

A TWO LEVEL APPROACH FOR SIMULATING BOSE–EINSTEIN CONDENSATES BY LOCALIZED ORTHOGONAL DECOMPOSITION [★]

CHRISTIAN DÖDING¹, PATRICK HENNING^{2,**} AND JOHAN WÄRNEGÅRD³

Abstract. In this work, we consider the numerical computation of ground states and dynamics of single-component Bose–Einstein condensates (BECs). The corresponding models are spatially discretized with a multiscale finite element approach known as Localized Orthogonal Decomposition (LOD). Despite the outstanding approximation properties of such a discretization in the context of BECs, taking full advantage of it without creating severe computational bottlenecks can be tricky. In this paper, we therefore present two fully-discrete numerical approaches that are formulated in such a way that they take special account of the structure of the LOD spaces. One approach is devoted to the computation of ground states and another one for the computation of dynamics. A central focus of this paper is also the discussion of implementation aspects that are very important for the practical realization of the methods. In particular, we discuss the use of suitable data structures that keep the memory costs economical. The paper concludes with various numerical experiments in $1d$, $2d$ and $3d$ that investigate convergence rates and approximation properties of the methods and which demonstrate their performance and computational efficiency, also in comparison to spectral and standard finite element approaches.

Mathematics Subject Classification. 35Q55, 65M60, 65M15, 81Q05.

Received April 26, 2023. Accepted May 21, 2024.

1. INTRODUCTION

In this paper we consider the numerical approximation of the stationary and the time-dependent Gross–Pitaevskii equation by advanced finite element techniques. The Gross–Pitaevskii equation has applications in quantum physics where it is used to model so-called Bose–Einstein condensates (BECs) which form when a dilute bosonic gas of particles is cooled down to near the absolute zero temperature [72]. In this extreme state of matter all particles within the condensate are in the same quantum state and are indistinguishable from each other so that the BEC itself behaves like a single quantum particle. This state, sometimes called a “super-atom”, gives access to quantum mechanical phenomena on an observable scale such as superfluidity and justifies why BECs became a highly active research area in modern quantum physics; see *e.g.*, [30, 40, 59, 69, 73] and the

Keywords and phrases. Bose–Einstein condensate, Gross–Pitaevskii equation, nonlinear Schrödinger equation, ground state, dynamics, finite element method.

[★] *Dedicated to the memory of Assyr Abdulle.*

¹ Institute for Numerical Simulation, University Bonn, DE-53115 Bonn, Germany.

² Department of Mathematics, Ruhr University Bochum, DE-44801 Bochum, Germany.

³ Department of Applied Physics and Applied Mathematics, Columbia University, New York, NY 10027, USA.

^{**}Corresponding author: patrick.henning@rub.de

references therein. An underlying mathematical model that describes the formation of BECs was first derived by Gross [41] and Pitaevskii [71] and later mathematically justified by Lieb *et al.* [61, 62].

We are interested in numerical simulations of real-world models in the context of BECs. For that reason it is required to consider two problems of different mathematical flavor. The first one is the time-dependent Gross–Pitaevskii equation (GPE),

$$i\partial_t u = -\frac{1}{2}\Delta u + Vu + \beta|u|^2 u, \quad (1)$$

which describes the time evolution of the quantum mechanical wave function u of the BEC. The function $V = V(x)$ models a magnetic or optical potential that traps the condensate in the physical experiment, and the constant β is a repulsion parameter so that the nonlinearity $\beta|u|^2 u$ models the interaction between the particles within the condensate. The second problem we have to tackle is the nonlinear Gross–Pitaevskii eigenvalue problem (GPEVP)

$$\lambda u = -\frac{1}{2}\Delta u + Vu + \beta|u|^2 u, \quad (2)$$

which characterizes energy extrema of the wave function. The unique minimal solution u , minimal in terms of λ , is called the ground state whereas all other solutions are called excited states. Besides their strong mathematical connection, there is also high interest in studying both problems in view of the application. The reason for this is the standard procedure followed in experiments with BECs; cf. [22, 40]. In a first step, a BEC is created by cooling down the gas of bosons to near absolute zero temperature. During this process, the condensate is held in place by a magnetic or optical trapping potential. As a result of this first stage, the condensate is prepared in the ground state w.r.t. the current trapping potential. In a second stage, the potential is tweaked or removed and the subsequent dynamics of the condensate are studied. In this stage the wave function of the condensate is described by the solution of the time-dependent GPE w.r.t. the tweaked trapping potential.

The main goal of this work is to present an advanced computational approach for simulating both stages of the aforementioned physical process and therefore for tackling both mathematical problems. The approach is based on a generalized finite element approximation in space known as the *Localized Orthogonal Decomposition* (LOD) which is a multiscale technique that was first introduced by Målqvist and Peterseim [46] in the context of highly heterogeneous diffusion problems and was further successfully applied to the GPEVP [48, 55] and the GPE [54]. In this work we further extend the in [48, 54, 55] achieved results and design two novel methods that combine the LOD with suitable iterative solvers depending on the nature of the considered problem, *i.e.*, the GPEVP or the time-dependent GPE. The designed fully-discrete methods come with several advantages in view of its approximation properties due to the LOD discretization. However, realizing them in an efficient way is a nontrivial task since the methods require the assembly of nonlinear terms in each iteration which may harm the theoretical advantages of the methodology in view of the computational costs. Therefore, we introduce a modification of the desired methods by adding an additional projection on the LOD space into the method. As we shall then demonstrate in numerical experiments this modification significantly decreases the computational cost and does not impose further substantial errors so that we achieve a very good overall performance of these novel methods.

To solve the nonlinear eigenvalue problem GPEVP, which is equivalent to the minimization of the associated energy functional, an adaptive Riemannian optimization method is used as an iterative solver and that is based on changing the metric in each iteration. The numerical algorithm of the iteration was originally proposed and analyzed in [51] and enjoys global convergence properties, which makes it very attractive in practice to obtain reliable approximations. For the time-dependent GPE, a continuous Galerkin (cG) ansatz is used for the time discretization, first introduced in combination with classical Lagrange finite elements in space by Karakashian and Makridakis [57] for the cubic nonlinear Schrödinger equation and further analyzed in [36]. The resulting time-stepping method is energy-conserving and it is of convergence order $\mathcal{O}(\tau^{2q})$ with respect to the time-step size τ and where q denotes the polynomial degree of the time ansatz functions (cf. [36, 57]). We observe numerically the same order of convergence when we combine the time-integrator with the more problem-specific LOD space.

The algebraic structure of the equation (1) comes with several time invariants that convey information about the physical state of the system on a global scale. With this in mind, it comes as no surprise that in order to achieve reliable numerical methods it is of utmost importance to conserve and accurately represent these quantities in the discrete setting. We recall that the accuracy with which these quantities are reproduced in the discrete setting is subject to a suitable spatial discretization whereas their conservation is subject to a suitable time discretization. As for the time discretization, the choice of conservative time-integrator schemes over non-conservative ones can have a tremendous advantage in terms of accuracy. This observation has been confirmed in various numerical experiments (cf. [52, 75]). Among others, mass conservative time discretizations have been studied in [81, 83], time integrators that are mass conservative and symplectic are investigated in [12, 47, 78, 80], energy conservative time discretizations in [57] and time discretizations that preserve mass and energy simultaneously are addressed in [12, 21, 25, 26, 31, 50, 53, 74].

As for the spatial discretization, a popular choice is to use a spectral approach which, for time integrations, often relies on splitting methods. These have been analyzed in for example [22, 64, 77]. Other common choices include finite differences, finite element methods or recently holistic discretizations [27]. Recent cross comparisons between different approaches have been made in [19, 22, 24, 52]. In particular, in high regularity regimes with smooth potentials and regular domains, the exponential convergence of a spectral approach leads to methods that are computationally very efficient. On the downside, they typically perform poorly in low regularity regimes such as in the case of a low regularity potential, which shall also be exemplarily illustrated in the present paper. Here, the method of LOD presents a hitherto unique opportunity to bridge this gap as it achieves high spatial accuracy even for discontinuous potentials, as well being computationally highly competitive for smooth solutions.

In many physically relevant settings it can also be reasonable to include the error in the initial discrete energy as an additional measure for the accuracy of a method. An example for such a setting was illustrated in [54], where excremental discrete energy in the system was converted into artificial kinetic energy that lead to spurious solutions. One way of achieving high accuracy with respect to the initial energy at low computational cost was addressed in [54] by using the aforementioned LOD. The LOD allows to build a discrete approximation space that is adapted to the underlying differential equation by incorporating problem-specific information (explained in Sect. 3). With this technique it is possible to approximate the initial value to 6th order accuracy with respect to the mesh size in terms of the time invariants of the equation. This superconvergence property only requires moderate regularity of the initial value u_0 , which is compatible with physically relevant problems where u_0 minimizes a certain energy. This observation makes the LOD extremely attractive in the context of the Gross–Pitaevskii equation and its conserved quantities, not only for the time-dependent equation as in [54], but also for the Gross–Pitaevskii eigenvalue problem [48, 55]. For further reading on the LOD we refer to the textbook by Målqvist and Peterseim [67] and the survey article [15]. Basic implementation aspects are discussed in [38] and for more recent developments on the LOD we exemplarily refer to [42, 44, 45, 60, 63, 65, 68, 82].

We finish the introduction with a brief outline of the paper. In Section 2 we set up the mathematical framework we are working in and formulate the aforementioned two problems in the context of BECs. Section 3 is then devoted to the spatial discretization by Localized Orthogonal Decomposition. There we start with the basic construction of the LOD space, apply it to the GPE and then discuss the approximation properties regarding ground states and time invariants. The section is closed with a description of the practical implementation of the LOD discretization. In Section 4 we then derive the fully-discrete method that we propose for solving the GPEVP using a damped inverse iteration based on LOD discretization. In Section 5 we state the fully-discrete method for solving the time-dependent GPE using the energy conserving time integrator. We finish the paper in Section 6 with numerical results presenting the performance and efficiency of our methods in several experiments. Further notes on the efficient implementation of the numerical methods is given in the Appendix A.

2. MATHEMATICAL SETTING OF THE GROSS–PITAEVSKII EQUATION

In this section we describe the mathematical framework of the Gross–Pitaevskii equation and formulate the two problems for the formation of ground states and for the dynamic evolution of BECs. Throughout this paper, we shall denote by $\mathcal{D} \subset \mathbb{R}^d$ (for $d = 1, 2, 3$) a bounded convex domain with polyhedral boundary. The complex Hilbert space $L^2(\mathcal{D}) = L^2(\mathcal{D}, \mathbb{C})$ of complex-valued square-integrable functions is equipped with the scalar product $(v, w) = \int_{\mathcal{D}} v(x) \overline{w(x)} dx$ where $\overline{w}$ is the complex-conjugate of w and we denote by $\|\cdot\|$ the induced norm for brevity. As usual, $H_0^1(\mathcal{D}) = H_0^1(\mathcal{D}, \mathbb{C})$ denotes the Sobolev space of complex-valued, weakly differentiable functions with zero trace on $\partial\mathcal{D}$ and L^2 -integrable partial derivatives. In the derivation of the problems (1) and (2) for modeling BECs, the energy of the system is a quantity that plays an important role. It is given by the Gross–Pitaevskii energy functional

$$E(u) = \int_{\mathcal{D}} \frac{1}{2} |\nabla u|^2 + V|u|^2 + \frac{\beta}{2} |u|^4 dx \quad (3)$$

for $u \in H_0^1(\mathcal{D})$ where we assume that the trapping potential is given by a bounded, real-valued, and non-negative function, *i.e.*, $V \in L^\infty(\mathcal{D}, \mathbb{R}_{\geq 0})$ and that the repulsion parameter $\beta \in \mathbb{R}$ is non-negative. The Gross–Pitaevskii energy functional (3) is derived from the many body problem of a cold and weakly interacting bosonic gas by mean-field approximation and a Hartree-ansatz for the wave function of the condensate. We refer to [22, 39, 41, 62, 71, 72] for precise details on the physical derivation and validation of the Gross–Pitaevskii energy functional.

