An adaptive Galerkin method for the time-dependent complex Schrödinger equation

A.I. Ávila[1], A. Meister[2], M. Steigemann[2]

[1] *Departamento de Ingeniería Matemática, CEMCC, Universidad de La Frontera, Temuco, Chile*
[2] *Department of Mathematics and Natural Sciences, University of Kassel, Germany*

Abstract. Nonlinear time-dependent Schrödinger equations (NLSE) model several important problems in quantum physics and morphogenesis. Recently, vortex lattice formation were experimentally found in Bose-Einstein condensate and Fermi superfluids, which are modeled by adding a rotational term in the NLSE equation. Numerical solutions have been computed by using separate approaches for time and space variables. If we see the complex equation as a system, wave methods can be used.
In this article, we consider finite element approximations using continuous Galerkin schemes in time and space by adaptive mesh balancing both errors. To get this balance, we adapt the dual weighted residual method used for wave equations and estimates of error indicators for adaptive space-time finite element discretization. The results show how important is dynamic refinement to control the degrees of freedom in space.

Key Words: nonlinear time-dependent Schrödinger equation, dual weighted residual method, adaptive Galerkin method.

1. Introduction

Dynamics of ultracold atoms is a complex phenomena, where parameter changes can involve phase transitions like superfluidity.When using external magnetic fields changing across the Feshbach resonance, the classical Bose-Einstein Condensate (BEC) for bosons can change to Barden-Cooper-Schrieffer superfluidity of fermions. To ensure experimental evdence, quantized vortices are observed. This phenomena has been experimentally studied by Zwierlein et al. [26] for a strongly interacting Fermi gas. The authors use the magnetic field to create lattices with different vortex number ranging from 1 to 40 and different lifetime showing a very rich problem. The vortex number depends on the rotational frequency. Finding the critical frequency which the vortex can last longer is another problem to model. To model this problem, a rotational term is added to the nonlinear Schrödinger equation (NLSE) to trigger vortex formation [3]

$$\mathrm{i}\varepsilon\partial_t u(x,t) = -\frac{\varepsilon^2}{2}\Delta u(x,t) + V(x)u(x,t) - f\big(x;|u(x,t)|^2\big)u(x,t) + \mathrm{i}\varepsilon\Omega L(x,\nabla_x)u(x,t), \tag{1}$$

where $x \in \mathbb{R}^d$, $t > 0$, ε is a small positive parameter, and the rotational term is modelled by the x_3-component of angular momentum

$$\mathrm{i}\Omega L(x,\nabla_x)u(x,t) = \mathrm{i}\Omega\big(x_1\partial_{x_2}u(x,t) - x_2\partial_{x_1}u(x,t)\big), \tag{2}$$

with rotational velocity $\Omega \in \mathbb{R}$. Different theoretical and numerical approaches have been used to understand this problem. Most of the abstract results deals with the search for ground states for the stationary NLSE. In case of the dynamic equation, results are scarce. We will focus on the numerical

In: Applied Numerical Mathematics, Volume 121 (2017), Page 149-169

solutions. Bao et al. [9] used polar coordinates to get a constant rotating term. Then, they define an algorithm based on a time splitting and Fourier pseudospectral discretization for the space-variable. The advantages of theses algorithms are: explicit, unconditionally stable, four-order accuracy in time. Ming et al [18] also studied a two-component Bose-Einstein condensates (BECs) using the time splitting sine pseudospectral method. Recently, Antoine & Duboscq [3] used Krylov subspace solvers to improve the speed of the iterative schem for the linear system. Also, FEM has been used to study the classical NSLE [1]. Rencently, Henning et al [16] propose a FEM approach for space variables and backward Euler discretization for the time variable to derive stability results. Using high order Time Splitting approach, Hofstätter et al. [17] studied the error behavior for this problem.
In this work we compute numerical solutions of the NLSE based on finite element approximations using continuous Galerkin schemes in time and in space. Some works on GP equaiton also studied both adaptivity (see [22]). A question in this context is always how to obtain precise solutions with less computational effort. Increasing the precision is always related with the refinement of the underlying discretization, but especially in time-dependent problems global refinement is not a realistic option, since the number of degrees of freedom will increase astronomically. Adaptive mesh refinement and an efficient strategy for balancing the temporal and spatial discretization error contributions is one of the a keys to compute precise numerical results. For mesh refinement in time, we need a mechanism to detect the time intervals for decreasing the time steps to get higher precision. One of the well-known approaches to gain such a time step control is embedded Runge-Kutta schemes as the Fehlberg method or the Cash-Karp method. See [15] for a review and many more details. Here we follow another concept. In many situations we are not directly interested in the precision of the numerical solution, but a quantity of physical interest which can be computed with the numerical solution, for example the energy or the mass. Adaptive mesh refinement based on error control of a given functional is the core of the dual weighted residual (DWR) method [10], [7] applied to wave equations. The DWR was applied to several physical problems in the last years. The main goal of this work is to adapt the DWR method to nonlinear time-dependent Schrödinger equations.
In section 2 we introduce the problem and notations. We present continuous Galerkin schemes in time and space in section 3. We show how the given Galerkin scheme can be reinterpreted as a Crank-Nicolson time stepping method and present the ideas of the DWR method. We calculate the temporal and spatial error indicators and use an algorithm from [20] for the balance of the different error contributions. Finally we present numerical results in section 4.

2. Description of the problem

Superfluidity is a phenomenon where a fluid preserves its kinetic energy. In case of ultracold gases, the solitons decay to a single or several quantized vortices depending on the rotational frequency [13]. At very low temperatures, a Bose-Einstein condensate in a rotational frame can be described by the complex-valued macroscopic wave function u whose evolution is governed by a mean field nonlinear Schrödinger equation known as the Gross-Pitaevskii equation (GPE) with an angular momentum rotational term:

$$\mathrm{i}\hbar\partial_t u(x,t) = \left(-\frac{\hbar^2}{2m}\Delta + \frac{m}{2}V(x) + NU_0|u(x,t)|^2 + \mathrm{i}\hbar\Omega L(x,\nabla_x)\right)u(x,t), \qquad x \in G, \quad t > 0. \tag{3}$$

$G \subset \mathbb{R}^d$ is a domain in $d = 2$ or $d = 3$ space dimension, $\hbar$ is the Planck constant, m is the atomic mass and N is the number of atoms in the condensate. The function V is a given real-valued external potential depending on the application. In this paper, we consider the function V as a harmonic trap potential

$$V(x) = \frac{\omega^2}{2}\left(\gamma_1^2 x_1^2 + \gamma_2^2 x_2^2 + \gamma_2^2 x_3^2\right),$$

where γ_j, $j = 1, 2, 3$, are the trap frequencies and $x = (x_1, x_2, x_3)^\top \in \mathbb{R}^3$. The number $U_0 = \frac{4\pi\hbar^2 a}{m}$ describes the interaction between the atoms with the wave scattering length a. If U_0 is positive, the interaction is called repulsive and if it is negative the interaction is called attractive. An example for a set of parameters used in experiments with ^{23}Rb presented in [2] is

$$\hbar = 1.05 \cdot 10^{-34} J, \qquad m \approx 1.44 \cdot 10^{-25} kg, \qquad N \approx 10^2 \dots 10^7, \qquad a = 5.1 \cdot 10^{-9} m.$$

By dividing equation (3) by the mass m, we introduce a dimensionless parameter $\varepsilon := \frac{\hbar}{m}$, $0 < \varepsilon \leq 1$, the so-called "scaled" Planck constant, see (1). Depending on the problem the parameter ε can be very small, see e.g. [9] and [4] for further details. In this work we do not focus on the parameter ε and assume $\varepsilon = 1$. The main goal of this work is a finite element scheme with adaptive mesh refinement in time and space for the time-dependent Schrödinger equation:

$$\mathrm{i}\partial_t u(x,t) = -\frac{1}{2}\Delta u(x,t) + V(x)u(x,t) - f\big(|u(x,t)|^2\big)u(x,t) + \mathrm{i}\Omega L(x,\nabla_x)u(x,t) \tag{4}$$

for $x \in G \subset \mathbb{R}^d$, $d = 2, 3$, $t > 0$, with initial data

$$u(x,0) = u_0(x), \qquad x \in G,$$

and subjected to homogeneous Dirichlet boundary conditions

$$u(x,t) = 0, \qquad x \in \partial G, \quad t \geq 0.$$

We assume further, that $f : \mathbb{R}^+ \to \mathbb{R}$, $f(s) \sim s$, is a real-valued, continuously differentiable function and refer to [8] for more details.

The energy functional associated with the NLSE (4) is defined as

$$E\big(u(t)\big) = \frac{1}{2}\int_G \left(\frac{1}{2}|\nabla u(x,t)|^2 + V(x)|u(t,x)|^2 - F\big(|u(x,t)|^2\big) + \mathrm{i}\Omega\overline{u(x,t)}L(x,\nabla_x)u(x,t)\right) dx$$

where $\overline{u}$ is the conjugate of u and $F(\rho) = \int_0^\rho f(s)\,ds$. A short calculation shows that the energy is a real number: Splitting the wave function into its real and imaginary part, $u = u^0 + \mathrm{i}u^1$, the rotational part of the energy is equal to

$$\begin{aligned}\int_G \mathrm{i}\overline{u(x,t)}L(x,\nabla_x)u(x,t)\,dx = &\int_G \big(u^1(x,t)L(x,\nabla_x)u^0(x,t) - u^0(x,t)L(x,\nabla_x)u^1(x,t)\big)\,dx \\ &+ \mathrm{i}\int_G \big(u^0(x,t)L(x,\nabla_x)u^0(x,t) + u^1(x,t)L(x,\nabla_x)u^1(x,t)\big)\,dx, \qquad t > 0.\end{aligned}$$

Because we assume homogeneous Dirichlet boundary conditions, the imaginary part vanishes applying the Gauss theorem. The Schrödinger equation preserves energy, $E(u(t)) = E(u(0))$, and mass

$$M(u(0)) = M\big(u(t)\big) = \int_G |u(x,t)|^2\,dx, \qquad t \geq 0.$$

We normalize the wave function by requiring that the initial data is normalized to

$$\|u^0\|^2 := \int_G |u^0(x)|^2\,dx = 1.$$

Especially for numerical investigations, we are interested in stationary states with low energy of the NLSE, so called *ground state solutions*. Their existence for general conditions is still an open problem. In our numerical experiments, we will fulfill the conditions given in [9] to get the existence and uniqueness.

Stationary solutions are decomposed by

$$u(x,t) = e^{-\mathrm{i}\mu t}\Phi(x)$$

where $\mu \in C$ is the chemical potential and Φ is a function independent of time. This separation of variables leaves the mass $\|u\|^2$ constant in time. We notice that the pair (μ, Φ) solves the nonlinear eigenvalue problem

$$\begin{aligned} \mu\Phi(x) &= -\frac{1}{2}\Delta\Phi(x) + V(x)\Phi(x) - f\big(|\Phi(x)|^2\big)\Phi(x) + \mathrm{i}\Omega L(x,\nabla)\Phi(x), \qquad x \in G, \\ \Phi(x) &= 0, \qquad x \in \partial G. \end{aligned} \tag{5}$$

under the normalization condition $\|\Phi\|^2 = 1$. For known Φ, the eigenvalue μ can be computed from the relation

$$\begin{aligned} \mu\|\Phi\|^2 &= \int_G \left(\frac{1}{2}|\nabla\Phi(x)|^2 + V(x)|\Phi(x)|^2 - f\big(|\Phi(x)|^2\big)|\Phi(x)|^2 + \mathrm{i}\Omega\overline{\Phi(x)}L(x,\nabla_x)\Phi(x)\right) dx \\ &= 2E(\Phi) + \int_G \Big(F(|\Phi(x)|^2) - f\big(|\Phi(x)|^2\big)|\Phi(x)|^2\Big)\, dx, \end{aligned} \tag{6}$$

and these functions correspond to critical points of the energy functional E on the unit sphere.

3. Error estimation and time step control

In this section we develop a framework for computing numerical solutions of the NLSE (4). Therefore, we formulate the approximation of solutions of (4) by continuous Galerkin finite elements in time and in space. We approximate the real and the imaginary parts by the same finite elements and use the dual weighted residual method for error and time step control. The dual weighted residual (DWR) method was mainly developed by Rannacher and his coworkers and it was adapted with great success to many different problems of mathematical physics in the last years. See e.g. [10], [11], [7], [5], [12] and the works cited in [7] for an introduction and more details. Indeed, we develop an error estimator of the form

$$J(u) - J(u_{kh}) \approx \eta_k + \eta_h$$

where J is a quantity of physical interest, for example the energy at some time, and η_k and η_h are bounds for the temporal and spatial discretization errors. These error bounds must be localized to time and space cells in order to use them for mesh adaption and we calculate the explicit form of these error indicators. The last step is then a strategy for mesh adaption providing a good balancing and an equilibrated reduction of the temporal and spatial discretization errors.

