UiT THE ARCTIC UNIVERSITY OF NORWAY

Faculty of Science and Technology
Department of Mathematics and Statistics

On the Ewald-Oseen scattering formulation for linear and nonlinear transient wave scattering

—

Anastasiia Kuzmina

MAT-3900 Master's Thesis in Mathematics ... May 2017

Contents

Introduction

Boundary integral formulations are well known in all areas of science and technology and leads to highly efficient numerical algorithms for solving partial differential equations. Their utility are in particular evident for scattering of waves from objects located in an unbounded space. For these situations, one whole space dimension is taken out of the problem by reducing the solution of the original PDEs to the problem of an integral equation located on the boundaries of the scattering objects.

However, this reduction rely on the use of Green's functions and is therefore only possible if the PDEs are linear. For computational reasons one is also usually restricted to situations where the Green's functions are given by explicit formulas, and this rules out most situations where the materials are inhomogeneous. Since many problems involves inhomogeneous materials with a nonlinear response, boundary integral methods have appeared to be of limited utility in computational science. Adding to this the somewhat advanced mathematical machinery that is needed to formulate PDEs in terms of boundary integral equations, it is perhaps not hard to understand why the method is not all that popular.

Domain-based methods, like the finite difference method and the finite element method, on the other hand, appears to be of much wider utility. Their simple formulation and wide applicability to many types of PDEs, both linear and nonlinear, has made them extremely popular in the scientific computing community. In the context of scattering problems they do, however, have problems of their own to solve. These problems are of two quite distinct types.

The first type of problem is related to the fact that, frequently, the scattering objects represents abrupt changes in material properties as compared to the properties of the surrounding homogeneous space. This abrupt change leads to PDEs with discontinuous or non-linear coefficients. Such features are hard to represent accurately using finite element or finite difference methods. The favoured approach is to introduce multiple, interlinked grids, that are adjusted so that they conform to the boundaries of the scattering objects. Generating these grids, that are tailored to the possibly complex shape of the scattering objects, linking them together in a correct way and designing them in such a way that the resulting numerical algorithm is accurate and stable, is a whole separate science with its own journals and conferences. The approach has been developed over many years and in general works quite well, but it certainly adds to the implementational complexity of these methods.

The second type of the problem is related to the fact that one can't grid the domain where the scattering objects are located for the simple reason that in almost all situations of interest this domain is unbounded. This problem is of course well known in the research community and the way it is resolved is to grid a computational box that is large enough to contain all scattering objects of interest. This can easily become a very large domain, leading to a very large number of degrees of freedom in the numerical algorithm. However, most of the time, the part of the numerical algorithm linked to this domain has a simple structure for which it is possible to design very fast implementations if the structure is taken into account. However, the introduction of the finite computational box in what is really a unbounded domain leads to the box in such a way that it is fully transparent to waves. This is not an easy thing to achieve, most approaches one can think of will in one way or another introduce an inhomogeneity that will partly reflect waves hitting the boundary. This problem was first solved in a fully satisfactory manner for the case of scattering of electromagnetic waves. The domain based method of choice for electromagnetic waves is the Finite Difference Time Domain method(FDTD) [4], [10], [11]. This is, as the name indicate, the finite difference method, but a method that has been designed to take into account the very special structure of Maxwell's equation. The removal of reflections from the the finite computational box was achieved by the introduction of a Perfectly Matched Layer(PML) [2], [5]. This amounts to adding a narrow layer of a specially constructed artificial material to the outside the computational box. The PML is however only perfectly transparent to wave propagation if the grid has infinite resolution. For any finite grid there is still a small, but nonzero, reflection from the boundary of the computational box. This can be reduced by making the PML thicker, but this leads to more degrees of freedom and thus an increasing computational load. However, overall PML works well, and certainly much better than anything that came before it. There is no doubt that the introduction of PML was a breakthrough.

The use of PML was closely linked to the special structure of Maxwell's equations. However, it was soon realized that the same effect could be achieved by complexifying the physical space outside the computational box and analytic continuing the fields into this complex spatial domain. Significantly, this realization made the benefits of a reflectionless boundary condition available to all kinds of scattering problems. However, the use of these reflexionless boundary conditions certainly leads to an increased computational load, increased implementational complexity and also to numerical stability issues that needs to be resolved. It is at this point worth recalling that the boundary of the computational box is not part of the original physical problem and that all the added implementational complexity and computational cost is spent trying to make it invisible after choice of a domain method forced us to put it there in first place.

What we propose in this paper is to only apply the domain-based method inside each scattering object. Firstly, this will reduce the size of the computational grid enormously since we now only need to grid the inside of the scattering objects. Secondly, our approach makes it possible to use different computational grids for each scattering objects that is tailored to their geometric structure, without having to worry about the inherent complexity caused by letting the different grids meet up. Thirdly, it makes the introduction of a large computational box, with its artificial boundary, redundant. In this way the computational load is reduced by a large amount and we rid of the implementational complexity and instabilities associated with the boundary of the computational box.

However, the domain based method resrticted to the inside of each scattering object requires field values on the boundaries of the scattering object in order to be able to propagate the fields forward in time. These boundary values will be supplied by a boundary integral method derived from a space-time integral formulation of the PDEs one are seeking to solve. This boundary integral method will take into account all the scattering and rescattering of the solution to the PDEs in the unbounded domain outside the scattering objects. Since the boundary integral method takes the radiation condition at infinity into account explicitly, no finite computational box with its artificial boundary is needed.

This kind of idea for solving scattering problems was to out knowledge first proposed in 1972 by Pattanayak and Wolf [6] for the case of electromagnetic waves. They discussed their ideas in the context of a generalization of the Ewald-Oseen optical extinction theorem and we will, because of this, refer to our method as the Ewald-Oseen Scattering(EOS) formulation.

However, the paper of Pattanayak and Wolf only discussed stationary linear scattering of electromagnetic waves and they therefore did their integral formulation in frequency space. This approach is not the right one when one is interested in transient scattering from objects that are in general inhomogeneous and have a possibly nonlinear response. What is needed for our approach is a space-time integral formulation of the PDEs of interest.

In the first chapter of this work we implement our EOS formulation for two different 1D scattering problems. Both cases can be thought of as toy models for the scattering of electromagnetic waves. This should not be taken to mean that only models that in some way is related to electromagnetic scattering can be subject to our approach, it merely reflects our particular interest in electromagnetic scattering. The way we see it, there is only one essential requirement for our method to be applicable, and that is that it must be possible to derive an explicit integral formulation for the PDEs of interest. This means that at some point one needs to find the explicit expression for a Green's function for some differential operator related to our PDEs. In general, it is difficult to find explicit expressions for Green's function belonging to nontrivial differential operators. However, the Green's function needed for our EOS formulation will always be of the infinite, homogeneous space type, and explicit expressions for such Green's functions can frequently be found.

The goal of this master thesis is twofold:

1. Firstly, the candidate is going to create a numerical implementation of the EOS formulation for the two toy models. For both models, the numerical implementations should be tested using a method known as artificial sources. This method has probably been around for a long time but apart from an application to the Navier-Stokes equations [3] we are not aware of any published work using this method. The method is based on the simple observation that you add arbitrary source terms to any system of PDEs the any function is a solution for some choice of the source. Adding a source term typically only introduce a trivial modification to whichever numerical method used to solve the PDEs. This essentially means that for any PDE of interest we can design particular functions to test various critical aspects of the numerical method related to numerical stability and accuracy.

2. Secondly, the candidate will develop the EOS formulation for the following system of PDEs

$$
\begin{aligned}
\partial_t \varphi &= \mu(\partial_y \psi + i\partial_x \psi) + f(x, y, t) \\
\partial_t \psi &= \nu(\partial_y \varphi - i\partial_x \varphi) + g(x, y, t)
\end{aligned}
$$

   implement the EOS formulation numerically and test the implementation using an artificial source test.

So in the first chapter of this work we will discuss two 1D-space toy models. We will determine the EOS formulation for both of toy models. For the first model we will try to solve test problem by several numerical methods: Central Finite Difference method, Modified Euler method, Lax-Wendroff method and by ODE solver which is in Python scientific library. After choosing more accurate and stable method we will use it to solve both toy models.

In the second part of this work we are going to only discuss EOS formulation for the 2D scattering problem.

All calculations were made using programming language Python3, a computer algebra system Wolfram Mathematica and using Intel(R) Atom(TM) CPU N570 clocked at 1.66GHz.

Chapter 1

Toy models

1.1 The first toy model

Our first toy model is

$$\begin{aligned} \varphi_t &= v\varphi_x + j + j_s \\ \rho_t &= -j_x \\ j_t &= (\alpha - \beta\rho)\varphi - \gamma j \end{aligned} \tag{1.1}$$

Where $\varphi = \varphi(x,t)$ – "electric field", $j = j(x,t)$ – "current density", $\rho = \rho(x,t)$ – "charge density". These quantities are analogues for the corresponding quantities in the Maxwell equations. We observe that the second equation in the model (1.1) is a 1D version of the equation of continuity from electromagnetics. The charge density and current density will be assumed to be confined to an interval $[a_0, a_1]$ on the real axis whereas φ is a continuous field defined on the whole real axis. The interval $[a_0, a_1]$ is the analogue of a compact scattering object in the electromagnetic situation. The function $j_s(x,t)$ is a fixed source, that has its support in a compact set in the interval $x > a_1$. The parameters α, β and γ are constants and the function

$$v = \begin{cases} c_1 & x \in [a_0, a_1] \\ c_2 & \text{otherwise} \end{cases}$$

is the propagation speed for the field.

Graphical representation of (1.1) is shown on Figure 1.1

1.1.1 The EOS formulation

In order to derive the EOS formulation for the model (1.1), we will firstly need a space-time integral identity involving the operator

$$L = \partial_t - c\partial_x$$

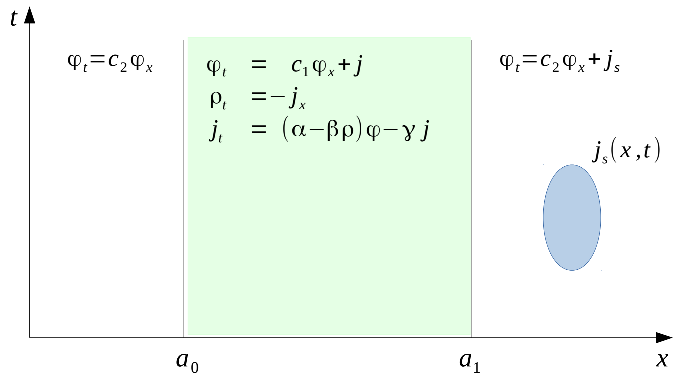

Figure 1.1: Graphical representation of the first toy model. The blue ellipse represents the source $j_s(x,t)$ outside the domain bounded by $x = a_0$ and $x = a_1$.

where c is some constant.

$$\begin{aligned}\int_{S\times T} \mathrm{d}x\,\mathrm{d}t L\varphi\psi &= \int_{S\times T} \mathrm{d}x\,\mathrm{d}t L(\varphi_t - c\varphi_x)\psi \\ &= \int_{S\times T} \mathrm{d}x\,\mathrm{d}t L\varphi_t\psi - \int_{S\times T} \mathrm{d}x\,\mathrm{d}t c\varphi_x\psi \\ &= \int_S \mathrm{d}x\psi\varphi\Big|_{t_0}^{t_1} - \int_T \mathrm{d}t\varphi\psi\Big|_{x_0}^{x_1} - \int_{S\times T} \mathrm{d}x\,\mathrm{d}t\varphi(\psi_t - c\varphi\psi_x)\end{aligned}$$

So we have integral identity

$$\int_{S\times T} \mathrm{d}x\,\mathrm{d}t\Big[L\varphi\psi - \varphi L^\dagger\psi\Big] = \int_S \mathrm{d}x\psi\varphi\Big|_{t_0}^{t_1} - \int_T \mathrm{d}t\varphi\psi\Big|_{x_0}^{x_1} \tag{1.2}$$

where $L^\dagger = -\partial_t + c\partial_x$ is the formal adjoin of L and where $S = (x_0, x_1)$ and $T = (t_0, t_1)$ are open space and time intervals.

The second item we need in order to derive the EOS formulation for the model (1.1), is the advanced Green's function for the operator $L^\dagger$. This is a function $G = G(x, t, x', t')$ which is solution to the equation

$$L^\dagger G(x, t, x', t') = \delta(t - t')\delta(x - x') \tag{1.3}$$

and that vanish when $t > t'$.

We use the one dimensional Fourier transform in x to solve the initial value problem for $G(x, t, x', t')$. Let the Fourier transform of $G(x, t, x', t')$ be denoted by $g(\lambda, t, x', t')$. Then

$$g(\lambda, t, x', t') = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \mathrm{d}x e^{i\lambda x} G(x, t, x', t') \tag{1.4}$$

To obtain an equation for $g(\lambda, t, x', t')$ we multiply (1.4) by $\frac{1}{\sqrt{2\pi}}e^{i\lambda x}$ and integrate from $-\infty$ to $+\infty$. Using

$$\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} \mathrm{d}x e^{i\lambda x} f^{(n)}(x) = -(i\lambda)^n F(\lambda), \quad n = 1, 2..$$

we obtain the equation.

$$\begin{aligned} -\frac{\partial g(\lambda, t, x', t')}{\partial t} + c\lambda g(\lambda, t, x', t') &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} dx e^{i\lambda x} G(x, t, x', t')\delta(t - t')\delta(x - x') \\ &= \frac{1}{\sqrt{2\pi}} e^{i\lambda x'}\delta(t - t') \end{aligned} \tag{1.5}$$

Also $g(\lambda, t, x', t')$ satisfies the condition $g(\lambda, 0, x', t') = 0$.

$$\begin{aligned} \left(e^{-ic\lambda t} g(\lambda, t, x', t')\right)_t &= -\frac{1}{\sqrt{2\pi}} e^{i\lambda x'} e^{-ic\lambda t}\delta(t - t') \\ &= -\frac{1}{\sqrt{2\pi}} e^{i\lambda x'} e^{-ic\lambda t'}\delta(t - t') \end{aligned}$$

$$g(\lambda, t, x', t') = \frac{1}{\sqrt{2\pi}} e^{ic\lambda(t-t')} e^{i\lambda x'}\Theta(t' - t)$$

So

$$\begin{aligned} G(x, t, x', t') &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} d\lambda e^{-i\lambda x} g(\lambda, t, x', t') \\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} d\lambda e^{-i\lambda x}\frac{1}{\sqrt{2\pi}} e^{ic\lambda(t-t')} e^{i\lambda x'}\theta(t' - t) \\ &= \frac{1}{2\pi}\theta(t' - t)\int_{-\infty}^{+\infty} d\lambda e^{i\lambda(x'-x)} e^{ic\lambda(t-t')} \\ &= \theta(t' - t)\delta(x' - x + c(t' - t)) \end{aligned} \tag{1.6}$$

where θ is the Heaviside step function with $\theta(x) = 1$ for $x > 0$ and zero otherwise.

We will now apply the integral identity (1.2) to each space interval $(-\infty, a_0)$, (a_0, a_1) and $(a_1, +\infty)$. For the function ψ we will substitute the advanced Green's function (1.6) and φ will be the solution to the equation

$$\varphi_t = v\varphi_x + j + j_s \tag{1.7}$$

with vanishing initial condition $\varphi(x, t_0) = 0$. We thus have a problem where all solutions are purely source-generated.

First interval $(-\infty, a_0)$ we let ψ be the Green's function

$$\psi = G_2(x,t,x',t') = \theta(t'-t)\delta(x'-x+c_2(t'-t)) \tag{1.8}$$

According to the already described properties of j and j_s, the function $\varphi = \varphi_2$ is in the interval $(-\infty, a_0)$ a solution to the equation

$$\begin{gathered} \varphi_{2_t} = v\varphi_{2_x} \\ \Updownarrow \\ L\varphi_2 = 0 \end{gathered} \tag{1.9}$$

Inserting (1.8), (1.9) and $S = (-\infty, a_0)$ into the integral identity (1.2), using the initial condition and the fact that the Green's function is advanced, we get for x in $(-\infty, a_0)$

$$\int_{-\infty}^{a_0}\int_{t_0}^{t} \mathrm{d}x\,\mathrm{d}t\Big[L\varphi_2 G_2 - \varphi_2 L^\dagger G_2\Big] = \int_{-\infty}^{a_0} \mathrm{d}x\varphi_2 G_2\Big|_{t_0}^{t} - c_2\int_{t_0}^{t}\mathrm{d}t\varphi_2 G_2\Big|_{-\infty}^{a_0}$$

$$\begin{aligned} \varphi_2(x,t) &= c_2\int_{t_0}^{t}\mathrm{d}t'\varphi_2(x',t')G_2(x',t',x,t)\Big|_{-\infty}^{a_0} \\ &= c_2\int_{t_0}^{t}\mathrm{d}t'\varphi_2(x',t')\theta(t-t')\delta(x-a_0+c_2(t-t')) \\ &- c_2\lim_{x'\to-\infty}\int_{t_0}^{t}\mathrm{d}t'\varphi_2(x',t')\theta(t-t')\delta(x-x'+c_2(t-t')) \\ &= c_2\int_{t_0}^{t}\mathrm{d}t'\varphi_2(a_0,t')\delta(x-a_0+c_2(t-t')) \end{aligned} \tag{1.10}$$

after interchanging the primed and unprimed variables. The last equality sign follows because $x - x' + c_2(t-t') > 0$ when $x' < x$ for all t' in the integration interval (t_0, t).

For the second integration interval, (a_0, a_1), we let ψ be the Green's function

$$\psi = G_1(x,t,x',t') \equiv \theta(t'-t)\delta(x'-x+c_1(t'-t)). \tag{1.11}$$

According to our specifications, the function $\varphi = \varphi_1$ is in the interval (a_0, a_1)

$$\begin{gathered} \varphi_{1_t} = v\varphi_{1_x} + j \\ \Updownarrow \\ L\varphi_1 = j \end{gathered} \tag{1.12}$$

with vanishing initial conditions. Inserting (1.11), (1.12) and $S = (a_0, a_1)$ into the integral identity (1.2), using the initial condition and the fact that the Green's function is advanced, we get for x in (a_0, a_1)

$$\int_{a_0}^{a_1}\int_{t_0}^{t} \mathrm{d}x'\,\mathrm{d}t'\Big[L\varphi_1 G_1 - \varphi_1 L^\dagger G_1\Big] = \int_{a_0}^{a_1} \mathrm{d}x'\varphi_1 G_1\Big|_{t_0}^{t} - c_1\int_{t_0}^{t}\mathrm{d}t'\varphi_1 G_1\Big|_{a_0}^{a_1}$$

$$\begin{aligned}\varphi_1(x,t) = c_1 &\int_{t_0}^{t} \mathrm{d}t' \varphi_1(a_1,t')\delta(x-a_1+c_1(t-t')) \\ - c_1 &\int_{t_0}^{t} \mathrm{d}xt' \varphi_1(a_0,t')\delta(x-a_0+c_1(t-t')) \\ + &\int_{a_0}^{a_1}\int_{t_0}^{t} \mathrm{d}x'\,\mathrm{d}t' j(x',t')\delta(x-x'+c_1(t-t'))\end{aligned} \tag{1.13}$$

after interchanging primed and unprimed variables. The last equality sign follows because $x - a_0 + c_1(t-t') > 0$ for all t' in the integration interval when $a_0 < x < a_1$.

Finally, for the third integration interval $(a_1, +\infty)$ we let ψ be the Green's function

$$\psi = G_2(x,t,x',t') \equiv \theta(t'-t)\delta(x'-x+c_2(t'-t)) \tag{1.14}$$

According to the properties of j and j_s, the function φ is in the interval (a_1, ∞) the solution to the equation

$$\begin{aligned}\varphi_{2_t} &= v\varphi_{2_x} + j_s \\ &\Updownarrow \\ L\varphi_2 &= j_s\end{aligned} \tag{1.15}$$

with vanishing initial conditions. Inserting (1.14), (1.15) and $S = (a_1, \infty)$ into the integral identity (1.2), using the initial conditions and the fact that the Green's function is advanced, we get for x in (a_1, ∞)

$$\int_{a_1}^{+\infty}\int_{t_0}^{t} \mathrm{d}x'\,\mathrm{d}t' \Big[L\varphi_2 G_2 - \varphi_2 L^{\dagger} G_2 \Big] = \int_{a_1}^{+\infty} \mathrm{d}x' \varphi_2 G_2 \Big|_{t_0}^{t} - c_2 \int_{t_0}^{t} \mathrm{d}t' \varphi_2 G_2 \Big|_{a_1}^{+\infty}$$

$$\begin{aligned}\varphi_2(x,t) = c_2 &\lim_{x'\to+\infty} \int_{t_0}^{t} \mathrm{d}t' \varphi_2(x',t')\delta(x-x'+c_2(t-t')) \\ - c_2 &\int_{t_0}^{t} \mathrm{d}t' \varphi_2(a_1,t')\delta(x-a_1+c_2(t-t')) \\ + &\int_{a_1}^{+\infty}\int_{t_0}^{t} \mathrm{d}x'\,\mathrm{d}t' j_s \delta(x-x'+c_2(t-t')) \\ = &\int_{a_1}^{+\infty}\int_{t_0}^{t} \mathrm{d}x'\,\mathrm{d}t' j_s \delta(x-x'+c_2(t-t'))\end{aligned} \tag{1.16}$$

after interchanging primed and unprimed variables. The third terms vanish because $x - a_1 + c_2(t-t') > 0$ for all t' in the integration interval when $x > a_1$. The second term vanish because $x - x' + c_2(t-t') < 0$ for all fixed $x > a_1$, $t > t_0$ and all t' in the integration interval (t_0, t) when x' is large enough.

We now investigate the limit of there integral identities as x approach the boundary points $\{a_0, a_1\}$ of the open interval (a_0, a_1) from inside the interval and outside the interval. This will give us four equations for the four boundary values

$$\varphi_1(a_0, t), \varphi_2(a_0, t), \varphi_1(a_1, t), \varphi_2(a_1, t)$$

However, by assumption, acceptable solutions of the toy model are continuous across the boundary points $\{a_0, a_1\}$. We therefore also have the two additional equations

$$\begin{aligned} \varphi_2(a_0, t) &= \varphi_1(a_0, t) \\ \varphi_1(a_1, t) &= \varphi_2(a_1, t) \end{aligned} \tag{1.17}$$

So we have four unknown boundary values and six linear equations. The problem is thus overdetermined and we would not normally expect there to be any nontrivial solutions.

On the other hand, the first of the equations in the toy model determine the function φ uniquely on the whole real axis in terms of the source $j + j_s$ and the vanishing initial conditions. The function satisfy, by construction, the integral identities whose limits yielded the overdetermined system. So the overdetermined linear system does in fact have a solution.

There is a more direct way to see why the overdetermined system will have a solution. Let us consider the inside of the scattering object, thus $x \in (a_0, a_1)$. Here, the field φ_1 is determined in terms of the current $j(x, t)$, and the boundary value $\varphi_1(a_1, t)$ by identity (1.13)

$$\begin{aligned} \varphi_1(x, t) = &\int_{a_0}^{a_1} \mathrm{d}x' \int_{t_0}^{t} \mathrm{d}t' j(x', t') \delta(x - x' + c_1(t - t')) \\ &+ c_1 \int_{t_0}^{t} \mathrm{d}t' \varphi_1(a_1, t') \delta(x - a_1 + c_1(t - t')) \end{aligned} \tag{1.18}$$

Natively, one would expect that we will get an equation determining the unknown boundary value $\varphi_1(a_1, t)$, by taking the limit of (1.18) as x approach a_1 from below. However, this would make the field inside the scattering object independent of the outside source, which from a scattering point of view must be patently wrong. After all, it is the outside source $j_s(x, t)$ which determine the field both outside and inside the scattering object. If this source is turned off the field would sipmly be zero everywhere. Note that if we actually take the limit of (1.18) we get the equation

$$\varphi_1(a_1, t) = 0,$$

which leaves the boundary value entirely arbitrary. If we analyse the rest of the overdetermined system in the same way, we find that one more equation for the boundary data is redundant

$$
\begin{aligned}
\varphi_1(a_0,t) &= \lim_{x\to a_0}\int_{a_0}^{a_1}\mathrm{d}x'\int_{t_0}^{t}\mathrm{d}t' j(x',t')\delta(x-x'+c_1(t-t')) \\
&+ c_1\lim_{x\to a_0}\int_{t_0}^{t}\mathrm{d}t'\varphi_1(a_1,t')\delta(x-a_1+c_1(t-t')) \\
&= \frac{1}{c_1}\int_{a_0}^{a_1}\mathrm{d}x' j(x',t-\frac{x'-a_0}{c_1})\theta(a_0-x'+c_1(t-t')) \\
&+ \varphi_1(a_1,t-\frac{a_1-a_0}{c_1})\delta(a_0-a_1+c_1(t-t'))
\end{aligned}
\tag{1.19}
$$

$$
\begin{aligned}
\varphi_2(a_0,t) &= c_2\lim_{x\to a_0}\int_{t_0}^{t}\mathrm{d}t'\varphi_2(a_0,t')\delta(x-a_0+c_2(t-t')) \\
&= -\varphi_2(a_0,t) = 0
\end{aligned}
\tag{1.20}
$$

$$
\begin{aligned}
\varphi_2(a_1,t) &= \lim_{x\to a_1}\int_{a_1}^{+\infty}\mathrm{d}x'\int_{t_0}^{t}\mathrm{d}t' j_s(x',t')\delta(x-x'+c_2(t-t')) \\
&= \frac{1}{c_2}\int_{a_1}^{+\infty}\mathrm{d}x' j_s(x',t-\frac{x'-a_1}{c_2})\theta(a_1-x'+c_2(t-t_0))
\end{aligned}
\tag{1.21}
$$

and thus the four unknown boundary values are uniquely determined by the following four equations

$$
\begin{aligned}
\varphi_1(a_0,t) &= \frac{1}{c_1}\int_{a_0}^{a_1}\mathrm{d}x' j(x',t-\frac{x'-a_0}{c_1})\theta(a_0-x'+c_1(t-t')) \\
&+ \varphi_1(a_1,t-\frac{a_1-a_0}{c_1})\theta(a_0-a_1+c_1(t-t'))
\end{aligned}
\tag{1.22}
$$

$$
\varphi_2(a_1,t) = \frac{1}{c_2}\int_{a_1}^{+\infty}\mathrm{d}x' j_s(x',t-\frac{x'-a_1}{c_2})\theta(a_1-x'+c_2(t-t_0))
\tag{1.23}
$$

$$
\begin{aligned}
\varphi_2(a_0,t) &= \varphi_1(a_0,t) \\
\varphi_1(a_1,t) &= \varphi_2(a_1,t)
\end{aligned}
$$

We emphasize the fact that we end up with an overdetermined system of linear equations for the boundary values because this is a generic outcome when we derive the EOS formulation for any given system of PDEs.