2.1. The Gross–Pitaevskii eigenvalue problem and ground states

We now derive the Gross–Pitaevskii eigenvalue problem (2) from the energy functional (3) and specify the notion of ground states of BECs. In particular, we are interested in finding the state of lowest energy of a BEC (in the given trapping potential V), *i.e.*, the state that minimizes the Gross–Pitaevskii energy (3) under the normalization constraint $\|u\| = 1$. Hence, we seek for $u_{\text{gs}} \in H_0^1(\mathcal{D})$ such that

$$u_{\text{gs}} = \arg \min_{u \in \mathbb{S}} E(u) \quad \text{where } \mathbb{S} := \{u \in H_0^1(\mathcal{D}) \text{ and } \|u\| = 1\}. \quad (4)$$

We call u_{gs} the *ground state* of the BEC (in the given configuration) and it can be shown that the problem is well-posed and u_{gs} is unique up to sign [28]. It is possible to characterize u_{gs} equivalently through a nonlinear eigenvalue problem. This perspective can be both beneficial for the analysis, as well as for numerical computations. To reformulate (4) as an eigenvalue problem, we simply consider the corresponding Euler–Lagrange equations of the form $E'(u_{\text{gs}}) = \lambda_{\text{gs}} S'(u_{\text{gs}})$, where $S(u) := \|u\|^2 - 1$ represents the constraint functional (as we demand $S(u_{\text{gs}}) = 0$) and λ_{gs} is the Lagrange multiplier for the constraint. Computing the Fréchet derivatives of the two functionals we recover the GPEVP in variational form as

$$\lambda_{\text{gs}}(u_{\text{gs}}, v) = \frac{1}{2} (\nabla u_{\text{gs}}, \nabla v) + (V u_{\text{gs}}, v) + \beta (|u_{\text{gs}}|^2 u_{\text{gs}}, v) \quad \text{for all } v \in H_0^1(\mathcal{D}),$$

with the constraint $u_{\text{gs}} \in \mathbb{S}$. In this setting, the eigenfunction u_{gs} associated with the smallest eigenvalue λ_{gs} is indeed the ground state in the sense of (4). This nontrivial observation was first made in [28]. Consequently, we can write the GPEVP compactly as: find the smallest eigenvalue $\lambda_{\text{gs}} > 0$ and corresponding eigenfunction $u_{\text{gs}} \in \mathbb{S}$ such that in the sense of distributions on $H_0^1(\mathcal{D})$:

$$\lambda_{\text{gs}} u_{\text{gs}} = -\frac{1}{2} \Delta u_{\text{gs}} + V u_{\text{gs}} + \beta |u_{\text{gs}}|^2 u_{\text{gs}}. \quad (5)$$

We note that the ground state u_{gs} is (up to sign) identical to the solution of (4). Natural regularity statements about u_{gs} that are important for optimal convergence rates of our numerical approximations are given in [48].

Remark 2.1. There are various iterative approaches for finding minimizers of (4), such as methods based on Riemannian optimization [16, 18, 20, 23, 32, 33, 43], based on self consistent field iterations (SCF) [35, 79], or the “ J -method” [14, 56]. In this paper, we apply the damped inverse iterations proposed in [51] due to its fast and global convergence properties when the damping parameter is chosen adaptively. Details of the method are given later in Section 4.

2.2. The time-dependent Gross–Pitaevskii equation and its invariants

Next, let us turn to the time-dependent GPE describing the dynamics of BECs on a relevant time interval $[0, T]$. We assume that the BEC is at time $t = 0$ in a state which is described by a wave function $u_0 \in H_0^1(\mathcal{D})$. The time evolution of the BEC is then determined by the Schrödinger equation $i\partial_t u = \frac{1}{2}E'(u)$ with the Gross–Pitaevskii energy functional E from (3). Taking the Fréchet derivative of the energy functional into account leads to the time-dependent GPE formulated in variational form as

$$(i\partial_t u, v) = \frac{1}{2}(\nabla u, \nabla v) + (Vu, v) + \beta(|u|^2 u, v) \quad \text{for all } v \in H_0^1(\mathcal{D}). \quad (6)$$

Hence we can formulate the time-dependent Gross–Pitaevskii problem as follows: find

$$u \in L^\infty([0, T], H_0^1(\mathcal{D})) \quad \text{and} \quad \partial_t u \in L^\infty([0, T], H^{-1}(\mathcal{D}))$$

such that $u(0) = u_0 \in H_0^1(\mathcal{D})$ and such that for almost every $t \in (0, T]$

$$i\partial_t u = -\frac{1}{2}\Delta u + Vu + \beta|u|^2 u. \quad (7)$$

Note that such a solution automatically belongs to $C([0, T], L^2(\mathcal{D}))$ and therefore the notion of the initial value is well-defined. Moreover, the initial value $u_0 \in H_0^1(\mathcal{D})$ can be chosen arbitrarily in the general setting. However, as motivated by our physical application, we shall mainly consider the physically reasonable case that $u_0 = u_{\text{gs}}$ is a ground state in a modified configuration in the sense of problem (5), *i.e.*, u_{gs} from (5) is the ground state trapped in a different potential than used for the time-dependent problem (7). This being said, it is worth to mention that the maximal existence time of solutions to the GPE (7) depends crucially on the initial value (in particular its regularity and the size of u_0 in the H^1 -norm), so that unphysical initial values easily lead to a chaotic behavior after short times that ultimately leads to the nonexistence of solutions. All aspects of existence and uniqueness of solutions to the GPE (7) are nicely elaborated in the textbook by Cazenave [29]. In particular, the GPE is at least locally well-posed, *i.e.*, for T sufficiently small. When talking about solutions and discussing convergence results for numerical methods, we shall always assume that problem (7) admits a *unique solution* that is *sufficiently regular*.

Since the problem (7) is derived as a Hamiltonian system from the Gross–Pitaevskii energy functional (3) its solution u is automatically energy-preserving over time, that is

$$E(u(t)) = E(u_0) \quad \text{for all } t \in [0, T].$$

For example, this can be derived directly from the variational form (6) when it is tested with $v = \partial_t u$ and imaginary parts are taken. In contrast to that, when (6) is tested with $v = u$, one infers the second fundamental conservation law

$$\|u(t)\|^2 = \|u_0\|^2 \quad \text{for all } t \in [0, T]$$

which is the preservation of mass. Other conservation laws hold true in less general settings such as the conservation of momentum and conservation of the mass center in the absence of the trapping potential. For details and more possible conservation laws we refer exemplarily to [22, 54]. From the numerical point of view it can be extremely important to choose numerical methods that approximate these quantities with high accuracy in order to capture the correct behavior and dynamics of BECs in numerical experiments; see *e.g.*, [47, 52, 75].

3. SPATIAL DISCRETIZATION BY LOCALIZED ORTHOGONAL DECOMPOSITION

As sketched in the introduction, our spatial discretization is based on the Localized Orthogonal Decomposition. In the following we will briefly describe the construction of LOD spaces and why it can be beneficial to use them in various applications. After that we review the approximation properties of these spaces in the context of the GPEVP and the time-dependent GPE and we describe one of the central computational issues in this context, namely the efficient assembly of nonlinear terms.

Before diving into the explicit construction of the spaces, we shall introduce some preliminaries that hold throughout the whole manuscript. The first essential ingredient for constructing a LOD space is a “coarse” finite element mesh $\mathcal{T}_H$ that corresponds to a shape-regular and quasi-uniform partition of the computational domain $\mathcal{D}$ with mesh size $H > 0$. In the following, we shall restrict the type of considered partitions to being simplicial (*i.e.*, intervals in 1D, triangles in 2D and tetrahedra in 3D), though the construction also directly generalizes to quadrilateral partitions. By a “coarse” mesh we mean that the number of elements in $\mathcal{T}_H$ is small compared to what would be usually necessary to obtain reasonable approximations in corresponding conventional finite element spaces. The aspect of coarseness will become clearer when describing the construction in more detail below.

Next, we consider a standard P1 finite element space on the mesh $\mathcal{T}_H$ given by

$$V_H := \{v \in C(\overline{\mathcal{D}}) \cap H_0^1(\mathcal{D}) \mid v|_K \in \mathbb{P}^1(K) \quad \text{for all } K \in \mathcal{T}_H\}.$$

We shall later modify this space (while keeping its dimension fixed), by correcting it based on problem-specific information entering through an elliptic differential operator. A crucial component of the construction is the choice of a set of quantities of interest which are represented by pairwise linear independent functionals $q_j : H_0^1(\mathcal{D}) \rightarrow \mathbb{R}$. Given such a set, the basic task is to construct “cheap” numerical approximations u_H^{LOD} that essentially conserve all the quantities of interest for the exact solution u , *i.e.*, with the property $q_j(u_H^{\text{LOD}}) = q_j(u)$ for all q_j . For linear problems, exact conservation is often possible, whereas for nonlinear problems (like the GPE), we have to content with almost conservation up to high accuracy. In the context of the GPE, quantities of interest that were identified as suitable are *local weighted averages* based on the coarse space V_H , cf. [48, 54, 55]. More precisely, we select

$$q_j(v) := (\varphi_j, v)_{L^2(\mathcal{D}_j)},$$

where $\varphi_j \in V_H$ denotes the (conventional) nodal basis function associated with the (interior) mesh node z_j and which is supported on a patch $\mathcal{D}_j \subset \mathcal{D}$. Before stating how this is exploited in the construction of a space, we require one more notation, which is the so-called detail space $W \subset H_0^1(\mathcal{D})$ that contains functions that do not change the quantities of interest when added to H^1 -functions and in particular to elements of V_H . It is given by

$$W := \{w \in H_0^1(\mathcal{D}) \mid q_j(w) = 0 \quad \text{for all } j\}.$$

Note that W is just the kernel of the L^2 -projection onto the finite element space V_H as its definition implies $(v_H, w) = 0$ for all $v_H \in V_H$ and all $w \in W$.

3.1. The ideal Localized Orthogonal Decomposition

To explain how to correct the finite element space V_H based on an elliptic differential operator and the quantities of interest encoded through the space W , we shall study an idealized setting. For that let $\mathcal{L}$ be a symmetric elliptic differential operator represented by a (symmetric) bilinear form $a(\cdot, \cdot)$ on $H_0^1(\mathcal{D})$ and let $f \in L^2(\mathcal{D})$ be a given source term. With this, consider the equation seeking $u \in H_0^1(\mathcal{D})$ with

$$a(u, v) = (f, v) \quad \text{for all } v \in H_0^1(\mathcal{D}). \quad (8)$$

A suitable discrete solution space is now constructed based on the $a(\cdot, \cdot)$ -orthogonal projection on W denoted by $\mathcal{Q} : H_0^1(\mathcal{D}) \rightarrow W$, *i.e.*, for $v \in H_0^1(\mathcal{D})$, the image $\mathcal{Q}v \in W$ is given as the solution to

$$a(\mathcal{Q}v, w) = a(v, w) \quad \text{for all } w \in W. \quad (9)$$

The operator $\mathcal{Q}$ is also called the *correction operator*, because we use it to “correct” V_H in the sense that the ideal approximation space is defined by

$$V_H^{\text{LOD}} := \{v_H - \mathcal{Q}v_H \mid v_H \in V_H\}, \quad (10)$$

or compactly $V_H^{\text{LOD}} = (I - \mathcal{Q})V_H$. Obviously, V_H^{LOD} has the same dimension as the coarse space V_H . Furthermore, Galerkin approximations in V_H^{LOD} conserve the quantities of interest. To see this, let $u^{\text{LOD}} \in V_H^{\text{LOD}}$ denote the solution to

$$a(u^{\text{LOD}}, v) = (f, v) \quad \text{for all } v \in V_H^{\text{LOD}}. \quad (11)$$

Testing in (8) with $v \in V_H^{\text{LOD}}$ and subtracting it from (11) yields the Galerkin orthogonality

$$a(u^{\text{LOD}} - u, v) = 0 \quad \text{for all } v \in V_H^{\text{LOD}}. \quad (12)$$

As $V_H^{\text{LOD}} = (I - \mathcal{Q})V_H$ is the $a(\cdot, \cdot)$ -orthogonal complement of W (which directly follows from (9)), we conclude from (12) that $u^{\text{LOD}} - u \in W$ and consequently

$$q_j(u^{\text{LOD}} - u) = 0, \quad \text{or equivalently} \quad q_j(u^{\text{LOD}}) = q_j(u),$$

for all quantities of interest q_j . By the choice of q_j as $q_j(v) = (\varphi_j, v)$, we also obtain error estimates that only depend on the regularity of the source term f . To be precise, let $s = 0, 1, 2$, where $s = 0$ if $f \in L^2(\mathcal{D})$ and $s = 1, 2$ if $f \in H_0^1(\mathcal{D}) \cap H^s(\mathcal{D})$, then it holds

$$\|u^{\text{LOD}} - u\| + H \|u^{\text{LOD}} - u\|_{H^1(\mathcal{D})} \leq C H^{s+2} \|f\|_{H^s(\mathcal{D})},$$

where the constant C does not depend on u or its regularity. For a detailed derivation of the estimate we refer to [54]. The essential observation to make here is that for elliptic PDEs it is possible to construct a problem-specific low-dimensional space V_H^{LOD} that admits super-approximation properties that do not depend on the regularity of the solution itself or possible structural variations of u . In particular, the underlying mesh $\mathcal{T}_H$ is generic and only needs to compensate $\|f\|_{H^s(\mathcal{D})}$, but it does not need to resolve the variations of u . The latter aspect is indirectly taken care of through the space V_H^{LOD} .