Notation and weak formulation

Splitting the wave function into its real and imaginary part, $u = u^0 + \mathrm{i}u^1$, the function u can be identified with the pair $\{u^0, u^1\} \in \mathcal{V} \times \mathcal{V}$, where $\mathcal{V}$ is an appropriate function space. The equations involving u^0 and u^1 read

$$\begin{aligned} \partial_t u^0 &= -\frac{1}{2}\Delta u^1 + Vu^1 - f\big(|u|^2\big)u^1 + \Omega L(\nabla)u^0 =: \mathscr{A}_2(u), \\ -\partial_t u^1 &= -\frac{1}{2}\Delta u^0 + Vu^0 - f\big(|u|^2\big)u^0 - \Omega L(\nabla)u^1 =: \mathscr{A}_1(u), \end{aligned} \qquad \text{in} \quad G \times I, \tag{7}$$

in a time interval $I = (0,T)$ and with initial condition

$$u^0(x,0) = u_0^0(x), \qquad u^1(x,0) = u_0^1(x), \qquad x \in G. \tag{8}$$

We start with a Gelfand triple $\mathcal{V} \hookrightarrow \mathcal{H} \hookrightarrow \mathcal{V}^*$. A natural choice for the energy space for real and imaginary parts of solutions is

$$\mathcal{V} := H_0^1(G) \quad \text{and} \quad \mathcal{H} := L^2(G).$$

Further we introduce the Hilbert space

$$\mathcal{X} := W(0,T) := \big\{v : v \in L^2(I,\mathcal{V}) \text{ and } \partial_t v \in L^2(I,\mathcal{V}^*)\big\}$$

which is continuously embedded in $C(\overline{I},\mathcal{H})$ (e.g. [25]). The inner product in $L^2(I,H)$ is given by

$$(\!(\phi,\psi)\!) := (\phi,\psi)_{L^2(I,\mathcal{H})} = \int_0^T \big(\phi(t),\psi(t)\big)_G \, dt, \qquad (\psi,\phi)_G := \int_G \psi(x)\phi(x)\,dx.$$

Let $A_j : \mathcal{V}\times\mathcal{V}\times\mathcal{V} \to \mathbb{R}$, $j = 1,2$, the spatial semilinear elliptic forms describing the weak formulation of problem (7):

$$A_1(u)(\phi) := \frac{1}{2}\big(\nabla u^0, \nabla\phi\big) + \big(Vu^0,\phi\big) - \big(f\big(|u|^2\big)u^0,\phi\big), -\Omega\big(L(\nabla)u^1,\phi\big),$$

$$A_2(u)(\phi) := \frac{1}{2}\big(\nabla u^1, \nabla\phi\big) + \big(Vu^1,\phi\big) - \big(f\big(|u|^2\big)u^1,\phi\big) + \Omega\big(L(\nabla)u^0,\phi\big).$$

Introducing the space-time semilinear forms $A_j : \mathcal{X}\times\mathcal{X}\times\mathcal{X} \to \mathbb{R}$, $j = 1,2$,

$$A_1(\!(u)(\phi)\!) := \int_0^T A_1\big(u(t)\big)\big(\phi(t)\big)\,dt, \qquad A_2(\!(u)(\phi)\!) := \int_0^T A_2\big(u(t)\big)\big(\phi(t)\big)\,dt,$$

the system (7) is equivalent to the following variational problem: Find $u = \{u^0,u^1\} \in \mathcal{X}\times\mathcal{X}$ satisfying

$$\begin{aligned} (\!(\partial_t u^0,\varphi^0)\!) &= A_2(\!(u)(\varphi^0)\!), & (\!(\partial_t u^1,\varphi^1)\!) &= -A_1(\!(u)(\varphi^1)\!), \\ u^0(0) &= u_0^0, & u^1(0) &= u_0^1, \end{aligned} \tag{9}$$

for all test functions $\varphi = \{\varphi^0,\varphi^1\} \in \mathcal{X}\times\mathcal{X}$, where $u_0 = \{u_0^0,u_0^1\} \in L^2(G)\times L^2(G)$ describes the initial condition.

Time discretization by continuous Galerkin finite elements

In order to discretize the time interval we choose a decomposition

$$\overline{I} = [0,T] = \{0\} \cup \bigcup_{m=1}^{M} I_m$$

with M time steps $I_m := (t_{m-1},t_m]$ of length $k_m := t_m - t_{m-1}$ using the time points

$$0 = t_0 < t_1 < \ldots < t_m < \ldots < t_M = T.$$

The maximal length $k := \max_m k_m$ and we denote the time discretization by $\mathbb{T}_k = \{I_m : 1 \le m \le M\}$. For reasons which become clear later, we require, that M is an even number: $M \in 2\mathbb{N}$. To allow

adaptive mesh refinement in space, we follow the Rothe approach and discretize in time first, then in space. For $r \in \mathbb{N}_0$ we define the function spaces

$$\mathcal{X}_k^r := \left\{ v_k \in C(\overline{I}, \mathcal{H}) : v_k\big|_{I_m} \in \mathcal{P}_r(I_m, \mathcal{V}) \right\} \subset \mathcal{X},$$

$$\widetilde{\mathcal{X}}_k^r := \left\{ v_k \in L^2(I, \mathcal{V}) : v_k\big|_{I_m} \in \mathcal{P}_r(I_m, \mathcal{V}) \text{ and } v_k(0) \in L^2(G) \right\},$$

where $\mathcal{P}_r(I_m, \mathcal{V})$ denotes the space of polynomials up to degree r on the interval I_m with values in $\mathcal{V}$. The space $\mathcal{X}_k^r$ consists of piece-wise polynomials which are continuous in time and this will be the trial space for the continuous (in time) Galerkin method. The elements of the space $\widetilde{\mathcal{X}}_k^r$ can have discontinuities at the end points of the intervals I_m. This space will be used as the test space for the Galerkin method. The degree of the test functions is chosen one degree lower as the trial functions, because one degree of freedom of the trial functions per time interval is fixed by the global continuity condition. By this construction we get a quadratic system of equations which can be decoupled in each time step. The resulting scheme can be reinterpreted as a time-stepping method.

With these definitions, the time-discrete variational problem can be stated to: Find $u_k = \{u_k^0, u_k^1\} \in \mathcal{X}_k^r \times \mathcal{X}_k^r$ satisfying

$$\begin{aligned} (\!(\partial_t u_k^0, \varphi^0)\!) - A_2((\!(u_k)\!)(\varphi^0)\!) + \left(u_k^0(0), \varphi_0^{0,-}\right)_G &= \left(u_0^0, \varphi_0^{0,-}\right)_G, \\ (\!(\partial_t u_k^1, \varphi^1)\!) + A_1((\!(u_k)\!)(\varphi^1)\!) + \left(u_k^1(0), \varphi_0^{1,-}\right)_G &= \left(u_0^1, \varphi_0^{1,-}\right)_G, \end{aligned} \tag{10}$$

for all $\varphi = \{\varphi^0, \varphi^1\} \in \widetilde{\mathcal{X}}_k^{r-1} \times \widetilde{\mathcal{X}}_k^{r-1}$. Here,

$$\varphi_m^{j,-} = \lim_{\tau \searrow 0} \varphi^j(t_m - \tau) = \varphi^j(t_m), \qquad j \in \{0, 1\},$$

denotes the value of a possible discontinuous test function at time point t_m.

For the discretization in space we choose a mesh $\mathbb{T}_h$ which consists of nonoverlapping quadrilateral cells K. The discretization parameter h is defined as the maximum of all cell diameters. We allow "hanging nodes", which means that cells can have nodes which lie on midpoints of faces or edges of neighboring cells. In addition, we require that the mesh consists of patches of cells. This means, that $\mathbb{T}_h$ is obtained by uniform refinement of a coarser mesh $\mathbb{T}_{2h}$, such that one can always combine four (in two dimensions) or eight (in three dimensions) adjacent cells of $\mathbb{T}_h$ to obtain one cell, called a *patch*, of $\mathbb{T}_{2h}$.

On the mesh $\mathbb{T}_h$ we construct a finite element space

$$\mathcal{V}_h^s := \left\{ v \in C(\overline{G}) : v\big|_K \in \mathcal{Q}_s(K) \text{ for } K \in \mathbb{T}_h \right\} \subset \mathcal{V}.$$

We use isoparametric elements on the space $\mathcal{Q}_s(K)$, which consists of shape functions obtained by transformations of polynomials $\widehat{\mathcal{Q}}_s(\widehat{K})^d$ on the reference cell $\widehat{K} = [0,1]^d$, see e.g. [7]. The mesh can be adapted in each time step, but the local time steps k_m are kept constant in space. With each point t_m in time we associate a mesh $\mathbb{T}_h^m$ and a corresponding (spatial) finite element space $\mathcal{V}_h^{s,m}$ and we define the space-time finite element space by

$$\widetilde{\mathcal{X}}_{kh}^{r,s} := \left\{ v_{kh} \in L^2(I, \mathcal{V}) : v_{kh}\big|_{I_m} \in \mathcal{P}_r(I_m, \mathcal{V}_h^{s,m}) \text{ and } v_k(0) \in \mathcal{V}_h^{s,0} \right\} \subset \widetilde{\mathcal{X}}_k^r.$$

In order to ensure the global continuity of the functions of the trial space, we use an approach from [20]: Let $\{\tau_0, \ldots, \tau_r\}$ be a basis of $\mathcal{P}_r(I_m, \mathbb{R})$ with the following property:

$$\tau_0(t_{m-1}) = 1, \quad \tau_0(t_m) = 0, \quad \tau_i(t_{m-1}) = 0, \quad i = 1, \ldots, r.$$

We set

$$\mathcal{X}_{kh}^{r,s,m} := \operatorname{span}\{\tau_i v_i : v_0 \in \mathcal{V}_h^{s,m-1}, v_i \in \mathcal{V}_h^{s,m}, i = 1,\dots,r\} \subset \mathcal{P}_r(I_m, \mathcal{V})$$

and define the trial space

$$\mathcal{X}_{kh}^{r,s} := \{v_{kh} \in C(\overline{I}, \mathcal{H}) : v_k|_{I_m} \in \mathcal{X}_{kh}^{r,s,m}\} \subset \mathcal{X}_k^r.$$

This construction ensures the continuity of all functions of the space $\mathcal{X}_{kh}^{r,s}$, because the vanishing spatial degrees of freedom in $\mathcal{V}_h^{s,m-1}$ are coupled only with the temporal basis function τ_0 which vanishes at the right end of each subinterval, see [20] for further details. The fully discrete so-called $cG(s)cG(r)$ *space-time finite element Galerkin formulation*, which reads: Find $u_{kh} = \{u_{kh}^0, u_{kh}^1\} \in \mathcal{X}_{kh}^{r,s} \times \mathcal{X}_{kh}^{r,s}$ satisfying

$$\begin{aligned} (\!(\partial_t u_{kh}^0, \varphi^0)\!) - A_2(\!(u_{kh})(\varphi^0)\!) + \big(u_{kh}^0(0), \varphi_0^{0,-}\big)_G &= \big(u_0^0, \varphi_0^{0,-}\big)_G, \\ (\!(\partial_t u_{kh}^1, \varphi^1)\!) + A_1(\!(u_{kh})(\varphi^1)\!) + \big(u_{kh}^1(0), \varphi_0^{1,-}\big)_G &= \big(u_0^1, \varphi_0^{1,-}\big)_G, \end{aligned} \tag{11}$$

for all $\varphi = \{\varphi^0, \varphi^1\} \in \widetilde{\mathcal{X}}_{kh}^{r-1,s} \times \widetilde{\mathcal{X}}_{kh}^{r-1,s}$.

The notation $cG(s)cG(r)$ describes a Galerkin method with a "continuous in space" discretization of order s and with a "continuous in time" discretization of order r, see for example [14].

Relation to a time stepping scheme

For the convenience of the reader we outline the connection to time stepping schemes. Let us consider the case $r = 1$, the local problem on the subinterval $I_m = (t_{m-1}, t_m]$ reads

$$\begin{aligned} \int_{I_m} \Big\{(\partial_t u^0, \varphi^0) - A_2(u)(\varphi^0)\Big\}\, dt = 0, \\ \int_{I_m} \Big\{(\partial_t u^1, \varphi^1) + A_1(u)(\varphi^1)\Big\}\, dt = 0, \end{aligned} \tag{12}$$

for all $\varphi^0 \in \mathcal{P}_0(I_m, \mathcal{V}_h)$ and $\varphi^1 \in \mathcal{P}_0(I_m, \mathcal{V}_h)$. With the nodal basis$\{\varphi_1, \dots, \varphi_N\}$ of the spatial finite element space $\mathcal{V}_h$ the trial functions $\{u_{kh}^0, u_{kh}^1\}$ at a time point t_m can be written as

$$U_m^j(x) := u_{kh}^j(x, t_m) = \sum_{l=1}^{N} y_{m,l}^j \varphi_l(x), \qquad j \in \{0, 1\}, \tag{13}$$

and on the interval I_m there holds

$$u_{kh}^j(x,t)\big|_{I_m} = \sum_{l=1}^{N} \left\{ y_{m-1,l}^j + \frac{(t - t_{m-1})}{k_m}(y_{m,l}^j - y_{m-1,l}^j) \right\} \varphi_l(x), \qquad j \in \{0, 1\}. \tag{14}$$

Inserting (13) into equations (12) and testing successively with the basis functions φ_i, $i = 1, \dots, N$, leads to the following set of fully discrete equations

$$\begin{aligned} \boldsymbol{M} y_m^0 &= \boldsymbol{M} y_{m-1}^0 + \frac{k_m}{2}\big(\boldsymbol{A}_2(U_{m-1}) + \boldsymbol{A}_2(U_m)\big), \\ \boldsymbol{M} y_m^1 &= \boldsymbol{M} y_{m-1}^1 - \frac{k_m}{2}\big(\boldsymbol{A}_1(U_{m-1}) + \boldsymbol{A}_1(U_m)\big), \end{aligned} \tag{15}$$

with the mass matrix and the vectors

$$\boldsymbol{M} = \big((\varphi_l, \varphi_i)\big)_{i,l=1}^{N}, \qquad \boldsymbol{A}_1\big(U_m\big) := \big(A_1(U_m)(\varphi_i)\big)_{i=1}^{N}, \qquad \boldsymbol{A}_2\big(U_m\big) := \big(A_2(U_m)(\varphi_i)\big)_{i=1}^{N}.$$

That means that $cG(s)cG(1)$ is algebraically equivalent to the Crank-Nicolson scheme, an implicit Runge-Kutta scheme with two stages and of order two. This result was shown for the wave equation in [5]. It is well-know that the Crank-Nicolson scheme is unconditionally stable in the L^2 norm, of convergence order two and energy conserving, all these properties hold for the $cG(s)cG(1)$ Galerkin scheme.

For practical reasons we need further requirements on the used meshes. The time stepping method (15) works without problems if the mesh does not change between two steps. But if we allow dynamic mesh adaption in each time step, we have to calculate inner products of functions of the spaces $\mathcal{V}_h^{s,m-1}$ and $\mathcal{V}_h^{s,m}$. Integrals over the different nodal basis functions cannot be evaluated cell-wise by quadrature rules because the integrands are not necessarily smooth inside each cell. To overcome this problem, we additionally assume, that all meshes $\mathbb{T}_h^m$ result from one mesh $\widetilde{\mathbb{T}}_h$ by hierarchical refinement or coarsening. In this case it is possible to build up a mesh $\mathbb{T}_h^{m-1/2}$ as the "union" of the two meshes. In other words, the mesh $\mathbb{T}_h^{m-1/2}$ is the set of the most refined cells of the meshes $\mathbb{T}_h^{m-1}$ and $\mathbb{T}_h^m$. For the corresponding finite element space $\mathcal{V}_h^{s,m-1/2}$ there holds $\mathcal{V}_h^{s,m-1} \subset \mathcal{V}_h^{s,m-1/2}$ and $\mathcal{V}_h^{s,m} \subset \mathcal{V}_h^{s,m-1/2}$. The nodal basis functions of $\mathcal{V}_h^{s,m-1}$ and of $\mathcal{V}_h^{s,m}$ can be expressed as linear combinations of the basis functions of $\mathcal{V}_h^{s,m-1/2}$. Due to this fact integrals can be evaluated cell-wise on $\mathbb{T}_h^{m-1/2}$, see the discussion in [20] and also [5] for more details.

