GASE: a high performance solver for the Generalized Nonlinear Schrödinger equation based on heterogeneous computing

Nuno Miguel Azevedo Silva
Mestrado em Física
Departamento de Física e Astronomia
2013

Orientador
Professor Ariel Guerreiro

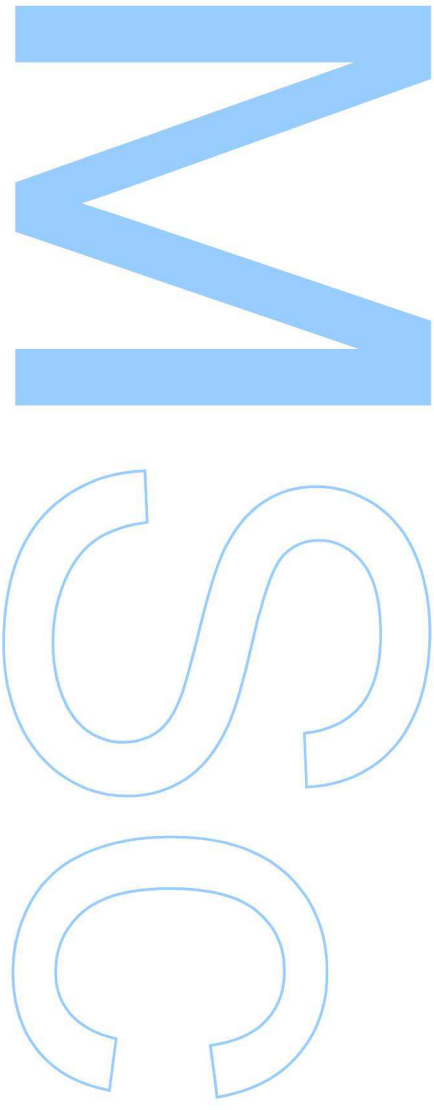
U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, _____/_____/_____



Agradecimentos

Chegando ao fim de um ciclo de um ano de trabalho, não posso deixar de aproveitar a ocasião para agradecer a todas as pessoas que contribuíram para esta dissertação. Primeiro, agradeço ao meu orientador, Professor Ariel Guerreiro, por toda a paciência e ajuda prestada tanto durante os trabalhos da dissertação como na sua elaboração. Diria mesmo que, sem o seu auxílio, esta dissertação só estaria completa daqui a dois meses. Todo o trabalho não teria sido possível também sem o apoio inestimável do Paulo, que me ensinou os primeiros passos da computação em placas gráficas. A ele, a todos os outros colegas do INESC e pessoas que contribuíram directamente para este trabalho, agradeço por todo o apoio prestado.

Finalmente, agradeço à minha família, aos meus pais e irmão, à Rute e a todos os meus amigos, especialmente os que me acompanharam nos anos em que passei na FCUP. Sem todos eles, teria talvez acabado esta dissertação há dois meses atrás. Mas tenho a certeza que sem eles, não teria sido a experiência tão agradável e reconfortante que foi.

Abstract

This dissertation presents a research work in the field of computational physics, namely the development, testing and benchmark of a high performance solver of the Generalized Nonlinear Schrödinger equation that can address problems with high dimensionality and complex geometries, based on massive parallel computing using graphical processing units.

The Generalized Nonlinear Schrödinger equation is an active topic of research and has attracted the attention of many researchers during the last decades. A major difficulty in this field is that the model is usually non-integrable and even perturbation methods are not valid in multidimensional and complex geometries. Instead, most of the research is done using numerical simulations to address this class of problems. However, in general, these have a high computational cost and can only be performed efficiently in costly computer clusters or supercomputers. In recent years, the field of computer sciences came up with a new computation concept called heterogeneous computing, that allows to use all the computing resources of a machine in an integrated way to do massive computing. This new computing paradigm is in the core of this dissertation as the enabling technology that supports the development of our solver. This dissertation begins with a general overview of the state-of-the-art in both Generalized Nonlinear Schrödinger equation and graphical processing units computing. The fundamental aspects of this equation are analyzed, presenting the most relevant analytical and numerical methods, which includes the overview of the Split-step Fourier method, the algorithm chosen for the development of the solver.

The algorithm was implemented in the *CUDA* language, which runs on *NVIDIA* graphical processing units, and the extensive tests performed on the solver revealed that it outperforms the serial version of the solver. It is shown that the solver developed turns the graphical processing units into low-budget solutions for high performance computation, many times faster than state-of-the-art central processing units, and with performances that can compete with expensive supercomputers. Also, two physical problems described by the Generalized Nonlinear Schrödinger equation are considered in

the last part of the dissertation and even though they have not been fully explored, given the limited duration of this dissertation project. The preliminary studies show some interesting results and illustrate the potential and versatility of the solver developed. Finally, some conclusions and future directions of research are discussed.

This dissertation hopes to contribute to the field of computational physics by showing how the new computing paradigms (such as heterogeneous computing) can be used to improve the performance of existing algorithms and methods and, through that, provide a tool of research to study more complex and demanding problems.

Resumo

Esta dissertação apresenta o desenvolvimento de um trabalho na área de física computacional, mais precisamente, o desenvolvimento, teste e análise de performance de um *solver* para a equação Não Linear de Schrödinger Generalizada, que seja capaz de resolver problemas de alta dimensionalidade e em geometrias complexas, baseado na computação paralela usando placas gráficas.

A equação Não Linear de Schrödinger Generalizada é um tópico bastante activo e tem atraído a atenção de muitos investigadores nas últimas décadas. Uma das maiores dificuldades na investigação destes sistemas é que geralmente o modelo é não-integrável e mesmo os métodos perturbativos são incapazes de oferecer soluções para estes problemas. Assim, a grande parte da investigação passa pela simulação numérica desta classe de problemas. No entanto, estes têm em geral um elevado custo computacional e só em clusters de computadores ou supercomputadores a sua simulação é eficiente. Nos últimos anos, surgiu um novo conceito na ciência de computadores denominado computação heterogénea, que permite a utilização integrada de todos os recursos de um computador para o cálculo de grandes tarefas numéricas. Este paradigma computacional constituí o núcleo desta dissertação ao ser a tecnologia que permite o desenvolvimento do *solver*. Esta dissertação começa com uma síntese geral do estado da arte da investigação tanto do caso da equação Não Linear de Schrödinger Generalizada como da computação em placas gráficas. Os aspectos fundamentais da equação são analisados e os métodos analíticos e numéricos mais importantes são apresentados, com especial atenção ao *Split-Step Fourier Method* , o algoritmo que escolhemos para utilizar no *solver*.

O algoritmo foi implementado na linguagem *CUDA*, que corre em placas gráficas da *NVIDIA*, e uma grande variedade de testes foram executados ao *solver,* revelando performances muito acima das obtidas para versões tradicionais com base no processamento em série. É mostrado também que o *solver* desenvolvido torna um computador com uma simples placa gráfica numa solução de baixo custo para obtenção de performances elevadas, muito mais rápidas que os habituais processadores em série e com resultados capazes de competir com supercomputadores muito mais caros. Além desta análise, dois

problemas descritos pela equação Não Linear de Schrödinger Generalizada são considerados na última parte da dissertação, mesmo não tendo sido completamente explorados devido à duração limitada deste projeto de dissertação. Os estudos preliminares mostram alguns resultados interessantes e acima de tudo ilustram o potencial e versatilidade do *solver* desenvolvido.

Deste trabalho espera-se sair uma contribuição para o ramo da física computacional ao demostrar como novos paradigmas computacionais (como a computação heterogénea) podem ser utilizados para melhorar a performance dos algoritmos e métodos existentes e, assim, oferecer uma ferramenta computacional capaz de resolver problemas computacionalmente custosos.

Contents

1 Introduction **1**

1.1 A soliton story . . . 2
1.2 1+1=3 and the Nonlinear Schrödinger Equation . . . 4
1.3 Optical solitons . . . 6
1.4 Analytical, variational and numerical methods . . . 10
1.5 GPU computing and prospects for the NLSE . . . 11
1.6 GASE - GPU Accelerated Soliton Explorer . . . 15
1.7 Outline and structure of the dissertation . . . 15

2 Nonlinear Schrödinger equation in a nutshell **17**

2.1 NLSE and analytical solutions . . . 17
2.2 GNLSE . . . 20
2.3 Noether's theorem and conservation laws in the GNLSE . . . 21
2.4 Variational methods . . . 23
2.4.1 Trial functions and solutions . . . 23
2.4.2 Perturbation methods . . . 24
2.4.3 Effective particle approach . . . 26
2.5 Numerical methods . . . 27
2.5.1 Explicit and implicit finite differences methods . . . 29
2.5.2 Pseudo-spectral methods and the SSFM . . . 31
2.5.3 Boundary conditions for the SSFM . . . 34
2.6 Concluding remarks . . . 36

3 Implementation of the GPU-based GNLSE solver **37**

3.1 Simple problem, high computational time . . . 37
3.2 How to plow a field? . . . 40
3.3 Gaming vs Scientific precision . . . 42

3.4 Some memories must be kept closer than others: memory considerations for GPU computing . . . 44
3.5 Compatibility issue . . . 45
3.6 Implementation of the GNLSE solver . . . 45
3.6.1 Outline of the code . . . 45
3.6.2 Integration step routine . . . 46
3.6.3 Code features . . . 49
3.7 Concluding remarks . . . 49

4 Benchmark of GASE **55**
4.1 Validation of the method . . . 55
4.2 Benchmark of the method: SSFM *versus* CN . . . 57
4.3 Benchmark of the code: GPU *versus* CPU . . . 58
4.3.1 (1+1)-d speedup results . . . 59
4.3.2 (2+1)-d speedup results . . . 61
4.3.3 (2+1+1)-d speedup results . . . 63
4.4 Concluding remarks . . . 65

5 Case study 1: Lightons: phonons with Light **67**
5.1 Motivation . . . 67
5.2 Physical model . . . 68
5.3 Numerical study of the normal modes in a soliton chain . . . 71
5.3.1 Other types of solutions . . . 75
5.3.2 Soliton chains in (2+1)-d . . . 76
5.4 Concluding remarks . . . 77

6 Case study 2: Soliton-soliton scattering in (2+1)-d **79**
6.1 Motivation . . . 79
6.2 Physical model . . . 80
6.3 Scattering of colliding (2+1)-dimensional spatial solitons . . . 81
6.4 Hard soliton scattering . . . 82
6.5 Soft soliton scattering . . . 84
6.6 Concluding remarks . . . 87

7 Conclusions **91**
7.1 Future work . . . 93
7.2 Publications . . . 93

List of Figures

1.1 a) On 1995 during a conference on nonlinear waves at Heriot-Watt University, the attending scientists recreated the first reported observation of a solitary wave, as part of a ceremony to honor Russell and name the new aqueduct with his name. Image from [1] b) Figure taken from the original report of the FPU problem [22], showing a simulation performed in the *MANIAC*. Fermi, Pasta and Ulam initialized the system with all energy in the lowest normal mode and observed the evolution. The energy is transferred into several modes and, after some time, the system returns to a state similar to the initial condition, with energy back to the first mode, unlike the expected thermalization. 3

1.2 Examples of soliton interaction (numerical and experimental) showing the particle behavior of solitons. a) Numerical simulation of a collision between two out-of-phase optical spatial solitons in a cubic media. b) Experimental results showing the collision of two solitons in a photorefractive medium [46]. Frame 1 shows the two solitons when they are launched one at a time. Frame 2 shows the output result when no nonlinearity exists and frame 3 is the output with nonlinear refractive index and when two solitons are launched at the same time. Frames 4,5 and 6 show different types of interactions depending on the initial phase difference between the two solitons: 4 shows fusion of two solitons caused by the attractive potential for in-phase solitons; frame 5 shows the result of the interaction between two solitons with phase difference $\pi/2$; frame 6 represents the out-of-phase case, where the solitons repel each other. 7

1.3 In (2+1)-d, high power supergaussian states constitute liquid light states, i.e., solitons showing liquid behavior. In this simulation is shown the coalescence of two high power solitons (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation) 9

1.4 a) GPUs have long surpassed peak performances of CPUs, which triggered the development of GPU computing. b) Also, they attain this peak performances without prohibitive power consumptions, characteristic of frequency scaling of the CPUs, using the parallel computing paradigm. . 12

2.1 Depending on the positive or negative sign of the nonlinearity, solitons can be either bright - figure a) - or dark - figure b) - respectively. Both were represented in arbitrary units. 19

2.2 Supergaussian shapes of: a) a low power soliton with $m = 1$; b) a high power soliton with $m = 1.9$. The competition between focusing cubic and defocusing quintic nonlinearities made high power solitons to spread, acquiring a flat-top form. Typically, the high power supergaussians exhibit liquid-like behavior, while low power exhibit other behaviors. 24

2.3 Visual scheme describing the split-step algorithm for evolving an initial field $\psi(z)$, described in equations (2.43-2.45). 29

2.4 Visual scheme of the simulation box, describing the concept of boundary layer. The blue interior of the box is where the field evolves. 36

3.1 a) Image description of the problem of a supergaussian state colliding with a small hole of linear material. b) Representation of the circle for various mesh sizes. 38

3.2 Conceptual comparison between two different computing paradigms, the high frequency and the high throughput computing paradigm. 40

3.3 Succinct description of the code structure. The code is divided in three parts: the first initializes the data and the simulation box, the second is the integration routine and the third is the post-simulation analysis of the data. 47

3.4 Integration routines for both versions of the solver. Figure a) shows the structure of the evolution routine for GASE, where it is possible to see the additional memory transfer needed but also the parallel structure of computations running both in CPU and GPU. Figure b) describes the integration procedure for the CPU version of the solver. 50

3.5 Sequence a)-d) shows a collision of a supergaussian state with velocity $\mu = 0.3$ with a hole of radius 2.4. The light state emerges as two smaller and low intensity beams after the scattering. (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation) 51

3.6 Sequence a)-d) shows a collision of a supergaussian state with velocity $\mu = 0.5$ with a hole of radius 2.4. The light state emerges as two smaller and low intensity beams after the scattering, with a different angle than the situation with velocity $\mu = 0.3$. (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation) 52

3.7 Sequence a)-d) shows a collision of a supergaussian state with velocity $\mu = 0.15$ with a hole of radius 2.4. The light state collides and is reflected by the hole. (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation) 52

3.8 Sequence a)-d) shows a collision of a supergaussian state with velocity $\mu = 0.3$ with a hole of radius 2.0. The light state after an intermediate division in two light states collapses again in one high power state. (*Pdf version only* - click twice on sub-figure a) for a clip of the simulation, where can also be seen that the state became trapped between two consecutive holes, that we simulate in the same box using periodic conditions) . . . 53

3.9 Sequence a)-d) shows a collision of a supergaussian state with velocity $\mu = 0.3$ with a hole of radius 1.0. The light state emerges almost as if not been scattered. (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation) 53

4.1 Initial conditions (a)) and analytical final state at $z_{final} = 100$ (b)) of the high power - dashed line - and low power - solid line - solitons. 56

4.2 Single (top) and Double (bottom) precision benchmarks for simulations of the (1+1)-d NLSE, with the results for computational time per step (left) and the corresponding speedup in comparison with the CPU version of the code (right). 60

4.3 Double (top) and single (bottom) precision benchmarks for simulations of the (2+1)-d GNLSE for a cubic-quintic media, with the results for computational time per step (left) and the corresponding speedup in comparison with the CPU version of the code (right). 62

4.4 Single precision benchmarks for simulations of the (2+1)-d GNLSE for a cubic-quintic media with absorbing boundaries, with the results for computational time per step (left) and the corresponding speedup in comparison with the CPU version of the code (right). 63

4.5 Double (top) and single (bottom) precision benchmarks for simulations of the (2+1+1)-d GNLSE for a cubic-quintic media, with the results for computational time per step (left) and the corresponding speedup in comparison with the CPU version of the code (right). 64

5.1 Evolution of the soliton chain for $n = 3$ (a)) and $n = 15$ (b)). Figures a) and b) show the intensity profile evolution where we can identify the phonon-like oscillations. 72

5.2 Evolution of the displacement of the second soliton of the chain from the right, for $n = 3$ (a)) and $n = 15$ (c)), comparing in detail the displacement obtained from the simulations (solid line) with the prediction of TLE (dashed line). Figures b) and d) display the Fourier transform of the displacements wave (numerical results with solid line, TLE with dashed line), where we can identify the generation of new frequencies for lower n. 73

5.3 Error analysis of the displacement wave for a soliton between the numerical simulation and TLE model. Solid line is the absolute value of displacement error, $|u_2 - u_2^{TLE}|$, dashed and dash-point line is the absolute value of error for phase difference between consecutive solitons and the prediction $|\Psi_{2,3} - \pi|$, $|\Psi_{1,2} - \pi|$ respectively. 74

5.4 Dispersion relation computed for a chain of $N = 40$ solitons (circular markers) and comparison with the TLE model (solid line). 74

5.5 Collapse of the standing wave for displacements with $n = 15$. The feedback process increases the amplitude of the oscillation and when it surpasses the mean separation between solitons, solitons start to collide and the phonon-like behavior is lost. 75

5.6 Evolution of the collision of a soliton with velocity $\mu = 0.2$ with a soliton chain with 4 solitons initially separated by $\Delta = 10$. The trapping potential allows to obtain results that resemble the Newton's cradle. . . 76

5.7 Evolution of an 1-dimensional chain of 2-dimensional spatial solitons with $\Delta = 20$. 78

6.1 Graphical description of the problem analyzed. Here, impact parameter b was exaggerated for better comprehension. Simulation box has limits $[0, 80] \times [0, 80]$ and a mesh of $N = 2^{10} \times 2^{10}$ points was used. 81

6.2 Typical numerical results for the evolution of two colliding solitons. Sequence a)-e) shows a collision between two out-of-phase solitons with $k_1 = 0.2$ and $b = 0$, sequence f)-j) displays the results for $b = 4$ and k)-o) for $b = 9$. Sequence p)-t) displays the coalescence of two colliding in-phase solitons with $b = 0$ and $k = 0.3$. Sequence u)-y) shows the results for $b = 5$ and $k = 0.3$. Sequence z)-dd) displays the destruction of two colliding in-phase solitons with $b = 0$ and $k = 0.8$. 83
6.3 Computational results for the relation between the scattering angle and the impact parameter, $\theta(b)$. This figure displays the results for colliding out-of-phase solitons for $k = 0.05$ (full line with markers), $k = 0.2$ (dashed line with markers) and $k = 0.3$ (pointed line with markers). The full line without markers shows the hard-sphere limit for a sphere with radius of 6. 84
6.4 Computational results for the relation between the scattering angle and the impact parameter, $\theta(b)$. This figure shows the results for colliding in-phase solitons with $k = 0.3$ (circles), $k = 0.25$ (crosses), $k = 0.23$ (triangles) and $k = 0.2$ (squares). Different behaviors are obtained depending on both the collision velocity and the impact parameter. Shaded region is the zone of coalescence for $k = 0.3$, where solitons reveal liquid-like behavior. 85
6.5 Sequence of data values for two coupled fields showing the collision of two in-phase solitons for the situation A in the figure (6.4). (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation) 86
6.6 Sequence of data values for two coupled fields showing the collision of two in-phase solitons for the situation B in the figure (6.4). (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation) 87
6.7 Sequence of data values for two coupled fields showing the collision of two in-phase solitons for the situation C in the figure (6.4). (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation) 88
6.8 Sequence of data values for two coupled fields showing the collision of two in-phase solitons for the situation D in the figure (6.4). (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation) 89

List of Tables

3.1 Comparison between different top-of-the-line GPUs of the NVIDIA consumer line Geforce. A model from the Fermi line of a professional computing dedicated GPUs is also presented. An increasing computational power is notorious over the years, as well the increase of chip memory and memory bandwidth. It is well patented that evolution of GPUs will reach another level in the next few years, with the new high performance Geforce Titan setting the pace. 43

4.1 Error analysis for the simulations with fixed integration step $h = 0.01$ and variable number of points, which introduces a variable discretization Δx. 57

4.2 Error analysis for the simulations with fixed number of points $N = 2^8$ and variable integration step h. 57

4.3 Accuracy and performance comparison between GASE solver, based on SSFM, and the CN solver. It can be easily seen that GASE outperforms in every aspect the CN method. 58

4.4 Specifications of both the two GPUs and the CPU used during the benchmarks. 59

4.5 A collection of results for simulation times and speedup of the solver for the (1+1)-d NLSE using double precision, for both GASE (running in the desktop GPU) and the CPU-based version of the solver. 61

4.6 A collection of results for simulation times and speedup of the solver for the (2+1)-d GNLSE for a cubic-quintic media, using double precision, for both GASE (running in the desktop GPU) and the CPU-based version of the solver. 63

4.7 A collection of results for simulation times and speedup of the solver for the (2+1+1)-d GNLSE for a cubic-quintic media, using double precision, for both GASE (running in the desktop GPU) and the CPU-based version of the solver. 65

List of abbreviations

GNLSE - Generalized Nonlinear Schrödinger Equation

GPU - Graphical Processing Unit

GPGPU - General Purpose GPU applications

CPU - Central Processing Unit

KdV - Korteweg-de-Vries

FPU - Fermi-Pasta-Ulam

CW - Continuous Wave

BEC - Bose-Einstein Condensate

IST - Inverse Scattering Transform

FD - Finite Differences

FFT - Finite Fourier Transform

PS - Pseudo-Spectral

SSFM - Split-Step Fourier Method

1 Introduction

The central problem of this dissertation is the development of a high performance solver of the Generalized Nonlinear Schrödinger Equation (GNLSE) based on heterogeneous programming using graphical processing units (GPU), that is able to address physical systems with high dimensionality (more than one spatial dimension) and complex geometries or structures with reasonable simulation times and that can run using low cost desktop computers. This constitutes a paradigm in computational physics since until few years ago this type of problems could only be addressed using costly state-of-the-art supercomputers and computer clusters, generally inaccessible to most scientists. GPU computing is bringing a revolution into scientific computing by allowing to do supercomputing by using several hundreds of processing units inside a desktop computers as a massive cluster. However many of the computational models and simulation codes previously developed cannot be straightforwardly adapted to heterogeneous computing, given its distinct computing architecture and the specific programming tools required.

Although the focus of this dissertation is on the development of the GNLSE solver, both as a proof of concept and as a simulation tool for future research, in the following chapters we will also present several case studies where we do some earlier exploration of soliton dynamics, mainly as a test and illustration of the potential of this code. These examples were not fully explored in terms of a complete scientific analysis since that goes beyond the scope of this dissertation and would require a longer research time.

The existence and propagation of solitons in nonlinear media attracted a substantial research interest for the past 50 years. While thousands of papers were published in this field and particularly in the study of the Nonlinear Schrödinger Equation (NLSE), new and more complex systems provide substance for prospective explorations. As current investigations focus on multidimensional solitons and spatial distribution of nonlinearity, described by non-integrable models, numerical simulations become mandatory. In a normal computer, simulation times for such systems are usually prohibitive and researchers that do not have access to a supercomputer are either limited in the research to smaller and simple systems. In this context, the development of new and high performance

computational tools for the study of the propagation of solitons is a subject of great importance for state-of-the-art problems.

Heterogeneous computing is one of the newest and fascinating trends in modern physics. It consists on the use of the central processing unit (CPU) in addition to other specialized hardware, usually the graphical processing units to get faster numerical simulations. Particularly, the use of the GPU for general purpose applications (GPGPU) created a buzz in recent years, as researchers from many areas attained overwhelming performances - up to 100 times faster than state-of-the-art CPUs. This potential, that is at the same level of the 1999-2000 timeframe best supercomputers, suggests that GPU based solvers of the NLSE could be the solution for simulating the cutting-edge demanding problems using only inexpensive personal machines, a hypothesis that is the motivation behind this dissertation.

1.1 A soliton story

The story of solitons is prolific in coincidences and fortuitous events. It all started in 1834 with a curious scottish engineer, John Scott Russell, hired to investigate how to improve the efficiency of boat designs at the Union Canal, near Edinburgh. In a fortuitous accident, a rope pulling a boat broke and Russell observed the formation of a wave that he described accurately in his report [69] as *"a large solitary elevation, a rounded, smooth and well-defined heap of water, which continued its course along the channel apparently without change of form or diminution of speed"*. Most probably, this was not the first time that a solitary wave was observed, but Russell was the first to report it. Believing that the discovery was important he did extensive experiments in a scale model constructed at his backyard. In 1895, Dutch physicists Diederick Korteweg and Gustav de Vries derived an equation [48] to match the observations reported by Russell. This partial differential equation, now called Korteweg-de Vries equation (KdV), contained both linear and nonlinear terms and although they were unable at that time to produce general solutions, they found a solitary-wave solution that resembled Russell's wave. Strangely, as it happened to Russell, their work fell into obscurity and was overlooked by mathematicians, physicists and engineers for more than 50 years.

The story continues in the early 1950s, at Los Alamos Scientific Laboratory when Enrico Fermi, John Pasta and Stanislaw Ulam used one of the earliest digital computers - the *MANIAC* (MAthematical Numerical Integrator And Computer) - to investigate the simple nonlinear system of a one dimensional chain of masses connected by springs with

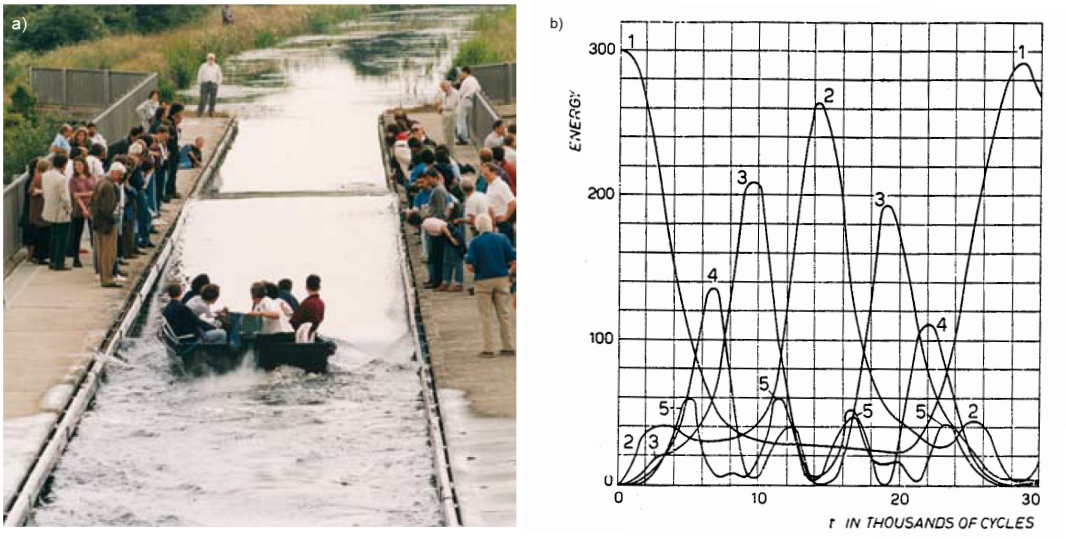


Figure 1.1: a) On 1995 during a conference on nonlinear waves at Heriot-Watt University, the attending scientists recreated the first reported observation of a solitary wave, as part of a ceremony to honor Russell and name the new aqueduct with his name. Image from [1] b) Figure taken from the original report of the FPU problem [22], showing a simulation performed in the *MANIAC*. Fermi, Pasta and Ulam initialized the system with all energy in the lowest normal mode and observed the evolution. The energy is transferred into several modes and, after some time, the system returns to a state similar to the initial condition, with energy back to the first mode, unlike the expected thermalization.

linear and small nonlinear restoring forces [22]. Exciting one normal mode of the linear system, they believed that the nonlinearity term would excite different modes and then at some point in time the system would "thermalize", i. e., the energy would be equally distributed among all the possible normal modes. Nevertheless, the results were unexpected: while it is true that after some periods the energy was shared between several modes, the prolongation of the simulation revealed a near return to the initial mode, as 97% of the energy focused again in the initial mode. This unexplained recurrence, later known as Fermi-Pasta-Ulam problem (FPU), did not convinced everyone, as some thought the system did not run enough time [8].

One of the first attempts to solve this puzzle was a phenomenological explanation suggested in 1965 by Zabusky and Kruskal [86]. Taking the FPU in the continuum limit, they found that the system was governed by a KdV equation, an equation analytically intractable at that time. Solving the equation using numerical simulations, they observed a breakdown of the initial periodic condition into a train of solitary waves.

Regardless of the initial condition tested, the recurrence of the initial wave suggested by the results of FPU was not observed during the simulations. Instead, they detected that the solitary waves started to move and collide. They also observed a quasi-recurrence to an intermediate state after a near chaotic behavior.

It is a common mistake to attribute the solution of the FPU problem to this study of Zabusky and Kruskal [46]. Despite obtaining a recurrence, their belief that the discovery provided a phenomenological explanation was poorly grounded, as they did not obtain any return to initial mode. A more concise explanation was only reached recently (1997) by Casetti et al. [13] with the discovery of two regimes for the dynamics of the FPU that depends on the energy per oscillator and on the total number of masses. For low oscillator energy and number of masses, dynamics are regular to weakly chaotic, like the FPU results. For higher oscillator energies and number of masses, dynamics are completely chaotic. Indeed, we should be thankful that Fermi and his co-workers did not simulate bigger systems nor used stronger nonlinearities, because if they did, they would have found an equipartition of the energy, and then maybe much of the understanding of nonlinearities that arose from their studies might not have existed.

In spite of not succeeding in explaining the FPU problem properly, Zabusky and Kruskal's publication still became very famous. They observed the survival of solitary waves after collisions - a behavior typical of a particle - and led them to coin one of the most successful terms in nonlinear science: the soliton.

1.2 1+1=3 and the Nonlinear Schrödinger Equation

When one starts to study physics it is common to think that nonlinear is synonym of anomalous, pertaining to something that diverges from "well behaved" physics. However, nonlinear systems are far more common in the real world than the linear ones. The study of nonlinear systems is the subject of nonlinear science, and the main idea is that *"the whole is more than a sum of its parts"*, or in physicist language, the superposition principle is not valid.

Another feature that boosted the development of the nonlinear science was its universality, not only in terms of being present in almost every phenomenon of the Universe, but also because it can be found in many fields of science. Indeed, models explaining phenomena like chaos or coherent structures can be used to describe a wide panoply of problems, not only in theoretical physics but also in mathematics, biology, neurosciences, sociology and more [71].