3.2. Application to the Gross–Pitaevskii equation

In order to approximate the solutions of the Gross–Pitaevskii problems (5) and (7) using the LOD discretization in space (as described in the previous subsection) we have to specify the bilinear form $a(\cdot, \cdot)$ on $H_0^1(\mathcal{D})$. For this purpose, we neglect the nonlinear term in the GPE and may choose the bilinear form $a(\cdot, \cdot)$ in (8) as

$$a_V(u, v) := \frac{1}{2}(\nabla u, \nabla v) + (Vu, v) \quad (13)$$

which is induced by the main part of the GPE and, in particular, incorporates the trapping potential V . Alternatively, it is also possible to include only parts of the potential V or to fully drop it which results in the choice

$$a_c(u, v) := (\nabla u, \nabla v) \quad (14)$$

to which we refer as the canonical bilinear form. There are a few practical differences regarding the performance of the LOD based on the choices (13) and (14). First of all, if V is a rough potential with low regularity, it is typically reasonable to include it in the construction of V_H^{LOD} to obtain optimal convergence rates, cf. [48]. If V is smooth, the guaranteed convergence rates are the same whether V is included or not, even though the absolute errors might differ quite significantly. The major practical difference between (13) and (14) lies in the computation of the corresponding LOD basis functions. If V is periodic or not included at all, then the LOD

basis functions $(I - \mathcal{Q})\varphi_j$ have to be computed only for a few nodes z_j , whereas the remaining basis functions are essentially obtained by translations. On the contrary, if V is heterogenous, then it is necessary to compute each LOD basis separately by solving an elliptic problem (cf. (9)), which causes considerable costs in the initial phase (“offline phase”). Last but not least, if V is included, then $(I - \mathcal{Q})\varphi_j$ has often a more complex structure. This calls for more accurate quadrature rules when computing products of (gradients of) LOD basis functions in mass and stiffness matrices.

Many of these aspects will become clearer after describing the method in full detail together with its implementation. We shall also revisit these issues in the numerical experiments in Section 6 to highlight them again and to support them by CPU times.

In the following we shall briefly review the relevant approximation properties of LOD spaces in the context of the Gross–Pitaevskii equation.

3.2.1. Ideal LOD-approximation of ground states

We consider the GPEVP for the ground state eigenfunction u_{gs} and we recall that this is equivalent to the energy minimization problem (4). In order to approximate u_{gs} , let V_H^{LOD} be the LOD space constructed as described in Section 3.1 based on a coarse P1 finite element space V_H and on the bilinear form $a(\cdot, \cdot)$ which is either chosen as in (13) or as in (14). A numerical approximation of u_{gs} is then given by a discrete minimizer in the LOD space, *i.e.*,

$$u_{\text{gs}}^{\text{LOD}} = \arg \min_{v \in \mathbb{S} \cap V_H^{\text{LOD}}} E(v).$$

As $u_{\text{gs}}^{\text{LOD}}$ and u_{gs} can be only unique up to sign, we can select them such that they are consistent in the sense that

$$(u_{\text{gs}}, u_{\text{gs}}^{\text{LOD}}) \geq 0.$$

Since we assumed $V \in L^\infty(\mathcal{D})$ to be non-negative and $\beta \geq 0$, we have the following *a priori* error estimates for the LOD approximations that were proved in [48]:

$$\|u_{\text{gs}} - u_{\text{gs}}^{\text{LOD}}\| + H \|u_{\text{gs}} - u_{\text{gs}}^{\text{LOD}}\|_{H^1(\mathcal{D})} \leq CH^4.$$

Here the constant $C > 0$ depends on the potential V , the ground state eigenvalue λ_{gs} , the constant β , the domain $\mathcal{D}$, the maximum value of the ground state $\|u_{\text{gs}}\|_{L^\infty(\mathcal{D})}$ and the mesh regularity of $\mathcal{T}_H$. For the corresponding ground state energy one has the estimate

$$|E(u_{\text{gs}}) - E(u_{\text{gs}}^{\text{LOD}})| \leq CH^6$$

and for the ground state eigenvalue

$$|\lambda_{\text{gs}} - \lambda_{\text{gs}}^{\text{LOD}}| \leq CH^6 \quad \text{where } \lambda_{\text{gs}}^{\text{LOD}} := \int_{\mathcal{D}} \frac{1}{2} |\nabla u_{\text{gs}}^{\text{LOD}}|^2 + V |u_{\text{gs}}^{\text{LOD}}|^2 + \beta |u_{\text{gs}}^{\text{LOD}}|^4 \, dx.$$

The above result shows that in the LOD space, we obtain superconvergence in various errors. For comparison, in the standard P1 finite element space V_H (which has the same dimension as V_H^{LOD}), the H^1 -error only converges with the order $\mathcal{O}(H)$ and the L^2 -error with the order $\mathcal{O}(H^2)$. This is the same for the errors in the ground state energy and in the ground state eigenvalue which also both converge with the order $\mathcal{O}(H^2)$ (compared to $\mathcal{O}(H^6)$ in V_H^{LOD}). The superior approximation properties of the LOD space make it very attractive for solving the Gross–Pitaevskii eigenvalue problem. If $u_{\text{gs}}^{\text{LOD}}$ is selected as an initial value in an energy-conserving discretization of the time-dependent equation (7), then we can guarantee that the energy of the discrete approximations accurately resembles the exact energy on arbitrarily large time scales.

3.2.2. Ideal LOD-approximation of time invariants

Let us now turn towards the question how accurate time invariants of the GPE, such as the energy and mass, can be approximated in the LOD space. This question particularly arises when we assume that the initial value u_0 in (7) is given by an analytical function in $H_0^1(\mathcal{D})$. Hence, since we aim to solve (7) numerically using the LOD discretization in space we first need to project the initial value onto the LOD space V_H^{LOD} . But the solutions of the GPE may be very sensitive to perturbations in energy (or mass) and therefore this error w.r.t. energy in the initial value caused by the spatial discretization is of high relevance.

In general, let us assume that a sufficiently smooth function $u \in H_0^1(\mathcal{D})$ is given and that the space V_H^{LOD} is constructed based on the bilinear form $a(\cdot, \cdot)$, selected either as in (13) or as in (14). With this, a suitable approximation of u in the LOD space is given by its $a(\cdot, \cdot)$ -orthogonal projection, *i.e.*, $P^{\text{LOD}}(u) \in V_H^{\text{LOD}}$ is the solution to

$$a(P^{\text{LOD}}(u), v) = a(u, v) \quad \text{for all } v \in V_H^{\text{LOD}}.$$

The central question is now how accurately $P^{\text{LOD}}(u)$ approximates the important time invariants introduced in Section 2.2 and if we can obtain similar rates as for the ground state. A positive answer was given in [54], where it was proved that for sufficiently smooth u and for a sufficiently smooth potential V , the energy (cf. (3)) and the mass are both approximated with 6th order accuracy, *i.e.*,

$$|E(u) - E(P^{\text{LOD}}(u))| \leq CH^6 \quad \text{and} \quad \left| \|u\|^2 - \|P^{\text{LOD}}(u)\|^2 \right| \leq CH^6, \quad (15)$$

where the constant C depends on u , V , β and parameters that characterize the quasi-uniformity and the regularity of the mesh. Similar results are obtained for the momentum and the center of mass in the absence of the trapping potential; see [54]. We note however that the optimal rates (15) can be only formally guaranteed for the solution u of the time-dependent GPE if the potential fulfills $V \in H^2(\mathcal{D})$, though this can be relaxed to some extent for suitable initial values (cf. Rem. 3.1 below).

In conclusion we can summarize that, both for analytically given (sufficiently regular) initial values u_0 and precomputed ground states, the LOD spaces allow to approximate the energy with 6th order accuracy. Preserving it over time is then subject to a suitable time discretization of the GPE.

Remark 3.1 (Well-prepared initial values for optimal accuracy). As mentioned before, the initial values for the time-dependent GPE are often selected as ground states. Now let us consider such a setting and denote by V^{gs} the potential used for describing the ground states and by V^{t} the potential in the time-dependent GPE. Let us assume that both potentials split into a regular and into a rough contribution, *i.e.*, $V^i = V_0^i + V_d^i$ with $V_0^i \in H^2(\mathcal{D})$ and $V_d^i \in L^\infty(\mathcal{D})$ where $i = \text{gs}, \text{t}$. If both problems are discretized with the same LOD space, then optimal convergence orders for the energy can only be guaranteed if the rough contributions of the potentials are identical (*i.e.*, $V_d^{\text{gs}} = V_d^{\text{t}}$) and if V_d^{gs} is included in the construction of the LOD space. In this case, we can interpret u_{gs} as a well-prepared initial value for the time-dependent GPE because V_d^{t} will only mildly reduce the time-regularity for $u(t)$ (cf. [50], Appendix A). A rigorous error analysis in this setting is however still open.

3.3. Practical approximation of the ideal LOD-spaces

Before we can start discussing implementation details, we need to revisit the idealized LOD spaces and how they are approximated in practice by a two level discretization. We will do this exemplarily for the space constructed based on the bilinear form $a(\cdot, \cdot) = a_V(\cdot, \cdot)$ from (13) but we can easily replace it by (14).

An approximation of the ideal space V_H^{LOD} is necessary since the canonical set of basis functions of V_H^{LOD} consists of functions $(I - \mathcal{Q})\varphi_j$, where φ_j are the nodal basis functions of V_H and $\mathcal{Q} : H_0^1(\mathcal{D}) \rightarrow W$ denotes the correction operator given by (9). As the computation of $\mathcal{Q}\varphi_j$ requires the solution of an elliptic problem in the infinite-dimensional Hilbert space W (cf. (9)), we need to discretize it on a second finite element mesh $\mathcal{T}_h$. The mesh $\mathcal{T}_h$ needs to be sufficiently fine and is usually selected as a subdivision of the coarse mesh $\mathcal{T}_H$ in order

to simplify the implementation. The corresponding P1 finite element space V_h on $\mathcal{T}_h$ thus contains the coarse space, *i.e.*, $V_H \subset V_h \subset H_0^1(\mathcal{D})$. With this, we could solve the corrector problem (9) in the discrete detail space

$$W_h := V_h \cap W,$$

leading to a discrete corrector problem of the form

$$\text{find } \mathcal{Q}_h \varphi_j \in W_h : \quad a(\mathcal{Q}_h \varphi_j, w_h) = a(\varphi_j, w_h) \quad \text{for all } w_h \in W_h. \quad (16)$$

However, in this case a second issue remains, that is, that even though the right hand side of (16) is local (due to the local support of φ_j), the solution $\mathcal{Q}_h \varphi_j$ has (in general) support on the whole domain $\mathcal{D}$, which would make the repeated solving for each basis function φ_j very expensive. Luckily, due to the particular solution space W_h , the functions $\mathcal{Q}_h \varphi_j$ are exponentially decaying (in units of the coarse mesh size H) outside of the support of φ_j , *i.e.*, outside the nodal patch $\mathcal{D}_j$. This property was first proved in [66] and justifies that the global corrector problems can be truncated to small local patches. This makes the solving feasible and the expenses for assembling an approximation of V_H^{LOD} remain economical. From the view point of approximation theory it is important that the local basis functions are constructed such that they still form a partition of unity in order to avoid numerical pollution [70].