This problem has been recognized by the research community in the context of space-time boundary integral formulation for the Maxwell equations, and a simple fix has been invented to resolve it.

Observe that equation (1.23) determine the value of the field at the boundary point a_1 in terms of the given source j_s, and the equation (1.22) determine the value of the field at the boundary point a_0 in terms of the current j which by definition is located inside the interval (a_0, a_1).

Equations (1.1) restricted to the open interval (a_0, a_1) together with the integral identities (1.22), (1.23) define the EOS formulation for toy model 1.

1.1.2 Numerical implementation of the first toy problem

Solving of the test problem

Before to solve original problem (1.1) let us consider test linear boundary problem

$$\begin{aligned} &\varphi_t = c\varphi_x + j + f_1(x,t), \quad a_0 < x < a_1, \quad 0 < t < T \\ &\rho_t = -j_x + f_2(x,t) \\ &j_t = \alpha\varphi - \gamma j + f_3(x,t) \end{aligned} \tag{1.24}$$

with zero initial conditions and boundary conditions $\varphi(a_0,t) = f(t)$ and $\varphi(a_1,t) = g(t)$ by different methods and to choose the most suitable one for this problem.

Actually original problem is delay problem because $\varphi(t, a_0)$ depends on values of $j(x,t)$ in previous time points. But it is difficult to find exact solution to delay system and compare it with numerical solution, so we are going to solve system of PDE.

We use exact boundary conditions for function φ because in original problem we have expressions for this function on the boundaries and know nothing about boundary values neither for function ρ or j.

We can choose exact solution for this problem (1.24) to find functions $f_1(x,t)$, $f_2(x,t)$ and $f_3(x,t)$.

Let us take the next exact solution to the problem (1.24)

$$\begin{aligned} \varphi(x,t) &= e^{-10(x+2t-7)^2} \\ \rho(x,t) &= -0.1e^{-10(x+2t-7)^2} \\ j(x,t) &= 0.1e^{-10(x+2t-7)^2} \end{aligned} \tag{1.25}$$

Then we can find that

$$\begin{aligned} f_1(x,t) &= ((20c-40)(x+2t-7) - 0.1)e^{-10(x+2t-7)^2} \\ f_2(x,t) &= 2(x+2t-7)e^{-10(x+2t-7)^2} \\ f_2(x,t) &= (-4(x+2t-7) - \alpha + 0.1\gamma)e^{-10(x+2t-7)^2} \end{aligned} \tag{1.26}$$

We are going to solve this problem by Central Finite Difference method (CD), Modified Euler method (ME), Lax-Wendroff method (LW) and by ODE solver which called `odeint` from Python3 library for scientific computing SciPy.

Central Finite Difference method First method is finite difference method because this method is the most popular and simple to realize. We will use Central Finite Difference rule for time and space derivative. We introduce a uniform spacetime grid (x_k, t_i) where $x_k = a_0 + kdx$, $k = 0, \dots, N+1$ and $t_i = idt$, $i = 0, \dots, M$ where $x_0 = a_0$, $x_{N+1} = a_1$, $dx = \dfrac{a_1 - a_0}{N+2}$. The solution will only be computing at the grid points.

The expressions for numerical solution at the grid points are

$$\begin{aligned} \varphi_k^{i+1} &= c\frac{\mathrm{d}t}{\mathrm{d}x}(\varphi_{k+1}^i - \varphi_{k-1}^i) + 2\,\mathrm{d}t(j_k^i + f_1(x_k, t_i)) + \varphi_k^{i-1} \\ \rho_k^{i+1} &= -\frac{\mathrm{d}t}{\mathrm{d}x}(j_{k+1}^i - j_{k-1}^i) + 2\,\mathrm{d}t f_2(x_k, t_i) + \rho_k^{i-1} \\ j_k^{i+1} &= 2\,\mathrm{d}t(\alpha\varphi_k^i - \gamma j_k^i + f_3(x_k, t_i)) + j_k^{i-1} \end{aligned} \tag{1.27}$$

Polynomial approximation We need values of ρ on the boundaries a_0 and a_1, but we cannot calculate them by usual way using Central Finite Difference method because we need values of j in outside points, which we do not have. So we need another way to find boundary values of ρ.

Let us take the polynomial approximation of function $U(x)$

$$U(x) = a_0 + a_1(x - x_k) + a_2(x - x_k)^2 + O(x^3)$$

. Then the x-derivative is

$$U_x(x) = a_1 + 2a_2(x - x_k)$$

$$U_x(x_k) = a_1$$

If we have points x_k, x_{k+1}, x_{k+2}, then $x_{k+1} - x_k = \mathrm{d}x$, $x_{k+2} - x_k = 2\,\mathrm{d}x$, then we can get linear system

$$\begin{aligned} U(x_k) &= a_0 \\ U(x_{k+1}) &= a_0 + a_1\,\mathrm{d}x + a_2\,\mathrm{d}x^2 \\ U(x_{k+2}) &= a_0 + 2a_1\,\mathrm{d}x + 4a_2\,\mathrm{d}x^2 \end{aligned}$$

with two unknown variables a_1 and a_2. If we solve this system we find that

$$U_x(x_k) = a_1 = \frac{-3U(x_k) + 4U(x_{k+1}) - U(x_{k+2})}{2\,\mathrm{d}x} \tag{1.28}$$

for the left boundary. Analogously we can construct system for the right boundary, but now we have points x_k, x_{k-1}, x_{k-2} and $x_{k-1} - x_k = -\,\mathrm{d}x$, $x_{k-2} - x_k = -2\,\mathrm{d}x$

$$\begin{aligned} U(x_k) &= a_0 \\ U(x_{k-1}) &= a_0 - a_1\,\mathrm{d}x + a_2\,\mathrm{d}x^2 \\ U(x_{k-2}) &= a_0 - 2a_1\,\mathrm{d}x + 4a_2\,\mathrm{d}x^2 \end{aligned}$$

If we solve this system we find that

$$U_x(x_k) = a_1 = \frac{3U(x_k) - 4U(x_{k-1}) + U(x_{k-2})}{2\,\mathrm{d}x} \tag{1.29}$$

for the right boundary.

Now we can use these two special rules (1.28) and (1.29) to calculate values of function ρ on the boundaries

$$\begin{aligned} \rho_0^{i+1} &= -\frac{\mathrm{d}t}{\mathrm{d}x}(-3j_0^i + 4j_1^i - j_2^i) + 2\,\mathrm{d}t f_2(x_0, t_i) + \rho_0^{i-1} \\ \rho_{N+1}^{i+1} &= -\frac{\mathrm{d}t}{\mathrm{d}x}(3j_{N+1}^i - 4j_N^i + j_{N-1}^i) + 2\,\mathrm{d}t f_2(x_{N+1}, t_i) + \rho_{N+1}^{i-1} \end{aligned} \tag{1.30}$$

Actually Central Finite Difference method in time and space (Leap-Frog) [7] is a two-level scheme, requiring records of values at time steps t_i and t_{i-1} to get values at time step t_{i+1}. This is clear from expressions (1.27). So we need two initial conditions at $t = t_0$ and $t = t_1$. For original problem (1.1) it doesn't matter because the solution stays zero or close to zero for several first time steps. But for problem (1.24) we need one more initial condition, which we can find from known exact solution.

So let us see now on Figures 1.2–1.4 the numerical solution of the problem (1.24) and compare them with exact solution in several time points.

In this test $T = 12$, $a_0 = -3$, $a_1 = 3$, $c = 0.5$, $\alpha = \gamma = 0.1$, $\mathrm{d}t = 0.01$, $\mathrm{d}x = 0.01$.

We can see that Central Finite Difference method solution overlaps exact solution perfectly until the signal gone. After the signal gone there are some oscillations around zero which possibly come from the numerical scheme. Also we can see artefact in solution to ρ on the left boundary. The first point grows up linearly. The nature of this artefact needs additional investigation and we are not going to do it in this work.

Modified Euler method Let us now apply Modified Euler (ME) method to the given test problem (1.24). Modified Euler method is an example of 2$^{\text{nd}}$ order Runge-Kutta method [9]. If we have initial value problem

$$y_t = f(t, y)$$

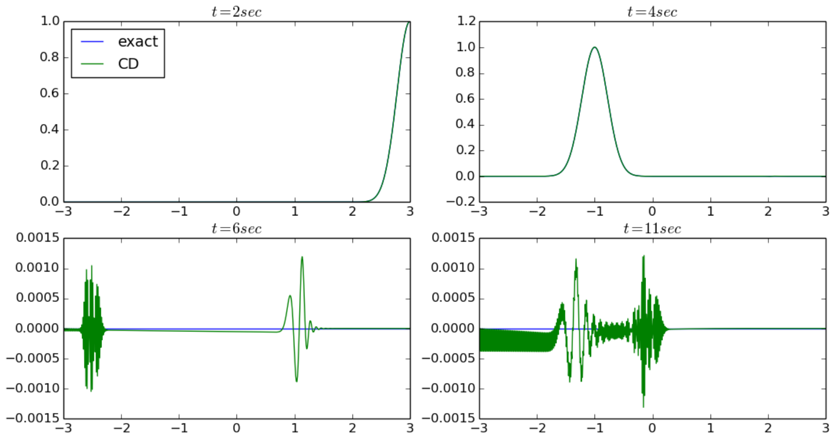

Figure 1.2: Exact (blue) solution and numerical (green) solution of test problem (1.24) for $\varphi(x,t)$ in time points $t = 2$ (upper left), $t = 4$ (upper right), $t = 6$ (lower left), $t = 11$ (lower right). Horizontal axis is x.

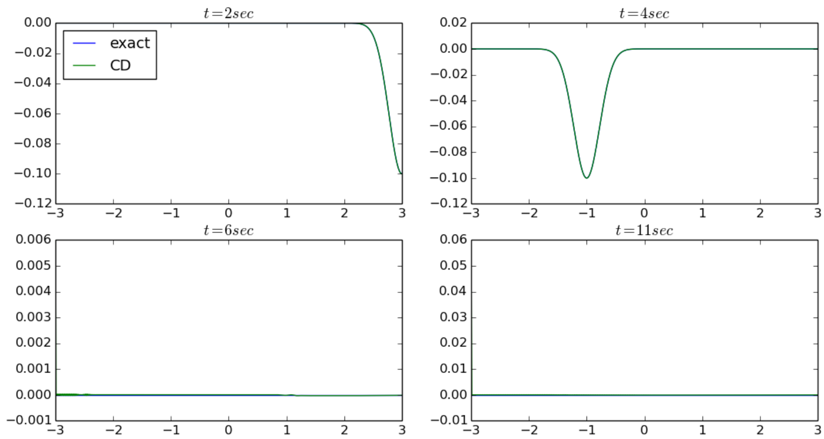

Figure 1.3: Exact (blue) solution and numerical (green) solution of test problem (1.24) for $\rho(x,t)$ in time points $t = 2$ (upper left), $t = 4$ (upper right), $t = 6$ (lower left), $t = 11$ (lower right). Horizontal axis is x.

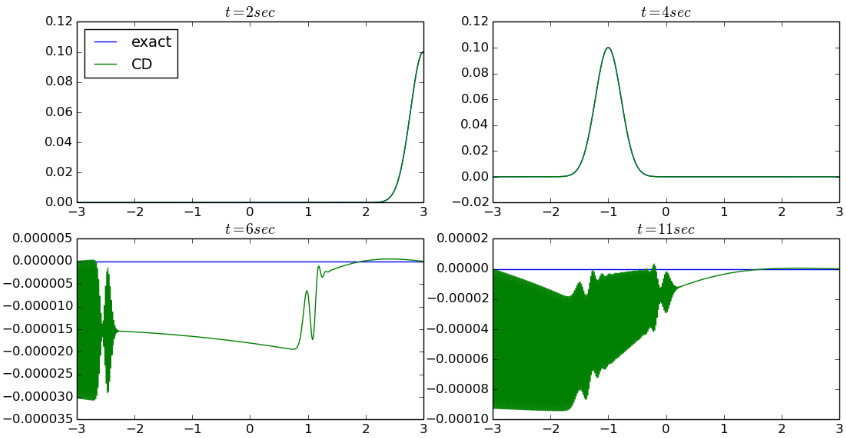

Figure 1.4: Exact (blue) solution and numerical (green) solution of test problem (1.24) for $j(x, t)$ in time points $t = 2$ (upper left), $t = 4$ (upper right), $t = 6$ (lower left), $t = 11$ (lower right). Horizontal axis is x.

the scheme of ME method takes form

$$K_1 = hf(t_i, y_i)$$
$$K_2 = hf(t_{i+1}, y_i + K_1)$$
$$y_{i+1} = y_i + \frac{K_1 + K_2}{2}$$

where h is time step, $t_i = ih$, $i = 0, \dots, n$.

If we apply this scheme, Central Finite Difference rule to x-derivatives and polynomial approximation for the boundary values from the previous section to the test problem we get the next scheme

$$
\begin{aligned}
K_1^\varphi &= \mathrm{d}t(\frac{c}{2\,\mathrm{d}x}(\varphi_{k+1}^i - \varphi_{k-1}^i) + j_k^i + f_1(x_k, t_i)) \\
K_2^\varphi &= \mathrm{d}t(\frac{c}{2\,\mathrm{d}x}(\varphi_{k+1}^i - \varphi_{k-1}^i) + j_k^i + f_1(x_j, t_{i+1})) \\
\varphi_k^{i+1} &= \varphi_k^i + \frac{K_1^\varphi + K_2^\varphi}{2} \\
K_1^\rho &= \mathrm{d}t(-\frac{1}{2\,\mathrm{d}x}(\varphi_{k+1}^i - \varphi_{k-1}^i)i + f_2(x_k, t_i)) \\
K_2^\rho &= \mathrm{d}t(-\frac{1}{2\,\mathrm{d}x}(\varphi_{k+1}^i - \varphi_{k-1}^i) + f_2(x_j, t_{i+1})) \\
\varphi_k^{i+1} &= \varphi_k^i + \frac{K_1^\rho + K_2^\rho}{2} \\
K_1^j &= \mathrm{d}t(\alpha\varphi_k^i - \gamma j_k^i + f_3(x_k, t_i)) \\
K_2^j &= \mathrm{d}t(\alpha\varphi_k^i - \gamma(j_k^i + K_1^j) + f_3(x_k, t_{i+1})) \\
j_k^{i+1} &= j_k^i + \frac{K_1^j + K_2^j}{2}
\end{aligned}
$$

where i, k, t_i, x_k are the same as for Central Finite Difference method which was described before.

So let us see now on Figures 1.5–1.7 the numerical solution of the problem (1.24) and compare them with exact solution in several time points.

In this test $T = 12$, $a_0 = -3$, $a_1 = 3$, $c = 0.5$, $\alpha = \gamma = 0.1$, $\mathrm{d}t = 0.01$, $\mathrm{d}x = 0.01$.

We can see that this method is unstable for the given problem and instability appears after the signal.

Lax-Wendroff method The next method which we are going to use is Lax-Wendroff method. This is finite difference method for first order equations. The Lax-Wendroff method can be used to approximate equations for φ and ρ from (1.24) by an explicit difference equation of a second-order accuracy.

Consider the problem (1.24) where c is a constant. By Taylor's expansion,

$$
\begin{aligned}
\varphi_k^i &= \varphi(t_i + dt, x_k) = \varphi_k^i + \mathrm{d}t(\varphi_t)_k^i + \frac{\mathrm{d}t^2}{2}(\varphi_{tt})_k^i + \dots \\
\rho_k^i &= \rho(t_i + dt, x_k) = \rho_k^i + \mathrm{d}t(\rho_t)_k^i + \frac{\mathrm{d}t^2}{2}(\rho_{tt})_k^i + \dots
\end{aligned}
$$

where $x_k = a_0 + k\,\mathrm{d}x$ and $t_i = i\,\mathrm{d}t$, $i = 0, 1, 2, \dots$, $k = 0, 1, 2, \dots$.

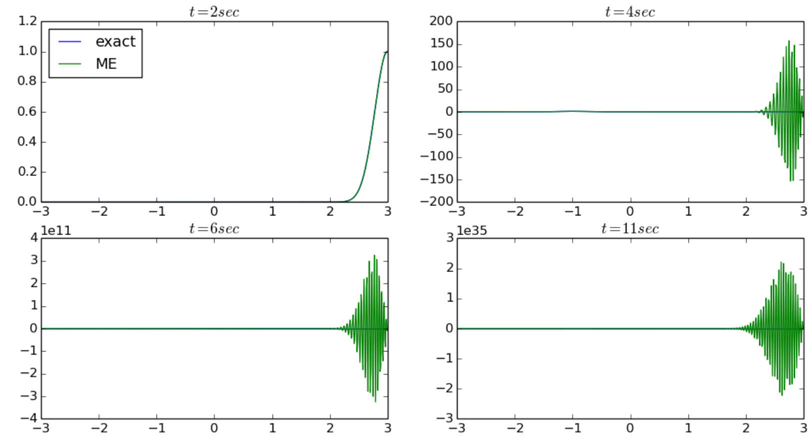

Figure 1.5: Exact (blue) solution and Modified Euler (green) solution of test problem (1.24) for $\varphi(x,t)$ in time points $t = 2$ (upper left), $t = 4$ (upper right), $t = 6$ (lower left), $t = 11$ (lower right). Horizontal axis is x.

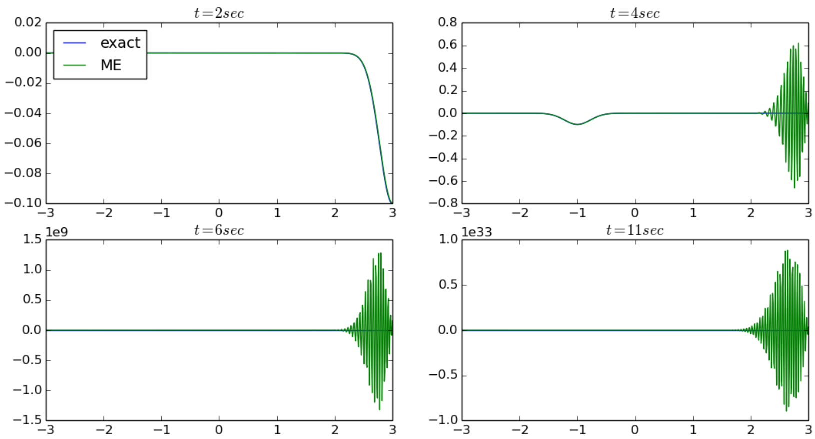

Figure 1.6: Exact (blue) solution and Modified Euler (green) solution of test problem (1.24) for $\rho(x,t)$ in time points $t = 2$ (upper left), $t = 4$ (upper right), $t = 6$ (lower left), $t = 11$ (lower right). Horizontal axis is x.

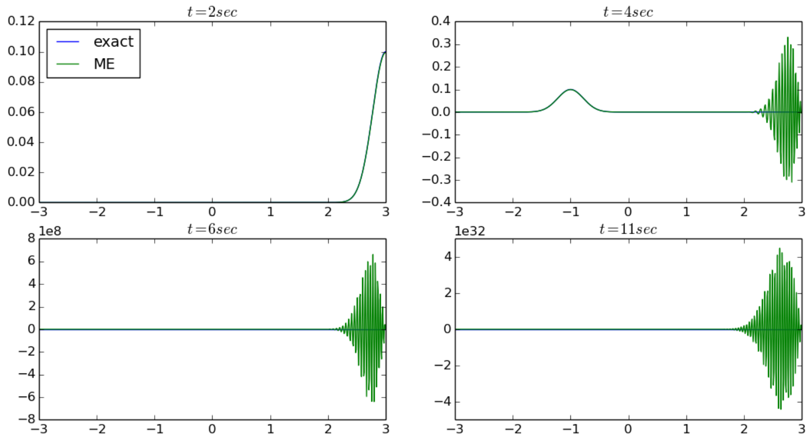

Figure 1.7: Exact (blue) solution and Modified Euler (green) solution of test problem (1.24) for $j(x,t)$ in time points $t = 2$ (upper left), $t = 4$ (upper right), $t = 6$ (lower left), $t = 11$ (lower right). Horizontal axis is x.

The differential equation can now be used to eliminate the t-derivatives

$$\begin{aligned}
\varphi_k^{i+1} &= \varphi_k^i + \mathrm{d}t(c\varphi_x + j + f_1(x,t))_k^i + \frac{\mathrm{d}t^2}{2}(c\varphi_{xt} + j_t + f_{1t}(x,t))_k^i \\
&= \varphi_k^i + \mathrm{d}t(c\varphi_x + j + f_1(x,t))_k^i + \frac{\mathrm{d}t^2}{2}(c\varphi_{tx} + (\alpha\varphi - \gamma j + f_3(x,t)) + f_{1t}(x,t))_k^i \\
&= \varphi_k^i + \mathrm{d}t(c\varphi_x + j + f_1(x,t))_k^i + \frac{\mathrm{d}t^2}{2}(c(c\varphi_x + j + f_1(x,t))_x \\
&+ (\alpha\varphi - \gamma j + f_3(x,t)) + f_{1t}(x,t))_k^i \\
&= \varphi_k^i + \mathrm{d}t(c\varphi_x + j + f_1(x,t))_k^i + \frac{\mathrm{d}t^2}{2}(c^2\varphi_{xx} + cj_x + cf_{1x}(x,t) \\
&+ (\alpha\varphi - \gamma j + f_3(x,t)) + f_{1t}(x,t))_k^i
\end{aligned}$$

for φ and for ρ

$$\begin{aligned}
\rho_k^{i+1} &= \rho_k^i + \mathrm{d}t(-j_x + f_2(x,t))_k^i + \frac{\mathrm{d}t^2}{2}(-j_{xt} + f_{2t}(x,t))_k^i \\
&= \rho_k^i + \mathrm{d}t(-j_x + f_2(x,t))_k^i + \frac{\mathrm{d}t^2}{2}(-j_{tx} + f_{2t}(x,t))_k^i \\
&= \rho_k^i + \mathrm{d}t(-j_x + f_2(x,t))_k^i + \frac{\mathrm{d}t^2}{2}(-(\alpha\varphi - \gamma j + f3(x,t))_x + f_{2t}(x,t))_k^i \\
&= \rho_j^i + \mathrm{d}t(-j_x + f_2(x,t))_k^i + \frac{\mathrm{d}t^2}{2}(-\alpha\varphi_x + \gamma j_x - f_{3x}(x,t) + f_{2t}(x,t))_k^i
\end{aligned}$$

Finally, the replacement of the x-derivatives by central-difference approximations gives, to terms in dt^2, the explicit difference equation

$$\begin{aligned}
\varphi_k^{i+1} &= \varphi_k^i + \mathrm{d}t\Big(\frac{c}{2\,\mathrm{d}x}\big(\varphi_{k+1}^i - \varphi_{k-1}^i\big) + j_k^i + f_1(x_k, t_i)\Big) \\
&+ \frac{\mathrm{d}t^2}{2}\Big(c^2\frac{1}{\mathrm{d}x^2}(\varphi_{k+1}^i - 2\varphi_k^i + \varphi_{k-1}^i) + c\frac{1}{2\,\mathrm{d}x}(j_{k+1}^i - j_{k-1}^i) \\
&+ \alpha\varphi_k^i - \gamma j_k^i + cf_{1x}(x_k, t_i) + f_3(x_k, t_i)\Big) \\
\rho_k^{i+1} &= \rho_k^i + \mathrm{d}t\Big(-\frac{1}{2\,\mathrm{d}x}\big(j_{k+1}^i - j_{k-1}^i\big) + f_2(x_k, t_i)\Big) \\
&+ \frac{\mathrm{d}t^2}{2}\Big(-\frac{\alpha}{2\,\mathrm{d}x}(\varphi_{k+1}^i - \varphi_{k-1}^i) + \frac{\gamma}{2\,\mathrm{d}x}(j_{k+1}^i - j_{k-1}^i) \\
&+ f_{2t}(x_k, t_i) - f_{3x}(x_k, t_i)\Big)
\end{aligned} \tag{1.31}$$

Also we need special formulas for calculation of ρ_t on the boundaries

$$\begin{aligned}
\rho_0^{i+1} &= \rho_0^i + \mathrm{d}t\Big(-\frac{1}{2\,\mathrm{d}x}\big(-3j_0^i + 4j_1^i - j_2^i\big) + f_2(x_0, t_i)\Big) \\
&+ \frac{\mathrm{d}t^2}{2}\Big(-\frac{\alpha}{2\,\mathrm{d}x}(-3\varphi_0^i + 4\varphi_1^i - \varphi_2^i) + \frac{\gamma}{2\,\mathrm{d}x}(-3j_0^i + 4j_1^i - j_2^i) \\
&+ f_{2t}(x_0, t_i) - f_{3x}(x_0, t_i)\Big) \\
\rho_{N+1}^{i+1} &= \rho_{N+1}^i + \mathrm{d}t\Big(-\frac{1}{2\,\mathrm{d}x}\big(3j_{N+1}^i - 4j_N^i + j_{N-1}^i\big) + f_2(x_{N+1}, t_i)\Big) \\
&+ \frac{\mathrm{d}t^2}{2}\Big(-\frac{\alpha}{2\,\mathrm{d}x}(3\varphi_{N+1}^i - 4\varphi_N^i + \varphi_{N-1}^i) + \frac{\gamma}{2\,\mathrm{d}x}(3j_{N+1}^i - 4j_N^i + j_{N-1}^i\big)
\end{aligned} \tag{1.32}$$

Also we will use Central Finite Difference method to calculate values of $j(x, t)$ in points of grid, so

$$j_k^{i+1} = 2\,\mathrm{d}t(\alpha\varphi_k^i - \gamma j_k^i + f_3(x_k, t_i)) + j_k^{i-1}$$

Now we can apply this numerical method to solve problem (1.24) for the same example functions like before. On Figures 1.8 – 1.10 we can see numerical solutions to (1.24) compare to the exact solutions. They overlap perfectly until the signal gone.