The Nonlinear Schrödinger Equation is one of this models. The NLSE was introduced first in 1964 by Chiao et al. [14] to describe the propagation and self-trapping of continuous wave light beams (CW) incident in a nonlinear Kerr media, both in one and two spatial dimensions. Soon, the first hints of the universality started to appear, when Hasewaga and Tappert [30] suggested a NLSE equation to describe the propagation of pulsed beams in optical fibers. Nowadays, it is normally accepted that NLSE

$$i\frac{\partial\psi}{\partial z} + \frac{1}{2}\nabla_\perp^2\psi + |\psi|^2\psi = 0 \tag{1.1}$$

and the more global GNLSE

$$i\frac{\partial\psi}{\partial z} + \frac{1}{2}\nabla_\perp^2\psi + F(|\psi|^2)\psi = 0 \tag{1.2}$$

are universal equations that describe the evolution of wave envelope in a weakly nonlinear medium. This equation arises in many and distinct areas, the most common being:

- Nonlinear optics: to describe the propagation of CW and pulsed beams in nonlinear medium [14, 46, 30];
- Bose Einstein Condensates (BEC): to describe the mean-field dynamics of the BEC [43, 27] where it is commonly known as Gross Pitaevski equation;
- Fluid dynamics:to describe for example the instability of Poiseuille flow [76], deep water waves [85] and Couette-Taylor flow [19]. It is commonly known as Complex Ginzburg-Landau equation;
- Plasma physics: to describe Langmuir waves [6];
- Protein chemistry: to model the vibrations of molecular chains [16].

This variety of applications reinforced the interest and the search for solutions for the NLSE.

In the seminal paper, Chiao and his co-workers said at some point that NLSE "appears to have no simple analytical solution" [14] leading them to search for numerical solutions. Although the solution had a bell shape similar to a solitary wave, the relationship was not explicitly noticed.

Meanwhile, the paper of Zabusky and Kruskal on solitons [86] led to the development of the Inverse Scattering Transform (IST) method in 1967 [26]. This elegant mathematical tool is an analog of the Fourier transform for nonlinear systems, such as the initial value

problem of the KdV equation. With IST, it is possible to obtain analytically the soliton solutions of KdV equation, previously predicted by Korteweg and Vries via ansatz. The relationship between NLSE and solitons was uncovered in 1972 by Zakharov and Shabat. In their publication [87] they showed that IST was applicable to the NLSE, explaining then the self-focusing and self-phase modulation of intense laser beams in nonlinear Kerr media by the means of stable soliton solutions. Together with the universality of the model, this achievement constituted a strong stimulus that pushed the research during the earlier years.

1.3 Optical solitons

During the past 50 years, thousands of papers devoted to the subject of NLSE were published and a complete review would be too long to be included in this dissertation. We will only summarize some of the most important results, focusing mainly in the field of nonlinear optics.

Solitons in optical media are usually referred to as optical solitons. Depending if the nonlinearity is self-focusing or defocusing they can either be bright or dark solitons, respectively [46]. This dissertation will focus on the most common: bright solitons. Optical solitons are usually classified into three categories:

- Spatial solitons: the confinement is in spatial dimensions, and the phenomenon is the result of the equilibrium between nonlinear self-focusing and diffraction of the beam. Predicted as a solution of the original NLSE derivation [14] in 1972, stable (1+1)-dimensional[1] solitons were observed for the first time in planar waveguides in 1988 by Maneuf et al. [54]. On the other hand, (2+1)-d solitons were only observed in a photorefractive crystal in 1993 by Duree et al. [21];

- Temporal solitons: the confinement occurs along the temporal dimension and this type of soliton is related with the compensation of dispersion by the self-phase modulation induced by the Kerr effect. They were observed experimentally in 1980 by Mollenauer et al.[61];

- Spatiotemporal solitons: the confinement occurs both in space and in time. Also called light bullets, they differ from the previous cases as they involve both the

[1] The notation (D+1)-dimensional or (D+1)-d solitons refers to solitons confined in D transverse spatial dimensions, with 1 standing for the propagation spatial dimension of the system. (D+1+1)-d are solitons confined in D transverse spatial dimensions and 1 temporal dimension. The last 1 is associated with propagation spatial dimension.

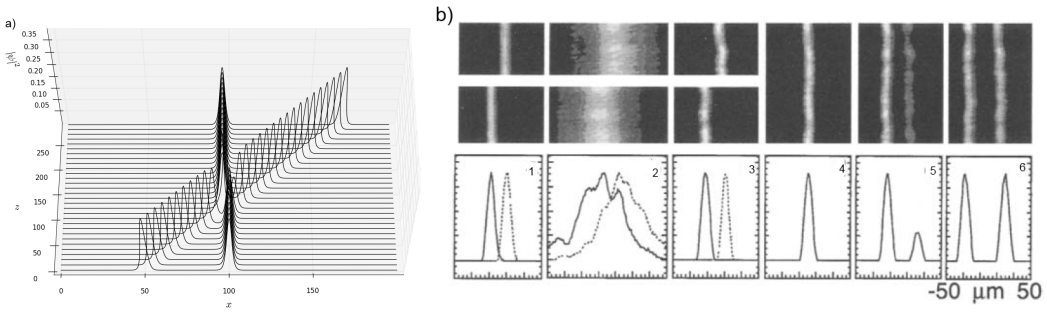


Figure 1.2: Examples of soliton interaction (numerical and experimental) showing the particle behavior of solitons. a) Numerical simulation of a collision between two out-of-phase optical spatial solitons in a cubic media. b) Experimental results showing the collision of two solitons in a photorefractive medium [46]. Frame 1 shows the two solitons when they are launched one at a time. Frame 2 shows the output result when no nonlinearity exists and frame 3 is the output with nonlinear refractive index and when two solitons are launched at the same time. Frames 4,5 and 6 show different types of interactions depending on the initial phase difference between the two solitons: 4 shows fusion of two solitons caused by the attractive potential for in-phase solitons; frame 5 shows the result of the interaction between two solitons with phase difference $\pi/2$; frame 6 represents the out-of-phase case, where the solitons repel each other.

spatial and time confinement of localized pulsed beams. Stable (2+1+1)-d solitons were predicted in quadratic media in 1981 [34] but an experimental realization still has not been accomplished.

There are two major fields of interest in the research of the existence and the propagation of solitons: solitons in nonlinear bulk medium and solitons in tailor made materials with a spatial distribution of nonlinearities, also called optical lattices.

For solitons in bulk nonlinear medium, a subject of great interest is the unique properties of interaction between solitons. For (1+1)-d solitons, there are many analytical studies of collision of solitons in Kerr media [5, 46], predicting elastic collision between solitons with exchange of momenta and preservation of shape, effects observed in various experiments since 1990 [68]. The case of soliton collision in media with higher order nonlinearities also revealed interesting properties, as it features inelastic collisions [72], energy exchange mechanisms [46] and fractal structure of the dependence of final velocity with the variation of phase difference between the two solitons [20]. In fact, the dependence of the interaction between solitons with their phase difference triggered the development of theoretical potential of interaction between two solitons [51]. The

interaction can be either coherent or incoherent whether it depends or not on the phase difference. Coherent interaction [51, 46] could lead to the mutual attraction for in-phase solitons or repulsion for out-of-phase ones. Incoherent interaction is obtained considering systems that support vector solitons [79, 49, 3], using two orthogonally polarized beams [56] or even a system of solitons so that their relative phase difference varies faster than the response of the medium [73, 74]. In such systems, incoherent interacting solitons can only attract each other[46].

Throughout the last decades, (1+1)-d solitons have been studied extensively and there have been many theoretical tools developed with great success for their analysis. It may seem that same results can be obtained straightforwardly for (2+1)-d solitons. It turns out that this intuition could not be wronger. However, contrary to what happens in the one dimensional case, it was shown [46, 87] that a (D+1)-dimensional GNLSE with focusing power-law nonlinearity $|\psi|^{2q}\psi$ blows up under the condition $qD \geq 2$. Thus, Kerr nonlinearity is enough to create a singularity for $D = 2$, which implies that (2+1)-d solitons are only realizable with different nonlinearities, such as quadratic media [53] or saturable media. Saturable media are obtained either exploiting the photorefractive effect [37] or higher order nonlinearities like the cubic-quintic media, where cubic is focusing and quintic a defocusing nonlinearity [18].

Like the one dimensional case, the interaction between solitons concentrates most of the research to the present days. Malomed et al. [50, 52] developed effective potential theories for coherent and incoherent interaction between two solitons, that ultimately supported discoveries like the conservation of angular momentum [17], the stable and unstable spiraling of solitons [10, 74, 46] and the rotation of N-soliton clusters in a ring like geometry [46]. Also, ring shaped solitons with angular momentum were predicted for saturable [46] and cubic-quintic media [60]. More recently, Michinel and his co-workers published a series of investigations on the dynamics of solitons in cubic-quintic media. They revealed the coalescence of two solitons [67], the splashing and the creation of small droplets when hitting a surface [58], the existence of an analogous to the Young-Laplace surface tension [59] and the capillarity phenomena [64]. These results suggested a liquid-like behavior of beams of light, that they called liquid light.

The generation of bullets of light in (D+1+1)-d systems is far more tricky, as the confinement is not only in space but also in time. In the NLSE both the spatial and temporal dimensions are treated alike, which suggests a similar mechanism of confinement of light along these dimensions. However the processes of diffraction and dispersion that affect analogously the beam along the spatial and temporal dimensions, respectively, occur

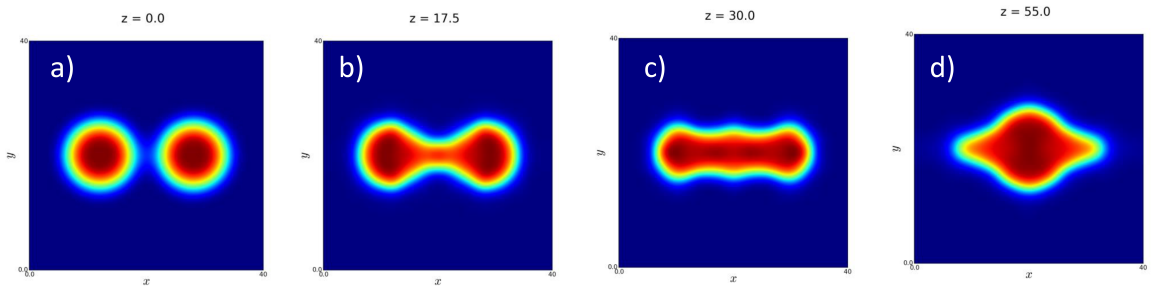


Figure 1.3: In (2+1)-d, high power supergaussian states constitute liquid light states, i.e., solitons showing liquid behavior. In this simulation is shown the coalescence of two high power solitons (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation)

with considerably different characteristic scales: dispersion is small while beam diffraction is much stronger [46, 53]. The stabilization of solitons due to the lack of a suitable media is an issue that seems to delay indefinitely their experimental observation. An extensive review of the subject can be found in [53].

The study of solitons in optical lattices is recent and is boosted mainly by the discovery and development of new media and metamaterials, that can have a complex spatial structure with different components, each having distinct optical properties, including linear and nonlinear refraction index. Here we use the term lattice loosely, as often the pattern is not periodic like an actual lattice. Electromagnetically induced transparency is one of the phenomena that makes such studies possible, with systems like Λ and N gases offering strong nonlinearities and refraction index controllable by a proper choice of control fields [24, 37, 28].

In the case of linear lattices, deep research has been done during the past two decades. One configuration that received much attention is the Bessel lattice, because of the non-diffracting properties of the Bessel beams that can provide large distance waveguides [36]. These have been shown to support new profiles of solitons [39], as well as rotating and orbiting solitons [38, 33]. In the same way, other interesting examples are the Laguerre lattices that produce ring and rotating solitons [28], periodic lattices - like the case of fiber Bragg grating - with gap soliton solutions [46] and random lattices, supporting Brownian-like motion [41]. The possibility of steering light by distributions of the linear refraction index arises as the most interesting application of these studies [78, 40].

The nonlinear lattices only captured the attention in the last eight years and are still in their first steps of research. New soliton profiles and steering of light are predicted in a variety of nonlinear lattices (see [37] for an extensive review), like for example, higher

order bound states in arrays of waveguides [37], oscillations of solitons in sinusoidal lattices [42] and surface solitons in arrays of nanowires [84]. The importance of such structures lies mostly on the achievement of stable solitons impossible in bulk media, like two dimensional spatial solitons in a metamaterial of nanowires embedded in a Kerr media [75] or in photonic crystals and lattices[46, 23].

Optical solitons are one of the most active topics in nonlinear science and that is not only due to the theoretical phenomena involved but also to possible applications of the technology. Robustness of temporal solitons made them a natural candidate for long-haul communication systems [46] where normally one uses pulsed beams to transmit binary digital data. Actually, fiber communications are the foundation of modern connections, with the record of data transfer rate set this year at 31 Tb/s over a distance of 7200 km, which is equivalent to sending almost 300 high definition movies in one second [70]. The theory of solitons comes into the practical world especially in the investigation of the timing jitter limitation due to soliton interaction. An extensive but outdated review of solitons in optical communications can be found in [32]. In turn, spatial solitons also have their applications. The newsworthy properties of soliton interaction, collision and also steering in optical lattices can be used for idealization of ultra-fast fully-optical devices like logic gates [55], appealing for the field of optical computing and optoelectronics.

1.4 Analytical, variational and numerical methods

Unfortunately, only the (1+1)-d NLSE is an integrable equation via IST. Usually the GNLSE cannot be investigated with this elegant analytical tool. Therefore, the analysis of solitons in multidimensional systems and optical lattices relies on one of two strategies: variational techniques [4] or full numerical integration [2].

The variational approach is something in between the analytical and the pure numerical methods of analysis. The idea is to assume a trial function and predict the dynamics of the parameters from the resulting Euler-Lagrange set of equations [4]. This technique can also be used to predict the shape of solitons in media with higher nonlinearities [56, 18]. When used to describe interactions between solitons [35] and dynamics in linear optical lattices [83, 36, 40, 31] this method is called effective particle approach. Variational methods have been successful in explaining several results, including the collapse of spatial beams [67], flat-top beams in saturable cubic-quintic materials [18] and the properties of liquid-light [59].

However, neither analytical nor variational tools are capable of explaining solitons

in more complex situations like nonlinear lattices. Also, even for the easiest cases the variational methods can be rather complicated to use[9]. To overcome these difficulties, numerical simulations can be very useful, providing an efficient way of doing elaborated research. Many methods and algorithms developed in the past [25, 2] fall into one of two categories: methods based on finite difference (FD) or based in pseudo-spectral (PS) methods. Both of these two classes of methods are based on the splitting of the evolution of the NLSE into linear and nonlinear steps. However, the PS methods use the Fast-Fourier Transform (FFT) - or an equivalent one - of the "pulse" to perform the linear step in Fourier space, while doing the nonlinear step in direct space. On the other hand, the FD methods do both these steps in the direct space. For slowly varying envelope situations, this distinction makes the PS methods faster by an order of magnitude for the same error, as the PS admits a bigger integration step [2]. However, when the dynamics of the envelope is faster and it becomes necessary to use a smaller integration step, FD frequently present similar or better performance [2].

In general the performance of each method or algorithm depends not only on the characteristics of the physical problem under study but also on the type of computer system that is going to be used. For example, the same PS method can have performances differing by several orders of magnitude when being computed in a single core desktop, a computer cluster, or a supercomputer. Therefore it is very important not only to carefully select the numerical methods and type of computer system to be used, but also to take into account the interplay between these two factors.

1.5 GPU computing and prospects for the NLSE

As discussed in earlier sections, computers are a fundamental tool of analysis of the nonlinear science problems, including the NLSE. Today, many interesting research problems require large simulations, which involve multi-core CPUs. Therefore we must look into the new paradigms of computer theory and technology that define the computing power required to address this more complex problems.

It is interesting to notice that even though the first computers were developed to perform intense numerical calculations (especially cracking German war codes during second world war), after the late 1980s computers were integrated in everyday life, from business to household desktops and personal laptops, performing many other tasks other than numerical and scientific calculations. This is currently the core of the computer industry, driving their research and their quest for higher performances. Though numer-

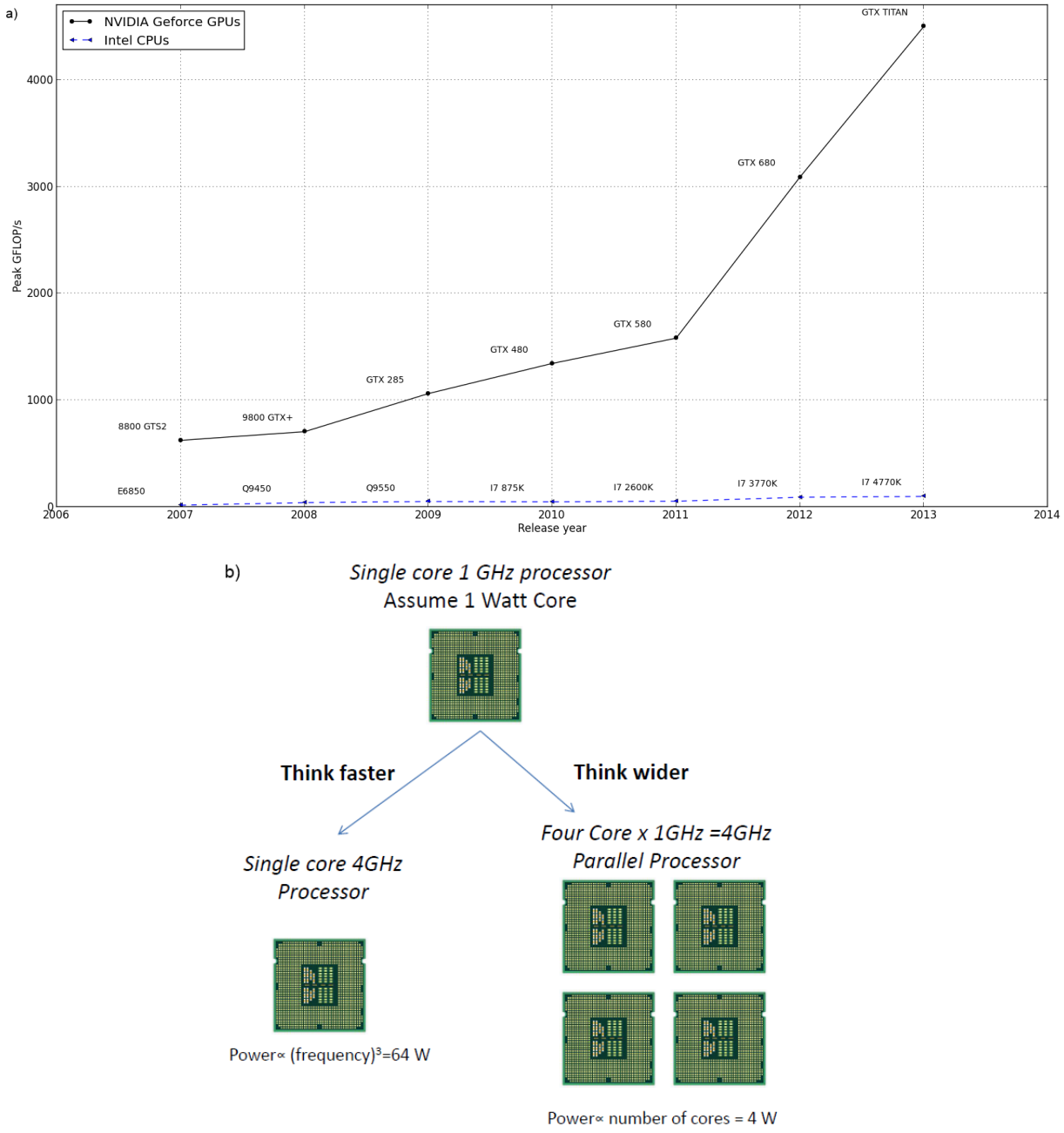


Figure 1.4: a) GPUs have long surpassed peak performances of CPUs, which triggered the development of GPU computing. b) Also, they attain this peak performances without prohibitive power consumptions, characteristic of frequency scaling of the CPUs, using the parallel computing paradigm.

ical scientific computing is no longer a nuclear objective of computer developers, it can still much benefit from the continuous waves of innovation and improvements constantly occurring in this technology.

During the past decade, the frantic demand for faster processors by software industries made the computer engineers to come up with the parallel programming paradigm for common purposes. This included the development of multi-core CPUs with ever increasing number of cores and the development of the necessary software to allow them to work in parallel. This helped to achieve computer performances as never before. The quest for parallel computation is not limited to CPUs. GPUs also experienced a similar evolution. Modern GPUs contain several hundreds of cores and they exploit the highly parallelizable task of calculating the value of a pixel, thus speeding up video games and image processing. The former is so important for the gaming industry that GPUs have long surpassed multi-core CPUs in both number of cores and computer performance, reaching incredible speeds of the order of a Teraflop/s, while the best CPUs are limited up to 50 GFLOP/s. GPUs have yet another advantage, as they cost the same or less than a CPU. This fact can get even striking, as the performance of a mid-range GPU is the same of the best supercomputers of year 2000, that cost 110$ million, for example ASCI White.

Excited by this technology, science world started to think about the use of GPU for general purposes (GPGPU) such as scientific computations, but the first approaches were challenging, as no easy and versatile programming framework was available. Recognizing the problem, NVIDIA made an effort to make this potential available to the industry and scientific community by developing a new programming framework for NVIDIA GPUs called CUDA. With CUDA and more recently with OPENCL - a platform-independent framework - the modern researcher can nowadays move his computational codes to the GPU of his personal computer and obtain speedups[2] worthy of a modern supercomputer. However life is not perfect yet! There are some obstacles for the average physicist to become a GPGPU user, as learning the architecture of the GPU and new programming paradigms can be time consuming and a truly jigsaw puzzle.

The first use of GPU computing for solving differential equations was probably by Mark Harris [29] and ever since GPU computations proliferated in many areas of physics. A few examples of scientific computation using GPU include:

- Fluid dynamics: the power of GPU is exploited to simulate large systems showing

[2]Speedup is a measure of the relative performance of a code developed in parallel when comparing to serial computations, defined usually as $\text{Speedup} = \frac{\text{serial computing execution time}}{\text{parallel computing execution time}}$.

turbulence obtaining simulations 22 times faster than CPUs [44];

- Statistical physics: Multidimensional Ising model simulations were done with speedups of 8 [47, 66], while Brownian dynamics and reaction-diffusion systems were simulated with speedups of 8 and 55 [65, 82];

- Electromagnetic waves: Maxwell equation was simulated with speedup of 50 and 60 in two [7] and three [57] dimensional systems, respectively.

NVIDIA also provides an extensive report [15] that shows the utilization of graphical cards for GPU-ready commercial software. From bioinformatics - sequence mapping software up to 100 times faster - to computational finances - financial analytic software 500 times faster - the applications are numberless.

It was in 2009-2012 timeframe that Ron Caplan developed the first approach of a NLSE solver using GPU computations during his PhD at San Diego University [11]. The code developed, that later became a package for MATLAB called NLSEmagic, served his purpose obtaining simulations up to 20 times faster than a CPU based script [12]. This code was a conceptual breakthrough by demonstrating the potential power of GPUs in solving the NLSE. However, it had many important limitations:

- First, is based on a Runge-Kutta scheme, which is an explicit FD method and then conditionally stable and slower than PS methods for most of the computations;

- NLSEmagic does not admit spatial distribution of the nonlinearities, which is very important for current research;

- Finally, NLSEmagic package is developed in MATLAB and although he used CMEX - a MATLAB interface for developing part of the code in C - MATLAB is still a scripting language, hence with low performance when compared with C or C++. Based on the experience acquired during the preparation of this master's dissertation, it is my opinion that even if Caplan has achieved a large speedup, he was strongly limited by using a MATLAB script, which can be easily outperformed by a CPU C++ based code.

In conclusion, GPU computing has an enormous potential for researchers and engineers. In particular, there is still space for improvements regarding high performance GNLSE solvers and integrators. Regardless of the NLSE simulation in GPU being already developed recently, it is our belief that there is still room for progress and higher performance, because it neither allows to solve the GNLSE nor uses high-performance C++ language,

nor it is based on spectral methods, which are usually more stable and have higher performance.

1.6 GASE - GPU Accelerated Soliton Explorer

The solver of the GNLSE developed during this dissertation project was named GASE. It consists in an executable file compiled from a CUDA C++ code compiled using Microsoft Visual Studio, which computes the numerical solutions of the GNLSE, as well as a series of complementary scripts in Python which do the analysis of the data and produce the graphical outputs. The solver can run in a normal computer having a NVIDIA GPU installed and enabled to use CUDA. This hardware component usually costs about few hundreds euros and is the element responsible for doing most of the massive computation in GASE.

The code GASE is capable of simulating physical problems with 1, 2 and 3 spatial dimensions in simulation boxes with a number of sampling points up to 2^{23}, although this value is only limited by the hardware and not by the code itself. GASE can simulate systems with any type of nonlinearity, including cubic, quintic and logarithmic, as well as nonlinearities that have a spatial dependence. Also, a recent upgrade of GASE allows to simulate a system of two coupled GNLSE (this can also be extended for more than two GNLSE).

The code is also prepared to simulate problems with periodic, reflective and absorbing boundary conditions. In short, this code has a high performance when compared with other sequential algorithms and is designed to be able to address a wide class of problems.

1.7 Outline and structure of the dissertation

This dissertation addresses aspects of two immense topics: GPU computing and the NLSE. Its main output is the development of a solver of the GNLSE based on CUDA in C++ framework, that uses GPU computing and is capable of addressing problems with high dimensionality and spatial distribution of nonlinearities, such as optical lattices. We hope that this output can give a contribution to other researchers by providing them with a tool to investigate computationally modern problems in nonlinear science. This is the result of one year of work whereas becoming an expert in both GPU computing and the NLSE requires years of dedication. Not surprisingly, there is still space for further improvement of the simulation code, as well as to explore its full scientific potential.

It has been a long way to reach this point, a road covered with many hours at the computer, testing different algorithms, making few mistakes and learning from them. In the words of Edison, "I have many results, I know many things that do not work". Indeed, the hardest tasks during the last year were to learn C++ and CUDA, to become familiar with GPU architecture and with the concepts needed in the development of the code, and to overcome all the difficulties that come with unexplored territory, without having any work as reference as it was one of the first GPU based codes developed at the department. These tasks consumed more than eight to nine full months but however they have no place in this dissertation, as it only reports what worked well. In an analogy, this dissertation is like a building: the outcome can be analyzed and we can tell how we built it, but the hard work and the needed strength can be wrongly overlooked.

The dissertation is structured as follows. In this first chapter, a general overview of the subject was given, in theory of solitons, NLSE and GPU computing. A small motivation and the framework was also discussed.

In Chapter 2 a brief synopsis of NLSE and solitons is presented, discussing succinctly the mathematical formulation and examples of variational methods and effective particle approach. The numerical methods, both finite-differences and a pseudo spectral method called Split Step Fourier Method (SSFM) are introduced, followed by a discussion of the boundary conditions. The code implementation is discussed in chapter 3, and a comparison of the performance of GPU-based versus CPU-based simulations is described in chapter 4. In chapter 5 and 6 we present two case studies as proof of concept of the developed tools. In chapter 5 we analyze a one dimensional chain of spatial solitons, predicting numerically and showing computationally the possibility of having phonon-like oscillations. Chapter 6 is devoted to the problem of soliton collision in (2+1)-d system, investigating both the in-phase and out-of-phase soliton collision. Finally, future perspectives and an outline of the main conclusions are provided in chapter 7.

2 Nonlinear Schrödinger equation in a nutshell

This chapter is devoted to briefly review some of the principal aspects of the NLSE and its solution, focusing on the main analytical results and introducing the most relevant numerical methods. Given the tremendous work done over the years and extensive literature in this topic, we restrain this review to the aspects which are most relevant to future chapters and more specifically to the development of the code GASE.

In particular, we mainly discuss the solutions of the (1+1)-d NLSE, which is (to our knowledge and so far) the only case with a generic method of obtaining exact analytical solutions via IST method. However, we also describe ways of obtaining approximate soliton-like solutions for the GNLSE in cases with higher dimensions. Finally, we discuss the most notorious successful numerical methods used to solve computationally the GNLSE, namely the Finite Differences (FD) methods and Pseudo-spectral (PS) methods. In particular, the review of the Split-step Fourier method (SSFM) establishes the ground base for the two following chapters, since it corresponds to the numerical method used in the development of our solver of the GNLSE. In the following chapter we describe how this numerical method was adapted and implemented to work on GPUs and make use of its tremendous computing power.

2.1 NLSE and analytical solutions

Traditionally, the NLSE refers to the Schrodinger equation where an extra term was added, corresponding to a cubic nonlinearity:

$$i\frac{\partial\psi}{\partial z} + \frac{1}{2}\nabla_\perp^2\psi + s|\psi|^2\psi = 0. \tag{2.1}$$

This model is widely spread in nonlinear science and describes the evolution of a dimensionless amplitude field ψ in a dispersive and weakly nonlinear medium. In this formulation, the coordinate z is usually associated to the longitudinal direction, along

which the field propagates, while the $\nabla_\perp^2$ is the Laplacian in the transverse directions. Also the number $s = \pm 1$ refers to the sign of the nonlinearity.

The (1+1)-d NLSE corresponds to a simplified version of equation (2.1) and is given by

$$i\frac{\partial\psi}{\partial z} + \frac{1}{2}\frac{\partial^2\psi}{\partial x^2} + s|\psi|^2\psi = 0. \tag{2.2}$$

It has soliton solutions for both values of s, which can be calculated using IST ([14]). In nonlinear optics equation 2.2 has two major applications. On one hand, when x refers to a spatial coordinate then, the NLSE describes the confinement of a CW light beam in a Kerr media [14]. On the other hand, when x refers to a temporal coordinate (sometimes x is replaced by τ in this equation to make the temporal character more evident), then, NLSE describes the propagation of a pulsed beam in an optical fiber [30]. From now on, and unless noted otherwise, only spatial solitons are considered.

For $s = 1$, the media is also called self-focusing and the supported solutions are called bright solitons. The one dimensional bright soliton solution of equation (2.2), centered at constant $x = \bar{x}_0$, is given by [46]

$$\psi(x,z) = 2\nu \mathrm{sech}\left[2\nu\left(x - \bar{x}_0\right)\right] \exp\left(2i\nu^2 z\right), \tag{2.3}$$

where ν is the amplitude of the soliton. It can be proven that the NLSE is invariant under the Galilean transformation [77]

$$\begin{aligned} x \mapsto x' &= x - \mu z \\ z \mapsto z' &= z \\ \psi(x,z) \mapsto \psi'(x',z') &= \psi'(x - \mu z, z) \exp\left(i\mu x - i\mu^2 z/2\right) \end{aligned} \tag{2.4}$$

allowing us to consider a more general solution of a moving soliton,

$$\begin{aligned} \psi(x,z) &= 2\nu \mathrm{sech}\left[2\nu\left(x - \bar{x}_0 - \mu z\right)\right] \exp\left\{i\mu\left(x - \bar{x}_0\right) + i\delta(z)\right\} \\ \delta(z) &= (2\nu^2 - \mu^2/2)z + \delta_0 \end{aligned} \tag{2.5}$$

where ν is the amplitude, $\bar{x}_0$ is the initial position of the centroid of the field distribution ψ, μ is the transverse velocity and δ_0 is the initial phase of the soliton. Bright solitons can exist for all values of ν and μ, constituting a two-parameter family of solutions.

