LABORATORY RESEARCH PROJECT (PRL)

Ultrafast nonlinear acoustics in free-standing membranes at the nanoscale

June 2024

Peter Sayegh[1]
Under the supervision of Dr. Vasily Temnov[2]



[1]Email: peter.sayegh@polytechnique.com
[2]Email: vasily.temnov@cnrs.fr

Contents

1 Introduction **1**

2 Outline **1**

3 Experimental Data **1**

4 Theoretical background **2**
4.1 Nonlinearity and Dispersion 2
4.2 Solitons and Energy Levels 2
4.3 Reflection, Transmission, and Acoustic Impedance 3

5 Soliton Dynamics **4**
5.1 The Korteweg-de Vries (KdV) Equation 4
5.2 Numerical Solution of the KdV Equation 6
5.2.1 Scaling the equation 6
5.2.2 Initialization and Parameters 7
5.2.3 Algorithm overview 7
5.2.4 Testing and Verification of the Numerical Solution 8
5.3 Compression and Time-Reversal of Solitons 9
5.4 Reflection from a free interface 11
5.4.1 The Hamiltonian of the system 11
5.4.2 Dynamics of the energy levels of an acoustic pulse 12
5.5 Acoustic Attenuation 12

6 Results **15**
6.1 Main findings 15
6.2 Influence of the initial conditions and parameters 15
6.3 Practical implications 17

7 Conclusion **18**

A Matlab code used for simulations **19**

Abstract

This paper explores the properties of nonlinear acoustic strain wave propagation in solid crystals, integrating both experimental data analysis and numerical modeling. By introducing the Korteweg-De Vries (KdV) equation, we describe soliton propagation within certain nonlinear systems. The study simulates several physical scenarios, including the compression of a sequence of acoustic pulses, reflection at a free interface, and acoustic attenuation. These simulations provide insights into the complex behavior of nonlinear acoustic waves, which could have significant implications for the design and analysis of advanced materials and acoustic devices.

1 Introduction

Nonlinear acoustics at the nanoscale describes the dynamics of femtolaser-generated acoustic vibrations in solid nanostructures. Understanding these dynamics is crucial for advancing various applications in materials science, nanotechnology, and acoustic device engineering. Recent experimental breakthroughs have enabled the fabrication of macroscopic specimens of quasi-free-standing metallic membranes and multilayers, with thicknesses on the order of 100 nanometers. These advancements open new avenues for exploring the complex behavior of acoustic waves at the nanoscale.

However, conventional theoretical descriptions of nonlinear acoustic propagation, such as those relying on unidirectional wave equations like the Korteweg-de Vries (KdV) or Burgers equations, face significant limitations in this context. These models often fall short due to the continuous change in the propagation direction of acoustic transients upon reflection, which is common in nanoscale systems.

This study aims to address these limitations by combining experimental data analysis with numerical modeling to explore various properties of nonlinear acoustic strain wave propagation in solid crystals.

2 Outline

The project timeline spans over the spring semester, divided into several key phases. The initial phase involves a comprehensive review of relevant literature and theoretical frameworks to establish a solid foundation for the research. Subsequently, numerical models are developed and refined to simulate nonlinear wave propagation dynamics, with a particular emphasis on solving both the Boussinesq and the Korteweg-De Vries (KdV) equation. MATLAB serves as the primary computational tool for these simulations, employing techniques such as the Runge-Kunta method or the Split Step Fourier Method (SSFM) to analyze the solutions of the KdV equation. Furthermore, as an additional goal, the project aims to examine and simulate an experiment involving optical pulse manipulation. This experiment entails inputting an optical pulse, such as a Gaussian pulse, through an optical transducer composed of materials like Chromium (Cr) or Cobalt (Co). The pulse is subjected to expansion into a sequence of solitons due to nonlinearity and dispersion combined then see what happens with time-reversal when the formed solitons propagate backwards int he cavity. This experimental simulation aims to provide insights into the dynamic behavior of acoustic waves generated by the compression and expansion of optical pulses within solid nanostructures.

3 Experimental Data

In this section, we present the experimental data for two materials: Magnesium Oxide (MgO) in direction [100] and Gold (Au) in direction [0001]. The parameters measured include the speed of sound c_S, non-linearity parameter C_3, equilibrium mass density ρ_0, the dispersive parameter γ, and attenuation coefficient α_0 at a reference frequency of 1 GHz. The data was collected from literature references [1] and [3] respectively for MgO and gold and is summarized in Table 1.

Parameter	**MgO [100]**	**Au [111]**
Speed of Sound c_s $(nm.ps^{-1})$	9.05	3.45
Non-linearity Parameter C_3 $(g.nm^{-1}ps^{-2})$	-4.02×10^{-18}	-2.63×10^{-18}
Mass Density ρ_0 $(g.nm^{-3})$	3.585×10^{-21}	1.91×10^{-20}
Dispersive Parameter γ $(nm.ps^{-2})$	1.6×10^{-2}	7.41×10^{-3}
Attenuation Coefficient α_0 (nm^{-1})	6.7×10^{-8}	-

Table 1: Experimental data for Magnesium Oxide (MgO) and Gold (Au) in the respective directions **[100]** and **[111]**.

4 Theoretical background

In this section, we investigate the physical concepts that were key to our study and that we relied on for our numerical models.

4.1 Nonlinearity and Dispersion

Nonlinearity refers to the phenomenon where the wave amplitude influences the wave's speed, leading to phenomena such as harmonic generation and self-focusing. In contrast, dispersion describes how waves of different frequencies travel at different velocities, which can cause a pulse to spread out over time.

The dispersion relation for such a system is given by:

$$\omega(k) = c_s k - \gamma k^3$$

where: $\omega(k)$ is the angular frequency,c_s is the speed of sound in the medium, γ is a constant related to the strength of dispersion,k is the wave number.

The right balance between dispersion and non-linearity leads to the formation of solitary wave packets refered to as solitons. They are stable, localized wave packets that maintain their shape while traveling at constant velocity. This stability arises from the nonlinear effects counteracting the dispersive spreading of the wave packet. Their formation does not impact energy conservation. Indeed, if we denote by $W(t)$ the total energy of the system at time t, then one must have the following at all times t,

$$W(t) \propto \int_{-L}^{L} |u(x,t)|^2 dx = const$$

4.2 Solitons and Energy Levels

Solitons typically are of the form $u(x,t) = -A\,\mathrm{sech}^2(\frac{x-vt}{\xi})$ where ξ is their width given

$$\xi = \sqrt{\frac{24\rho_0 c_s \gamma}{A|C_3|}}$$

and v their velocity expressed as follows:

$$v = c_s \left(1 + \frac{|C_3|A}{6\rho_0 c_s^2}\right)$$

.
The connection between the energy levels of an acoustic system and solitons is rooted in the mathematical structure of the equations describing wave propagation. In particular, solitons in the KdV equation can be associated with the eigenvalues of a related Hamiltonian operator. This association is most easily seen through the context of the Schrödinger equation, given by:

$$\frac{d^2\psi}{dx^2} + (E - V(x))\psi = 0$$

where ψ denotes the wave function, E the energy eigenvalue and $V(x)$ is the potential related to the initial pulse profile.

The eigenvalues E_n of this equation (or more precisely its number of bound states) indicate the number of solitons and their amplitude. Moreover, as described by H.-J. Maris and W. Singhsomroje Hao in reference [2], if the initial pulse has amplitude A and width 2σ, then n solitons will appear if the following inequality is satisfied:

$$(n-1)^2 \leq \frac{C_3\sigma^2 A}{3\pi^2\rho_0 c_s\gamma} \leq n^2$$

Moreover, the arising solitons will have strain amplitudes

$$A_n = \frac{24\rho_0 c_s\gamma E_n}{C_3}$$

Both C_3 and E_n being negative leads to the amplitude being positive.

4.3 Reflection, Transmission, and Acoustic Impedance

When an acoustic wave encounters a boundary between two different media, part of the wave is reflected back into the original medium, and part is transmitted into the new medium. The behavior of the wave at the boundary is governed by the acoustic impedance of the two media.

Acoustic impedance, Z, is a property of a medium that describes the resistance an acoustic wave encounters as it travels through the medium. It is defined as:

$$Z = \rho c_s$$

where ρ is the density of the medium and c_s is the speed of sound in the medium. Acoustic impedance is crucial in determining the reflection and transmission of waves at an interface between two media.