In the following we shall present a suitable localization strategy to approximate $\mathcal{Q}_h \varphi_j$. The strategy is commonly associated with the classical LOD and was suggested in [46, 49]. For that, let $v_H \in V_H$ be an arbitrary function in the coarse space V_H (in practice selected as a nodal basis function φ_j) and let $K \in \mathcal{T}_H$ be a coarse simplex. For the localization we require a truncation parameter $\ell \in \mathbb{N}_{>0}$ that characterizes the size of local patches. To be precise, we define the ℓ -layer patch around K iteratively by

$$S_\ell(K) := \bigcup \left\{ \hat{K} \in \mathcal{T}_H \mid \hat{K} \cap S_{\ell-1}(K) \neq \emptyset \right\} \quad \text{and} \quad S_0(K) := K.$$

This means that $S_\ell(K)$ consists of K and ℓ layers of grid elements around it. This also implies that $S_\ell(K)$ has a diameter of order $\mathcal{O}(\ell H)$. The restriction of W_h to $S_\ell(K)$ is given by

$$W_h(S_\ell(K)) := W_h \cap H_0^1(S_\ell(K))$$

and the restriction of the bilinear form $a(\cdot, \cdot)$ to some local subset $S \subset \mathcal{D}$ by

$$a_S(u, v) := \frac{1}{2}(\nabla u, \nabla v)_{L^2(S)} + (Vu, v)_{L^2(S)}.$$

With all these notations, the K -localized corrector $\mathcal{Q}_{h,K}^\ell : V_H \rightarrow W_h(S_\ell(K))$ is given by

$$a_{S_\ell(K)}(\mathcal{Q}_{h,K}^\ell v_H, w_h) = a_K(v_H, w_h) \quad \text{for all } w_h \in W_h(S_\ell(K)). \quad (17)$$

Note that computing $\mathcal{Q}_{h,K}^\ell v_H$ involves solving a local finite element problem on the (small) domain $S_\ell(K)$. The global approximation of the idealized corrector $\mathcal{Q}_h$ (cf. (16)) is then defined by

$$\mathcal{Q}_h^\ell v_H := \sum_{K \in \mathcal{T}_H} \mathcal{Q}_{h,K}^\ell v_H.$$

It is easy to see that if ℓ is large enough such that $S_\ell(K) = \mathcal{D}$, then we have

$$\mathcal{Q}_h^\ell v_H = \mathcal{Q}_h v_H,$$

i.e., the approximated corrector coincides with the ideal corrector. In general, the error between $\mathcal{Q}_h^\ell$ and $\mathcal{Q}_h$ can be quantified by the aforementioned exponential decay as

$$\|\mathcal{Q}_h^\ell v_H - \mathcal{Q}_h v_H\|_{H^1(\mathcal{D})} \leq C \exp(-\rho \ell) \|v_H\|_{H^1(\mathcal{D})}, \quad (18)$$

where $C > 0$ and $\rho > 0$ are generic constants that depend on the mesh regularity and on the coercivity and continuity constants of $a(\cdot, \cdot)$. For a proof we refer to [46, 49].

In consequence, we can define the approximated LOD space by

$$V_{H,\text{LOD}}^{h,\ell} := \{(I - \mathcal{Q}_h^\ell)v_H \mid v_H \in V_H\},$$

which is spanned by the basis functions $(I - \mathcal{Q}_h^\ell)\varphi_j$. Naturally, the localization introduced through $\mathcal{Q}_h^\ell$ becomes a source of numerical error. However, as indicated by (18), the approximation properties of $V_{H,\text{LOD}}^{h,\ell}$ are the same as for V_H^{LOD} , up to an exponential error term of the form $\exp(-\rho\ell)$. For rigorous proofs of this claim, we refer to [67]. As a practical remark, selecting ℓ proportional to $|\log(H)|$ with a suitable prefactor allows to fully reproduce the high convergence orders stated in Sections 3.2.1 and 3.2.2.

Hence, in practice we can work with the space $V_{H,\text{LOD}}^{h,\ell}$ without sacrificing significant accuracy. The advantages of $V_{H,\text{LOD}}^{h,\ell}$ are apparent: On the one hand, the basis functions $(I - \mathcal{Q}_h^\ell)\varphi_j$ have a local support (of order $|\log(H)|H$) which is important to obtain sparse system matrices after the spatial discretization, both for the eigenvalue problem and for the time-dependent problem. On the other hand, a basis function $(I - \mathcal{Q}_h^\ell)\varphi_j$ is cheap to compute. It requires to solve for $\mathcal{Q}_{h,K}^\ell\varphi_j$ for every coarse element K that is in the nodal patch $\mathcal{D}_j$. Each of the problems (17) is small (only formulated on $S_\ell(K)$) and they are all independent from each other and can be solved in parallel. Also note that problem (17) can be written as a saddle point problem where the constraint entering through the space $W_h(S_\ell(K))$ is represented by a Lagrange multiplier. The solution space for the saddle point problem now becomes the unconstraint space $V_h(S_\ell(K))$. This is exploited in practical computations.

The implementation of the above constructed LOD space is described comprehensively in [38] which is why we will not repeat it here. There is only one aspect that we would like to mention as it becomes important when discussing the implementation of the numerical schemes for the Gross–Pitaevskii equation: Once the LOD basis functions are computed, they are stored and processed through a matrix. To be a bit more specific, let φ_i , $i = 1, \dots, N_H$ denote the (nodal) basis functions of V_H , let ψ_j , $j = 1, \dots, N_h$ denote the (nodal) basis functions of V_h , and let $\mathbf{P} \in \mathbb{R}^{N_H \times N_h}$ denote the interpolation matrix containing the values of each coarse basis function φ_i , in every fine node, so that

$$\varphi_i = \sum_{j=1}^{N_h} \mathbf{P}_{ij} \psi_j \quad \text{for all } 1 \leq i \leq N_H.$$

Furthermore, we can store the coefficients of the functions $\mathcal{Q}_h^\ell\varphi_i \in V_h$ (for each $1 \leq i \leq N_H$) row-wise in a corrector matrix $\mathbf{Q} \in \mathbb{R}^{N_H \times N_h}$ such that

$$\mathcal{Q}_h^\ell\varphi_i = \sum_{j=1}^{N_h} \mathbf{Q}_{ij} \psi_j \quad \text{for all } 1 \leq i \leq N_H.$$

Then the representation of the LOD-basis functions on the fine mesh is given by the *LOD-matrix* $\Phi^{\text{LOD}} = \mathbf{P} - \mathbf{Q}$, such that

$$\varphi_i^{\text{LOD}} = \sum_{j=1}^{N_h} \Phi_{ij}^{\text{LOD}} \psi_j \quad \text{for all } 1 \leq i \leq N_H$$

is a basis of the LOD space $V_{H,\text{LOD}}^{h,\ell}$. With this, the standard stiffness matrix can be assembled easily through matrix multiplications, *e.g.*, for the stiffness matrix in the LOD space $\mathbf{A}_{ij}^{\text{LOD}} = (\nabla \varphi_i^{\text{LOD}}, \nabla \varphi_j^{\text{LOD}})$ we have $\mathbf{A}^{\text{LOD}} = (\mathbf{P} - \mathbf{Q})\mathbf{A}(\mathbf{P} - \mathbf{Q})^\top$ where $\mathbf{A}_{ij} = (\nabla \psi_i, \nabla \psi_j)$ is the stiffness matrix on the fine mesh. Similarly, we can assemble the mass matrix and the matrix representing the potential term in the GPE. However, this simple strategy fails once nonlinear terms are considered as they will appear in the GPE. In the next subsection we present a practical solution for this issue.

3.4. Treatment of nonlinear terms

The super-approximation properties of the LOD-space reviewed in Section 3.2 can be further leveraged to substantially speed up the nonlinear computations. This is however not obvious since the GPE involves a cubic nonlinearity of the form $(|u|^2 u, v)$ which has to be discretized in the LOD space. With a standard discretization (and suitable linearization), this would essentially involve a product of the form $(|v^{\text{LOD}}|^2 \varphi_i^{\text{LOD}}, \varphi_m^{\text{LOD}})$, where $v^{\text{LOD}} \in V_{H, \text{LOD}}^{h, \ell}$ is a function in the LOD space and $\varphi_i^{\text{LOD}} = (I - \mathcal{Q}_h^\ell) \varphi_i$ again denotes the LOD basis function belonging to the node with index i . The necessary precomputations for easily evaluating such a term would require to assemble and store a 4-valence tensor in the LOD space, which easily exceeds the computing and memory resources due to the curse of dimensionality.

To this end, we introduce the *sparse* 3-valence tensor

$$\omega_{ijk} := (\varphi_i^{\text{LOD}} \varphi_j^{\text{LOD}}, \varphi_k^{\text{LOD}}), \quad 1 \leq i, j, k \leq N_H \quad (19)$$

and replace throughout expressions of the type $(|v|^2 u, w)$ in the GPE with $(P_{L^2}^{\text{LOD}}(|v|^2) u, w)$, where $P_{L^2}^{\text{LOD}}$ is the L^2 -projection onto the LOD space, i.e., $P_{L^2}^{\text{LOD}}(|v|^2) \in V_{H, \text{LOD}}^{h, \ell}$ solves

$$(P_{L^2}^{\text{LOD}}(|v|^2), \varphi_i^{\text{LOD}}) = (|v|^2, \varphi_i^{\text{LOD}}) \quad \text{for all } 1 \leq i \leq N_H. \quad (20)$$

With this notation, the projection $P_{L^2}^{\text{LOD}}(|v_{\text{LOD}}|^2)$ for an LOD function v_{LOD} is easily computed using ω . In fact, the LOD-mass matrix describing the left hand side of (20) is available (typically preassembled as sketched in the previous subsection) and the entries of the right hand side vector $(|v_{\text{LOD}}|^2, \varphi_i^{\text{LOD}})$ are obtained from ω . To be precise, if $v_{\text{LOD}} = \sum_k \alpha_k \varphi_k^{\text{LOD}}$ for $\alpha_k \in \mathbb{C}$, then the entries of the right hand side vector in (20) can be expressed as

$$\begin{aligned} (|v_{\text{LOD}}|^2, \varphi_i^{\text{LOD}}) &= \left(\sum_k \sum_j \alpha_k \overline{\alpha_j} \varphi_k^{\text{LOD}} \varphi_j^{\text{LOD}}, \varphi_i^{\text{LOD}} \right) = \sum_{k \leq j} \Re(\alpha_k \overline{\alpha_j}) (2 - \delta_{kj}) (\varphi_k^{\text{LOD}} \varphi_j^{\text{LOD}}, \varphi_i^{\text{LOD}}) \\ &= \sum_{k \leq j} \Re(\alpha_k \overline{\alpha_j}) (2 - \delta_{kj}) \omega_{kji}, \quad \text{where } \omega_{kji} = (\varphi_k^{\text{LOD}} \varphi_j^{\text{LOD}}, \varphi_i^{\text{LOD}}). \end{aligned}$$

Now that (20) is algebraically assembled, we can cheaply solve the arising system of equations to obtain a vector representation of $P_{L^2}^{\text{LOD}}(|v_{\text{LOD}}|^2)$, say,

$$P_{L^2}^{\text{LOD}}(|v_{\text{LOD}}|^2) = \sum_k \rho_k \varphi_k^{\text{LOD}} \quad \text{for some } \rho \in \mathbb{C}^{N_H}.$$

With this, nonlinear terms of the form $(P_{L^2}^{\text{LOD}}(|v_{\text{LOD}}|^2) u_{\text{LOD}}, \varphi_i^{\text{LOD}})$ are then straightforwardly assembled as

$$(P_{L^2}^{\text{LOD}}(|v_{\text{LOD}}|^2) u_{\text{LOD}}, \varphi_i^{\text{LOD}}) = \sum_{kj} \rho_k \alpha_j \omega_{kji}.$$

We note that the threefold symmetry of ω implies that only nonzero elements for which $i \leq j \leq k$, must be computed. Using the 3-valence tensor instead of the canonical 4-valence tensor reduced the preassembly time and the memory demand tremendously and ultimately makes the implementation of the LOD-based numerical methods efficient. In the Appendix A we give more details on the implementation of the 3-valence tensor ω .

Remarkably, the error introduced through replacing $|v|^2$ by $P_{L^2}^{\text{LOD}}(|v|^2)$ at all occurrences seems to be only of order $\mathcal{O}(H^8)$ and is hence negligible. Heuristically, this can be motivated by the following simple calculation. First, we observe that for any $u, v \in H_0^1(\mathcal{D})$ it holds:

$$\int_{\mathcal{D}} |u|^2 v - P_{L^2}^{\text{LOD}}(|u|^2) v \, dx = \int_{\mathcal{D}} (|u|^2 - P_{L^2}^{\text{LOD}}(|u|^2)) (v - P_{L^2}^{\text{LOD}}(v)) \, dx.$$

If u, v and V are sufficiently smooth, that is $(-\frac{1}{2}\Delta + V)|u|^2 \in H_0^1(\mathcal{D}) \cap H^2(\mathcal{D})$ and $(-\frac{1}{2}\Delta + V)v \in H_0^1(\mathcal{D}) \cap H^2(\mathcal{D})$, the approximation properties of the LOD space hence yield

$$\left| \int_{\mathcal{D}} |u|^2 v - P_{L^2}^{\text{LOD}}(|u|^2) v \, dx \right| \leq \| |u|^2 - P_{L^2}^{\text{LOD}}(|u|^2) \| \|v - P_{L^2}^{\text{LOD}}(v)\| \leq CH^8. \quad (21)$$

To make this argument rigorous for a particular method requires knowledge of the regularity of u and v (which are typically both related to the exact solution at certain time instances). In [54], it was rigorously demonstrated for a modified Crank–Nicolson method for the time-dependent GPE.