For $s = -1$, the media is called self-defocusing and the supported solitons are called

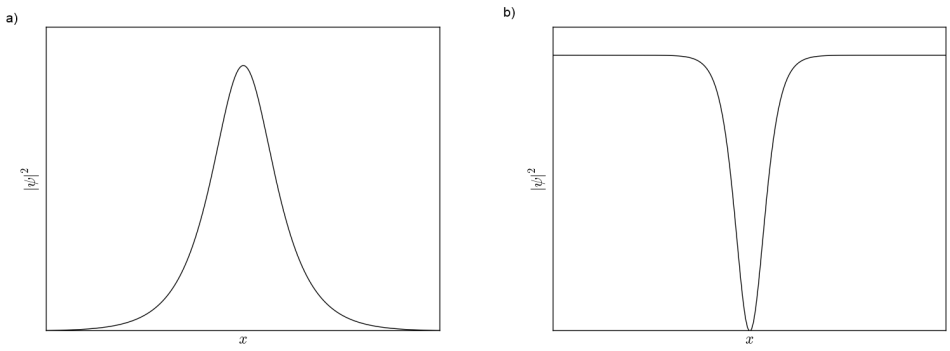


Figure 2.1: Depending on the positive or negative sign of the nonlinearity, solitons can be either bright - figure a) - or dark - figure b) - respectively. Both were represented in arbitrary units.

dark solitons. Using the IST and the boundary condition $|\psi| = \psi_0$ as $x \to \infty$, these dark solitons solutions correspond to a localized intensity reduction in an otherwise constant CW background. They can be expressed analytically [46] as

$$\psi(x,z) = \psi_0 \left\{ B \tanh\left[\psi_0 B \left(x - A\psi_0 z\right)\right] + iA \right\} \exp\left(-i\psi_0^2 z\right) \tag{2.6}$$

where the two parameters A and B obey the relation $A^2 + B^2 = 1$. Even though it is possible to interpret the parameter A as being related with the velocity of the dark soliton, the similarities end here. Dark and bright solitons have very distinct properties since they are not the dual of each other. Such properties will not be discussed here since they fall out of the scope of this dissertation. From now on, the discussion is focused on the case of bright solitons, that shall be designated simply as solitons.

The case of (1+1)-d solitons is well studied in the literature, much due to the development of IST. However, the dimensionality of the physical system has a key role on the nature of the solutions of NLSE. Indeed, it is proven that for the (D+1)-d NLSE, with a Kerr-type nonlinearity, a generic localized envelope-like solution collapses into a singularity for $D = 2$ [46, 87]. This behavior is indicative of the difficulty of finding envelope and soliton-like solutions of the NLSE in systems with high dimensionality. In fact, stable (2+1)-d solitons are only possible considering higher and saturable nonlinearities, described by the GNLSE.

2.2 GNLSE

The GNLSE is generalization of the NLSE obtained by replacing the cubic nonlinearity with a generic nonlinear term, say

$$i\frac{\partial\psi}{\partial z} + \frac{1}{2}\nabla_\perp^2\psi + F(|\psi|^2)\psi = 0, \tag{2.7}$$

where $F(|\psi|^2)$ is a real valued function describing the nonlinearity. The interest of the GNLSE is that for certain nonlinearities it allows for (D+1)-d soliton-like solutions with $D > 1$, and specially for (2+1)-d. Various types of non-Kerr law nonlinearities had been studied, these include the following:

- *Parabolic law:* $F(|\psi|^2) = |\psi|^2 + s|\psi|^4$

  For many optical materials and media, the refractive index begins to deviate from the Kerr type for large intensities of ψ. For example, a polydiacetene para-toluene sulfonate (PTS) crystal has a parabolic law dependence for the refraction index with $s < 0$ [18]. This situation, usually called cubic-quintic media, relies on the competition between the two nonlinearities to stabilize (2+1)-d solitons. In fact, at low intensities the self-focusing dominates the system, but for high intensities the beam collapse is avoided by the self-defocusing effect.

- *Saturating law:* $F(|\psi|^2) = \lambda\left(1 - \frac{1}{(1+|\psi|^2/|\Psi_{sat}|^2)}\right)$

  Simple two-level atomic systems [9] or photorefractive materials [37] displays a type of nonlinearity which saturates for field intensities above $|\psi_{sat}|^2$.

On the other hand, when the nonlinear term in the (2+1)-d GNLSE depends explicitly on the spatial coordinates (described by a formal dependence of F on the transverse and longitudinal coordinates, say $F(|\psi|^2; r_\perp, z)$) it describes the so called optical lattices, which can also support the formation and propagation of solitons. Typically optical lattices can be classified into two main classes:

- *Linear lattices:*

  Considering that $F(|\psi|^2; r, z) = F_1(|\psi|^2) + V(r, z)$ we obtain an equation that governs the solitons in a linear lattice

  $$i\frac{\partial\psi}{\partial z} + \frac{1}{2}\nabla_\perp^2\psi + F(|\psi|^2)\psi + V(r, z)\psi = 0 \tag{2.8}$$

  described by the potential $V(r, z)$. In BECs, $V(r, z)$ is the trapping potential and this equation is usually called Gross Pitaevski equation.

- *Nonlinear lattices:*

  Nonlinear lattices are spacial distributions of the nonlinearities. It is not possible to present a general formula for this case but for example, one can consider a cubic-quintic nonlinearity modulated in space by the function $R(r,z)$, being the problem governed by

$$i\frac{\partial\psi}{\partial z}+\frac{1}{2}\nabla_{\perp}^{2}\psi+R(r,z)\left(|\psi|^{2}-|\psi|^{4}\right)\psi=0. \tag{2.9}$$

Unfortunately, it is only for very specific cases that GNLSE constitutes an integrable model with analytical soliton solutions. In a rigorous sense, solutions of nonintegrable systems are not solitons, but it is common to use the term because most of the solutions normally tested are soliton-shaped waves. As nonintegrable GNLSEs are important models in nonlinear physics, it is necessary to develop methods capable of analyzing the properties of solitons in such systems. In the following sections some of these methods are briefly reviewed.

2.3 Noether's theorem and conservation laws in the GNLSE

Unlike the (1+1)-d NLSE, the GNLSE does not possess an infinite number of conserved quantities [14]. However, the existence of some conserved quantities is of great importance for the analysis of the GNLSE, as it can provide clues about soliton behavior.

An investigation of the conservation laws is based on the structure of the Lagrangian associated with GNLSE described by equation (2.7). This Lagrangian density is defined as

$$\mathcal{L} \;=\; \frac{i}{2}\left(\psi^{*}\psi_{z}-\psi\psi_{z}^{*}\right)-\frac{1}{2}\nabla_{\perp}\psi\nabla_{\perp}\psi^{*}+G\left(|\psi|^{2}\right), \tag{2.10}$$

with

$$G(\lambda)=\int_{0}^{\lambda}F(\lambda)d\lambda, \tag{2.11}$$

where we introduced the notation $\frac{\partial\psi}{\partial z}=\psi_{z}$ and $\mathbf{x}$ refers to the transverse coordinates [77].

The corresponding action is

$$S\left\{\psi,\psi^*\right\} = \int_{z_0}^{z_1} \int_{all\ space} \mathcal{L}\, d\mathbf{x}\, dz. \tag{2.12}$$

According to the formulation of field theory of Noether's theorem, an action invariant under the following infinitesimal transformation

$$\begin{aligned} z &\mapsto z' = z + \delta z(\mathbf{x}, z, \psi) && (2.13)\\ \mathbf{x} &\mapsto \mathbf{x}' = \mathbf{x} + \delta\mathbf{x}(\mathbf{x}, z, \psi) && (2.14)\\ \psi(\mathbf{x}, z) &\mapsto \psi(\mathbf{x}', z') = \psi(\mathbf{x}, z) + \delta\psi(\mathbf{x}, z) && (2.15) \end{aligned}$$

must conserve

$$\int \left[\frac{\partial\mathcal{L}}{\partial\psi_z}\left(\psi_z\delta z + \nabla_\perp\psi\cdot\delta\mathbf{x} - \delta\psi\right) + \frac{\partial\mathcal{L}}{\partial\psi_z^*}\left(\psi_z^*\delta z + \nabla_\perp\psi^*\cdot\delta\mathbf{x} - \delta\psi^*\right) - \mathcal{L}\delta t\right] d\mathbf{x}. \tag{2.16}$$

From this result, it can be proven that the GNLSE has three conserved quantities, namely

$$\begin{aligned} E &= \int |\psi|^2 d\mathbf{x} && (2.17)\\ H &= \int \left[\frac{1}{2}\nabla_\perp\psi\nabla_\perp\psi^* - G(|\psi|^2)\right] d\mathbf{x} && (2.18)\\ M &= \frac{i}{2}\int \left[\psi\nabla_\perp\psi^* - \psi^*\nabla_\perp\psi)\right] d\mathbf{x} && (2.19) \end{aligned}$$

Quantity E results from the invariance of the action under a phase shift transformation and can be interpreted as a conservation of an energy type quantity. It is usually referred as soliton energy, mass, plasmon number or, in the case of optical solitons, as wave power.

The second quantity H corresponds to the Hamiltonian of the soliton and its conservation reflects the invariance of the action under time shift.

Finally the invariance of the action under spacial shifts or translations leads to the conservation of the linear momentum, represented by M.

These conservation laws and the Lagrangian formalism behind them have an important role in the development of the variational methods used to study the soliton solutions in GNLSE systems discussed in the following section.

2.4 Variational methods

2.4.1 Trial functions and solutions

An important method to obtain approximate stationary solutions of the GNLSE is to consider trial functions that depend on a set of parameters that are determined using the Ritz optimization procedure. For example, when looking for stationary solutions, it is possible to use the method of separation of variables to write $\psi(\mathbf{x}, z) = g(\mathbf{x})\exp(i\delta z)$, where g is the shape function of soliton and the second factor is the phase of the soliton. Typically g depends on a set of parameters $\mathbf{p}$ (in a formal way $g \equiv g(\mathbf{x};\mathbf{p})$) besides the coordinates $\mathbf{x}$ which determine the soliton amplitude and width, among others. Then the substitution of ψ in the GNLSE, yields the nonlinear eigenvalue problem

$$\frac{1}{2}\nabla_{\perp}^{2} g + F(|g|^2)g - \delta g = 0. \tag{2.20}$$

Then using the Lagrangian density of the GNLSE , corresponding to equation (2.10), it is possible to compute

$$\langle L\rangle = \int\limits_{all\ space} \mathcal{L}(g)d\mathbf{x}. \tag{2.21}$$

Now, the Ritz optimization procedure allows to determine the values of the parameters $\mathbf{p}$ which maximize the resemblance between the trial functions and the exact solutions of GNLSE. This method consists in solving the following equation

$$\frac{\partial \langle L\rangle}{\partial \mathbf{p}} = 0. \tag{2.22}$$

The method appears to be very simple but its success depends on the right choice of the trial function. It has been proven successfully in many situations by providing a good insight of the properties of the solutions of the GNLSE, particularly for the (2+1)-d cubic-quintic GNLSE, by helping to explain flat-top beams [18] and liquid-light [58]. For a cubic-quintic media described by $F(|g|^2) = |g|^2 - |g|^4$, the states are supergaussians of order m described by

$$g(\mathbf{x}) = A\exp\left(-\frac{1}{2}\left(\frac{\mathbf{x}-\bar{\mathbf{x}}_{\mathbf{0}}}{a}\right)^{2m}\right), \tag{2.23}$$

where A is the amplitude, a is the width, $\bar{x}_0$ is the initial position of the centroid of the field distribution g. With the Ritz optimization procedure, expressions for the amplitude and the width can be given by [18]

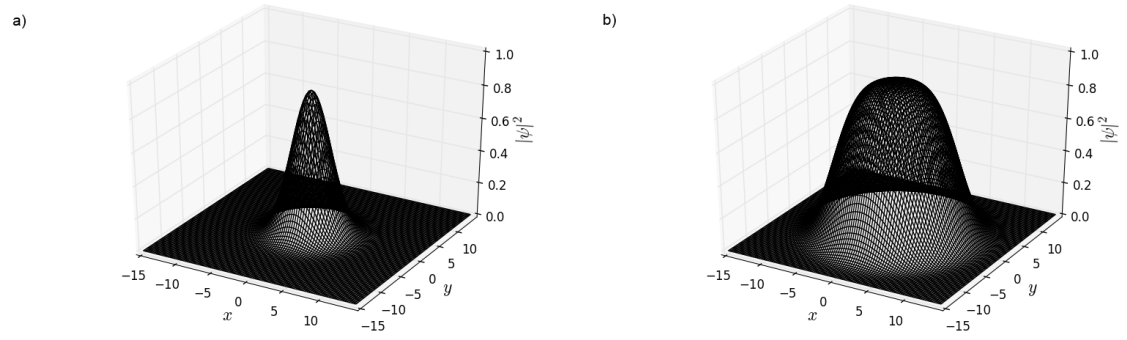


Figure 2.2: Supergaussian shapes of: a) a low power soliton with $m = 1$; b) a high power soliton with $m = 1.9$. The competition between focusing cubic and defocusing quintic nonlinearities made high power solitons to spread, acquiring a flat-top form. Typically, the high power supergaussians exhibit liquid-like behavior, while low power exhibit other behaviors.

$$A = \sqrt{\left(\frac{3}{2}\right)^{1/m} \frac{3}{2} \frac{m - \ln 2}{2m - \ln 3}} \tag{2.24}$$

$$a = \frac{1}{A} \sqrt{\frac{2^{1+1/m} m}{\Gamma(1 + 1/m)} \frac{2m - \ln 3}{\ln(4/3)}} \tag{2.25}$$

both dependent on the order m. It can be proven that the parameter m is directly proportional to power and thus, high power states are those with higher values of m.

2.4.2 Perturbation methods

Another technique used for obtaining approximate solutions of the GNLSE are perturbation methods. Many of these approaches have been developed over the years, not only for NLSE and GNLSE but for other similar nonlinear equations [46]. This section is focused on a simple formulation of the adiabatic perturbation theory developed by Anjwan Biswas [9] for (1+1)-d, which can be easily extended to systems with higher dimensions. In some physical systems, after some manipulations, it is possible to derive a perturbed (1+1)-d GNLSE of the form

$$i\psi_z + \frac{1}{2}\frac{\partial^2 \psi}{\partial x^2} + F(|\psi|^2)\psi = i\epsilon R[\psi, \psi^*] \tag{2.26}$$

where R is the function describing a perturbation term and ϵ is a small parameter. It is also assumed that the general form of the soliton solution of the previous equation is of the form

$$\psi(x,z) = A(z)g[B\{x-\bar{x}(z)\}]\exp(-ik(z)\{x-\bar{x}(z)\}+i\theta(z)), \tag{2.27}$$

where A, B, g, k, θ,$\bar{x}(t)$ are the amplitude, the width, the shape, the frequency, the phase and the centroid of the soliton, respectively. These parameters can be calculated from ψ according to the following expressions:

$$A(z) = \left[\frac{I_{0,2,0,0,0}\int_{-\infty}^{\infty}|\psi|^4 dx}{I_{0,4,0,0,0}\int_{-\infty}^{\infty}|\psi|^2 dx}\right]^{1/2} \tag{2.28}$$

$$B(z) = \left[\frac{I_{2,2,0,0,0}\int_{-\infty}^{\infty}|\psi|^2 dx}{I_{0,2,0,0,0}\int_{-\infty}^{\infty}x^2|\psi|^2 dx}\right]^{1/2} \tag{2.29}$$

$$k(z) = \frac{i\int_{-\infty}^{\infty}(\psi\psi_x^* - \psi^*\psi_x)dx}{2\int_{-\infty}^{\infty}|\psi|^2 dx} \tag{2.30}$$

$$\bar{x}(z) = \frac{\int_{-\infty}^{\infty}x|\psi|^2 dx}{\int_{-\infty}^{\infty}|\psi|^2 dx}, \tag{2.31}$$

where the integrals $I_{l,m,n,p,r}$ are given by

$$I_{l,m,n,p,r} = \int_{-\infty}^{\infty}\tau^l g^m(\tau)(\frac{dg}{d\tau})^n(\frac{d^2g}{d\tau^2})^p(\frac{d^3g}{d\tau^3})^r d\tau. \tag{2.32}$$

The evolution of these parameters resulting from the perturbative analysis of equation (2.26). In detail, differentiating equation (2.17) and (2.28-2.31) with respect to z and considering the perturbed GNLSE (2.26) it is possible to deduce that, if the shape of soliton is kept unchanged under a perturbation R, then the characteristic parameters of the soliton must satisfy the following evolution equations

$$\frac{dE}{dz} = \epsilon\int_{-\infty}^{\infty}(\psi^*R+\psi R^*)dx \tag{2.33}$$

$$\frac{dA}{dz} = \frac{\epsilon}{I_{0,4,0,0,0}}\frac{B}{A^3}\int_{-\infty}^{\infty}|\psi|^2(\psi^*R+\psi R^*)dx \tag{2.34}$$

$$\frac{dB}{dz} = \frac{2\epsilon}{I_{0,4,0,0,0}}\frac{B^2}{A^4}\int_{-\infty}^{\infty}|\psi|^2(\psi^* R + \psi R^*)dx - \frac{\epsilon}{I_{0,2,0,0,0}}\frac{B^2}{A^2}\int_{-\infty}^{\infty}(\psi^* R + \psi R^*)dx \tag{2.35}$$

$$\frac{dk}{dz} = \frac{i\epsilon}{I_{0,2,0,0,0}}\frac{B}{A^2}\int_{-\infty}^{\infty}(\psi_x^* R - \psi_x R^*)dx - \frac{\epsilon}{I_{0,2,0,0,0}}\frac{kB}{A^2}\int_{-\infty}^{\infty}(\psi^* R + \psi R^*)dx \tag{2.36}$$

$$\frac{d\bar{x}}{dz} = -k + \frac{\epsilon}{I_{0,2,0,0,0}}\frac{B}{A^2}\int_{-\infty}^{\infty}x(\psi^* R + \psi R^*)dx \tag{2.37}$$

$$\begin{aligned}\frac{d\theta}{dz} &= -\frac{k^2}{2} - \frac{B}{2}\frac{I_{0,0,2,0,0}}{I_{0,2,0,0,0}} + \frac{1}{I_{0,2,0,0,0}}\int_{-\infty}^{\infty}F(A^2 g^2(s))g^2(s)ds + \\ &+\frac{i\epsilon}{I_{0,2,0,0,0}}\frac{B}{A^2}\int_{-\infty}^{\infty}(\psi^* R - \psi R^*)dx.\end{aligned} \tag{2.38}$$

This system of equations describes completely the dynamics of weakly perturbed soliton and can be extrapolated for higher dimensional systems. A variety of situations can be studied under this formalism, such as soliton-soliton interactions [56] and soliton steering in linear optical lattices [41, 38]. This method is sometimes called the effective particle approach.

2.4.3 Effective particle approach

The effective particle approach [83] is a special case of the perturbed GNLSE discussed in the previous section and it describes the behavior of a soliton in a (1+1)-d linear optical lattice governed by the equation

$$i\frac{\partial\psi}{\partial z} + \frac{1}{2}\frac{\partial^2\psi}{\partial x^2} + F(|\psi|^2)\psi + V(r,z)\psi = 0, \tag{2.39}$$

where V is a real valued function that describes the linear refraction index profile. For small values of modulation of the refraction index, it is possible to apply the formalism derived in section (2.4.2). With the help of equation (2.26) it is possible to identify $R = iV\psi/\epsilon$ and use the formulas (2.33-2.38) to prove that the amplitude, the width and the energy of the soliton remain constant.

Using this approach it is possible to recover the notorious result from quantum mechanics known as the Ehrenfest theorem, which basically expresses the correspondence principle between classical and quantum mechanics. In particular, the equation of the evolution of the centroid of the soliton, obtained using effective particle approach is

$$\frac{d^2\bar{x}}{dz^2} = -\frac{1}{E}\int_{-\infty}^{\infty} |\psi|^2 \frac{\partial V}{\partial x} dx. \tag{2.40}$$

In short, it describes a quantum equivalent of Newton second law for a classical particle under the influence of a potential V, namely

$$\frac{d^2x}{dz^2} = -\nabla V. \tag{2.41}$$

The effective particle approach can be used not only to study the trajectory of solitons in optical lattices, but also can provide a theoretical framework to study the interaction of several weakly overlapping solitons. However, all this approximative analytical methods have strong limitations in addressing systems with high dimensionality and complexity. In these cases, it is necessary to use an approach with long tradition in nonlinear science and use numerical methods. In the following section some of the most relevant of such methods are discussed.

2.5 Numerical methods

Numerical methods are not so powerful as analytical methods in the sense that one cannot obtain the complete set of solutions for the problem. However, they are still useful since they can avoid cumbersome or impossible calculations by solving the problem for a specific configuration of the solution. This is specially true if one has some intuition about the type and behavior of the solution. In the last three decades most of the theoretical results in soliton propagation were supported by numerical investigations.

In the theory of differential equations, the GNLSE is a nonlinear second-order parabolic partial differential equation and if we remove the nonlinear part, the GNLSE becomes a linear parabolic equation or a diffusion equation. The diffusion equation can be solved numerically using different numerical methods that convert the partial differential equation into an algebraic equation, via some form of discretization. Then the problem can be treated using methods from computational linear algebra which in most cases consist in methods of matrix inversion. Among the algebraic methods used the most com-

mon are the Crank-Nicholson implicit scheme and pseudo-spectral methods which are both unconditionally stable [25]. The presence of the nonlinear term changes drastically the situation and the otherwise known methods are no longer unconditionally stable, becoming important to choose carefully the integration step.

Usually, integration schemes for nonlinear problems rely on the split of the integration step into a linear and a nonlinear sub-steps. The splitting procedure is a well known mathematical method and the idea is to decompose a complex model into a sequence of simple sub-problems. The method has an associated error that can be theoretically estimated. One of the most popular splitting methods is the Strang-Splitting algorithm which constitutes a second order splitting [2]. To explain how the splitting procedure can be applied to the GNLSE we consider that it can be written as:

$$\frac{\partial \psi}{\partial z} = \left(\hat{D} + \hat{N}\right)\psi, \tag{2.42}$$

where $\hat{D} = \frac{i}{2}\nabla_{\perp}^2$ is a linear operator containing the linear terms of the GNLSE and $\hat{N} = iF(|\psi|^2)$ is its nonlinear counterpart. When applied to this equation, Strang-Splitting algorithm is as follows:

$$\frac{\partial \psi}{\partial z} = \hat{D}\psi, \text{ with } z \in [z, z+h/2], \qquad \psi(z) = \psi(z) \tag{2.43}$$

$$\frac{\partial \psi^{NL}}{\partial z} = \hat{N}\psi^{NL}, \text{ with } z \in [z, z+h], \qquad \psi^{NL}(z) = \psi(z+h/2) \tag{2.44}$$

$$\frac{\partial \psi^{L}}{\partial z} = \hat{D}\psi^{L}, \text{ with } z \in [z+h/2, z+h], \quad \psi^{L}(z+h/2) = \psi^{NL}(z+h) \tag{2.45}$$

where h is the integration step. It can be proven that this algorithm is second order accurate. The nonlinear problem is then divided into two simple problems. The linear sub-step requires the integration of the diffusion equation which can be done as previously discussed. The nonlinear sub-step can be integrated numerically using the Euler method, the Runge-Kutta or other methods.

The splitting method is commonly referred in beam propagation studies as the split-step. The idea is that the linear and nonlinear parts of the dynamics can be treated separately considering small integration steps. Depending on the strategy to solve the linear sub-problem of the algorithm, beam propagation methods can be either finite difference or pseudo-spectral.

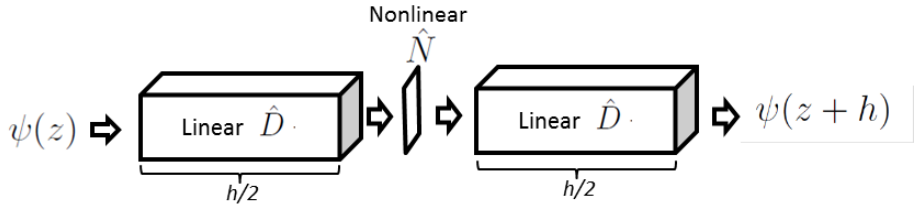


Figure 2.3: Visual scheme describing the split-step algorithm for evolving an initial field $\psi(z)$, described in equations (2.43-2.45).

2.5.1 Explicit and implicit finite differences methods

Finite difference methods can be grouped into two broad categories: explicit or implicit schemes. During this section we briefly present an example of each.

The fourth-order Runge-Kutta (RK4) scheme was used by Ron Caplan in the first CUDA solver of the NLSE [11]. The RK4 is an explicit method that can be used also for the GNLSE. Writing the GNLSE as

$$\frac{\partial \psi}{\partial z} = f(\psi) = i\left[\frac{1}{2}\nabla_{\perp}^{2}\psi + F(|\psi|^{2})\psi\right] \tag{2.46}$$

the RK4 scheme is defined by [12]

$$\begin{aligned} k_1 &= f(\psi(z)) && (2.47)\\ k_2 &= f(\psi(z) + \frac{h}{2}k) && (2.48)\\ k_3 &= f(\psi(z) + \frac{h}{2}k_2) && (2.49)\\ k_4 &= f(\psi(z) + \frac{h}{2}k_3) && (2.50)\\ \psi(z+h) &= \psi(z) + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right). && (2.51) \end{aligned}$$

To convert the GNLSE into an algebraic equation, the domain of ψ is replaced by a discrete set of points that lay on a regular grid. For example, in the case of (1+1)-d GNLSE, the transverse spatial variable is reduced to a set of spatial points separated by a step Δx, while the longitudinal variable becomes a discrete set of values separated by the integration step h. Then, after discretization, the solution ψ is replaced by a discrete set of values $\psi(j\Delta x, nh) = \psi_j^n$ with n and j being integer numbers. This case can be generalized to other spatial dimensions, but we restrict ourselves to the study of the easiest case. In finite differences the Laplacian is computed by means of a stencil

(considering a 3-point stencil)

$$\frac{\partial^2 \psi_j^n}{\partial x^2} = \frac{\psi_{j-1}^n - 2\psi_j^n + \psi_{j+1}^n}{\Delta x^2}. \tag{2.52}$$

Once the Laplacian is computed, nonlinear term corresponds only to a point-to-point vector multiplication and the implementation of the method is complete.

It turns out that this method is not only considerably unstable but also non conservative in the sense that it does not conserve the wave energy defined in equation (2.17) during the system evolution[25]. However it can be shown that an implicit method

$$\frac{\psi_j^{n+1} - \psi_j^n}{h} = \theta f_j\left(\psi^{n+1}\right) + (1-\theta) f_j\left(\psi^n\right) \tag{2.53}$$

is conservative under the condition $\theta = 1/2$, which corresponds to a scheme commonly known as the Crank-Nicholson (CN) scheme. The split-step CN scheme can be obtained from the algorithm (2.43-2.45). The most common strategy is to use a simple Euler or RK4 method for the nonlinear sub-step and a CN for solving the linear sub-steps. In (1+1)-d, using the discretization previously discussed and the 3-point stencil, the linear step can be written as

$$\frac{\psi_j^{n+1/2} - \psi_j^n}{h} = i\left(\frac{\psi_{j-1}^{n+1/2} - 2\psi_j^{n+1/2} + \psi_{j+1}^{n+1/2}}{2\Delta x^2}\right) + i\left(\frac{\psi_{j-1}^n - 2\psi_j^n + \psi_{j+1}^n}{2\Delta x^2}\right) \tag{2.54}$$

Then, grouping the $n + 1/2$ and n terms,

$$\frac{\psi_j^{n+1/2}}{h} - i\frac{\psi_{j-1}^{n+1/2} - 2\psi_j^{n+1/2} + \psi_{j+1}^{n+1/2}}{2\Delta x^2} = \frac{\psi_j^n}{h} + i\frac{\psi_{j-1}^n - 2\psi_j^n + \psi_{j+1}^n}{2\Delta x^2} \tag{2.55}$$

the problem is reduced to the solution of the following linear system for $\psi^{n+1/2}$

$$A\psi^{n+1/2} = A^*\psi^n \tag{2.56}$$

with

$$A = \frac{i}{2\Delta x^2}\begin{pmatrix} \Delta x^2/h + 2 & -1 & 0 & 0 & \dots & & 0 \\ -1 & \Delta x^2/h + 2 & -1 & 0 & \dots & & 0 \\ 0 & & \ddots & & & & \\ \vdots & & \ddots & & & & \vdots \\ 0 & \cdots & & & -1 & \Delta x^2/h + 2 & -1 \\ 0 & \cdots & & & & -1 & \Delta x^2/h + 2 \end{pmatrix} \tag{2.57}$$

This system is usually solved using iterative methods, that, depending on the problem,

can either be fast or slow. Normally, when the matrix is sparse (for example it is a 3-diagonal for (1+1)-d and 5-diagonal for (2+1)-d) there can be considerable speedups if the solver uses sparse matrix algorithms. A solver based on the CN method using GPU computing was implemented by Paulo Alcino in 2012 at INESC Porto.

Even tough the CN scheme relies on the use of iterative methods that are slower than matrix multiplications of RK4, the performance of CN is usually better than RK4 [25] because RK4 is not conservative. In general, to improve the solutions obtained by RK4, it is necessary to use a smaller integration step h, which increases the number of integration steps needed for the simulation and drastically reduces the performance of the method.

2.5.2 Pseudo-spectral methods and the SSFM

Pseudo-spectral methods rely on the utilization of the decomposition of the field ψ in an orthogonal basis of functions, where it is easy to compute the linear sub-step [2]. From direct integration, the exact solution of the equation (2.42) is given by

$$\psi(z+h,\mathbf{x}) = \exp\left(h\left(\hat{D}+\hat{N}\right)\right)\psi(z,\mathbf{x}), \tag{2.58}$$

with $\hat{D} = \frac{i}{2}\nabla_{\perp}^2$ a linear operator relative to the dispersion and $\hat{N} = iF(|\psi|^2)$ relative to the nonlinearities of the media. Using the Strang-Splitting algorithm we can reach an approximation for the solution of the GNLSE as

$$\psi\left(z+h,\mathbf{x}\right) \approx \exp\left(\frac{h}{2}\hat{D}\right)\exp\left(h\hat{N}\right)\exp\left(\frac{h}{2}\hat{D}\right)\psi\left(z,\mathbf{x}\right). \tag{2.59}$$

This means that computationally the solution $\psi\left(z+h,\mathbf{x}\right)$ is calculated from $\psi\left(z,\mathbf{x}\right)$ by applying sequentially the operators $\exp\left(h\hat{D}/2\right)$, $\exp\left(h\hat{N}\right)$ and $\exp\left(h\hat{D}/2\right)$ again. The Baker-Hausdorff formula [2] for two operators $\hat{a}$ and $\hat{b}$ is

$$\exp\left(\hat{a}\right)\exp\left(\hat{b}\right) = \exp\left(\hat{a}+\hat{b}+\frac{1}{2}\left[\hat{a},\hat{b}\right]+\frac{1}{12}\left[\hat{a}-\hat{b},\left[\hat{a},\hat{b}\right]\right]+...\right) \tag{2.60}$$

and can give us a good insight of the error of the method. Indeed, applying the formula two times using $\hat{a} = h\hat{D}/2$ and $\hat{b} = h\hat{N}$, it can be obtained

$$\exp\left(\frac{h}{2}\hat{D}\right)\exp\left(h\hat{N}\right)\exp\left(\frac{h}{2}\hat{D}\right) = \exp\left(h\hat{D}+h\hat{N}+\mathcal{O}\left(h^3\left[\hat{D}-\hat{N},\left[\hat{D},\hat{N}\right]\right]\right)+...\right) \tag{2.61}$$

suggesting that the dominant error term is of the order of h^3 and that the method is accurate up to the second order.