The reflection coefficient, r, and transmission coefficient, t, describe the fractions of the wave's amplitude that are reflected and transmitted, respectively, when the wave encounters an interface. They are given by:

$$r = \frac{Z_2 - Z_1}{Z_2 + Z_1}$$

$$t = \frac{2Z_2}{Z_2 + Z_1}$$

where Z_1 and Z_2 are the acoustic impedances of the first and second media, respectively.

The reflection and transmission coefficients are related through the conservation of energy. For an acoustic wave, the sum of the squared magnitudes of the reflection and transmission coefficients

should equal 1, assuming there are no energy losses at the interface. Mathematically, this is expressed as:

$$|r|^2+\frac{Z_1}{Z_2}|t|^2= 1$$

By substituting the expressions for r and t into this equation, we can verify the energy conservation:

$$|r|^2= \left(\frac{Z_2 - Z_1}{Z_1 + Z_2}\right)^2$$

$$|t|^2= \left(\frac{2Z_2}{Z_2 + Z_1}\right)^2$$

$$|r|^2+\frac{Z_1}{Z_2}|t|^2= \left(\frac{Z_2 - Z_1}{Z_2 + Z_1}\right)^2 + \frac{Z_1}{Z_2}\left(\frac{2Z_2}{Z_2 + Z_1}\right)^2$$

$$|r|^2+\frac{Z_1}{Z_2}|t|^2= \frac{(Z_2 - Z_1)^2 + 4Z_1Z_2}{(Z_2 + Z_1)^2}$$

Thus, we get:

$$|r|^2+\frac{Z_1}{Z_2}|t|^2= \frac{(Z_2 + Z_1)^2}{(Z_2 + Z_1)^2} = 1$$

This confirms that the energy is conserved at the boundary, and the relationship between the reflection and transmission coefficients is physically valid. One can further introduce the reflectance $R = |r|^2$ and the transmittance $T = \frac{Z_1}{Z_2}|t|^2$ such that energy conservation translates to $R + T = 1$.

To represent a free boundary, one can consider a this surface is between the MgO crystal cavity and air or vacuum in which transmission would be very low. The speed of sound in air is $c_{air} = 340m.s^{-1} = 0.343nm.ps^{-1}$ and the equilibium density of air is $\rho_{air} = 1.225kg.m^{-1} = 1.225 \times 10^{-6}g.nm^{-3}$.

Thus, if 1 denotes the MgO cavity and 2 is for air, we can compute the refraction and reflection coefficients:

$$r = \frac{\rho_{air}c_{air} - \rho_0 c_s}{\rho_{air}c_{air} + \rho_0 c_s}, t = \frac{2\rho_{air}c_{air}}{\rho_{air}c_{air} + \rho_0 c_s}$$

The theoretical values we obtain are: $r \approx -1, t = 1.7 \times 10^{-13} \approx 0$. So one can deduce that more or less the incident wave will be totally reflected except for a very small fraction of it given by $|t|^2$, which is characteristic of a free interface.

5 Soliton Dynamics

5.1 The Korteweg-de Vries (KdV) Equation

In 1895, Diederik Johannes Korteweg together with his Ph.D student, Gustav de Vries, derived analytically a nonlinear partial differential equation, well known now as the "KdV equation." The KdV equation is used to model the disturbance of the surface of shallow water in the presence of solitary waves. The KdV equation is a generic model for the study of weakly nonlinear long waves, incorporating leading order nonlinearity and dispersion. The KdV equation in its simplest form is given by $u_t + auu_x + u_{xxx} = 0$. This equation incorporates two competing effects: nonlinearity represented by uu_x and linear dispersion represented by u_{xxx}. Nonlinearity tends to localize the wave while dispersion spreads it out.[4] For the rest of the section we consider the following KdV equation:

$$u_t + c_s u_x + \frac{C_3}{2\rho_0 c_s} u u_x + \gamma u_{xxx} = 0 \tag{1}$$

where u represents the strain relative displacement and the subscript denotes the partial derivative of u, i.e. u_t denotes the time-derivative of u, u_x the spatial derivative and u_{xxx} the third order spatial derivative. Our simulation can be depicted by the sketch below.

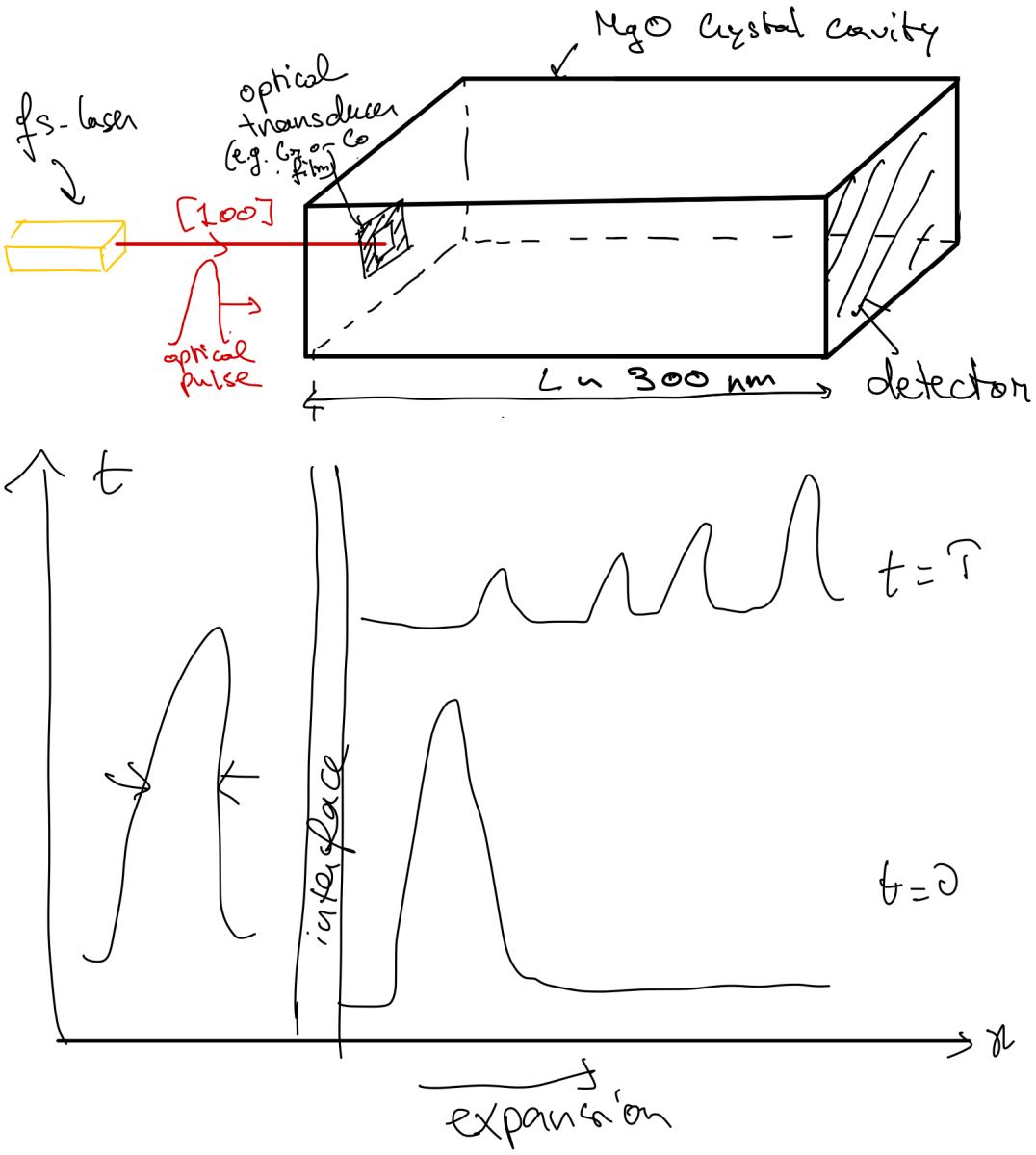


Figure 1: Sketch of the experiment we simulate and expected outcome

5.2 Numerical Solution of the KdV Equation

5.2.1 Scaling the equation

To make Equation 1 dimensionless, we introduce the following dimensionless variables:

$$\bar{x} = \frac{x}{x_c}, \bar{t} = \frac{t}{t_c}, \bar{u} = \frac{u}{u_c}$$

where x_c, t_c, and u_c are respectively characteristic position, time and strain value to be found by plugging them in our original KdV equation 1 such that we get to following equation:

$$u_t + uu_x + u_{xxx} = 0 \tag{2}$$

By doing so, one gets:

$$\frac{\partial u}{\partial t} = u_c \frac{d\bar{u}}{dt} = u_c \frac{d\bar{u}}{d\bar{t}} \frac{d\bar{t}}{dt} = \frac{u_c}{t_c} \frac{\partial \bar{u}}{\partial \bar{t}},$$

$$\frac{\partial u}{\partial x} = u_c \frac{d\bar{u}}{dx} = u_c \frac{d\bar{u}}{d\bar{x}} \frac{d\bar{x}}{dx} = \frac{u_c}{x_c} \frac{\partial \bar{u}}{\partial \bar{x}},$$

$$\frac{\partial^3 u}{\partial x^3} = \frac{\partial^2}{\partial x^2} \left(\frac{u_c}{x_c} \frac{\partial \bar{u}}{\partial \bar{x}} \right) = \frac{u_c}{x_c^3} \frac{\partial^3 \bar{u}}{\partial \bar{x}^3}$$