4. FULLY-DISCRETE APPROXIMATION OF GROUND STATES

In this section we discuss an efficient method for computing ground states in the LOD space. For this, let $V_{H,\text{LOD}}^{h,\ell}$ denote the corresponding space, typically constructed based on the bilinear form $a(\cdot, \cdot) = a_V(\cdot, \cdot)$ given by (13). Recalling the setting of Section 3.2.1, we are interested in finding an approximation to the discrete ground state

$$u_{\text{gs}}^{\text{LOD}} = \arg \min_{v \in \mathbb{S} \cap V_{H,\text{LOD}}^{h,\ell}} E(v). \quad (22)$$

This can be achieved with a Riemannian gradient method that was originally proposed in [51]. To keep the presentation and the motivation short, we will however introduce the method here as an inverse iteration with damping. Before we formulate a corresponding fully-discrete version of the method in the LOD space, let us first sketch the idea in the idealized analytical setting (4), where we seek $u_{\text{gs}} \in H_0^1(\mathcal{D})$ with

$$u_{\text{gs}} = \arg \min_{v \in \mathbb{S}} E(v).$$

Recall here that $\mathbb{S}$ is the Riemannian manifold $\mathbb{S} := \{v \in H_0^1(\mathcal{D}) \text{ and } \|v\| = 1\}$. As the minimization problem can be equivalently written as a nonlinear eigenvalue problem (5) seeking the eigenfunction to the smallest eigenvalue λ_{gs} , a natural choice is the classical inverse iteration method, where the approximation from the previous iteration can be used to linearize the differential operator. For that, let us denote by

$$\mathcal{A}_z^{-1} : H^{-1}(\mathcal{D}) \rightarrow H_0^1(\mathcal{D})$$

the inverse of the linearized Gross–Pitaevskii operator in a linearization point $z \in H_0^1(\mathcal{D})$. To make the definition of $\mathcal{A}_z^{-1}$ precise, we consider the bilinear form associated with the z -linearized Gross–Pitaevskii operator given by

$$a_z(v, w) := \frac{1}{2}(\nabla v, \nabla w) + (V + \beta|z|^2)v, w). \quad (23)$$

With this, the preimage $\mathcal{A}_z^{-1}F \in H_0^1(\mathcal{D})$ for some $F \in H^{-1}(\mathcal{D})$ is given as the solution to the linear elliptic problem

$$a_z(\mathcal{A}_z^{-1}F, v) = \langle F, v \rangle \quad \text{for all } v \in H_0^1(\mathcal{D}).$$

If $\mathcal{I} : H_0^1(\mathcal{D}) \rightarrow H^{-1}(\mathcal{D})$ denotes the canonical L^2 -identity given by $\langle \mathcal{I}u, v \rangle := (u, v)$, then the standard inverse iteration would read

$$u^{(k+1)} := \frac{\mathcal{A}_{u^{(k)}}^{-1} \mathcal{I}u^{(k)}}{\|\mathcal{A}_{u^{(k)}}^{-1} \mathcal{I}u^{(k)}\|}.$$

However, in our nonlinear setting, this is not sufficient to guarantee global convergence to the ground state. For that we combine the inverse iteration with a damping strategy to obtain the modified iterations as

$$u^{(k+1)} := \frac{(1 - \theta_k)u^{(k)} + \theta_k \gamma_k \mathcal{A}_{u^{(k)}}^{-1} \mathcal{I}u^{(k)}}{\|(1 - \theta_k)u^{(k)} + \theta_k \gamma_k \mathcal{A}_{u^{(k)}}^{-1} \mathcal{I}u^{(k)}\|}, \quad \text{where } \gamma_k := \left(u^{(k)}, \mathcal{A}_{u^{(k)}}^{-1} \mathcal{I}u^{(k)}\right)^{-1} \quad (24)$$

and the damping parameter $\theta_k \in (0, 2)$ is determined numerically by line search (cf. [51]) such that the energy becomes minimal, *i.e.*,

$$\theta_k := \arg \min_{\theta > 0} E(u^{(k+1)}(\theta)),$$

where $u^{(k+1)}(\theta)$ is defined by (24) if θ_k is replaced by the variable parameter θ . The efficient implementation of the line search (which comes at negligible costs) is described in [51]. The appearance of the parameter γ_k in addition to the damping parameter θ_k can be explained from the view point of Riemannian gradient methods, where we again refer to [51] for details. The fundamental advantage of the damped inverse iteration (24) is that it converges globally to the ground state for any nonnegative starting value $u^{(0)} \in \mathcal{S}$. Furthermore, the method is strictly energy diminishing (cf. [51]). The guaranteed convergence is a unique feature, which makes the approach very attractive for practical purposes.

It is straightforward to discretize the damped inverse iterations (24) by replacing the analytical functions $u^{(k)}$ by corresponding approximations in the LOD space. However, as discussed in Section 3.4, this direct discretization would require the storing of a 4-valence tensor of products of LOD-basis functions, which is computationally not feasible. Hence, we follow the strategy described in Section 3.4 and replace $|z|^2$ in the definition of $a_z(\cdot, \cdot)$ (23) by $P_{L^2}^{\text{LOD}}(|z|^2)$, where we recall $P_{L^2}^{\text{LOD}}$ as the L^2 -projection into the LOD space given by (20).

With this we can now explicitly formulate the fully-discrete method in the LOD space. For that, let $u_{\text{LOD}}^{(0)} \in V_{H,\text{LOD}}^{h,\ell}$ be a given initial value and $u_{\text{LOD}}^{(k)} \in V_{H,\text{LOD}}^{h,\ell}$ the current iterate. The corresponding linearized bilinear form is given by

$$a_k^{\text{LOD}}(v, w) := \frac{1}{2}(\nabla v, \nabla w) + \left(\left(V + \beta P_{L^2}^{\text{LOD}} \left(|u^{(k)}|^2 \right) \right) v, w \right).$$

To keep the notation short, we denote the discretized inverse (as an approximation of $\mathcal{A}_{u^{(k)}}^{-1}$) by $\mathcal{D}_k^{-1}$, *i.e.*, for $F \in H^{-1}(\mathcal{D})$, the inverse $\mathcal{D}_k^{-1}F \in V_{H,\text{LOD}}^{h,\ell}$ is defined as the solution to

$$a_k^{\text{LOD}}(\mathcal{D}_k^{-1}F, v) = \langle F, v \rangle \quad \text{for all } v \in V_{H,\text{LOD}}^{h,\ell}.$$

Straightforwardly, we now obtain the next iterate $u_{\text{LOD}}^{(k+1)} \in V_{H,\text{LOD}}^{h,\ell}$ by the corresponding discrete damped inverse iteration:

$$u_{\text{LOD}}^{(k+1)} := \frac{(1 - \theta_k)u_{\text{LOD}}^{(k)} + \theta_k \gamma_k^{\text{LOD}} \mathcal{D}_k^{-1} \mathcal{I} u_{\text{LOD}}^{(k)}}{\left\| (1 - \theta_k)u_{\text{LOD}}^{(k)} + \theta_k \gamma_k^{\text{LOD}} \mathcal{D}_k^{-1} \mathcal{I} u_{\text{LOD}}^{(k)} \right\|}, \quad \text{where } \gamma_k^{\text{LOD}} := \left(u_{\text{LOD}}^{(k)}, \mathcal{D}_k^{-1} \mathcal{I} u_{\text{LOD}}^{(k)} \right)^{-1}. \quad (25)$$

The optimal value of the damping parameter $\theta_k \in (0, 2)$ is determined by the property

$$\theta_k := \arg \min_{\theta > 0} E_{\text{LOD}}(u_{\text{LOD}}^{(k+1)}(\theta))$$

for the modified energy

$$E_{\text{LOD}}(v) := \int_{\mathcal{D}} \frac{1}{2} |\nabla v|^2 + V|v|^2 + \frac{\beta}{2} |v|^2 P_{L^2}^{\text{LOD}}(|v|^2) \, dx. \quad (26)$$

Note that each iteration (25) requires the solving of one elliptic problem involving the computation of $P_{L^2}^{\text{LOD}}(|u^{(k)}|^2)$ and afterwards one more equation to compute $\mathcal{D}_k^{-1} \mathcal{I} u_{\text{LOD}}^{(k)}$. Recall here from Section 3.4 that the computation of $P_{L^2}^{\text{LOD}}(|u^{(k)}|^2)$ is cheap, as it only involves the (nicely conditioned and precomputed) mass matrix in the LOD-space and the precomputed 3-valence tensor. Similarly, all components that are needed

to assemble the system matrix $\mathcal{D}_k^{-1} \mathcal{I} u_{\text{LOD}}^{(k)}$ are also precomputed. This makes the iterations in the LOD-space highly efficient. Finally, let us note that the LOD-specific iterations proposed in equation (25) have (in this particular modification) not yet been studied in the literature, neither the convergence in terms of iterations nor the approximation properties of the limit function for $k \rightarrow \infty$. However, our numerical results (cf. Sect. 6) indicate that we can expect the same order of accuracy (cf. Sect. 3.2.1) as for the ideal LOD ground state $u_{\text{gs}}^{\text{LOD}}$ given by (22).

5. FULLY-DISCRETE APPROXIMATION OF THE TIME-DEPENDENT PROBLEM

In this section we present a fully-discrete method to solve the time-dependent GPE (7). For the spatial discretization we use again the generalized finite element space $V_{H,\text{LOD}}^{h,\ell}$ based on the Localized Orthogonal Decomposition which was discussed in the previous sections. Hence, we now focus on the time discretization for which we use an energy-conserving time integrator based on continuous Galerkin time stepping. The time stepping method was first introduced and analyzed in [57] for the nonlinear Schrödinger equation in two dimensions and generalized in [36] for the Gross–Pitaevskii equation in up to three dimensions. However, both works assume standard finite element spaces of order p for the space discretization such as the space of H^1 -conforming Lagrange finite elements. In contrast to that, we now combine this time integrator with the LOD discretization in space. For that purpose, we consider (7) on a time interval $[0, T]$ and introduce an equidistant time grid $0 = t_0 < t_1 < \dots < t_N = T$ of n time intervals $I_n = (t_n, t_{n+1}]$ with step size $\tau = t_{n+1} - t_n > 0$. Of course, we could also choose a non-equidistant time grid, cf. [36, 57], but we assume a constant step size τ for simplicity. Next we choose a polynomial degree $q \in \mathbb{N}$ and define the space

$$\mathcal{W}_{\tau,q,\text{LOD}}^{H,h,\ell} := \left\{ v : \mathcal{D} \times (0, T] \rightarrow \mathbb{C} : v|_{\mathcal{D} \times I_n}(x, t) = \sum_{j=0}^q t^j \varphi_{nj}(x), \text{ where } \varphi_{nj} \in V_{H,\text{LOD}}^{h,\ell} \right\}.$$

Hence, $\mathcal{W}_{\tau,q,\text{LOD}}^{H,h,\ell}$ consists of piecewise polynomials of degree q in time with values in the LOD space $V_{H,\text{LOD}}^{h,\ell}$. With this, the numerical solution scheme for the time-dependent GPE, to which we refer as the cG-LOD-method, reads as follows:

For a given initial value $u_0 \in H_0^1(\mathcal{D})$ we seek $u_\tau^{\text{LOD}} \in \mathcal{W}_{\tau,q,\text{LOD}}^{H,h,\ell}$ such that for all $n = 0, \dots, N-1$

$$\begin{aligned} \int_{I_n} (\mathrm{i} \partial_t u_\tau^{\text{LOD}}, v) - a_V(u_\tau^{\text{LOD}}, v) - \beta \left(P_{L^2}^{\text{LOD}}(|u_\tau^{\text{LOD}}|^2) u_\tau^{\text{LOD}}, v \right) dt &= 0 \quad \forall v \in \mathcal{W}_{\tau,q-1,\text{LOD}}^{H,h,\ell}, \\ u_\tau^{\text{LOD},n+} &= u_\tau^{\text{LOD}}(t_n), \end{aligned} \quad (27)$$

where $u_\tau^{\text{LOD},n+} = \lim_{t \searrow t_n} u_\tau^{\text{LOD}}(t)$ and $u_\tau^{\text{LOD},0+} = P^{\text{LOD}}(u_0)$ and the bilinear form $a_V(\cdot, \cdot)$ is given by (13). We note that according to the previous sections the cG-LOD-method (27) proposed here uses the modified nonlinearity $P_{L^2}^{\text{LOD}}(|u|^2)u$ instead of the original cubic nonlinearity $|u|^2u$. As it was discussed in Section 3.4, this decreases the computational effort significantly when assembling the nonlinear terms. Nevertheless, this causes that only the modified energy E_{LOD} from (26) is preserved by the time integration rather than the exact energy E . More precise, when testing in (27) with $v = \partial_t u_\tau^{\text{LOD}}$ one finds

$$E_{\text{LOD}}(u_\tau^{\text{LOD}}(t_n)) = E_{\text{LOD}}(P^{\text{LOD}}(u_0)) \quad \text{for all } n = 0, \dots, N.$$

For the Crank–Nicolson time integrator it was shown in [54] that this is enough to approximate the exact energy $E(u)$ with an accuracy of $\mathcal{O}(H^6)$ independent on τ . However, a corresponding result for the cG-LOD-method (27), as well as a rigorous error analysis, is still left for future work.