Before advancing, the spatial discretization must be considered. Considering a (3+1)-d system, the discretization on spatial coordinates can be introduced by a grid of integers (j,k,l) where $0 \leq j < N_x$,$0 \leq k < N_y$,$0 \leq l < N_t$. Thus, any point in the continuous spatial space $\mathbf{x} = (x,y,t)$[1] is represented by the corresponding $\mathbf{X} = (j\Delta x, k\Delta y, l\Delta t)$. Is also useful to define a discretization vector $\Delta\mathbf{X} = (\Delta x, \Delta y, \Delta t)$ and the vector number of points $\mathbf{N} = (N_x, N_y, N_t)$.

The Fourier transform of the field ψ is the decomposition of the field ψ in an orthogonal basis of plane waves. Usually, the computational Fast Fourier transform (FFT) maps the complex-valued vector ψ into its frequency domain representation by

$$\hat{\psi}(z,\mathbf{k}) = \sum_{j=0}^{N_x-1}\sum_{k=0}^{N_y-1}\sum_{l=0}^{N_t-1}\psi(z,\mathbf{X})\exp\left(-i\mathbf{k}\cdot\mathbf{X}\right), \tag{2.62}$$

where $\mathbf{X}$ is the discretized space vector. The discretization of the spatial domain reflects as a discretization in the $\mathbf{k} = (k_x, k_y, k_t)$ frequency domain. For even values of the components of the vector $\mathbf{N}$, the discretization can be done in terms of three integers, $(\hat{j},\hat{k},\hat{l})$, within the limits defined by $\mathbf{N}$; however it is not linear like the spatial discretization. For example, k_x is discretized under the formula

$$k_x = \begin{cases} \frac{2\pi\hat{i}}{N_x\Delta x} & ,\text{for } 0 \leq \hat{i} \leq \frac{N_x}{2} \\ \frac{2\pi(\hat{i}-N_x)}{N_x\Delta x} & ,\text{for } \frac{N_x}{2} < \hat{i} < N_x \end{cases}. \tag{2.63}$$

This allows to build a complete map between the field ψ in the discretized direct space and the frequency discretized version $\hat{\psi}$. The advantage of the using the Fourier transforms is that in the frequency space the Laplacian is algebraic, namely

$$\hat{D}(\mathbf{k}) = \frac{-i}{2}\mathbf{k}\cdot\mathbf{k}. \tag{2.64}$$

Therefore, it is possible to evaluate the linear sub-step in Fourier space using

[1] In this sense we are considering t as a spatial variable.

$$\hat{\psi}(z,\mathbf{k}) = F_T\left\{\psi\left(z,\mathbf{X}\right)\right\} \tag{2.65}$$

$$\exp\left(\frac{h}{2}\hat{D}\right)\psi\left(z,\mathbf{X}\right) = F_T^{-1}\left\{\exp\left(-ih\frac{\mathbf{k}\cdot\mathbf{k}}{2}\right)\hat{\psi}\left(z,\mathbf{k}\right)\right\} \tag{2.66}$$

where F_T denotes the FFT operation.

The nonlinear sub-step can be evaluated in direct space using the formula

$$\psi^{NL}(z+h/2,\mathbf{X}) = \exp\left(ihF(|\psi|^2,\mathbf{X})\right)\psi(z+h/2,\mathbf{X}) \tag{2.67}$$

which completes the Split-step Fourier method (SSFM). It is important to notice that both the linear and nonlinear sub-steps are computationally solved in a discretized grid. This discussion about the discretization and the vectors $\mathbf{X}$ and $\mathbf{k}$ concludes that both equations (2.66) and (2.67) can be done by point-to-point calculations. In summary the SSFM algorithm is as follows:

$$\begin{aligned}
\hat{\psi}(z,\mathbf{k}) &= F_T\left\{\psi\left(z,\mathbf{X}\right)\right\} \\
\psi\left(z+h/2,\mathbf{X}\right) &= F_T^{-1}\left\{\exp\left(-ih\frac{\mathbf{k}\cdot\mathbf{k}}{2}\right)\hat{\psi}\left(z,\mathbf{k}\right)\right\} \\
\psi^{NL}(z+h/2,\mathbf{X}) &= \exp\left(ihF(|\psi|^2,\mathbf{X})\right)\psi(z+h/2,\mathbf{X}) \\
\hat{\psi}^{NL}(z+h/2,\mathbf{k}) &= F_T\left\{\psi^{NL}\left(z+h/2,\mathbf{X}\right)\right\} \\
\psi\left(z+h,\mathbf{X}\right) &= F_T^{-1}\left\{\exp\left(-ih\frac{\mathbf{k}\cdot\mathbf{k}}{2}\right)\hat{\psi}^{NL}\left(z+h/2,\mathbf{k}\right)\right\}.
\end{aligned} \tag{2.68}$$

Before concluding this section we shall notice three important features of the SSFM. First, in a usual problem the interest is not to do only a single integration step but several of them. In that situation, it can be shown that, except for the first step, we need only to compute one linear sub-step per integration step instead of two. In fact, the First Same As Last [62] property allows to concatenate the linear sub-steps as follows:

$$\psi\left(z+2h,\mathbf{x}\right) \approx \exp\left(\frac{h}{2}\hat{D}\right)\exp\left(h\hat{N}\right)\underbrace{\exp\left(\frac{h}{2}\hat{D}\right)\exp\left(\frac{h}{2}\hat{D}\right)}_{FSAL}\exp\left(h\hat{N}\right)\exp\left(\frac{h}{2}\hat{D}\right)\psi\left(z,\mathbf{x}\right)$$

$$= \exp\left(\frac{h}{2}\hat{D}\right)\exp\left(h\hat{N}\right)\exp\left(h\hat{D}\right)\exp\left(h\hat{N}\right)\exp\left(\frac{h}{2}\hat{D}\right)\psi\left(z,\mathbf{x}\right). \tag{2.69}$$

Thus, considering several integration steps, the cost of this second order method is basically the same of the first order method.

Secondly, the calculation of the linear sub-step, the most time consuming step of the method, is done by using the FFT. If the dimensions $\mathbf{N}$ are all powers of 2, this method has a computational cost of order $O\left(N_{tot} log N_{tot}\right)$ where $N_{tot} = N_x \cdot N_y \cdot N_t$. As most FD methods rely in $O(N_{tot}^2)$ matrix operations, SSFM are usually faster, especially for multidimensional or large systems.

Last but not least, the SSFM is conservative and normally admits larger integration steps than the FD methods for obtaining the same accuracy. In fact, the only situation where FD are preferable over SSFM is when the system is small and the dynamics of the envelope is fast, introducing a limitation to smaller integration steps [2].

The SSFM is not the faster neither the most accurate method for every situation [2, 25]. However, it constitutes the best performance GNLSE solver for the majority of the problems. Thus, it is our choice for the implementation of a high performance GNLSE solver using GPU computing.

2.5.3 Boundary conditions for the SSFM

An important aspect of any solver of the GNLSE are the boundary conditions. In general, a soliton like solution of the GNLSE extends well beyond the limits of the simulation box even though in that portion of space the amplitude of the field can be very close to zero, and therefore negligible. Also, it is possible that the solitons propagate to close proximity (and scatter through) to the boundaries of the simulation box.

The boundary represents a discontinuity in the simulation box and can interact with the soliton-like pulses yielding diverse, and many times unwanted, effects. These may include reflection and numerical dispersion, depending on the type of the solver being developed. Therefore, great care must be put in addressing the boundary-field interaction. This is specially important when considering problems where the medium is supposed to be infinite or the relevant interaction is restricted to a small region of space after which the field evolves into far regions, as occurs during soliton scattering. If no attention is put to boundary conditions then extremely large simulation boxes must be used, which are costly in terms of computational resources, efficiency and simulation times.

To avoid these problems, several types of numerical solvers for the GNLSE have been developed which control the physics of boundary-field interaction, namely allowing to produce periodic, reflective and absorbing boundaries.

Periodic boundary conditions allow that, when the field reaches one of the boundaries

of the simulation box, then it emerges from the opposing boundary. This implies that a two dimensional simulation box corresponds topologically to a torus. This type of boundary condition arises natively from the SSFM algorithm given its calculation of the linear step in the Fourier space.

Reflecting boundary conditions force the field that reaches one of the boundaries to bounce back. This type of boundary condition can be implemented by dividing the simulation box into two domains, one at the center where the simulation occurs and another corresponding to a thin layer of points, as shown in figure (2.4). The idea is to replace the original GNLSE with an altered version, say

$$i\frac{\partial\psi}{\partial z} + \frac{1}{2}\nabla_\perp^2\psi + F(|\psi|^2)\psi + V_r\psi = 0 \tag{2.70}$$

where

$$V_r = \begin{cases} v_r & \text{at the boundary layer} \\ 0 & \text{elsewhere} \end{cases} \tag{2.71}$$

where v_r is a real-valued constant larger than any of the other terms in the GNLSE.

The absorbing boundary conditions correspond to the case where a field reaching the boundaries of the box is totally (or almost totally) absorbed and disappears from the simulation box. This boundary condition is specially indicated to simulate the propagation of solitons in infinite or very large domains.

Again this boundary condition is obtained by replacing the original GNLSE with

$$i\frac{\partial\psi}{\partial z} + \frac{1}{2}\nabla_\perp^2\psi + F(|\psi|^2)\psi + iV_a\psi = 0 \tag{2.72}$$

where V_a is a positive real valued function, chosen to maximize the absorption of the radiation. The most common absorbing potentials are the Gaussian [62]

$$V_a = Ae^{-\frac{(\mathbf{x}-\mathbf{x}_0)^2}{2w_0}} \tag{2.73}$$

and the hyperbolic tangent [67]

$$V_a = A\left(1 + \tanh\left(w_0\left(r - \mathbf{x}_0\right)\right)\right). \tag{2.74}$$

Parameters A, w_0 and $\mathbf{x}_0$ are characteristic of the absorbing potential whose choice depends on the problem and must be optimized to maximize the absorption of the outgoing radiation.

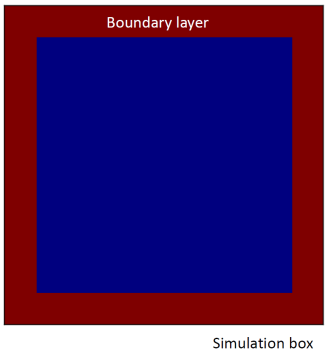


Figure 2.4: Visual scheme of the simulation box, describing the concept of boundary layer. The blue interior of the box is where the field evolves.

2.6 Concluding remarks

By the end of this chapter we have presented some of the most important aspects of the structure, solutions and numerical methods of the GNLSE. Particularly, the introduction of the SSFM provides the cornerstone for the next chapter, where we describe the implementation of our GNLSE solver based on GPU computing, GASE. In the following chapter we discuss how the SSFM can be adapted to operate in a GPU architecture, taking to account the allocation of data into the memory available. Also, we address how the resources of a computer can be used in heterogeneous programming to boost code efficiency.

3 Implementation of the GPU-based GNLSE solver

In the previous chapter we reviewed the basic framework of the GNLSE, including the type of equations and situations that have been studied, the analytical methods used, and some of the most relevant numerical methods developed to solve it. In this chapter we focus on the implementation of a GNLSE solver based on a SSFM using GPU computing. We start by analyzing the advantages and disadvantages of this approach, especially when compared with CPU-based computing, which constitutes the most common base approach in the past. The core of this chapter is devoted to the description of the algorithm used, its computational implementation and to its performance analysis. As it shall be shown, we have obtained computational speedup factors of almost 100 when compared to the same CPU-based solver, demonstrating the high potential of GPU computing for numerical analysis of the GNLSE.

3.1 Simple problem, high computational time

The calculation of solutions of the GNLSE in systems with dimensionality higher than (1+1)-d and specially, if it involves complex geometries and higher order nonlinearities, is a very large computational problem. This is mostly due to the large number of points of the spatial mesh used to sample the field and the local optical properties, which need to be determined and computed on each time step. Using serial programming, such as used in single core CPUs, the solution of this type of problems requires vast running times as most calculations must be done sequentially for a very large set of sampling points of the field. To illustrate the immense challenge at hands, consider a simple problem, consisting of the investigation of the dynamics of a supergaussian (2+1)-d soliton in a cubic-quintic media limited in a small square domain with side length a having a small circle - we will call it a *hole* - of linear material at the center, with diameter of value $a/25$, as shown in figure 3.1. Consider also that the soliton has a characteristic size of $a/4$ and

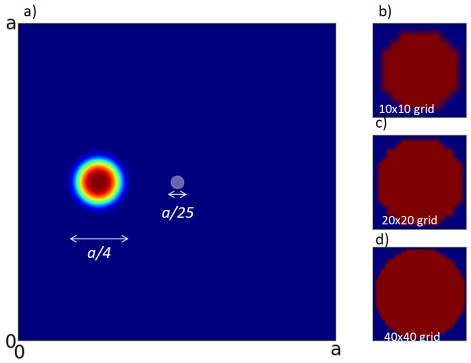


Figure 3.1: a) Image description of the problem of a supergaussian state colliding with a small hole of linear material. b) Representation of the circle for various mesh sizes.

scatters with a small angle towards the hole. Such problem has no known analytical solutions, including those obtained using perturbation methods (section 2.4.2), which leaves numerical simulations as the only choice of analysis method.

Numerical simulations rely on the adequate discretization of the simulation box which is determined with the spatial scales of the problem. For the problem in figure 3.1 the discretization mesh size is determined by the need to preserve the circular shape of the linear defect. Figure (3.1) shows that this requires using at least 40 points per diameter of the object. Since 40 discretization points are equivalent to $a/25$, then the domain is described by at least a grid of 1000x1000=1 million points.

In a modern 3 GHz processor the multiplication by an exponential factor - equivalent operation to the nonlinear step of the SSFM - of a vector of 1 million points lasts at least 2 seconds. In our problem we have three different nonlinear steps to do, two relative to cubic and quintic nonlinearities and other relative to the refraction index hole, giving a computational time of 6 seconds per nonlinear step. If we are interested in seeing what is the behavior of the soliton with small velocities interacting with a sequence of holes, we must integrate the problem by a larger number of steps, say 100 000. Considering all these results, the total running time of the simulation would greatly exceed a week. Now imagine that we want three or four other simulations with different wire sizes and that along the way we may have to do some intermediate tests. The simulation would then be impossible to solve using ordinary computers.

Many of the recent research in solitons considers problems such the one previously discussed. Using serial programming to solve them on a computer would involve prohibitive running times. The alternative is to use parallel programming in a computer

cluster and doing the calculations simultaneously on several interconnected computers instead of sequentially on a single computer. The idea is to break up the original domain into smaller chunks or sub-domains and doing the calculations for each of them in a separate computer. The best about this is that the process is scalable, which means that increasing the number of sampling points of the domain mesh can be solved by just adding more computers to the cluster. Unfortunately, in reality things are not so simple, in fact they can be even worse than using a single machine. At every integration step the computers need to exchange information about the values of the field at the boundaries of their sub-domains. This data transfer is usually even slower than the computational process and thus clusters are not a solution. Also, good clusters are extremely expensive.

GPU computing is a recent trend that seems to solve both of these problems. Modern GPUs have thousands of cores connected directly to the same device RAM memory, what seems to minimize the memory transfer problems while still having the possibility of the parallelization of the simulation over many chip cores. Also, they are cheap devices, which cost almost the same price of modern CPUs.

The only flaw is that the cores of GPUs are not as powerful as CPUs. GPUs cores have peak operation frequency of 1GHz, which is three times slower than the normal 3 GHz processors. CPU and GPU are the result of two distinct strategies to achieve higher computing power. Typically, CPUs result from an approach pursued over the last decades to attain high computing frequencies and through that, reduce the time needed to perform each operation. Unfortunately, this strategy has been so successful that it has reached the limit between classical and quantum physics. As computer electronics became smaller and computer clocking became faster, the physics of the electrons swirling through the computer wiring starts to exhibit quantum features and some severe problems of thermal dissipation. As a result, this approach has reached an efficiency barrier.

The increasing demand for devices capable to do massive computing, specially from the game industry, has led to a different approach based on parallel programming, using clusters of simple cores integrated in a single device, the GPU. The idea is to optimize not single operations but the overall computing time. For example, it is more interesting to minimize the overall computational time of 1 million pixels than to minimize the time of computing a single pixel. This new paradigm appears to be the new trend in high performance computing, replacing high frequency computing with a high throughput computing paradigm.

Many of the GPU cards are developed for running computer games with higher perfor-

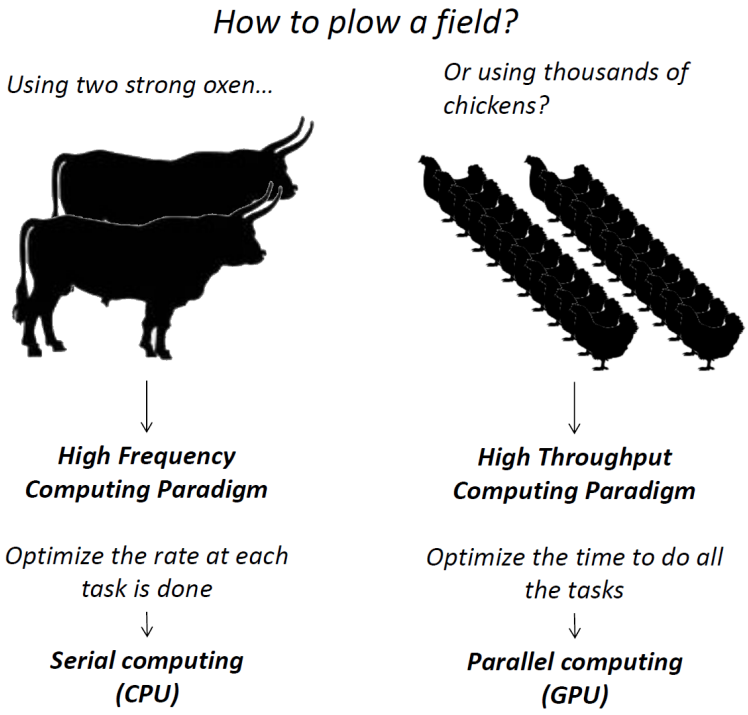


Figure 3.2: Conceptual comparison between two different computing paradigms, the high frequency and the high throughput computing paradigm.

mance graphics and rely on the same algebraic operations needed to do many scientific calculations. As a result they constitute a recent and almost untapped resource in computational physics. The following section describes some of the aspects, challenges and results of using this technology to develop the high performance solver of the GNLSE.

3.2 How to plow a field?

Seymour Cray, for many the father of supercomputing, always resisted to massive parallel computing as a credible solution for better computational performances. To sustain his opinion he even joked once saying "If you were plowing a field, which would you rather use: Two strong oxen or 1024 chickens?". This question reflects the conceptual problem of using the two different paradigms. If you put some numbers to Cray's problem then you realize that if each ox is able to yield a force power of 150 kg while each chicken yields 0.5 kg, then, as surprising as it might have been to Cray himself, chickens would win!

In this analogy the oxen represent the CPU, powerful and able to plow data quickly,

and chickens represent GPUs and other forms of distributed computing, slower but in larger number. Of course the problem of high performance computing is not as simple as this analogy seems to suggest. The answer to Cray's challenge depends on the size of the field to be plowed (the analogy is a good one and can be used to explain many of the ideas, so we keep it for now). If the field is small, the oxen are better because they are able to plow quickly, but if the field is very large, then the combined power of the chickens and their ability to do work simultaneously and independently (this is called *concurrence*) can be preferable.

If you know chicken, you know how hard it would be to organize and coordinate a brood of chickens to plow a field. With GPUs the problem is the same, and it is necessary a computer model and the corresponding software to make the many GPU cores to work as a unit to solve a numerical problem. The development of the CUDA by NVIDIA answered this problem. CUDA is an extension to C/C++ for GPU computing using NVIDIA devices. This extension allows programmers to access GPU memory and compute capabilities. A normal code is composed by a part to be performed in the CPU (*host*) and a part that is performed in the GPU (*device*).

Although CUDA can help in coordinating the operations of different GPU cores, it is limited to basic algebraic operations and it is not the ideal tool for scientific programming. Being closer to machine code than scripting languages, such as Python and MATLAB, it allows better control of machine operations but yields more complex and extensive coding that require longer development, programming and debugging times. Therefore, for GPU computing to be used for scientific calculations in an efficient way, it would be necessary to develop numerical packages and libraries similar to *Lapack*, *Blas*, *Scipy* (Python libraries) and others. These libraries use serial programming so the challenge is to adapt the functions to make full use of parallelization capabilities of GPUs. Such packages and libraries are currently being developed and new progresses are being made available to computational physicists every day. Fortunately, such packages already include efficient versions of the numerical tools necessary to implement a GNLSE solver.

In particular, in the solver developed during the preparation of this dissertation, the following numerical packages were used, both of them developed by NVIDIA and included in the most recent CUDA toolkit (version 5.0) :

- *Thrust* is a library that provides a collection of parallel data operations that allows the transformation and operation between vectors. Thrust also defines the vector containers and can work both in the CPU and GPU depending if the vector is

in the *host* or in the *device* memory, respectively, and permit to make complex vector operations with a high-level of abstraction, selecting automatically the most efficient parallel implementation.

- *Cufft* is a library that allows the user to apply one, two or three dimension FFT transformations of vectors in the GPU. It is the analog of *Fftw* library for CPU and it only works in the graphics card.

Even though much of these packages facilitate the development of the GNLSE solvers and allow a good level of abstraction from what is actually occurring in the hardware during the calculations, the way the code is written is not independent of the parallelization in the hardware.

In fact there are three aspects that need to be considered when writing a GNLSE solver to work in massively parallel GPUs: the fact that GPUs are designed to operate mainly with single precision floating point numbers, the existence of different types of memories distributed in the graphics card that have to be used adequately, and portability and compatibility issues of the code developed.

This factors are addressed in the three following sections.

3.3 Gaming vs Scientific precision

The GPUs have long surpassed the performances of the CPUs. The performance can be measured in the number of floating point operations that can be done in a second, a unit named FLOP/s. A typical modern CPU is capable of no more than 100 GFLOP/s while GPUs are now reaching TFLOP/s performances. But these peak performances are only obtained in single precision.

As mentioned, the main application of massive parallel programming using GPUs is the acceleration of the graphics in computer games. For this reason, many of the calculations need only to be performed using floating point numbers with single precision. However, for scientific calculations, where it is mandatory to maintain numeric errors under control during the computations, the use of double precision numbers is preferable, if not necessary. The most recent devices and numerical packages developed for GPU already allow to use double precision numbers. But as GPUs are not optimized for this type of work, the immense single precision performance is strongly reduced to more modest values when double precision is used. Yet, the peak performance of GPUs at double precision is still better than those of CPUs and even CPU clusters, and a speedup

Type of the card	Release year	Model	Bandwidth (GB/s)	GFLOP/s (Single)	GFLOP/s (Double)	Memory (Mb)	Core Clock (MHz)	# Cuda cores	Release price (€, approx.)
Consumer GPUs	2007	8800 GTS	68	624	-	512	650	128	500
	2008	9800 GTX+	70	705	-	512	738	128	500
	2009	GTX 285	159	1062	133	1024	648	240	550
	2010	GTX 480	177	1344	168	1536	700	480	450
	2011	GTX 580	192	1581	198	1536	772	512	600
	2012	GTX 680	192	3090	129	2048	1006	1536	500
	2013	GTX 780	288	3977	166	3072	863	2304	500
	2013	GTX TITAN	288	4500	1500	6144	836	2688	1000
Pro GPUs	2012	TESLA K20X	250	4591	1530	6144	732	2688	6000

Table 3.1: Comparison between different top-of-the-line GPUs of the NVIDIA consumer line Geforce. A model from the Fermi line of a professional computing dedicated GPUs is also presented. An increasing computational power is notorious over the years, as well the increase of chip memory and memory bandwidth. It is well patented that evolution of GPUs will reach another level in the next few years, with the new high performance Geforce Titan setting the pace.

can still be obtained. For example, the calculation of a multiplication by an exponential of a vector of 1 million points has a speedup of 90 in single precision and 40 for double. In this case a speedup is still obtained and the reduction in speedup is not so dramatic as predicted. That is due to the fact that usual operations are frequently limited by the *bandwidth* of the memory, i.e., the transfer rate of data from the memory to the processing unit.

To understand the concept of bandwidth we focus on the simple example of the sum of two vectors. For the Intel I7 processor, the memory bandwidth is 25,6 GB/s which means that the system can only transfer 25,6 GB of data from memory to the CPU during one second of operation. Considering that a double precision float occupies 8 bytes of memory, processing units have access to less than 3500 million of floats per second. Considering that a sum has to read two values and store one, the bandwidth limit the peak performance of a sum to $3500/3 \approx 1.16$ GFLOP/s far away from the predicted peak performance of 98,78 GFLOP/s. Thus, the bandwidth of the GPU is a determinant factor in achieving higher performances in numerical experiments.

Although the need to use double precision in scientific calculations can diminish significantly the true power of the GPUs, it is important to notice also that these are recent developments in these technologies. Not only GPUs are becoming faster with ever-increasing number of cores (reaching almost 3000 cores) but also the developments that allow to do double precision calculations are quite recent (less than 4 years), still in their infancy. Therefore, although there is still a long way to go, scientific computation

based on GPU appears to have a bright future ahead.

3.4 Some memories must be kept closer than others: memory considerations for GPU computing

Other key aspect of hardware that is necessary to account when programming is memory allocation. In more detail, different types of data needed for the calculations must be stored in distinct types of memory of the GPU architecture, if optimal (or close to optimal) performance is to be achieved.

Inside the GPU card there are three types of memory, with different access times:

- Shared memory: small amount of memory (48 kB) with fast access but not accessible by all cores in a GPU;
- Constant memory : small amount of memory (64 kB = 8192 double precision floats) with fast access and accessible by all cores. The speed makes it the ideal to store constant data, such as physical constants used by the solver.
- Global memory: big amount of memory (few GB, see table 3.1) with slow access and accessible by all cores. Its size makes it ideal to store the field distribution and the nonlinearities.

As mentioned, global memory, the RAM of the device, is the memory used to store fields. It should be noticed that the amount of memory of the GPU is fixed from factory and cannot be augmented afterward, unlike the CPU, where one can always add larger RAM memories. This can limit the capabilities of the GPU in larger simulation systems. Generally, for each GB of RAM memory, 2^{22} double precision numbers can be stored and operated in the GPU. Hopefully, future GPU models will have increasing memory that will make GPUs more capable of addressing larger simulations.

Finally let us give a word about data storage, namely about saving the numerical data produced during calculation in the GPU to a more permanent storage, in this case a hard disk, for later analysis. The transference of data between different groups of cores in the GPU is fast (the bandwidth concept mentioned in the last section) resulting in a short latency time between two time steps of integration of the GNLSE. However, transferring that information from the GPU to be stored in an external hard disk is time consuming, and must be avoided during the calculations. This problem will be addressed in more detail in the next sections.

3.5 Compatibility issue

The use of GPUs for general purpose computing, and more specifically for scientific computing, is something new in computer sciences and is a part of a broader concept called heterogeneous computing. To put it simple, heterogeneous computing aims to use all the resources available in a machine (GPUs, CPUs or others) in an integrated way to do massive computing. The early precursors of this concept included the engineers of NVIDIA, one of the most important graphics cards company and the developer of CUDA. Although CUDA is currently perhaps most advanced platform for heterogeneous computing, it operates only on specific GPU cards from NVIDIA. There is some irony in this fact: heterogeneous computing has a main goal in promoting portability between different devices but CUDA and programs written in CUDA can only be used in hardware from a specific manufacturer. Therefore, our code is not portable to machines that have other types of GPU cards. To address this limitation, an alternative to CUDA is being developed in recent years by a consortium of GPU card and CPU manufacturers (especially AMD) called OPEN CL. The idea is that OPEN CL can become a standard language for heterogeneous computing capable of operating in any type of device.

Unfortunately, OPEN CL is still in its earlier versions and even though it can already compete with CUDA in terms of managing parallel computation in distinct devices, it still lacks numerical packages that support scientific computing. This short-come of OPEN CL is expected to be overcome in future years as it becomes more used and software developers bridge the gap between it and CUDA. It is not possible to consider OPEN CL to support the development of a GNLSE solver yet, however the structure of GASE should (in principle) be easily transposed from CUDA as OPEN CL reaches later stages of development.

3.6 Implementation of the GNLSE solver

This section is devoted to explain the general structure of the numerical code developed during this dissertation's research.

3.6.1 Outline of the code

The code is basically composed of three parts, each corresponding to a specific stage of the calculation as shown in figure 3.3.

The first part encompasses the initialization of the data structures that store the data. These include the lists containing the spatial coordinates, the value of the field and the optical properties and parameters in each point of the mesh of the simulation box. Within these structures are also included the data necessary to implement the boundary conditions.

During the second part of the code the actual numerical calculations necessary to integrate the GNLSE are performed using the SSFM. In a very simple way, it consists in a loop which repeats the integration step. This is the most time consuming part of the code and the duration of each step will be discussed in the following sections. During this second stage, the results are stored into the hard disk to be used in the third and last stage of the code where the data is analyzed. In order to reduce both the memory requirements of the hard disk and the latency of the process of transferring data from the memory to the hard disk, data is only stored after doing an user-defined number of integration steps. While the first and second part are written in C++ with CUDA, this last part of the code is written in Python (instead of CUDA) by reasons of convenience, since not only Python allows to produce graphics with high quality but also is much easier to program, given that it is a scripting language.

It should be noticed that CUDA and the associated numerical toolkit allow us to do most of the calculations both in the GPU and CPU without significative changes to the structure of the code. In order to compare our solver GASE with a CPU version of a GNLSE solver we adapted and developed also a CPU solver using the same structure and the *Fftw* library for the Fourier transforms. Therefore, and to benchmark the GPU computations relative to the CPU version, this code is prepared to operate in both platforms with specifications that we discuss in the following section.