We plug these results in the original equation and get:

$$\frac{\partial \bar{u}}{\partial \bar{t}} + \frac{c_s t_c}{x_c} \frac{\partial \bar{u}}{\partial \bar{x}} + \frac{C_3}{2\rho_0 c_s} \frac{u_c t_c}{x_c} \bar{u} \frac{\partial \bar{u}}{\partial \bar{x}} + \frac{\gamma t_c}{x_c^3} \frac{\partial^3 \bar{u}}{\partial \bar{x}^3} = 0$$

We solve the following system to obtain the dimensionless equation:

$$\begin{cases} x_c = c_s t_c \\ t_c = \frac{x_c^3}{\gamma} \\ u_c = \frac{x_c}{t_c} \frac{2\rho_0 c_s}{C_3} \end{cases}$$

This can be written as:

$$\begin{cases} x_c = c_s t_c \\ t_c = \frac{c_s^3 t_c^3}{\gamma} \\ u_c = \frac{2\rho_0 c_s^2}{C_3} \end{cases}$$

Solving these, we get the solution:

$$\begin{cases} x_c = \sqrt{\frac{\gamma}{c_s}} \\ t_c = \sqrt{\frac{\gamma}{c_s^3}} \\ u_c = \frac{2\rho_0 c_s^2}{C_3} \end{cases}$$

We then disregard the second term in equation 1 and the solution we will obtain will correspond to the propagation of the pulse in a frame moving at the speed of sound in the cavity c_s. We drop the bar notation for simplicity and obtain the desired equation 2.

5.2.2 Initialization and Parameters

The simulation begins by initializing various parameters governing the system, including the spatial domain discretization, physical constants, and initial conditions for the wave profile.

Grid Parameters: The spatial domain is discretized using $N = 512$ points, which is used to define a discretized space so N being a power of 2 optimizes computations. The spatial grid $\bar{x}$ is defined as follows:

$$\bar{x} = m\left(\frac{2\pi}{N}\right)\left(-\frac{N}{2} : \frac{N}{2} - 1\right) \iff \bar{x} \in [-m\pi, m\pi]$$

We choose this particular domain for x to facilitate FFT handling. Here, $m = 1.5$ is a scaling factor for the grid spacing, ensuring appropriate coverage of the spatial domain. The grid spacing is given by:

$$\Delta x = \frac{\pi}{N}$$

For the time integration, the time step dt is calculated to be:

$$\Delta t = \frac{0.3}{N^2}$$

.

Indeed, it satisfies the Courant-Friedrichs-Lewy (CFL) condition which is a stability criterion for numerical solutions of partial differential equations. It can be expressed as $\Delta t \leq C(\Delta x)^\alpha$ where $C > 0$ depends on the equation and α on the solving method, $\alpha = 2$ for the FSSM. Given that ($\Delta x \propto \frac{1}{N}$), the CFL condition would suggest ($\Delta t \propto (\Delta x)^2 \propto \frac{1}{N^2}$). Then, the specific factor 0.3 was tuned empirically. It ensures that the numerical solution remains stable across a range of scenarios while also maintaining a reasonable runtime for simulations. For higher-order numerical schemes like the one we have, the time step often needs to be very small compared to the spatial step to ensure accuracy. The ($\frac{1}{N^2}$) dependence indicates a careful balance is being made to manage error propagation over the simulation. Moreover, we consider a propagation time $\tau \approx 300$ ps.

Physical Constants: The physical constants used in the simulation are the ones given in Table 1 mainly for Magnesium Oxyde crystal in direction **[100]**.

Initial Condition: The initial condition for the wave profile $u(x, t = 0)$ is defined by a Gaussian pulse centered at x_0, with amplitude A and full-width at half maximum (FWHM) 2σ. Mathematically, this is expressed as:

$$u(x, t = 0) = A \exp\left(-\frac{(x - x_0)^2}{2\sigma^2}\right)$$

5.2.3 Algorithm overview

The purpose of this algorithm is to solve the dimensionless Korteweg-de Vries (KdV) equation using a spectral method. By transforming the equation into the Fourier domain, we can efficiently handle both the linear and nonlinear parts at each iteration.

Starting from the dimensionless KdV equation (2), we apply the Fourier transform, denoted by the hat notation, to yield:

$$\frac{d\hat{u}}{dt} = -\frac{ik}{2}(\hat{u^2}) + ik^3\hat{u}$$

where $\hat{u}$ is the Fourier transform of u, k is the wavenumber, and t is time.
At each iteration, we solve the linear and nonlinear parts separately:
Linear Part:

$$\hat{u}_t = i\gamma k^3 \hat{u}$$

For the linear part, we see that $\hat{u}$ evolves according to the cubic term in the Fourier domain.
Nonlinear Part:

$$\hat{u}_t = -\frac{ik}{2}(\hat{u^2})$$

The nonlinear part involves a convolution in the Fourier domain, represented by the term $(\hat{u^2})$.
Iteration Step:
We combine these equations to perform the following iteration step:

$$\begin{cases} \hat{u}_1(k, t+\Delta t) = \hat{u}(k,t)e^{-ik^3\Delta t} \\ \hat{u}(k, t+\Delta t) = \hat{u}_1(k, t+\Delta t) - ik\Delta t\left(\mathcal{F}\left(\left(\mathcal{F}^{-1}(\hat{u}_1(k, t+\Delta t))\right)^2\right)\right) \end{cases}$$

$\hat{u}_1(k, t+\Delta t)$ represents the solution after the linear step. $\hat{u}(k, t+\Delta t)$ is the final updated solution incorporating the nonlinear step. $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and its inverse, respectively.

By alternating between these linear and nonlinear transformations, the algorithm efficiently steps forward in time to simulate the dynamics described by the KdV equation.

5.2.4 Testing and Verification of the Numerical Solution

To validate the accuracy of the obtained solutions, we performed a verification process by comparing the energy levels derived from solving the Schrödinger equation against the formation of solitons. This testing process ensures that the numerical solutions are consistent with theoretical expectations. The detailed steps and results are presented below.

Verification of Energy Levels The Schrödinger equation was solved numerically to determine the energy levels. The spatial domain was discretized, and appropriate boundary conditions were applied to compute the eigenvalues and eigenfunctions. The solitons were formed based on the initial conditions and parameters set in the numerical simulation, which included the amplitude, width of the initial pulse, and the coefficients related to the physical constants.

Figure 2 shows the potential and energy levels plotted alongside the eigenfunctions. The energy levels are depicted as dashed lines, and the soliton solutions are plotted to show their correspondence with these levels.

The graph in Figure 2 shows that the energy levels match closely with the positions of the formed solitons. This agreement confirms that the numerical solutions obtained are accurate and consistent with the theoretical predictions. Each eigenfunction (ψ_1, ψ_2, ψ_3) corresponds to a specific soliton, verifying that the energy levels are correctly computed and represented. We will be investigating the dynamics of energy levels of acoustics pulses further in the report.

Energy Conservation Energy conservation is a critical aspect of validating numerical simulations, especially for physical systems where energy is a key invariant.