The dual weighted residual method

We consider an output functional $J : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which represents the quantity of physical interest:

$$J(u) = \int_0^T J_1\big(u(t)\big)\, dt + J_2\big(u(T)\big),$$

where $J_1 : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ and $J_2 : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ are three times continuously differentiable functionals. The functional J_1 or J_2 may be zero. If u_{kh} is the solution of the discrete problem (11), the goal of this section is the a posteriori estimation with respect to J in the following form:

$$J(u) - J(u_{kh}) \approx \eta_k + \eta_h.$$

The quantity η_k describes the error due to the time discretization and η_h the error due to the discretization in space. We start with an abstract result and refer to [7] for more details.

Theorem 1 (Becker & Rannacher [11]). *Let $u \in \mathcal{V}$ the solution of the variational problem*

$$A(u)(\psi) = 0 \quad \text{for all} \quad \psi \in \mathcal{V}.$$

This equation is approximated by a Galerkin method using a finite dimensional subspace $\mathcal{V}_h \subset \mathcal{V}$:

$$u_h \in \mathcal{V}_h : \quad A(u_h)(\psi_h) = 0 \quad \text{for all} \quad \psi_h \in \mathcal{V}_h.$$

If J is a three-times differentiable functional on $\mathcal{V}$ and a pair $\{u, z\} \in \mathcal{V} \times \mathcal{V}$ (the primal and the dual solution) fulfills

$$A'(u)(\phi, z) = J'(u)(\phi),$$

$$A(u)(\psi) = F(\psi),$$

for all test functions $\{\phi,\psi\} \in \mathcal{V}\times\mathcal{V}$, there holds the error representation

$$J(u) - J(u_h) = \frac{1}{2}\rho(u_h)(z-\psi_h) + \frac{1}{2}\rho^*(u_h,z_h)(u-\phi_h) + R_h^{(3)}$$

for arbitrary $\varphi_h, \psi_h \in \mathcal{V}_h$. The primal and dual residuals are

$$\rho(u_h)(\cdot) := F(\cdot) - A(u_h)(\cdot),$$

$$\rho^*(u_h,z_h)(\cdot) := J'(u_h)(\cdot) - A'(u_h)(\cdot,z_h),$$

and the remainder is cubic in the primal and dual errors $e := u - u_h$ and $e^* := z - z_h$:

$$R_h^{(3)} = \frac{1}{2}\int_0^1 \Big\{ J'''(u_h+se)(e,e,e) - A'''(u_h+se)(e,e,e,z_h+se^*) \\ - 3A''(u_h+se)(e,e,e^*)\Big\} s(1-s)\,ds$$

The derivative

$$A'(u)(\phi,z) := \lim_{\tau\neq 0,\tau\to 0} \tau^{-1}\big\{A(u+\tau\phi)(z) - A(u)(z)\big\}$$

is the so-called Gâteaux derivative with respect to u. To apply Theorem 1, we require that the output functional J is three-times differentiable.

In view of Theorem 1 and summing up both equations, problem (9) is equivalent to: Find $u = \{u^0,u^1\} \in \mathcal{X}\times\mathcal{X}$ with

$$A(\!(u)\!)(\varphi)) := (\!(\partial_t u^0,\varphi^0)\!) - A_2(\!(u)\!)(\varphi^0)) + (\!(\partial_t u^1,\varphi^1)\!) + A_1(\!(u)\!)(\varphi^1)) \\ + \big(u^0(0),\varphi_0^{0,-}\big)_G + \big(u^1(0),\varphi_0^{1,-}\big)_G = \big(u_0^0,\varphi_0^{0,-}\big)_G + \big(u_0^1,\varphi_0^{1,-}\big)_G =: F(\varphi) \quad (16)$$

for all test functions $\varphi = \{\varphi^0,\varphi^1\} \in \mathcal{X}\times\mathcal{X}$. The error of the output functional can be represented as

$$J(u) - J(u_{kh}) = J(u) - J(u_k) + J(u_k) - J(u_{kh}).$$

Applying Theorem 1 and neglecting the remainder terms, this may be rewritten as

$$J(u) - J(u_k) \approx \frac{1}{2}\big(\rho(u_k)(z-\tilde z_k) + \rho^*(u_k,z_k)(u-\tilde u_k)\big),$$

$$J(u_k) - J(u_{kh}) \approx \frac{1}{2}\big(\rho(u_{kh})(z_k-\tilde z_{kh}) + \rho^*(u_{kh},z_{kh})(u_k-\tilde u_{kh})\big).$$

The functions $\{\tilde u_k,\tilde z_k\} \in \big(\widetilde{\mathcal{X}}_k^r \times \widetilde{\mathcal{X}}_k^r\big)\times\big(\widetilde{\mathcal{X}}_k^r \times \widetilde{\mathcal{X}}_k^r\big)$ and $\{\tilde u_{kh},\tilde z_{kh}\} \in \big(\widetilde{\mathcal{X}}_{kh}^{r,s} \times \widetilde{\mathcal{X}}_{kh}^{r,s}\big)\times\big(\widetilde{\mathcal{X}}_{kh}^{r,s} \times \widetilde{\mathcal{X}}_{kh}^{r,s}\big)$ can be chosen arbitrarily.

Because the functions $\tilde u_k$, $\tilde z_k$, $\tilde u_{kh}$ ad $\tilde z_{kh}$ can be chosen arbitrarily, the so-called *weights* $u-\tilde u_k$ and $z-\tilde z_h$ are essentially interpolation errors and we approximate these errors by higher order reconstruction from the discrete solutions. Therefore, we introduce the operators Π_k and Π_h, which map computed solutions to approximations of the interpolation errors:

$$z - z_k \approx \Pi_k z_k, \qquad u - u_k \approx \Pi_k u_k,$$

$$z_k - z_{kh} \approx \Pi_h z_{kh}, \qquad u_k - u_{kh} \approx \Pi_h u_{kh}.$$

We use a patch-wise higher-order interpolation: in time piece-wise quadratic interpolation,

$$\Pi_k u_k := I_{2k}^{(2)} u_k - u_k$$

and in space

$$I_{2h}^{(2l)} : \mathcal{V}_h^l \to \mathcal{V}_{2h}^{2l}, \qquad \Pi_h u_{kh} := I_{2h}^{(2l)} u_{kh} - u_{kh}.$$

We recall our assumption that the underlying mesh consists of patches. The piece-wise quadratic interpolation operator in time can be explicitly written:

$$I_{2k}^{(2)} v(t) := \frac{(t_m - t)(t_{m+1} - t)}{k_m(k_m + k_{m+1})} v(t_{m-1}) + \frac{(t - t_{m-1})(t_{m+1} - t)}{k_m k_{m+1}} v(t_m) + \frac{(t - t_{m-1})(t_m - t)}{(k_m + k_{m+1})k_{m+1}} v(t_{m+1})$$

for $t \in (t_{m-1}, t_{m+1}]$. For the practical evaluation of the derived error estimator, we approximate the semidiscrete solutions u_k, z_k by the fully discrete ones:

$$\rho(u_k)(z - z_k) \approx \rho(u_{kh})(\Pi_k z_{kh}),$$

$$\rho^*(u_k, z_k)(u - u_k) \approx \rho^*(u_{kh}, z_{kh})(\Pi_k u_{kh}),$$

and we get at an error representation of the form

$$J(u) - J(u_{kh}) \approx \eta_k + \eta_h,$$

where

$$\eta_k := \frac{1}{2}\big(\rho(u_{kh})(\Pi_k z_{kh}) + \rho^*(u_{kh}, z_{kh})(\Pi_k u_{kh})\big),$$

$$\eta_h := \frac{1}{2}\big(\rho(u_{kh})(\Pi_h z_{kh}) + \rho^*(u_{kh}, z_{kh})(\Pi_h u_{kh})\big)$$

are the contributions of the temporal and the spatial discretization errors.

The dual problem

The evaluation of the error indicators requires the calculation of residuals involving the derivative $A'(u_h)(u - \phi_h, z_h)$. For the derivative of the functional, we introduce the following notation:

$$J'(u)(\varphi) = j_1(u)(\varphi^0) + j_2(u)(\varphi^1), \qquad \varphi = \{\varphi^0, \varphi^1\} \in \mathcal{X} \times \mathcal{X}.$$

Defining the space-time semilinear forms

$$B_1(\!(u)(\phi, z)\!) := \int_0^T B_1(u)(\phi, z)\, dt, \qquad B_2(\!(u)(\phi, z)\!) := \int_0^T B_2(u)(\phi, z)\, dt,$$

with $B_j : \mathcal{V} \times \mathcal{V} \times \mathcal{V} \times \mathcal{V} \times \mathcal{V} \to \mathbb{R}$, $j = 1, 2$,

$$\begin{aligned} B_1(u)(\phi, z) := {} & \frac{1}{2}\big(\nabla\phi, \nabla z^0\big) + \big(\phi, V z^0\big) - \Big(\phi, 2f'(|u|^2)(u^1 z^0 - u^0 z^1)u^1 + f(|u|^2) z^0\Big) \\ & - \Omega\big(\phi, L(\nabla) z^1\big), \\ B_2(u)(\phi, z) := {} & \frac{1}{2}\big(\nabla\phi, \nabla z^1\big) + \big(\phi, V z^1\big) - \Big(\phi, 2f'(|u|^2)(u^0 z^1 - u^1 z^0)u^0 + f(|u|^2) z^1\Big) \\ & + \Omega\big(\phi, L(\nabla) z^0\big), \end{aligned}$$

the derivative of the variational form in (16) can be computed to:

$$A'((u)(\varphi,z)) = ((\partial_t\varphi^0, z^0)) + B_2((u)(\varphi^0,z)) + ((\partial_t\varphi^1, z^1)) - B_1((u)(\varphi^1,z)) \\ + \big(\varphi^0(0), z^0(0)\big) + \big(\varphi^1(0), z^1(0)\big)$$

for all $\varphi = \{\varphi^0, \varphi^1\} \in \mathcal{X} \times \mathcal{X}$. We see that the variational problem for computing the dual solution $z = \{z^0, z^1\} \in \mathcal{X} \times \mathcal{X}$ reads

$$((\partial_t\varphi^0, z^0)) + B_2((u)(\varphi^0,z)) + \big(\varphi^0(0), z^0(0)\big) = j_1(u)(\varphi^0),$$
$$((\partial_t\varphi^1, z^1)) - B_1((u)(\varphi^1,z)) + \big(\varphi^1(0), z^1(0)\big) = j_2(u)(\varphi^1),$$

for all $\varphi = \{\varphi^0, \varphi^1\} \in \mathcal{X} \times \mathcal{X}$. Integrating by parts in each time interval leads to

$$-((\varphi^0, \partial_t z^0)) + \big(\varphi^0(t), z^0(t)\big)\Big|_{t=0}^{t=T} + B_2((u)(\varphi^0,z)) + \big(\varphi^0(0), z^0(0)\big) = j_1(u)(\varphi^0),$$
$$-((\varphi^1, \partial_t z^1)) + \big(\varphi^1(t), z^1(t)\big)\Big|_{t=0}^{t=T} - B_1((u)(\varphi^1,z)) + \big(\varphi^1(0), z^1(0)\big) = j_2(u)(\varphi^1),$$

and we arrive at the following form for dual variational problem: Find a solution $z = \{z^0, z^1\} \in \mathcal{X} \times \mathcal{X}$ satisfying

$$\begin{aligned} -((\varphi^0, \partial_t z^0)) + B_2((u)(\varphi^0,z)) + \big(\varphi^0(T), z^0(T)\big) &= j_1(u)(\varphi^0), \\ -((\varphi^1, \partial_t z^1)) - B_1((u)(\varphi^1,z)) + \big(\varphi^1(T), z^1(T)\big) &= j_2(u)(\varphi^1), \end{aligned} \tag{17}$$

for all $\varphi = \{\varphi^0, \varphi^1\} \in \mathcal{X} \times \mathcal{X}$. We define the operator for the dual problem to

$$\mathscr{B}_1(u)(z) := -\frac{1}{2}\Delta z^0 + Vz^0 - \Big(2f'(|u|^2)(u^1z^0 - u^0z^1)u^1 + f(|u|^2)z^0\Big) - \Omega L(\nabla)z^1,$$
$$\mathscr{B}_2(u)(z) := -\frac{1}{2}\Delta z^1 + Vz^1 - \Big(2f'(|u|^2)(u^0z^1 - u^1z^0)u^0 + f(|u|^2)z^1\Big) + \Omega L(\nabla)z^0,$$

and the equations determining the dual solution read:

$$\begin{aligned} -\partial_t z^0 + \mathscr{B}_2(u)(z) &= 0, \\ -\partial_t z^1 - \mathscr{B}_1(u)(z) &= 0, \end{aligned} \qquad x \in G, \quad t \in I,$$

with terminal conditions

$$z^0(x,T) = j_1(u(T)), \qquad z^1(x,T) = j_2(u(T)), \qquad x \in G.$$

The dual problem is a modified system of Schrödinger's type but running back in time and involving of the primal solution. That makes it difficult to solve. First, one has to solve the primal system for all time steps and then one has to go backwards in time and compute the dual solutions. The dual solution transports information backwards in time. This especially means that one has to store the primal solution for all time steps, at least as long as one has not computed the dual solution for this time step.