Next we describe how the time stepping procedure of the cG-LOD-method (27) can be realized in an implementation. For this purpose, the method is reformulated using sufficiently high quadrature rules that integrate the time integrals in (27) exactly. We denote by s_j and w_j for $j = 1, \dots, q$ the nodes and weights of the q -stage

Gauss–Legendre quadrature rule on the unit interval $[0, 1]$ which is exact for polynomials of order $2q - 1$. Further, we set formally $s_0 = 0$ and define the Lagrange basis polynomials L_i , $i = 1, \dots, q$ of degree $q - 1$ associated with $s_1, \dots, s_q$ and the Lagrange basis polynomials $\hat{L}_i$, $i = 0, \dots, q$ of degree q associated with $s_0, \dots, s_q$ by

$$L_i(s) = \prod_{j=1, j \neq i}^q \frac{s - s_j}{s_i - s_j}, \quad \hat{L}_i(s) = \prod_{j=0, j \neq i}^q \frac{s - s_j}{s_i - s_j} \quad \text{for } s \in [0, 1].$$

The nodes, weights and basis polynomials are easily transformed to $\bar{I}_n$ so that we define further

$$\begin{aligned} t_{n,i} &:= t_n + \tau s_i, & w_{n,i} &:= \tau w_i, \\ L_{n,i}(t_n + s\tau) &:= L_i(s), & \hat{L}_{n,i}(t_n + \tau s) &:= \hat{L}_i(s). \end{aligned}$$

Then u_τ^{LOD} is uniquely expressed on I_n via

$$u_\tau^{\text{LOD}}(t) = \sum_{j=0}^q \hat{L}_{n,j}(t) u_\tau^{n,j} \quad \text{for } t \in I_n \quad (28)$$

with suitable $u_\tau^{n,j} := u_\tau^{\text{LOD}}(t_{n,j}) \in V_{H,\text{LOD}}^{h,\ell}$. Testing in (27) with $v(t) = L_{n,i}(t)\psi$ for arbitrary $\psi \in V_{H,\text{LOD}}^{h,\ell}$, we can rewrite the cG-LOD-method (27) as

$$\sum_{j=0}^q m_{ij}(u_\tau^{n,j}, \psi) + i\tau w_i a_V(u_\tau^{n,i}, \psi) + i\beta \int_{I_n} L_{n,i}(t) \left(P_{L^2}^{\text{LOD}}(|u_\tau^{\text{LOD}}|^2) u_\tau^{\text{LOD}}, \psi \right) dt = 0 \quad (29)$$

for all $\psi \in V_{H,\text{LOD}}^{h,\ell}$ and $i = 1, \dots, q$. Here the coefficients m_{ij} are given by

$$m_{ij} := \int_0^1 \hat{L}'_j(s) L_i(s) ds.$$

Next we observe that the integrand of the remaining time integral in (29) is of degree $4q - 1$ in time. Hence the integral is integrated exactly by a $2q$ -stage Gauss–Legendre quadrature rule for which we denote by $\tilde{s}_j$ and $\tilde{w}_j$ for $j = 1, \dots, 2q$ the corresponding nodes and weights on the unit interval $[0, 1]$. Setting formally $g(u) := P_{L^2}^{\text{LOD}}(|u|^2)u$ we observe, in view of (28), that (27) is equivalent to

$$\sum_{j=0}^q m_{ij}(u_\tau^{n,j}, \psi) + i\tau w_i a_V(u_\tau^{n,i}, \psi) + i\beta \tau \sum_{\nu=1}^{2q} \tilde{w}_\nu L_i(\tilde{s}_\nu) \left(g\left(\sum_{j=0}^q \hat{L}_j(\tilde{s}_\nu) u_\tau^{n,j}\right), \psi \right) = 0 \quad (30)$$

for all $\psi \in V_{H,\text{LOD}}^{h,\ell}$ and $i = 1, \dots, q$. Hence in each time step one has to solve for given $u_\tau^{\text{LOD}}(t_n) = u_\tau^{n,0}$ the nonlinear system (30) for $u_\tau^{n,j}$, $j = 1, \dots, q$ so that by the continuity condition in (27) the solution at the new time instance t_{n+1} is given by

$$u_\tau^{\text{LOD}}(t_{n+1}) = \sum_{j=0}^q \hat{L}_j(1) u_\tau^{n,j}.$$

Note that the system (30) is a fully coupled system for the unknowns $u_\tau^{n,j}$, $j = 1, \dots, q$. However, as proposed in [57], the system can be decoupled by observing that for $\mathcal{M} := (m_{ij})_{i,j=1,\dots,q}$ and $W := \text{diag}(w_1, \dots, w_q)$ the matrix $\mathcal{M}^{-1}W$ is the coefficient matrix of the q -stage Gauss–Legendre Runge–Kutta method. Therefore, it is

diagonalizable (cf. [34]) such that $\Sigma \mathcal{M}^{-1} W \Sigma^{-1} = \Gamma = \text{diag}(\gamma_1, \dots, \gamma_q)$ for some matrix $\Sigma \in \mathbb{R}^{q \times q}$. Defining the new unknowns

$$U^{n,i} = \sum_{j=1}^q \Sigma_{ij} u_\tau^{n,j}, \quad i = 1, \dots, q$$

in each time step one infers that (30) is equivalent to

$$(U^{n,i}, \psi) + i\tau\gamma_i a_V(U^{n,i}, \psi) = A_i(u_\tau^{n,0}, \psi) - i\beta\tau \sum_{\nu=1}^q B_{i\nu} \left(g \left(C_{0\nu} u_\tau^{n,0} + \sum_{j=1}^q C_{j\nu} U^{n,j} \right), \psi \right) \quad (31)$$

for all $\psi \in V_{H,\text{LOD}}^{h,\ell}$ and $i = 1, \dots, q$. Here the coefficients $A_i, B_{i\nu}, C_{0\nu}, C_{i\nu}$ for $i = 1, \dots, q$ and $\nu = 1, \dots, 2q$ are given by

$$A_i = \sum_{j=1}^q \Sigma_{ij}, \quad B_{i\nu} = \tilde{w}_\nu \sum_{j=1}^q (\Sigma \mathcal{M})_{ij}^{-1} L_j(\tilde{s}_\nu), \quad C_{0\nu} = \hat{L}_0(\tilde{s}_\nu), \quad C_{i\nu} = \sum_{j=1}^q \hat{L}_j(\tilde{s}_\nu) \Sigma_{ji}^{-1}.$$

In order to solve the nonlinear system of equations (31) in each time step we propose to use a fixed-point iteration. For this, we define $\mathcal{L}_{\text{LOD}} : V_{H,\text{LOD}}^{h,\ell} \rightarrow V_{H,\text{LOD}}^{h,\ell}$ via $(\mathcal{L}_{\text{LOD}} v, w) = a_V(v, w)$ so that we can formulate (31) as

$$(I + i\tau\gamma_i \mathcal{L}_{\text{LOD}}) U^{n,i} = A_i u_\tau^{n,0} - i\beta \sum_{\nu=1}^q B_{i\nu} g \left(C_{0\nu} u_\tau^{n,0} + \sum_{j=1}^q C_{j\nu} U^{n,j} \right), \quad i = 1, \dots, q.$$

The solution of this equation $U^{n,j}$, $j = 1, \dots, q$, is now approximated via an explicit fixed-point iteration reading for $k \geq 0$ as

$$(I + i\tau\gamma_i \mathcal{L}_{\text{LOD}}) U_{k+1}^{n,i} = A_i u_\tau^{n,0} - i\beta \sum_{\nu=1}^q B_{i\nu} g \left(C_{0\nu} u_\tau^{n,0} + \sum_{j=1}^q C_{j\nu} U_k^{n,j} \right), \quad i = 1, \dots, q. \quad (32)$$

Here we may choose $U_0^{n,j} = U^{n-1,j}$ from the previous time step as an initial value for the fixed-point iteration. Note that the operator $(I + i\tau\gamma_i \mathcal{L}_{\text{LOD}})$ is independent of the current time step (at least on an equidistant time grid). Hence, the resulting matrix associated with this operator in the LOD basis can be **LU**-decomposed in a pre-process before the time stepping. Using the **LU**-decomposition the system of linear equations in (32) can be solved efficiently.

6. NUMERICAL EXPERIMENTS

In this section we verify, by means of numerical experiments, the theoretical properties, efficient implementation and promising outlook of the proposed method. The tests are divided into three major parts, namely, first is verified the convergence of the damped inverse iteration in the LOD-space, subsequently the high order convergence of the time integration is verified using a test problem with known analytical solution, and finally the methods are combined to solve a physical setup in 3d. Additionally, these major parts are subdivided into smaller sections in order to compare different LOD-spaces and make cross comparisons to other spatial discretizations such as spectral and P1-Lagrange elements. Further, we demonstrate how the replacement of the density $|u_{\text{LOD}}|^2$ by its LOD projection $P_{L_2}^{\text{LOD}}(|u_{\text{LOD}}|^2)$ in our methods highly improves their performance. All CPU times were measured on a laptop with a 11th Gen Intel® Core™ i7-1165G7 @ 2.80 GHz $\times$ 8 processing unit and 64 GB of RAM.

6.1. Computation of ground states

In this section, we compute the ground state using the damped inverse iteration (25) for a smooth potential and for a discontinuous potential. Throughout we set the tolerance of the damped inverse iteration to 10^{-10} in terms of energy.

6.1.1. Smooth potential

Our first experiment considers the computation of the minimal energy and the corresponding ground state to a double-well potential in $2d$. More precise, we consider the minimization problem for the ground state

$$u_{\text{gs}} = \arg \min_{u \in \mathbb{S}} \int_{\mathcal{D}} \frac{1}{2} |\nabla v|^2 + V|v|^2 + \frac{\beta}{2} |v|^4 \, dx \quad (33)$$

with $\mathcal{D} = (-6, 6)^2$ and where the trapping potential and repulsion parameter are given by

$$V(x, y) = \frac{1}{2}(x^2 + y^2) + 4e^{-x^2/2} + 4e^{-y^2/2} \quad \text{and} \quad \beta = 50. \quad (34)$$

Recall that $\mathbb{S}$ is the Riemannian manifold $\mathbb{S} := \{v \in H_0^1(\mathcal{D}) \text{ and } \|v\| = 1\}$. We approximate the ground state solution u_{gs} of (33), (34) by a discrete minimizer in a suitable LOD-space given by (22) which is then computed with the damped inverse iteration method (25) proposed in Section 4. Besides demonstrating the performance of the damped inverse iteration method in the LOD-space, we also compare the two different choices of the bilinear form for the construction of the LOD-space, *i.e.*,

$$a_V(u, v) = \frac{1}{2}(\nabla u, \nabla v) + (Vu, v) \quad \text{and} \quad a_c(u, v) = (\nabla u, \nabla v)$$

from (13) and (14). Since V is chosen as a smooth potential we expect that both choices result in the same order of convergence w.r.t. the coarse mesh size, *i.e.*, $\mathcal{O}(H^6)$. After that, we compare the computational cost for our proposed method (25) with the method proposed in [48, 55] that does not replace the density $|u_{\text{LOD}}|^2$ by $P_{L^2}^{\text{LOD}}(|u_{\text{LOD}}|^2)$.

For the computations the fine mesh size is consistently chosen as $h = H/50$ so as to isolate the convergence in H . From a computational point of view, this is too small when H is coarse, since one would ideally choose $h \propto H^3$ so that the error in representing the LOD basis functions on the fine mesh is of the same magnitude as the best approximation error of the solution in the ideal LOD space. However, the choice of the fine mesh $h = H/50$ is chosen to isolate the best approximation error in H .