In our simulation, we evaluated the energy conservation by calculating the energy at each time step using the trapezoidal integration method. The relative gap between the first and last value of the energy is 0.72%. This small change can be attributed to numerical approximation errors inherent in

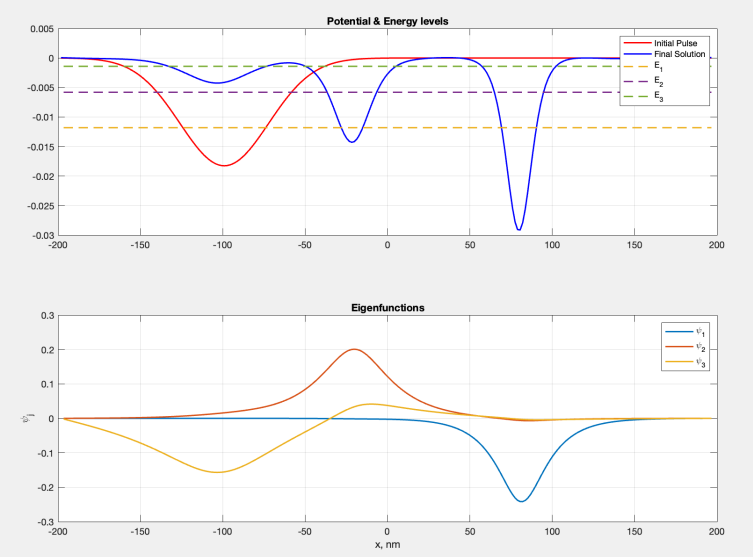


Figure 2: Graphical representation of the potential, energy levels, and eigenfunctions with N=512, A=0.018, $\sigma = 27nm, x_0 = -100nm$. The top subplot shows the potential and energy levels, while the bottom subplot displays the corresponding eigenfunctions.

the discretization and integration schemes used. Despite this minor discrepancy, the energy remains relatively stable, indicating that the numerical method employed is sufficiently accurate for capturing the essential dynamics of the system.

5.3 Compression and Time-Reversal of Solitons

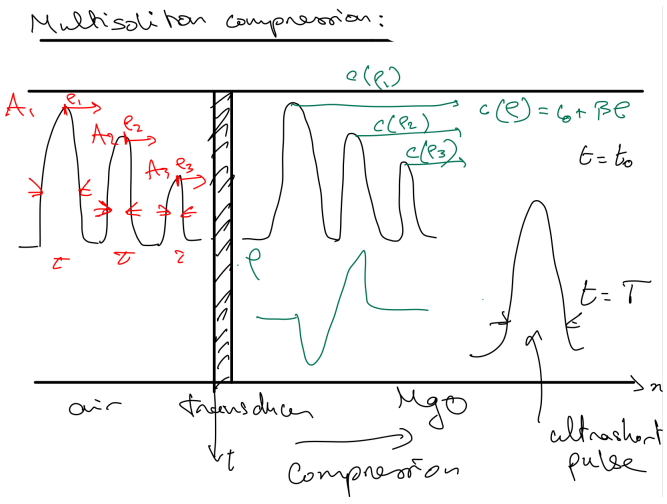


Figure 3: Explanatory drawing of the compression of a sequence of pulses to a single ultrashort pulse

In this part of the study, we investigate whether the solitons formed from an initial Gaussian pulse can be compressed back into the original pulse by time-reversing the process. This involves taking the final state of the forward simulation (where solitons have formed) and evolving it backward in

time. Indeed, we use the final state of the forward simulation as the initial condition for a reverse simulation. The drawing in Figure 3 illustrates the compression process.

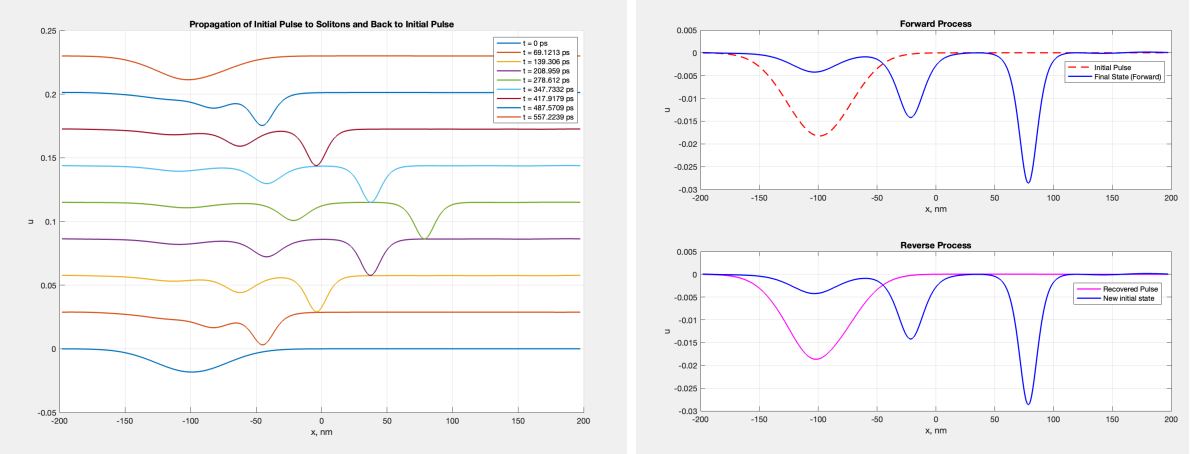


Figure 4: Evolution of the Gaussian pulse forwards and backwards with time going upwards (left) and comparison between forward propagation and time-reversal process(right)

Theory suggests that solitons with larger amplitudes propagate faster than those with smaller amplitudes. The resulting graphs in Figure 4 illustrate the process. The left plot shows the evolution of the Gaussian pulse forward and backward in time, demonstrating how the solitons initially form and then converge back into the original pulse during the time-reversal process. The right plot provides a comparison of the forward propagation and time-reversal process, highlighting the effective recovery of the initial pulse. The outcome of this numerical experiment is depicted the graphs in 4, obtained with the same initial conditions and grid parameters i.e. N=512, A=125, $\sigma = 27nm, x_0 = -100nm$.

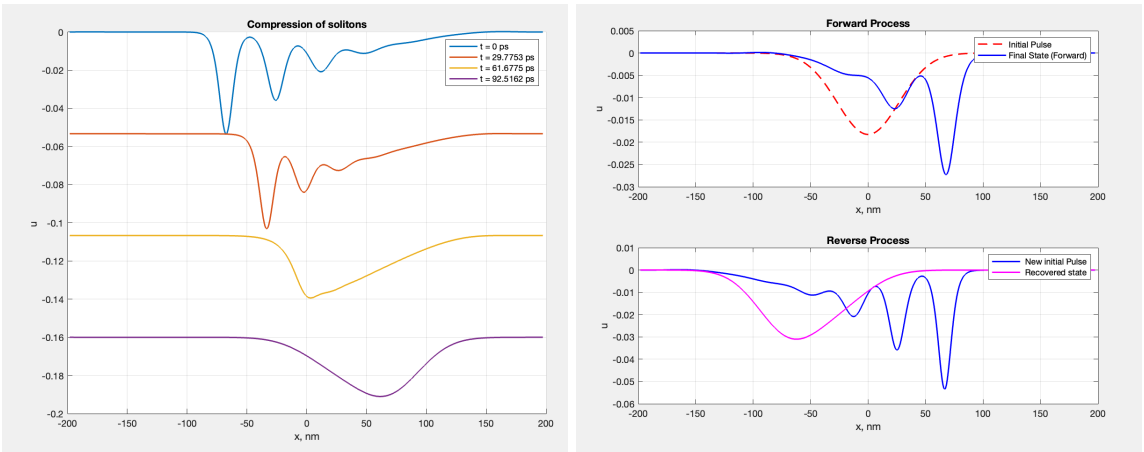


Figure 5: Evolution plot strain wave propagation in time and space (left) and comparison between initial and recovered pulse (right)

We slightly change our algorithm by locating the peaks positions and storing their amplitudes to approximate solitons by Gaussian pulses with the same width as the initial pulse and investigate

the recovered pulse through the time-reversal process instead of directly taking the final shape of the strain pulse as a new initial condition. We illustrate the result in Figure 5.

As one can see in Figure 5, we are still able to recover a giant ultrashort at the end of the propagation time, which closely resembles the initial pulse but not exactly. This gap can be explained by the approximation we used but also showcases the convergence the sequence of three Gaussian pulses to a this large pulse.

5.4 Reflection from a free interface

In this part, we are interested in the impact of reflection from a free membrane on the strain wave propagation and on the energy levels of the system.

5.4.1 The Hamiltonian of the system

The Schrödinger equation is a fundamental equation in quantum mechanics that describes how the quantum state of a physical system changes over time. It is given by:

$$\hat{H}\psi(x) = E\psi(x)$$

where $\hat{H}$ is the Hamiltonian operator, $\psi(x)$ is the wave function, and E is the energy eigenvalue.

The Hamiltonian operator $\hat{H}$ can be expressed as:

$$\hat{H} = -\frac{\partial^2}{\partial x^2} + V(x)$$

where $V(x)$ is the potential function.