Explicit form of error bounds

The last step to obtain a cell-wise error estimator which can be used for mesh refinement is the localization of the error bounds to time and space cells. We start with the computation of the temporal contributions

$$\eta_k = \sum_{m=1}^{M} \eta_k^m = \frac{1}{2}\big(\rho(u_{kh})(\Pi_k z_{kh}) + \rho^*(u_{kh}, z_{kh})(\Pi_k u_{kh})\big)$$

with

$$\rho(u_{kh})(\Pi_k z_{kh}) = -\sum_{m=1}^{M} \int_{I_m} \Big\{ \big(\partial_t u_{kh}^0(t), \Pi_k z_{kh}^0(t)\big) - A_2\big(u_{kh}(t)\big)\big(\Pi_k z_{kh}^0(t)\big) \\ + \big(\partial_t u_{kh}^1(t), \Pi_k z_{kh}^1(t)\big) + A_1\big(u_{kh}(t)\big)\big(\Pi_k z_{kh}^1(t)\big)\Big\}\, dt$$

and

$$\rho^*(u_{kh}, z_{kh})(\Pi_k u_{kh}) = j_1(u_{kh})(\Pi_k u_{kh}^0) + j_2(u_{kh})(\Pi_k u_{kh}^1) \\ - \sum_{m=1}^{M} \int_{I_m} \Big\{ -\big(\Pi_k u_{kh}^0(t), \partial_t z_{kh}^0(t)\big) + B_2\big(u_{kh}(t)\big)\big(\Pi_k u_{kh}^0(t), z_{kh}(t)\big) \\ - \big(\Pi_k u_{kh}^1(t), \partial_t z_{kh}^1(t)\big) - B_1\big(u_{kh}(t)\big)\big(\Pi_k u_{kh}^1(t), z_{kh}(t)\big)\Big\}\, dt$$

We approximate the temporal integrals involving u_{kh}^j and z_{kh}^j by the trapezoidal rule whereas those involving the quadratic terms $I_{2k}^{(2)} u_{kh}^j$ and $I_{2k}^{(2)} z_{kh}^j$ are evaluated with the three-point Newton-Côtes quadrature rule (Simpson's rule). We use the notation

$$U_m^j(x) := u_{kh}^j(x, t_m), \qquad Z_m^j(x) := z_{kh}^j(x, t_m),$$

the point $t_{m-1/2} := \frac{1}{2}(t_{m-1} + t_m)$ is the midpoint of the time interval and

$$U_{m-1/2}^j(x) := \frac{1}{2}\big(U_{m-1}^j(x) + U_m^j(x)\big), \qquad Z_{m-1/2}^j(x) := \frac{1}{2}\big(Z_{m-1}^j(x) + Z_m^j(x)\big)$$

for $j \in \{0, 1\}$. The solutions u_k and z_k are (piecewise) linear polynomials in time and the function value at the midpoint $t_{m-1/2}$ is the average. We outline how to compute the integrals. For two scalar functions ϕ, ψ, linear in the interval I_m, the three-point Newton-Cotes quadrature rule leads to

$$\int_{I_m} \phi(t) I_{2k}^{(2)} \psi(t)\, dt = \frac{k_m}{6}\Big(\phi(t_{m-1}) I_{2k}^{(2)} \psi(t_{m-1}) + 4\phi(t_{m-1/2}) I_{2k}^{(2)} \psi(t_{m-1/2}) + \phi(t_m) I_{2k}^{(2)} \psi(t_m)\Big).$$

At the end points of the interval, there holds

$$I_{2k}^{(2)} \psi(t_{m-1}) = \psi_{m-1}, \qquad I_{2k}^{(2)} \psi(t_m) = \psi_m,$$

and the value at the midpoint,

$$\widetilde{\psi}_{m-1/2} := I_{2k}^{(2)} \psi(t_{m-1/2}),$$

can be computed from the definition of the interpolation operator: If $m-1 \in 2\mathbb{N}$, we have

$$\widetilde{\psi}^j_{m-1/2} = \frac{(k_m + 2k_{m+1})}{4(k_m + k_{m+1})}\psi^j_{m-1} + \frac{(k_m + 2k_{m+1})}{4k_{m+1}}\psi_m - \frac{k_m k_m}{4(k_m + k_{m+1})k_{m+1}}\psi_{m+1},$$

if $m \in 2\mathbb{N}$, we have

$$\widetilde{\psi}^j_{m-1/2} = -\frac{k_m k_m}{4k_{m-1}(k_{m-1} + k_m)}\psi^j_{m-2} + \frac{(2k_{m-1} + k_m)}{4k_{m-1}}\psi^j_{m-1} + \frac{(2k_{m-1} + k_m)}{4(k_{m-1} + k_m)}\psi^j_m.$$

Putting all this together shows

$$\int_{I_m} \phi(t)\Pi_k\psi(t)\,dt = \frac{k_m}{3}\Big(2\phi_{m-1/2}\widetilde{\psi}_{m-1/2} - \phi_{m-1}\psi_{m-1} - \phi_m\psi_m\Big).$$

The integrals involving derivatives in time calculate to

$$\int_{I_m} \big(\partial_t\phi, \Pi_k\psi\big)\,dt = \frac{1}{3}\Big(2\big(\phi_m - \phi^j_{m-1}, \widetilde{\psi}_{m-1/2}\big) - \big(\phi_m - \phi_{m-1}, \psi_{m-1} + \psi_m\big)\Big),$$

We put

$$d_m U^j_m = \frac{U^j_m - U^j_{m-1}}{k_m}, \qquad d_m Z^j_m = \frac{Z^j_m - Z^j_{m-1}}{k_m},$$

and using the previous calculations, the temporal error indicators have the explicit representation

$$\begin{aligned}
\eta^m_k = \frac{1}{2}\frac{k_m}{3}\Bigg\{ &\sum_{j=0}^{1}\Big\{\big(2\widetilde{U}^j_{m-1/2} - U^j_{m-1} - U^j_m, d_m Z^j_m\big) - \big(d_m U^j_m, 2\widetilde{Z}^j_{m-1/2} - Z^j_{m-1} - Z^j_m\big)\Big\} \\
&+ \Big\{\Big(2A_2(U_{m-1/2})\big(\widetilde{Z}^0_{m-1/2}\big) - \big(A_2(U_{m-1})(Z^0_{m-1}) + A_2(U_m)(Z^0_m)\big)\Big) \\
&- \Big(2A_1(U_{m-1/2})\big(\widetilde{Z}^1_{m-1/2}\big) - \big(A_1(U_{m-1})(Z^1_{m-1}) + A_1(U_m)(Z^1_m)\big)\Big) \\
&+ \Big(2j_1(U_{m-1/2})(\widetilde{U}^0_{m-1/2}) - j_1(U_{m-1})(U^0_{m-1}) - j_1(U_{m-1/2})(U^0_m)\Big) \\
&+ \Big(2j_2(U_{m-1/2})(\widetilde{U}^0_{m-1/2}) - j_2(U_{m-1})(U^0_{m-1}) - j_2(U_{m-1/2})(U^0_m)\Big) \\
&+ \Big(2B_1(U_{m-1/2})(\widetilde{U}^1_{m-1/2}, Z_{m-1/2}) - \big(B_1(U_{m-1})(U^1_{m-1}, Z_{m-1}) + B_1(U_m)(U^1_m, Z_m)\big)\Big) \\
&- \Big(2B_2(U_{m-1/2})(\widetilde{U}^0_{m-1/2}, Z_{m-1/2}) - \big(B_2(U_{m-1})(U^0_{m-1}, Z_{m-1}) + B_2(U_m)(U^0_m, Z_m)\big)\Big)\Big\}\Bigg\}.
\end{aligned}$$

The next step is the calculation of the spatial contributions

$$\eta_h = \sum_{m=0}^{M}\eta^m_h = \frac{1}{2}\big(\rho(u_{kh})(\Pi_h z_{kh}) + \rho^*(u_{kh}, z_{kh})(\Pi_h u_{kh})\big)$$

which reads

$$\eta_h = \sum_{m=0}^{M} \eta_h^m = \frac{1}{2}\Big((u_0^0 - U_0^0, \Pi_h Z_0^0) + (u_0^1 - U_0^1, \Pi_h Z_0^1) \Big)$$

$$+ \frac{1}{2} \sum_{m=1}^{M} \frac{k_m}{2} \Bigg\{ \sum_{j=0}^{1} \Big\{ \big(\Pi_h U_{m-1/2}^j, d_m Z_m^j\big) - \big(d_m U_m^j, \Pi_h Z_{m-1/2}^j\big) \Big\}$$

$$+ \Big\{ A_2(U_{m-1})(\Pi_h Z_{m-1}^0) + A_2(U_m)(\Pi_h Z_m^0) - A_1(U_{m-1})(\Pi_h Z_{m-1}^1) - A_1(U_m)(\Pi_h Z_m^1)_K \Big\}$$

$$+ \Big\{ + j_1(U_{m-1})(\Pi_h Z_{m-1}^0) + j_1(U_m)(\Pi_h Z_m^0) + j_2(U_{m-1})(\Pi_h Z_{m-1}^1) + j_2(U_m)(\Pi_h Z_m^1) \Big\}$$

$$+ \Big\{ B_1(U_{m-1})(\Pi_h U_{m-1}^1, Z_{m-1}) + B_1(U_m)(\Pi_h U_m^1, Z_m)$$

$$- B_2(U_{m-1})(\Pi_h U_{m-1}^0, Z_{m-1}) - B_2(U_m)(\Pi_h U_m^0, Z_m) \Big\} \Bigg\}.$$

As mentioned previously, we see here that the spatial error indicators depend linearly on the time step length. To get error indicators which can be used for mesh refinement, this spatial contributions have to be localized further. As widely discussed in the literature, there exist several strategies. Here we follow a standard procedure leading to cell-wise contributions and we refer to [10], [7] and the literature cited there for more details. Summing up over all cells,

$$\sum_{m=0}^{M} \eta_h^m = \sum_{m=0}^{M} \sum_{K \in \mathbb{T}_h^m} \eta_{h,K}^m,$$

we integrate by parts on each single cell. In the spatial semilinear forms this approach leads to contributions

$$\begin{aligned} A_{2-j}(u_{kh})(\varphi^j)\Big|_K &= -\frac{1}{2}\big(\Delta u_{kh}^{1-j}, \varphi^j\big)_K + \big(V u_{kh}^{1-j}, \varphi^j\big)_K - \big(f\big(|u_h|^2\big) u_{kh}^{1-j}, \varphi^j\big)_K \\ &\quad + (-1)^j \Omega \big(L(\nabla) u_{kh}^j, \varphi^j\big)_K + \frac{1}{2}\big(\nabla_n u_{kh}^{1-j}, \varphi^j\big)_{\partial K} \\ &= \big(\mathscr{A}_{2-j}(u_{kh}), \varphi^j\big)_K + \frac{1}{4}\big([\nabla_n u_{kh}^{1-j}], \varphi^j\big)_{\partial K}, \qquad j \in \{0,1\}. \end{aligned}$$

Here, $\big[\nabla_n u_{kh}\big] := \big(\nabla u_{kh}|_{\partial K' \cap \Gamma} - \nabla u_{kh}|_{\partial K \cap \Gamma}\big) \cdot n$ denotes the jump of the gradient ∇u_{kh} across a common cell interface Γ of two neighbor cells $K, K' \in \mathbb{T}_h$ into the direction of the normal unit vector n pointing from K to K'. The cell-wise spatial error contributions can be computed now from the representations

$$\Big(A_2(U_m)(\Pi_h Z_m^0) - A_1(U_m)(\Pi_h Z_m^1)\Big)\Big|_K = \big(\mathscr{A}_2(U_m), \Pi_h Z_m^0\big)_K - \big(\mathscr{A}_1(U_m), \Pi_h Z_m^1\big)_K$$

$$+ \frac{1}{4}\Big\{ \big([\partial_n U_m^1], \Pi_h Z_m^0\big)_{\partial K \setminus \partial G} - \big([\partial_n U_m^0], \Pi_h Z_m^1\big)_{\partial K \setminus \partial G} \Big\}$$

$$\Big(B_1(U_m)(\Pi_h U_m^1, Z_m) - B_2(U_m)(\Pi_h U_m^0, Z_m)\Big)\Big|_K = \big(\mathscr{B}_1(U_m)(Z_m), \Pi_h U_m^1\big)_K$$
$$- \big(\mathscr{B}_2(U_m)(Z_m), \Pi_h U_m^0\big)_K + \frac{1}{4}\Big\{\big([\partial_n Z_m^0], \Pi_h U_m^1\big)_{\partial K\setminus\partial G} - \big([\partial_n Z_m^1], \Pi_h U_m^0\big)_{\partial K\setminus\partial G}\Big\}$$

for $m \geq 1$.

Discretization refinement strategy

Adaptive mesh refinement in time and space always has the goal to reduce the temporal and spatial parts of the discretization errors. This can be done by refining both discretizations as long as the corresponding error contribution is larger than half of a given tolerance to reach

$$J(u) - J(u_{kh}) \approx \eta_k + \eta_h \approx \mathrm{TOL}$$

but this may need many rounds of mesh refinement. We follow a strategy presented in [20] which balances the temporal and spatial discretization error parts:

$$|\eta_k| \approx |\eta_h|.$$

For an efficient equilibrated reduction of the different error parts the temporal error indicators have to be independent of the spatial error indicators and vice versa. This is the case for

$$\eta_k = \sum_{m=1}^{M} \eta_k^m, \qquad \eta_h = \sum_{m=0}^{M} \eta_h^m,$$

but the local contributions

$$\eta_h^m = \sum_{K\in\mathbb{T}_h^m} \eta_{h,K}^m$$

depend linearly on the time step size k_m of the interval I_m as we have seen in the last section. The spatial contributions would decrease, if the time step size decrease but the spatial discretization is fixed. To overcome this, we introduce a reference time step and introduce the error contributions

$$\widetilde{\eta}_{h,K}^m := \frac{T}{M}\frac{1}{k_m}\eta_{h,K}^m, \qquad K \in \mathbb{T}_h^m, \quad m = 0, \ldots, M.$$

This leads to two sets of error indicators which can be used for mesh refinement:

$$\big\{|\eta_k^m| : m = 1, \ldots, M\big\}, \qquad \bigcup_{m=0}^{M} \big\{|\widetilde{\eta}_{h,K}^m| : K \in \mathbb{T}_h^m\big\}$$

There are many different strategies how these error indicators can used for the decision which of the cells have to be refined, see e.g. [12]. We use a "fixed-number" strategy, which means that we refine as many cells as we reach some given growth rate of the meshes.

The adaptive refinement algorithm presented in [20] introduces an equilibration factor c for the decision which of the discretization has to be refined in each adaptation cycle. This factor has to be chosen carefully. If c is chosen too small, the reduction of the error is slow, because only one discretization is refined although both error parts are of the same size. On the other hand, if c is chosen too large, both discretizations will be refined although one part of the error dominates the overall error.