In this test $T = 12$, $a_0 = -3$, $a_1 = 3$,$c = 0.5$, $\alpha = \gamma = 0.1$, $\mathrm{d}t = 0.01$, $\mathrm{d}x = 0.01$.

Again we can see the artefact on the left boundary in solution to ρ, which possibly appears because of the same nature as before in Central Finite Difference method solution.

ODE solver In this paragraph we are going to apply ODE solver `odeint` from Python library SciPy [1]. This ODE solver integrates a system of ordinary differential equations and solves ordinary differential equations using Isoda from the FORTRAN library odepack. So we need to rewrite our problem (1.24) to system of ODE.

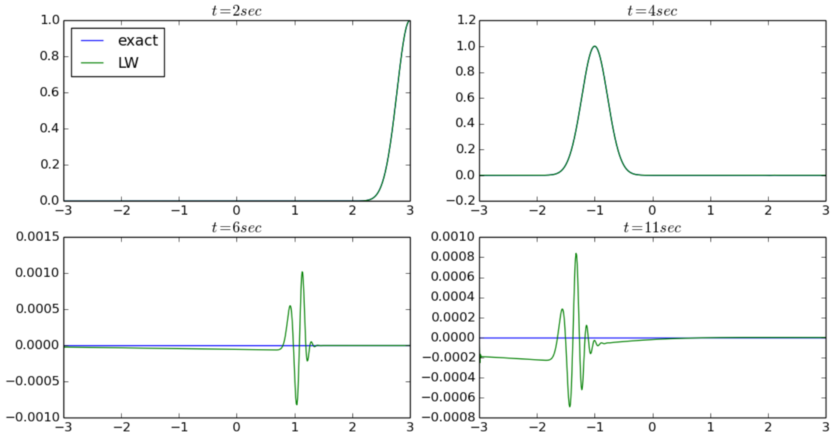

Figure 1.8: Exact (blue) solution and Lax-Wendroff numerical (green) solution of test problem (1.24) for $\varphi(x,t)$ in time points $t = 2$ (upper left), $t = 4$ (upper right), $t = 6$ (lower left), $t = 11$ (lower right). Horizontal axis is x.

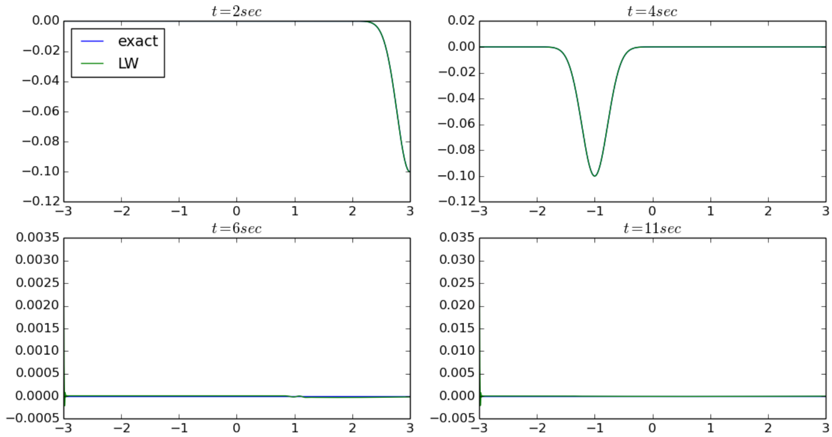

Figure 1.9: Exact (blue) solution and Lax-Wendroff numerical (green) solution of test problem (1.24) for $\rho(x,t)$ in time points $t = 2$ (upper left), $t = 4$ (upper right), $t = 6$ (lower left), $t = 11$ (lower right). Horizontal axis is x.

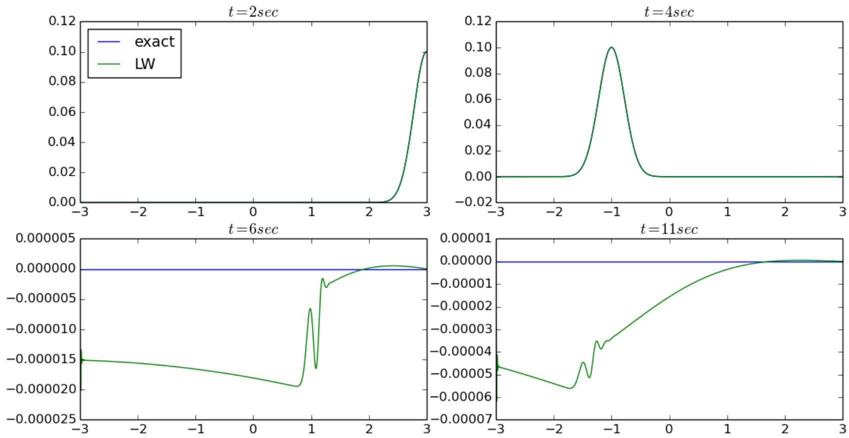

Figure 1.10: Exact (blue) solution and Lax-Wendroff numerical (green) solution of test problem (1.24) for $j(x,t)$ in time points $t = 2$ (upper left), $t = 4$ (upper right), $t = 6$ (lower left), $t = 11$ (lower right). Horizontal axis is x.

Let us introduce space grid $x_0, \dots, x_{N+1}$, where $x_0 = a_0$, $x_{N+1} = a_1$. Also let

$$\begin{aligned} \underline{\varphi} &= (\varphi_1, \dots, \varphi_N) \\ \underline{\rho} &= (\rho_0, \dots, \rho_{N+1}) \\ \underline{j} &= (j_0, \dots, j_{N+1}) \end{aligned}$$

where $\varphi_i = \varphi(t, x_i)$, $i = 1, \dots, N$, $\rho_k = \rho(t, x_k)$, $j_k = j(t, x_k)$, $k = 0, \dots, N+1$. We didn't include $\varphi_0 = \varphi(t, x_0)$ and $\varphi_{N+1} = \varphi(t, x_{N+1})$ to system as unknown functions because they are known from boundary conditions.

Also to rewrite the problem (1.24) to ODE system we need to change x - derivatives by approximations. Let us use Central Finite Difference rule for φ_x, j_x and special polynomial approximation of j_x for $\rho_t(t, a_0)$ and $\rho_t(t, a_1)$ as before in Central Finite Difference method.

Now we can rewrite the problem as

$$\begin{aligned} \underline{\varphi}_t &= M_1 \underline{\varphi} + \underline{j} + \underline{f}_1(t) \\ \underline{\rho}_t &= M_2 \underline{j} + \underline{f}_2(t) \\ \underline{j}_t &= \alpha \underline{\varphi} - \gamma \underline{j} + \underline{f}_3(t) \end{aligned}$$

where $\underline{f}_1(t) = (f_1(x_1, t), \dots, f_1(x_N, t))$, $\underline{f}_i(t) = (f_i(a_0, t), f_i(x_1, t), \dots, f_i(x_N, t), f_i(a_1, t))$,

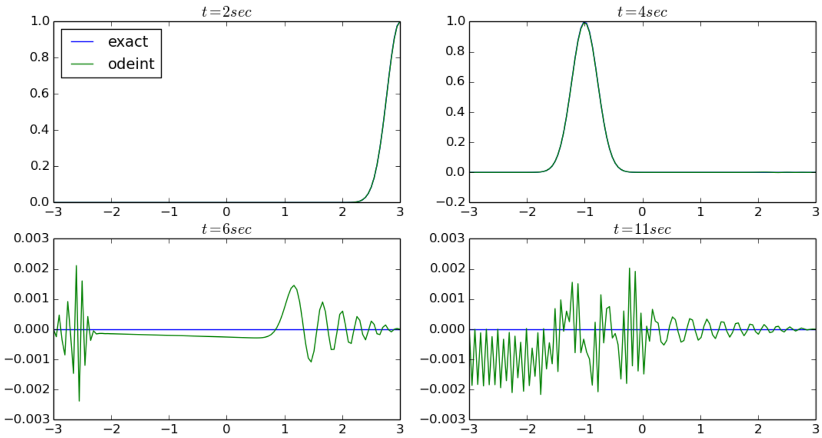

Figure 1.11: Exact (blue) solution and `odeint` numerical (green) solution of test problem (1.24) for $\varphi(x,t)$ in time points $t = 2$ (upper left), $t = 4$ (upper right), $t = 6$ (lower left), $t = 11$ (lower right). Horizontal axis is x.

$i = 2, 3$, M_1 is $N \times N$ matrix

$$M_1 = \frac{c}{2\,\mathrm{d}x}\begin{pmatrix} 0 & 1 & 0 & 0 & \dots \\ -1 & 0 & 1 & 0 & \dots \\ & \ddots & \ddots & \ddots & \\ 0 & \dots & -1 & 0 & 1 \\ 0 & \dots & 0 & -1 & 0 \end{pmatrix}$$

and M_2 is $(N+2) \times (N+2)$ matrix

$$M_2 = \frac{1}{2\,\mathrm{d}x}\begin{pmatrix} 3 & -4 & 1 & 0 & \dots & 0 \\ 1 & 0 & -1 & 0 & & \vdots \\ 0 & 1 & 0 & -1 & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 & 0 & -1 \\ 0 & \dots & 0 & -1 & 4 & -3 \end{pmatrix}$$

Let us now apply the constructed scheme and use system as argument to `odeint`. In this test $T = 12$, $a_0 = -3$, $a_1 = 3$, $c = 0.5$, $\alpha = \gamma = 0.1$, $\mathrm{d}t = 0.01$, $\mathrm{d}x = 0.05$. We take $\mathrm{d}x = 0.05$ instead of $\mathrm{d}x = 0.01$ because the ODE solver works slow and the process of calculations takes several hours. We can see on Figures 1.11 – 1.13 solutions to the (1.24) by `odeint`. There are some artefacts in solution of ρ on the both boundaries. Actually

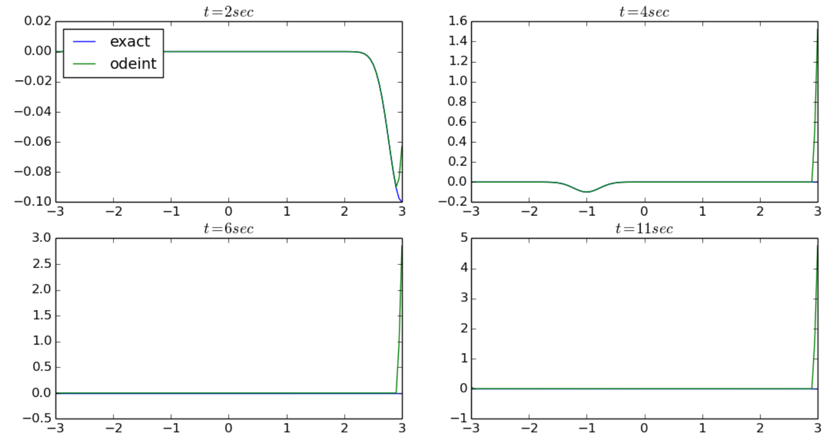

Figure 1.12: Exact (blue) solution and `odeint` numerical (green) solution of test problem (1.24) for $\rho(x,t)$ in time points $t = 2$ (upper left), $t = 4$ (upper right), $t = 6$ (lower left), $t = 11$ (lower right). Horizontal axis is x.

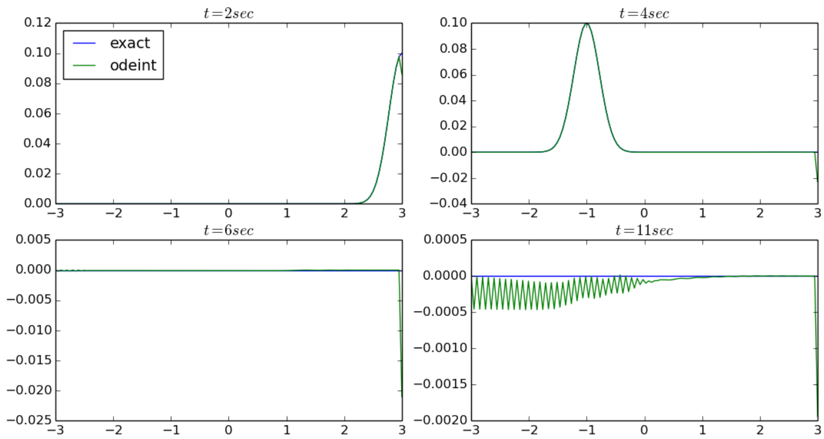

Figure 1.13: Exact (blue) solution and `odeint` numerical (green) solution of test problem (1.24) for $j(x,t)$ in time points $t = 2$ (upper left), $t = 4$ (upper right), $t = 6$ (lower left), $t = 11$ (lower right). Horizontal axis is x.

Table 1.1: Compare of numerical solutions

Method	**Maximum difference**			**Process time (sec)**
	φ	ρ	j	
CD	0.005377	0.000655	0.000626	47.213
LW	0.005022	0.003481	0.000565	178.616
`odeint`	0.007259	0.076554	0.000626	2764.312

artefact on the left boundary appeared in CD and LW methods. Possibly artefact on the right boundary appears because of the similar problem.

Let us now compare process time and maximum difference of numerical solutions which were founded by Central Finite Difference method, Lax-Wendroff method and ODE solver from exact solution. We need to remind that for Central Finite Difference method and Lax-Wendroff method we used $\mathrm{d}x = 0.01$ and for `odeint` we used $\mathrm{d}x = 0.05$ because of slow process of calculations.

Also we didn't include Modified Euler method because we saw that it is unstable.

We didn't include difference between exact solutions and numerical methods on the boundaries because of artefacts. So in Table 1.1 we compare difference only inside the domain.

We can see that Central Finite Difference method and Lax-Wendroff method are more accurate, but they have big difference in ρ. But also Lax-Wendroff method is more stable than Central Finite Difference method [7]. On Figure 1.14 we can see exact solution, numerical solution by Central Finite Difference method and numerical solution by Lax-Wendroff of test problem (1.24) with parameters $T = 12$, $a_0 = -3$, $a_1 = 3$, $dx = 0.05$, $dt = 0.01$, $c = 2$ for φ in several time points. We can see that for the same parameters Central Finite Difference method solution has oscillations unlike solution by Lax-Wendroff method.

Solution by ODE solver is less accurate because we used $\mathrm{d}x = 0.05$ instead of $\mathrm{d}x = 0.01$ but unfortunately this solver too slow to take another value. So we are going to use Lax-Wendroff method to solve original Toy model (1.1).

Numerical solution of the first toy problem

In this section we are going to apply Lax-Wendroff numerical method to model (1.1).

Let us first rewrite numerical scheme which we used for test problem by taking $c = c_1$

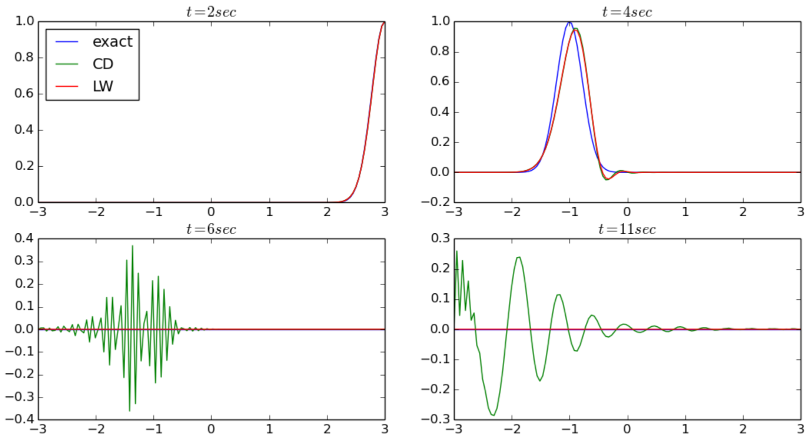

Figure 1.14: Exact (blue) solution, Central Finite Difference method numerical solution (green) and Lax-Wendroff method numerical solution (red) of test problem (1.24) for φ in time points $t = 2$ (upper left), $t = 4$ (upper right), $t = 6$ (lower left), $t = 11$ (lower right). Horizontal axis is x.

and $f_1(x,t) = f_2(x,t) = f_3(x,t) = 0$

$$
\begin{aligned}
\varphi_k^{i+1} &= \varphi_k^i + \mathrm{d}t\Big(\frac{c_1}{2\,\mathrm{d}x}(\varphi_{k+1}^i - \varphi_{k-1}^i) + j_k^i\Big)\\
&+ \frac{\mathrm{d}t^2}{2}\Big(\frac{c_1^2}{\mathrm{d}x^2}(\varphi_{k+1}^i - 2\varphi_k^i + \varphi_{k-1}^i) + \frac{c_1}{2\,\mathrm{d}x}(j_{k+1}^i - j_{k-1}^i)\\
&+ (\alpha - \beta\rho_k^i)\varphi_k^i - \gamma j_k^i\Big)\\
\rho_k^{i+1} &= \rho_k^i + \mathrm{d}t\Big(-\frac{1}{2\,\mathrm{d}x}(j_{k+1}^i - j_{k-1}^i)\Big)\\
&+ \frac{\mathrm{d}t^2}{2}\Big(\frac{\beta}{2\,\mathrm{d}x}(\rho_{k+1}^i - \rho_{k-1}^i)\varphi_k^i - \frac{1}{2\,\mathrm{d}x}(\alpha - \beta\rho_k^i)(\varphi_{k+1}^i - \varphi_{k-1}^i)\\
&+ \gamma\frac{1}{2\,\mathrm{d}x}(j_{k+1}^i - j_{k-1}^i)\Big)
\end{aligned}
\tag{1.33}
$$

and special rules for calculation values of ρ_t on the boundaries

$$\begin{aligned}
\rho_0^{i+1} &= \rho_0^i + \mathrm{d}t\Big(-\frac{1}{2\,\mathrm{d}x}\big(-3j_0^i + 4j_1^i - j_2^i\big)\Big) \\
&+ \frac{\mathrm{d}t^2}{2}\Big(\frac{\beta}{2\,\mathrm{d}x}(-3\rho_0^i + 4\rho_1^i - \rho_2^i)\varphi_0^i \\
&- \frac{1}{2\,\mathrm{d}x}(\alpha - \beta\rho_0^i)(-3\varphi_0^i + 4\varphi_1^i - \varphi_2^i) \\
&+ \frac{\gamma}{2\,\mathrm{d}x}(-3j_0^i + 4j_1^i - j_2^i)\Big) \\
\rho_{N+1}^{i+1} &= \rho_{N+1}^i + \mathrm{d}t\Big(-\frac{1}{2\,\mathrm{d}x}(3j_{N+1}^i - 4j_N^i + j_{N-1}^i)\Big) \\
&+ \frac{\mathrm{d}t^2}{2}\Big(\frac{\beta}{2\,\mathrm{d}x}(3\rho_{N+1}^i - 4\rho_N^i + \rho_{N-1}^i)\varphi_{N+1}^i \\
&- \frac{1}{2\,\mathrm{d}x}(\alpha - \beta\rho_{N+1}^i)(3\varphi_{N+1}^i - 4\varphi_N^i + \varphi_{N-1}^i) \\
&+ \frac{\gamma}{2\,\mathrm{d}x}(3j_{N+1}^i - 4j_N^i + j_{N-1}^i)\Big)
\end{aligned} \tag{1.34}$$

Also we will use Central Finite Difference method to calculate values of $j(x,t)$ in points of grid, so

$$j_k^{i+1} = 2\,\mathrm{d}t\big((\alpha - \beta\rho_k^i)\varphi_k^i - \gamma j_k^i\big) + j_k^{i-1}$$

Let for example take

$$j_s(x,t) = ae^{-b(x-x_0)^2}e^{-c(t-t_0)^2}$$

where $a = 2$, $b = c = 10$, $x_0 = 10$ and $t_0 = 4$. Also we take $T = 12$, $a_0 = -3$, $a_1 = 3$, $dt = 0.01$, $dx = 0.01$, $\alpha = \beta = \gamma = 0.1$, $c_1 = 1.4$, $c_2 = 2$. Numerical solution of the First toy problem with described parameters is on Figures 1.15 – 1.17 Again we can see artefact on the left boundary of ρ.

1.2 The second toy model

Our second toy model is

$$\begin{aligned}
\varphi_t &= \mu(x)\psi_x + j + j_s \\
\psi_t &= \nu(x)\varphi_x \\
\rho_t &= -j_x \\
j_t &= (\alpha - \beta\rho)\varphi - \gamma j
\end{aligned} \tag{1.35}$$

Where $\varphi = \varphi(x,t)$ is the "electric field", $\psi = \psi(x,t)$ is the "magnetic field", $j = j(x,t)$ is the "current density", $\rho = \rho(x,t)$ is the "charge density". These quantities are analogues to the corresponding quantities in the Maxwell equation. The charge density and current

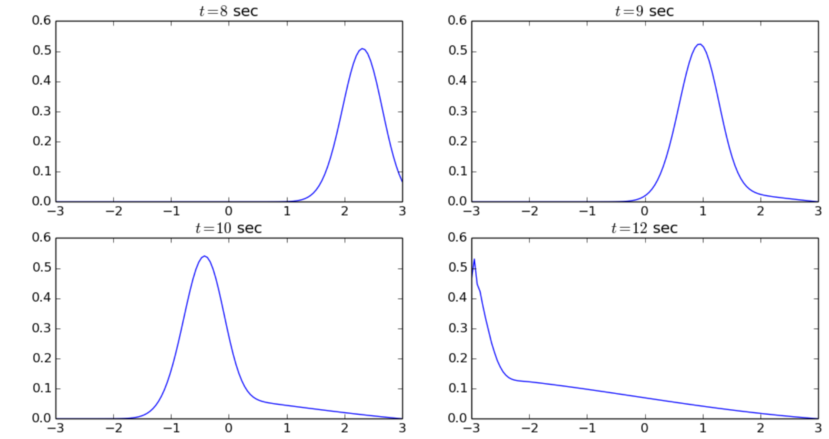

Figure 1.15: Numerical solution of model (1.1) for φ in time points $t = 8$ (upper left), $t = 9$ (upper right), $t = 10$ (lower left), $t = 12$ (lower right). Horizontal axis is x.

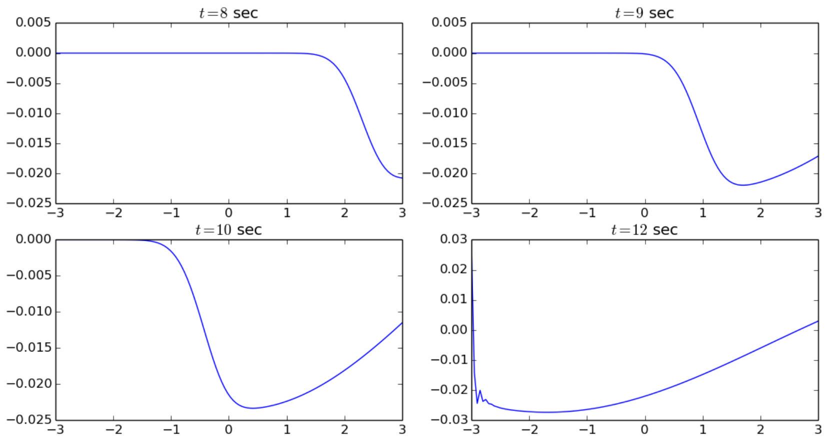

Figure 1.16: Numerical solution of model (1.1) for ρ in time points $t = 8$ (upper left), $t = 9$ (upper right), $t = 10$ (lower left), $t = 12$ (lower right). Horizontal axis is x.

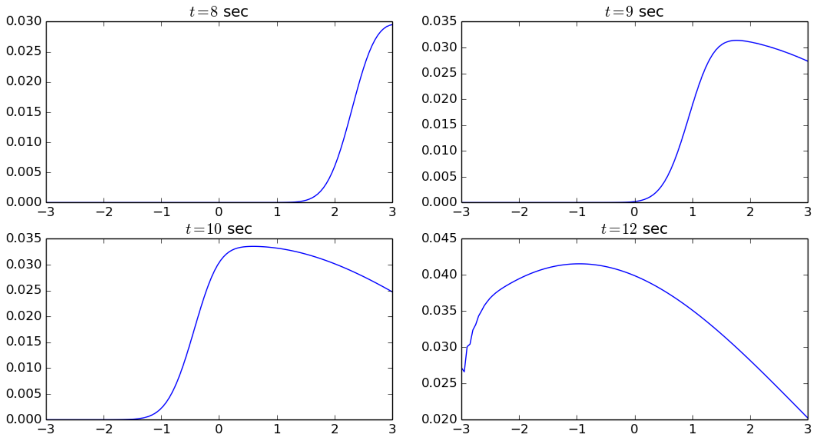

Figure 1.17: Numerical solution of model (1.1) for j in time points $t = 8$ (upper left), $t = 9$ (upper right), $t = 10$ (lower left), $t = 12$ (lower right). Horizontal axis is x.

density will be assumed to be confined to an interval $[a_0, a_1]$ on the real axis whereas the fields φ and ψ are continuous on the whole real axis. This interval is, like for the first toy model, the analogue of a compact scattering object in the electromagnetic situation. The function $j_s(x, t)$ is a given source that has support on a compact set in the interval $x > a_1$. The parameters α, β and γ are constant inside the interval $[a_0, a_1]$ and ero outside. The functions $\mu(x)$ and $\nu(x)$ are "material" parameters analogous to electric and magnetic susceptibilities in the electromagnetic case.

$$\mu(x) = \begin{cases} \mu_1 & x \in [a_0, a_1] \\ \mu_2 & x \notin [a_0, a_1] \end{cases}$$

$$\nu(x) = \begin{cases} \nu_1 & x \in [a_0, a_1] \\ \nu_2 & x \notin [a_0, a_1] \end{cases}$$

Graphical representation of the Second Toy model is shown on Figure 1.18.