3.6.2 Integration step routine

One of the most important features of GASE is that it can address systems with higher dimensionality, i.e., (2+1)-d and (2+1+1)-d domains (although it can be easily extended for higher dimensions if necessary). This implies the discretization of the domain into a regular mesh of points on which the field and the optical properties of the medium must be evaluated. In principle, these values could be stored in multidimensional lists of data (similar to tensors) but this is not the most efficient way to use data in GPU computing. In fact, the numerical libraries built on CUDA and used to develop GASE operate only with one dimensional lists, including the numerical library which takes care of the FFT. Therefore, the multidimensional list must be spanned into a single

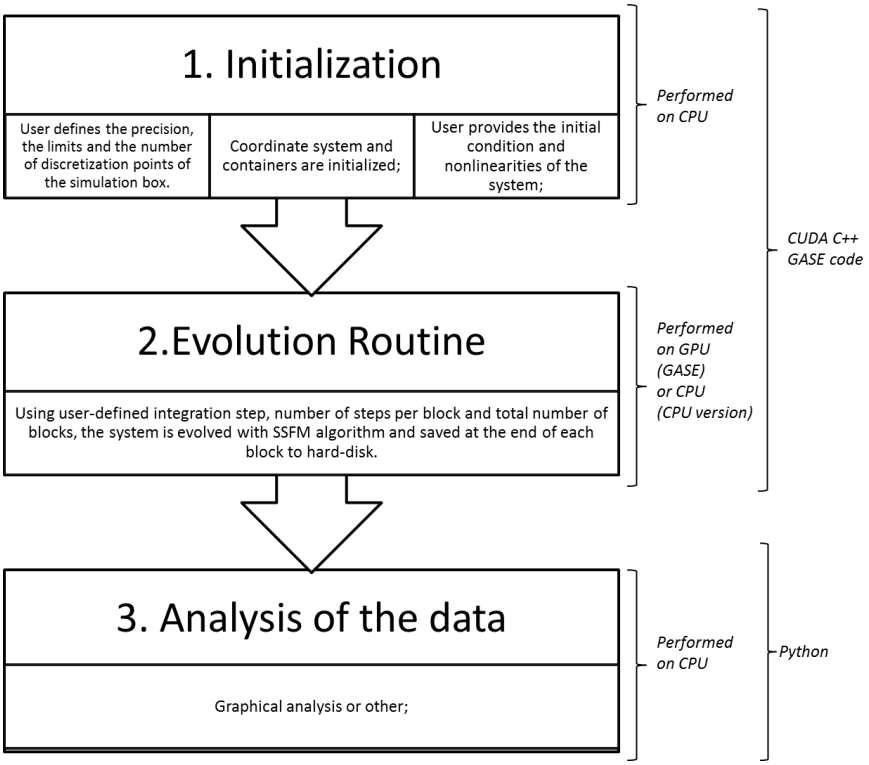


Figure 3.3: Succinct description of the code structure. The code is divided in three parts: the first initializes the data and the simulation box, the second is the integration routine and the third is the post-simulation analysis of the data.

one dimensional list (which is quite simple to do). More importantly, it is necessary to convert the identifying index of the data in the multidimensional list (j,k,l) into the index in the one dimensional list $I = j + k \times N_x + l \times N_x \times N_y$. The conversion between the index I to (j,k,l) is easily done with the help of *floor* and *mod* functions, using the expressions

$$\begin{aligned} j &= I \mod (N_x) \\ k &= \text{floor}\left(\frac{I}{N_x}\right) \mod (N_y) \\ l &= \text{floor}\left(\frac{I}{N_x N_y}\right) \end{aligned}$$

In the simulation mesh, this allows to compute the spatial coordinates of the point to which the data pertains to. For example in three dimensional mesh, coordinates are obtained as $\mathbf{X} = (j\Delta x, k\Delta y, l\Delta t)$. Also, having this methodology in mind, the nonlinear step can be easily calculated from expression (2.68).

The FFT routine transforms the field 1D lists with the field data $\Psi(I)$ into another, say $\hat{\Psi}(I)$, where each element corresponds to a specific spatial frequency. According to the documentation of *Cufft*, the relation between true index of the list I and the corresponding spatial frequency is given computing (j,k,l) and then using the correspondence formulas (here shown for k_x but the same is extended to the other dimensions)

$$k_x = \begin{cases} \frac{2\pi j}{N_x \Delta x} & \text{, for } 0 \leq j \leq \frac{N_x}{2} \\ \frac{2\pi (j - N_x)}{N_x \Delta x} & \text{, for } \frac{N_x}{2} < j < N_x \end{cases}$$

allows us to compute the Laplacian of the field and then computing the linear step using expression (2.68).

The second stage of the code is basically a loop which operates consecutive linear and nonlinear steps of the SSFM on the field data and is depicted in figure . However, it should be noticed that these steps are grouped *(blocks)* into sequences of about a few hundred steps *(steps per block)* during which no data is registered in the hard disk. As explained before, the transference of data from the GPU to the disk is time consuming and if done after each step it would eat away the performance of doing parallel computing. Also, the data on GPU cannot be transferred directly to the hard disk. Rather, it is first copied to the RAM of the host and then it undergoes some simple processing in the CPU. Namely, the field intensity and phase are computed from the complex field

amplitude profile. Only then, the data is transferred from the RAM to the hard disk. Together, the computation of field intensity and phase and the transfer of the data from the RAM to the hard disk, constitute a very slow process. This would kill some of the performance of the GPU code so a new feature was added to GASE, allowing to perform this operations in the CPU while GPU is already and simultaneously running the next step of the SSFM. It is important, but not mandatory, to choose sequences of SSFM steps sufficiently long to give time for the CPU to finish its task, but short enough to allow a good insight of the evolution of the field. As a result of using the CPU for part of the workload, the GPU is relieved from part of the calculations and from latency times during data storage, resulting in faster computing processes as a whole. In fact, this is a good example of heterogeneous computing where all the resources of the machines are used to promote overall efficiency.

3.6.3 Code features

Before ending this chapter it is important to make a synthesis of the capabilities of GASE. Therefore, summarizing all the work developed, GASE is currently able to perform integrations of any given GNLSE in (1+1)-d, (2+1)-d and (2+1+1)-d geometry, with any user given initial condition. The nonlinearities are chosen by the user and might be of arbitrary power and can be either a constant number or a spatial distribution. The nonlinearities can also be given by a function to have a dependance on the propagation distance.

All of these systems can be simulated in a simulation box either using periodic, reflective or absorbing boundary conditions. Moreover, there is also a recent and still under development, feature that allows the simulation of two coupled GNLSE. Therefore, GASE is a powerful tool and we believe it to be ready to investigate the majority of the most state-of-the-art problems in solitons and in GNLSE subject.

3.7 Concluding remarks

In the beginning of this chapter we have introduced a problem and conclude that simulating such system would imply very high computational running times, which make the problem very hard to investigate. We proceed trying to develop a new tool to address the problem and presented the GPUs as a recent technology that could be useful for such systems.

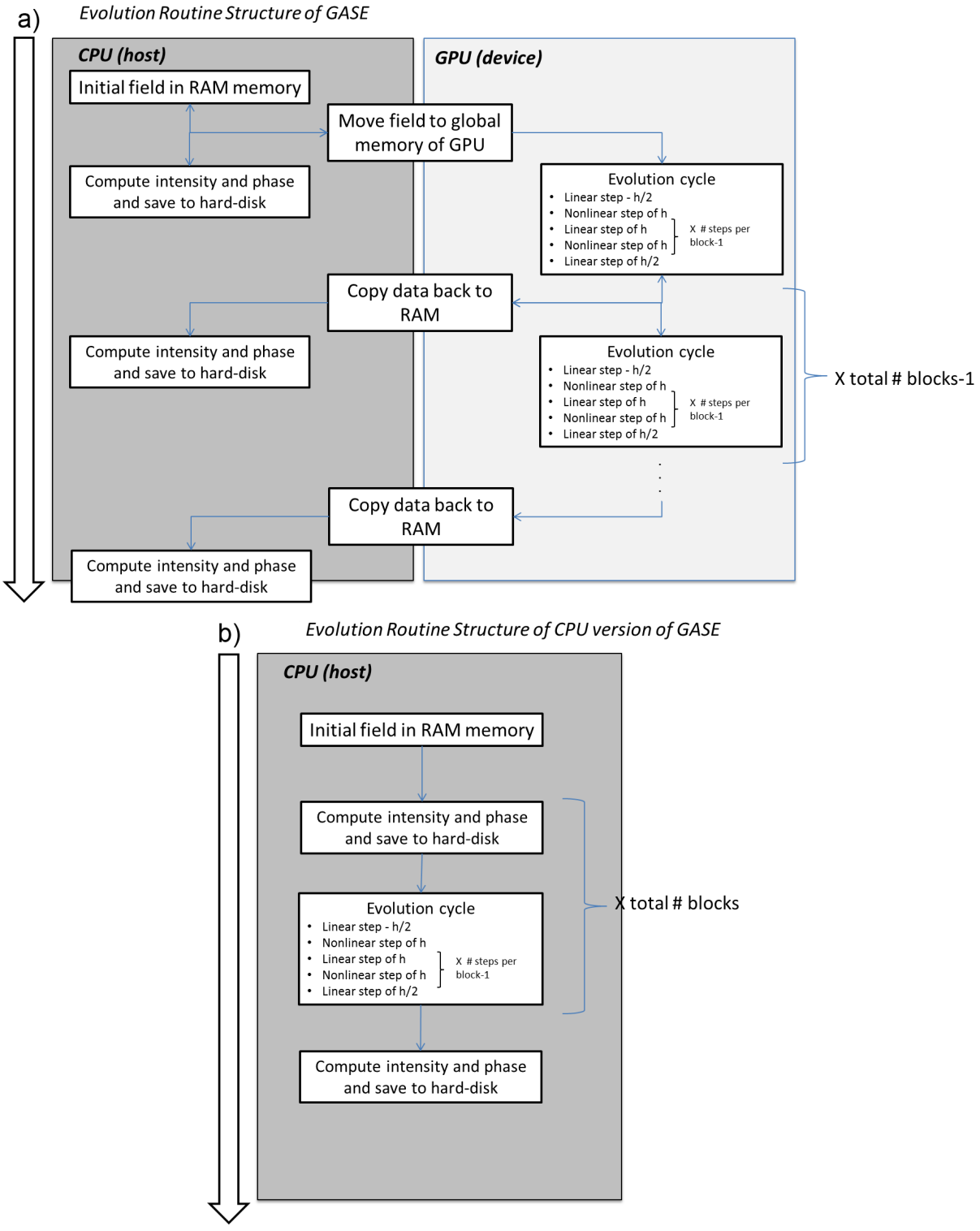


Figure 3.4: Integration routines for both versions of the solver. Figure a) shows the structure of the evolution routine for GASE, where it is possible to see the additional memory transfer needed but also the parallel structure of computations running both in CPU and GPU. Figure b) describes the integration procedure for the CPU version of the solver.

After a small discussion on GPU computing framework, we described succinctly the structure of GASE, the GNLSE solver based on GPU computing developed during this dissertation. Full details were not given as this description is intended to be simple to the reader, avoiding to enter in the complicated world of CUDA programming.

At the end of this chapter an obvious question arises: what is the speedup, if any at all, that GASE can achieve?

To answer the question we recover the initial problem introduced at the begin of the chapter. The propagation of a supergaussian of order $m = 1.9$ with parameters described in section 2.4.1 was done using GASE. In conformity with the initial description of the problem, shown in figure (3.1), we choose a simulation box with limits $[0, 120] \times [0, 120]$. Also, the center hole has a radius of 2.4 and refractive index $n = 0$. After some initial numerical tests we found that the system could be solved in single precision using an integration step of $h = 0.02$. The total number of integration steps used is 100 000.

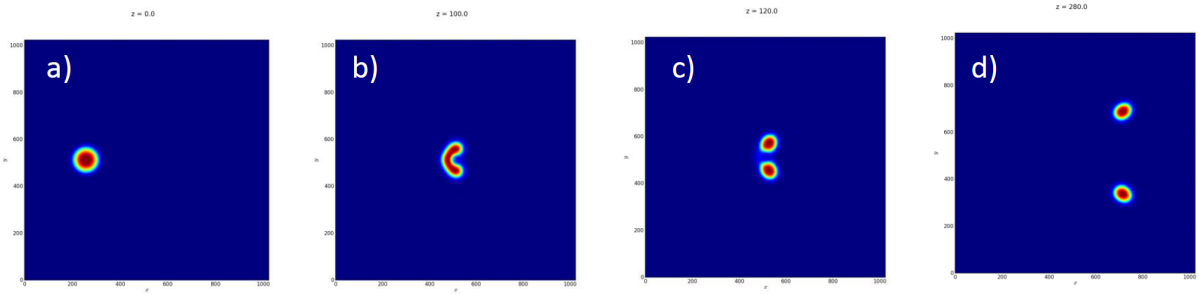


Figure 3.5: Sequence a)-d) shows a collision of a supergaussian state with velocity $\mu = 0.3$ with a hole of radius 2.4. The light state emerges as two smaller and low intensity beams after the scattering. (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation)

A series of simulations were performed and results are presented in figures (3.5 - 3.9). The physics of the results are interesting as they resemble a collision of a drop of liquid with a circular object, which is not unexpected given the liquid behavior of high power supergaussian states [58]. Also, the results change depending on the initial velocity and the hole radius, and a plethora of different behaviors is achieved. For the initial hole radius, a soliton with velocity $\mu = 0.15$ rebound on the hole, but with higher velocity, such as $\mu = 0.3$ and $\mu = 0.5$, is decomposed in two smaller and lower intensity light beams. Diminishing the hole radius to 2.0, a light beam with a velocity $\mu = 0.3$ is momentarily divided into two low power light beams but regroup as one after the collision with the hole. A smaller hole of 1.0 seems even to not affect the propagation. These results are interesting but are not our main goal in dissertation. Instead, they

illustrate the power of the developed GNLSE solver.

As a matter of fact, we have not performed only the 5 presented simulations but a series of simulations that, including the earlier investigations and those which were wrongly set up, took more than one day to run in our computer. An initial comparison between GPU and CPU versions of the solver show that GASE runs the problem almost 100 times faster than the CPU version. The conclusion is obvious: if we had chosen to investigate this problem in a normal CPU then it would have taken almost 100 days, a third of the duration of this dissertation.

But how fast can we go and for which problems? The answer is addressed in the next chapter, where we perform the benchmarking of GASE.

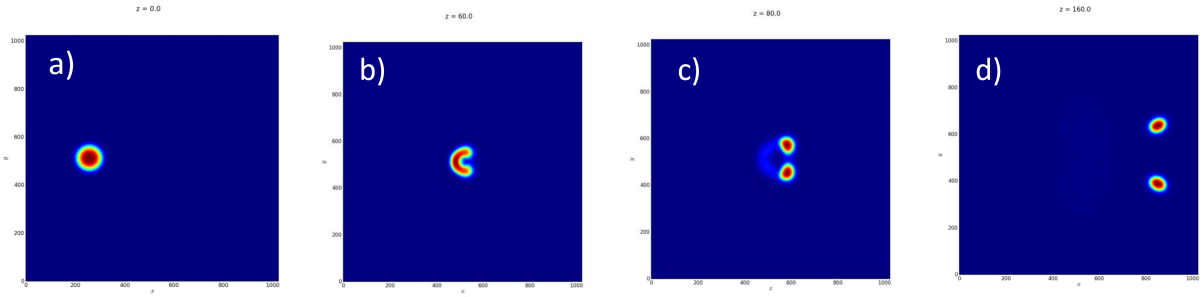


Figure 3.6: Sequence a)-d) shows a collision of a supergaussian state with velocity $\mu = 0.5$ with a hole of radius 2.4. The light state emerges as two smaller and low intensity beams after the scattering, with a different angle than the situation with velocity $\mu = 0.3$. (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation)

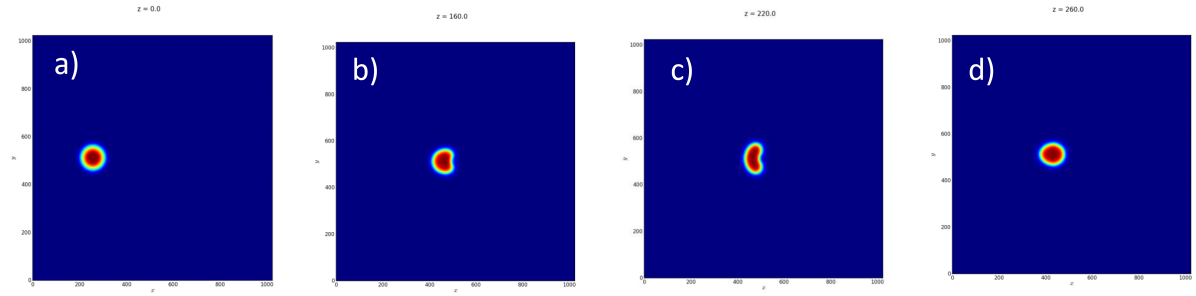


Figure 3.7: Sequence a)-d) shows a collision of a supergaussian state with velocity $\mu = 0.15$ with a hole of radius 2.4. The light state collides and is reflected by the hole. (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation)

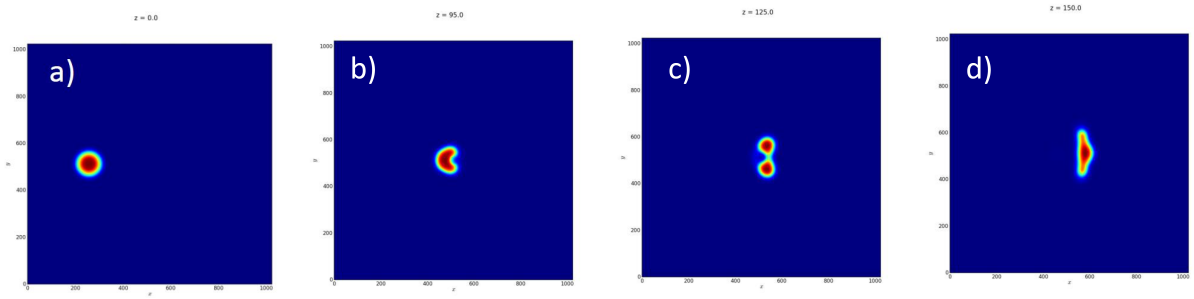


Figure 3.8: Sequence a)-d) shows a collision of a supergaussian state with velocity $\mu = 0.3$ with a hole of radius 2.0. The light state after an intermediate division in two light states collapses again in one high power state. (*Pdf version only* - click twice on sub-figure a) for a clip of the simulation, where can also be seen that the state became trapped between two consecutive holes, that we simulate in the same box using periodic conditions)

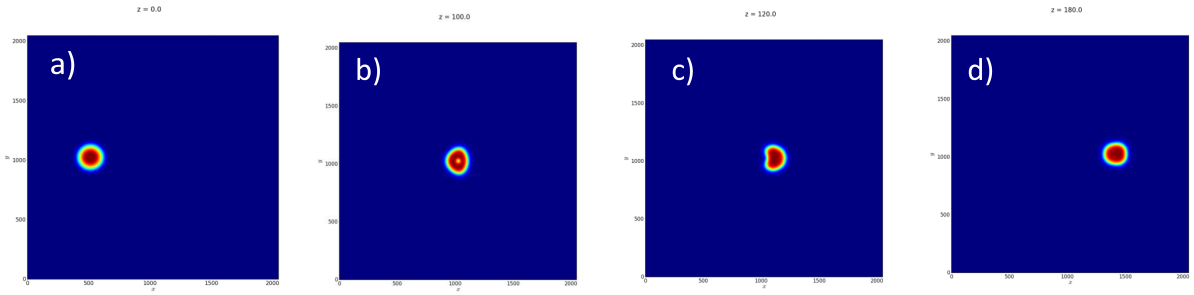


Figure 3.9: Sequence a)-d) shows a collision of a supergaussian state with velocity $\mu = 0.3$ with a hole of radius 1.0. The light state emerges almost as if not been scattered. (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation)

4 Benchmark of GASE

During the last chapter the implementation of GASE and the features and specifications of the code were succinctly described. This chapter is focused on testing and benchmarking of the code, analyzing the obtained speedups in comparison with the CN algorithm and with CPU-based version of the code. These results, that are the central and most critical results of this dissertation, show that the GPU is many times faster than the CPU version of the code, which demonstrates the utility and efficiency of the GASE, especially for multidimensional and complex systems.

4.1 Validation of the method

A crucial step in the development of numerical codes for physical simulations is their validation. The most direct methodology is to use the codes to simulate cases where there are known analytical solutions and compare them. As discussed in chapter 2, only the (1+1)-d NLSE admits analytical solutions using IST method, so our accuracy tests are restricted to solitons in Kerr media. The soliton solutions of the NLSE for this system have the form

$$\begin{aligned} \psi(x,z) &= 2\nu \mathrm{sech}\left[2\nu\left(x-\bar{x}_0-\mu z\right)\right]\exp\left\{i\mu\left(x-\bar{x}_0\right)+i\delta(z)\right\} \\ \delta(z) &= (2\nu^2-\mu^2/2)z+\delta_0, \end{aligned} \tag{4.1}$$

where ν is the amplitude, $\bar{x}_0$ is the initial position of the centroid of the field distribution ψ, μ is the transverse velocity and δ_0 is the initial phase of the soliton. In order to analyze the accuracy of the simulations, the relative error can be computed

$$i_2 = \frac{\int_{-\infty}^{\infty}|\psi(x,z_{final})-\psi_n(x,z_{final})|^2dx}{\int_{-\infty}^{\infty}|\psi_n(x,z_{final})|^2dx},$$

where ψ_n and ψ are the numerical and analytical solutions of equation (2.1) respectively.

The code was validated considering the evolution of the solitons with different initial

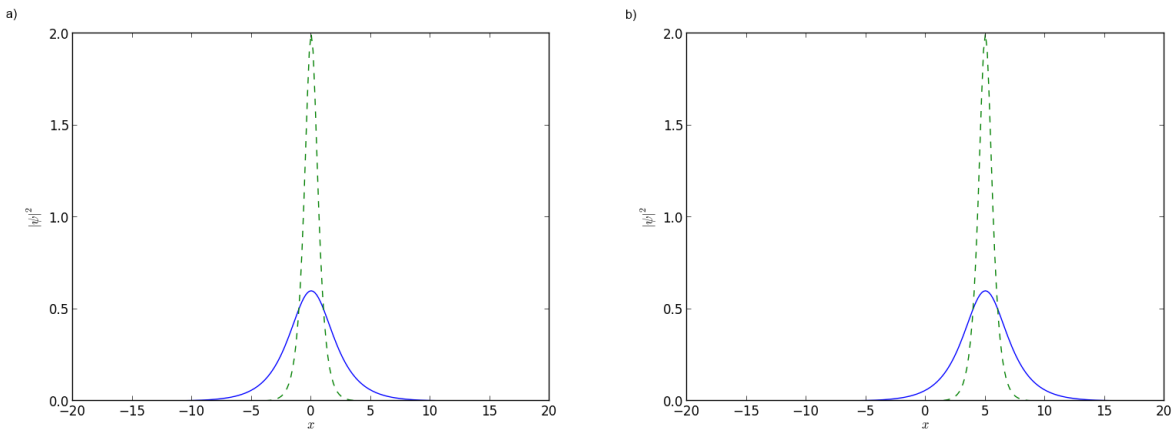


Figure 4.1: Initial conditions (a)) and analytical final state at $z_{final} = 100$ (b)) of the high power - dashed line - and low power - solid line - solitons.

conditions, as shown in figure (4.1). The first is a low power soliton with $\nu = 0.3$ while the second is a high power soliton with $\nu = 1$. Both have a small initial velocity $\mu = 0.05$ and were initially centered at $x = 20$. The simulation box corresponds to $x \in [0, 40]$ and is discretized in a grid of N points. The evolution is computed using an integration step h until reaching the final propagation distance $z_{final} = 100$, where the data is retrieved to hard disk in order to perform the error analysis.

In table (4.1) are presented the numerical errors as a function of the spatial discretization along the x dimension, introduced by the variation of the number of points N of the simulation grid. It is shown that, the case of low power soliton needs lower spatial definition for getting the same accuracy as the high power soliton, which is explained by the steeper variation in amplitude of the second soliton. When the number of points is $N = 2^8$ or bigger, i.e., spatial discretization around $\Delta x \lesssim 0.2$, we obtained very accurate results for both solitons.

Using a grid with $N = 2^8$ points, the influence of integration step h in the accuracy of simulations can be investigated. Results are presented in table (4.2) and show again that the higher power soliton requires a smaller integration step for same accuracy results. It is then shown that optimal spatial discretization is problem dependent, but that for solitons with similar powers to those considered, good results can be obtained using a spatial discretization with step $\Delta x = 0.2$. Optimal integration step h is also dependent on the problem and must be investigated case by case, comparing the results obtained to those expected.

The results presented in this section are quite simple and might seem accessory but

N	i_2 ($\nu = 0.3$)	i_2 ($\nu = 1$)
2^5	0.016	1.98
2^6	1.5e-8	1.97
2^7	1.4e-8	0.11
2^8	1.4e-8	1.9e-8
2^9	1.4e-8	1.9e-8
2^{10}	1.4e-8	1.9e-8
2^{11}	1.4e-8	1.9e-8
2^{12}	1.4e-8	1.9e-8

Table 4.1: Error analysis for the simulations with fixed integration step $h = 0.01$ and variable number of points, which introduces a variable discretization Δx.

h	i_2 ($\nu = 0.3$)	i_2 ($\nu = 1$)
1.0	0.00025	1.6
0.5	6.4e-6	1.4
0.1	1.9e-8	0.9
0.05	1.3e-8	0.04
0.01	1.4e-8	1.8e-8

Table 4.2: Error analysis for the simulations with fixed number of points $N = 2^8$ and variable integration step h.

they are still important as they constitute a validation of the method and the code, showing that GASE is accurate. Also, the information of tables (4.1) and (4.2) allows us to have a prediction of the necessary discretization and integration step for future simulations.

4.2 Benchmark of the method: SSFM *versus* CN

In chapters 1 and 2 it was mentioned that SSFM usually outperforms the CN algorithm for majority of simulations of the GNLSE. Having access to a code developed by Paulo Alcino, a GPU version of the CN scheme, it was possible to do a direct comparison of both accuracy and speedups performances of the two algorithms. The test simulation is done for the (1+1)-d NLSE, similar to that performed in last section, with soliton amplitude $\nu = 0.3$ but with final propagation distance $z_{final} = 1000$. The simulations were performed in a simulation box with $x \in [0, 40]$ described by a grid of $N = 2^8$ points.

Results of computational runs are presented in table (4.3) for various integration steps h. Direct analysis shows that not only SSFM is far more accurate than CN, allowing

h	i_2 SSFM	i_2 CN	time CN (s)	time SSFM (s)
0.1	1.9e-8	0.8	0.5	0.4
0.05	1.3e-8	0.8	1.0	0.9
0.01	1.4e-8	1.8e-2	6.0	2.0
0.005	1.0e-8	2.3e-3	64.1	5.1
0.001	1.0e-8	1.0e-4	132.6	28.1

Table 4.3: Accuracy and performance comparison between GASE solver, based on SSFM, and the CN solver. It can be easily seen that GASE outperforms in every aspect the CN method.

to carry out simulations with bigger integration steps and thus reduced computational times, but also that the code developed is faster than the CN code even for the same integration step, contrary to what was expected [2]. This might be related either with the low performance of iterative methods in GPUs comparing to Fourier transforms, or with implementation problems in the CN code.

Also, during early investigations, it was noted that CN algorithm is neither conservative nor reliable, as the convergence of the solution is difficult and relies on the right choice of an iterator. Then, it can be concluded that SSFM is the adequate choice between the two of them for a high performance solver of the GNLSE using GPUs.

4.3 Benchmark of the code: GPU *versus* CPU

After some initial tests, the attention is now focused on comparing GASE with the CPU version of the same code. Before advancing, the methodology used must be discussed. In this section we compare results for computational times obtained in the desktop GPU, GTX 660 Ti, the laptop GPU, GT 640M, and the desktop CPU, Intel I7 3770K, with specifications given in table (4.4). With these results we compute the speedup of GASE comparing the performance of each GPU with the CPU running time. Speedup tests were done for (1+1)-d, (2+1)-d and (2+1+1)-d systems. The (1+1)-d tests were done in a simple cubic nonlinear media using *sech-shaped* solitons. For the (2+1)-d and the (2+1+1)-d case were used *supergaussian* solitons as approximate stable soliton solutions. To avoid the soliton collapse for high dimensions it was considered that the media also has a quintic nonlinearity.

	# cores	Core Clock (GHz)	GFLOP/s Peak	Bandwidth (GB/s)	Memory	Price (approx. 2/2013)
CPU Intel I7-3770K	4	3.9	98.78	25.6	(RAM)	300€
GPU Geforce GTX660 Ti (Desktop)	1344	1.0	2459	144.2	2GB DDR5 (on-chip memory)	280€
GPU Geforce GT 640M (Laptop)	384	0.6	480	64.0	1GB DDR5 (on-chip memory)	200€

Table 4.4: Specifications of both the two GPUs and the CPU used during the benchmarks.

4.3.1 (1+1)-d speedup results

For the (1+1)-d results we ran a series of simulations of bright *sech-shaped* soliton, equal to the one used in section 4.1. The final propagation distance is also $z_{total} = 1000$ and the simulation box has a variable number of points N with constant spatial discretization $\Delta x = 0.2$, which defines the limits of the simulation box. The solution is only retrieved to the hard disk at the final propagation distance, which is not the usual situation since in most cases it is important to keep track of the evolution of the solution at intermediate integration steps. However, for performance testing this is not necessary. Also, the results obtained constitute a lower bound to performance speedup of GASE relative to other codes since it benefits from the improved performance in managing the storage of data to disk, as discussed in section 3.6.2.

Computational running times and speedups for GPU and CPU versions of the code are shown in figure (4.2) and table (4.5). For keeping the computational times under control, different integration steps h were chosen and consequently different number of total integration steps were performed for obtaining the same final propagation distance. Then, for comparing the computational times, the variable time per step, that is just the total computational time for each simulation divided by the total number of steps performed, is introduced. Results obtained show increasing speedup with increasing number of points, with best results showing a speedup of 35 for $N = 2^{17}$. Also, even the low end laptop GPU still outperforms the CPU, with a speedup factor over 7 for sufficiently large systems.