Algorithm 1: Adaptive refinement algorithm

Data: Initial temporal discretization $\mathbb{T}_{k_0}$ and spatial meshes $\mathbb{T}_{h_0}^m$.

begin

  Set $n = 0$.

  while $|\eta| > \mathrm{TOL}$ *and* $\sum_{m=0}^{M_n} N_m < N_{\max}$ **do**

    Compute primal and dual solutions $u_{k_n h_n}$ and $z_{k_n h_n}$.

    Evaluate the error indicators η_{k_n} and η_{h_n}.

    if $|\eta_{k_n}| > c|\eta_{h_n}|$ **then**

      Refine temporal discretization.

    else if $|\eta_{h_n}| > c|\eta_{k_n}|$ **then**

      Refine spatial discretization.

    else if $\frac{1}{c} \leq \frac{|\eta_{k_n}|}{|\eta_{h_n}|} \leq c$ **then**

      Refine temporal and spatial discretizations.

    end

    Increase $n = n + 1$.

  end

end

Running several tests show, that a good choice for the equilibration constant is $c \approx 4$. The algorithm which is presented below stops, if the error $|\eta| = |\eta_k + \eta_h|$ reaches some given tolerance or if the number of degrees of freedom reaches a maximal value $N_{\max}$, which should be determined by the given computer resources.

The presented algorithm is costly when bringing back to mind that in each round of refinement many finite element solutions have to be computed. As mentioned in a previous section, the computed error indicators give a posteriori error bounds for the functional value of interest. These cell-wise indicators point to cells which have to be refined in order to reduce the overall error. But there is no information in these indicators how much the overall error is being reduced by refining a special cell. It is only possible to evaluate these error bounds, refine the cells where these indicators are largest and start a new round until a given precision is not reached. This process is expensive and usually needs several rounds of refinement. But even if computing so many finite element solutions is expensive, we have an error bound for the functional value and only the parts of the time and space disrectizations are being refined, which contribute to the overall error. This is the main benefit over a classical time stepping method.

4. Examples

All the numerical examples presented in this section have been calculated using a program based on the open source C++ finite element library *deal.II*. For more details see [6] and the project website `http://www.dealii.org`.

The Schrödinger equation preserves energy and mass and using the energy as control functional is a very good indicator to show that the refinement scheme works. Computing the energy functional $E\big(u(T)\big)$ at the end of the time interval, we see that the derivative used in the dual problem is

$$J'\big(u\big)(\varphi) = E'\big(u(T)\big)(\varphi) = A_1\big(u(T)\big)(\varphi^0) + A_2\big(u(T)\big)(\varphi^1)$$

We present results for $cG(1)cG(2)$ disrectizations and leave other combinations for future work. In each time step we use a BiCGstab linear solver [23] combined with an algebraic multigrid preconditioner

[19], [24], [21] for computing explicit stages, which involve only the mass matrix here. For implicit stages we use Newton's method and compute solutions of linear systems also with a BiCGstab solver. The used criterion for stopping the Newton iteration is a defect of $1.00 \cdot 10^{-8}$. It turned out that in combination with SOR preconditioning with $\omega = 1$ this was reached in some few steps in all shown results. **The ordering was optimized for the preconditioner and the algorithm of Cuthill and McKee was used with deal.II.**

4.1 Ground state solutions

As a first example, we consider the framework given in [9]. We start with a ground state solution as initial value and compute the wave solution of the Gross-Pitaevskii equation with a harmonic trap potential in two dimensions:

$$\mathrm{i}\partial_t u = -\frac{\varepsilon^2}{2}\Delta u + Vu + \beta|u|^2 u, \qquad V(x) = \frac{1}{2}\big(\gamma_1^2 x_1^2 + \gamma_2^2 x_2^2\big), \qquad G \subset \mathbb{R}^2.$$

If the nonlinearity vanishes, $\beta = 0$, one can directly calculate [9] that a ground state is given by

$$u(x,t) = e^{-\mathrm{i}\mu t}\Phi(x), \qquad \Phi(x) = \frac{1}{\sqrt{\pi}}(\gamma_1\gamma_2)^{1/4} e^{-\frac{\gamma_1 {x_1}^2 + \gamma_2 {x_2}^2}{2}}, \qquad \mu = \frac{1}{2}(\gamma_1 + \gamma_2). \tag{18}$$

We set $\beta = 0$, $\gamma_1 = \gamma_2 = 1$ and choose the domain $G = [-10, 10] \times [-10, 10]$ together with $T = 1$ and before presenting results showing the equilibration of the time and space errors we give a numerical justification for splitting the total discretization error into temporal and spatial parts in Table 1 for $cG(1)cG(2)$ discretizations. With N_m we denote the number of degrees of freedom of the space discretization at time point m, $N := \sum_{m=0}^{M} N_m$ is the overall number and $N_{\max} = \max N_m$ the maximal number of degrees of freedom of the space discretizations. Table 1 demonstrates the independence of the two parts of the error estimator on the refinement of the other part. Remember that η_h represents space errors and η_k time error.

M	$N_{\max}$	$\lvert\eta_h\rvert$		$\lvert\eta_k\rvert$	
		cG(1)	cG(2)	cG(1)	cG(2)
10	2,178	$6.0072 \cdot 10^{-04}$	$2.6357 \cdot 10^{-05}$		
20	2,178	$6.1328 \cdot 10^{-04}$	$3.9285 \cdot 10^{-05}$		
40	2,178	$6.1740 \cdot 10^{-04}$	$3.5041 \cdot 10^{-05}$		
80	2,178	$6.1839 \cdot 10^{-04}$	$3.3185 \cdot 10^{-05}$		
160	2,178	$6.1866 \cdot 10^{-04}$	$3.3501 \cdot 10^{-05}$		
10	2,178			$1.4976 \cdot 10^{-03}$	$1.4976 \cdot 10^{-03}$
10	8,450			$1.3943 \cdot 10^{-03}$	$1.3943 \cdot 10^{-03}$
10	33,282			$1.3704 \cdot 10^{-03}$	$1.3704 \cdot 10^{-03}$
10	132,098			$1.3646 \cdot 10^{-03}$	$1.3646 \cdot 10^{-03}$
10	526,338			$1.3631 \cdot 10^{-03}$	$1.3632 \cdot 10^{-03}$

Table 1: *Example 1:* Independence of the temporal and the spatial part of the error estimator

Next we present the results obtained using the equilibration algorithm. We choose the equilibration factor $c = 4$ and a fraction of 25% of all space cells for refinement in space and 50% of all time cells for refinement in time. If the algorithm provides refinement in time and space, we choose half of both fractions. The ground state fulfills $M(\Phi) = 1$ and the energy can be computed to $E(\Phi) = \frac{1}{2}\mu\|\Phi\|^2 = \frac{1}{2}$.

With $M(u_{kh})$ and $E(u_{kh})$ we denote the computed mass and energy, respectively.
As the results in Table 2 show, the algorithm needs 9 adaptation cycles to reach an error $|\eta| < 1.00{\cdot}10^{-04}$ with the $cG(1)cG(2)$ method. We want to remark that the error indicators are computed for the energy functional and that there is no guarantee that the error in the mass also diminishes in each refinement step. Figure 1 shows that the dynamic refinement in time. As we expect the refinement in time has many ups and downs and the algorithm refines only time steps where it is needed. Because we compute a ground state, the wave evolution is not interesting and we only show one picture of the solution and the space discretization at $t = 0.5$ in Figure 2.

M	N	$N_{\max}$	$\lvert M(u_{kh}) - M(\Phi)\rvert$	$\lvert E(u_{kh}) - E(\Phi)\rvert$	$\lvert\eta_h\rvert$	$\lvert\eta_k\rvert$
10	92,950	8,450	$1.00 \cdot 10^{-04}$	$1.93 \cdot 10^{-04}$	$2.64 \cdot 10^{-05}$	$1.37 \cdot 10^{-03}$
16	143,650	8,450	$1.00 \cdot 10^{-04}$	$1.93 \cdot 10^{-04}$	$4.07 \cdot 10^{-05}$	$4.89 \cdot 10^{-04}$
25	219,700	8,450	$1.00 \cdot 10^{-04}$	$1.93 \cdot 10^{-04}$	$4.41 \cdot 10^{-05}$	$2.25 \cdot 10^{-04}$
38	329,550	8,450	$1.00 \cdot 10^{-04}$	$1.93 \cdot 10^{-04}$	$3.86 \cdot 10^{-05}$	$8.76 \cdot 10^{-05}$
48	577,474	12,610	$2.81 \cdot 10^{-05}$	$1.51 \cdot 10^{-04}$	$6.74 \cdot 10^{-04}$	$9.30 \cdot 10^{-04}$
61	1,060,396	19,714	$3.43 \cdot 10^{-04}$	$1.35 \cdot 10^{-04}$	$3.90 \cdot 10^{-04}$	$4.28 \cdot 10^{-04}$
77	1,904,004	27,754	$7.22 \cdot 10^{-05}$	$3.18 \cdot 10^{-05}$	$1.40 \cdot 10^{-04}$	$7.60 \cdot 10^{-05}$
97	3,516,580	49,282	$1.71 \cdot 10^{-05}$	$5.44 \cdot 10^{-06}$	$1.52 \cdot 10^{-04}$	$8.55 \cdot 10^{-05}$
122	6,616,966	82,250	$5.24 \cdot 10^{-06}$	$1.59 \cdot 10^{-06}$	$7.84 \cdot 10^{-05}$	$3.97 \cdot 10^{-05}$
153	12,264,044	139,082	$2.85 \cdot 10^{-06}$	$3.33 \cdot 10^{-07}$	$5.41 \cdot 10^{-05}$	$2.47 \cdot 10^{-05}$

Table 2: *Example 1:* Dynamic mesh refinement with equilibration

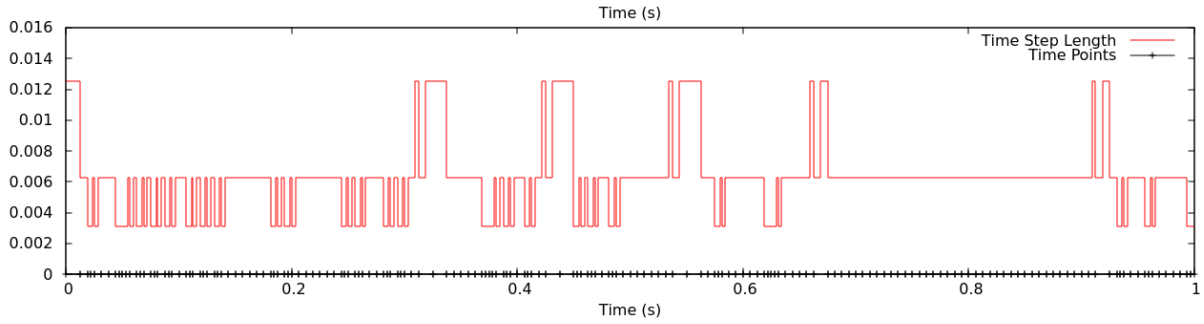

Figure 1: *Example 1:* Dynamic refinement in time for the ground state.

4.2 Adding a repulsive interaction and angular momentum

Now we add a repulsive interaction to the Gross-Pitaevskii by setting $\beta = 5$. We start with the ground state solution (18) with $\mu = 1$ as initial value, normalized to $\|u_0\|^2 = 1$ and extend the time interval to $T = 3$ to see the repulsion. For better comparison, we always run eight adaptation cycles in the following examples.
The adaptive algorithm starts with refinement in time as one can see from Table 3. After four refinement steps in time also the space meshes are refined further. The equilibration of the spatial and temporal parts of the error estimator can be clearly seen. The computed energy after eight rounds of adaptation is $E(u_{kh}) = 0.698839$ with an error estimate $|\eta| < 1.3068 \cdot 10^{-03}$, the mass is conserved up to four digits.
What we see from the picture below in Figure 3 and also in the following examples is, that the number of degrees of freedom always grows with time. As mentioned in a previous chapter, if we add a new

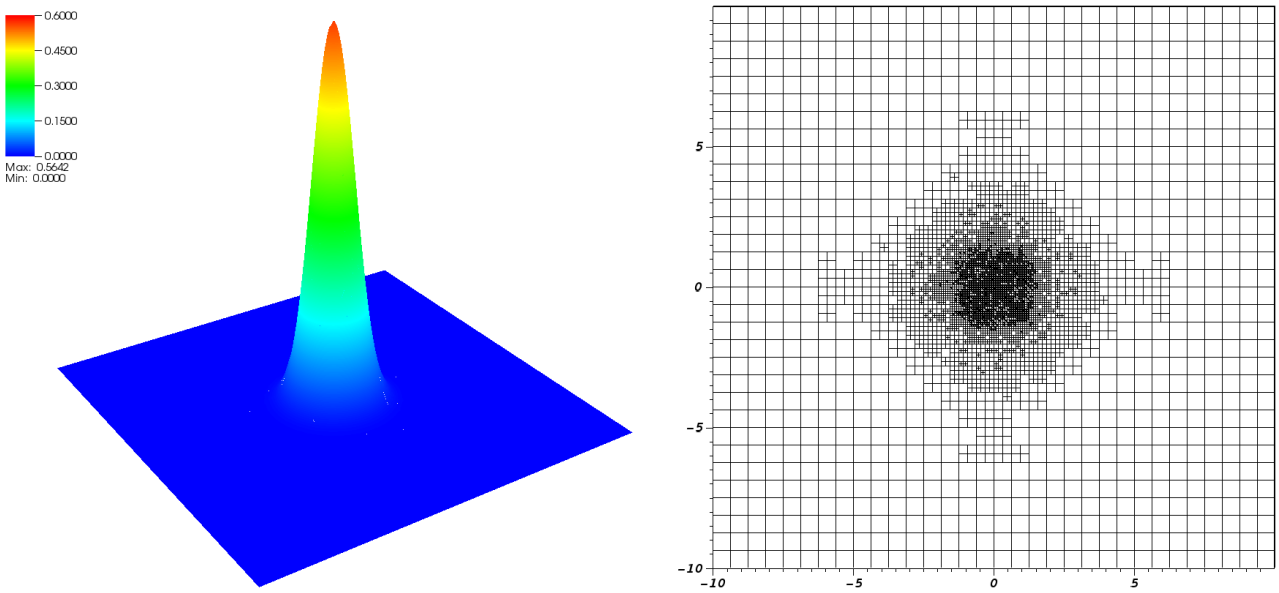

Figure 2: *Example 1:* mass $|u|^2$ and dynamically refined mesh, $t = 0.5$

time step, we always insert a union of the meshes from the neighbor time steps in order to realize dynamical mesh refinement. The mesh in an inserted time step is the set of the finest cells from both neighbor time step. A new space grid is always finer and, even if it is possible that the mesh will be getting coarser in a later step, the number of degrees of freedom has the tendency to grow with time. We do not see a way to overcome when allowing dynamic mesh refinement.