1.2.1 The EOS formulation

In order to derive the EOS formulation for model (1.35), we will firstly need a space-time integral identity involving the matrix operator

$$L = \begin{pmatrix} \partial_t & -\mu\partial_x \\ -\nu\partial_x & \partial_t \end{pmatrix} \tag{1.36}$$

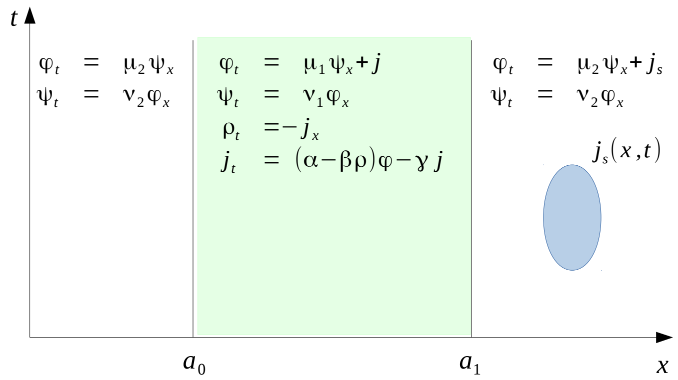

Figure 1.18: Graphical representation of the second toy model. The blue ellipse represents the source $j_s(x,t)$ outside the domain bounded by $x = a_0$ and $x = a_1$.

where μ, ν are constants. The operator (1.36) acts on vector valued functions in the usual way and we can write the first two equations from (1.35) in the compact form

$$L\begin{pmatrix}\varphi\\ \psi\end{pmatrix}(x,t) = \begin{pmatrix}j + j_s\\ 0\end{pmatrix}(x,t) \tag{1.37}$$

Let $A = \begin{pmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\end{pmatrix}$ is a matrix of functions $A(x,y,t)$

$$\begin{aligned}\int_{S\times T} dxdtAL\begin{pmatrix}\varphi\\ \psi\end{pmatrix} &= \int_{S\times T} dxdt\begin{pmatrix}A_{11}(\varphi_t - \mu\psi_x) + A_{12}(-\nu\varphi_x + \psi_t)\\ A_{21}(\varphi_t - \mu\psi_x) + A_{22}(-\nu\varphi_x + \psi_t)\end{pmatrix}\\ &= \int_{S\times T} dxdt\begin{pmatrix}-\varphi\partial_t A_{11} + \mu\psi\partial_x A_{11} + \nu\varphi\partial_x A_{12} - \psi\partial_t A_{12}\\ -\varphi\partial_t A_{21} + \mu\psi\partial_x A_{21} + \nu\varphi\partial_x A_{22} - \psi\partial_t A_{22}\end{pmatrix}\\ &+ \int_S dx\begin{pmatrix}A_{11}\varphi + \psi A_{12}\\ A_{21}\varphi + \psi A_{22}\end{pmatrix}\Bigg|_{t_0}^{t} + \int_T dt\begin{pmatrix}-\mu\psi A_{11} - \nu\varphi A_{12}\\ -\mu\psi A_{21} - \nu\varphi A_{22}\end{pmatrix}\Bigg|_{x_0}^{x_1}\\ &= \int_{S\times T} dxdtL^\dagger A\begin{pmatrix}\varphi\\ \psi\end{pmatrix} + \int_S dxA\begin{pmatrix}\varphi\\ \psi\end{pmatrix}\Bigg|_{t_0}^{t} + \int_T dtB\begin{pmatrix}\varphi\\ \psi\end{pmatrix}\Bigg|_{x_0}^{x_1}\end{aligned}$$

So we have integral identity

$$\int_{S\times T} dxdt\left[AL\begin{pmatrix}\varphi\\ \psi\end{pmatrix} - L^\dagger A\begin{pmatrix}\varphi\\ \psi\end{pmatrix}\right] = \int_S dxA\begin{pmatrix}\varphi\\ \psi\end{pmatrix}\Bigg|_{t_0}^{t_1} - \int_T dxB\begin{pmatrix}\varphi\\ \psi\end{pmatrix}\Bigg|_{x_0}^{x_1} \tag{1.38}$$

where $S = (x_0, x_1)$ and $T = (t_0, t_1)$ are open space and time intervals and where φ and ψ are smooth functions on the space-time interval $S \times T$. Also $A = A(x,t)$ is a 2×2 matrix valued function

$$A = \begin{pmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\end{pmatrix}$$

$L^\dagger$ is the formal adjoint to the operator L, and acts on the matrix valued function A in the following way

$$L^\dagger A = \begin{pmatrix} -\partial_t A_{11} + \nu\partial_x A_{12} & \mu\partial_x A_{11} - \partial_t A_{12} \\ -\partial_t A_{21} + \nu\partial_x A_{22} & \mu\partial_x A_{21} - \partial_t A_{22} \end{pmatrix} \tag{1.39}$$

B is the 2×2 matrix valued function

$$B = \begin{pmatrix} -\nu A_{12} & -\mu A_{11} \\ -\nu A_{22} & -\mu A_{21} \end{pmatrix} \tag{1.40}$$

The second item we need in order to derive the EOS formulation for model (1.35), is the advanced Green's function for the operator $L^\dagger$. This is a 2×2 matrix valued function $G(x, t, x', t')$ that satisfy the equation

$$L^\dagger G(x, t, x', t') = \delta(t - t')\delta(x - x')I \tag{1.41}$$

and that vanish for $t > t'$. In (1.41), I is the 2×2 identity matrix.

Using (1.39) in (1.41) we have the following system of four equations for the components G.

$$\begin{aligned} &\partial_t G_{11} - \nu\partial_x G_{12} = -\delta(t - t')\delta(x - x') \\ &\partial_t G_{12} - \mu\partial_x G_{11} = 0 \\ &\partial_t G_{21} - \nu\partial_x G_{22} = 0 \\ &\partial_t G_{22} - \mu\partial_x G_{21} = -\delta(t - t')\delta(x - x') \end{aligned} \tag{1.42}$$

We use the one dimensional Fourier transform in x to solve the initial value problem for 1.42. Let the Fourier transform of G_{ij} be denoted by $g_{ij}(\lambda, t, x', t')$. Then we get system

$$\begin{aligned} &\partial_t g_{11} - \nu\lambda g_{12} = -\frac{1}{\sqrt{2\pi}} e^{i\lambda x}\delta(t - t') \\ &\partial_t g_{12} - \mu\lambda g_{11} = 0 \\ &\partial_t g_{21} - \nu\lambda g_{22} = 0 \\ &\partial_t g_{22} - \mu\lambda g_{21} = -\frac{1}{\sqrt{2\pi}} e^{i\lambda x}\delta(t - t') \end{aligned} \tag{1.43}$$

If we solve system 1.43 we get

$$\begin{aligned} G_{11} &= \frac{1}{2}\theta(t' - t)\Big[\delta(x' - x + \sqrt{\mu\nu}(t - t')) - \delta(x' - x - \sqrt{\mu\nu}(t - t'))\Big] \\ G_{12} &= \frac{1}{2}\sqrt{\frac{\mu}{\nu}}\theta(t' - t)\Big[\delta(x' - x + \sqrt{\mu\nu}(t - t')) - \delta(x' - x - \sqrt{\mu\nu}(t - t'))\Big] \\ G_{21} &= \frac{1}{2}\sqrt{\frac{\nu}{\mu}}\theta(t' - t)\Big[\delta(x' - x + \sqrt{\mu\nu}(t - t')) + \delta(x' - x - \sqrt{\mu\nu}(t - t'))\Big] \\ G_{22} &= \frac{1}{2}\theta(t' - t)\Big[\delta(x' - x + \sqrt{\mu\nu}(t - t')) - \delta(x' - x - \sqrt{\mu\nu}(t - t'))\Big] \end{aligned} \tag{1.44}$$

So we have

$$G(x,t,x',t') = -\frac{\theta(t'-t)}{2c}\{\begin{pmatrix} c & \mu \\ \nu & c \end{pmatrix}\delta(x-x'+c(t-t')) + \begin{pmatrix} c & -\mu \\ -\nu & c \end{pmatrix}\delta(x-x'-c(t-t'))\} \tag{1.45}$$

where $c^2 = \mu\nu$. This is analogue to the corresponding identity for the speed of light in electromagnetics.

We will now apply the integral identity (1.38) to each space interval $(-\infty, a_0)$, (a_0, a_1), (a_1, ∞) with A equal to the advances Green's function (1.45) and where φ and ψ are solutions to the system

$$\begin{aligned} \varphi_t &= \mu(x)\psi_x + j + j_s \\ \psi_t &= \nu(x)\varphi_x \end{aligned} \tag{1.46}$$

with vanishing initial conditions $\varphi(x,t_0) = \psi(x,t_0) = 0$.

For the first interval, $(-\infty, a_0)$, we let A be the Green's function

$$G_2(x,t,x',t') = -\frac{\theta(t'-t)}{2c_2}\{\begin{pmatrix} c_2 & \mu_2 \\ \nu_2 & c_2 \end{pmatrix}\delta(x-x'+c_2(t-t')) + \begin{pmatrix} c_2 & -\mu_2 \\ -\nu_2 & c_2 \end{pmatrix}\delta(x-x'-c_2(t-t'))\} \tag{1.47}$$

According to the described properties of j and j_s, the function $\varphi = \varphi_2$, $\psi = \psi_2$ solves, in the interval $(-\infty, a_0)$, the system

$$\begin{aligned} \varphi_{2t} &= \mu_2\psi_{2x} \\ \psi_{2t} &= \nu_2\varphi_{2x} \\ &\Updownarrow \\ L\begin{pmatrix} \varphi_2 \\ \psi_2 \end{pmatrix} &= 0 \end{aligned} \tag{1.48}$$

Inserting (1.47), (1.48) and $S = (-\infty, a_0)$ in the integral identity (1.38), using the initial conditions and the fact that the Green's function is advanced, we get fo x in the interval $(-\infty, a_0)$.

$$\int_{S\times T} dxdt\{G_2L\begin{pmatrix} \varphi_2 \\ \psi_2 \end{pmatrix}(x,t) - L^\dagger G_2\begin{pmatrix} \varphi_2 \\ \psi_2 \end{pmatrix}(x,t)\} = \int_S dxG_2\begin{pmatrix} \varphi_2 \\ \psi_2 \end{pmatrix}(x,t)\Big|_{t_0}^{t_1} + \int_T dtB_2\begin{pmatrix} \varphi_2 \\ \psi_2 \end{pmatrix}(x,t)\Big|_{-\infty}^{a_0}$$

So after interchanging primed and unprimed variables

$$\begin{aligned}\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(x,t) &= -\int_{t_0}^{t_1} dt' B_2(a_0,t',x,t)\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(a_0,t')\\ &+ \lim_{R\to-\infty}\int_{t_0}^{t_1} dt' B_2(R,t',x,t)\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(R,t')\end{aligned}\tag{1.49}$$

The function B_2 is from (1.40)

$$\begin{aligned}B_2(x',t',x,t) = -\frac{\theta(t-t')}{2}\{&\begin{pmatrix}c_2 & \mu_2\\ \nu_2 & c_2\end{pmatrix}\delta(x-x'+c_2(t-t'))\\ +&\begin{pmatrix}-c_2 & \mu_2\\ \nu_2 & -c_2\end{pmatrix}\delta(x-x'-c_2(t-t'))\}\end{aligned}\tag{1.50}$$

From (1.50) it is evident that the last term in (1.49) vanish. This is because for large R the argument of the delta function does not charge sign in the interval of integration. Inserting the expression (1.50) into (1.50) and changing to variable defining the argument of the delta function in the two integrals, we get that for x in $(-\infty, a_0)$

$$\begin{aligned}\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(x,t) &= -\int_{t_0}^{t_1} dt' B_2(a_0,t',x,t)\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(a_0,t')\\ &= \frac{1}{2}\begin{pmatrix}c_2 & \mu_2\\ \nu_2 & c_2\end{pmatrix}\int_{t_0}^{t_1} dt'\delta(x-a_0+c_2(t-t'))\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(a_0,t')\\ &= \frac{1}{2c_2}\begin{pmatrix}c_2 & \mu_2\\ \nu_2 & c_2\end{pmatrix}\theta(x-a_0+c_2(t-t'))\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(a_0,t+\frac{x-a_0}{c_2})\end{aligned}\tag{1.51}$$

For the second interval, (a_0, a_1), we let A be the Green's function

$$\begin{aligned}G_1(x,t,x',t') = -\frac{\theta(t'-t)}{2c_1}\{&\begin{pmatrix}c_1 & \mu_1\\ \nu_1 & c_1\end{pmatrix}\delta(x-x'+c_1(t-t'))\\ +&\begin{pmatrix}c_1 & -\mu_1\\ -\nu_1 & c_1\end{pmatrix}\delta(x-x'-c_1(t-t'))\}\end{aligned}\tag{1.52}$$

According to the described properties of j and j_s, the functions $\varphi = \varphi_1$, $\psi = \psi_1$ solves, in the interval (a_0, a_1), the system

$$\begin{aligned}\varphi_{1t} &= \mu_1\psi_{1x} + j\\ \psi_{1t} &= \nu_1\varphi_{1x}\\ &\Updownarrow\\ L\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix} &= \begin{pmatrix}j\\ 0\end{pmatrix}\end{aligned}\tag{1.53}$$

Inserting (1.52), (1.53) and $S = (a_0, a_1)$ in the integral identity (1.38), using the initial conditions and the fact that the Green's function is advanced, we get fo x in the interval (a_0, a_1).

$$\int_{S\times T} dxdt\{G_1 L \begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(x,t) - L^\dagger G_1 \begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(x,t)\} = \int_S dx G_1 \begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(x,t)\Big|_{t_0}^{t_1} + \int_T dt B_1 \begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(x,t)\Big|_{a_0}^{a_1}$$

So after interchanging primed and unprimed variables

$$\begin{aligned}\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(x,t) &= \int_{S\times T} dx'dt' G_1(x',t',x,t)\begin{pmatrix}j\\ 0\end{pmatrix}(x',t')\\ &- \int_{t_0}^{t_1} dt' B_1(a_1,t',x,t)\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_1,t')\\ &+ \int_{t_0}^{t_1} dt' B_1(a_0,t',x,t)\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_0,t')\end{aligned} \tag{1.54}$$

The function B_1 is from (1.40)

$$\begin{aligned}B_1(x',t',x,t) = -\frac{\theta(t-t')}{2}\{&\begin{pmatrix}c_1 & \mu_1\\ \nu_1 & c_1\end{pmatrix}\delta(x-x'+c_1(t-t'))\\ +&\begin{pmatrix}-c_1 & \mu_1\\ \nu_1 & -c_1\end{pmatrix}\delta(x-x'-c_1(t-t'))\}\end{aligned} \tag{1.55}$$

Inserting (1.52) and (1.55) into (1.54), we get after changing variables to the arguments in the delta functions that for x in (a_0, a_1)

$$\begin{aligned}
\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(x,t) = \frac{1}{2c_1^2}&\begin{pmatrix}c_1 & -\mu_1\\ -\nu_1 & c_1\end{pmatrix}\\
&\int_{a_0}^{x} dx'\theta(c_1(t-t_0)-(x-x'))\begin{pmatrix}j\\ 0\end{pmatrix}(x', t-\frac{x-x'}{c_1})\\
&+\begin{pmatrix}c_1 & \mu_1\\ \nu_1 & c_1\end{pmatrix}\\
&\int_{x}^{a_1} dx'\theta(c_1(t-t_0)-(x'-x))\begin{pmatrix}j\\ 0\end{pmatrix}(x', t-\frac{x'-x}{c_1})\\
&+\frac{1}{2c_1}\begin{pmatrix}c_1 & \mu_1\\ \nu_1 & c_1\end{pmatrix}\theta(c_1(t-t_0)-(a_1-x))\\
&\begin{pmatrix}\varphi_1\psi_1\end{pmatrix}(a_1, t-\frac{a_1-x}{c_1})\\
&-\frac{1}{2c_1}\begin{pmatrix}-c_1 & \mu_1\\ \nu_1 & -c_1\end{pmatrix}\theta(c_1(t-t_0)-(x-a_0))\\
&\begin{pmatrix}\varphi_1\psi_1\end{pmatrix}(a_1, t-\frac{x-a_0}{c_1})
\end{aligned} \tag{1.56}$$

For the third interval, (a_1, ∞), we let A be the Green's function

$$\begin{aligned}
G_2(x,t,x',t') = -\frac{\theta(t'-t)}{2c_2}\{&\begin{pmatrix}c_2 & \mu_2\\ \nu_2 & c_2\end{pmatrix}\delta(x-x'+c_2(t-t'))\\
&+\begin{pmatrix}c_2 & -\mu_2\\ -\nu_2 & c_2\end{pmatrix}\delta(x-x'-c_2(t-t'))\}
\end{aligned} \tag{1.57}$$

According to the described properties of j and j_s, the function $\varphi = \varphi_2$, $\psi = \psi_2$ solves, in the interval (a_1, ∞), the system

$$\begin{aligned}
\varphi_{2t} &= \mu_2\psi_{2x} + j_s\\
\psi_{2t} &= \nu_2\varphi_{2x}\\
&\Updownarrow\\
L\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix} &= \begin{pmatrix}j_s\\ 0\end{pmatrix}
\end{aligned} \tag{1.58}$$

Inserting (1.57), (1.58) and $S = (a_1, \infty)$ in the integral identity (1.38), using the initial conditions and the fact that the Green's function is advanced, we get fo x in the interval

(a_1, ∞).

$$\int_{S\times T} dxdt\{G_2 L \begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(x,t) - L^\dagger G_2 \begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(x,t)\} =$$
$$\int_S dx G_2 \begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(x,t)\Big|_{t_0}^{t_1} + \int_T dt B_2 \begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(x,t)\Big|_{a_1}^{\infty}$$

So after interchanging primed and unprimed variables

$$\begin{aligned}\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(x,t) &= \int_{S\times T} dx'dt' G_2(x',t',x,t)\begin{pmatrix}j_s\\ 0\end{pmatrix}(x',t')\\ &- \lim_{R\to\infty}\int_{t_0}^{t_1} dt' B_2(R,t',x,t)\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(R,t')\\ &+ \int_{t_0}^{t_1} dt' B_2(a_1,t',x,t)\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(a_1,t')\end{aligned} \tag{1.59}$$

The function B_2 is from (1.40)

$$\begin{aligned}B_2(x',t',x,t) = -\frac{\theta(t-t')}{2}\{&\begin{pmatrix}c_2 & \mu_2\\ \nu_2 & c_2\end{pmatrix}\delta(x-x'+c_2(t-t'))\\ &+\begin{pmatrix}-c_2 & \mu_2\\ \nu_2 & -c_2\end{pmatrix}\delta(x-x'-c_2(t-t'))\}\end{aligned} \tag{1.60}$$

Since the arguments of the delta functions in B_2 does not change sign in the interval of integration for R big enough, the second term in (1.59) will vanish. Inserting (1.57) and (1.60) into the remaining terms of (1.59), we get after changing variables to the arguments in the delta functions that for x in (a_1, ∞)

$$\begin{aligned}\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(x,t) &= \begin{pmatrix}\varphi_i\\ \psi_i\end{pmatrix}(x,t)\\ &- \frac{1}{2c_2}\begin{pmatrix}-c_2 & \mu_2\\ \nu_2 & -c_2\end{pmatrix}\theta(c_2(t-t_0)-(x-a_1))\\ &\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(a_1, t-\frac{x-a_1}{c_2})\end{aligned} \tag{1.61}$$

where φ_i and ψ_i are fields that are entirely determined by the given source j_s

$$\begin{aligned}\begin{pmatrix}\varphi_i\\ \psi_i\end{pmatrix}(x,t) &= \int_{S\times T} dx'dt' G_2(x',t',x,t)\begin{pmatrix}j_s\\ 0\end{pmatrix}(x',t')\\ &= \frac{1}{2c_2^2}\begin{pmatrix}c_2 & -\mu_2\\ -\nu_2 & c_2\end{pmatrix}\\ &\int_{a_1}^{x}\theta(c_2(t-t_0)-(x-x'))\begin{pmatrix}j_s\\ 0\end{pmatrix}(x',t-\frac{x-x'}{c_2})\\ &+\frac{1}{2c_2^2}\begin{pmatrix}c_2 & \mu_2\\ \nu_2 & c_2\end{pmatrix}\\ &\int_{x}^{\infty}\theta(c_2(t-t_0)-(x'-x))\begin{pmatrix}j_s\\ 0\end{pmatrix}(x',t-\frac{x'-x}{c_2})\end{aligned} \tag{1.62}$$

Taking the limit of the integral identities (1.51), (1.56) and (1.61) as x approach the boundary points $\{a_0, a_1\}$ from inside and outside the interval (a_0, a_1) we get

$$\begin{gathered}\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(a_0,t) = \frac{1}{2c_2}\begin{pmatrix}c_2 & \mu_2\\ \nu_2 & c_2\end{pmatrix}\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(a_0,t)\\ \Updownarrow\\ \begin{pmatrix}c_2 & -\mu_2\\ -\nu_2 & c_2\end{pmatrix}\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(a_0,t) = 0\end{gathered} \tag{1.63}$$

$$\begin{gathered}\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_0,t) = \frac{1}{2c_1^2}\begin{pmatrix}c_1 & \mu_1\\ \nu_1 & c_1\end{pmatrix}\\ \int_{a_0}^{a_1} dx'\theta(c_1(t-t_0)-(x'-a_0))\begin{pmatrix}j\\ 0\end{pmatrix}(x',t-\frac{x'-a_0}{c_1})\\ +\frac{1}{2c_1}\begin{pmatrix}c_1 & \mu_1\\ \nu_1 & c_1\end{pmatrix}\theta(c_1(t-t_0)-(a_1-a_0))\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_1,t-\frac{a_1-a_0}{c_1})\\ +\frac{1}{2c_1}\begin{pmatrix}c_1 & \mu_1\\ \nu_1 & c_1\end{pmatrix}\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_0,t)\\ \Updownarrow\\ \begin{pmatrix}c_1 & \mu_1\\ \nu_1 & c_1\end{pmatrix}\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_0,t) =\\ \frac{1}{c_1}\begin{pmatrix}c_1 & \mu_1\\ \nu_1 & c_1\end{pmatrix}\int_{a_0}^{a_1} dx'\theta(c_1(t-t_0)-(x'-a_0))\begin{pmatrix}j\\ 0\end{pmatrix}(x',t-\frac{x'-a_0}{c_1})\\ +\begin{pmatrix}c_1 & \mu_1\\ \nu_1 & c_1\end{pmatrix}\theta(c_1(t-t_0)-(a_1-a_0))\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_1,t-\frac{a_1-a_0}{c_1})\end{gathered} \tag{1.64}$$

$$
\begin{aligned}
&\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_1,t) = \frac{1}{2c_1^2}\begin{pmatrix}c_1 & -\mu_1\\ -\nu_1 & c_1\end{pmatrix}\\
&\int_{a_0}^{a_1} dx'\theta(c_1(t-t_0)-(a_1-x'))\begin{pmatrix}j\\ 0\end{pmatrix}(x',t-\frac{a_1-x'}{c_1})\\
&-\frac{1}{2c_1}\begin{pmatrix}-c_1 & \mu_1\\ \nu_1 & -c_1\end{pmatrix}\theta(c_1(t-t_0)-(a_1-a_0))\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_0,t-\frac{a_1-a_0}{c_1})\\
&+\frac{1}{2c_1}\begin{pmatrix}c_1 & \mu_1\\ \nu_1 & c_1\end{pmatrix}\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_1,t)\\
&\Updownarrow\\
&\begin{pmatrix}c_1 & -\mu_1\\ -\nu_1 & c_1\end{pmatrix}\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_1,t) =\\
&\frac{1}{c_1}\begin{pmatrix}c_1 & -\mu_1\\ -\nu_1 & c_1\end{pmatrix}\int_{a_0}^{a_1} dx'\theta(c_1(t-t_0)-(a_1-x'))\begin{pmatrix}j\\ 0\end{pmatrix}(x',t-\frac{a_1-x'}{c_1})\\
&-\begin{pmatrix}-c_1 & \mu_1\\ \nu_1 & -c_1\end{pmatrix}\theta(c_1(t-t_0)-(a_1-a_0))\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_0,t-\frac{a_1-a_0}{c_1})
\end{aligned}
\tag{1.65}
$$

$$
\begin{aligned}
&\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(a_1,t) = \begin{pmatrix}\varphi_i\\ \psi_i\end{pmatrix}(a_1,t) - \frac{1}{2c_2}\begin{pmatrix}-c_2 & \mu_2\\ \nu_2 & -c_2\end{pmatrix}\\
&\Updownarrow\\
&\begin{pmatrix}c_2 & \mu_2\\ \nu_2 & c_2\end{pmatrix}\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(a_1,t) = 2c_2\begin{pmatrix}\varphi_i\\ \psi_i\end{pmatrix}(a_1,t)
\end{aligned}
\tag{1.66}
$$

Continuity of the fields at the boundary points $\{a_0, a_1\}$, gives us two additional equations

$$\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(a_0,t) = \begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_0,t) \tag{1.67}$$

$$\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_1,t) = \begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(a_1,t) \tag{1.68}$$

Altogether we have six linear equations for the four vectors

$$\begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(a_0,t), \begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_0,t), \begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_1,t), \begin{pmatrix}\varphi_2\\ \psi_2\end{pmatrix}(a_1,t)$$

Thus our system is overdetermined and contains equation that are reduntant. Mathematically this is reflected in the fact that the determinant of the matrices

$$\begin{pmatrix}c_j & \pm\mu_j\\ \pm\nu_j & c_j\end{pmatrix}, j = 1, 2 \tag{1.69}$$

are all zero. For the first toy model it was obvious which two equations were redundant, here it is not immediately clear which equations we can remove, and this will also be the case when we write down the EOS formulation for more general systems of PDEs, like, for example, Maxwell equations.