To understand the implications of the speedups, imagine that we need to define a big simulation box with $x \in [0, 830000]$. With a $\Delta x \approx 0.2$ discretization, at least a grid of $N = 2^{22}$ points is needed. Table (4.5) shows that such simulation runs in 48.4 seconds in the GPU, while it takes more than 26 minutes to run in the CPU. To

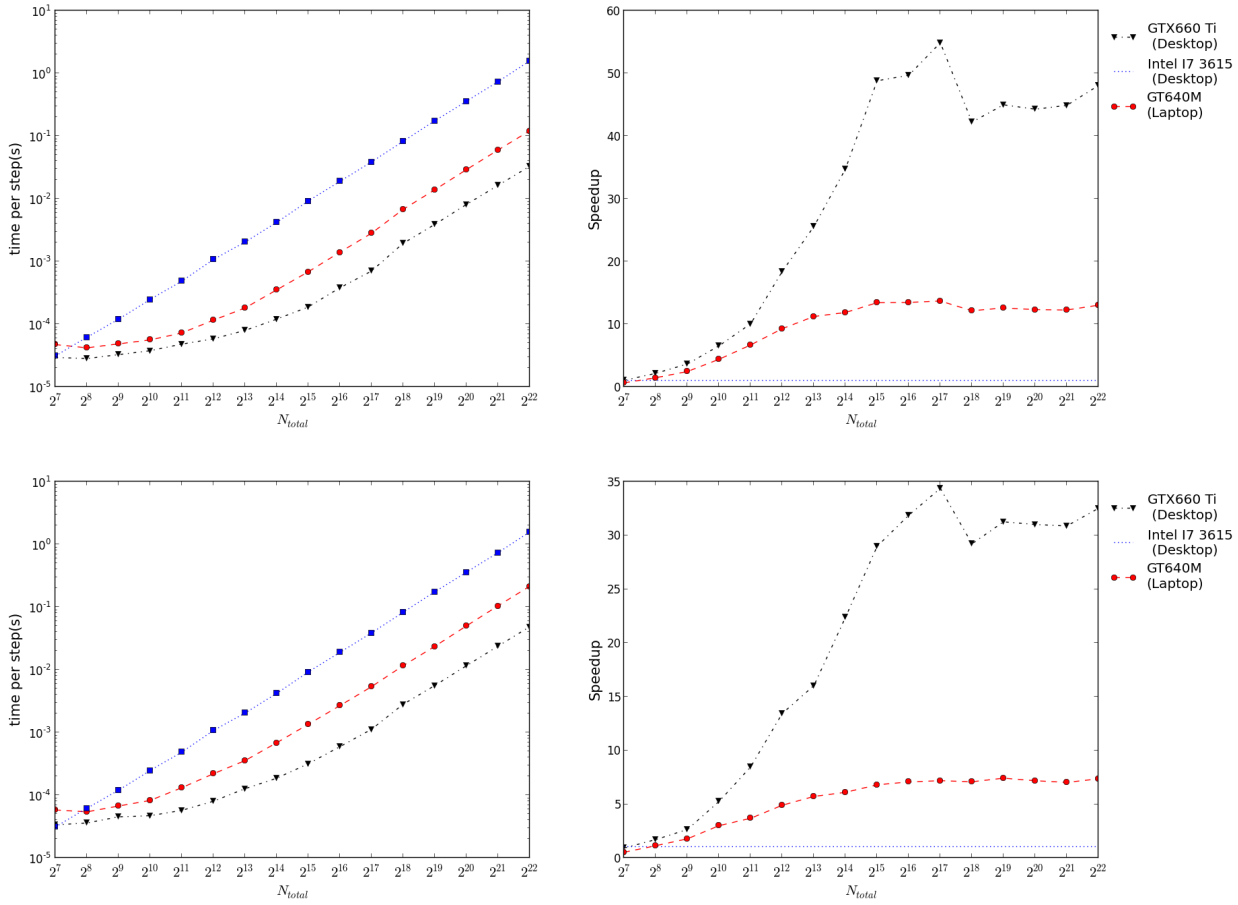


Figure 4.2: Single (top) and Double (bottom) precision benchmarks for simulations of the (1+1)-d NLSE, with the results for computational time per step (left) and the corresponding speedup in comparison with the CPU version of the code (right).

Number of Steps	h	N	GPU (s)	CPU (s)	Speedup
100 000	0.01	2^8	3.4	6.2	1.8
10 000	0.1	2^{12}	0.8	10.8	13.5
1 000	1.0	2^{18}	2.8	82.1	29.3
1 000	1.0	2^{22}	48.4	1575.6	32.5

Table 4.5: A collection of results for simulation times and speedup of the solver for the (1+1)-d NLSE using double precision, for both GASE (running in the desktop GPU) and the CPU-based version of the solver.

understand the performance of this code in comparison with another code written in a scripting language, we developed a similar version of the code in Python. For the same simulation, Python took more than 11 days, more than 20 000 times slower than our GASE code, which provides a demonstration that Ron Caplan's NLSEmagic [12] is strongly limited by using a scripting language.

It is noticeable that the performance when using double precision is reduced to less than a half when compared with single precision performance. This was expected but still good speedup results are obtained. Single performance can be used for some problems but should be used carefully, and only after being sure that results obtained are the same to those obtained using double precision. Simulations involving many integrations steps should be done using double precision, in order to keep numerical errors under control.

4.3.2 (2+1)-d speedup results

For the (2+1)-d speedup tests we choose to run a simulation of a cubic-quintic medium described by the GNLSE of equation (2.2) with $F(|\psi|^2) = |\psi|^2 - |\psi|^4$. For the initial condition, it is used a supergaussian of order $m = 1$, with parameters and form given by the equations introduced in section (2.4.1). The final propagation distance is considered $z_{total} = 10$ and the integration step is now fixed at $h = 0.01$. The simulation box has a constant spatial discretization $\Delta x = \Delta y = 0.2$ and is defined over a grid of $N_x \times N_y$ points, corresponding to the number of the mesh points in the x and y dimension, respectively. Again, speedup results are analyzed with the help of a time per step variable, as done for the (1+1)-d case. Running times and speedup results are shown in figure (4.3) and table (4.6) and again significant speedups were obtained, reaching in double precision a top speedup factor of just over 40. For example, for a computational grid with $N = 2^{11} \times 2^{11}$ points, GPU took just one minute to solve the system while CPU took more than 40 minutes. If we consider a simulation that took a day to solve in GPU

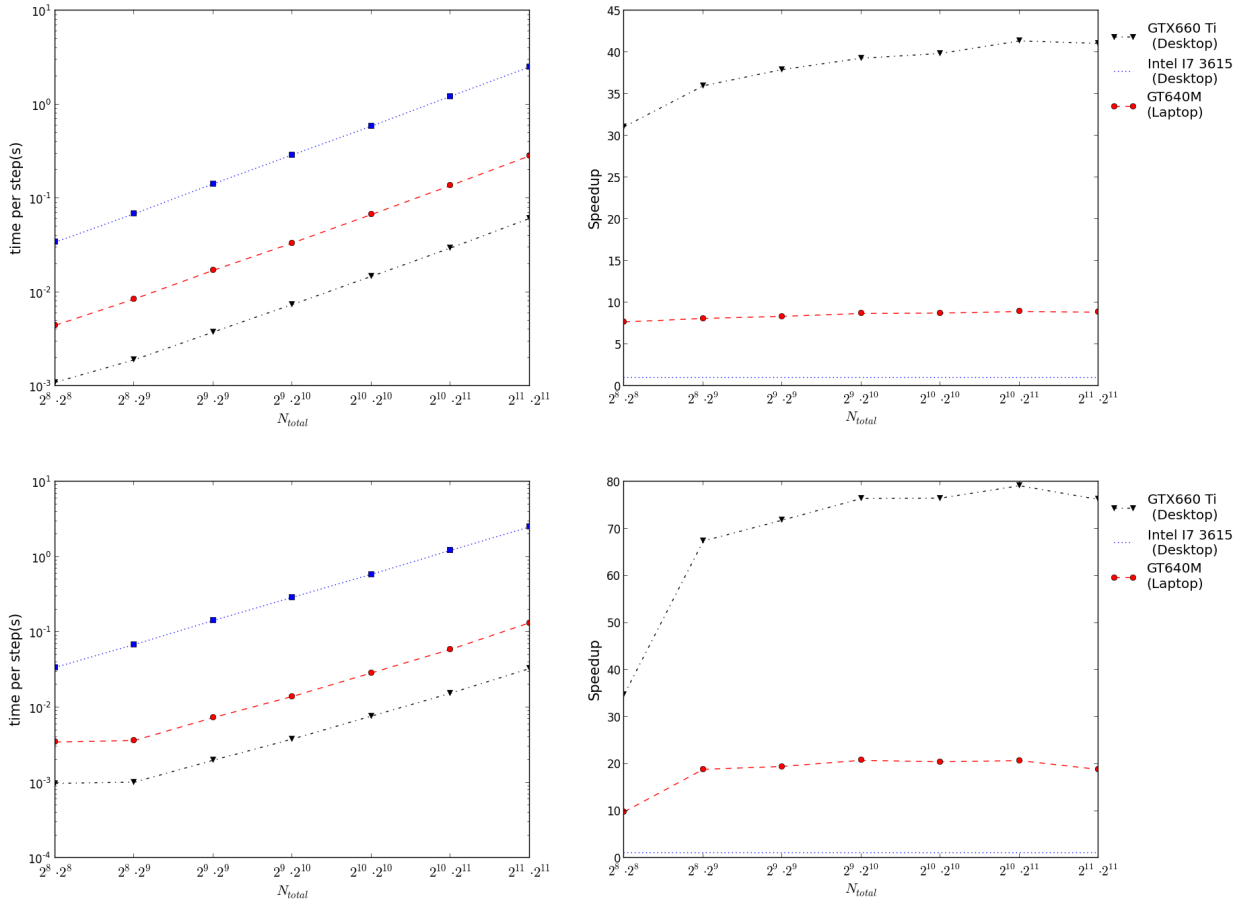


Figure 4.3: Double (top) and single (bottom) precision benchmarks for simulations of the (2+1)-d GNLSE for a cubic-quintic media, with the results for computational time per step (left) and the corresponding speedup in comparison with the CPU version of the code (right).

- just consider for example the same mesh and final propagation time of $z_{total} = 14400$ - solving in the CPU will took more than one month, which is a prohibitive duration for the majority of research.

It is important to note that these results are even better than the (1+1)-d case. This is related to the fact that now we consider two nonlinear terms and the system is then more complex. This becomes more clear with the results of figure (4.4), corresponding to the same initial value problem but now solved in a simulation box with boundary conditions as introduced in section 2.5.3. For this it is used an absorbing potential $V_a = (1 + \tanh(r - L))$, with L the limits of the simulation box . Speedup factors for this problem show a top value of over 80 which is really significative.

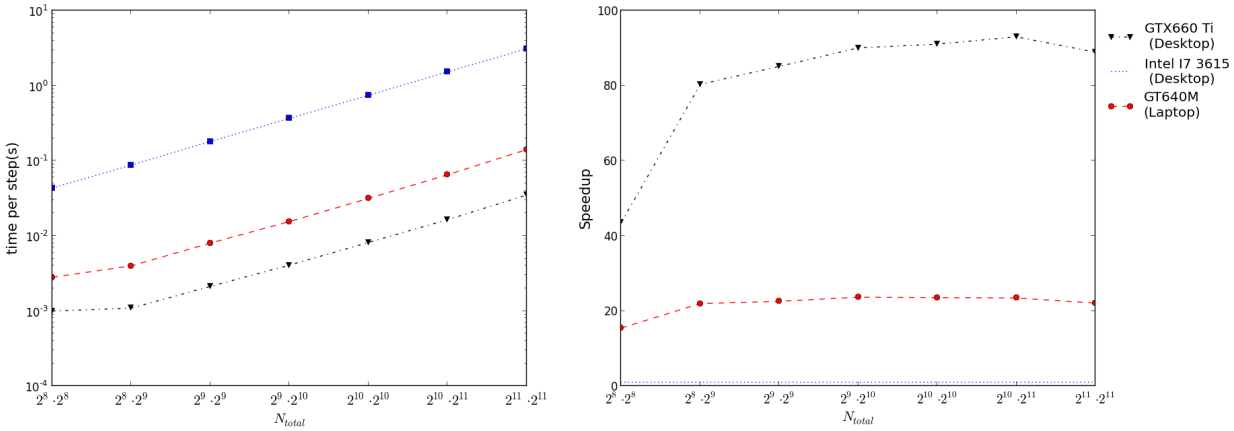


Figure 4.4: Single precision benchmarks for simulations of the (2+1)-d GNLSE for a cubic-quintic media with absorbing boundaries, with the results for computational time per step (left) and the corresponding speedup in comparison with the CPU version of the code (right).

Number of Steps	h	N	GPU (s)	CPU (s)	Speedup
1 000	0.01	$2^8 \times 2^8$	1.1	34.1	31.0
1 000	0.01	$2^9 \times 2^9$	3.8	142.9	37.6
1 000	0.01	$2^{10} \times 2^{10}$	14.7	587.5	39.9
1 000	0.01	$2^{11} \times 2^{11}$	61.0	2506.5	41.1

Table 4.6: A collection of results for simulation times and speedup of the solver for the (2+1)-d GNLSE for a cubic-quintic media, using double precision, for both GASE (running in the desktop GPU) and the CPU-based version of the solver.

As obtained for (1+1)-d, figure (4.3) shows that single precision simulations are twice faster than double precision, with overwhelming speedup factor of almost 80. In (2+1)-d cases is common to be interested in evolving the system much less steps than the (1+1)-d problems. Therefore, single precision can be used for some cases but again, this must be done with caution.

4.3.3 (2+1+1)-d speedup results

For the three dimensional simulations, or as called before, the (2+1+1)-d GNLSE, we use a supergaussian state of order $m = 1$ as one in the last section. The final propagation distance is considered $z_{final} = 10$ and the integration step is fixed as $h = 0.01$. We use the discretization of the space as before, $\Delta x = \Delta y = \Delta t = 0.2$ and define a simulation box represented by the mesh $N_x \times N_y \times N_t$.

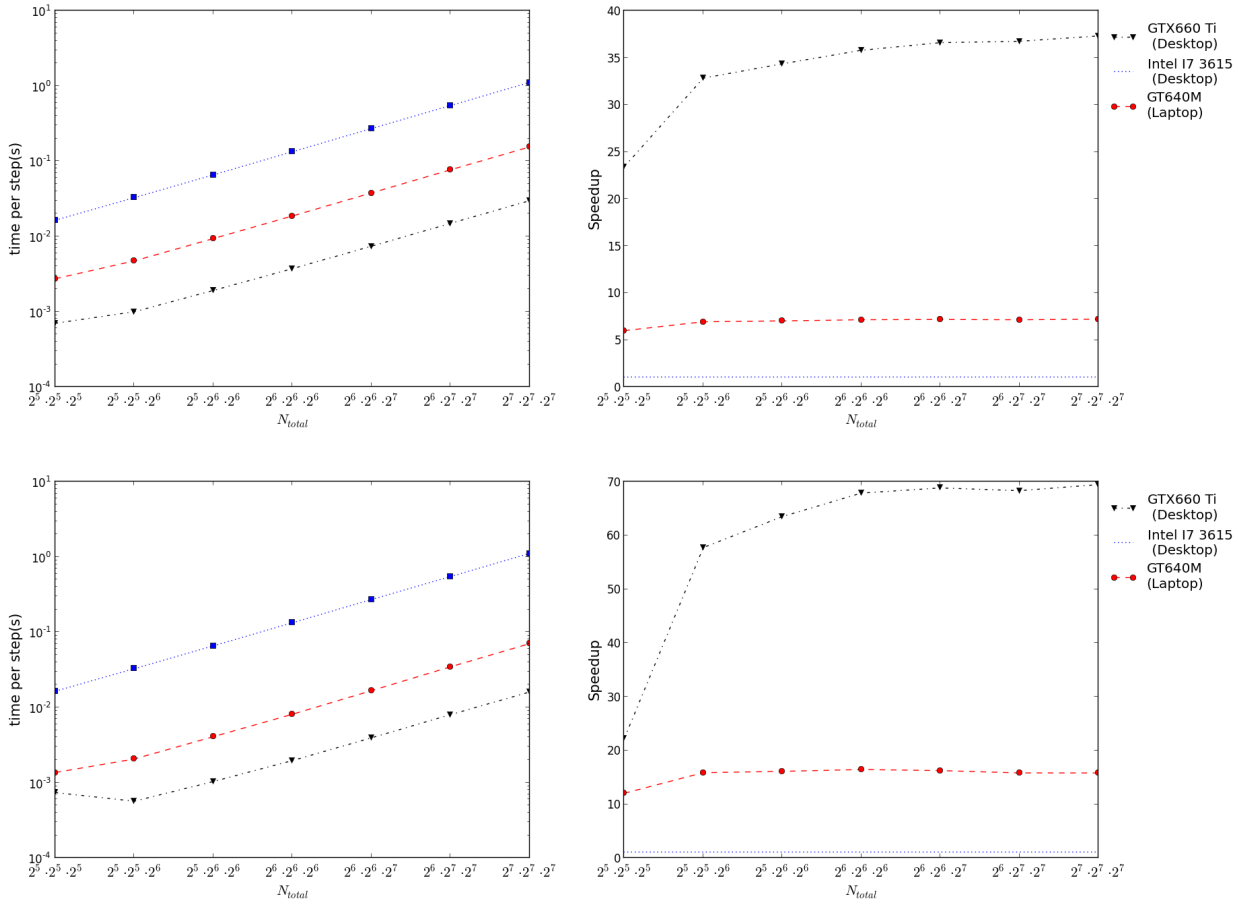


Figure 4.5: Double (top) and single (bottom) precision benchmarks for simulations of the (2+1+1)-d GNLSE for a cubic-quintic media, with the results for computational time per step (left) and the corresponding speedup in comparison with the CPU version of the code (right).

Number of Steps	h	N	GPU (s)	CPU (s)	Speedup
1 000	0.01	$2^5 \times 2^5 \times 2^5$	0.7	16.4	23.4
1 000	0.01	$2^6 \times 2^6 \times 2^6$	3.7	133.5	36.1
1 000	0.01	$2^7 \times 2^7 \times 2^7$	29.8	1112.7	37.3

Table 4.7: A collection of results for simulation times and speedup of the solver for the (2+1+1)-d GNLSE for a cubic-quintic media, using double precision, for both GASE (running in the desktop GPU) and the CPU-based version of the solver.

Results for speedups compared with serial CPU performances are shown in figure (4.5) and table (4.7). It can be observed that speedups are slight smaller than those obtained for (2+1)-d case, which is possibly related with the lower performances of the GPU for computing Fourier transforms in three dimensions. Still, a considerable speedup factor of over 35 is obtained for the larger simulation boxes. For single precision, speedup factor is almost twice of that obtained for the double precision, which as said in sections before, was already expected.

4.4 Concluding remarks

Along this chapter we presented the tests done to validate the GNLSE solver developed during this dissertation - GASE. We started by comparing the results obtained with GASE with the exact solution, in situations where an analytical solution exists. Afterward, we proceed to a series of tests intended to benchmark the performance of GASE in comparison with the CPU version developed. We have obtained maximum speedups of around 80 for single precision and around 40 for double precision. Also we have observed that increasing the dimensionality of the system or the complexity leads to a larger speedup factor. Therefore, the results of this chapter are important as they demonstrate the realization of the goal of this dissertation: the development of a high performance GNLSE solver, capable of addressing efficiently problems in multidimensional or complex systems, using only low-cost computers and based on GPU computing.

5 Case study 1: Lightons: phonons with Light

In the present and the following chapter we present two case studies used to evaluate the potential of our implementation of a GNLSE solver based on GPU computing. These case studies have not been fully explored in terms of scientific analysis for two reasons. First, because the main objective of this dissertation is the development of a GPU based GNLSE solver, which is by itself very challenging and required much effort. Secondly, each of these topics can constitute a subject of a master thesis on their own. However, we already present many and interesting new results.

In this chapter we analyze a chain of interacting solitons which can support collective excitations similar to phonons in chains of masses. This problem is exceptionally difficult to address using conventional SSFM implemented on a single CPU due mainly to the large number of discretization points and integration steps required to simulate several dozens of solitons and investigate the continuous limit of this excitations. Moreover, we also present results for (2+1)-d chains of solitons, not just (1+1)-d, which truly demonstrate the potential of GPU computing.

5.1 Motivation

At a fundamental point of view, soliton appear as stable solution of nonlinear wave equations, such as the Nonlinear Schrodinger equation (NLSE), balancing wave dispersion with nonlinear wave interaction, and resulting in a coherent wave package that can behave both like a particle and a wave.

In some sense, solitons bridge the gap between waves and particles. Being a wave phenomenon in its fundamental aspects, solitons can behave like a particle by avoiding the effects of dispersion and keeping most of its intensity confined to a small region of space. Also, like particles, in some situations they can scatter each other as if they where solid blocks of light.

Just as waves can exhibit particle like behavior, particles can also support waves, such as mechanical waves. For example, a chain of interacting particles can display collective oscillatory motions which depend on the nature of the interaction forces. Elastic linear interaction allows small displacements of the particles from their equilibrium positions to form mechanical waves, known as phonons, that propagate throughout the chain. As waves, phonons can not only propagate but also superpose and interfere.

From this duality, where nonlinear wave packages display particle behavior and particles can support waves, arises a question: can solitons aligned in a chain support mechanical waves similar to phonons?

In this chapter, we focus our attention in the collective oscillations of a 1-dimensional chain of N optical solitons, supported by a medium with a cubic nonlinearity as described by the NLSE. We investigate the nature of these oscillations using GASE and extend the models existing in the literature not only by identifying its limitations but also by presenting illustrations of their extension to (2+1)-d systems.

5.2 Physical model

The evolution of optical solitons in a media with cubic non-linearity is governed by the (1+1)-dimensional NLSE

$$i\frac{\partial\psi}{\partial z}+\frac{1}{2}\frac{\partial^2\psi}{\partial x^2}+|\psi|^2\psi=0 \tag{5.1}$$

where z is the propagation distance and x is coordinate along the transverse direction. The single soliton solution for this equation is

$$\psi_j(x,z)=2\nu_j\text{sech}\left\{2\nu_j\left(x-\bar{x_j}\right)\right\}\exp\left\{i2\mu_j\left(x-\bar{x_j}\right)+i\delta_j\right\} \tag{5.2}$$

where the parameters ν_j, $\bar{x_j}$, μ_j, and δ_j refer to the amplitude, the position, the frequency and the phase of the soliton, respectively.

We consider the limit of large distance and small overlap between consecutive solitons, where the N-soliton solution for the equation (5.1) can be given as a sum of N single soliton solutions

$$\psi(x,z)=\sum_{j=1}^{N}\psi_j(x,z) \tag{5.3}$$

When replacing (5.3) into equation (5.1) it becomes clear that the nonlinear term is responsible for the interaction between solitons. To solve the resulting equation, we use an *ansatz* and separate equation (5.1) into the following N coupled equations, one for

each single soliton:

$$i\frac{\partial\psi_j}{\partial z}+\frac{1}{2}\frac{\partial^2\psi_j}{\partial x^2}+\left|\psi_j\right|^2\psi_j=-R_j\left[\psi\right] \tag{5.4}$$

It is simple to see that the sum of the solutions of each of these equations is a solution of equation (5.1). Each of these equations is just a NLSE with a small perturbation R_j associated to solitonic interaction. Considering that only first neighbors can interact and expanding the first order in the overlap $\mathcal{O}\left(\psi_j\psi_{j+1}^*,\psi_j\psi_{j-1}^*\right)$ we get

$$R_j\left[\psi\right]=2\left|\psi_j\right|^2\left(\psi_{j+1}+\psi_{j-1}\right)+\psi_j^2\left(\psi_{j+1}^*+\psi_{j-1}^*\right) \tag{5.5}$$

The initial problem is now reduced to a form which is suitable to be analyzed using a quasiparticle approach which results in the following evolution equations for the parameters of each j-th soliton[81]

$$\frac{d\mu_j}{dz}=16\nu^3\left[\cos\left(\phi_{j+1,j}\right)\exp\left(-\Delta_{j,j+1}\right)-\cos\left(\phi_{j-1,j}\right)\exp\left(-\Delta_{j-1,j}\right)\right] \tag{5.6}$$

$$\frac{d\bar{x_j}}{dz}=2\mu_j-4\nu\left[\sin\left(\phi_{j+1,j}\right)\exp\left(-\Delta_{j,j+1}\right)-\sin\left(\phi_{j-1,j}\right)\exp\left(-\Delta_{j-1,j}\right)\right] \tag{5.7}$$

$$\begin{aligned}\frac{d\delta_j}{dz} &= 2\left(\nu^2+\mu_j^2\right)+2\mu\left(\frac{d\bar{x_j}}{dz}-2\mu_j\right)+ \\ &\quad +24\nu^2\left[\cos\left(\phi_{j+1,j}\right)\exp\left(-\Delta_{j,j+1}\right)+\cos\left(\phi_{j-1,j}\right)\exp\left(-\Delta_{j-1,j}\right)\right]\end{aligned} \tag{5.8}$$

Here $\phi_{j,l}=2\mu\left(\bar{x_l}-\bar{x_j}\right)+\Psi_{j,l}$ is the complex phase between solitons, with $\Psi_{j,l}=\delta_j-\delta_l$, the quantity $\Delta_{j,l}=2\nu\left|\bar{x_l}-\bar{x_j}\right|$ is the spacing between two solitons and ν and μ are the mean values for the amplitude and frequency, respectively. The derivation of these equations takes into account the assumptions that the overlap is small, i.e. $\nu\left|\bar{x_l}-\bar{x_j}\right|\gg 1$, and that the solitons have similar frequencies and amplitudes, i.e. $\left|\mu_j-\mu_l\right|\ll\mu$, $\left|\nu_j-\nu_l\right|\ll\nu$. Also it is assumed that the amplitude ν_j of each soliton is approximately constant, a fact which occurs in the adiabatic limit and is supported by small amplitude variation observed in numerical simulations.

We also adopt a conjecture commonly made [81], where it is assumed that if the system is initialized with consecutive solitons having a phase difference of 0 or π, this difference remains constant during the motion of the solitons. This assumption leads us

to the Toda Lattice Equation (TLE) [80]

$$\frac{\partial^2 u_j}{\partial z^2} = 32\cos(\Psi)\,\nu^3\exp(-2\nu\Delta)\left\{\exp\left(-2\nu\left(u_{j+1}-u_j\right)\right)-\exp\left(-2\nu\left(u_j-u_{j-1}\right)\right)\right\} \tag{5.9}$$

where $u_j = \bar{x_j} - x_j^0$ is the displacement of the $j-th$ soliton from its position of equilibrium x_j^0. The phase difference between consecutive solitons is represented by Ψ and the equilibrium lattice spacing is $\Delta = x_{j+1}^0 - x_j^0 = L/N$ with $[0, L]$ the limits of the simulation box. We impose the later condition considering that our problem has periodic boundary conditions and that each soliton is at a stable point of the lattice.

For small displacements we can expand the TLE as

$$\frac{\partial^2 u_j}{\partial z^2} \simeq C\left(u_{j+1} - 2u_j + u_{j-1}\right) + \mathcal{O}\left(u^2\right) \tag{5.10}$$

with the constant C given by $C = -64\cos(\Psi)\,\nu^4\exp(-2\nu\Delta)$. This equation resembles the equation of motion of phonons which have the traveling waves solution

$$u_j = \sum_k A_k e^{-i\left(kx_j^0 - \omega_k z\right)} \tag{5.11}$$

obeying the dispersion relation

$$\omega_k = \sqrt{\frac{2C}{M}\left(1-\cos k\Delta\right)} \tag{5.12}$$

with $M = 1$, that implies that C must be positive, which happens for $\Psi = \pi$.

It is important to discuss the validity of the assumption that the relative phase between consecutive solitons Ψ remains constant throughout the evolution of the system. Equation (5.8) implies that the phase of a moving soliton must change. Since the motion of two consecutive solitons is different then, their phase difference cannot remain constant. Therefore equation (5.9) and (5.8) are inconsistent[81], fact also stated by Novoa et. al [63] which lead them to consider an analog ideal binary system that eliminates the phase difference dependence fixing the interaction between consecutive solitons. Due to these questions and also to the nonlinearity of the TLE, it seems odd to expect a phonon-like behavior with traveling waves as normal modes. However, even knowing that the normal modes purposed above cannot describe perfectly the dynamics of the system, we expect that for small displacements and short evolutions the solitons can support collective oscillations. With the evolution of the system, there is an accumula-

tion of phase difference between solitons, which changes the nature of their interaction, resulting in a distortion of the phonon-like modes and ultimately destroying them.

5.3 Numerical study of the normal modes in a soliton chain

Several simulations were made of a chain of $N = 40$ solitons in a one dimensional domain with a cubic nonlinearity. The initial field condition provided was:

$$v = \sum_{j=1}^{N} 2\nu \text{sech}\left\{2\nu\left(x - x_j^0 - u_j\right)\right\} e^{ij\pi} \tag{5.13}$$

where $x_j^0 = \left(j + \frac{1}{2}\right)\Delta$. The u_j are chosen as $u_j = \sin\left(kj\Delta\right)$, where $k = 2\pi n/N$ is the wave number for this wave. Initializing the soliton frequencies as $\mu_j = 0$, we expect solutions of the type of standing waves, which lead us to a dispersion relation $\omega(k)$ that could be compared with the theoretical dispersion defined in equation (5.12).

The GPU used is a standard GeForce GTX 660 Ti. In all 1-dimensional simulations reported in the next section will be used the values $\Delta = 20$ and $\nu = 0.35$ for the soliton parameters. The spatial discretization used was $\Delta x = 800/2^{12}$ and the integration step was $h = 0.02$. The typical simulation time is about 11 minutes, being more than 13 times faster than the CPU versions.

Figure (5.1) shows the results of the simulations for standing waves in the solitonic chain with wave number equal to $k = 2\pi n/N$, $n = 3$ and $n = 15$. In the insets of figure (5.1) we can identify in more detail the oscillatory and wave-like motion of the solitons. We have chosen small oscillation amplitudes $A = 0.5$, compared to the distance between solitons, to satisfy the validity conditions of the model described in section 5.2.

From the results presented in figure (5.1) and to compare them with the model discussed in the previous section, we determine numerically the position of the geometric center of each soliton at a given integration step. In figure (5.2) we can compare the motion of a soliton observed in the simulations with that predicted by the Toda model. The match is closer for larger n where the phonon-like motion is more stable. In both cases, the amplitude of the oscillation increases during the evolution, implying that the system is not really undergoing a periodic motion. However, the characteristic time of the oscillation, or pseudo-period, defined as the period of return of the soliton to the position of equilibrium $u = 0$, remains stable throughout the simulation. For $n = 3$, the

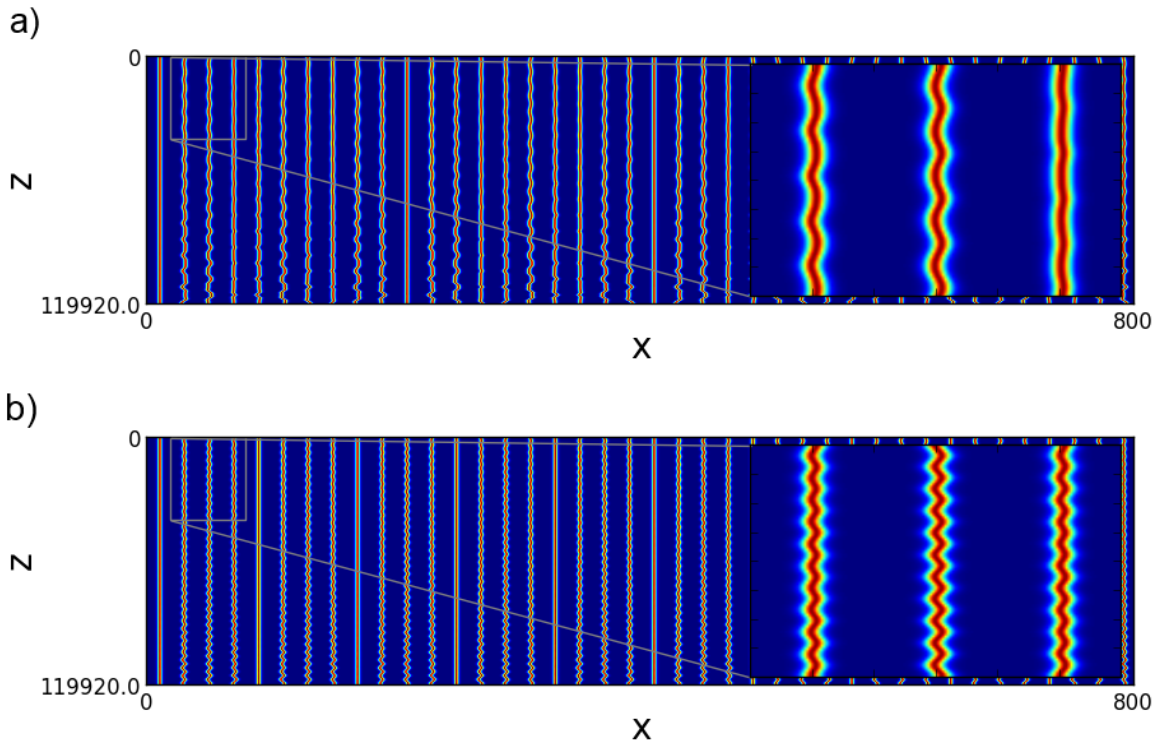


Figure 5.1: Evolution of the soliton chain for $n = 3$ (a)) and $n = 15$ (b)). Figures a) and b) show the intensity profile evolution where we can identify the phonon-like oscillations.

