.

Nobili, R., F. Mammano, and J. Ashmore (1998). How well do we understand the cochlea? *Trends in neurosciences 21*(4), 159–167.

Patuzzi, R. (1996). *Cochlear Micromechanics and Macromechanics*. New York, NY: Springer New York.

Pickles, J. (2012). *An Introduction to the Physiology of Hearing, Fourth Edition*. Bingley: Emerald Group.

Purves D, Augustine GJ, F. D. e. a. e. (2001). Hair cells and the mechanoelectrical transduction of sound waves. Available from: `https://www.ncbi.nlm.nih.gov/books/NBK10867/`.

Rubinstein, J. T. (1995). Threshold fluctuations in an n sodium channel model of the node of ranvier. *Biophysical journal 68*(3), 779–785.

Russell, I. J. and K. E. Nilsen (1997). The location of the cochlear amplifier: Spatial representation of a single tone on the guinea pig basilar membrane. *Proceedings of the National Academy of Sciences 94*(6), 2660–2664.

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems* (2nd ed.). Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Strassberg, A. F. and L. J. DeFelice (1993). Limitations of the hodgkin-huxley formalism: effects of single channel kinetics on transmembrane voltage dynamics. *Neural computation 5*(6), 843–855.

Sumner, C. J., E. A. Lopez-Poveda, L. P. O'Mard, and R. Meddis (2002). A revised model of the inner-hair cell and auditory-nerve complex. *The Journal of the Acoustical Society of America 111*(5), 2178–2188.

Tebbens, J. D., I. Hnetynková, M. Plešinger, Z. Strakoš, and P. Tichỳ (2012). Analỳza metod pro maticové vỳpocty: Základnı metody. *Prvnı vydánı. Matfyzpress, Praha*.

Ticháček, O. (2014). Computer modeling of the cochlea and the hearing nerve. *Bc thesis*.

v. Békésy, G. (1953). Description of some mechanical properties of the organ of corti. *The Journal of the Acoustical Society of America 25*(4), 770–785.

Vetešník, A. and A. W. Gummer (2012). Transmission of cochlear distortion products as slow waves: A comparison of experimental and model data. *The Journal of the Acoustical Society of America 131*(5), 3914–3934.

Von Békésy, G. and E. G. Wever (1960). *Experiments in hearing*, Volume 8. McGraw-Hill New York.

Zampini, V., S. L. Johnson, C. Franz, M. Knipper, M. C. Holley, J. Magistretti, G. Russo, W. Marcotti, and S. Masetto (2014). Fine tuning of cav1. 3 ca2+ channel properties in adult inner hair cells positioned in the most sensitive region of the gerbil cochlea. *PloS one 9*(11), e113750.

Zwislocki, J. J., S. C. Chamberlain, and N. B. Slepecky (1988). Tectorial membrane i: Static mechanical properties in vivo. *Hearing research 33*(3), 207–222.

Abbreviations

BM	basilar membrane
TM	tectorial membrane
CP	cochlear partition
CPo	characteristic position
MC	mechanotransducer channel
RL	reticular lamina
CF	characteristic frequency
NT	neurotransmitter
EE	explicit Euler method
RK6	6-stage explicit Runge-Kutta method
IE	implicit Euler method
CN	Crank-Nicolson method
BDF2	backward difference formula of the second order
BDF3	backward difference formula of the third order