For our system, we model the potential function $V(x)$ using the initial strain wave profile $u(x, 0)$:

$$V(x) = -\frac{C_3}{12\rho_0 c_s \gamma} u(x, 0)$$

To numerically solve the Schrödinger equation, we discretize the system and construct the Hamiltonian matrix. Here's a step-by-step outline of the process:

The Laplacian operator is a second-order differential operator commonly used in PDEs to describe diffusion, heat conduction, and wave propagation. In numerical simulations, the Laplacian operator is often discretized using finite difference methods. For a one-dimensional domain discretized into N points with spatial step size Δx, the second derivative with respect to x can be approximated using a central difference scheme. The discrete Laplacian matrix $\hat{D}$ for a one-dimensional system is given by:

$$\hat{D} = \frac{1}{\Delta x^2} \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & -2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -2 \end{pmatrix}$$

Then, the potential energy matrix $\hat{V}$ is a diagonal matrix where each diagonal element corresponds to the potential at a specific spatial point:

$$\hat{V} = \text{diag}(V(x_1), V(x_2), \ldots, V(x_{N-1}))$$

So the discrete Hamiltonian matrix can be rewritten $\hat{H} = -\hat{D} + \hat{V}$. From there, one can use Matlab built-in numerical solvers find the eigenvalues and eigenvectors of $\hat{H}$.

5.4.2 Dynamics of the energy levels of an acoustic pulse

A free interface in the context of wave propagation refers to a boundary condition where the medium is free to move without any external constraint like stress pressure. In a physical system, this could represent a membrane or a boundary where there is no restoring force acting on the medium. We are interested in simulating the propagation of an optical pulse between two free interfaces at $x = -L$ and $x = 0$. What one would expect is that the pulse would eventually decompose into solitons that would progressively reverse due to the phase inversion induced by reflection at a free interface. At the end of the propagation time, one should recover the initial pulse with an opposite sign. However, we don't know exactly how would that affect the energy levels of the pulse. Our results were not conclusive for this simulation as our algorithm runs on iterative time-steps we faced hardships locating the free interfaces spatially to apply the boundary conditions; our theoretical predictions are outlined in the drawing Figure 6 below.

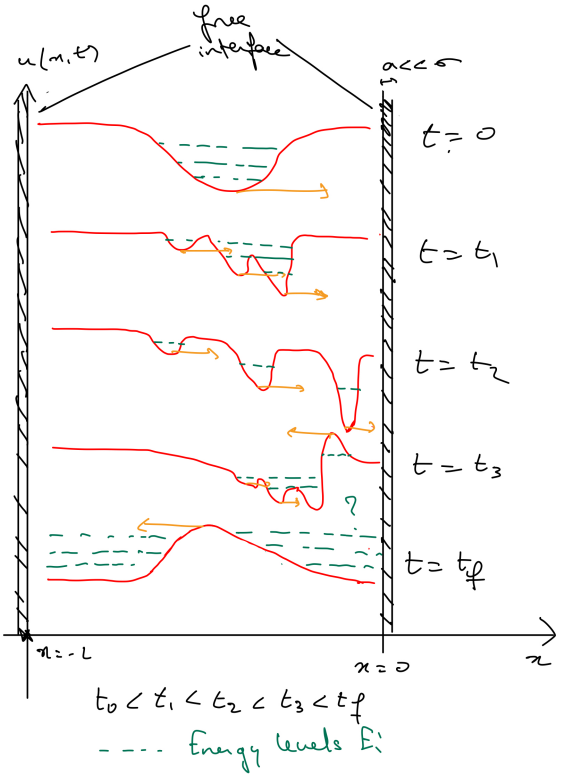


Figure 6: Sketch representing reflection of soliton or acoustic pulse at a free interface and possible impact and energy levels

5.5 Acoustic Attenuation

Attenuation refers to the gradual loss of intensity as a wave propagates through a medium. In the context of nonlinear acoustic waves in nanostructures, attenuation arises due to various mechanisms, including scattering, absorption, and intrinsic material properties. It is a crucial factor in determining the distance over which a wave can travel before its energy is significantly dissipated. One can also draw a sketch of how damping would afffect a soliton's shape and propagation, like in Figure 7.

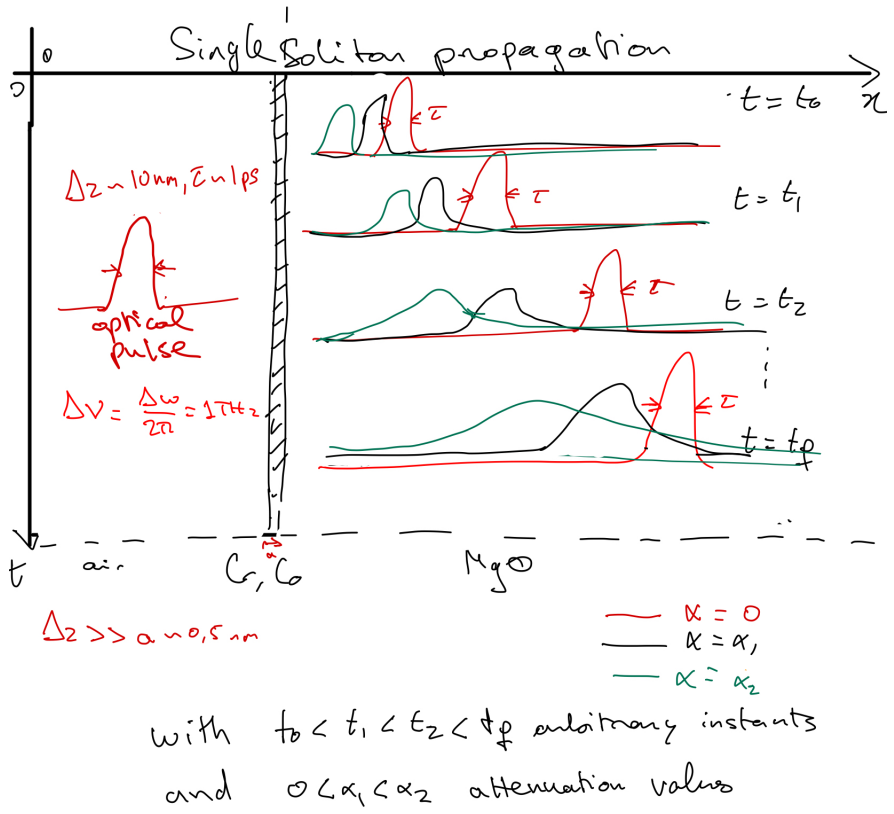


Figure 7: Qualitative drawing highlighting the damping effect on soliton propagation

The attenuation coefficient, $\alpha(k)$, quantifies the rate at which the wave's amplitude decreases with distance. For a given wave number k, the attenuation coefficient can be expressed as:

$$\alpha(k) = \alpha_0 \left(\frac{\omega(k)}{\omega_0} \right)^2$$

where α_0 is the reference attenuation coefficient at the reference frequency ω_0, and $\omega(k)$ is the dispersion relation of the system, given by:

$$\omega(k) = c_s k - \gamma k^3$$

Combining these, the attenuation coefficient becomes:

$$\alpha(k) = \alpha_0 \left(\frac{c_s k - \gamma k^3}{\omega_0} \right)^2$$

By introducing the attenuation term, the modified KdV equation becomes:

$$u_t + c_s u_x + \frac{C_3}{2\rho_0 c_s} u u_x + \gamma u_{xxx} + \alpha(k) c_s \eta = 0$$

where $\alpha(k)$ introduces the effect of damping. We plot this attenuation function applied on a localized range of wave number k and obtain Figure 8.

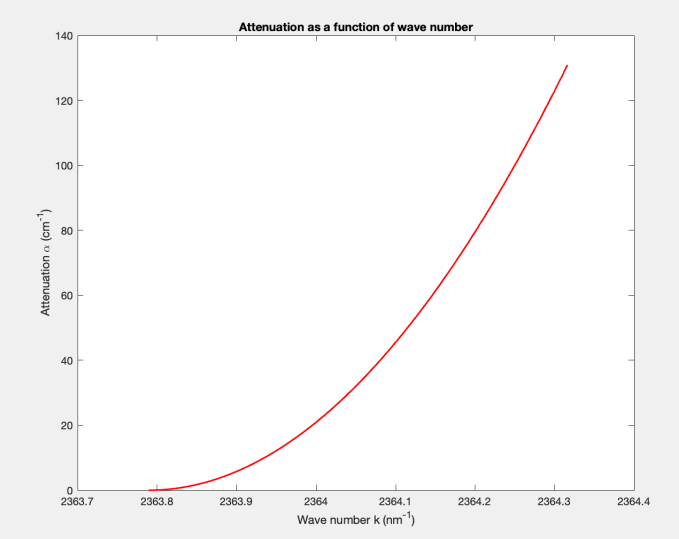


Figure 8: Plot of Acoustic attenuation with respect to wave number

In the Fourier domain, this becomes:

$$\frac{d\hat{u}}{dt} = -ikc_s\hat{u} - ik\frac{C_3}{2\rho_0 c_s}\widehat{(u^2)} + i\gamma k^3\hat{u} - \alpha(k)c_s\hat{u}$$

We consider this modification in the iterative time-step of the Fourier Split-Step algorithm used for compression and plot the results in Figure 9.