For the system (1.63) - (1.66), it is not very hard to identify the redundant equations, but will rather introduce a different approach that is in general quite useful when working with the EOS formulations of PDEs. This is the method that has been used by the research community that calculate electromagnetic scattering from linear homogeneous scattering objects using a time dependent integral formulation of Maxwell's equations. The reason why this method has been used for the Maxwell equations has not been clearly stated in the research literature, it has rather taken the form of a trick that is needed in order achieve stability and accuracy for the numerical implementation of the boundary formulation of the electromagnetic scattering.

The point is that even though the system (1.63) - (1.66) is singular we know from its construction that it has a solution which consists of the boundary values coming from the unique solution to the scattering problem (1.46).

In terms of linear algebra the situation is that for two singular matrices A and B, the system

$$\begin{aligned} A\mathbf{x} &= \mathbf{b}_1 \\ B\mathbf{x} &= \mathbf{b}_2 \end{aligned} \tag{1.70}$$

has a solution, $\mathbf{x}$. Let us assume that there are numbers a and b such that

$$\det(aA + bB) \neq 0$$

Given (1.70) it is clear that $\mathbf{x}$ is a solution to the linear system

$$(aA + bB)\mathbf{x} = \mathbf{b}_1 + \mathbf{b}_2 \tag{1.71}$$

and since the system (1.71) is nonsingular, $\mathbf{x}$ is the unique solution to the system. Finding numbers such that $aA + bB$ is nonsingular is in general not difficult.

Let us apply the approach to the system (1.63) - (1.66). Simply adding together the equations give us a matrix

$$\begin{pmatrix} c_2 & -\mu_2 \\ -\nu_2 & c_2 \end{pmatrix} + \begin{pmatrix} c_1 & \mu_1 \\ \nu_1 & c_1 \end{pmatrix} = \begin{pmatrix} c_1 + c_2 & \mu_1 - \mu_2 \\ \nu_1 - \nu_2 & c_1 + c_2 \end{pmatrix}$$

and

$$\det \begin{pmatrix} c_1 + c_2 & \mu_1 - \mu_2 \\ \nu_1 - \nu_2 & c_1 + c_2 \end{pmatrix} = 2c_1c_2 + \mu_2\nu_1 + \mu_1\nu_2$$

which is nonzero since all the numbers ν_i, μ_i, c_i are positive by assumption. In a similar way, adding together (1.65) and (1.66) will result in a nonsingular system. Thus, from the singular system (1.63) - (1.66) we get the nonsingular system

$$\begin{gathered}\begin{pmatrix} c_1+c_2 & \mu_1-\mu_2 \\ \nu_1-\nu_2 & c_1+c_2 \end{pmatrix}\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_0,t)= \\ \frac{1}{c_1}\begin{pmatrix} c_1 & \mu_1 \\ \nu_1 & c_1 \end{pmatrix}\int_{a_0}^{a_1} dx'\theta(c_1(t-t_0)-(x'-a_0))\begin{pmatrix}j\\ 0\end{pmatrix}(x',t-\frac{x'-a_0}{c_1}) \\ +\theta(c_1(t-t_0)-(a_1-a_0))\begin{pmatrix} c_1 & \mu_1 \\ \nu_1 & c_1 \end{pmatrix}\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_1,t-\frac{a_1-a_0}{c_1})\end{gathered} \tag{1.72}$$

$$\begin{gathered}\begin{pmatrix} c_2+c_1 & \mu_2-\mu_1 \\ \nu_2-\nu_1 & c_2+c_1 \end{pmatrix}\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_1,t)= \\ \frac{1}{c_1}\begin{pmatrix} c_1 & -\mu_1 \\ -\nu_1 & c_1 \end{pmatrix}\int_{a_0}^{a_1} dx'\theta(c_1(t-t_0)-(a_1-x'))\begin{pmatrix}j\\ 0\end{pmatrix}(x',t-\frac{a_1-x'}{c_1}) \\ -\theta(c_1(t-t_0)-(a_1-a_0))\begin{pmatrix} -c_1 & \mu_1 \\ \nu_1 & -c_1 \end{pmatrix}\begin{pmatrix}\varphi_1\\ \psi_1\end{pmatrix}(a_0,t-\frac{a_1-a_0}{c_1}) \\ +2c_2\begin{pmatrix}\varphi_i\\ \psi_i\end{pmatrix}(a_1,t)\end{gathered} \tag{1.73}$$

The system (1.72), (1.73), which determine the boundary values of the fields in term of the internal and external currents, together with the partial differential equations (1.35), restricted to the interior of the interval (a_0, a_1), defines the EOS formulation for the toy model 2.

1.2.2 Numerical implementation of the Second toy model

In this section we are going to apply Lax-Wendroff numerical method to model (1.35). Let us first describe numerical scheme

$$\begin{aligned}\varphi_k^{i+1} &= \varphi_k^i + \mathrm{d}t\Big(\frac{\mu_1}{2\,\mathrm{d}x}\big(\psi_{k+1}^i-\psi_{k-1}^i\big)+j_k^i\Big) \\ &+ \frac{\mathrm{d}t^2}{2}\Big(\frac{c_1^2}{\mathrm{d}x^2}(\varphi_{k+1}^i-2\varphi_k^i+\varphi_{k-1}^i)+(\alpha-\beta\rho_k^i)\varphi_k^i-\gamma j_k^i\Big) \\ \psi_k^{i+1} &= \psi_k^i + \mathrm{d}t\Big(\frac{\nu_1}{2\,\mathrm{d}x}\big(\varphi_{k+1}^i-\varphi_{k-1}^i\big)\Big) \\ &+ \frac{\mathrm{d}t^2}{2}\Big(\nu_1\big(\frac{\mu_1}{\mathrm{d}x^2}(\psi_{k+1}^i-2\psi_k^i+\psi_{k-1}^i)+\frac{1}{2\,\mathrm{d}x}(j_{k+1}^i-j_{k-1}^i)\big)\Big) \\ \rho_k^{i+1} &= \rho_k^i + \mathrm{d}t\Big(-\frac{1}{2\,\mathrm{d}x}\big(j_{k+1}^i-j_{k-1}^i\big)\Big) \\ &+ \frac{\mathrm{d}t^2}{2}\Big(\frac{\beta}{2\,\mathrm{d}x}(\rho_{k+1}^i-\rho_{k-1}^i)\varphi_k^i-\frac{1}{2\,\mathrm{d}x}(\alpha-\beta\rho_k^i)(\varphi_{k+1}^i-\varphi_{k-1}^i) \\ &+ \gamma\frac{1}{2\,\mathrm{d}x}(j_{k+1}^i-j_{k-1}^i)\Big)\end{aligned} \tag{1.74}$$

and special rules for calculation values of ρ_t on the boundaries

$$\begin{aligned}
\rho_0^{i+1} &= \rho_0^i + \mathrm{d}t\Big(- \frac{1}{2\,\mathrm{d}x}\big(- 3j_0^i + 4j_1^i - j_2^i\big)\Big) \\
&+ \frac{\mathrm{d}t^2}{2}\Big(\frac{\beta}{2\,\mathrm{d}x}(-3\rho_0^i + 4\rho_1^i - \rho_2^i)\varphi_0^i \\
&- \frac{1}{2\,\mathrm{d}x}(\alpha - \beta\rho_0^i)(-3\varphi_0^i + 4\varphi_1^i - \varphi_2^i) \\
&+ \frac{\gamma}{2\,\mathrm{d}x}(-3j_0^i + 4j_1^i - j_2^i)\Big) \\
\rho_{N+1}^{i+1} &= \rho_{N+1}^i + \mathrm{d}t\Big(- \frac{1}{2\,\mathrm{d}x}\big(3j_{N+1}^i - 4j_N^i + j_{N-1}^i\big)\Big) \\
&+ \frac{\mathrm{d}t^2}{2}\Big(\frac{\beta}{2\,\mathrm{d}x}(3\rho_{N+1}^i - 4\rho_N^i + \rho_{N-1}^i)\varphi_{N+1}^i \\
&- \frac{1}{2\,\mathrm{d}x}(\alpha - \beta\rho_{N+1}^i)(3\varphi_{N+1}^i - 4\varphi_N^i + \varphi_{N-1}^i) \\
&+ \frac{\gamma}{2\,\mathrm{d}x}(3j_{N+1}^i - 4j_N^i + j_{N-1}^i)\Big)
\end{aligned} \tag{1.75}$$

Also, we will use Central Finite Difference method to calculate values of $j(x,t)$ in points of grid, so

$$j_k^{i+1} = 2\,\mathrm{d}t\big((\alpha - \beta\rho_k^i)\varphi_k^i - \gamma j_k^i\big) + j_k^{i-1}$$

Let for example take

$$j_s(x,t) = ae^{-b(x-x_0)^2}e^{-c(t-t_0)^2}$$

where $a = 2$, $b = c = 10$, $x_0 = 6$ and $t_0 = 4$. Also, we take $T = 12$, $a_0 = -3$, $a_1 = 3$, $dt = 0.01$, $dx = 0.01$, $\alpha = \beta = \gamma = 0.1$, $\mu_1 = 1.4$, $\mu_2 = 2$, $\nu_1 = 1$ and $\nu_2 = 1$. Numerical solution of the Second toy problem with described parameters is on Figures 1.19 – 1.21 Again we can see artefact on the left boundary of ρ which possibly is reason of oscillations in solutions for φ, ρ and j.

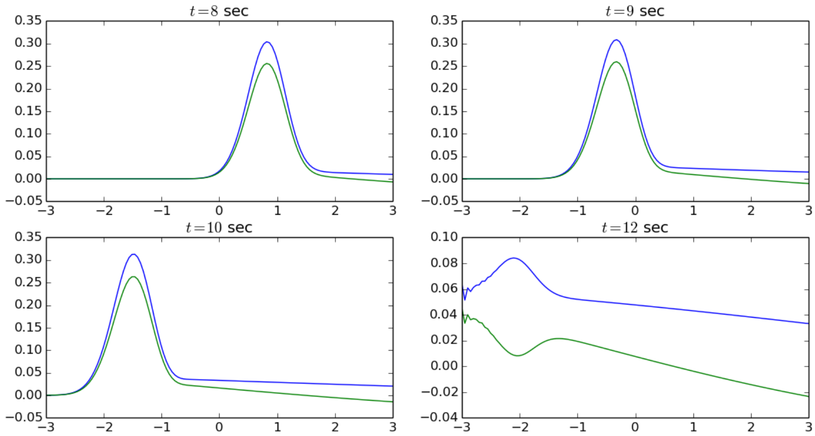

Figure 1.19: Numerical solution of (1.35) of problem (1.35) for φ (blue) and ρ (green) in time points $t = 8$ (upper left), $t = 9$ (upper right), $t = 10$ (lower left), $t = 12$ (lower right). Horizontal axis is x.

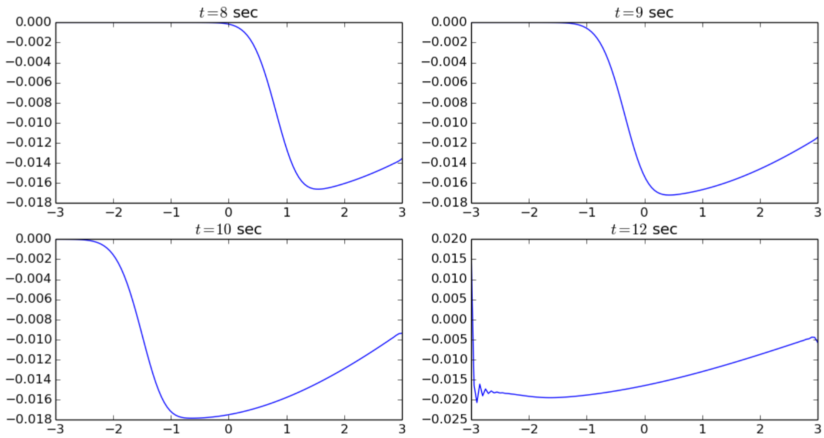

Figure 1.20: Numerical solution of (1.35) for ρ in time points $t = 8$ (upper left), $t = 9$ (upper right), $t = 10$ (lower left), $t = 12$ (lower right). Horizontal axis is x.

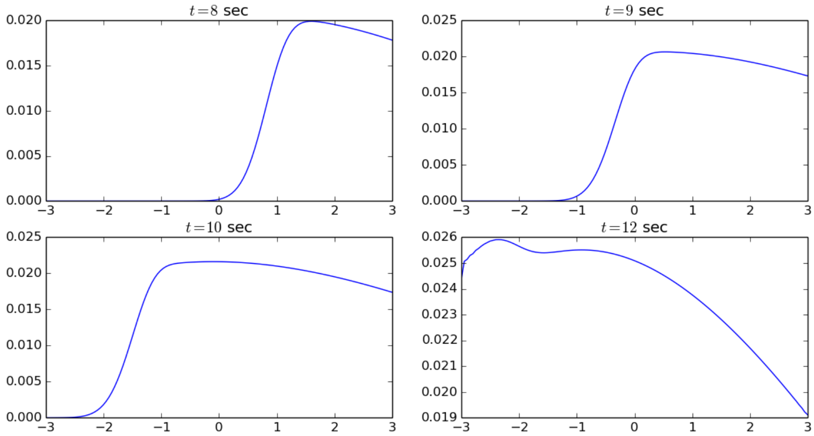

Figure 1.21: Numerical solution of (1.35) for j in time points $t = 8$ (upper left), $t = 9$ (upper right), $t = 10$ (lower left), $t = 12$ (lower right). Horizontal axis is x.

Chapter 2

2D scattering problem

We consider the scattering problem

$$\begin{aligned} \partial_t \varphi &= \mu(\partial_y \psi + i\partial_x \psi) + f(x,y,t) \\ \partial_t \psi &= \nu(\partial_y \varphi - i\partial_x \varphi) + g(x,y,t) \end{aligned} \tag{2.1}$$

where $\mu = \mu(x,y)$, $\nu = \nu(x,y)$. Eventually we want to do a boundary formulation for the geometry shown on Figure 2.1.

Observe that (2.1) imply $f = g = 0$ that

$$\begin{aligned} \partial_{tt}\varphi &= \mu(\partial_t\partial_y\psi + i\partial_t\partial_x\psi) \\ &= \mu(\partial_y\partial_t\psi + i\partial_x\partial_t\psi) \\ &= \mu(\partial_y(\nu(\partial_y\varphi - i\partial_x\varphi)) + i\partial_x(\nu(\partial_y\varphi - i\partial_x\varphi))) \\ &= \mu\nu(\partial_{yy}\varphi - i\partial_{yx}\varphi + i\partial_{xy}\varphi + \partial_{xx}\varphi) \\ &= \mu\nu(\partial_{yy}\varphi + \partial_{xx}\varphi) \end{aligned}$$

This is the 2D wave equation. In a similar way we show that ψ also satisfy the 2D wave

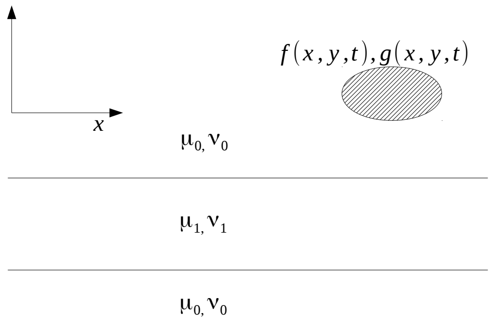

Figure 2.1: Geometry of the given 2D scattering problem.

equation.

$$
\begin{aligned}
\partial_{tt}\psi &= \mu(\partial_t\partial_y\varphi - i\partial_t\partial_x\varphi) \\
&= \nu(\partial_y\partial_t\varphi - i\partial_x\partial_t\varphi) \\
&= \nu(\partial_y(\mu(\partial_y\psi + i\partial_x\psi)) - i\partial_x(\mu(\partial_y\psi + i\partial_x\psi))) \\
&= \mu\nu(\partial_{yy}\psi + i\partial_{yx}\psi - i\partial_{xy}\psi + \partial_{xx}\psi) \\
&= \mu\nu(\partial_{yy}\psi + \partial_{xx}\psi)
\end{aligned}
$$

Our ultimate goal is to investigate the scattering of electromagnetic waves from metals. This will involve the Maxwell equations and these have of course wave solutions.

2.1 The EOS formulation

In order to derive the EOS formulation for the problem (2.1), we will firstly need a space-time integral identity involving the matrix operator

$$
L = \begin{pmatrix} \partial_t & -\mu(i\partial_x + \partial_y) \\ -\nu(i\partial_x - \partial_y) & \partial_t \end{pmatrix} \tag{2.2}
$$

where μ and ν are constants. The operator (2.2) acts on vector valued functions in the usual way and we can write the system (2.1) in the compact form

$$
L\begin{pmatrix}\varphi \\ \psi\end{pmatrix}(x,y,t) = \begin{pmatrix} f \\ g\end{pmatrix}(x,y,t) \tag{2.3}
$$

In order to get a boundary formulation of this equation we need an integral identity.

Let $S = X \times Y = (x_0, x_1) \times (y_0, y_1)$ be a domain in the plane and let $T = (t_0, t_1)$ is time interval. Then we have

$$
\begin{aligned}
\int_{S\times T} dxdydtA\Big(L\begin{pmatrix}\varphi \\ \psi\end{pmatrix}\Big)^* &= \int_{S\times T} dxdydt \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22}\end{pmatrix}\begin{pmatrix}\partial_t\varphi^* - \mu(\partial_y - i\partial_x)\psi^* \\ -\mu(\partial_y + i\partial_x)\varphi^* + \partial_t\psi^*\end{pmatrix} \\
&= \int_{S\times T} dxdydtL^\dagger A\begin{pmatrix}\varphi \\ \psi\end{pmatrix}^* + \int_S dxdyA\cdot\begin{pmatrix}\varphi \\ \psi\end{pmatrix}^*\Bigg|_{t_0}^{t_1} \\
&+ \int_T dt\int_S dxdy\nabla\cdot\Big(B\cdot\begin{pmatrix}\varphi \\ \psi\end{pmatrix}^*\Big)
\end{aligned}
$$

So we have integral identity

$$
\begin{aligned}
&\int_{S\times T} dxdydt\Big[A\Big(L\begin{pmatrix}\varphi \\ \psi\end{pmatrix}\Big)^* - L^\dagger A\begin{pmatrix}\varphi \\ \psi\end{pmatrix}^*\Big] \\
&= \int_S dxdyA\cdot\begin{pmatrix}\varphi \\ \psi\end{pmatrix}^*\Bigg|_{t_0}^{t_1} + \int_T dt\int_S dxdy\nabla\cdot\Big(B\cdot\begin{pmatrix}\varphi \\ \psi\end{pmatrix}^*\Big)
\end{aligned} \tag{2.4}
$$

where φ and ψ are smooth functions on the space-time interval $S \times T$. Also, $A = A(x,t)$ is a 2×2 matrix valued function

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

$L^\dagger$ is the formal adjoint to the operator L, and acts on the matrix valued function A in the following way

$$L^\dagger A = \begin{pmatrix} -\partial_t A_{11} + \nu(i\partial_x + \partial_y)A_{12} & \mu(-i\partial_x + \partial_y)A_{11} - \partial_t A_{12} \\ -\partial_t A_{21} + \nu(i\partial_x + \partial_y)A_{22} & \mu(-i\partial_x + \partial_y)A_{21} - \partial_t A_{22} \end{pmatrix} \tag{2.5}$$

and $B = (B_{ijk})$ is a 3-tensor

$$\begin{aligned} B_{111} &= -i\nu A_{12}, \quad B_{121} = -i\nu A_{22}, \\ B_{211} &= -\nu A_{12}, \quad B_{221} = -\nu A_{22}, \\ B_{112} &= i\mu A_{11}, \quad B_{122} = i\mu A_{21}, \\ B_{212} &= -\mu A_{11}, \quad B_{222} = -\mu A_{21}, \end{aligned}$$

The second item we need to derive the EOS formulation for the problem (2.1) is the advanced Green's function for the operator $L^\dagger$:

$$L^\dagger G(x,y,t,x',y',t') = \delta(t)\delta(x)\delta(y)I \tag{2.6}$$

Using (2.5) and (2.6) we have the following system of four equations for the components of G.

$$\begin{aligned} &\partial_t G_{11} - \nu(i\partial_x + \partial_y)G_{12} = -\delta(t)\delta(x)\delta(y) \\ &\partial_t G_{12} - \mu(i\partial_x + \partial_y)G_{11} = 0 \\ &\partial_t G_{21} - \nu(i\partial_x + \partial_y)G_{22} = 0 \\ &\partial_t G_{22} - \mu(i\partial_x + \partial_y)G_{21} = -\delta(t)\delta(x)\delta(y) \end{aligned} \tag{2.7}$$

The two first and the two last equations can be solve separately. We use the Fourier transform in x, y and t to solve the initial value problem for 2.7.

$$\begin{aligned} \hat{\varphi}(k_x,k_y,\omega) &= \int_{-\infty}^{\infty} \mathrm{d}x\,\mathrm{d}y\,\mathrm{d}t\varphi(x,y,t)e^{-i(k_x x + k_y y - \omega t)} \\ \varphi(k_x,k_y,\omega) &= \frac{1}{8\pi^3}\int_{-\infty}^{\infty} \mathrm{d}k_x\,\mathrm{d}k_y\,\mathrm{d}\omega\hat{\varphi}(k_x,k_y,\omega)e^{i(k_x x + k_y y - \omega t)} \end{aligned}$$

Using these two transforms on the two first equations we get

$$i\omega\hat{G}_{11} - \nu(k_x - ik_y)\hat{G}_{12} = 1$$
$$i\omega\hat{G}_{12} + \mu(k_x + ik_y)\hat{G}_{11} = 0$$
$$\Updownarrow$$
$$\underbrace{\begin{bmatrix} i\omega & -\nu(k_x - ik_y) \\ \mu(k_x + ik_y) & i\omega \end{bmatrix}}_{M} \begin{bmatrix} \hat{G}_{11} \\ \hat{G}_{12} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\det M = -(\omega^2 - \mu\nu(k_x^2 + k_y^2)) = -(\omega^2 - c^2k^2)$$

where $k = (k_x^2 + k_y^2)^{\frac{1}{2}}$, $c^2 = \mu\nu$.

$$\begin{bmatrix} \hat{G}_{11} \\ \hat{G}_{12} \end{bmatrix} = \frac{-1}{\omega^2 - c^2k^2} \begin{bmatrix} i\omega & \nu(k_x - ik_y) \\ -\mu(k_x + ik_y) & i\omega \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Then we can find that

$$\hat{G}_{11}(k_x, k_y, \omega) = \frac{-i\omega}{\omega^2 - c^2k^2}$$
$$\hat{G}_{12}(k_x, k_y, \omega) = \frac{\mu(k_x + ik_y)}{\omega^2 - c^2k^2}$$

Also, we can find $G_{11}(x, y, t)$ and $G_{12}(x, y, t)$ like

$$G_{11}(x, y, t) = \frac{1}{8\pi^3}\int_{-\infty}^{\infty} \mathrm{d}k_x\, \mathrm{d}k_y g_{11}(k_x, k_y, t)e^{i(k_x x + k_y y)}$$
$$G_{12}(x, y, t) = \frac{1}{8\pi^3}\int_{-\infty}^{\infty} \mathrm{d}k_x\, \mathrm{d}k_y g_{12}(k_x, k_y, t)e^{i(k_x x + k_y y)}$$

where

$$g_{11}(k_x, k_y, t) = \int_{-\infty}^{\infty} \mathrm{d}\omega \frac{-i\omega}{\omega^2 - c^2k^2} e^{-i\omega t}$$
$$g_{12}(k_x, k_y, t) = \int_{-\infty}^{\infty} \mathrm{d}\omega \frac{\mu(k_x + ik_y)}{\omega^2 - c^2k^2} e^{-i\omega t}$$

We want to find the advanced Green's function and use the contour shown on Figure 2.2 Thus

$$g_{11}(k_x, k_y, t) = \int_{-\infty}^{\infty} \mathrm{d}\omega \frac{-i\omega}{\omega^2 - c^2k^2} e^{-i\omega t}$$
$$g_{12}(k_x, k_y, t) = \int_{-\infty}^{\infty} \mathrm{d}\omega \frac{\mu(k_x + ik_y)}{\omega^2 - c^2k^2} e^{-i\omega t}$$

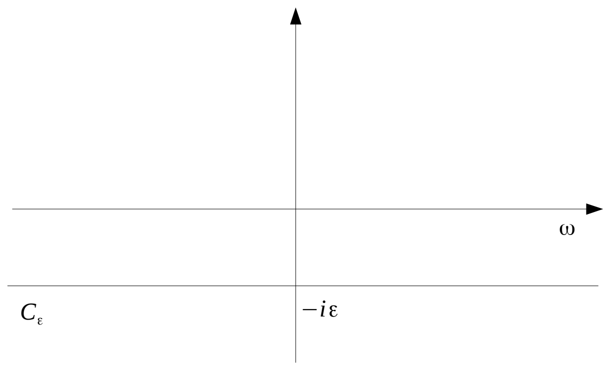

Figure 2.2: Contour which is used to find the advanced Green's function for problem (2.1)