Appendix: list of selected Matlab scripts

```
function [ t, y ] = cochleaSolverIE_lin( t, y, h, ...
                    N, L, U, P, in, AM, TMa, om, om2, fac  )

 t = t + h_t;
 gf  = g( t,N,in,TMa,AM,om,om2,fac );
 RHS = P*(gf*h_t+y);
 y = U\(L\RHS);

end


function [ t, y ] = cochleaSolverCN_lin( t, y, y_j, h_t, ...
                    N, L, U, P, in, AM, TMa, om, om2, fac  )

 gf1  = g( t,N,in,TMa,AM,om,om2,fac );
 t = t + h_t;
 gf2  = g( t,N,in,TMa,AM,om,om2,fac );
 RHS = P*(y+h_t/2*(gf1+gf2+y_j));
 y = U\(L\RHS);

end


function [ t, y ] = cochleaSolverBDF_2_lin( t, y2, y1, h_t, ...
                    N, L, U, P, in, AM, TMa, om, om2, fac)

t = t + h_t;
gf  = g( t,N,in,TMa,AM,om,om2,fac );
RHS = P*(h_t*gf+2*y2-1/2*y1);
y = U\(L\RHS);

end


function [ t, y ] = cochleaSolverBDF_3_lin( t, y3, y2, y1, h_t, ...
                    N, L, U, P, in, AM, TMa, om, om2, fac  )

 t = t + h_t;
 gf  = g( t,N,in,TMa,AM,om,om2,fac );
 RHS = P*(h_t*gf+3*y3-3/2*y2+1/3*y1);
 y = U\(L\RHS);

end


function [ t, y ] = cochleaSolverBDF_2_nonlin_M1( t, y_prev, y2,...
                    y1, h_t, N, L,U,P, in, AM, TMa, BMy, om, om2, fac)
```

```
48
49  iter = 0;
50  tol = 0.1;
51  error1 = tol+1;
52  t = t + h_t;
53  gf  = g( t,N,in,TMa,AM,om,om2,fac );
54  z1 = y_prev;
55    while error1 > tol
56            iter = iter+1;
57            RHS = P*(h_t*gf+2*y2-1/2*y1+...
58            h_t*nonlinearity(z1,N,TMa,BMy));
59            y = U\(L\RHS);
60            error1 = norm(y-z1)/norm(y);
61            z1 = y;
62    end
63
64  end
65
66
67  function [ t, y] = cochleaSolverBDF_2_nonlin_M2( t, y_prev, y2,...
68                y1, h_t, N, A, L, U, in, AM, TMa, BMy, om, om2, fac )
69
70  iter = 0;
71  tol_Newton = 0.1;
72  tol_gmres = 1e-4;
73  error1 = tol_Newton+1;
74  t=t+h_t;
75  gf  = g( t,N,in,TMa,AM,om,om2,fac );
76      i = [N+1:2*N, 3*N+1:4*N]';
77      j = [2*N+1:3*N, 2*N+1:3*N]';
78  z1 = y_prev;
79      while error1 > tol_Newton
80          iter = iter + 1;
81          [term1, term2] = nonlinterms(N,z1,TMa,BMy);
82          v = [term1;term2];
83
84          Dh_DY = sparse(i,j,v,4*N,4*N);
85          B = A-h_t*Dh_DY;
86
87          z1TMdispl = z1(2*N+1:3*N);
88          T1 = term1.*z1TMdispl;
89          T2 = term2.*z1TMdispl;
90       d = 2*y2-1/2*y1+h_t*gf+h_t*nonlinearity(z1,N,TMa,BMy) -...
91       h_t*[zeros(N,1);
92            T1;
93            zeros(N,1);
94            T2];
95          [y,~] = gmres(B,d,[],tol_gmres,100,L,U,z1);
96          error1 = norm(y-z1)/norm(y);
97          z1 = y;
98      end
99
100 end
```

```
101
102
103 function [ t, y ] = cochleaSolverIE_nonlin_M1( t, y_prev, y_mass,...
104                     h_t, N, L, U, P, in, AM, TMa, BMy, om, om2, fac  )
105
106  iter = 0;
107  tol = 0.01;
108  error1 = tol+1;
109  t = t + h_t;
110  gf  = g( t,N,in,TMa,AM,om,om2,fac );
111  z1 = y_prev;
112  while error1 > tol
113              iter = iter+1;
114              RHS = P*(gf*h_t+y_mass+h_t*nonlinearity(z1,N,TMa,BMy));
115              y = U\(L\RHS);
116              error1 = norm(y-z1)/norm(y);
117              z1 = y;
118  end
119
120 end
121
122
123 function [ t, y] = cochleaSolverIE_nonlin_M2( t, y_prev, y_mass,...
124                     h_t, N, A, L, U, in, AM, TMa, BMy, om, om2, fac )
125
126 iter = 0;
127 tol_Newton = 1e-4;
128 tol_gmres = 1e-6;
129 error1 = tol_Newton+1;
130 t=t+h_t;
131 gf  = g( t,N,in,TMa,AM,om,om2,fac );
132     i = [N+1:2*N, 3*N+1:4*N]';
133     j = [2*N+1:3*N, 2*N+1:3*N]';
134 z1 = y_prev;
135     while error1 > tol_Newton
136         iter = iter + 1;
137         [term1, term2] = nonlinterms(N,z1,TMa,BMy);
138         v = [term1;term2];
139