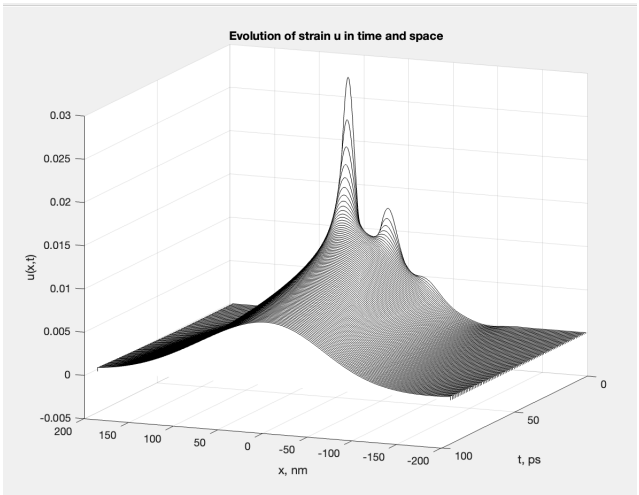


Figure 9: Compression of a sequence of Gaussian pulses in the damped case

One can notice that the convergence of the pulses is significantly faster than in the damped regime. However, damping results in a drastic energy loss, estimated to be of 60% in this case. Indeed, one would hence prefer to limit the damping to avoid energy loss since we know that the expansion into solitons is reversible even in the undamped case.

6 Results

6.1 Main findings

Our study reveals several key findings regarding the formation of solitons, the reversibility of soliton propagation, and the effects of reflection at a free interface. In our simulations, we observed the formation of solitons as a result of the balance between nonlinearity and dispersion in the system. This is evidenced by the sustained waveforms that emerge from the initial Gaussian pulse. The formation of solitons is crucial for understanding the dynamics of the system, as they represent a stable solution to the nonlinear wave equation.

An important characteristic of solitons is their reversibility. In our simulations, we reversed the direction of time to observe the behavior of the solitons. The results demonstrate that the solitons retrace their paths accurately, reaffirming the integrable nature of the system governed by the Korteweg-de Vries (KdV) equation.

6.2 Influence of the initial conditions and parameters

Changing the initial shape of the pulse In our simulations, we explored various initial conditions for the pulse shape, including Gaussian, Lorentz, Triangular, and Bessel pulses. Each pulse shape has a specific amplitude, width, and center. The mathematical formulations for these pulses are as follows, where A is the amplitude, σ is the width, and x_0 is the center. We showcase the resulting pulses we obtain given respectively those initial conditions in the following graphs.
The **Lorentz pulse** is defined as:

$$u(x) = \frac{A}{1 + \left(\frac{x - x_0}{\sigma}\right)^2}$$

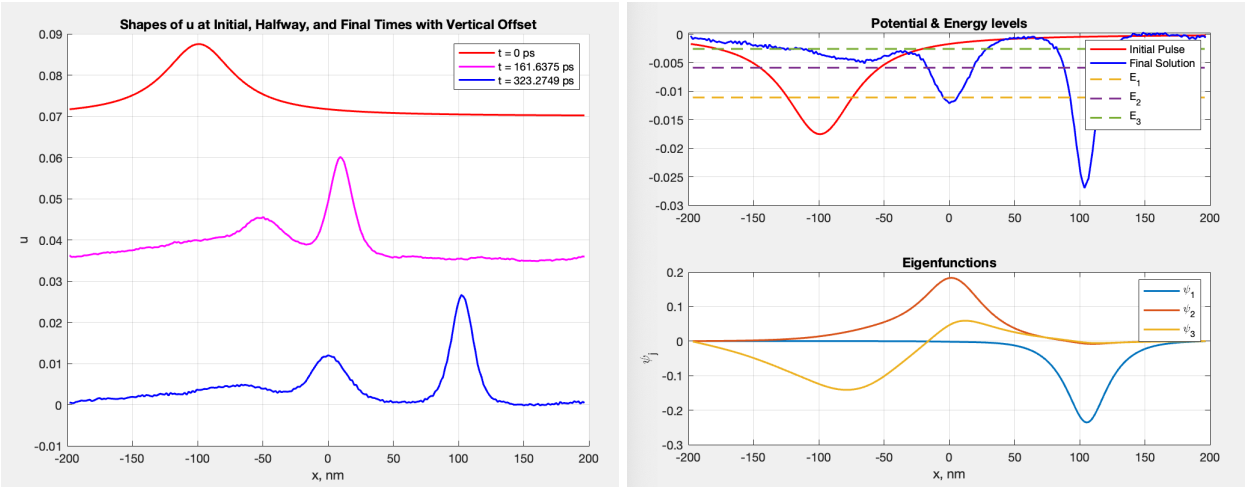


Figure 10: Results for a Lorentzian pulse with $A = 0.017$,$\sigma = 32nm$, $x_0 = -100nm$

The **Triangular pulse** is defined as:

$$u(x) = A \max\left(0, 1 - \left|\frac{x - x_0}{\sigma}\right|\right)$$

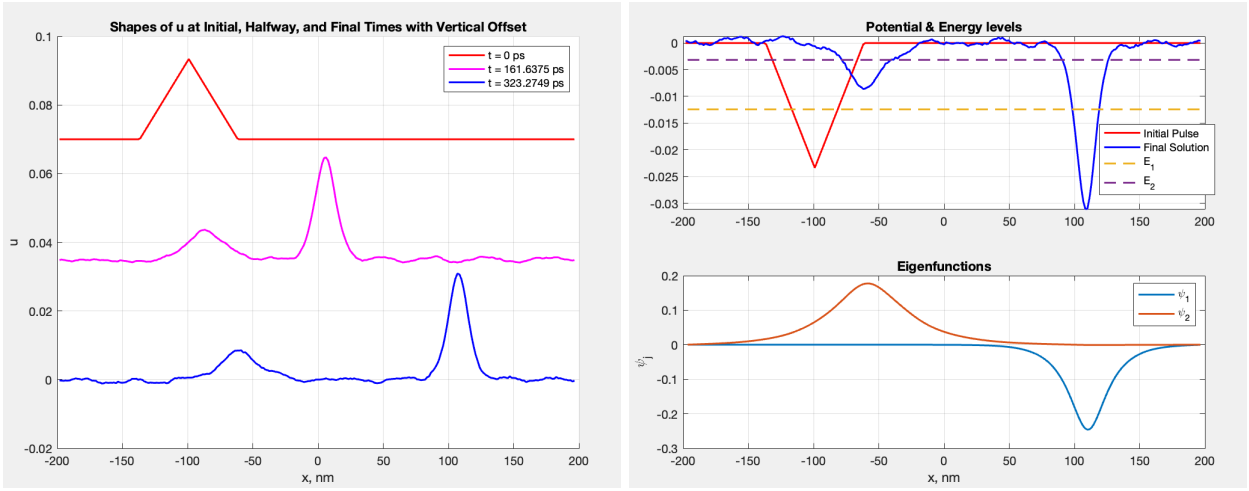


Figure 11: Results for a Triangular pulse with $A = 0.022$,$\sigma = 38nm$, $x_0 = -100nm$

The **Bessel pulse** is defined as:

$$u(x) = AJ_0\left(\frac{x - x_0}{\sigma}\right)$$

Here, J_0 represents the zeroth-order Bessel function of the first kind.