The denominator in g_{11} and g_{12} has zeros on the real axis at $\pm kc$. Therefore, by Cauchy

$$g_{11} = g_{12} = 0 \quad t > 0$$

For $t \leq 0$ we close the contour in the upper halfplane and how the two poles, $z = \pm kc$, of the integrands are inside the contour. By Cauchy we get

$$\begin{aligned} g_{11}(k_x, k_y, t) &= 2\pi i \left(\frac{-i(-kc)}{2(-kc)} e^{-i(-kc)t} + \frac{-i(kc)}{2(kc)} e^{-i(kc)t} \right) \\ &= \pi \left(e^{ikct} + e^{-ikct} \right) \end{aligned}$$

$$\begin{aligned} g_{12}(k_x, k_y, t) &= 2\pi i \left(\frac{\mu(k_x + ik_y)}{2(-kc)} e^{-i(-kc)t} + \frac{\mu(k_x + ik_y)}{2(kc)} e^{-i(kc)t} \right) \\ &= -\frac{\pi\mu(ik_x - k_y)}{ck} \left(e^{ikct} - e^{-ikct} \right) \end{aligned}$$

$$\begin{aligned} \Rightarrow g_{11}(k_x, k_y, t) &= \pi\theta(-t) \left(e^{ikct} + e^{-ikct} \right) \\ &= 2\pi\theta(-t)\cos(kct) \\ g_{12}(k_x, k_y, t) &= -\frac{\pi\mu(ik_x - k_y)}{ck} \theta(-t) \left(e^{ikct} - e^{-ikct} \right) \\ &= \frac{2\pi\mu(k_x + ik_y)}{ck} \theta(-t)\sin(kct) \end{aligned}$$

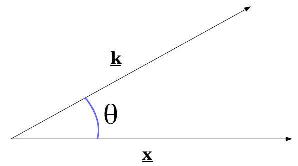

Figure 2.3: Introducing polar coordinates to find $G_{11}(\underline{\mathbf{x}}, t)$.

$$\Rightarrow G_{11}(x,y,t) = \frac{1}{8\pi^3}\int_{-\infty}^{\infty} \mathrm{d}k_x\,\mathrm{d}k_y 2\pi\theta(-t)\cos(kct)e^{i(k_x x+k_y y)}$$
$$= \frac{\theta(-t)}{4\pi^2}\int_{\mathbb{R}^2} \mathrm{d}\underline{\mathbf{k}}\cos kct e^{i\underline{\mathbf{k}}\cdot\underline{\mathbf{x}}}$$

We introduce polar coordinates with axis centered on $\underline{\mathbf{x}}$ like on Figure 2.3 where $r = ||\underline{\mathbf{x}}||$, $k = ||\underline{\mathbf{k}}||$

$$\Rightarrow G_{11}(\underline{\mathbf{x}}, t) = \frac{\theta(-t)}{4\pi^2}\int_0^{2\pi}\mathrm{d}\theta\int_0^{\infty}\mathrm{d}k\cos(kct)e^{ikr\cos\theta}$$
$$= \frac{\theta(-t)}{4\pi^2 c}\partial_t\int_0^{2\pi}\mathrm{d}\theta\int_0^{\infty}\mathrm{d}k\sin(kct)e^{ikr\cos\theta}$$
$$= \frac{\theta(-t)}{4\pi^2 c}\partial_t\int_0^{2\pi}\mathrm{d}\theta\int_0^{\infty}\mathrm{d}k\frac{1}{2i}\left(e^{ik(r\cos\theta+ct)} - e^{ik(r\cos\theta-ct)}\right)$$
$$= \frac{\theta(-t)}{8\pi^2 ic}\partial_t\int_0^{2\pi}\mathrm{d}\theta\left(I(r\cos\theta+ct) - I(r\cos\theta-ct)\right)$$

where

$$I(y) = \int_0^{\infty}\mathrm{d}k e^{iky}$$

This is a divergent integral. We introduce the regularized integral

$$I_\varepsilon(y) = \int_0^{\infty}\mathrm{d}k e^{ik(y+i\varepsilon)}, \quad \varepsilon > 0$$

and observe that

$$I(y) = \lim_{\varepsilon\to 0} I_\varepsilon(y)$$

Evidently

$$I_\varepsilon(y) = \frac{1}{i(y+i\varepsilon)}e^{ik(y+i\varepsilon)}\Big|_0^{\infty} = \frac{i}{y+i\varepsilon}$$

Defining

$$J_\varepsilon^\pm = \int_0^{2\pi} \mathrm{d}\theta I_\varepsilon(r\cos\theta \pm ct)$$

we have then

$$J_\varepsilon^\pm = \int_0^{2\pi} \mathrm{d}\theta \frac{i}{r\cos\theta \pm ct + i\varepsilon}$$

Let $z = e^{i\theta}$, $\mathrm{d}\theta = \dfrac{-i\,\mathrm{d}z}{z}$, $\cos\theta = \dfrac{1}{2}(z + z^{-1})$

$$\begin{aligned}\Rightarrow J_\varepsilon^\pm &= \oint_{S_1} \frac{\mathrm{d}z}{z(\frac{r}{2}(z+z^{-1}) \pm ct + i\varepsilon)} \\ &= \oint_{S_1} \frac{2\,\mathrm{d}z}{rz^2 + 2(\pm ct + i\varepsilon)z + r}\end{aligned}$$

where S_1 is the unit circle. This integral can be solved using Cauchy. Observe that the poles of the integrand in the complex plane occur when

$$p(z) = rz^2 + 2(\pm ct + i\varepsilon)z + r$$

Observe that

$$p(z) = 0 \Leftrightarrow z^2 + \alpha z + 1 = 0, \quad \alpha = \frac{2(\pm ct + i\varepsilon)}{r}$$

Let z_1, z_2 be the two roots of p. Then from the theory of polynomials we have

$$z_1 z_2 = 1$$

$$\begin{aligned}\text{Let } z_1 &= r_1 e^{i\theta_1}, \quad z_2 = r_2 e^{i\theta_2} \\ &\Rightarrow r_1 r_2 e^{i(\theta_1+\theta_2)} = 1 \\ &\Rightarrow r_1 = \frac{1}{r_2}, \quad \theta_1 = -\theta_2\end{aligned}$$

Therefore either both poles of the integrand is on S_1 or exactly one pole is inside S_1.

In order to write down formulas for the poles we must choose a branch of the complex square root. We choose to put the branch cut along the negative real axis.

Let $f(z)$ be the unique square root defined by

$$f(z) = \sqrt{|z|}e^{i\frac{\theta}{2}}, \quad \text{where } -\pi \leqslant \theta < \pi$$

The other square root corresponding to this branch cut is then $-f(z)$.

Let us first consider $J_\varepsilon^\pm$. The two poles are

$$z_1^+(\varepsilon) = \frac{1}{r}(c|t| - i\varepsilon + f(c^2t^2 - r^2 - \varepsilon^2 - 2ic|t|\varepsilon))$$
$$z_2^+(\varepsilon) = \frac{1}{r}(c|t| - i\varepsilon - f(c^2t^2 - r^2 - \varepsilon^2 - 2ic|t|\varepsilon))$$

Mathematical analysis show that z_2^+ is inside the unit circle. Let $z^+(\varepsilon) = z_2^+(\varepsilon)$.
For J_ε^- the two poles are

$$z_1^-(\varepsilon) = \frac{1}{r}(-c|t| - i\varepsilon + f(c^2t^2 - r^2 - \varepsilon^2 + 2ic|t|\varepsilon))$$
$$z_2^-(\varepsilon) = \frac{1}{r}(-c|t| - i\varepsilon - f(c^2t^2 - r^2 - \varepsilon^2 + 2ic|t|\varepsilon))$$

We now find that $z_1^-(\varepsilon)$ is inside the unit circle. Let $z^-(\varepsilon) = z_1^-(\varepsilon)$.
In summary for $J_\varepsilon^\pm$ we have exactly one pole inside S_1 given by

$$z^+(\varepsilon) = \frac{1}{r}(c|t| - i\varepsilon - f(c^2t^2 - r^2 - \varepsilon^2 - 2ic|t|\varepsilon))$$
$$z^-(\varepsilon) = \frac{1}{r}(-c|t| - i\varepsilon + f(c^2t^2 - r^2 - \varepsilon^2 + 2ic|t|\varepsilon))$$

Observe that

$$2rz^\pm(\varepsilon) + 2(\mp ct + i\varepsilon) = \mp 2f(c^2t^2 - r^2 - \varepsilon^2 \mp 2ic|t|\varepsilon)$$

Cauchy now give

$$J_\varepsilon^\pm = 2\pi i\left(\frac{2}{\mp 2f(c^2t^2 - r^2 - \varepsilon^2 \mp 2ic|t|\varepsilon)}\right)$$
$$= \frac{\mp 2\pi i}{f(c^2t^2 - r^2 - \varepsilon^2 \mp 2ic|t|\varepsilon)}$$

Because of the presence of the branch cut on the negative real axis the limits of $J_\varepsilon^\pm$ will depend on the sign of $c^2t^2 - r^2$.

i) $c^2t^2 - r^2 > 0$

Then $c^2t^2 - r^2 - \varepsilon^2 \mp 2ic|t|\varepsilon$ are on the right halfplane and thus away from the branch cut.

$$J_\varepsilon^\pm \longrightarrow \frac{\mp 2\pi i}{\sqrt{c^2t^2 - r^2}}$$
$$J_\varepsilon^+ - J_\varepsilon^- \longrightarrow \frac{-4\pi i}{\sqrt{c^2t^2 - r^2}}$$

ii) $c^2t^2 - r^2 < 0$

Then $c^2t^2 - r^2 - \varepsilon^2 \mp 2ic|t|\varepsilon$ are on each side of the branch cut along the negative real axis.

$$J_\varepsilon^\pm \longrightarrow \frac{2\pi}{\sqrt{|c^2t^2 - r^2|}}$$
$$J_\varepsilon^+ - J_\varepsilon^- \longrightarrow 0$$

These two results can be collected into

$$J_\varepsilon^+ - J_\varepsilon^- \longrightarrow -4\pi i \frac{\theta(c^2t^2 - r^2)}{\sqrt{c^2t^2 - r^2}}$$

This gives finally the following expression for G_{11}

$$G_{11}(\underline{\mathbf{x}}, t) = \frac{\theta(-t)}{2\pi c}\partial_t \frac{\theta(c^2t^2 - r^2)}{\sqrt{c^2t^2 - r^2}}$$

Let us next find G_{12}

$$\begin{aligned}
G_{12}(x, y, t) &= -\frac{1}{8\pi^3}\int_{-\infty}^{\infty} \mathrm{d}k_x\, \mathrm{d}k_y \frac{2\pi i\mu(-k_y + ik_x)\theta(-t)}{ck}\sin(kct)e^{i(k_x x + k_y y)} \\
&= -\frac{\theta(-t)i\mu}{4\pi^2 c}\int_{\mathbb{R}^2} \mathrm{d}\underline{\mathbf{k}}\sin(kct)\frac{-k_y + ik_k}{k}e^{i\underline{\mathbf{k}}\cdot\underline{\mathbf{x}}} \\
&= -\frac{\theta(-t)i\mu}{4\pi^2 c}\int_0^{2\pi} \mathrm{d}\theta \int_0^\infty \mathrm{d}kk\sin(kct)(-\sin\theta + i\cos\theta)e^{ikr\cos\theta} \\
&= \frac{\theta(-t)i\mu}{4\pi^2 c^2}\partial_t \int_0^{2\pi} \mathrm{d}\theta \int_0^\infty \mathrm{d}k\cos(kct)(-\sin\theta + i\cos\theta)e^{ikr\cos\theta} \\
&= \frac{\theta(-t)\mu}{8\pi^2 c^2}\partial_t \int_0^{2\pi} \mathrm{d}\theta \left(\int_0^\infty \mathrm{d}ke^{ik(r\cos\theta + ct)} + \int_0^\infty \mathrm{d}ke^{ik(r\cos\theta - ct)} \right) \\
&\quad (-\sin\theta + i\cos\theta) \\
&= \frac{\theta(-t)\mu}{8\pi^2 c^2}\partial_t \int_0^{2\pi} \mathrm{d}\theta(-\sin\theta + i\cos\theta)\Big(I(r\cos\theta + ct) - I(r\cos\theta - ct) \Big)
\end{aligned}$$

where $I(y) = \int_0^\infty \mathrm{d}ke^{iky}$.

The same regularizations as before for G_{11} leads to integrals

$$J_\varepsilon^\pm = -\int_0^{2\pi} \mathrm{d}\theta \frac{\cos\theta + i\sin\theta}{r\cos\theta \pm ct + i\varepsilon}$$

and $z = e^{i\theta}$, $\mathrm{d}\theta = \dfrac{-i\,\mathrm{d}z}{z}$, $\cos\theta = \dfrac{1}{2}(z + z^{-1})$ and $\sin\theta = \dfrac{1}{2i}(z - z^{-1})$

$$\begin{aligned} J_\varepsilon^\pm &= \oint_{S_1} \frac{i\,\mathrm{d}z}{z} \frac{\frac{1}{2}(z+z^{-1}) + \frac{1}{2i}(z-z^{-1})}{\frac{1}{2}r(z+z^{-1}) \pm ct + i\varepsilon} \\ &= 2i \oint_{S_1} \mathrm{d}z \frac{z}{rz^2 + 2(\pm ct + i\varepsilon)z + r} \end{aligned}$$

The poles inside S_1 are the same as when we calculated G_{11}. Cauchy gives

$$\begin{aligned} J_\varepsilon^\pm &= 2\pi i \left(\frac{2iz^\pm(\varepsilon)}{2rz^\pm(\varepsilon + 2(\pm ct + i\varepsilon))} \right) \\ &= -4\pi \frac{\frac{1}{r}\Big(\pm c|t| - i\varepsilon \mp f(c^2t^2 - r^2 - \varepsilon^2 \mp 2ic|t|\varepsilon) \Big)}{\mp 2f(c^2t^2 - r^2 - \varepsilon^2 \mp 2ic|t|\varepsilon)} \end{aligned}$$

i) $c^2t^2 - r^2 > 0$

$$\begin{aligned} J_\varepsilon^\pm &\longrightarrow \frac{2\pi}{r} \frac{ct - \sqrt{c^2t^2 - r^2}}{\mp\sqrt{c^2t^2 - r^2}} \\ J_\varepsilon^+ + J_\varepsilon^- &\longrightarrow \frac{4\pi}{r} \frac{ct - \sqrt{c^2t^2 - r^2}}{\sqrt{c^2t^2 - r^2}} \end{aligned}$$

ii) $c^2t^2 - r^2 < 0$

$$\begin{aligned} J_\varepsilon^\pm &\longrightarrow -\frac{2\pi}{r} \frac{cti + \sqrt{|c^2t^2 - r^2|}}{\sqrt{|c^2t^2 - r^2|}} \\ J_\varepsilon^+ + J_\varepsilon^- &\longrightarrow -\frac{4\pi}{r} \frac{cti + \sqrt{|c^2t^2 - r^2|}}{\sqrt{|c^2t^2 - r^2|}} \end{aligned}$$

$$\begin{aligned} G_{12}(\underline{\mathbf{x}}, t) &= -\frac{\theta(-t)\mu}{8\pi^2c^2} \partial_t \begin{cases} \dfrac{4\pi}{r} \dfrac{ct - \sqrt{c^2t^2 - r^2}}{\sqrt{c^2t^2 - r^2}}, & c^2t^2 - r^2 > 0 \\ -\dfrac{4\pi}{r} \dfrac{cti + \sqrt{|c^2t^2 - r^2|}}{\sqrt{|c^2t^2 - r^2|}}, & c^2t^2 - r^2 < 0 \end{cases} \\ &= -\frac{\theta(-t)\mu}{2\pi c^2} \partial_t \frac{ct - \sqrt{c^2t^2 - r^2}}{r\sqrt{c^2t^2 - r^2}} \theta(c^2t^2 - r^2) \end{aligned}$$

Let us next turn to G_{21} and G_{22}. They solve the system

$$\begin{aligned} -\partial_t G_{22} + \mu(\partial_y - i\partial_x)G_{21} &= \delta(x-x')\delta(y-y')\delta(t-t') \\ -\partial_t G_{21} + \nu(\partial_y + i\partial_x)G_{22} &= 0 \end{aligned}$$

This system turns into the system for G_{11} and G_{12} if we let $x \to -x$, $\mu \leftrightarrow \nu$. Therefore,

$$\begin{aligned} G_{21}(\underline{\mathbf{x}},t) &= -\frac{\theta(-t)\nu}{2\pi c^2}\partial_t \frac{ct - \sqrt{c^2t^2-r^2}}{r\sqrt{c^2t^2-r^2}}\theta(c^2t^2-r^2) \\ G_{22}(\underline{\mathbf{x}},t) &= \frac{\theta(-t)}{2\pi c}\partial_t \frac{\theta(c^2t^2-r^2)}{\sqrt{c^2t^2-r^2}} \end{aligned}$$

therefore we finally have

$$\begin{aligned} G(\underline{\mathbf{x}},t) &= \begin{bmatrix} G_{11}(\underline{\mathbf{x}},t) & G_{12}(\underline{\mathbf{x}},t) \\ G_{21}(\underline{\mathbf{x}},t) & G_{22}(\underline{\mathbf{x}},t) \end{bmatrix} \\ &= \theta(-t)\partial_t\theta(c^2t^2-r^2)\begin{bmatrix} p(\underline{\mathbf{x}},t) & \mu q(\underline{\mathbf{x}},t) \\ \nu q(\underline{\mathbf{x}},t) & p(\underline{\mathbf{x}},t) \end{bmatrix} \end{aligned}$$

where

$$\begin{aligned} p(\underline{\mathbf{x}},t) &= \frac{1}{2\pi c}(c^2t^2 - ||\underline{\mathbf{x}}||^2)^{-1/2} \\ q(\underline{\mathbf{x}},t) &= -\frac{1}{2\pi c^2}\frac{ct - \sqrt{c^2t^2-||\underline{\mathbf{x}}||^2}}{||\underline{\mathbf{x}}||\sqrt{c^2t^2-||\underline{\mathbf{x}}||^2}} \end{aligned}$$

Let us now consider the following geometry shown on Figure 2.4 S is characterized at the linear level by μ, ν and S_R by μ_0, ν_0. The source is located inside $S_R \times T$. Eventually we will let $R \to \infty$.

Our general integral identity is

$$\begin{aligned} &\int_{D\times T} \mathrm{d}x'\,\mathrm{d}y'\,\mathrm{d}t' \left(A\left(L\begin{pmatrix}\varphi\\ \psi\end{pmatrix}\right)^* - L^\dagger A\begin{pmatrix}\varphi\\ \psi\end{pmatrix}^* \right) \\ &= \int_D \mathrm{d}x'\,\mathrm{d}y' A\begin{pmatrix}\varphi\\ \psi\end{pmatrix}^* \Bigg|_{t_0}^{t_1} + \int_T \mathrm{d}t' \int_{\partial D} \mathrm{d}l' \underline{n}\cdot B\cdot \begin{pmatrix}\varphi\\ \psi\end{pmatrix}^* \end{aligned}$$

where the 3-tensor B was described before.

Let us choose $A = G$ and apply the integral identity inside $S \times T$.

Our equations are

$$\begin{aligned} L\begin{pmatrix}\varphi\\ \psi\end{pmatrix}(x',y',t') &= 0 \\ L^\dagger G(x',y',t',x,y,t) &= \delta(x'-x)\delta(y'-y)\delta(t'-t)I \end{aligned}$$

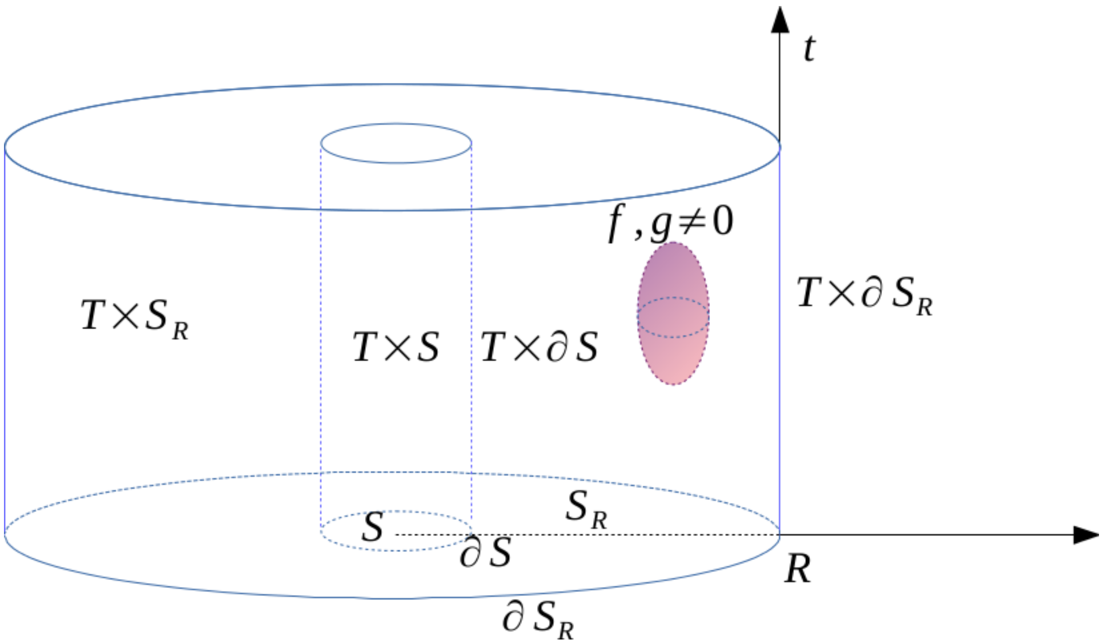

Figure 2.4: Graphical representation of the given scattering problem (2.1). S is characterized at the linear level by μ, ν and S_R by μ_0, ν_0. The source is located inside $S_R \times T$.

$$\begin{aligned}\begin{pmatrix}\varphi\\ \psi\end{pmatrix}(x,y,t) = \int_S \mathrm{d}x'\,\mathrm{d}y' \Bigg(& G^*(x',y',t_1,x,y,t)\begin{pmatrix}\varphi\\ \psi\end{pmatrix}(x',y',t_1) \\ & - G^*(x',y',t_0,x,y,t)\begin{pmatrix}\varphi\\ \psi\end{pmatrix}(x',y',t_0)\Bigg) \\ & - \int_T \mathrm{d}t' \int_{\partial S} \mathrm{d}l_{\underline{\mathbf{x}}'}\underline{n}(x',y')\cdot B^*(x',y',t',x,y,t)\cdot\begin{pmatrix}\varphi\\ \psi\end{pmatrix}(x',y',t')\end{aligned}$$

Assuming as usual that all fields are generated by the source and the fact that we are using an advanced Green's function the expressions.

$$\begin{aligned}\begin{pmatrix}\varphi\\ \psi\end{pmatrix}(x,y,t) = & - \int_T \mathrm{d}t' \int_{\partial S} \mathrm{d}l_{\underline{\mathbf{x}}'}\underline{n}(x',y')\cdot B^*_-(x',y',t',x,y,t)\cdot\begin{pmatrix}\varphi\\ \psi\end{pmatrix}_-(x',y',t'), \\ & (x,y,t)\in S\times T\end{aligned}$$

where a "-" subscript means that the limit of the quantity is taken from inside the boundary $\partial S \times T$. Similarly, for a "+" meaning the limit is from outside a boundary

Applying the integral identity to $S_R \times T$ we get

$$\begin{aligned}
\begin{pmatrix}\varphi\\ \psi\end{pmatrix}(x,y,t) = &\int_{D\times T} \mathrm{d}x'\,\mathrm{d}y'\,\mathrm{d}t' G_0(x',y',t',x,y,t)\begin{pmatrix}f\\ g\end{pmatrix}(x',y',t')\\
&+\int_T \mathrm{d}t' \int_{\partial S} \mathrm{d}l_{\underline{\mathbf{x}}'} \underline{n}\cdot B^*_{0+}(x',y',t',x,y,t)\\
&\cdot \begin{pmatrix}\varphi\\ \psi\end{pmatrix}_+ (x',y',t')\\
&-\int_T \mathrm{d}t' \int_{\partial S_R} \mathrm{d}l_{\underline{\mathbf{x}}'} \underline{n}\cdot B^*_{0-}(x',y',t',x,y,t)\\
&\cdot \begin{pmatrix}\varphi\\ \psi\end{pmatrix}_- (x',y',t')
\end{aligned}$$

We let $R \to \infty$ for finite T. Then the last term vanish because B is generating from matrix elements $G_{ij}(x',y',t',x,y,t)$ and these goes to the zero as $||x-x'|| \to \infty$

$$\begin{aligned}
\begin{pmatrix}\varphi\\ \psi\end{pmatrix}(x,y,t) = &\int_{R^2\times T} \mathrm{d}x'\,\mathrm{d}y'\,\mathrm{d}t' G_0(x',y',t',x,y,t)\begin{pmatrix}f\\ g\end{pmatrix}(x',y',t')\\
&+\int_T \mathrm{d}t' \int_{\partial S} \mathrm{d}l_{\underline{\mathbf{x}}'} \underline{n}\cdot B^*_{0+}(x',y',t',x,y,t)\\
&\cdot \begin{pmatrix}\varphi\\ \psi\end{pmatrix}_+ (x',y',t')\\
&= \begin{pmatrix}\varphi\\ \psi\end{pmatrix}^{(i)} (x,y,t)\\
&+\int_T \mathrm{d}t' \int_{\partial S} \mathrm{d}l_{\underline{\mathbf{x}}'} \underline{n}\cdot B^*_{0+}(x',y',t',x,y,t)\\
&\cdot \begin{pmatrix}\varphi\\ \psi\end{pmatrix}_+ (x',y',t')
\end{aligned}$$

A index "0" on B refer to the fact that G and as a consequence B depends on μ, ν, whose values in $S_R \times T$ are μ_0, ν_0.