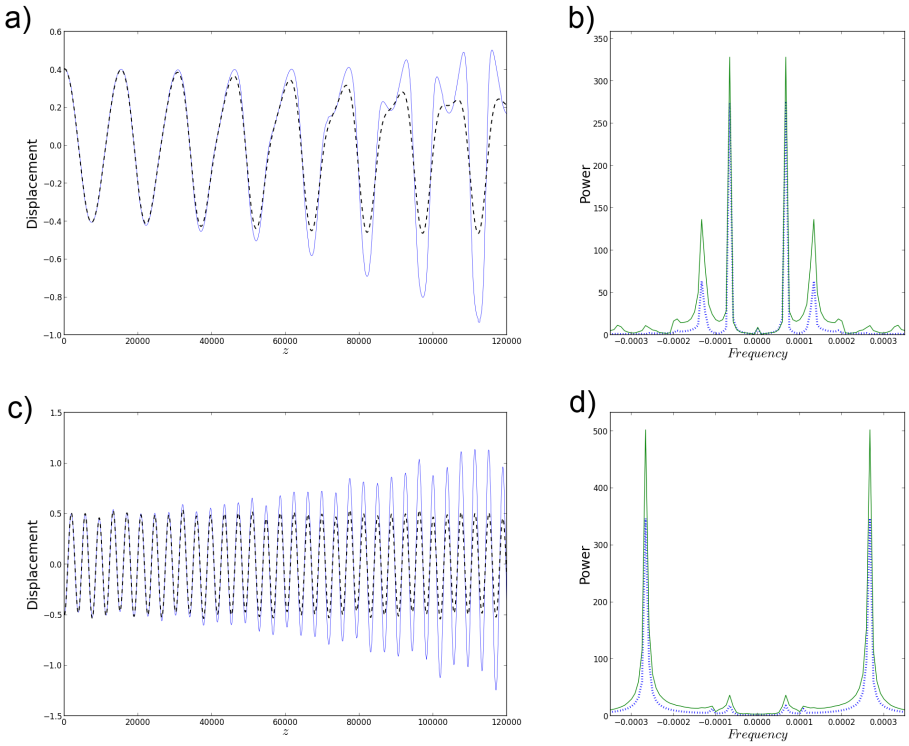


Figure 5.2: Evolution of the displacement of the second soliton of the chain from the right, for $n = 3$ (a)) and $n = 15$ (c)), comparing in detail the displacement obtained from the simulations (solid line) with the prediction of TLE (dashed line). Figures b) and d) display the Fourier transform of the displacements wave (numerical results with solid line, TLE with dashed line), where we can identify the generation of new frequencies for lower n.

nonlinear character of equation (5.9) is stronger and leads to the generation of higher harmonics in the spectrum of the motion of the soliton, as seen in figure (5.2b)).

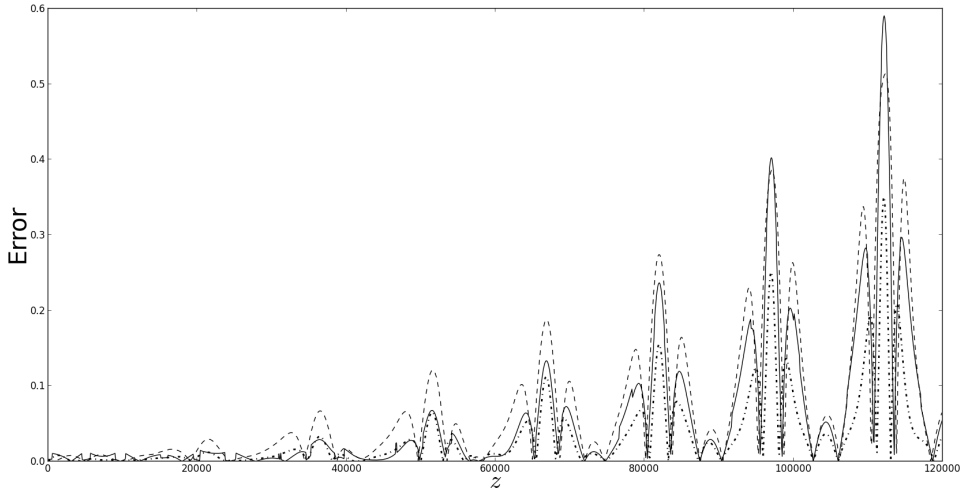


Figure 5.3: Error analysis of the displacement wave for a soliton between the numerical simulation and TLE model. Solid line is the absolute value of displacement error, $|u_2 - u_2^{TLE}|$, dashed and dash-point line is the absolute value of error for phase difference between consecutive solitons and the prediction $|\Psi_{2,3} - \pi|$, $|\Psi_{1,2} - \pi|$ respectively.

Figure (5.3) allows to verify the limitation of the TLE model by comparing the absolute difference between the displacement predicted and the calculated in the simulations, with the deviation of the phase difference between consecutive solitons from its initial value π. Clearly there is a good correlation which indicates the connection between the accumulation of error in phase difference and degradation of the phonon-like oscillations.

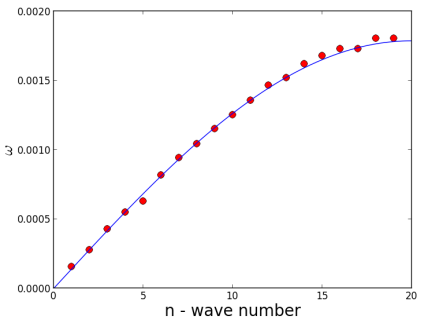


Figure 5.4: Dispersion relation computed for a chain of $N = 40$ solitons (circular markers) and comparison with the TLE model (solid line).

Simulating standing waves for an integer parameter $n \in]0, 20[$ we can obtain the right-side branch of the dispersion relation, shown in figure (5.4). As we can see, the data obtained from the pseudo-period values for each wave number leads to the expected theoretical dispersion, which confirms again the possibility of having standing waves propagating in chains of solitons. Similar results can be obtained for chains with as many solitons as wanted.

5.3.1 Other types of solutions

In the previous section we presented how a chain of solitons can sustain phonon-like excitations. Unlike the predictions of the model, the oscillations associated with soliton motion are not stable, but appear to reveal some type of feedback that yield increasing oscillation amplitudes. As a result, after some integration steps, the amplitude of the oscillation surpasses the mean distance between solitons and the chain collapses as soliton scatter each other, as seen in figure (5.5). This suggests the existence of a second regime, distinct from the spring-like interacting that supports soliton oscillations, where the nature of their interaction is closer to collisions.

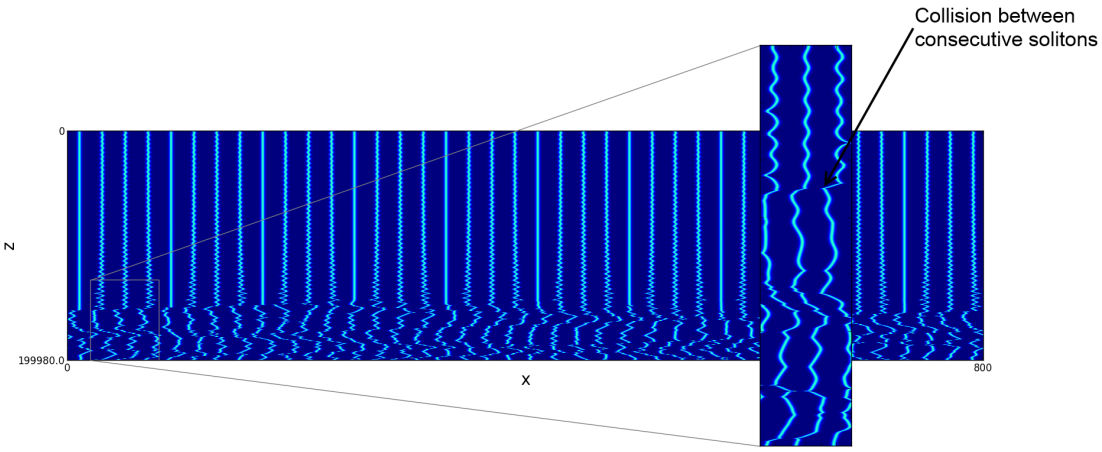


Figure 5.5: Collapse of the standing wave for displacements with $n = 15$. The feedback process increases the amplitude of the oscillation and when it surpasses the mean separation between solitons, solitons start to collide and the phonon-like behavior is lost.

To investigate this, we look into the collisional response of the chain, by considering it in equilibrium configuration where all the solitons are still equally apart. Then we send one soliton with a given velocity against the chain of solitons. To prevent the solitons of the chain from escaping, a weak linear potential of the form

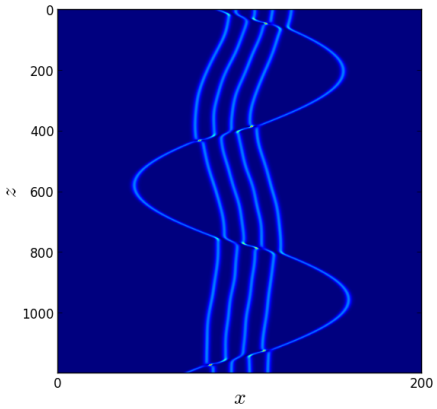


Figure 5.6: Evolution of the collision of a soliton with velocity $\mu = 0.2$ with a soliton chain with 4 solitons initially separated by $\Delta = 10$. The trapping potential allows to obtain results that resemble the Newton's cradle.

$$V(x) = 3 \times 10^{-5}(x - 100)^2$$

is added to the system.

The results are shown in figure (5.6). The chain now responds similar to a Newton's cradle, where each soliton transfers its momentum to the following and replaces it in the chain. This behavior is closer to a particle than to the mechanical-like waves, previously observed in the simulation. This appears to suggest that the system can exhibit many types of behavior, from wave-like to particle-like. This is a result of the nonlinear nature of the system which allows for distinct types of phenomena.

5.3.2 Soliton chains in (2+1)-d

Finally we show that these effects are not limited to soliton chains in (1+1)-d but can be generalized to higher spatial dimensions. In figure (5.7) we show a chain in a (2+1)-dimensional space. Here the spatial solitons are Gaussian shapes supported by a cubic-quintic media and the early results show that it is possible to propagate energy and momentum with wave-like phenomena in a chain of solitons.

5.4 Concluding remarks

We have explored the dynamics of an 1-dimensional chain of optical solitons in a cubic nonlinear media described by the NLSE. Using perturbative methods we show that the interaction between consecutive solitons could support wave-like oscillations of the positions of the optical solitons in the chain. We name these waves lightons: phonons of light. Numerical simulations show it is possible to create standing wave oscillations in the considered system that follows a predicted relation of dispersion. For small propagation distances and displacement amplitudes, the system revealed a controllable sinusoidal motion for the soliton position which is obtained only with light-light interaction. For higher amplitudes, the solitons enter a collisional regime and their behavior changes from a phonon type excitation to something that resembles a Newton cradle.

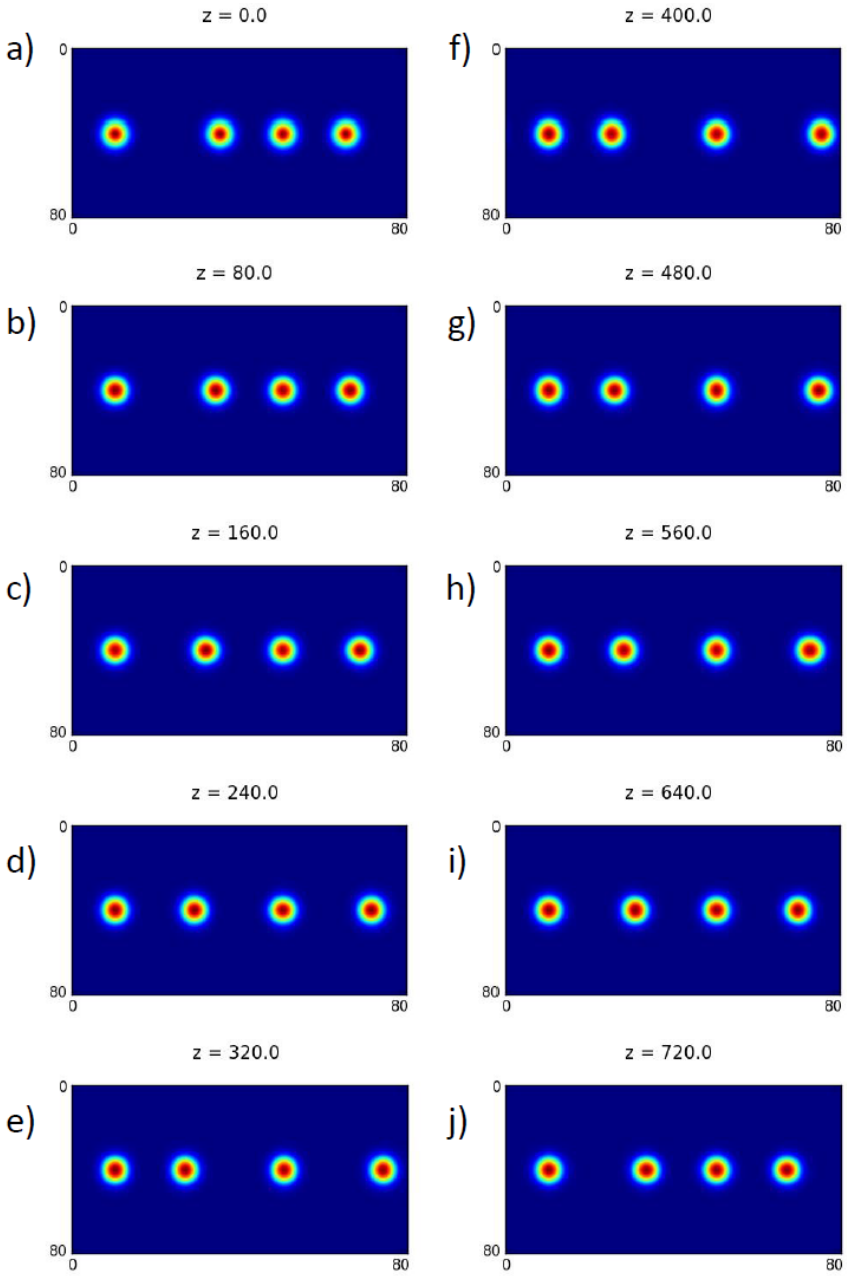


Figure 5.7: Evolution of an 1-dimensional chain of 2-dimensional spatial solitons with $\Delta = 20$.

6 Case study 2: Soliton-soliton scattering in (2+1)-d

In the previous chapter we analyzed an example of how the developed GNLSE solver can be used to study problems with high complexity, specifically, with a large number of solitons. In this chapter we demonstrate the potential of this computing approach in addressing systems with higher dimensionality. In particular, we study the collisions of (2+1)-d solitons in the physics of the dynamics in two dimensional plane. Unlike the case with (1+1)-d collisions, it is possible to study a wide range of collision parameters, including the impact parameter, the phase difference, the initial energies, etc.. We have chosen to analyze the transition between solid and liquid-light behavior and their influence on the collision dynamics.

6.1 Motivation

The example of the previous chapter illustrated how solitons can have a versatile nature, exhibiting both wave and particle-like behavior. However, the diversity of soliton behavior extends well beyond that. In systems with higher dimensionality, (2+1)-d or more, soliton-like solutions of the GNLSE must be supported by a mix of higher nonlinearities, as discussed in chapter 2.

Depending on the relative global phase of the soliton and type of nonlinear media, two solitons can scatter each other like rigid bodies, go right through each other like a wave, or coalesce in a wider soliton, like the coalescence of two droplets of light.

Exploring the dynamics of interacting solitons is a process with major interest in nonlinear optics. However, most studies have been restricted to the one dimensional simulations, where the simulations can be performed in a state-of-the-art computer. Exploring systems with higher dimensions was limited to the use of clusters and supercomputers. The use of GPU computing in GASE allows to explore these situations and take into account a wider and richer diversity of situations. In this chapter we explore

the scattering of solitons in cubic-quintic media in (2+1)-d scenarios. As a result, we are able to explore the interplay between solitons phase, their original velocities and impact parameters, thus providing a better insight of the physical processes.

6.2 Physical model

The starting point of analysis is again the GNLSE

$$i\frac{\partial\psi}{\partial z} + \frac{1}{2}\nabla_{\perp}^{2}\psi + F(|\psi|^2)\psi = 0, \tag{6.1}$$

which describes the evolution of the dimensionless amplitude of a light field ψ in a nonlinear media with properties given by the function $F(|\psi|^2)$. Here z is the longitudinal coordinate parallel to the propagation and $\nabla_{\perp}^{2}$ is the Laplacian in the transverse directions.

As discussed in chapter 2, for systems with (2+1) or more dimensions it is necessary to consider special types of nonlinearities for soliton-like solutions is assumed that the solutions of equation (6.1) can be described by a general soliton solution

$$q(x,z) = A(z)g[B\{x-\bar{x}(z)\}]\exp(-ik(z)\{x-\bar{x}(z)\}+i\theta(z)), \tag{6.2}$$

where A, B, g, k, θ,$\bar{x}(t)$ are the amplitude, the width, the shape, the frequency, the phase and the center of the soliton, respectively. The existence and the dynamics of the soliton can be then be investigated using variational methods in terms of the variation of these parameters for specific forms of F, corresponding to different nonlinearities. Media with cubic-quintic nonlinearities are known to support solitons in more than one dimension and have been extensively studied because they are described by a simple nonlinear potential of the form $F\left(|\psi|^2\right) = |\psi|^2 - |\psi|^4$. In this case, it is possible to obtain approximate solutions in the form of supergaussians pulses [18], defined by

$$\psi(r,z) = A\exp\left[-B^2(r-\bar{r})^{2m}\right]\exp\left(i\delta z\right), \tag{6.3}$$

where r is the vector with transverse coordinates and m a parameter related to pulse energy. For the soliton to be stable, these parameters must be mutually related by the following conditions,

$$A = \sqrt{\left(\frac{3}{2}\right)^{1/m} \frac{3}{2} \frac{m - \ln 2}{2m - \ln 3}} \tag{6.4}$$

$$B = \left(\frac{1}{A}\sqrt{\frac{2^{1/m}}{\Gamma(1+1/m)} \frac{2m - \ln 3}{\ln(4/3)}}\right)^{-1} \tag{6.5}$$

$$\delta = \frac{2B^2 m}{\Gamma(1+1/m)} \frac{m + \ln(2/3)}{\ln(4/3)}, \tag{6.6}$$

expressions derived from those presented in section 2.4.1 and from [18]. As the system is non integrable, studying the dynamics of the solitons is only possible using either perturbative methods or numerical simulations. Here, we focus on the numerical simulations because they provide a more direct way of studying the interactions of solitons in wider range of situations.

6.3 Scattering of colliding (2+1)-dimensional spatial solitons

We are interested in the computational analysis of two (2+1)-dimensional soliton collisions in the xy plane. The spatial solitons dynamics is described by the cubic-quintic GNLSE . The solitons have parameter $m = 1$, and the other parameters set according to equations (6.4-6.6). In this study, we initialize the solitons with a phase difference $\delta_1 - \delta_2 = 0$ or π and with opposing but equal velocities $|k_1|, |k_2| = k$ (see figure (6.1)), by multiplying the supergaussian shape () by $\exp\left(-ik(z)\{x - \bar{x}(z)\} + i\theta(z)\right)$.

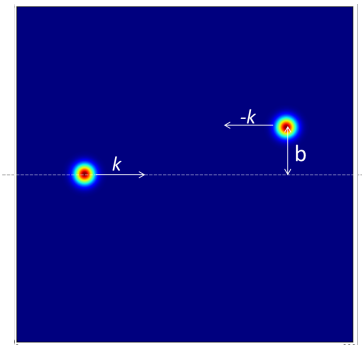


Figure 6.1: Graphical description of the problem analyzed. Here, impact parameter b was exaggerated for better comprehension. Simulation box has limits $[0, 80] \times [0, 80]$ and a mesh of $N = 2^{10} \times 2^{10}$ points was used.

The global phase difference $\delta_1 - \delta_2$ determines the nature of the interaction between the two solitons and ultimately their behavior, switching between particle, wave and liquid-like. As shown in the literature [46], the phase difference between solitons determines whether the interaction is attractive or repulsive. In the situations considered here, the two solitons and their trajectories are completely symmetric (point reflective symmetry relative to the origin), therefore their phase difference and the character of their interaction remains constant throughout the simulations. Another relevant parameter in the soliton-soliton scattering is the angular momentum, which introduces an effective repulsion between the solitons, and is determined by the initial velocity k and impact parameter b of the solitons. For large impact parameters and large velocities, the interaction is very weak and both solitons almost are not deflected from a straight trajectory, being nearly impossible to classify their behavior. Instead, for small impact parameters, the interaction is stronger and a wide variety of soliton behaviors can be observed, as shown in figure (6.2).

To help understand the different types of phenomena found, we introduce the following classification of soliton scattering:

- *hard soliton* scattering occurs for $\delta_1 - \delta_2 = \pi$, where their interaction is mainly repulsive and they interact as if they were rigid spheres for high velocities;
- *soft soliton* scattering occurs for $\delta_1 - \delta_2 = 0$ and the solitons have distinct behaviors, dominated by mutual attraction. Soft solitons can exhibit a wide variety of behaviors, from liquid light (when they coalesce into a single droplet of light) and "planetary-like", when they orbit for a few moments around each other. When the collision velocity is very large, solitons can also destroy each other giving rise to radiation.

The designation of hard and soft light introduced here follows a trend found in the literature and originated by several authors, including Michinel et. al [64] who introduced the concept of liquid light.

In the following sections we describe in more detail the behavior identified in the simulations.

6.4 Hard soliton scattering

For $\delta_1 - \delta_2 = \pi$, the two solitons are set to be out-of-phase and their interaction is strongly repulsive, as seen in figure (6.2) in sequences (a-j). This is confirmed in figure

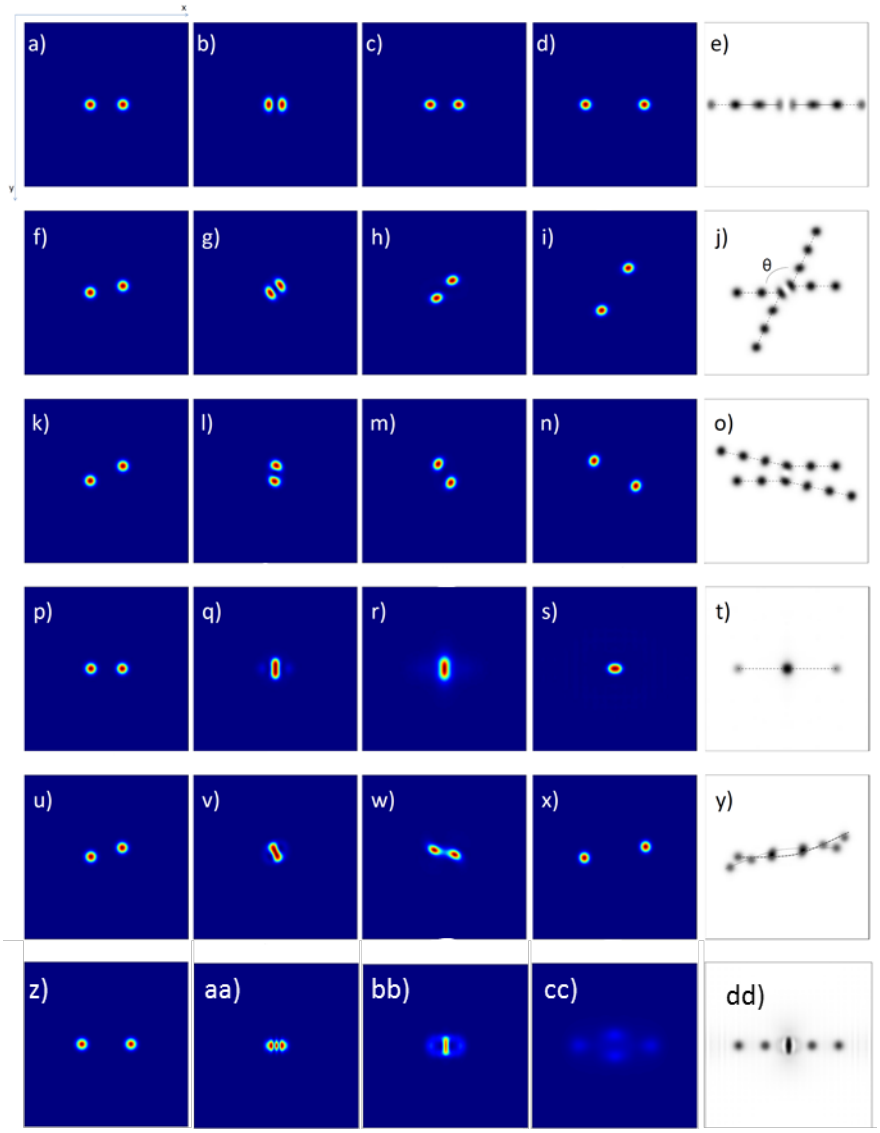


Figure 6.2: Typical numerical results for the evolution of two colliding solitons. Sequence a)-e) shows a collision between two out-of-phase solitons with $k_1 = 0.2$ and $b = 0$, sequence f)-j) displays the results for $b = 4$ and k)-o) for $b = 9$. Sequence p)-t) displays the coalescence of two colliding in-phase solitons with $b = 0$ and $k = 0.3$. Sequence u)-y) shows the results for $b = 5$ and $k = 0.3$. Sequence z)-dd) displays the destruction of two colliding in-phase solitons with $b = 0$ and $k = 0.8$.

(6.3), which shows the dependence of the scattering angle θ (measured as a deflection of the original straight trajectory) on the impact parameter.

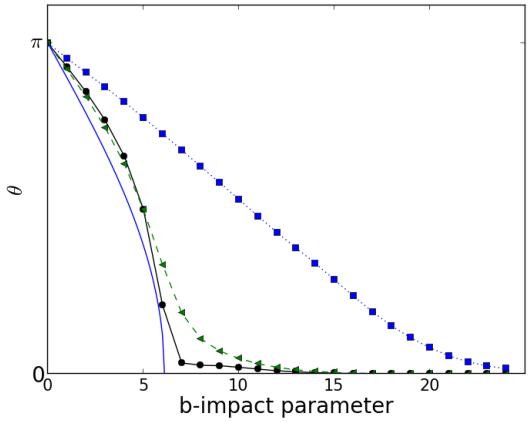


Figure 6.3: Computational results for the relation between the scattering angle and the impact parameter, $\theta(b)$. This figure displays the results for colliding out-of-phase solitons for $k = 0.05$ (full line with markers), $k = 0.2$ (dashed line with markers) and $k = 0.3$ (pointed line with markers). The full line without markers shows the hard-sphere limit for a sphere with radius of 6.

The increase of collision velocity further strengths the repulsive nature of the interaction but also forces the solitons to come closer to each other. As a result, in this limit the scattering angle approaches the results predicted by the scattering model for hard-spheres [45], revealing the particle-like behavior of solitons.

6.5 Soft soliton scattering

For $\delta_1 - \delta_2 = 0$, the two solitons are set to be in-phase and their interaction is attractive, as seen in figure (6.2) in sequences (k-y). However the anticipated particle-like dynamics does not hold for small impact parameter as the light pulses tend to coalesce, revealing a liquid-like behavior of solitons. In figure (6.4) it is seen that with the increase of the impact parameter the system undergoes a transition to a particle-like dynamics. For large impact parameters it is seen that the solitons almost do not interact, and thus the scattering angle is approximately null. With the decreasing of the impact parameter we see the expected increase of θ. The peak in figure (6.4) seems to be related with the formation of a quasi-stable state with angular momentum.

Unfortunately, the complete analysis of these processes is difficult since using the

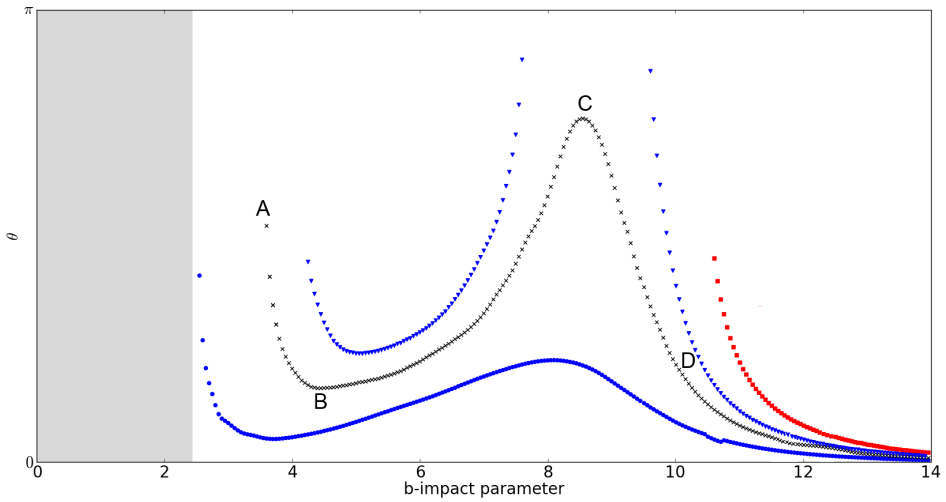


Figure 6.4: Computational results for the relation between the scattering angle and the impact parameter, $\theta(b)$. This figure shows the results for colliding in-phase solitons with $k = 0.3$ (circles), $k = 0.25$ (crosses), $k = 0.23$ (triangles) and $k = 0.2$ (squares). Different behaviors are obtained depending on both the collision velocity and the impact parameter. Shaded region is the zone of coalescence for $k = 0.3$, where solitons reveal liquid-like behavior.

basic algorithm of GASE it is impossible to separate the light intensity pertaining to each soliton. Only the total intensity in each point can be calculated.