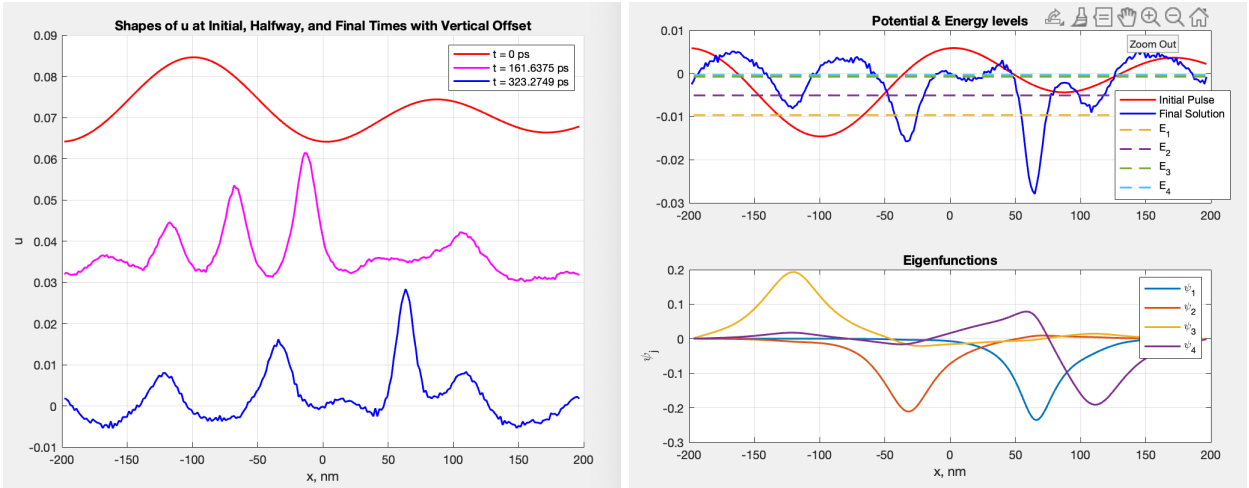


Figure 12: Results for a Bessel pulse with $A = 0.014$,$\sigma = 27nm$, $x_0 = -100nm$

One can deduce that our simulation is stable within a range of initial conditions and we see solitons forming and matching energy levels in all of the four cases we treated so far.

Changing the material's properties

Let's apply our propagation simulation with gold's material constants.

We see from Figure 13 that our simulation works for more than one material if the parameters are tuned and scaled correctly. The notable difference between gold and MgO is the speed of decomposition into solitons: it took 800 ps in gold whereas it would take less than 300 ps in MgO crystal.

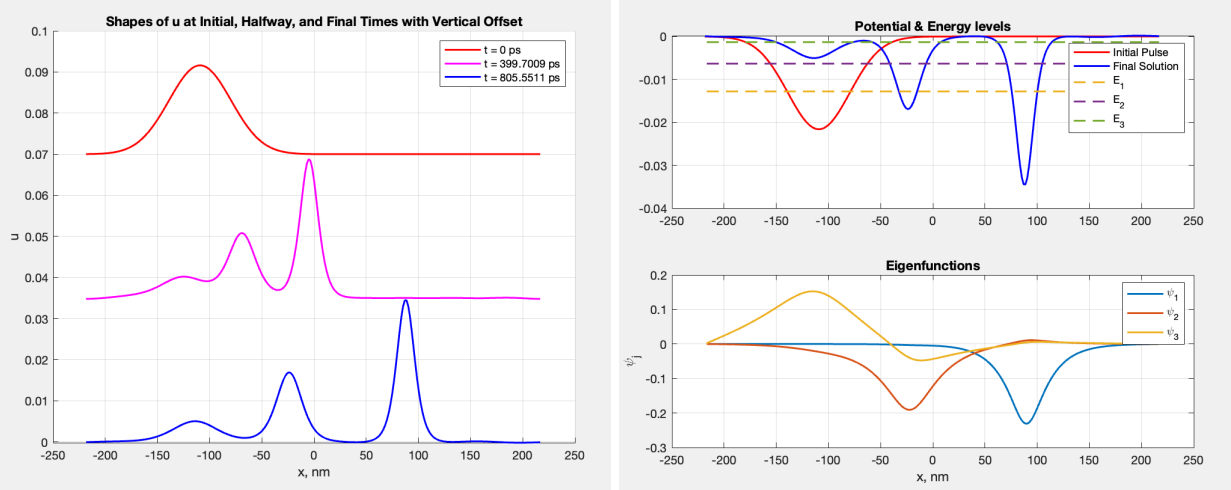


Figure 13: Evolution of Gaussian initial pulse of amplitude $A = 0.018$, half-width $\sigma = 27nm$, centered at $x_0 = -100nm$ in Gold (cf. Table 1) and Energy levels

6.3 Practical implications

The findings from this study have several important practical implications, particularly in the fields of telecommunications, materials science, and nonlinear optics.

The stable and reversible nature of solitons makes them highly suitable for applications in telecommunications. Solitons can propagate over long distances without changing their shape, which is essential for maintaining signal integrity in optical fiber communication systems. The ability to reverse soliton propagation implies that data encoded in soliton waveforms can be reliably transmitted and retrieved, reducing signal loss and distortion. This property can be leveraged to design more efficient and robust communication networks.

The principles observed in the behavior of acoustic solitons can also be extended to the field of nonlinear optics. Solitons in optical systems, known as optical solitons, exhibit similar properties of stability and phase inversion upon reflection. These characteristics are valuable for applications such as pulse shaping, optical switching, and the development of soliton-based lasers. The insights gained from this study can contribute to the advancement of nonlinear optical technologies, leading to more precise and efficient optical devices and strutural change detectors.

7 Conclusion

Overall, this paper has provided a comprehensive analysis of the dynamics of nonlinear acoustic waves in nanostructures, shedding light on the complex interplay between nonlinearity, dispersion, reflection, and attenuation. These findings have important implications for the design and optimization of nanostructured materials and devices in various technological applications.

Throughout this research, we faced several challenges. Initially, getting familiar with MATLAB and scaling partial differential equations was a significant hurdle. Solving these equations numerically required a deep understanding of both the mathematical framework and the computational techniques. Addressing stiff problems in numerical simulations was particularly challenging, but these experiences were invaluable for one's learning and development.

To push the study further, one can consider the three-dimensional problem instead of solving the one-dimension KdV equation. Moreover, the algorithms used in our simulations can be computationally heavy for a large number of data points so one may be interested in taking optimization to the next step and perform some algebraic reduction that would facilitate computations. Also, one could conceive another numerical approach relying on a spatial time for than a time-step iteration to simulate reflection in order to evaluate its impact on the energy levels of the system.

Acknowledgements

Special thanks to Dr. Vasily Temnov for supervising the project and for providing key guidance and insights all along the semester.

References

[1] H-Y Hao and Humphrey J Maris. Experiments with acoustic solitons in crystalline solids. *Physical review B*, 64(6):064302, 2001.

[2] Wisit Singhsomroje and Humphrey J. Maris. Generating and detecting phonon solitons in mgo using picosecond ultrasonics. *Phys. Rev. B*, 69:174303, May 2004.

[3] Vasily Temnov, Christoph Klieber, Keith Nelson, et al. Femtosecond nonlinear ultrasonics in gold probed with ultrashort surface plasmons. *Nature Communications*, 4:1468, 2013.

[4] Norman J Zabusky and Martin D Kruskal. Interaction of "solitons" in a collisionless plasma and the recurrence of initial states. *Phys. Rev. Lett.*, 15(6):240–243, 1965.