From the definition of G we have

$$\begin{aligned}
G(x',y',t',x,y,t) &= \theta(t-t')\partial_U g(x'-x,y'-y,U)\Big|_{U=t'-t}\\
&= -\theta(t-t')\partial_t g(x'-x,y'-y,t'-t)
\end{aligned}$$

and since B is constructed from the matrix elements of G we have

$$B(x',y',t',x,y,t) = \theta(t-t')\partial_t b(x',y',t',x,y,t)$$

for some 3-tensor b.

$$\int_T \mathrm{d}t' \int_{\partial S} \mathrm{d}l_{\underline{\mathbf{x}}'} \underline{n}(x',y') \cdot B^*_-(x',y',t',x,y,t)$$
$$\cdot \begin{pmatrix}\varphi\\ \psi\end{pmatrix}_- (x',y',t')$$
$$= \int_{\partial S} \mathrm{d}l_{\underline{\mathbf{x}}'} \underline{n}(x',y') \cdot \int_{t_0}^{t_1} \mathrm{d}t' \theta(t-t')\partial_t b(x',y',t',x,y,t)$$
$$\cdot \begin{pmatrix}\varphi\\ \psi\end{pmatrix}_- (x',y',t')$$
$$= \int_{\partial S} \mathrm{d}l_{\underline{\mathbf{x}}'} \underline{n}(x',y')$$
$$\cdot \int_{t_0}^{t_1} \mathrm{d}t' \Big(\partial_t\big(\theta(t-t')b(x',y',t',x,y,t)\big) - \partial_t\theta(t-t')b(x',y',t',x,y,t)\Big)$$
$$\cdot \begin{pmatrix}\varphi\\ \psi\end{pmatrix}_- (x',y',t')$$
$$= \partial_t \int_{t_0}^{t} \mathrm{d}t' \int_{\partial S} \mathrm{d}l_{\underline{\mathbf{x}}'} \underline{n}(x',y') \cdot b(x',y',t',x,y,t) \cdot \begin{pmatrix}\varphi\\ \psi\end{pmatrix}_- (x',y',t')$$
$$- \int_{\partial S} \mathrm{d}l_{\underline{\mathbf{x}}'} \underline{n}(x',y') \cdot \int_{t_0}^{t_1} \mathrm{d}t' \delta(t-t') b(x',y',t',x,y,t) \cdot \begin{pmatrix}\varphi\\ \psi\end{pmatrix}_- (x',y',t')$$

But b contains $\theta(c^2(t-t')^2 - ||x-x'||^2) = 0$ when $t = t'$ then last term is zero.

$$\to \partial_t \int_{t}^{t_1} \mathrm{d}t' \int_{\partial S} \mathrm{d}l_{\underline{\mathbf{x}}'} \underline{n}(x',y') \cdot b(x',y',t',x,y,t) \cdot \begin{pmatrix}\varphi\\ \psi\end{pmatrix}_- (x',y',t')$$

Therefore, the fundamental integral identities can be written as

$$\begin{aligned}
\begin{pmatrix}\varphi\\ \psi\end{pmatrix}(\underline{\mathbf{x}},t) &= -\partial_t \int_{t_0}^{t} \mathrm{d}t' \int_{\partial S} \mathrm{d}l_{\underline{\mathbf{x}}'} \underline{n}(\underline{\mathbf{x}}') \cdot b(\underline{\mathbf{x}}',t',\underline{\mathbf{x}},t) \\
&\quad \cdot \begin{pmatrix}\varphi\\ \psi\end{pmatrix}_- (\underline{\mathbf{x}}',t'), \quad (\underline{\mathbf{x}}',t') \in S\times T \\
\begin{pmatrix}\varphi\\ \psi\end{pmatrix}(\underline{\mathbf{x}},t) &= \begin{pmatrix}\varphi\\ \psi\end{pmatrix}^{(i)}(\underline{\mathbf{x}},t) + \partial_t \int_{t_0}^{t_1} \mathrm{d}t' \int_{\partial S} \mathrm{d}l_{\underline{\mathbf{x}}'} \underline{n}(\underline{\mathbf{x}}') \cdot b_0(\underline{\mathbf{x}}',t',\underline{\mathbf{x}},t) \\
&\quad \cdot \begin{pmatrix}\varphi\\ \psi\end{pmatrix}_- (\underline{\mathbf{x}}',t'), \quad (\underline{\mathbf{x}}',t') \in (R^2\times S)\times T
\end{aligned} \tag{2.8}$$

We must now let $\underline{\mathbf{x}}$ approach ∂S from inside and outside S.

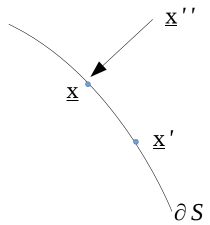

Figure 2.5: Denote a generic point in S by $\underline{\mathbf{x}}''$.

Denote a generic point in S by $\underline{\mathbf{x}}''$
Thus

$$\begin{pmatrix}\varphi\\ \psi\end{pmatrix}(\underline{\mathbf{x}}'', t) = -\partial_t \int_{t_0}^{t} \mathrm{d}t' \int_{\partial S} \mathrm{d}l_{\underline{\mathbf{x}}'} \underline{n}(\underline{\mathbf{x}}') \cdot b(\underline{\mathbf{x}}', t', \underline{\mathbf{x}}', t) \cdot \begin{pmatrix}\varphi\\ \psi\end{pmatrix}_{-}(\underline{\mathbf{x}}', t')$$

Let now $\underline{\mathbf{x}}$ be a point on ∂S (Figure 2.5).

The integrand will be a singular function of $\underline{\mathbf{x}}'$ as $\underline{\mathbf{x}} \to \underline{\mathbf{x}}''$ but this singularity is weak and we will see that all expressions we encounter will be well defined as improper integrals.

From how this limiting procedure is implied whenever we write integrals. The boundary integral equations are thus

$$\begin{pmatrix}\varphi\\ \psi\end{pmatrix}(\underline{\mathbf{x}}, t) = -\partial_t \int_{t_0}^{t} \mathrm{d}t' \int_{\partial S} \mathrm{d}l_{\underline{\mathbf{x}}'} \underline{n}(\underline{\mathbf{x}}') \cdot b(\underline{\mathbf{x}}', t', \underline{\mathbf{x}}, t) \cdot \begin{pmatrix}\varphi\\ \psi\end{pmatrix}(\underline{\mathbf{x}}', t')$$

$$\begin{pmatrix}\varphi\\ \psi\end{pmatrix}(\underline{\mathbf{x}}, t) = \begin{pmatrix}\varphi\\ \psi\end{pmatrix}^{(i)}(\underline{\mathbf{x}}, t) + \partial_t \int_{t_0}^{t} \mathrm{d}t' \int_{\partial S} \mathrm{d}l_{\underline{\mathbf{x}}'} \underline{n}(\underline{\mathbf{x}}') \cdot b_0(\underline{\mathbf{x}}', t', \underline{\mathbf{x}}, t) \cdot \begin{pmatrix}\varphi\\ \psi\end{pmatrix}(\underline{\mathbf{x}}', t'), \quad (\underline{\mathbf{x}}, t) \in \partial S \times T$$

where we have assumed that $\begin{bmatrix}\varphi\\ \psi\end{bmatrix}$ is continuous across $\partial S \times T$

$$\begin{pmatrix}\varphi\\ \psi\end{pmatrix}_{-}(\underline{\mathbf{x}}', t') = \begin{pmatrix}\varphi\\ \psi\end{pmatrix}_{+}(\underline{\mathbf{x}}', t'), \quad (\underline{\mathbf{x}}', t') \in \partial S \times T$$

We will in the following assume that ∂S is smooth clearly the generalization to piecewise smooth is simple.

If $\partial S = \cup_{i=1}^n c_i$ where each c_i is smooth we merely write

$$\int_{\partial S} = \sum_{i=1}^{n} \int_{c_i}$$

We will mainly consider simple boundaries in our calculations and therefore assume that the boundary is parametrized by some known function $\underline{\gamma}(s)$

$$\underline{\gamma} : [a,b] \to \mathbb{R}^2, \quad \underline{\gamma}([a,b]) = \partial S$$

Define

$$\begin{aligned}
\underline{U}(s,t) &= \begin{pmatrix} \varphi \\ \psi \end{pmatrix} (\underline{\gamma}(s),t), \quad U^{(i)}(s,t) = \begin{pmatrix} \varphi \\ \psi \end{pmatrix} (\underline{\gamma}(s),t) \\
A(s',t',s,t) &= \underline{n}(\underline{\gamma}(s')) \cdot b(\underline{\gamma}(s'),t',\underline{\gamma}(s),t)||\underline{\gamma}'(s')|| \\
B(s',t',s,t) &= \underline{n}(\underline{\gamma}(s')) \cdot b_0(\underline{\gamma}(s'),t',\underline{\gamma}(s),t)||\underline{\gamma}'(s')||
\end{aligned}$$

$$\begin{aligned}
\underline{U}(s,t) &= -\partial_t \int_{t_0}^{t} \mathrm{d}t' \int_a^b \mathrm{d}s' A(s',t',s,t) \cdot \underline{U}(s',t') \\
\underline{U}(s,t) &= \underline{U}^{(i)}(s,t) + \partial_t \int_{t_0}^{t} \mathrm{d}t' \int_a^b \mathrm{d}s' F(s',t',s,t) \cdot \underline{U}(s',t')
\end{aligned}$$

We must now discretize our system.

Introduce

$$\begin{aligned}
\alpha_i &= a + i\Delta s, \quad i = 0,\ldots,n, \quad \Delta s = \frac{b-a}{n} \\
I_i &= [\alpha_{i-1}, \alpha_i] \\
s_i &= \frac{1}{2}(\alpha_i + \alpha_{i-1}) = a + (i - \frac{1}{2})\Delta s, \quad i = 1,\ldots,n
\end{aligned}$$

Also introduce

$$\begin{aligned}
\underline{U}_i(t) = \underline{U}(s_i,t), &\quad \underline{U}_i^{(i)}(t) = \underline{U}^{(i)}(s_i,t) \\
A_i(s',t',t) = A(s',t',s_i,t), &\quad B_i(s',t',t) = B(s',t',s_i,t)
\end{aligned}$$

Then

$$\begin{aligned}
\underline{U}_i(t) &= -\partial_t \int_{t_0}^{t} \mathrm{d}t' \sum_{j=1}^{n} \int_{I_j} \mathrm{d}s' A_i(s',t',t) \cdot \underline{U}(s',t') \\
\underline{U}_i(t) &= \underline{U}_i^{(i)}(t) \\
&\qquad + \partial_t \int_{t_0}^{t} \mathrm{d}t' \sum_{j=1}^{n} \int_{I_j} \mathrm{d}s' B_i(s',t',t) \cdot \underline{U}(s',t')
\end{aligned}$$

If n is large enough Δs is so small that $\underline{U}(s',t')$ vary little over I_j and can therefore to a good approximation be taken outside the integral

$$\Rightarrow \underline{U}_i(t) = -\sum_{j=1}^{n} \partial_t \int_{t_0}^{t} \mathrm{d}t' M_{ij}(t',t) \cdot \underline{U}_j(t')$$

$$\underline{U}_i(t) = \underline{U}_i^{(i)}(t) + \sum_{j=1}^{n} \partial_t \int_{t_0}^{t} \mathrm{d}t' N_{ij}(t',t) \cdot \underline{U}(t')$$

where

$$M_{ij}(t',t) = \int_{I_j} \mathrm{d}s' A_i(s',t',t)$$

$$N_{ij}(t',t) = \int_{I_j} \mathrm{d}s' B_i(s',t',t)$$

Let us next introduce a timegrid $t_k = t_0 + k\Delta t$, $k = 1, 2, ...$

Let $J_k = [t_{k-1}, t_k]$ and let $(\partial_t)_k$ be a discrete approximation to the derivative at t_k. Also define

$$\underline{U}_i^k = \underline{U}_i(t_k), \quad \underline{U}_i^{(i)k} = \underline{U}_i^{(i)}(t_k)$$

then

$$\underline{U}_i^k = -\sum_{j=1}^{n} (\partial_t)_k \sum_{l=1}^{k} \int_{J_l} \mathrm{d}t' M_{ij}(t',t_k) \cdot \underline{U}_j(t')$$

$$\underline{U}_i^k = \underline{U}_i^{(i)k} + \sum_{j=1}^{n} (\partial_t)_k \sum_{l=1}^{k} \int_{J_l} \mathrm{d}t' N_{ij}(t',t_k) \cdot \underline{U}_j(t')$$

Clean up hese expressions by defining

$$M_{ij}^{kl} = \int_{J_l} \mathrm{d}t' M_{ij}(t',t_k) = \int_{J_l} \mathrm{d}t' \int_{I_j} \mathrm{d}s' A_i(s',t',t_k)$$

$$N_{ij}^{kl} = \int_{J_l} \mathrm{d}t' N_{ij}(t',t_k) = \int_{J_l} \mathrm{d}t' \int_{I_j} \mathrm{d}s' B_i(s',t',t_k)$$

Then

$$\underline{U}_i^k = -(\partial_t)_k \sum_{l=1}^{k} \sum_{j=1}^{n} M_{ij}^{kl} \cdot \underline{U}_j^l$$

$$\underline{U}_i^k = \underline{U}_i^{(i)k} + (\partial_t)_k \sum_{l=1}^{k} \sum_{j=1}^{n} N_{ij}^{kl} \cdot \underline{U}_j^l$$

where we assume $\underline{U}_j(t')$ vary little over the small interval J_l.

Recall that the integrands have a term of the type

$$(c^2(t-t')^2 - ||\underline{\gamma}(s) - \underline{\gamma}(s')||^2)^{-1/2}$$

Then when computing the matrix elements we are lead to integrals of the type

$$\begin{aligned} \alpha_{ij}^{kl} = \int_{J_l} \mathrm{d}t' \int_{I_j} \mathrm{d}s' (c^2(t-t')^2 - ||\underline{\gamma}(s) - \underline{\gamma}(s')||^2)^{-1/2} \\ \theta(c^2(t-t')^2 - ||\underline{\gamma}(s) - \underline{\gamma}(s')||^2) \end{aligned} \tag{2.9}$$

Different approximations for the matrix elements must be used depending on whether the integrand is singular inside $J_l \times I_j$ or not.

If it is not singular we can simply use the approximation

$$\begin{aligned} M_{ij}^{kl} &\approx \Delta t \Delta s A_i(s_j, t_l, t_k) \\ N_{ij}^{kl} &\approx \Delta t \Delta s B_i(s_j, t_l, t_k) \end{aligned} \tag{2.10}$$

If the integrand is singular we must use different approximating formulas depending on whether $(i,k) = (j,l)$ or not and what the geometry of the singularity look at.

We can use an automatic test to find (j,l) for the given (i,k) for which integrand (2.9) has singularity. For example we can use Python3 function:

```
def test(i,k):
    singlist = []
    for j in range(1,np.size(s)):
        for l in range(1,np.size(t)):
            try:
                INT = quad(lambda tt:
                            quad(lambda ss:
                            Hvs(c**2*(t[k]-tt)**2
                            - (gamma(s[i])- gamma(ss))**2)/
                            np.sqrt(c**2*(t[k]-tt)**2
                            - (gamma(s[i]) - gamma(ss))**2),
```

```
                    s[j-1],s[j], epsabs = ds, limit = 10)[0],
                    t[l-1], t[l],epsabs = dt, limit = 2)
        except ZeroDivisionError:
            singlist.append([np.nan,j,l])
            #Division by zero error
        if (INT[0] == np.inf):
            singlist.append([np.inf,j,l])
            #Infinity result of integration
    return singlist
```

where `tt` is t', `ss` is s', `gamma` is $\gamma(s)$, `quad(func,a,b)` – a Python3 function to integrate from scipy.integrate library, `Hvs(x)` – Heaviside function:

```
Hvs = lambda x: 0.5*(np.sign(x)+1)
```

Function `test(i,k)` returns list of pairs (j,l) where integral has singularity. It is not fast, but it can be improved by using another integration function, another programming language or more powerful processor.

Let us first assume that there is a singularity in $I_j \times J_l$ and $(i,k) \neq (j,l)$.

Introduce new coordinates

$$\begin{aligned} t' &= t_l - \eta \Delta t \\ s' &= s_j + \xi \Delta s \end{aligned} \tag{2.11}$$

where $\eta \in [0,1]$, $\xi \in [-\frac{1}{2}, \frac{1}{2}]$. Then

$$\begin{aligned} & c^2(t_k - t')^2 - ||\underline{\gamma}(s_i) - \underline{\gamma}(s')||^2 \\ & \approx c^2(t_k - t_l)^2 - ||\underline{\gamma}(s_i) - \underline{\gamma}(s_j)||^2 + 2c^2(t_k - t_l)\Delta t\eta \\ & + 2\underline{\gamma}'(s_j)(\underline{\gamma}(s_i) - \underline{\gamma}(s_j))\Delta s\xi \\ & = a_{ij}^{kl} + b^{kl}\eta + c_{ij}\xi \end{aligned} \tag{2.12}$$

where

$$\begin{aligned} a_{ij}^{kl} &= c^2(t_k - t_l)^2 - ||\underline{\gamma}(s_i) - \underline{\gamma}(s_j)||^2 \\ b^{kl} &= 2c^2(t_k - t_l)\Delta t \neq 0 \\ c_{ij} &= 2\underline{\gamma}'(s_j)(\underline{\gamma}(s_i) - \underline{\gamma}(s_j))\Delta s \end{aligned} \tag{2.13}$$

then

$$\begin{aligned} \alpha_{ij}^{kl} = -\Delta t \Delta s \int_0^t \mathrm{d}\eta \int_{-\frac{1}{2}}^{\frac{1}{2}} \mathrm{d}\xi (a_{ij}^{kl} + b^{kl}\eta + c_{ij}\xi)^{-\frac{1}{2}} \\ \theta(a_{ij}^{kl} + b^{kl}\eta + c_{ij}\xi) \end{aligned} \tag{2.14}$$

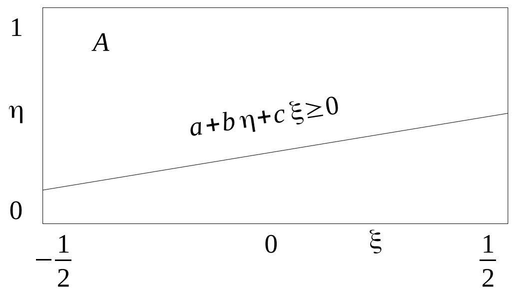

Figure 2.6: Geometry of the case when $(i,k) \neq (j,l)$ and the integrand (2.9) has singularity inside $I_j \times J_l$.

Let us simplify the notation by removing the indices i, j, k, l that are constant during this derivation.

Let us consider the case represented on Figure 2.6

$$\alpha = -\Delta t \Delta s \int_A \mathrm{d}\eta \, \mathrm{d}\xi (a + b\eta + c\xi)^{-\frac{1}{2}}$$

This is an 2D improper integral since $a + b\eta + c\xi = 0$ on the lover boundary of A, but the improper integral converge as we will see.

The lower boundary is given by

$$\eta = -\frac{a}{b} - \frac{c}{b}\xi \equiv \eta(\xi)$$

then

$$\begin{aligned}
\alpha &= -\Delta t \Delta s \int_{-\frac{1}{2}}^{\frac{1}{2}} \mathrm{d}\xi \int_{\eta(\xi)}^{1} \mathrm{d}\eta (a + b\eta + c\xi)^{\frac{1}{2}} \\
&= -\frac{2}{b} \Delta t \Delta s \int_{-\frac{1}{2}}^{\frac{1}{2}} \mathrm{d}\xi (a + b\eta + c\xi)^{\frac{1}{2}} \Big|_{\eta(\xi)}^{1} \\
&= -\frac{2}{b} \Delta t \Delta s \int_{-\frac{1}{2}}^{\frac{1}{2}} \mathrm{d}\xi (a + b + c\xi)^{\frac{1}{2}} \qquad (2.15) \\
&= -\frac{4}{3bc} \Delta t \Delta s (a + b + c\xi)^{\frac{3}{2}} \Big|_{-\frac{1}{2}}^{\frac{1}{2}} \\
&= \frac{4\Delta t \Delta s}{3bc} \left((a + b - \frac{1}{2}c)^{\frac{3}{2}} - (a + b + \frac{1}{2}c)^{\frac{3}{2}} \right)
\end{aligned}$$

Let us next look at the situation there is a singularity in $I_j \times J_l$ and $(i,k) = (j,l)$.

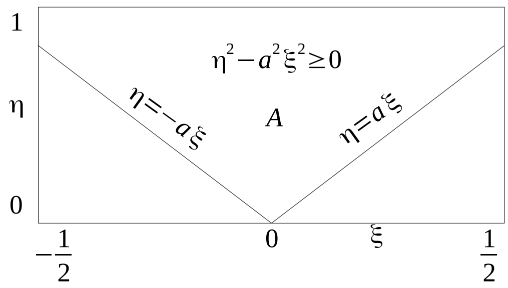

Figure 2.7: Geometry of the case when $(i,k) = (j,l)$ and the integrand (2.9) has singularity inside $I_j \times J_l$.

Using the same variables as before we now get

$$\begin{aligned} & c^2(t_k - t')^2 - ||\underline{\gamma}(s_i) - \underline{\gamma}(s')||^2 \\ & \approx c^2 \Delta t^2 \eta^2 - ||\underline{\gamma}'(s_i)||^2 \Delta s^2 \xi^2 \\ & = c^2 \Delta t^2 (\eta^2 - a_i^2 \xi^2), \quad a = \frac{||\underline{\gamma}'(s_i)|| \Delta s}{c \Delta t} \end{aligned} \tag{2.16}$$

Dropping indices we are lead to an integral

$$\alpha = -\frac{\Delta s}{c} \int_0^1 \mathrm{d}\eta \int_{-\frac{1}{2}}^{\frac{1}{2}} \mathrm{d}\xi (\eta^2 - a^2 \xi^2)^{-\frac{1}{2}} \theta(\eta^2 - a^2 \xi^2) \tag{2.17}$$

Let us consider the geometric situation shown on Figure 2.7

$$\alpha = -\frac{\Delta s}{c}\int_A \mathrm{d}\eta\,\mathrm{d}\xi(\eta^2 - a^2\xi^2)^{-\frac{1}{2}}$$
$$= -\frac{2\Delta s}{c}\int_0^{\frac{1}{2}}\mathrm{d}\xi\int_{a\xi}^{1}\mathrm{d}\eta(\eta^2 - a^2\xi^2)^{-\frac{1}{2}}$$

$$\eta = a\xi x, \quad \mathrm{d}\eta = a\xi\,\mathrm{d}x$$

$$\begin{aligned}
&= -\frac{2\Delta s}{c}\int_0^{\frac{1}{2}}\mathrm{d}\xi\int_1^{\frac{1}{a\xi}}\mathrm{d}x(x^2-1)^{-\frac{1}{2}}\\
&= -\frac{2\Delta s}{c}\int_0^{\frac{1}{2}}\mathrm{d}\xi\ln(x+\sqrt{x^2-1}\Big|_1^{\frac{1}{a\xi}}\\
&= -\frac{2\Delta s}{c}\int_0^{\frac{1}{2}}\mathrm{d}\xi\ln(\frac{1}{a\xi}+\sqrt{\frac{1}{a\xi}-1})\\
&= -\frac{2\Delta s}{c}\int_0^{\frac{1}{2}}\mathrm{d}\xi\ln\Big(\frac{1}{a\xi}(1+\sqrt{1-a^2\xi^2})\Big)\\
&= -\frac{2\Delta s}{c}\int_0^{\frac{1}{2}}\mathrm{d}\xi\Big(-\ln a-\ln\xi+\ln(1+\sqrt{1-a^2\xi^2})\Big)\\
&= -\frac{2\Delta s}{c}\Big(-\frac{1}{2}\ln a-(\xi\ln\xi-\xi)\Big|_0^{\frac{1}{2}}+\int_0^{\frac{1}{2}}\mathrm{d}\xi\ln(1+\sqrt{1-a^2\xi^2})\Big)\\
&= -\frac{2\Delta s}{c}\Big(\frac{1}{a}\sin^{-1}(\frac{a}{2})-\frac{1}{2}\ln\frac{a}{2}+\frac{1}{2}\ln(1+\sqrt{1-\frac{a^2}{4}})\Big)
\end{aligned} \tag{2.18}$$

Equations (2.1) restricted to the S together with the integral identities (2.8) define the EOS formulation for the given 2D scattering problem.

Conclusion

In this thesis work we discussed two 1D scattering problems and one 2D scattering problem. For every problem was constructed EOS formulation. For both of the 1D scattering problems were made numerical implementation using Python3. Before numerical solution of toy models we solved test problem to choose more stable and accurate numerical method. The test problem was solved by using several methods: Central Finite Difference method, Modified Euler method, Lax-Wendroff method and by using ODE solver `odeint` from library for scientific calculations of Python3.

We found that in every solution of test problem by using different methods has an artefact in solution for $\rho(x,t)$ on the left boundary of the domain. Possibly oscillations of solutions for $\varphi(x,t)$ and $j(x,t)$ on the left boundary appeared because of this artefact in $\rho(x,t)$. To understand the nature of this artefact and to fix this problem we need to do additional research.

This work doesn't comprise numerical implementation of 2D scattering problem, so this is also subject for the futher research.