M	N	$N_{\max}$	$M(u_{kh})$	$E(u_{kh})$	$\lvert\eta_h\rvert + \lvert\eta_k\rvert$	$\lvert\eta_h\rvert$	$\lvert\eta_k\rvert$
30	261,950	8,450	1.000363	0.699355	$6.48 \cdot 10^{-02}$	$1.24 \cdot 10^{-03}$	$6.36 \cdot 10^{-02}$
46	397,150	8,450	1.000121	0.698994	$2.63 \cdot 10^{-02}$	$1.55 \cdot 10^{-03}$	$2.48 \cdot 10^{-02}$
70	599,950	8,450	1.000016	0.698917	$1.34 \cdot 10^{-02}$	$1.85 \cdot 10^{-03}$	$1.15 \cdot 10^{-02}$
106	904,150	8,450	1.000039	0.698934	$7.11 \cdot 10^{-03}$	$2.23 \cdot 10^{-03}$	$4.88 \cdot 10^{-03}$
133	1,572,892	13,218	1.000053	0.699076	$4.98 \cdot 10^{-03}$	$1.79 \cdot 10^{-03}$	$3.19 \cdot 10^{-03}$
167	2,845,136	21,122	0.999459	0.698387	$3.19 \cdot 10^{-03}$	$1.06 \cdot 10^{-03}$	$2.14 \cdot 10^{-03}$
209	5,112,268	32,178	0.999790	0.698688	$2.20 \cdot 10^{-03}$	$8.67 \cdot 10^{-04}$	$1.34 \cdot 10^{-03}$
262	9,145,566	54,562	0.999935	0.698839	$1.31 \cdot 10^{-03}$	$4.56 \cdot 10^{-04}$	$8.51 \cdot 10^{-04}$

Table 3: *Example 2:* Dynamic mesh refinement with equilibration, $\beta = 5$, $\Omega = 0$

In the next step we add angular momentum and set $\Omega = 1$, so called *slow speed*. The behavior of the adaptive algorithm and the resulting dynamically refined meshes as well as the obtained results in Table 4 only change slightly. The adaptive refined meshes in space "follow" the rotation of the system, as one can see from Figure 11, but the rotation has only a minor influence here.

4.3 Comparison with adaptive methods

Checking the literature, we considered [1] as a similar approach, where a the fully discrete Galerkin method with fixed point iterations for nonlinearities is given. To compare with our algorithm, we implemented a rotational version using the software package *FreeFem++*, see `http://www.freefem.org`, To compare it with time adaptive mesh, we must choose fixed time steps. From Figure 3, we choose the three most common values $\Delta t = 0.025, 0.0125, 0.00625$, which corresponds to $M = 120, 240, 480$.

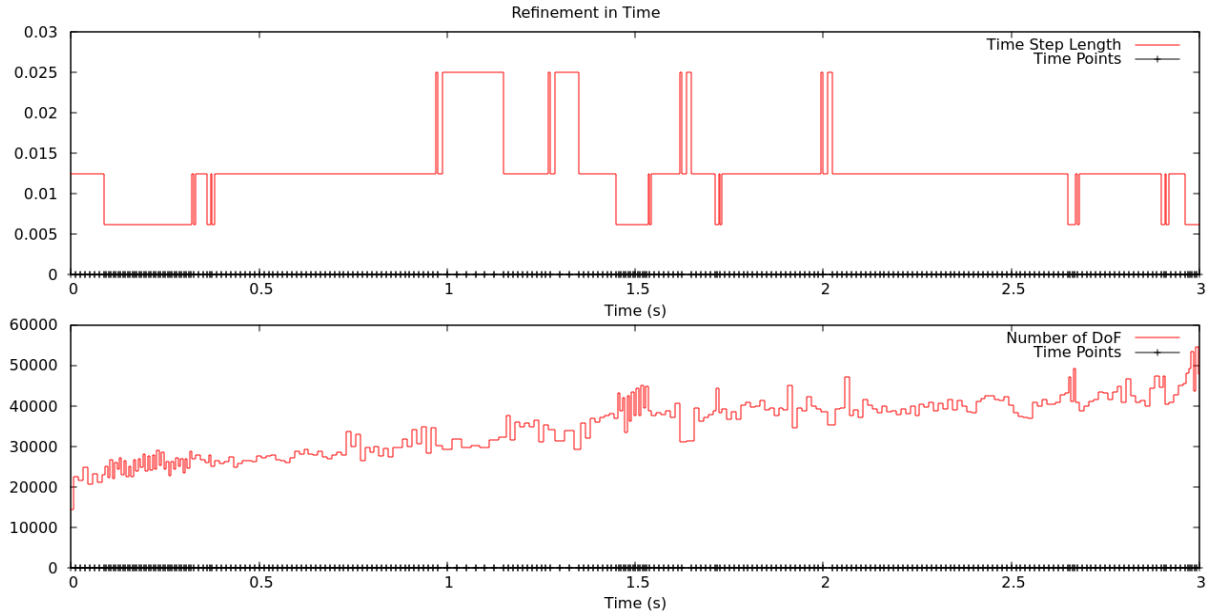

Figure 3: *Example 2:* Dynamic refinement in time with repulsive element.

M	N	$N_{\max}$	$M(u_{kh})$	$E(u_{kh})$	$\lvert\eta_h\rvert + \lvert\eta_k\rvert$	$\lvert\eta_h\rvert$	$\lvert\eta_k\rvert$
30	261,950	8,450	1.000363	0.699355	$6.48 \cdot 10^{-02}$	$1.22 \cdot 10^{-03}$	$6.36 \cdot 10^{-02}$
46	397,150	8,450	1.000122	0.698994	$2.63 \cdot 10^{-02}$	$1.52 \cdot 10^{-03}$	$2.48 \cdot 10^{-02}$
70	599,950	8,450	1.000016	0.698917	$1.34 \cdot 10^{-02}$	$1.89 \cdot 10^{-03}$	$1.16 \cdot 10^{-02}$
106	904,150	8,450	1.000039	0.698934	$7.12 \cdot 10^{-03}$	$2.24 \cdot 10^{-03}$	$4.88 \cdot 10^{-03}$
133	1,587,532	13,314	0.998848	0.697978	$5.21 \cdot 10^{-03}$	$1.99 \cdot 10^{-03}$	$3.22 \cdot 10^{-03}$
167	2,878,008	21,506	0.999501	0.698312	$3.22 \cdot 10^{-03}$	$1.15 \cdot 10^{-03}$	$2.08 \cdot 10^{-03}$
209	5,143,108	32,130	0.999835	0.698730	$2.21 \cdot 10^{-03}$	$8.87 \cdot 10^{-04}$	$1.33 \cdot 10^{-03}$
262	9,022,258	53,826	0.999934	0.698833	$1.38 \cdot 10^{-03}$	$5.57 \cdot 10^{-04}$	$8.25 \cdot 10^{-04}$

Table 4: *Example 3:* Dynamic mesh refinement with equilibration, $\beta = 5$, $\Omega = 1$

To refine the mesh, we first adapt it up to error 1.2068×10^{-03} and next we use the fixed point step with error 10^{-07} to approximate the nonlinearities. Table 5 shows the total number of triangles N, the maximum number of triangles in time N_{max}, mass and energy minimum values and average errors. The detailed numerical behavior given by the adaptive time mesh are not shown, but we will describe it in each case.

Δt	N	$N_{\max}$	$\min M_a(u_k)$	$\min E_a(u_k)$	$\lvert M(u_{kh}) - M_a(u_k)\rvert$	$\lvert E(u_{kh}) - E_a(u_k)\rvert$
0.02500	2,143,240	17,983	1.002531	0.692712	$1.36 \cdot 10^{-04}$	$7.42 \cdot 10^{-03}$
0.01250	4,297,525	17,984	1.002483	0.692493	$1.24 \cdot 10^{-04}$	$8.13 \cdot 10^{-04}$
0.00625	8,598,781	17,984	1.000023	0.692366	$1.31 \cdot 10^{-04}$	$1.51 \cdot 10^{-03}$

Table 5: *Example 3:* Adaptive mesh refinement comparison , $\beta = 5$, $\Omega = 1$

The results are the following. For $\Delta t = 0.025$, the adaptive method has $N = 2,143,240$ elements and $N_{max} = 17,978$ for all time steps except from two outliers: $t = 0.000$ and at $t = 2.225$. In case of mass, its value is also almost steady in time steps at 1.00268 with an error of 10^{-04}, except for an outlier at

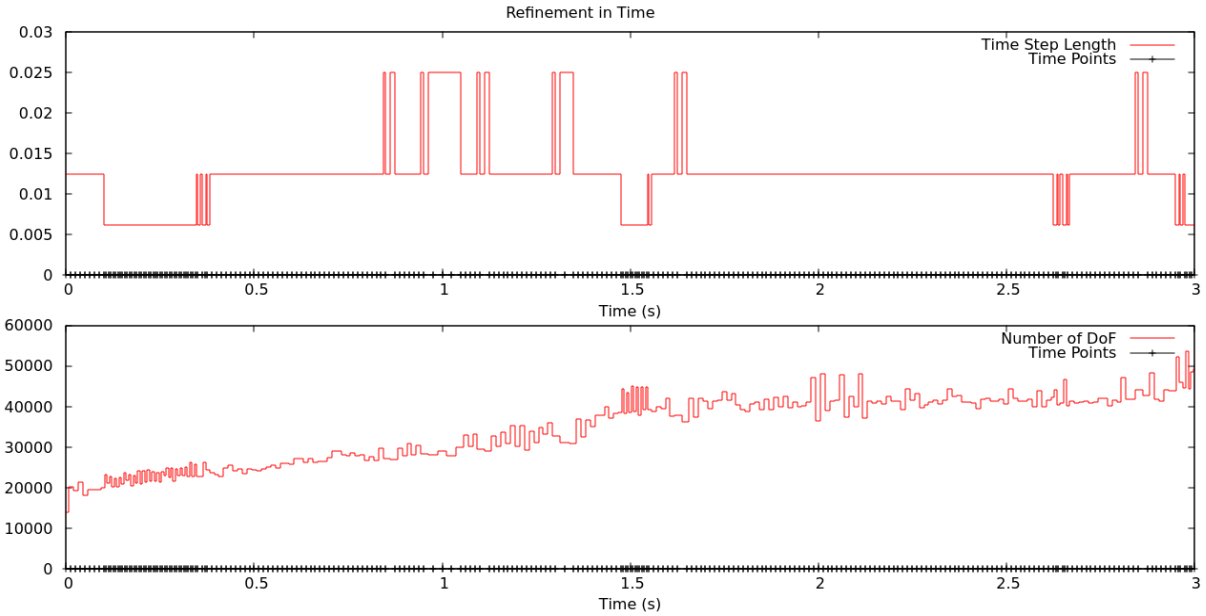

Figure 4: *Example 3:* Dynamic refinement in time with nonsymmetrical potential.