140         Dh_DY = sparse(i,j,v,4*N,4*N);
141         B = A-h_t*Dh_DY;
142
143         z1TMdispl = z1(2*N+1:3*N);
144         T1 = term1.*z1TMdispl;
145         T2 = term2.*z1TMdispl;
146     d = y_mass+h_t*gf+h_t*nonlinearity(z1,N,TMa,BMy) -...
147     h_t*[zeros(N,1);
148          T1;
149          zeros(N,1);
150          T2];
151          [y,~] = gmres(B,d,[],tol_gmres,100,L,U,z1);
152          error1 = norm(y-z1)/norm(y);
153          z1 = y;
```

```
154      end
155
156  end
157
158
159  function [ t, y ] = cochleaSolverCN_nonlin_M1( t, y_prev, y_mass,...
160                  y_j, h_t, N, L, U, P, in, AM, TMa, BMy, om, om2, fac )
161
162  iter = 0;
163  tol = 0.01;
164  error1 = tol+1;
165   gf1  = g( t,N,in,TMa,AM,om,om2,fac );
166   t = t + h_t;
167   gf2  = g( t,N,in,TMa,AM,om,om2,fac );
168   z1 = y_prev;
169   while error1 > tol
170                iter = iter+1;
171  RHS = P*(y_mass+h_t/2*(gf1+gf2+y_j+nonlinearity(y_prev,N,TMa,BMy))+
         h_t/2*nonlinearity(z1,N,TMa,BMy));
172                y = U\(L\RHS);
173                error1 = norm(y-z1)/norm(y);
174                z1 = y;
175   end
176  end
177
178
179  function [ t, y] = cochleaSolverCN_nonlin_M2( t, y_prev, y_mass,...
180                  y_j, h_t, N, A, L, U, in, AM, TMa, BMy, om, om2, fac )
181
182  iter = 0;
183  tol_Newton = 1e-4;
184  tol_gmres = 1e-6;
185  error1 = tol_Newton+1;
186   gf1  = g( t,N,in,TMa,AM,om,om2,fac );
187   t = t + h_t;
188   gf2  = g( t,N,in,TMa,AM,om,om2,fac );
189      i = [N+1:2*N, 3*N+1:4*N]';
190      j = [2*N+1:3*N, 2*N+1:3*N]';
191  z1 = y_prev;
192      while error1 > tol_Newton
193          iter = iter + 1;
194          [term1, term2] = nonlinterms(N,z1,TMa,BMy);
195          v = [term1;term2];
196
197          Dh_DY = sparse(i,j,v,4*N,4*N);
198          B = A-h_t/2*Dh_DY;
199
200          z1TMdispl = z1(2*N+1:3*N);
201          T1 = term1.*z1TMdispl;
202          T2 = term2.*z1TMdispl;
203      d = y_mass+h_t/2*(gf1+gf2+y_j+nonlinearity(y_prev,N,TMa,BMy))+...
204        h_t/2*nonlinearity(z1,N,TMa,BMy)-h_t/2*[zeros(N,1);
205                                                T1;
```

```
206                                                     zeros(N,1);
207                                                     T2];
208             [y,~] = gmres(B,d,[],tol_gmres,100,L,U,z1);
209             error1 = norm(y-z1)/norm(y);
210             z1 = y;
211         end
212
213     end
214
215
216     function [ t, y] = cochleaSolverCN_nonlin_M2_fixed( t, y_prev,...
217           y_mass, y_j, h_t, N, Matrix, in, AM, TMa, BMy, om, om2, fac)
218
219
220     iter = 0;
221     tol_Newton = 1e-4;
222     error1 = tol_Newton+1;
223     gf1  = g( t,N,in,TMa,AM,om,om2,fac );
224     t = t + h_t;
225     gf2  = g( t,N,in,TMa,AM,om,om2,fac );
226         i = [N+1:2*N, 3*N+1:4*N]';
227         j = [2*N+1:3*N, 2*N+1:3*N]';
228     z1 = y_prev;
229                 [term1, term2] = nonlinterms(N,z1,TMa,BMy);
230                 v = [term1;term2];
231                 Dh_DY = sparse(i,j,v,4*N,4*N);
232                 B = A-h_t/2*Dh_DY;
233                 [V,W,P] = lu(B);
234             while error1 > tol_Newton
235                 iter = iter + 1;
236                 z1TMdispl = z1(2*N+1:3*N);
237                 T1 = term1.*z1TMdispl;
238                 T2 = term2.*z1TMdispl;
239     d = P*(y_mass+h_t/2*(gf1+gf2+y_j+nonlinearity(y_prev,N,TMa,BMy))+...
240                 h_t/2*nonlinearity(z1,N,TMa,BMy)-h_t/2*[zeros(N,1);
241                                                         T1;
242                                                         zeros(N,1);
243                                                         T2]);