Appendix A

Test problem. Python3 code

```
import numpy as np
import scipy as sp
from scipy import integrate
from scipy.integrate import odeint

dt = 0.01 #time step
dx = 0.05 #space step
T = 12 #time maximum
X = 3 #boundary value
c = 2
alpha = 0.1
gamma = 0.1
b = 10

t = np.linspace(0,T,T/dt+1)
x = np.linspace(-X,X,2*X/dx + 2)

exactphi = lambda t,x: np.exp(-b*(x + 2*t - 7)**2)
exactjf = lambda t,x: 0.1*np.exp(-b*(x + 2*t - 7)**2)
exactrho = lambda t,x: -0.1*np.exp(-b*(x + 2*t - 7)**2)

j1 = lambda t,x: ((20*c - 40)*(x + 2*t - 7) - 0.1)
        *np.exp(-10*(x + 2*t -7)**2)
j2 = lambda t,x: 2*(x + 2*t - 7)*np.exp(-10*(x + 2*t -7)**2)
j3 = lambda t,x: (-4*(x + 2*t - 7) - alpha + 0.1*gamma)
        *np.exp(-10*(x + 2*t -7)**2)

#Central Finite Difference method
CDphi = [[0]*np.size(x) for i in range(np.size(t))]
```

```
CDrho = [[0]*np.size(x) for i in range(np.size(t))]
CDj = [[0]*np.size(x) for i in range(np.size(t))]

for i in range(np.size(t)):
    CDphi[i][0] = exactphi(t[i],x[0])
    CDphi[i][-1] = exactphi(t[i],x[-1])

for j in range(np.size(x)):
    CDphi[0][j] = exactphi(t[0],x[j])
    CDphi[1][j] = exactphi(t[1],x[j])
    CDrho[0][j] = exactrho(t[0],x[j])
    CDrho[1][j] = exactrho(t[1],x[j])
    CDj[0][j] = exactjf(t[0],x[j])
    CDj[1][j] = exactjf(t[1],x[j])

for i in range(1,np.size(t)-1):
    for j in range(1,np.size(x)-1):
        CDphi[i+1][j] = c*dt/dx*(CDphi[i][j+1]
                        - CDphi[i][j-1])
                        + 2*dt*(j1(t[i],x[j]) + CDj[i][j])
                        + CDphi[i-1][j]
        CDrho[i+1][j] = -dt/dx*(CDj[i][j+1]
                        - CDj[i][j-1]) + 2*dt*j2(t[i],x[j])
                        + CDrho[i-1][j]
        CDj[i+1][j] = 2*dt*(alpha*CDphi[i][j]
                      - gamma*CDj[i][j]
                      + j3(t[i],x[j]))+ CDj[i-1][j]

    j=0
    CDrho[i+1][j] = -dt/dx*(-3*CDj[i][j] + 4*CDj[i][j+1]
                    - CDj[i][j+2]) + CDrho[i-1][j]
                    + 2*dt*j2(t[i],x[j])
    CDj[i+1][j] = 2*dt*(alpha*CDphi[i][j] - gamma*CDj[i][j]
                  + j3(t[i],x[j])) + CDj[i-1][j]

    j=-1
    CDrho[i+1][j] = dt/dx*(-3*CDj[i][j] + 4*CDj[i][j-1]
                    - CDj[i][j-2]) + CDrho[i-1][j]
                    + 2*dt*j2(t[i],x[j])
    CDj[i+1][j] = 2*dt*(alpha*CDphi[i][j] - gamma*CDj[i][j]
```

```
                    + j3(t[i],x[j])) + CDj[i-1][j]

#Lax-Wendroff method
LWphi = [[0]*np.size(x) for i in range(np.size(t))]
LWrho = [[0]*np.size(x) for i in range(np.size(t))]
LWj = [[0]*np.size(x) for i in range(np.size(t))]

for i in range(np.size(x)):
    LWrho[0][i] = exactrho(0,x[i])
    LWj[0][i] = exactjf(0,x[i])
    LWphi[0][i] = exactphi(0,x[i])

j1_dt = lambda t,x: 0.5/dt*(j1(t+dt,x) - j1(t-dt,x))
j1_dx = lambda t,x: 0.5/dx*(j1(t,x+dx) - j1(t,x-dx))
j2_dt = lambda t,x: 0.5/dt*(j2(t+dt,x) - j2(t-dt,x))
j2_dx = lambda t,x: 0.5/dx*(j2(t,x+dx) - j2(t,x-dx))
j3_dx = lambda t,x: 0.5/dx*(j3(t,x+dx) - j3(t,x-dx))

for i in range(1,np.size(t)):
    LWphi[i][-1] = exactphi(t[i],x[-1])
    LWphi[i][0] = exactphi(t[i],x[0])

for i in range(np.size(t)-1):
    for j in range(1,np.size(x)-1):

        LWphi[i+1][j] = LWphi[i][j] + dt*(0.5*c/dx*(LWphi[i][j+1]
        - LWphi[i][j-1]) + LWj[i][j]
        + j1(t[i],x[j]))+ 0.5*(dt**2)*(((c/dx)**2)*(LWphi[i][j+1]
        - 2*LWphi[i][j] + LWphi[i][j-1]) + c*j1_dx(t[i],x[j])
        + 0.5*c/dx*(LWj[i][j+1]- LWj[i][j-1])
        + alpha*LWphi[i][j] - gamma*LWj[i][j]
        + j1_dt(t[i],x[j]) + j3(t[i],x[j]))

        LWrho[i+1][j] = LWrho[i][j] + dt*(-0.5/dx*(LWj[i][j+1]
        - LWj[i][j-1]) + j2(t[i],x[j]))
        + 0.5*(dt**2)*(-alpha*0.5/dx*(LWphi[i][j+1] - LWphi[i][j-1])
        + 0.5*gamma/dx*(LWj[i][j+1] - LWj[i][j-1])
        - j3_dx(t[i],x[j]) + j2_dt(t[i],x[j]))

        LWj[i+1][j] = LWj[i-1][j] + 2*dt*(alpha*LWphi[i][j]
        - gamma*LWj[i][j]+ j3(t[i],x[j]))
```

```
    j = -1
    LWj[i+1][j] = LWj[i-1][j] + 2*dt*(alpha*LWphi[i][j]
    - gamma*LWj[i][j]+ j3(t[i],x[j]))
    LWrho[i+1][j] = LWrho[i][j] + dt*(-0.5/dx*(3*LWj[i][j]
    - 4*LWj[i][j-1] + LWj[i][j-2]) + j2(t[i],x[j]))
    + 0.5*(dt**2)*(-alpha*0.5/dx*(3*LWphi[i][j]
    - 4*LWphi[i][j-1] + LWphi[i][j-2])
    + 0.5*gamma/dx*(3*LWj[i][j]
    - 4*LWj[i][j-1] + LWj[i][j-2]) - j3_dx(t[i],x[j])
    + j2_dt(t[i],x[j]))

    j = 0
    LWj[i+1][j] = LWj[i-1][j] + 2*dt*(alpha*LWphi[i][j]
    - gamma*LWj[i][j]+ j3(t[i],x[j]))

    LWrho[i+1][j] = LWrho[i][j] + dt*(-0.5/dx*(-3*LWj[i][j]
    + 4*LWj[i][j+1] - LWj[i][j+2]) + j2(t[i],x[j]))
    - 0.5*(dt**2)*(-alpha*0.5/dx*(-3*LWphi[i][j]
    + 4*LWphi[i][j+1] - LWphi[i][j+2])
    + 0.5*gamma/dx*(-3*LWj[i][j]
    + 4*LWj[i][j+1] - LWj[i][j+2]) - j3_dx(t[i],x[j])
    + j2_dt(t[i],x[j]))

#Modified Euler method
MEphi = [[0]*np.size(x) for i in range(np.size(t))]
MErho = [[0]*np.size(x) for i in range(np.size(t))]
MEj = [[0]*np.size(x) for i in range(np.size(t))]

for i in range(np.size(x)):
    MErho[0][i] = exactrho(0,x[i])
    MEj[0][i] = exactjf(0,x[i])
    MEphi[0][i] = exactphi(0,x[i])

for i in range(1,np.size(t)):
    MEphi[i][-1] = exactphi(t[i],x[-1])
    MEphi[i][0] = exactphi(t[i],x[0])


for i in range(np.size(t)-1):
    for j in range(1,np.size(x)-1):
```

```
        K1 = dt*(0.5*c/dx*(MEphi[i][j+1] - MEphi[i][j-1])
            + MEj[i][j] + j1(t[i],x[j]))
        K2 = dt*(0.5*c/dx*(MEphi[i][j+1] - MEphi[i][j-1])
            + MEj[i][j] + j1(t[i+1],x[j]))
        MEphi[i+1][j] = MEphi[i][j] + 0.5*(K1+K2)

        K1 = dt*(-0.5/dx*(MEj[i][j+1] - MEj[i][j-1])
            + j2(t[i],x[j]))
        K2 = dt*(-0.5/dx*(MEj[i][j+1] - MEj[i][j-1])
            + j2(t[i+1],x[j]))
        MErho[i+1][j] = MErho[i][j] + 0.5*(K1+K2)

        K1 = dt*(alpha*MEphi[i][j] - gamma*MEj[i][j]
            + j3(t[i],x[j]))
        K2 = dt*(alpha*MEphi[i][j] - gamma*(MEj[i][j]+K1)
            + j3(t[i+1],x[j]))
        MEj[i+1][j] = MEj[i][j] + 0.5*(K1+K2)

    j = -1
    K1 = dt*(alpha*MEphi[i][j] - gamma*MEj[i][j]
        + j3(t[i],x[j]))
    K2 = dt*(alpha*MEphi[i][j] - gamma*(MEj[i][j]+K1)
        + j3(t[i+1],x[j]))
    MEj[i+1][j] = MEj[i][j] + 0.5*(K1+K2)

    K1 = dt*(-0.5/dx*(3*MEj[i][j] - 4*MEj[i][j-1]
        + MEj[i][j-2]) + j2(t[i],x[j]))
    K2 = dt*(-0.5/dx*(3*MEj[i][j] - 4*MEj[i][j-1]
        + MEj[i][j-2]) + j2(t[i+1],x[j]))
    MErho[i+1][j] = MErho[i][j] + 0.5*(K1+K2)

    j = 0
    K1 = dt*(alpha*MEphi[i][j] - gamma*MEj[i][j]
        + j3(t[i],x[j]))
    K2 = dt*(alpha*MEphi[i][j] - gamma*(MEj[i][j]+K1)
        + j3(t[i+1],x[j]))
    MEj[i+1][j] = MEj[i][j] + 0.5*(K1+K2)

    K1 = dt*(-0.5/dx*(-3*MEj[i][j] + 4*MEj[i][j+1]
        - MEj[i][j+2]) + j2(t[i],x[j]))
```

```
    K2 = dt*(-0.5/dx*(-3*MEj[i][j] + 4*MEj[i][j+1]
        - MEj[i][j+2]) + j2(t[i+1],x[j]))
    MErho[i+1][j] = MErho[i][j] + 0.5*(K1+K2)

#ODE solver
ODEphi = [[0]*np.size(x) for i in range(np.size(t))]
ODErho = [[0]*np.size(x) for i in range(np.size(t))]
ODEj = [[0]*np.size(x) for i in range(np.size(t))]
Nx = np.size(x)
M = [[0]*(Nx-2) for i in range(Nx-2)]
for i in range(1,Nx-2):
    M[i-1][i] = 0.5/dx
    M[i][i-1] = - 0.5/dx

for i in range(np.size(x)):
    ODErho[0][i] = exactrho(0,x[i])
    ODEj[0][i] = exactjf(0,x[i])
    ODEphi[0][i] = exactphi(0,x[i])

for i in range(1,np.size(t)):
    ODEphi[i][-1] = exactphi(t[i],x[-1])
    ODEphi[i][0] = exactphi(t[i],x[0])

def f(y,t):
    Y = y

    derives = []

    derives.extend([0.5*c/dx*(Y[1] - exactphi(t,x[0]))
                    + Y[2*Nx-1] + j1(t,x[1])])
    for i in range(1,Nx-2):
        derives.extend([c*M[i][0]*Y[0] + Y[2*Nx-1+i]
                    + j1(t,x[i+1])])
        for j in range(1,Nx-2):
            derives[i] = derives[i] +  c*M[i][j]*Y[j]

    derives[-1] = derives[-1] + 0.5*c/dx*exactphi(t,x[-1])

    derives.extend([-0.5/dx*(-3*Y[2*Nx-2]
                  + 4*Y[2*Nx-1] - Y[2*Nx])
                  + j2(t,x[0])])
```

```
    derives.extend([-0.5*(Y[2*Nx] - Y[2*Nx-2])/dx +
                    j2(t,x[1])])
    for i in range(1,Nx-2):
        derives.extend([-M[i][0]*Y[2*Nx-2+i] +
                        j2(t,x[i+1])])
        for j in range(1,Nx-2):
            derives[Nx+i-1] = derives[Nx+i-1]
                        - M[i][j]*Y[j-1+2*Nx]
    derives[-1] = derives[-1] - 0.5/dx*Y[-1]
    derives.extend([-0.5/dx*(3*Y[-1] - 4*Y[-2] + Y[-3])
                    + j2(t,x[-1])])

    derives.extend([alpha*exactphi(t,x[0]) - gamma*Y[2*Nx-2]
                    + j3(t,x[0])])
    for i in range(Nx-1):
        derives.extend([alpha*Y[i] - gamma*Y[i+2*Nx-1]
                        + j3(t,x[i+1])])
    derives.extend([alpha*exactphi(t,x[-1]) - gamma*Y[-1]
                   + j3(t,x[-1])])
    return derives

#Initial values
y0 = ODEphi[0][1:-1]+ODErho[0]+ODEj[0]

psoln = odeint(f, y0, t)

for i in range(np.size(t)):
    for j in range(1,Nx-1):
        ODEphi[i][j] = psoln[i][j-1]
    for j in range(Nx):
        ODErho[i][j] = psoln[i][j+Nx-2]
        ODEj[i][j] = psoln[i][j+Nx*2-2]
```

Appendix B

The first toy model. Python3 code

```
import numpy as np

Hvs = lambda t: 0.5*(np.sign(t)+1) #Heaviside function

a0 = -3 #left boundary
a1 = 3 #right boundary
dx = 0.05 # space step
dt = 0.01 # time step
T = 12 # Time maximum
alpha = 0.1
beta = 0.1
gamma = 0.1

t = np.linspace(0,T,T/dt+1)
x = np.linspace(a0,a1,(a1-a0)/dx+2)

c1 = 1.4
c2 = 2

#source function
a = 2
b = 10
c = 10
x0 = 10
t0 = 4
js = lambda x,t: a*np.exp(-b*(x-x0)**2)*np.exp(-c*(t-t0)**2)

phi = [[0]*np.size(x) for i in range(np.size(t))]
```

```
rho = [[0]*np.size(x) for i in range(np.size(t))]
jf = [[0]*np.size(x) for i in range(np.size(t))]
#integration formula for the right boundary
xnew = np.linspace(3,1000,997/dx+1)
for i in range(1,np.size(t)):
    phi[i][-1] = 1/c2*integrate.simps(Hvs(a1-xnew+c2*t[i])
                *js(xnew,t[i]-(xnew - a1)/c2),xnew,dx)

for i in range(np.size(t)-1):
    for j in range(1,np.size(x)-1):
        phi[i+1][j] = phi[i][j] + dt*(0.5*c1/dx*(phi[i][j+1]
        - phi[i][j-1]) + jf[i][j])+ 0.5*(dt**2)
        *(((c1/dx)**2)*(phi[i][j+1]- 2*phi[i][j]
        + phi[i][j-1])
        + 0.5*c1/dx*(jf[i][j+1]- jf[i][j-1])
        + (alpha - beta*rho[i][j])*phi[i][j]
        - gamma*jf[i][j])

        rho[i+1][j] = rho[i][j] + dt*(-0.5/dx*(jf[i][j+1]
        - jf[i][j-1]))
        + 0.5*(dt**2)*(beta*phi[i][j]*0.5/dx*(rho[i][j+1]
        - rho[i][j-1])
        - (alpha - beta*rho[i][j])*0.5/dx*(phi[i][j+1]
        - phi[i][j-1])
        + 0.5*gamma/dx*(jf[i][j+1] - jf[i][j-1]))

        jf[i+1][j] = jf[i-1][j] + 2*dt*((alpha
        - beta*rho[i][j])*phi[i][j] - gamma*jf[i][j])

    j = -1
    jf[i+1][j] = jf[i-1][j] + 2*dt*((alpha
    - beta*rho[i][j])*phi[i][j]
    - gamma*jf[i][j])
    rho[i+1][j] = rho[i][j] + dt*(-0.5/dx*(3*jf[i][j]
    - 4*jf[i][j-1]
    + jf[i][j-2]))+ 0.5*(dt**2)*(beta*phi[i][j]
    *0.5/dx*(3*rho[i][j]
    - 4*rho[i][j-1]      + rho[i][j-2]) - (alpha
    - beta*rho[i][j])
    *0.5/dx*(3*phi[i][j]- 4*phi[i][j-1] + phi[i][j-2])
```

```
+ 0.5*gamma/dx*(3*jf[i][j] - 4*jf[i][j-1] + jf[i][j-2]))

j = 0
jf[i+1][j] = jf[i-1][j] + 2*dt*((alpha
- beta*rho[i][j])*phi[i][j]
- gamma*jf[i][j])

ti = int((t[i+1] - 1/c1*(a1-a0))/dt)

phi[i+1][j] = Hvs(a0-a1+c1*t[i+1])*phi[ti][-1] +
1/c1*dx*((Hvs(c1*t[i+1])*jf[i+1][0]
+ Hvs(a0-a1+c1*t[i+1])*jf[int((t[i+1]
- 1/c1*(a1-a0))/dt)][-1])/2
+ np.sum(Hvs(a0-x[k]+c1*t[i+1])*jf[int((t[i+1]
- 1/c1*(x[k]-a0))/dt)][k]
for k in range(1,np.size(x))))

rho[i+1][j] = rho[i][j] + dt*(-0.5/dx*(-3*jf[i][j]
+ 4*jf[i][j+1]
- jf[i][j+2]))+ 0.5*(dt**2)*(beta*phi[i][j]
*0.5/dx*(-3*rho[i][j] + 4*rho[i][j+1]- rho[i][j+2])
- (alpha - beta*rho[i][j])*0.5/dx*(-3*phi[i][j]
+ 4*phi[i][j+1] - phi[i][j+2]) + 0.5*gamma/dx
*(-3*jf[i][j]
+ 4*jf[i][j+1]- jf[i][j+2]))
```

Appendix C

The second toy model. Python3 code

```
import numpy as np

Hvs = lambda t: 0.5*(np.sign(t)+1) #Heaviside function

def ind(i):
    if i<0:
        return(0)
    else:
        return(i)

a0 = -3 #left boundary
a1 = 3 #right boundary
dx = 0.05 # space step
dt = 0.01 # time step
T = 12 # Time maximum
alpha = 0.1
beta = 0.1
gamma = 0.1

t = np.linspace(0,T,T/dt+1)
x = np.linspace(a0,a1,(a1-a0)/dx+2)

Nx = np.size(x)
Nt = np.size(t)

m1 = 1.4
m2 = 2
```

```
n1 = 1
n2 = 1
c1 = np.sqrt(m1*n1)
c2 = np.sqrt(m2*n2)

#source function
a = 2
b = 10
c = 10
x0 = 6
t0 = 4
js = lambda x,t: a*np.exp(-b*(x-x0)**2)*np.exp(-c*(t-t0)**2)

phi = [[0]*np.size(x) for i in range(np.size(t))]
psi = [[0]*np.size(x) for i in range(np.size(t))]
phi_i = [0]*np.size(t)
psi_i = [0]*np.size(t)
rho = [[0]*np.size(x) for i in range(np.size(t))]
jf = [[0]*np.size(x) for i in range(np.size(t))]

xnew = np.linspace(a1,1000,(1000-a1)/dx+1)
for i in range(1,np.size(t)):
    phi_i[i] = 0.5/c2*integrate.simps(Hvs(np.sign(c2*t[i]
    - (xnew - a1)))*js(xnew,t[i]-(xnew - a1)/c2),xnew,dx)
    psi_i[i] = n2/c2*phi_i[i]

A0 = np.array([[c1+c2, m1-m2],[n1-n2,c1+c2]])
A1 = np.array([[c1+c2, m2-m1],[n2-n1,c1+c2]])
A0s = np.array([[c1,m1],[n1,c1]])
A1s = np.array([[c1,-m1],[-n1,c1]])

for i in range(1,Nt-1):
    for j in range(1,Nx-1):
        phi[i+1][j] = phi[i][j] + dt*(0.5*m1/dx*(psi[i][j+1]
        - psi[i][j-1]) + jf[i][j])
        + 0.5*(dt**2)*(((c1/dx)**2)*(phi[i][j+1]
        - 2*phi[i][j] + phi[i][j-1])
        + (alpha - beta*rho[i][j])*phi[i][j] - gamma*jf[i][j])

        psi[i+1][j] = psi[i][j] + dt*(0.5*n1/dx
```

```
    *(phi[i][j+1]
    - phi[i][j-1])) + 0.5*(dt**2)*(n1*(m1*(1/dx)**2
    *(psi[i][j+1]
    - 2*psi[i][j] + psi[i][j-1]) + 0.5/dx*(jf[i][j+1]
    - jf[i][j-1])))

    rho[i+1][j] = rho[i][j] + dt*(-0.5/dx*(jf[i][j+1]
    - jf[i][j-1]))
    + 0.5*(dt**2)*(beta*phi[i][j]*0.5/dx*(rho[i][j+1]
    - rho[i][j-1])
    - (alpha - beta*rho[i][j])*0.5/dx*(phi[i][j+1]
    - phi[i][j-1])
    + 0.5*gamma/dx*(jf[i][j+1] - jf[i][j-1]))

    jf[i+1][j] = 2*dt * (alpha - beta*rho[i][j])
    *phi[i][j]
    - 2*dt*gamma*jf[i][j] + jf[i-1][j]

j = -1
jf[i+1][j] = 2*dt*(alpha - beta*rho[i][j])*phi[i][j]
- 2*dt*gamma*jf[i][j] + jf[i-1][j

rho[i+1][j] = rho[i][j] + dt*(-0.5/dx*(3*jf[i][j]
- 4*jf[i][j-1]
+ jf[i][j-2]))+ 0.5*(dt**2)*(beta*phi[i][j]*0.5/dx
*(3*rho[i][j]
- 4*rho[i][j-1]      + rho[i][j-2]) - (alpha
- beta*rho[i][j])*0.5/dx*(3*phi[i][j]- 4*phi[i][j-1]
+ phi[i][j-2]) + 0.5*gamma/dx*(3*jf[i][j] - 4*jf[i][j-1]
+ jf[i][j-2]))

INT1 = dx*(0.5*(Hvs(c1*t[i+1] - (a1-a0))
*jf[ind(int((t[i+1]
- (a1-a0)/c1)/dt))][0] + Hvs(c1*t[i+1])*jf[i+1][-1])
+ np.sum(Hvs(c1*t[i+1] - (a1 - x[k]))*jf[ind(int((t[i+1]
- (a1-x[k])/c1)/dt))][k] for k in range(1,Nx-1)))

INT11 = 1/c1*(np.dot(A1s,[INT1,0]))
INT12 = Hvs(c1*t[i+1] - (a1-a0))
*np.dot(-A1s,[phi[ind(int((t[i+1]
```

```
- (a1-a0)/c1)/dt))][0],psi[ind(int((t[i+1]
- (a1-a0)/c1)/dt))][0]])

B1 = INT11+INT12+2*c2*np.array([phi_i[i+1],psi_i[i+1]])

phi[i+1][-1],psi[i+1][-1] = np.linalg.solve(A1,B1)

j = 0
jf[i+1][j] = 2*dt*(alpha - beta*rho[i][j])*phi[i][j]
- 2*dt*gamma*jf[i][j] + jf[i-1][j]

rho[i+1][j] = rho[i][j] + dt*(-0.5/dx*(-3*jf[i][j]
+ 4*jf[i][j+1]
- jf[i][j+2]))+ 0.5*(dt**2)*(beta*phi[i][j]*0.5/dx
*(-3*rho[i][j]
+ 4*rho[i][j+1]       - rho[i][j+2]) - (alpha
- beta*rho[i][j])*0.5/dx*(-3*phi[i][j]+ 4*phi[i][j+1]
- phi[i][j+2]) + 0.5*gamma/dx*(-3*jf[i][j]
+ 4*jf[i][j+1]- jf[i][j+2]))

INT0 = dx*(0.5*(Hvs(c1*t[i+1])*jf[i+1][0] + Hvs(c1*t[i+1]
- (a1-a0))*jf[ind(int((t[i+1] - (a1-a0)/c1)/dt))][-1])
+ np.sum(Hvs(c1*t[i+1] - (x[k] - a0))*jf[ind(int((t[i+1]
- (x[k]-a0)/c1)/dt))][k] for k in range(1,Nx-1)))

INT01 = 1/c1*(np.dot(A0s,[INT0,0]))
INT02 = Hvs(c1*t[i+1] - (a1-a0))
*np.dot(A0s,[phi[ind(int((t[i+1]
- (a1-a0)/c1)/dt))][-1],psi[ind(int((t[i+1]
- (a1-a0)/c1)/dt))][-1]])

B0 = INT01+INT02
phi[i+1][0],psi[i+1][0] = np.linalg.solve(A0,B0)
```

References

[1] SciPy.org. 2017. `scipy.integrate.odeint` – SciPy v0.18.1 Reference Guide. [ONLINE] Available at: `https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.integrate.odeint.html`. [Accessed 3 May 2017].

[2] J. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. Journal of computational physics, 114(2):185, 1994.

[3] H. F. C. Brehm. Immersed interface method for solving the incompressible Navier-Stokes equations with moving boundaries. 49th AIAA Aerospace Science meeting Including the New Horizons Forum and Aerospace Science Exposition 2011, 2011.

[4] A. T. et. al. Computational Electrodynamics: the Finite-Difference Time Domain Method, 3rd Ed. Artech House Publishers, 2005.

[5] S. Gedney. An anisotropic perfectly matched layer absorbing media for the truncation of FTDT lattices. IEEE Transactions on Antennas and Propagation, 44(12):1630, 1996.

[6] D. N. Pattanayak and E. Wolf. General form and new interpretation of the Ewald-Oseen extinction theorem. Optics communications, 6(3):217, 1972.

[7] L. Rezolla, Numerical Method for the Solution of partial differential equations, Lecture Notes for the COMPSTAR School on Computational Astrophysics, 8-13/02/10, Caen, France, 2011.

[8] G.D. Smith, Numerical Solution of partial differential equations: Finite Difference Methods, Brunel University, 3^{d} ed., Clarendon Press, Oxford, 1986.

[9] E. Süli and D. F. Mayers, An Introduction to Numerical Analysis, University of Oxford, Cambridge University Press,2003.

[10] A. Taflove. Application of the finite-difference time-domain method to sinusoidal steady state electromagnetic penetration problems. IEEE Transaction on Electromagnetic Compatibility, 22(3):191, 1980.

[11] K. Yee. Numerical solution of initial boundary value problems involving Maxwell's equation in isotropic media. IEEE Transaction on Antennas and Propagation, 14(3):302, May 1966.