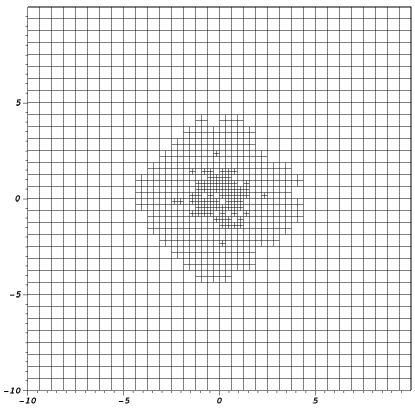

Figure 5: * $t = 0.00$

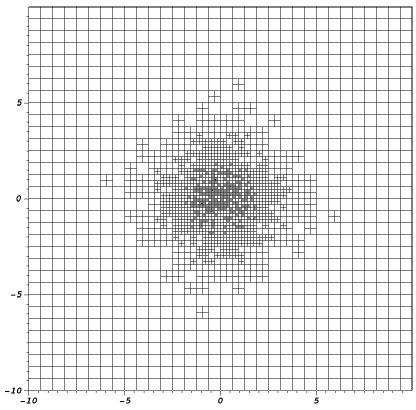

Figure 6: * $t = 0.75$

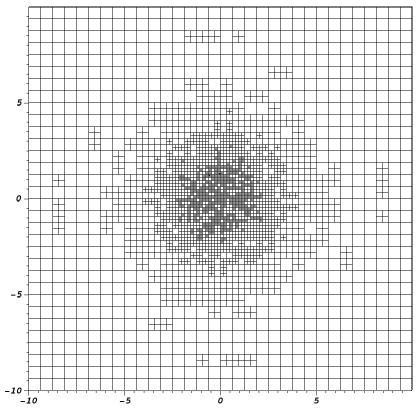

Figure 7: * $t = 1.45$

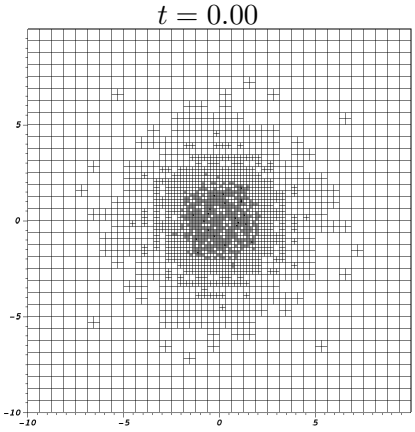

Figure 8: * $t = 1.55$

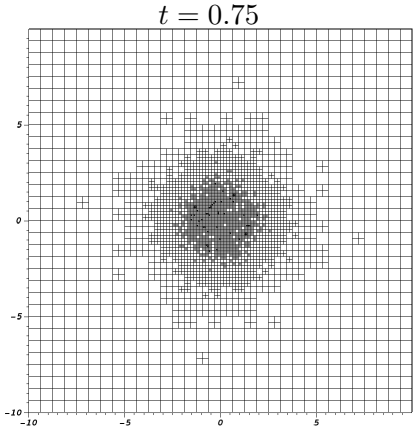

Figure 9: * $t = 2.25$

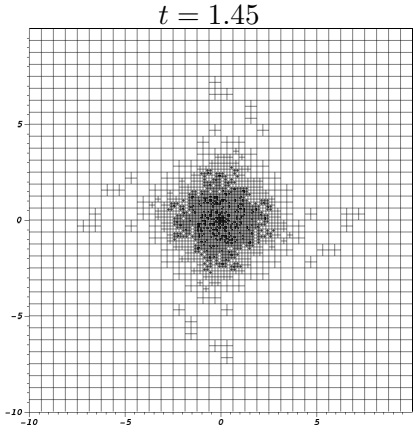

Figure 10: * $t = 3.00$

Figure 11: *Example 3:* Dynamically refined meshes, $\beta = 5$, $\Omega = 1$

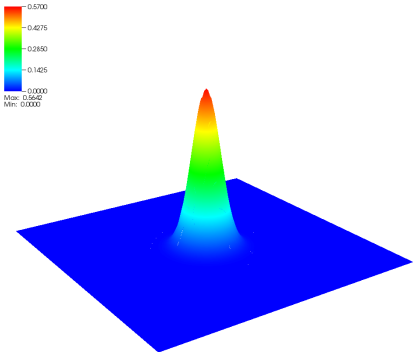

Figure 12: *
$t = 0.00$

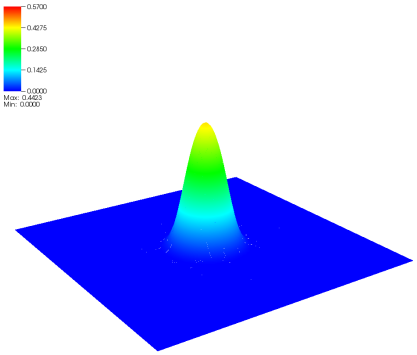

Figure 13: *
$t = 0.75$

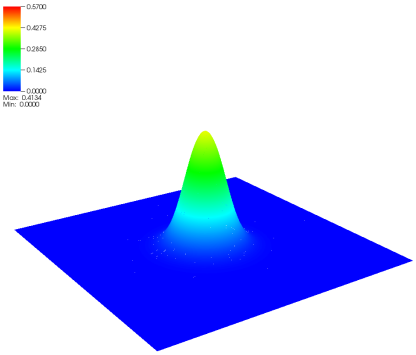

Figure 14: *
$t = 1.45$

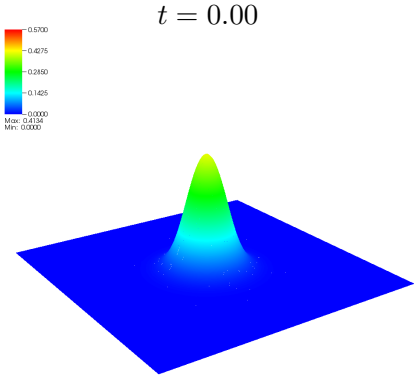

Figure 15: *
$t = 1.55$

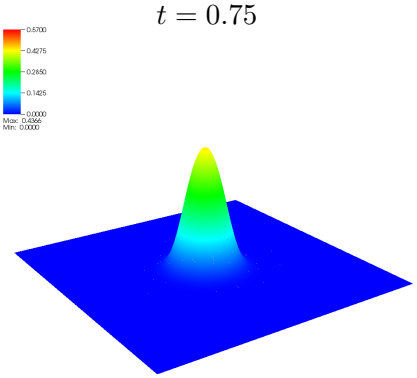

Figure 16: *
$t = 2.25$

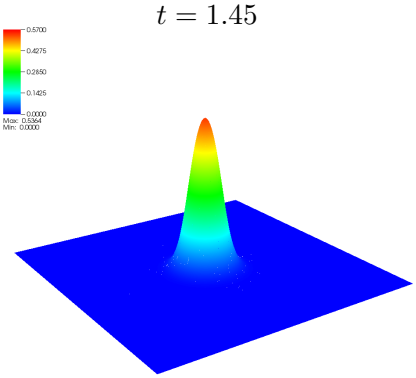

Figure 17: *
$t = 3.00$

Figure 18: *Example 3:* mass $|u|^2$, $\beta = 5$, $\Omega = 1$

0.575. The energy increases from 1.20472 up to 1.33816 and then decrease to 1.23923 for $t = 3.0$. Its error oscillates with the worst case with 2×10^{-03} decreasing to 10^{-03} and an average of 7×10^{-04}. For $\Delta t = 0.0125$, N_{max} has a similar behavior: two outliers at $t = 0.000$ and $t = 0.2875$, the remaining time steps about $17,978$, and a total element number of $N = 4,297,252$. Mass is about 1.00248 and its error oscillates about 5×10^{-04} with few cases over 10^{-03}. Energy behaves similarly to case $\Delta t = 0.025$, since it starts at 1.20529 and increases up to 1.35123 and then decreases to 1.22454 at $t = 3.00$. Errors also have an oscillating behavior about 7×10^{-04} with several cases with error 10^{-03}. Finally, for $\Delta = 0.00625$, N_{max} oscillates about $17,970$ with four outliers, $t = 0.000$, $t = 0.70625, 0.75625$ and 0.081875. The total number of elements rises to $N = 8,598,781$. Mass follows the same previous behavior with errors about 5×10^{-05}. For the energy, we also detect small oscillating values about the trend with errors about 10^{-03}.
In summary, we can see that energy and mass do not improve its value using the time and mesh refinement given in the previous section. Also, errors do not decrease for the increasing number of DoF, which is almost four times given by the time refinement. From Table 2, we can see that for values about $M = 120$ and $N = 2,143,240$ we get lower errors for mass and energy. It seems that mesh error η_h is leading the energy and mass error like in Table 2. We have not tested the relation between adaptation and errors, but we see that under higher mesh elements N, energy and mass error did not improve.

Comparing the case $M = 120$ in Table 5 with $M = 133$ in Table 4, we obtain similar mass an energy values, but with lower N. Now comparing the case $M = 240$ in Table 5 with $M = 262$ in Table 4, we also get similar values but now with larger N. We have no comparison for $M = 480$ in in Table 4.

4.4 A nonsymmetric potential

We change the trap potential setting two different weights

$$V(x) = \frac{1}{2}(\gamma_1^2 x_1^2 + \gamma_2^2 x_2^2), \qquad \gamma_1 = 2, \quad \gamma_2 = 1.$$

The trap has the shape of an ellipse and after eight rounds of adaptation the computed energy is $E(u_{kh}) = 1.031094$ with an error estimate $|\eta| < 4.2120 \cdot 10^{-03}$. This error is a slightly larger error than the previous example using about $300,000$ more DoF. See Table 6. Adding angular momentum $\Omega = 1$, it does not change the computed energy noteworthly, see Table 7. On the other hand, the adaptive algorithm starts much earlier to refine not only in time but also in space. After eight adaptation cycles, we use $14,468,680$ to reach $|\eta| < 1.1076 \cdot 10^{-02}$. In comparison with Figure 19, the adaptation of the time grid changes in Figure 20 following the rotation and the evolution of the wave intensity as we see from Figure 27 and Figure 34.

M	N	$N_{\max}$	$M(u_{kh})$	$E(u_{kh})$	$\lvert\eta_h\rvert + \lvert\eta_k\rvert$	$\lvert\eta_h\rvert$	$\lvert\eta_k\rvert$
30	261,950	8,450	1.001806	1.034039	$2.43 \cdot 10^{-01}$	$5.34 \cdot 10^{-03}$	$2.38 \cdot 10^{-01}$
46	397,150	8,450	1.001837	1.033928	$1.04 \cdot 10^{-01}$	$6.93 \cdot 10^{-03}$	$9.70 \cdot 10^{-02}$
70	599,950	8,450	1.000479	1.032047	$4.67 \cdot 10^{-02}$	$8.63 \cdot 10^{-03}$	$3.80 \cdot 10^{-02}$
106	904,150	8,450	1.000104	1.031511	$2.73 \cdot 10^{-02}$	$1.08 \cdot 10^{-02}$	$1.65 \cdot 10^{-02}$
133	1,578,844	12,834	0.999798	1.031270	$1.52 \cdot 10^{-02}$	$3.70 \cdot 10^{-03}$	$1.15 \cdot 10^{-02}$
167	2,830,464	19,858	0.998370	1.028915	$9.69 \cdot 10^{-03}$	$2.72 \cdot 10^{-03}$	$6.97 \cdot 10^{-03}$
209	5,056,020	31,650	0.999799	1.031001	$6.79 \cdot 10^{-03}$	$2.31 \cdot 10^{-03}$	$4.49 \cdot 10^{-03}$
262	9,390,422	50,674	0.999883	1.031094	$4.21 \cdot 10^{-03}$	$1.27 \cdot 10^{-03}$	$2.95 \cdot 10^{-03}$

Table 6: *Example 4:* Dynamic mesh refinement with equilibration, $\beta = 5$, $\Omega = 0$, nonsymmetric potential V

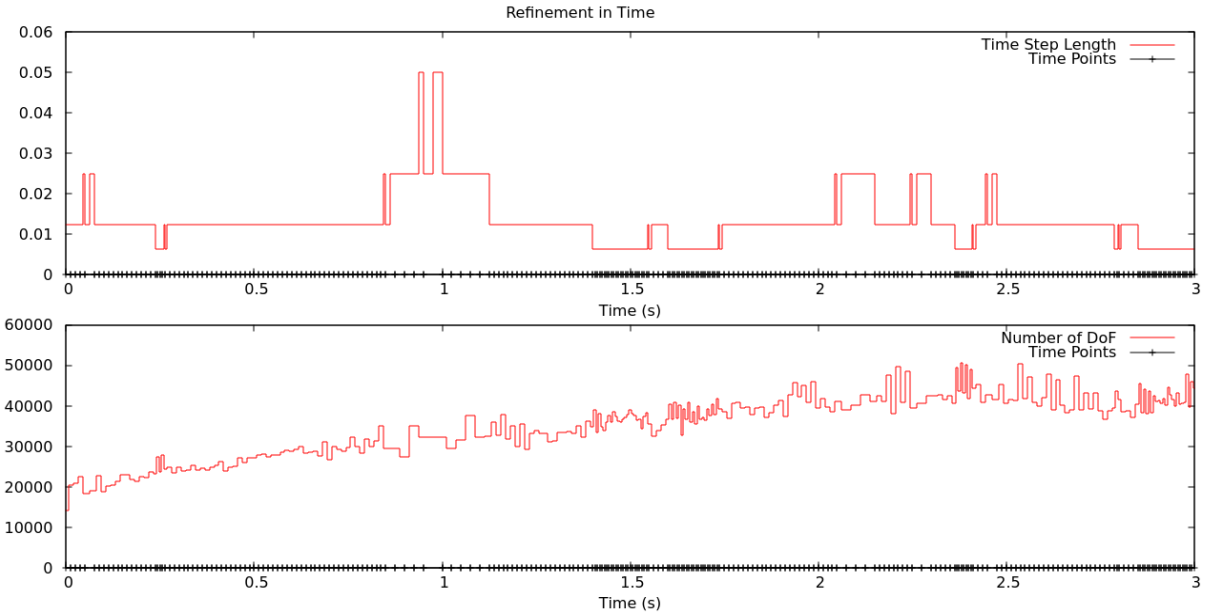

Figure 19: *Example 4:* Dynamic refinement in time

M	N	$N_{\max}$	$M(u_{kh})$	$E(u_{kh})$	$\lvert\eta_h\rvert + \lvert\eta_k\rvert$	$\lvert\eta_h\rvert$	$\lvert\eta_k\rvert$
30	261,950	8,450	1.004150	1.036606	$2.41 \cdot 10^{-01}$	$3.80 \cdot 10^{-02}$	$2.03 \cdot 10^{-01}$
46	397,150	8,450	1.002381	1.034183	$1.28 \cdot 10^{-01}$	$4.68 \cdot 10^{-02}$	$8.09 \cdot 10^{-02}$
58	698,390	12,674	0.999606	1.034444	$1.04 \cdot 10^{-01}$	$3.35 \cdot 10^{-02}$	$7.05 \cdot 10^{-02}$
73	1,268,180	23,266	0.998627	1.030291	$6.42 \cdot 10^{-02}$	$2.68 \cdot 10^{-02}$	$3.75 \cdot 10^{-02}$
92	2,304,970	44,098	1.000278	1.031683	$4.13 \cdot 10^{-02}$	$1.61 \cdot 10^{-02}$	$2.52 \cdot 10^{-02}$
116	4,253,482	73,858	1.000131	1.031511	$2.63 \cdot 10^{-02}$	$1.22 \cdot 10^{-02}$	$1.41 \cdot 10^{-02}$
146	7,934,486	119,954	1.000138	1.031506	$1.67 \cdot 10^{-02}$	$8.49 \cdot 10^{-03}$	$8.21 \cdot 10^{-03}$
183	14,468,680	184,898	1.000085	1.031420	$1.11 \cdot 10^{-02}$	$6.32 \cdot 10^{-03}$	$4.76 \cdot 10^{-03}$

Table 7: *Example 5:* Dynamic mesh refinement with equilibration, $cG(1)cG(2)$, $\beta = 5$, $\Omega = 1$