244                  y = W\(V\d);
245                  error1 = norm(y-z1)/norm(y);
246                  z1 = y;
247             end
248
249     end
250
251
252     function [ t, y ] = cochleaSolverBDF_3_nonlin_M1( t, y_prev, y3,...
253             y2, y1, h_t, N, L, U, P, in, AM, TMa, BMy, om, om2, fac  )
254
255
256      iter = 0;
257      tol = 0.1;
258      error1 = tol+1;
```

```
 t = t + h_t;
 gf  = g( t,N,in,TMa,AM,om,om2,fac );
 z1 = y_prev;
while error1 > tol
                iter = iter+1;
 RHS = P*(h_t*gf+3*y3-3/2*y2+1/3*y1+h_t*nonlinearity(z1,N,TMa,BMy));
                y = U\(L\RHS);
                error1 = norm(y-z1)/norm(y);
                z1 = y;
end

end


function [ t, y] = cochleaSolverBDF_3_nonlin_M2( t, y_prev, y3,...
        y2, y1, h_t, N, A, L, U, in, AM, TMa, BMy, om, om2, fac )

iter = 0;
tol_Newton = 1e-4;
tol_gmres = 1e-6;
error1 = tol_Newton+1;
t=t+h_t;
gf  = g( t,N,in,TMa,AM,om,om2,fac );
     i = [N+1:2*N, 3*N+1:4*N]';
     j = [2*N+1:3*N, 2*N+1:3*N]';
z1 = y_prev;
     while error1 > tol_Newton
          iter = iter + 1;
          [term1, term2] = nonlinterms(N,z1,TMa,BMy);
          v = [term1;term2];

          Dh_DY = sparse(i,j,v,4*N,4*N);
          B = A-h_t*Dh_DY;

          z1TMdispl = z1(2*N+1:3*N);
          T1 = term1.*z1TMdispl;
          T2 = term2.*z1TMdispl;
    d = h_t*gf+3*y3-3/2*y2+1/3*y1+h_t*nonlinearity(z1,N,TMa,BMy) -...
          h_t*[zeros(N,1);
                  T1;
                  zeros(N,1);
                  T2];
           [y,~] = gmres(B,d,[],tol_gmres,500,L,U,z1);
           error1 = norm(y-z1)/norm(y);
           z1 = y;
     end

end


function  x  = gaussgrid(N, plotflag)

if nargin<2,
```

```
312             plotflag=0;
313 end
314
315 if nargin<1,
316             N=500;
317 end
318
319 stred=0.35;
320 sig1=2*(1.2)^2;
321 sig2=2*(0.46)^2;
322
323 xo=(1:N)/N;
324
325 ind=ceil(stred/xo(1));
326 dx(1:ind)=exp(((xo(1:ind)-xo(ind)).^2)./sig1);
327 dx(ind+1:N)=exp(((xo(ind+1:N)-xo(ind)).^2)./sig2);
328 dx=dx/sum(dx);
329
330 x(1)=dx(1);
331
332 for i=2:N,
333             x(i)=x(i-1)+dx(i);
334 end
335
336 if (x(N) > 1),
337             x(N) = 1;
338 end
339
340 dens=ones(size(dx))./dx;
341 Nmx=max(dens);
342 Nmn=min(dens);
343
344 if plotflag==1,
345             [xm, im] = min(abs(x-0.5));
346             Nmdl=dens(im);
347
348             figure(2);
349             clf,
350             plot(x, zeros(size(x))+Nmx/2,'.w',x,dens,'-r'),
351             axis([0,1,0,Nmx]);
352             text(0.2, 0.95*Nmx, ['Max.density = ', num2str(Nmx)])
353             text(0.2, 0.85*Nmx,  ['Mid.density = ', num2str(Nmdl)])
354             text(0.2, 0.75*Nmx, ['Min.density = ', num2str(Nmn)])
355             ylabel('Point density')
356         xlabel('Fractional distance from stapes')
357 end
358
359
360 function [Y] = nonlin(Y)
361
362 y1 = 0.01139;
363 y2 = 0.03736;
364 c1 = 0.7293;
```

```
365 c2 = 1.4974;
366 b = 0.30991;
367
368
369 Y=(1./(1+c1*exp(-Y./y1)+c2*exp(-Y./y2)))-b;
370
371 Y=0.1*Y;
372
373
374 function [ z ] = nonlinearity( y, N, TMa, BMy )
375
376         fun   = BMy.*nonlin(y(2*N+1:3*N));
377
378                 z   = [zeros(N,1);...
379                     -fun;...
380                     zeros(N,1);...
381                     TMa.*fun];
382
383 end
384
385
386 function [ term1,term2 ] = nonlinterms( N,y,TMa,BMy )
387
388 y1 = 0.01139;
389 y2 = 0.03736;
390 c1 = 0.7293;
391 c2 = 1.4974;
392
393 yTM = y(2*N+1:3*N);
394 der = 0.1*(c1/y1*exp(-yTM./y1)+c2/y2*exp(-yTM./y2))./...
395 (1+c1*exp(-yTM./y1)+c2*exp(-yTM./y2)).^2;
396
397 term1 = -BMy.*der;
398 term2 = TMa.*BMy.*der;
399 end
```