To overcome this limitation, we developed a version of GASE algorithm where the original GNLSE is decomposed into two equations, say

$$i\frac{\partial\psi_1}{\partial z} + \frac{1}{2}\nabla_\perp^2\psi_1 + F(|\psi|^2)\psi_1 = 0, \tag{6.7}$$

$$i\frac{\partial\psi_2}{\partial z} + \frac{1}{2}\nabla_\perp^2\psi_2 + F(|\psi|^2)\psi_2 = 0, \tag{6.8}$$

each describing the evolution of the field associated with each soliton. Notice that the value of F is computed from $\psi = \psi_1 + \psi_2$ corresponding to the sum of the fields of both solitons. It is trivial to show that if ψ_1 and ψ_2 satisfy the equations (6.7) and (6.8) respectively then, $\psi = \psi_1 + \psi_2$ satisfies the original GNLSE (6.1).

This upgrade to the original GASE code allows to identify the evolution of each field. Figure (6.5-6.8) shows the results obtained for the scattering of two solitons with "planetary-like" trajectories. The results show a transfer of intensity between these two solitons. This suggests that during the interaction, the two solitons exchange energy

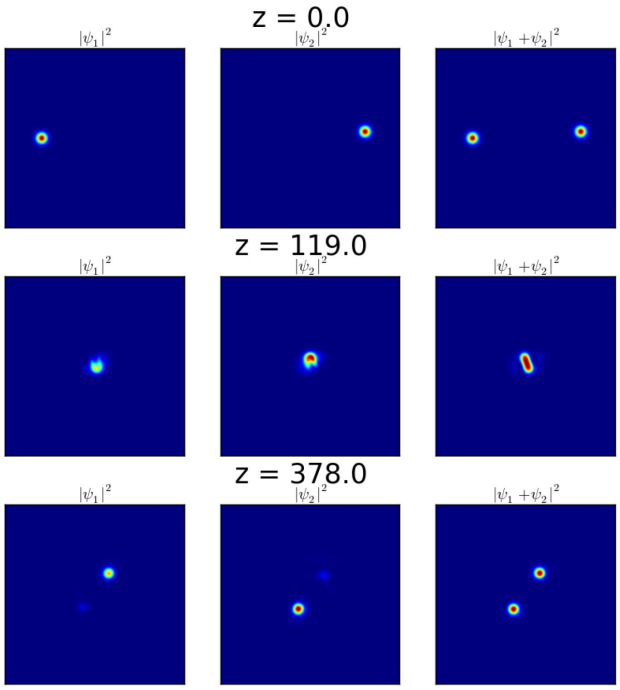


Figure 6.5: Sequence of data values for two coupled fields showing the collision of two in-phase solitons for the situation A in the figure (6.4). (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation)

and momentum. The development of this upgrade to the original code was done very recently and it was not possible to use it extensively to explore and analyze soliton dynamics. However, the results shown in figures (6.5-6.8) are illustrative of its potential.

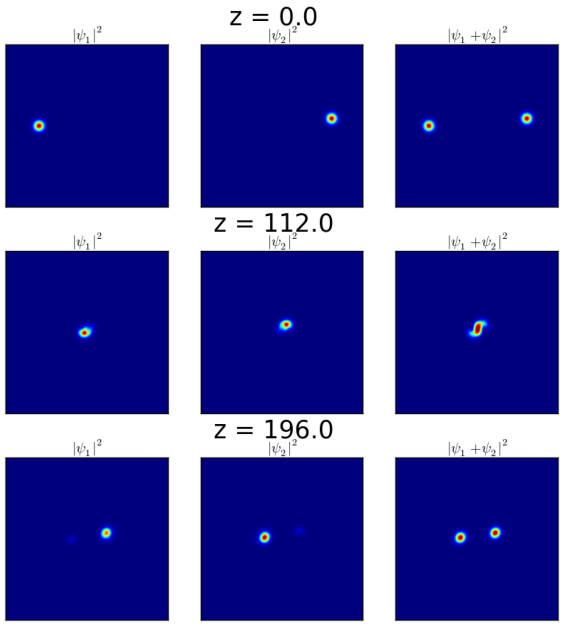


Figure 6.6: Sequence of data values for two coupled fields showing the collision of two in-phase solitons for the situation B in the figure (6.4). (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation)

6.6 Concluding remarks

The problem of soliton scattering introduced in this chapter appear to be an interesting and relevant topic of research presenting a wide diversity of phenomena and meriting a closer investigation that goes beyond the scope of this dissertation. We have chosen to present the results of our preliminary analysis to demonstrate the potential of GASE as tool of research in the field of soliton and nonlinear optics. As we hope to have demonstrated, it allows to solve complex and computationally intensive problems, that otherwise require expensive and massive computational resources. In fact, had we used a state-of-the-art CPU to solve it using a standard SSFM code, the estimated running time of the performed simulations would exceed 240 days. Almost as much as the duration of this dissertation project!. Instead, using GASE, it accounted for only 6 days of running

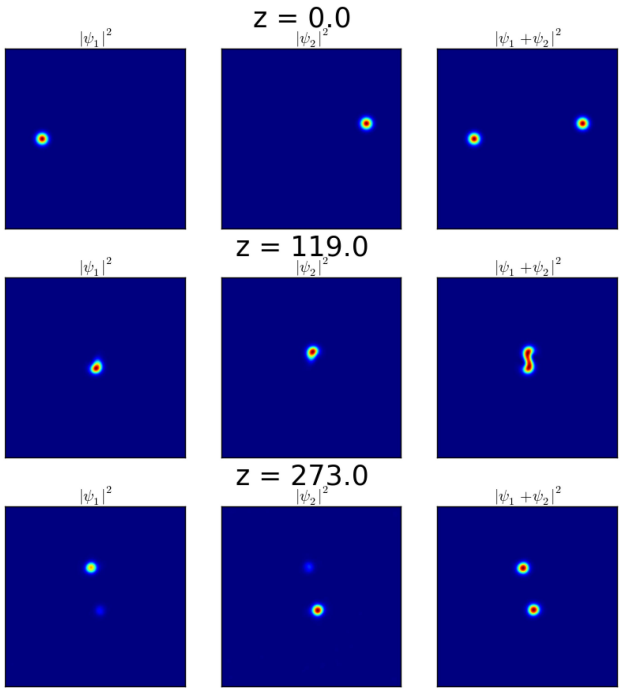


Figure 6.7: Sequence of data values for two coupled fields showing the collision of two in-phase solitons for the situation C in the figure (6.4). (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation)

time.

In the next chapter we discuss the future work, including the development of the topics of this and the previous chapters, and present our concluding remarks.

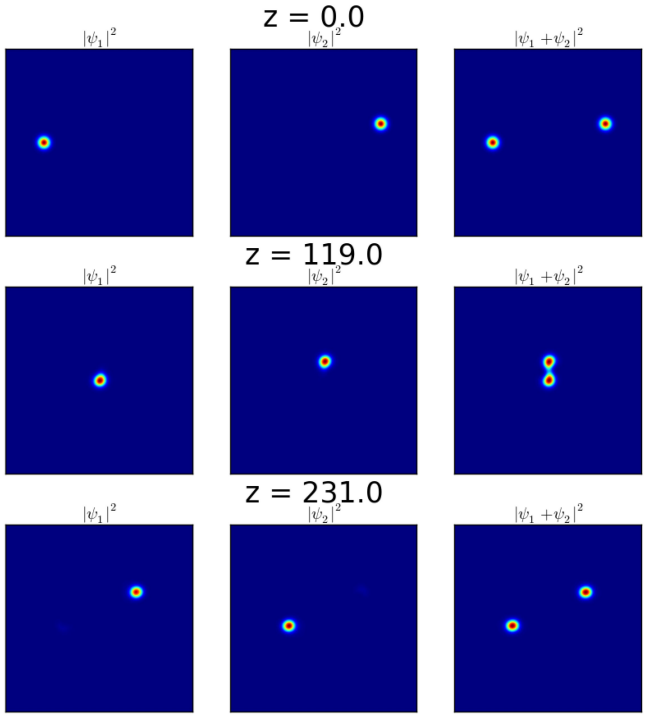


Figure 6.8: Sequence of data values for two coupled fields showing the collision of two in-phase solitons for the situation D in the figure (6.4). (*Pdf version only* - click twice on sub-figure a) for a small clip of the simulation)

7 Conclusions

The study of systems governed by the GNLSE is an active subject of nonlinear physics. The majority of the investigation on the GNLSE relies on the use of numerical methods, as analytical and variational methods often fail to provide solutions. For multidimensional systems and optical lattices, spatial distributions of linear or nonlinear refractive index, such research can became rather difficult and inefficient, as the computational running times for such problems are too long and can only be solved using modern and costly supercomputers. Thus, the main goal of the present dissertation is to develop a high performance numerical solver for the GNLSE, capable of addressing those problems in normal computers and, through that, provide a basis for future research.

The GPU computing technology is recent, still taking its first steps, and relies on a whole new paradigm of computing, known as heterogeneous computing, to achieve massive parallel computation. Particularly, GPUs constitute an interesting new tool for scientific computing, as they not only have an incredible computational power and potential, but also because their evolution is fast as it is fed by the gaming industry, one of the most powerful industries of the world. This allows to foresee that in the near future, even higher performances could be obtained.

With this in mind, we developed GASE - GPU Accelerated Soliton Explorer - a solver of the GNLSE that uses the SSFM and it is based on GPU computing. In fact, this solver can run in a low-budget desktop or laptop, having only as a "special ingredient" a recent NVIDIA graphical processing unit. These devices are standard in modern computers and for a desktop a current high-line GPU costs around 300 euros. It was observed that the performance obtained by our solver in such low-cost solutions competes with the usual computer clusters and supercomputers, that costs thousands of euros. Then, the methodology approached in this dissertation should become increasingly important in the following years.

However, the development of computational codes for scientific purposes using the GPU could not be done straightforwardly from the old algorithms, numerical methods and codes used in "normal" CPU computation. Also, the type of parallelism used in

computer clusters is different, being necessary to re-adapt the numerical tools to work in GPU. In particular, it must be taken into account in which type of memory the data is saved in other to avoid possible memory bottlenecks. On the other hand, the GPU computing belongs to a more general computing paradigm: the heterogeneous computing. The heterogeneous computing aims to use every resources of a computer (CPU, GPU and others) to do massive parallel computations. In the GASE, this concept was used to improve even more the performance of the solver by allocating different jobs to the CPU and the GPU, having in mind the times associated to the data transfer between them. This way, all of the resources of the machine are used simultaneously during the process and yields speedups in excess of 100 when comparing to simulations run using a top-of-the-line CPU. With the future improvement of these technologies it is expected that this optimal value of GASE could be even larger.

Besides the performance results of GASE, we should also underline its versatility in the analysis of physical problems described by the GNLSE. The code is capable of addressing simulation boxes with 1, 2 and 3 dimensions and the number of sampling points of the field in excess of 2^{23}, which allows reasonably large simulation domains. Also, several types of nonlinearities can be used simultaneously. These include not only quadratic, cubic, quintic, logarithmic, etc. nonlinearities but also spatial distributions (including the dependence on the propagation coordinate) of the nonlinear refraction index - called nonlinear lattices. Basically, any type of nonlinearity could be simulated. We have also included three types of boundary conditions, namely periodic, reflective or absorbing boundary conditions. Finally, in the last stage of the development, we included the possibility of simulating several fields satisfying their own GNLSE but coupled to each other.

The development of the code GASE included extensive testing to validate the numerical accuracy of the results and to benchmark the code in relation to other numerical approaches: Crank-Nicholson GPU-based solver and CPU SSFM. In all these cases, GASE was superior in performances. We concluded this dissertation by presenting the application of GASE to the study of two physical problems.

Although the results obtained in both problems correspond to a preliminary study, they illustrate the versatility and potential of GASE and GPU computing. Had we more time and we could have explored both these problems in more depth, as well as investigate others. In the following section we give a brief overview of future work that could be developed with the help of GASE.

7.1 Future work

At the end of this dissertation, there is a multitude of extensions to this work that could be done. From the point of view of the solver, GASE can be extended to being capable of addressing the problems with many coupled GNLSE, a feature that can be useful for problems where the field has a vector character. As novices in GPU computing, we believe that there are still certain optimizations that could be done and lend to an even faster solver, such as asynchronous memory transfer from the GPU to the CPU or others. The extension of GASE to work in other GPUs from other manufacturers rather than NVIDIA can also be done, passing the developed CUDA code to the OPEN CL language. The extension of GASE to multi-GPU architectures is also another point of interest.

From the point of view of the study of the GNLSE, there is still plenty of room for further developments. In fact, as GASE is capable of addressing almost any problem described by the GNLSE, the solver is ready to tackle a multitude of research problems in the subject of solitons, either in multidimensional systems or in the case of optical lattices. As the GNLSE has a certain character of universality, being present in many fields of physics, the same solver could also constitute a tool for studying phenomena in Bose-Einstein condensates, plasma physics and fluid dynamics, among many others. The solver can also be proven useful in other areas of physics, such as general relativity, where it can be used to analyze the propagation of gravitational waves, a high nonlinear type of wave. It is our hope that in the short term GASE will become an important tool of research in these problems.

7.2 Publications

Conference proceedings

- Nuno A. Silva, M. I. Carvalho, A. Guerreiro, Lightons: Phonons with light, Proceedings of RIAO/OPTILAS 2013, Porto, Portugal, 2013
- Nuno A. Silva, M. I. Carvalho, A. Guerreiro, Spatial soliton dynamics in cubic-quintic media, Proceedings of RIAO/OPTILAS 2013,Porto, Portugal, 2013

Poster presentations

- Nuno A. Silva, M. I. Carvalho, A. Guerreiro, Lightons: Phonons with light, NANOPT 2013, Porto, Portugal, 2013

- Nuno A. Silva, M. I. Carvalho, A. Guerreiro, Lightons: Phonons with light, RIAO/OPTILAS 2013, Porto, Portugal, 2013
- Nuno A. Silva, M. I. Carvalho, A. Guerreiro, Spatial soliton dynamics in cubic-quintic media, RIAO/OPTILAS 2013, Porto, Portugal, 2013

Bibliography

[1] Soliton wave receives crowd of admirers. *Nature*, 376:373, August 1995.

[2] G. Agrawal. *Nonlinear Fiber Optics*. Academic Press, 3 edition, January 2001.

[3] Adrian Alexandrescu, Jose R. Salgueiro, Victor M. Perez-Garcia, and Humberto Michinel. Nonperturbative vector solitary waves in four-level coherent media. *Phys. Rev. A*, 79:063843, Jun 2009.

[4] D. Anderson. Variational approach to nonlinear pulse propagation in optical fibers. *Phys. Rev. A*, 27:3135–3145, Jun 1983.

[5] David W. Aossey, Steven R. Skinner, Jamie L. Cooney, James E. Williams, Matthew T. Gavin, David R. Andersen, and Karl E. Lonngren. Properties of soliton-soliton collisions. *Phys. Rev. A*, 45:2606–2610, Feb 1992.

[6] Naruyoshi Asano, Tosiya Taniuti, and Nobuo Yajima. Perturbation method for a nonlinear wave modulation. ii. *Journal of Mathematical Physics*, 10(11):2020–2024, 1969.

[7] A. Balevic, L. Rockstroh, A. Tausendfreund, S. Patzelt, G. Goch, and S. Simon. Accelerating simulations of light scattering based on finite-difference time-domain method with general purpose gpus. In *Computational Science and Engineering, 2008. CSE '08. 11th IEEE International Conference on*, pages 327–334, 2008.

[8] G. P. Berman and F. M. Izrailev. The Fermi-Pasta-Ulam problem: Fifty years of progress. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 15(1):015104, 2005.

[9] Anjan Biswas. Adiabatic dynamics of non-Kerr law solitons. *Applied Mathematics and Computation*, 151(1):41 – 52, 2004.

[10] Alexander V. Buryak, Yuri S. Kivshar, Ming-feng Shih, and Mordechai Segev. Induced coherence and stable soliton spiraling. *Phys. Rev. Lett.*, 82:81–84, Jan 1999.

[11] Ron Caplan. Study of vortex ring dynamics in the nonlinear Schrodinger equation utilizing gpu-accelerated high-order compact numerical integrators, 2003.

[12] Ronald M. Caplan. Nlsemagic: Nonlinear Schrodinger. *CoRR*, abs/1203.1263, 2012.

[13] Lapo Casetti, Monica Cerruti-Sola, Marco Pettini, and E. G. D. Cohen. The Fermi-Pasta-Ulam problem revisited: Stochasticity thresholds in nonlinear hamiltonian systems. *Phys. Rev. E*, 55:6566–6574, Jun 1997.

[14] R. Y. Chiao, E. Garmire, and C. H. Townes. Self-trapping of optical beams. *Phys. Rev. Lett.*, 13:479–482, Oct 1964.

[15] NVIDIA Corporation. Popular gpu accelerated applications, 2012.

[16] A S Davydov. Solitons in molecular systems. *Physica Scripta*, 20(3-4):387, 1979.

[17] A S Desyatnikov and Andrei I Maimistov. Conservation of the angular momentum for multidimensional optical solitons. *Quantum Electronics*, 30(11):1009, 2000.

[18] Kristian Dimitrevski, Erik Reimhult, Erik Svensson, Anders Ohgren, Dan Anderson, Anders Berntson, Mitek Lisak, and Manuel L. Quiroga-Teixeiro. Analysis of stable self-trapping of laser beams in cubic-quintic nonlinear media. *Physics Letters A*, 248(5-6):369–376, 1998.

[19] R. C. Diprima, W. Eckhaus, and L. A. Segel. Non-linear wave-number interaction in near-critical two-dimensional flows. *Journal of Fluid Mechanics*, 49:705–744, 10 1971.

[20] Sergey V. Dmitriev, Yuri S. Kivshar, and Takeshi Shigenari. Fractal structures in multi-soliton collisions. *Physica B: Condensed Matter*, 316-317(0):139–142, 2002. Proceedings of the 10th International Conference on Phonon Scattering in Condensed Matter.

[21] Galen C. Duree, John L. Shultz, Gregory J. Salamo, Mordechai Segev, Amnon Yariv, Bruno Crosignani, Paolo Di Porto, Edward J. Sharp, and Ratnakar R. Neurgaonkar. Observation of self-trapping of an optical beam due to the photorefractive effect. *Phys. Rev. Lett.*, 71:533–536, Jul 1993.

[22] J.; Ulam S. Fermi, E.; Pasta. Studies of nonlinear problems. Technical report, 1955.

[23] Jason W. Fleischer, Mordechai Segev, Nikolaos K. Efremidis, and Demetrios N. Christodoulides. Observation of two-dimensional discrete solitons in optically induced nonlinear photonic lattices. *Nature*, 422(6928):147–150, March 2003.

[24] Michael Fleischhauer, Atac Imamoglu, and Jonathan P. Marangos. Electromagnetically induced transparency: Optics in coherent media. *Rev. Mod. Phys.*, 77:633–673, Jul 2005.

[25] Victor M. Perez Garcia and Xiao yan Liu. Numerical methods for the simulation of trapped nonlinear Schrodinger systems. *Applied Mathematics and Computation*, 144(2-3):215–235, 2003.

[26] Clifford S. Gardner, John M. Greene, Martin D. Kruskal, and Robert M. Miura. Method for solving the Korteweg-deVries equation. *Phys. Rev. Lett.*, 19:1095–1097, Nov 1967.

[27] E.P. Gross. Structure of a quantized vortex in boson systems. *Il Nuovo Cimento Series 10*, 20(3):454–477, 1961.

[28] Chao Hang and V. V. Konotop. Spatial solitons in a three-level atomic medium supported by a laguerre-gaussian control beam. *Phys. Rev. A*, 83:053845, May 2011.

[29] Mark Jason Harris. Real-time cloud simulation and rendering, 2003.

[30] Akira Hasegawa and Frederick Tappert. Transmission of stationary nonlinear optical pulses in dispersive dielectric fibers. i. anomalous dispersion. *Applied Physics Letters*, 23(3):142–144, 1973.

[31] Rainer W. Hasse. Schrödinger solitons and kinks behave like newtonian particles. *Phys. Rev. A*, 25:583–584, Jan 1982.

[32] Hermann A. Haus and William S. Wong. Solitons in optical communications. *Rev. Mod. Phys.*, 68:423–444, Apr 1996.

[33] Simon Huang, Peng Zhang, Xiaosheng Wang, and Zhigang Chen. Observation of soliton interaction and planetlike orbiting in Bessel-like photonic lattices. *Opt. Lett.*, 35(13):2284–2286, Jul 2010.

[34] A.A. Kanashov and A.M. Rubenchik. On diffraction and dispersion effect on three wave interaction. *Physica D: Nonlinear Phenomena*, 4(1):122 – 134, 1981.

[35] V.I. Karpman and V.V. Solov'ev. A perturbational approach to the two-soliton systems. *Physica D: Nonlinear Phenomena*, 3(3):487 – 502, 1981.

[36] Y. V. Kartashov, V. A. Vysloukh, and L. Torner. Solitons in complex optical lattices. *The European Physical Journal Special Topics*, 173(1):87–105, 2009.

[37] Yaroslav V. Kartashov, Boris A. Malomed, and Lluis Torner. Solitons in nonlinear lattices. *Rev. Mod. Phys.*, 83:247–305, Apr 2011.

[38] Yaroslav V. Kartashov, Victor A. Vysloukh, and Lluis Torner. Rotary solitons in Bessel optical lattices. *Phys. Rev. Lett.*, 93:093904, Aug 2004.

[39] Yaroslav V. Kartashov, Victor A. Vysloukh, and Lluis Torner. Stable ring-profile vortex solitons in Bessel optical lattices. *Phys. Rev. Lett.*, 94:043902, Feb 2005.

[40] Yaroslav V. Kartashov, Victor A. Vysloukh, and Lluis Torner. Soliton control in fading optical lattices. *Opt. Lett.*, 31(14):2181–2183, Jul 2006.

[41] Yaroslav V. Kartashov, Victor A. Vysloukh, and Lluis Torner. Brownian soliton motion. *Phys. Rev. A*, 77:051802, May 2008.

[42] Yaroslav V. Kartashov, Victor A. Vysloukh, and Lluis Torner. Soliton modes, stability, and drift in optical lattices with spatially modulated nonlinearity. *Opt. Lett.*, 33(15):1747–1749, Aug 2008.

[43] P.G. Kevrekidis, D.J. Frantzeskakis, and R. Carretero-González. *Emergent Nonlinear Phenomena in Bose-Einstein Condensates: Theory and Experiment.* Atomic, Optical, and Plasma Physics. Springer, 2007.

[44] Ali Khajeh-Saeed and J. Blair Perot. Direct numerical simulation of turbulence using {GPU} accelerated supercomputers. *Journal of Computational Physics*, 235(0):241 – 257, 2013.

[45] Tom W. B. Kibble and Frank H. Berkshirel. *Classical Mechanics.* Imperial College Press, London, fifth edition, 2004.

[46] Yuri S. Kivshar, Govind P. Agrawal, and Govind P. Agrawal. *Optical Solitons.* Academic Press, Burlington, 2004.

[47] Yukihiro Komura and Yutaka Okabe. Gpu-based single-cluster algorithm for the simulation of the ising model. *Journal of Computational Physics*, 231(4):1209 – 1215, 2012.

[48] D. J. Korteweg and G. de Vries. Xli. on the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves. *Philosophical Magazine Series 5*, 39(240):422–443, 1895.

[49] Andrei I Maimistov. Solitons in nonlinear optics. *Quantum Electronics*, 40(9):756, 2010.

[50] Andrey Maimistov, Boris Malomed, and Anton Desyatnikov. A potential of incoherent attraction between multidimensional solitons. *Physics Letters A*, 254(3-4):179–184, 1999.

[51] Boris A. Malomed. Bound solitons in the nonlinear schrödinger–ginzburg-landau equation. *Phys. Rev. A*, 44:6954–6957, Nov 1991.

[52] Boris A. Malomed. Potential of interaction between two- and three-dimensional solitons. *Phys. Rev. E*, 58:7928–7933, Dec 1998.

[53] Boris A Malomed, Dumitru Mihalache, Frank Wise, and Lluis Torner. Spatiotemporal optical solitons. *Journal of Optics B: Quantum and Semiclassical Optics*, 7(5):R53, 2005.

[54] S. Maneuf, R. Desailly, and C. Froehly. Stable self-trapping of laser beams: Observation in a nonlinear planar waveguide. *Optics Communications*, 65(3):193 – 198, 1988.

[55] Robert McLeod, Kelvin Wagner, and Steve Blair. (3+1)-dimensional optical soliton dragging logic. *Phys. Rev. A*, 52:3254–3278, Oct 1995.

[56] Robert McLeod, Kelvin Wagner, and Steve Blair. Variational approach to orthogonally-polarized optical soliton interaction with cubic and quintic nonlinearities. *Physica Scripta*, 59(5):365, 1999.

[57] David Michea and Dimitri Komatitsch. Accelerating a three-dimensional finite-difference wave propagation code using gpu graphics cards. *Geophysical Journal International*, 182(1):389–402, 2010.

[58] H. Michinel, J. Campo-Taboas, R. Garcia-Fernandez, J. R. Salgueiro, and M. L. Quiroga-Teixeiro. Liquid light condensates. *Phys. Rev. E*, 65:066604, Jun 2002.

[59] Humberto Michinel, Maria J. Paz-Alonso, and Victor M. Perez-Garcia. Turning light into a liquid via atomic coherence. *Phys. Rev. Lett.*, 96:023903, Jan 2006.

[60] Humberto Michinel, Jose R. Salgueiro, and Maria J. Paz-Alonso. Square vortex solitons with a large angular momentum. *Phys. Rev. E*, 70:066605, Dec 2004.

[61] L. F. Mollenauer, R. H. Stolen, and J. P. Gordon. Experimental observation of picosecond pulse narrowing and solitons in optical fibers. *Phys. Rev. Lett.*, 45:1095–1098, Sep 1980.

[62] Gaspar D. Montesinos and Victor M. Perez-Garcia. Numerical studies of stabilized townes solitons. *Math. Comput. Simul.*, 69(5):447–456, August 2005.

[63] David Novoa, Boris A. Malomed, Humberto Michinel, and Victor M. Perez-Garcia. Supersolitons: Solitonic excitations in atomic soliton chains. *Phys. Rev. Lett.*, 101:144101, Sep 2008.

[64] David Novoa, Humberto Michinel, and Daniele Tommasini. Pressure, surface tension, and dripping of self-trapped laser beams. *Phys. Rev. Lett.*, 103:023903, Jul 2009.

[65] Carolyn L. Phillips, Joshua A. Anderson, and Sharon C. Glotzer. Pseudo-random number generation for brownian dynamics and dissipative particle dynamics simulations on {GPU} devices. *Journal of Computational Physics*, 230(19):7191 – 7201, 2011.

[66] Tobias Preis, Peter Virnau, Wolfgang Paul, and Johannes J. Schneider. {GPU} accelerated monte carlo simulation of the 2d and 3d ising model. *Journal of Computational Physics*, 228(12):4468 – 4477, 2009.

[67] M. L. Quiroga-Teixeiro, A. Berntson, and H. Michinel. Internal dynamics of nonlinear beams in their ground states: short- and long-lived excitation. *J. Opt. Soc. Am. B*, 16(10):1697–1704, Oct 1999.

[68] F. Reynaud and A. Barthelemy. Optically controlled interaction between two fundamental soliton beams. *EPL (Europhysics Letters)*, 12(5):401, 1990.

[69] J. Scott Russel. Report on waves. Technical report, 1844.

[70] M. Salsi, O. Bertran-Pardo, J. Renaudier, W. Idler, H. Mardoyan, P. Tran, G. Charlet, and S. Bigo. Wdm 200gb/s single-carrier pdm-qpsk transmission over 12,000km. In *Optical Communication (ECOC), 2011 37th European Conference and Exhibition on*, pages 1–3, 2011.

[71] A. Scott. *Encyclopedia of Nonlinear Science*. Routledge and Sons, 2005.

[72] Wen-Rui Shan, Feng-Hua Qi, Rui Guo, Yu-Shan Xue, Pan Wang, and Bo Tian. Conservation laws and solitons for the coupled cubic quintic nonlinear Schrodinger equations in nonlinear optics. *Physica Scripta*, 85(1):015002, 2012.

[73] Ming Feng Shih and Mordechai Segev. Incoherent collisions between two-dimensional bright steady-state photorefractive spatial screening solitons. *Opt. Lett.*, 21(19):1538–1540, Oct 1996.

[74] Ming-feng Shih, Mordechai Segev, and Greg Salamo. Three-dimensional spiraling of interacting spatial solitons. *Phys. Rev. Lett.*, 78:2551–2554, Mar 1997.

[75] Mário G. Silveirinha. Effective medium response of metallic nanowire arrays with a Kerr-type dielectric host. *Phys. Rev. B*, 87:165127, Apr 2013.

[76] K. Stewartson and J. T. Stuart. A non-linear instability theory for a wave system in plane poiseuille flow. *Journal of Fluid Mechanics*, 48:529–545, 8 1971.

[77] C. Sulem and P.L. Sulem. *The Nonlinear Schrodinger Equation: Self-Focusing and Wave Collapse*. Number v. 139 in Applied Mathematical Sciences. Springer, 1999.

[78] Qingqing Sun, Yuri V. Rostovtsev, and M. Suhail Zubairy. Optical beam steering based on electromagnetically induced transparency. *Phys. Rev. A*, 74:033819, Sep 2006.

[79] Zhi-Yuan Sun, Yi-Tian Gao, Xin Yu, Wen-Jun Liu, and Ying Liu. Bound vector solitons and soliton complexes for the coupled nonlinear schrödinger equations. *Phys. Rev. E*, 80:066608, Dec 2009.

[80] Morikazu Toda. Vibration of a chain with nonlinear interaction. *Journal of the Physical Society of Japan*, 22(2):431–436, 1967.

[81] I.M. Uzunov, V.S. Gerdjikov, M. Golles, and F. Lederer. On the description of n-soliton interaction in optical fibers. *Optics Communications*, 125(4):237–242, 1996.

[82] Sibo Wang, Junbo Xu, and Hao Wen. Accelerating dissipative particle dynamics with multiple {GPUs}. *Computer Physics Communications*, 184(11):2454 – 2461, 2013.

[83] Lluis Torner Yaroslav V. Kartashov, Anna S. Zelenina and Victor A. Vysloukh. Spatial soliton switching in quasi-continuous optical arrays. *Opt. Lett.*, 29(7):766–768, Apr 2004.

[84] F. Ye, D. Mihalache, B. Hu, and N. C. Panoiu. Subwavelength plasmonic lattice solitons in arrays of metallic nanowires. *Phys. Rev. Lett.*, 104:106802, Mar 2010.

[85] Henry C. Yuen and Bruce M. Lake. Nonlinear deep water waves: Theory and experiment. *Physics of Fluids*, 18(8):956–960, 1975.

[86] N. J. Zabusky and M. D. Kruskal. Interaction of "solitons" in a collisionless plasma and the recurrence of initial states. *Phys. Rev. Lett.*, 15:240–243, Aug 1965.

[87] V. E. Zakharov and A. B. Shabat. Exact theory of two-dimensional self-focusing and one-dimensional self-modulation of waves in nonlinear media. *Sov. Phys*, 34(3):62–69, jan 1972.