As a reference solution we choose the result of the damped inverse iteration method (25) with $h = H/80$, $H = 0.3$ and $\ell = 7$. The energy of the reference solution is computed to be $E^0 = 7.0823112$. Table 1 shows the obtained results for different values of H (and ℓ) in terms of the minimal energy together with the estimated errors and the CPU time required in the online phase. The errors are subsequently plotted in Figure 1, which clearly demonstrates the $\mathcal{O}(H^6)$ convergence rates of both choices of LOD-space. The canonical construction of the LOD-space with a_c fares better than the problem specific construction incorporating the potential. A striking example of this difference is shown in Figure 3 at which discretization level the difference in accuracy is one order of magnitude. The CPU time in the offline phase is dominated by the computation of the LOD-space and the tensor ω , these values are tabulated in Table 2. For the canonical the LOD-space it suffices, in essence (except for the near $\partial\mathcal{D}$), to compute one basis function which can then be reused and interpolated. We report similar number of iterations for both LOD-spaces and consequently similar CPU times in the online phase as tabulated in the two final columns of Table 1.

Next, we do the same experiment using the operator-adapted LOD-space but without replacement of $|u_{\text{LOD}}|^2$ by $P_{L^2}^{\text{LOD}}(|u_{\text{LOD}}|^2)$, *i.e.*, corresponding to the method analyzed in [48, 55]. The results are shown in Table 3. First of all we see that the error caused by replacing the density $|u_{\text{LOD}}|^2$ by $P_{L^2}^{\text{LOD}}(|u_{\text{LOD}}|^2)$ converge to zero in terms of the energy as the coarse mesh size tends to zero. In particular, the error is negligible compared to the error caused by the spatial discretization shown in Table 1 since it is roughly one magnitude smaller. However, the

TABLE 1. Energy E_{LOD} from (26) of the ground state obtained by (25) for different discretizations using the two different LOD-spaces constructed by a_c and a_V . The fine grid discretization was 50 times finer than the coarse, *i.e.*, $h = H/50$, except for the reference energy E^0 , for which $h = H/80$ and $H = 0.3$. Asterisk marks estimated value. Tabulated CPU time denotes to the total online time.

H	ℓ	a_c		a_V		CPU [s]	N° it
		E_{LOD}	$E_{\text{LOD}} - E^0$	E_{LOD}	$E_{\text{LOD}} - E^0$		
2.0	1	7.1131522	0.0308409	7.4010560	0.3187454	3	18
1.2	2	7.0897461	0.0074349	7.1211954	0.0388848	3	18
1.0	3	7.0841921	0.0018808	7.0897783	0.0074678	4	19
0.75	4	7.0826330	0.0003217	7.0843425	0.0020320	4	21
0.60	5	7.0823724	0.0000612	7.0825553	0.0002448	7	21
0.50	6	7.0823286	0.0000173	7.0824111	0.0001005	16	22
0.40	7	7.0823166	0.0000054	7.0823290	0.0000184	35	22
0.30	7	7.0823112	0.000001*			75	22

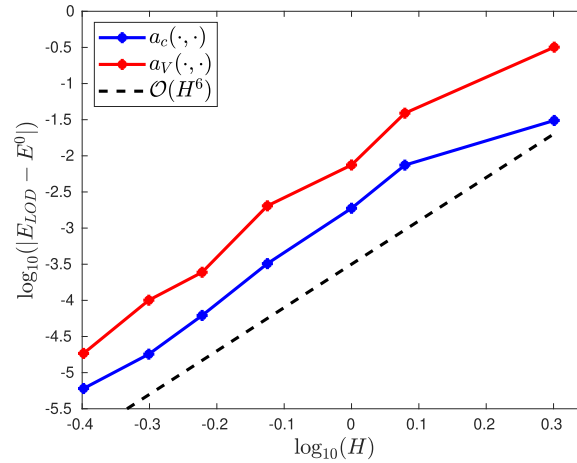


FIGURE 1. Convergence plot of values in Table 1 as measured logarithmically in error *versus* mesh size H .

computational time decreases enormously when including the projected density into the method. Our method proposed in this work is roughly 50 times faster than the method from [48, 55]. At the same time this comes with almost no loss w.r.t. its approximation properties, though this still has to be investigated analytically. This indicates a significant boost in efficiency.

6.1.2. Discontinuous potential

In the previous test case, nothing was gained by incorporating the potential into the construction of the LOD-space. For smooth problems with analytical solutions the canonical LOD-space is, computationally, a better choice. Furthermore, the smooth potential makes the previous problem well-suited for spectral methods. However, in the case of potentials with only L^∞ -regularity, such as potentials with a discontinuity, one can at most expect linear convergence w.r.t. energy using a spectral method as the Fourier coefficients of the potential decay as $\mathcal{O}(1/N)$, N being the number of Fourier modes in each dimension. In contrast, the LOD-method achieves 6th order convergence even for L^2 -potentials, provided the low-regularity potential is incorporated into

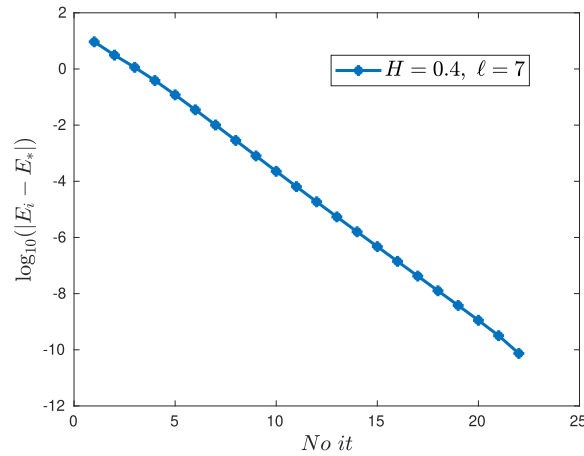


FIGURE 2. Convergence to ground state of method (24), initial guess is a normalized Gaussian.

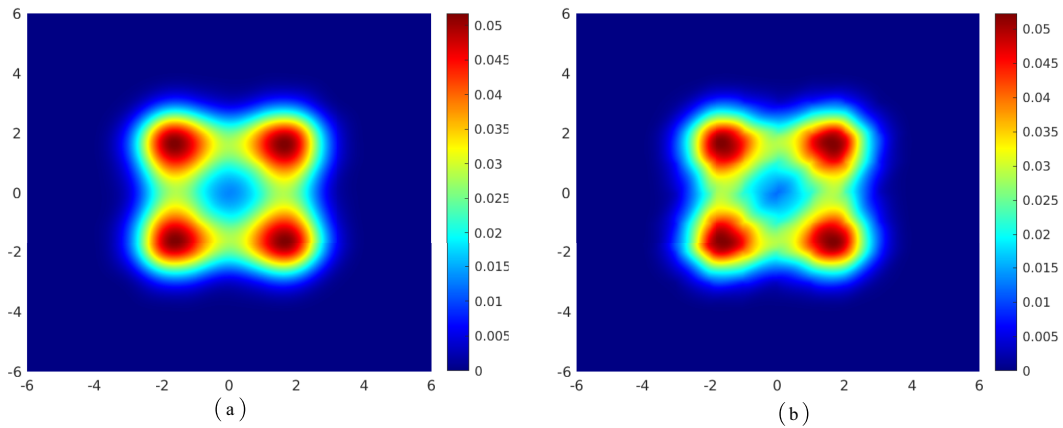


FIGURE 3. Density plot of minimizer, $|u_{\text{gs}}^{\text{LOD}}|^2$. At $H = 1.0$, $l = 3$, the difference in accuracy is more than one order of magnitude. (a) Canonical choice of inner-product, a_c . (b) Problem specific choice of inner-product, a_V .

the construction of the space [48]. To exemplify this we consider the minimization problem

$$u_{\text{gs}} = \arg \min_{v \in \mathbb{S}} \int_{\mathcal{D}} \frac{1}{2} |\nabla v|^2 + V|v|^2 + \frac{\beta}{2} |v|^4 \, dx \quad (35)$$

where again $\mathcal{D} = (-6, 6)^2$ and $\beta = 50$ but with a discontinuous trapping potential given by

$$V = V_0 + V_d, \quad V_0(x, y) = \frac{1}{2}(x^2 + y^2), \quad V_d(x, y) = \mathbb{1}_{x \geq 0}(x, y). \quad (36)$$

Here $\mathbb{1}_{x \geq 0}$ denotes the indicator function on the set $\{(x, y) \in \mathbb{R}^2 : x > 0\}$. Again we approximate the ground state solution u_{gs} of (35), (36) by a discrete minimizer in the LOD-space given by (22) which is then computed with the damped inverse iteration method (25) proposed in Section 4. We construct the LOD-space either with

TABLE 2. Computational times in the offline phase. The basis functions are computed to a relative tolerance of 10^{-7} . In theory, the number of basis functions necessary to compute with inner product a_c is only of $\mathcal{O}(\ell)$ whereas with a_V it is of $\mathcal{O}(H^{-2})$. Memory usage, all LOD-basis functions are listed as represented on the fine mesh. Similarly, all values of ω are listed, in spite of many being equal.

H	ℓ	Precomputations				mem ω
		CPU [s]/basis	ω CPU [s]	mem LOD-matrix	Φ^{LOD}	
2.0	1	4	5	11 MB		11 kB
1.2	2	8	4	82 MB		800 kB
1.0	3	13	7	283 MB		3.4 MB
0.75	4	25	18	580 MB		18 MB
0.6	5	54	46	1.4 GB		93 MB
0.5	6	120	140	2.6 GB		180 MB
0.4	7	170	468	5.5 GB		350 MB
0.3	7	509	729	27.1 GB		1.2 GB

TABLE 3. Energy E from (3) for the ground state obtained for different discretizations using the operator-adapted LOD-space without replacement of $|u_{\text{LOD}}|^2$ by $P_{L^2}^{\text{LOD}}(|u_{\text{LOD}}|^2)$, *i.e.*, corresponding to the method analyzed in [48, 55]. The values for E_{LOD} are taken from Table 1. The fine grid discretization was 50 times finer than the coarse, *i.e.*, $h = H/50$. Tabulated CPU time denotes to the total online time.

H	ℓ	a_V		CPU [s]	N° it
		E	$E - E_{\text{LOD}}$		
2.0	1	7.4459532	0.044897	12	18
1.2	2	7.1305263	0.009331	30	18
1.0	3	7.0913014	0.001523	47	19
0.75	4	7.0844227	0.000080	189	21
0.60	5	7.0825835	0.000028	398	21
0.50	6	7.0824042	-0.000007	682	22
0.40	7	7.0823313	0.000002	1651	22

the canonical bilinear form

$$a_c(u, v) = (\nabla u, \nabla v)$$

from (13) or with

$$a_{V_d}(u, v) = \frac{1}{2}(\nabla u, \nabla v) + (\mathbb{1}_{x \geq 0} u, v)$$

which incorporates the discontinuity of the potential. Due to the symmetry of the discontinuous potential only a handful of basis functions need to be computed, more precisely the $\mathcal{O}(\ell)$ basis functions whose support intersect the line $x = 0$. The rest can be interpolated from these, save for basis functions near the boundary.

Next we compare the LOD-method with a standard P1-FEM approach and with the spectral method implemented in the highly successful and widely used package GPELab [17]. A reference value for the minimal energy is computed to 10 digit accuracy using Richardson extrapolation and was found to be $E^0 = 3.341711792$.