A Matlab code used for simulations

```
clear all; close all;
% Constants
N = 512; % Increase the number of points
c = 9.05; % [nm/ps]
gamma = 1.6e-2; % [nm^3 ps^-1]
C3 = -40.2e-19; % [g nm^-1 ps^-2]
rho = 3.585e-21; % [g nm^-3]
omega0 = 1e-3; % [ps^-1]
B2= C3/(2*rho*c);

%Scaling factors:
xc= 1000*sqrt(gamma/c); %[nm]
tc=1e6*sqrt(gamma/c^3); % [ps]
uc=0.001*c/B2; % dimensionless
Vc=c/gamma;
A=125; %Amplitude of initial pulse [dimensionless]
sg=sqrt(0.41);% width of initial pulse [dimensionless]


init_cond='gauss'; %change initial condition here
m=1.5;
x_bar = m*(2*pi/N)*(-N/2:N/2-1)';
dx = (x_bar(2) - x_bar(1));
dt = 0.3/N^2; % time step
T = 0.07; % final time
 % number of time steps

delta_k = 2*pi/(N*dx);
k = [0:delta_k:(N/2-1)*delta_k,0,-(N/2-1)*delta_k:delta_k:-delta_k];

x0 = -m*pi/2;
% Initial condition: Gaussian pulse
if strcmp(init_cond, 'gauss')
    A =120; % Amplitude of initial pulse [dimensionless]
    sg = sqrt(0.41); % width of initial pulse [dimensionless]
    u0 = @(x) A * exp(-(x-x0).^2 / (2 * sg^2));
elseif strcmp(init_cond, 'lorentz')
    A =120; % Amplitude of initial pulse [dimensionless]
    sg = sqrt(0.6); % width of initial pulse [dimensionless]
    u0 = @(x) A ./ (1 + ((x - x0) / sg).^2);
elseif strcmp(init_cond, 'triangular')
    A=160;
    sg=sqrt(0.8);
    u0 = @(x) A * max(0, 1 - abs((x - x0) /sg));
elseif strcmp(init_cond, 'bessel')
    A = 100;
    sg=sqrt(0.4);
    u0 =@(x) A * besselj(0, (x - x0) / sg);
elseif strcmp(init_cond, 'raised-cos')
    A=100;
    sg=sqrt(0.3);

    u0=@(x) A * (abs(x - x0) <= 2 * sg) .* (1 + cos(pi * (x - x0) / (2 * sg)));
```

```
elseif strcmp(init_cond, 'airy')
    A=150;
    sg=sqrt(0.8);
    a=0.06;
    x0=-m*pi/4;
    u0=@(x) A * airy((x - x0) / sg) .* exp(a * (x - x0) / sg);
end
u = u0(x_bar)';
disp(x0*xc)
disp(sg*xc)

% Precompute the FFT of the differentiation matrix
kx = 1i*k;
k3 = -k.^3;
Nt=round(T/dt);

% Determine the interval for storing data
store_interval = 100; % Save every 100 time steps
stored_steps = floor(Nt/store_interval) + 1;
usol = zeros(stored_steps, N);
time_values = zeros(stored_steps, 1);

% Store the initial condition
usol(1, :) = u;
time_values(1) = 0;

% Solve the Schrodinger equation to obtain initial eigenvalues and eigenfunctions
% Use a simple finite difference method for a potential well
P = @(x) Vc * u0(x); % Potential for simplicity


% Time-stepping loop
for n = 1:Nt
    u_hat = fft(u); % Compute the FFT of the current solution
    u_hat = u_hat .* exp(1i * k.^3 * dt);

    % Solve the non-linear part
    u_hat = u_hat - dt * (0.5 * 1i * k .* fft(real(ifft(u_hat)).^2));
    u = real(ifft(u_hat));

    % Store the solution at specified intervals
    if mod(n, store_interval) == 0
        idx = n/store_interval + 1;
        usol(idx, :) = u;
        time_values(idx) = n * dt;
    end
end
```

Listing 1: Solving the KdV with the FSSM algorithm

```

D=zeros(N- 1,N-1);
for k=1:size(D,1)
    D(k,k)= -2;
end
for k=1:size(D,1)-1
    D(k,k+1)= 1;
```

```
    D(k+1,k)= 1;
end
D=D/(dx^2);

B=zeros(N-1,N-1);
for k=1:size(B,1)
    B(k,k)= 10;
end
for k=1:size(B,1)-1
    B(k,k+1)= 1;
    B(k+1,k)= 1;
end
B=B/12;

V0=(B2/(6*gamma));
g= -V0*u*uc;

V=zeros(N-1,N-1);
for k=1:size(V,1)-1
    V(k,k)= - g(k);
 end


 [Y,s] = eig((B^-1)*D+V);
 E= -diag(s);

 Sss=sortrows([E,Y']);

 %Sss(1:5,1)

 figure;
  subplot(2,1,1);
  plot(x_bar*xc, usol(1,:)*uc, 'Color','r', LineWidth=1.5, DisplayName='Initial Pulse
    '); hold on;
  plot(x_bar*xc,-g/V0,DisplayName='Final Solution', Color= 'b', LineWidth=1.5); grid
    on; hold on;
  indexx=1;
  while Sss(indexx,1)<0
  plot(x_bar(2:end)*xc,-ones(1,length(x_bar)-1)*Sss(indexx,1)/V0,'--','LineWidth'
    ,1.5, DisplayName=['E_',num2str(indexx)]); hold on;
  indexx=indexx+1;
  end
  title('Potential & Energy levels')
  legend()
 indexx=1;
  subplot(2,1,2)
   while Sss(indexx,1)<0
  plot(x_bar(2:end)*xc,(Sss(indexx,2:end)),'LineWidth',1.5, 'DisplayName',['\psi_',
    num2str(indexx)]); grid on; hold on;
  indexx=indexx+1;
  end
 title('Eigenfunctions')
 xlabel('x, nm')
 ylabel('\psi_j')
 legend()
```

Listing 2: Find the eigenvalues and eigenfunctions

```
1
2  % Time-stepping loop for the forward process
3  for n = 1:Nt
4      u_hat = fft(u); % Compute the FFT of the current solution
5      u_hat = u_hat .* exp(1i * k.^3 * dt);
6
7      % Solve the non-linear part
8      u_hat = u_hat - dt * (0.5 * 1i * k .* fft(real(ifft(u_hat)).^2));
9      u = real(ifft(u_hat));
10     % Store the solution at specified intervals
11     if mod(n, store_interval) == 0
12         idx = n/store_interval + 1;
13         usol(idx, :) = u;
14         time_values(idx) = n * dt;
15     end
16 end
17
18 % Identify solitons by finding peaks in the final propagated waveform
19 final_waveform = (usol(end, :));
20
21 % Use findpeaks to identify peaks (solitons)
22 [peak_amplitudes, peak_locations] = findpeaks(final_waveform, 'MinPeakProminence',
       0.006);
23
24 % Extract soliton parameters
25 num_solitons = length(peak_amplitudes);
26 soliton_amplitudes = peak_amplitudes;
27 soliton_positions = x_bar(peak_locations);
28
29 % Estimate soliton widths (assuming Gaussian shape)
30 soliton_widths = ones(num_solitons, 1) * sg; % Assuming same width for simplicity
31
32 % Replace solitons with Gaussian pulses
33 new_initial_condition = zeros(length(final_waveform),1);
34 for i = 1:num_solitons
35     new_initial_condition = new_initial_condition + ...
36         soliton_amplitudes(i) * exp(-(x_bar - soliton_positions(i)).^2 / (2 *
       soliton_widths(i).^2));
37 end
38
39
40
41 % Time-reverse the solitons
42 u_reverse = (usol(end,:));
43 u_reverse =new_initial_condition'; %change to gaussian
44
45
46
47
48 % % Store the initial condition for the reverse process
49 usol_reverse = zeros(stored_steps, N);
50 time_values_reverse = zeros(stored_steps, 1);
51 usol_reverse(1, :) = u_reverse;
52 time_values_reverse(1) = 0;
53
54 omega_k = c * k - gamma * k.^3; % Dispersion relation
55 alpha_k = alpha0 * (omega_k / omega0).^2; % Attenuation coefficient
```

```
56
57
58 % Time-stepping loop for the reverse process
59 for n = 1:Nt
60     u_hat = fft(u_reverse); % Compute the FFT of the current solution
61     u_hat = u_hat .* exp(-1i * k.^3 * dt); % Reverse the linear part
62
63     % Solve the non-linear part
64     u_hat = u_hat + dt * (0.5 * 1i * k .* fft(real(ifft(u_hat)).^2)); % Reverse the
       non-linear part
65     u_reverse = real(ifft(u_hat));
66
67
68     % Store the solution at specified intervals
69     if mod(n, store_interval) == 0
70         idx = n/store_interval + 1;
71         usol_reverse(idx, :) = u_reverse;
72         time_values_reverse(idx) = n * dt;
73     end
74 end
75
76 % Plot results
77 figure;
78 subplot(2, 1, 1);
79 plot(x_bar*xc, usol(1,:)*uc, '--r', 'DisplayName', 'Initial Pulse', 'LineWidth',1.5);
80 hold on;
81 plot(x_bar*xc, usol(end, :)*uc, 'b', 'DisplayName', 'Final State (Forward)','
       LineWidth',1.5);
82 xlabel('x [nm]');
83 ylabel('u');
84 title('Forward Process');
85 legend;
86 grid on;
87
88 subplot(2, 1, 2);
89 plot(x_bar*xc, fliplr(usol_reverse(end, :)*uc), 'b', 'DisplayName', ' New initial
       Pulse','LineWidth',1.5); hold on;
90 plot(x_bar*xc, fliplr(usol_reverse(1, :)*uc), 'm', 'DisplayName', ' Recovered state',
       'LineWidth',1.5);
91 xlabel('x [nm]');
92 ylabel('u');
93 title('Reverse Process');
94 legend;
95 grid on;
```

Listing 3: Compression and Time-reversal simulation