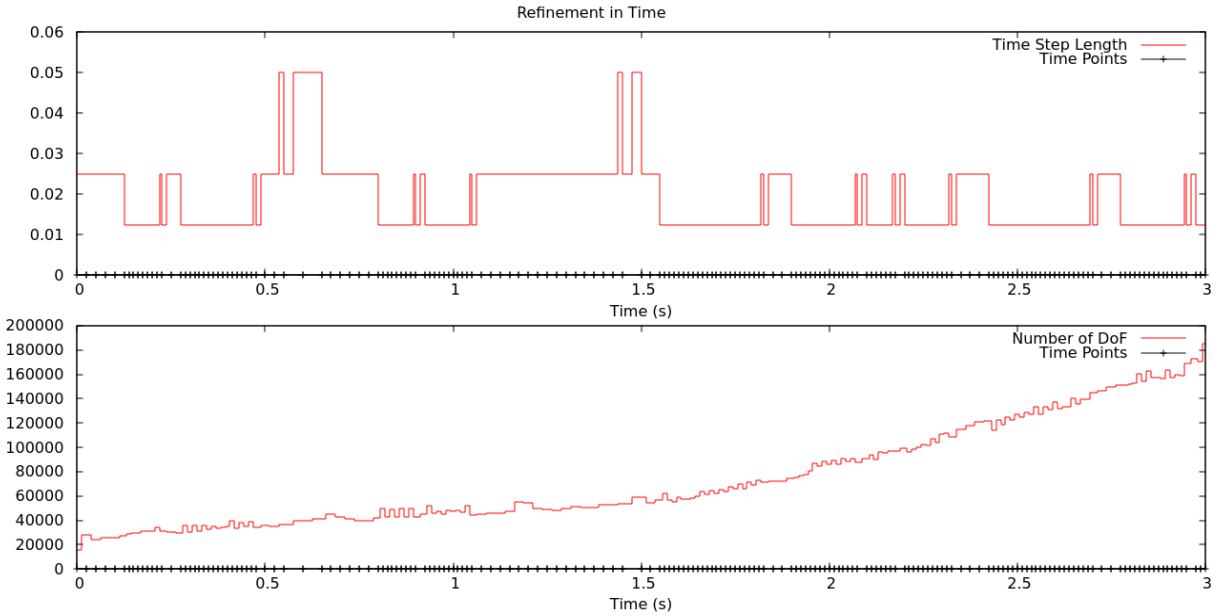

Figure 20: *Example 5:* Dynamic refinement in time

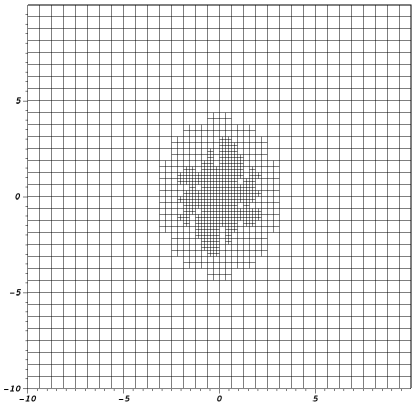

Figure 21: * $t = 0.00$

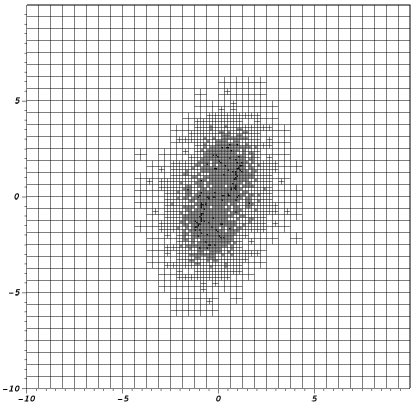

Figure 22: * $t = 1.00$

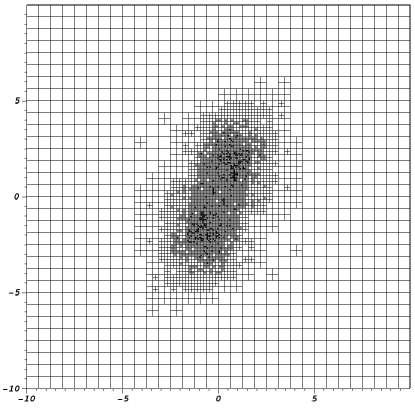

Figure 23: * $t = 1.50$

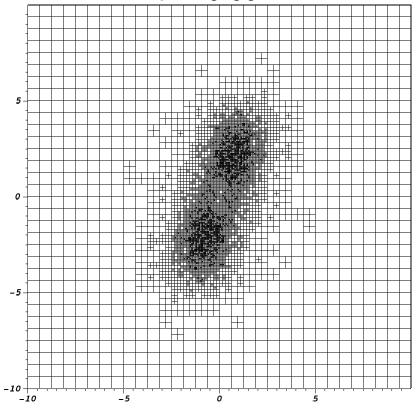

Figure 24: * $t = 2.00$

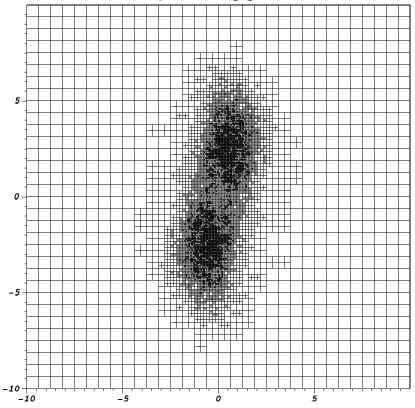

Figure 25: * $t = 2.50$

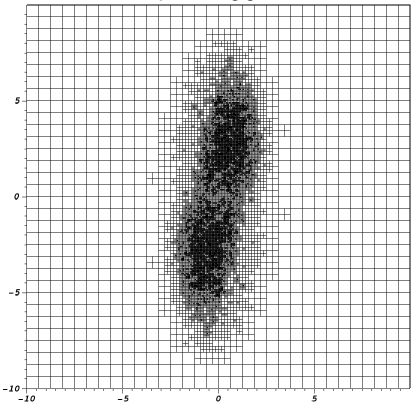

Figure 26: * $t = 3.00$

Figure 27: *Example 5:* Dynamically refined meshes, $\beta = 5$, $\Omega = 1$

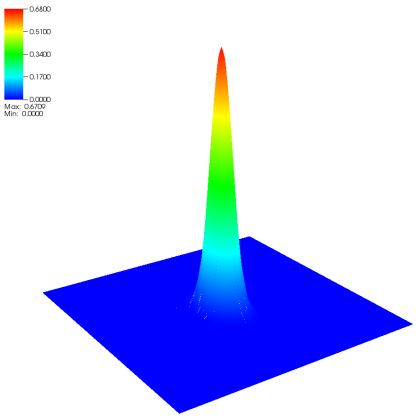

Figure 28: *
$t = 0.00$

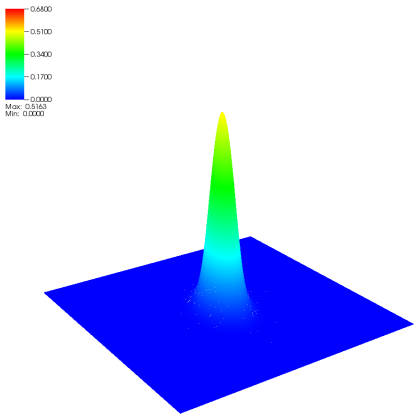

Figure 29: *
$t = 1.00$

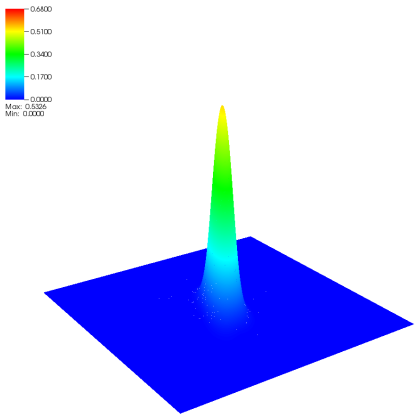

Figure 30: *
$t = 1.50$

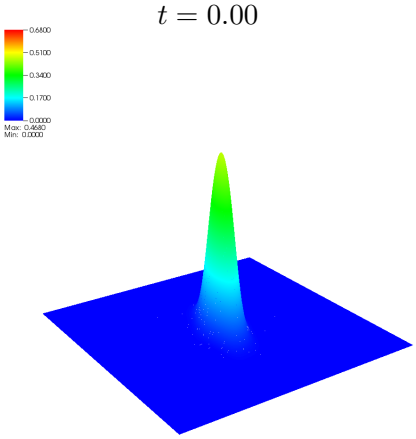

Figure 31: *
$t = 2.00$

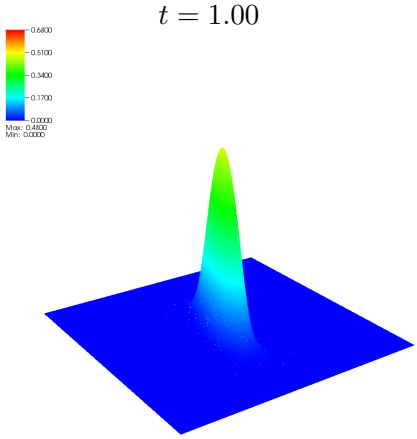

Figure 32: *
$t = 2.50$

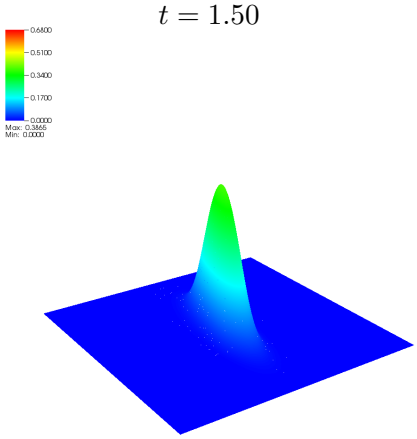

Figure 33: *
$t = 3.00$

Figure 34: *Example 5:* mass $|u|^2$, $\beta = 5$, $\Omega = 1$

5. Summary and Outlook

In this paper we have presented a refinement strategy for finite element schemes in time and space for nonlinear time-dependent Schrödinger equations. The refinement algorithm is based on the dual weighted residual method and an equilibration strategy for the temporal and spatial error estimators. We have shown numerical results built on the deal.II library for the Gross-Pitaevskii equation with an external harmonic trap potential V in a rotational frame and repulsive interaction.
There are different directions for future work. The algorithm presented here does not depend on the dimension of the underlying domain and also works in dimension three, but with higher computational effort. Another important aspect is the parameter ε, which is much smaller than one in many problems. Here the algorithm also works and some numerical tests show that the discretizations in time and space are adapted as expected, but the computational effort is also much higher because of possible concentrations of the solution. Moreover, more adaptation rounds are needed than in the presented examples.

Acknowledgments: The authors A. Meister and M. Steigemann gratefully acknowledge financial support by the Deutsche Forschungsgemeinschaft (DFG) through grant ME 1889/6-1.

References

[1] G.D. Akrivis, V.A. Dougalis, and O.A. Karakashian, *On fully discrete Galerkin methods of second-order temporal accuracy for the nonlinear Schrödinger equation*, Numer. Math. **59** (1991), no. 1, 31–54.

[2] M.H. Anderson, J.R. Ensher, M.R. Matthews, C.E. Wieman, and E.A. Cornell, *Observation of Bose-Einstein condensation in a dilute atomic vapor*, Science **269** (1995), no. 5221, 198–201.

[3] X. Antoine and R. Duboscq, *Robust and efficient preconditioned Krylov spectral solvers for computing the ground states of fast rotating and strongly interacting Bose-Einstein condensates*, J. Comput. Phys. **258** (2014), 509–523.

[4] A.I. Avila, A. Meister, and M. Steigemann, *On numerical methods for nonlinear singularly perturbed Schrödinger problems*, Appl. Num. Math. **86** (2014), 22–42.

[5] W. Bangerth, M. Geiger, and R. Rannacher, *Adaptive Galerkin finite element methods for the wave equation*, Comp. Meth. Appl. Math. **10** (2010), no. 1, 3–48.

[6] W. Bangerth, R. Hartmann, and G. Kanschat, *deal.II — a general-purpose object-oriented finite element library*, ACM Trans. Math. Softw. **33** (2007), no. 4, 24.

[7] W. Bangerth and R. Rannacher, *Adaptive finite element methods for differential equations*, Lectures in Mathematics, Birkhäuser Verlag, Basel, Boston, Berlin, 2003.

[8] W. Bao and Y. Cai, *Mathematical theory and numerical methods for Bose-Einstein condensation*, Kinet. Relat. Mod. **6** (2013), no. 1, 1–135.

[9] W. Bao, H. Wang, and P. A. Markowich, *Ground, symmetric and central vortex states in rotating Bose-Einstein condensates*, Comm. Math. Sci. **3** (2005), no. 1, 57–88.

[10] R. Becker and R. Rannacher, *A feed-back approach to error control in finite element methods: basic analysis and examples*, East-West J. Numer. Math. **4** (1996), 237–264.

[11] R. Becker and R. Rannacher, *An optimal control approach to a posteriori error estimation in finite element methods*, Acta Numerica **10** (2001), 1–102.

[12] M. Besier and R. Rannacher, *Goal-oriented space-time adaptivity in the finite element Galerkin method for the computation of nonstationary incompressible flow*, Int. J. Numer. Meth. Fluids **70** (2011), no. 9, 1139–1166.

[13] I. Bloch, J. Dalibard, and W. Zwerger, *Many-body physics with ultracold gases*, Rev. Mod. Phys. **80** (2008), no. 3, 885–944.

[14] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Introduction to adaptive methods for differential equations*, Acta Numerica **4** (1995), 105-158.

[15] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration*, Series in Computational Mathematics, vol. 31, Springer, Berlin, Heidelberg, 2002.

[16] P. Henning and A. Malqvist, *The finite element method for the instationary Gross-Pitaevskii equation with angular momentum rotation*, arXiv:1502.05025v2 (2015), 1–30.

[17] H. Hofstätter, O. Koch, and M. Thalhammer, *Convergence analysis of high-order time-splitting pseudo-spectral methods for rotational Gross-Pitaevskii equations*, Numer. Math. **127** (2014), 315–364.

[18] J. Ming, Q. Tang, and Y. Zhang, *An efficient spectral method for computing dynamics of rotating two-component Bose-Einstein condensates via coordinate transformation*, J. Comput. Phys. **258** (2014), no. 1, 538–554.

[19] J.W. Ruge and K. Stüben *Multigrid methods*, Frontiers in Applied Mathematics **3** (1987), 73-130.

[20] M. Schmich and B. Vexler, *Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations*, SIAM J. Sci. Comput. **30** (2008), no. 1, 369–393.

[21] K. Stüben, *A review of algebraic multigrid*, Journal of Computational and Applied Mathematics **128** (2001), no. 1, 281-309.

[22] M. Thalhammer and J. Abhau, *A numerical study of adaptive space and time discretisations for Gross-Pitaevskii equations*, J. Comput. Phys. **231** (2012), no. 20, 6665–6681.

[23] H.A. van der Vorst, *A fast and smoothly converging variant of Bi-CG fir the solution of nonsymmetric linear systems*, SIAM J. Sci. and Stat. Comput. **13** (1992), no. 2, 631–644.

[24] R. Webster, *An algebraic multigrid solver for Navier-Stokes problems in the discrete second-order approximation*, International journal for numerical methods in fluids **22** (1996) no. 11, 1103-1124.

[25] J. Wloka, *Partial differential equations*, Cambridge University Press, 1987.

[26] M.W. Zwierlein, J.R. Abo-Sheer, A. Schirotzek, C.H. Schunck, and W. Ketterle, *Vortices and superfluidity in a strongly interacting Fermi gas*, Nature **435** (2005), 1047–1051.