The convergence rates are illustrated in Figure 4a, where downward pointing triangles indicate the canonical LOD-space and upward indicate LOD-space with discontinuity incorporated. The apparent 8th order convergence of the LOD-method is believed to transient and due to the error introduced when replacing the density

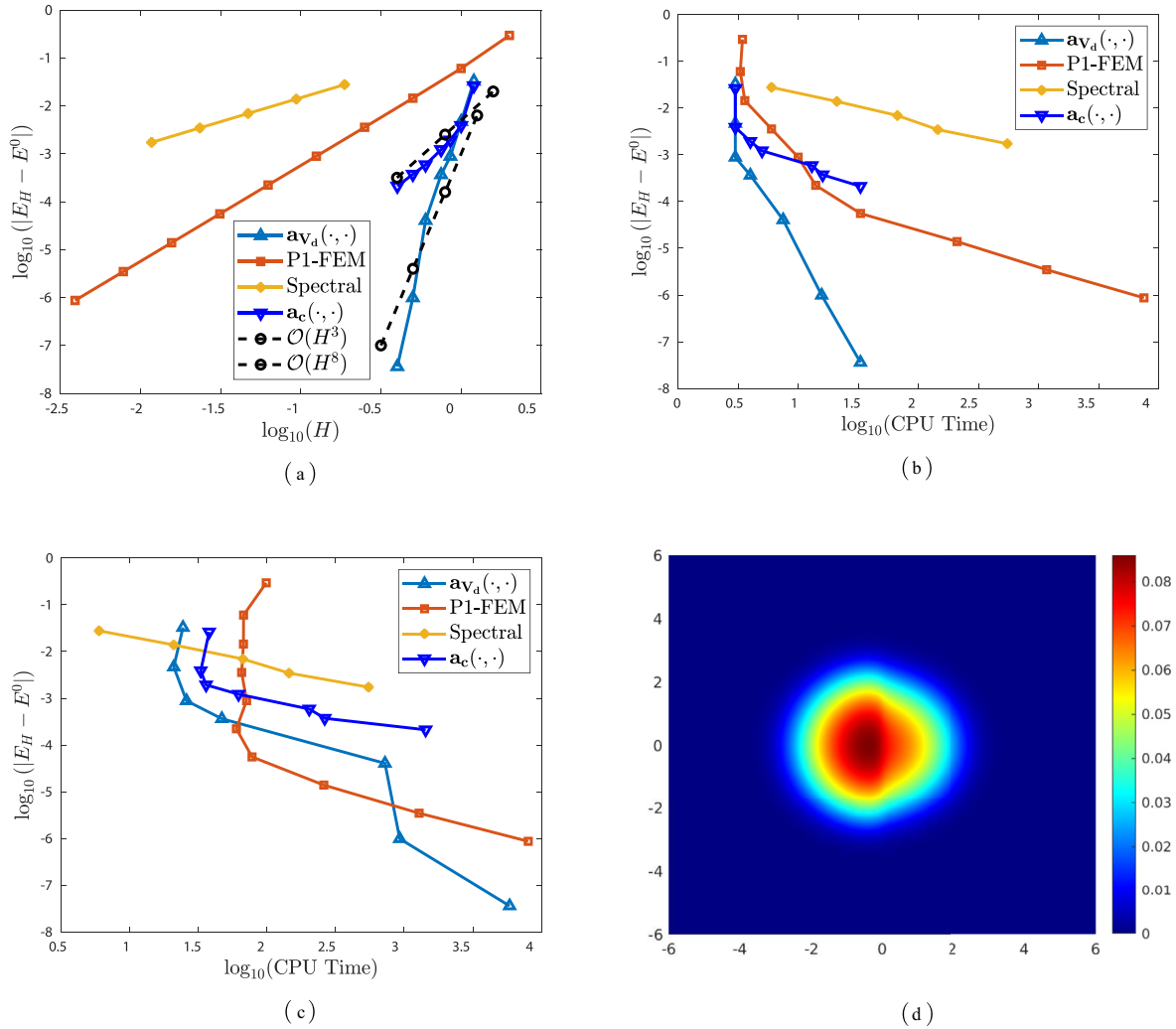


FIGURE 4. Comparison of different methods in terms of convergence and accuracy per CPU times. The blue line with downward pointing triangles in Figures 4a through 4b show the performance of the canonical LOD. The blue line with upwards pointing triangles shows the performance of incorporating the discontinuity in the LOD-space. The choice of h and ℓ is according to the triplets $(H, H/h, \ell)$: $(1.2, 25, 1)$; $(1, 25, 3)$; $(6/7, 25, 4)$; $(3/4, 25, 5)$; $(0.6, 40, 6)$; $(0.5, 40, 6)$; $(0.4, 50, 8)$. (a) Logarithmic convergence plot of error *versus* mesh size H . (b) Logarithmic plot of error *versus* CPU time in the online phase. (c) Log-log plot of error *versus* total CPU time including computation of basis functions and tensor. (d) Density of solution in the LOD-space for $H = 0.4$, $Nh = 50$, $\ell = 8$.

with its projection onto the LOD-space, *i.e.*, (21). In Figure 4b is plotted the accuracy *versus* CPU times in the online phase for the different methods. In this respect the LOD-method outclasses the other methods and we note that in the LOD-space the minimization algorithm converged in around 20 iterations, whereas for the P1 discretization it required around 40. However, due the expensive offline phase, it is not until an accuracy

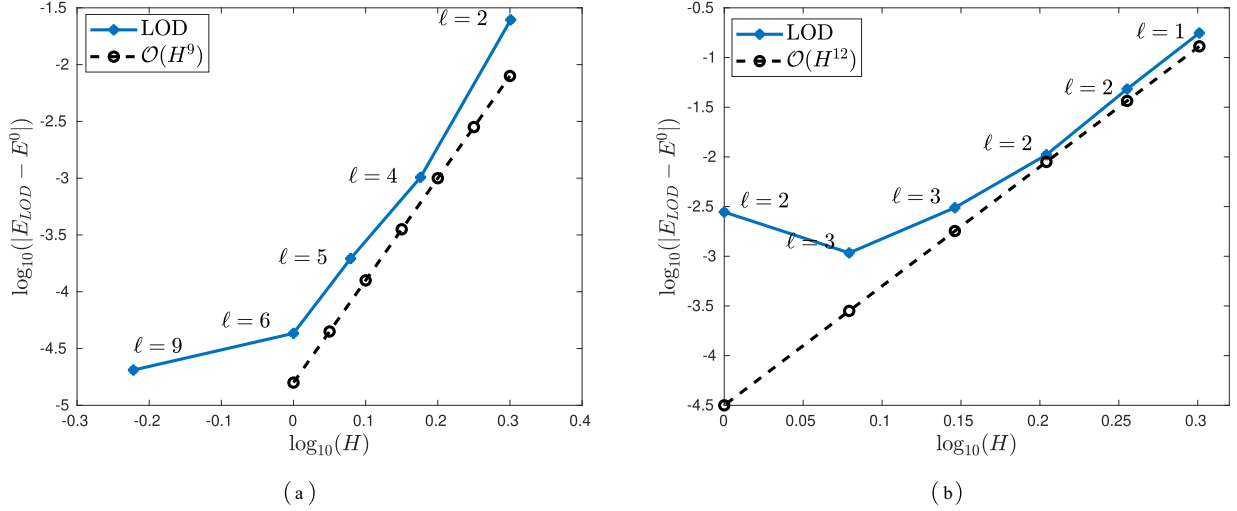


FIGURE 5. Log-log plot of error *versus* coarse mesh size H in the case of harmonic potential. (a) $d = 2$. (b) $d = 3$.

of 10^{-6} , that something is gained from using the current implementation of the LOD-method, as illustrated in Figure 4c where the accuracy is plotted *versus* the total CPU time. Finally, the solution is plotted in Figure 4d.

6.1.3. Harmonic potential

We observe even higher order convergence than the predicted $\mathcal{O}(H^6)$ for purely harmonic potentials. Namely in $2d$ we observe $\mathcal{O}(H^9)$, as illustrated in Figure 5a and in $3d$, $\mathcal{O}(H^{12})$ is observed, see Figure 5b. We stress that the observation only holds for approximately global basis functions. In fact, the convergence is so fast that we are unable to conclusively say that this is the asymptotic behavior rather than some incidental initial convergence. This is mostly due to the severe limitations in the offline phase stemming from the representation of the LOD-basis functions as P1-elements. Specifically we consider the following problem:

$$u_{\text{gs}} = \arg \min_{u \in \mathbb{S}} \int_{\mathcal{D}} \frac{1}{2} |\nabla v|^2 + V|v|^2 + \frac{\beta}{2} |v|^4 \, dx$$

with $V(x) = \frac{1}{2}|x|^2$, $x \in \mathbb{R}^d$ and $\beta = 50$. The domain is $(-6, 6)^2$ in $2d$ and $(-5, 5)^3$ in $3d$. Owing to the exponential decay of the ground state, the solution may be well approximated by the solution to the same problem but posed in the full space $\mathbb{R}^3$. Due to the symmetry, this can be further simplified to a problem in $1d$, namely the following:

$$\min_{\tilde{u} \in H^1(\mathbb{R}_+)} 2^{d-1} \pi \int_0^\infty \left(\frac{1}{2} |\partial_r \tilde{u}|^2 + \frac{1}{2} r^2 |\tilde{u}|^2 + \frac{\beta}{2} |\tilde{u}|^4 \right) r^{d-1} \, dr \quad \text{s.t.} \quad \|\tilde{u}\| = 1,$$

where d denotes the dimension. The corresponding energy can then easily be computed using *e.g.*, a standard FDM such as described in Section 3.4 in [22]. The result, which to 14 digits accuracy is $E^0 = 2.3734292669786$ in $3d$ and $E^0 = 2.896031852200792$ in $2d$, is used as a reference solution.

For brevity, the results are summarized in Figure 5a and in Figure 5b. From these figures it is gathered that in $2d$, $H = 1$, *i.e.*, a grid of 6×6 , achieves an error of 0.000045 . More impressive still is the fact that in $3d$ an error in energy of only $3 \cdot 10^{-3}$ is achieved for $H = 1.4$.

6.2. Computations of dynamics

In this section we present numerical experiments for the time-dependent GPE and the combined GPEVP-GPE problem. For the latter, we consider the GPE in three dimensions and simulate the dynamics of BEC in a harmonic trapping potential with the methods presented in this work. Before doing so, we investigate the performance of the LOD discretization combined with the continuous Galerkin time integrator which results in the cG-LOD-method presented in Section 5. For the performance test of the cG-LOD-method we choose a one dimensional problem from [13] that was used for benchmarking in [54].

6.2.1. Benchmark problem in 1d

The following experiment is only one dimensional, yet it is extremely hard to solve numerically on long time scales. The formal simplicity of the problem also makes it very well suited for benchmarking. The problem is a cubic nonlinear Schrödinger equation and takes the form

$$i\partial_t u = -\partial_{xx} u - 2|u|^2 u \quad \text{in } \mathbb{R} \times (0, T],$$

with initial value

$$u(x, 0) = \frac{8(9e^{-4x} + 16e^{4x}) - 32(4e^{-2x} + 9e^{2x})}{-128 + 4e^{-6x} + 16e^{6x} + 81e^{-2x} + 64e^{2x}}.$$

As derived in [13], the exact solution is given by

$$u(x, t) = \frac{8e^{4it}(9e^{-4x} + 16e^{4x}) - 32e^{16it}(4e^{-2x} + 9e^{2x})}{-128 \cos(12t) + 4e^{-6x} + 16e^{6x} + 81e^{-2x} + 64e^{2x}}$$

and consists of two interacting but standing solitons. What makes it hard to solve numerically is its sensitiveness to energy perturbations. As demonstrated in [54], any initial error as measured in energy, $E - E_h$, will result in a splitting of the two solitons which travel with a velocity proportional to $\sqrt{|E - E_h|}$ in opposite directions. Consequently there is, on long time scales T , a pre-asymptotic regime that is only resolved if the error in energy fulfills $\sqrt{|E - E_h|} \ll T^{-1}$. For example, if the energy converges as $\mathcal{O}(h^2)$ then, refining h by a factor of 2 only decreases the drifting distance between the solitons by a factor of 2.

We consider the fixed spatial discretization $H = 40/2^{11}$, $\ell = 12$ on the computational domain $\mathcal{D} = (-20, 20)$ with homogenous Dirichlet boundary conditions, motivated by the exponential decay of the solution. First we solve the problem on a small time scale, $T = 10$ and compare the cG-LOD scheme from (27) for different choices of q with the standard Crank–Nicolson (CN) time integrator used in [54] which is also based on the LOD discretization in space. The convergence rates for $q = 1, 2, 3$ are plotted in Figure 6a and we obtain numerically the expected convergence rate in time of $\mathcal{O}(\tau^{2q})$. However, for $q = 3$ it is noted that the relative L^2 -error quickly becomes polluted by the spatial discretization. Nevertheless, as expected the higher order schemes ($q = 2$ and $q = 3$) perform enormously better compared to the CN-scheme in terms of the error and fixed step-size. For $q = 1$ the cG-LOD scheme performs roughly as well as the CN-scheme due to the same order of convergence. It is noted that, the larger q is, the smaller becomes the region of stability of the fixed-point iteration that we proposed in Section 5 to solve the nonlinear system of equations required in every time step. An example of this is seen in Figure 6a where a data point is missing for $q = 3$ and $\tau = 10/2^{10} \approx 10^{-2}$.

Figure 6b shows the relative L^2 -error *versus* the CPU time of the cG-LOD scheme for the different choices of q and again compared with the CN-scheme from [54]. When comparing $q = 2$ and $q = 3$ at a fixed computational time of $10^{2.8} \text{ s} \approx 10 \text{ min}$ the difference in relative L^2 -error is of roughly 2 orders of magnitude whereas when comparing $q = 3$ with the CN-scheme we gain more than 3 orders of magnitude in the relative L^2 -error. A higher order q leads to a gain in CPU-time but imposes a smaller τ . Thus, for a given accuracy one should choose q as large as possible under the constraint that the corresponding time-step size is within the region of stability. For our choice of spatial discretization, selecting $q = 3$ is optimal.

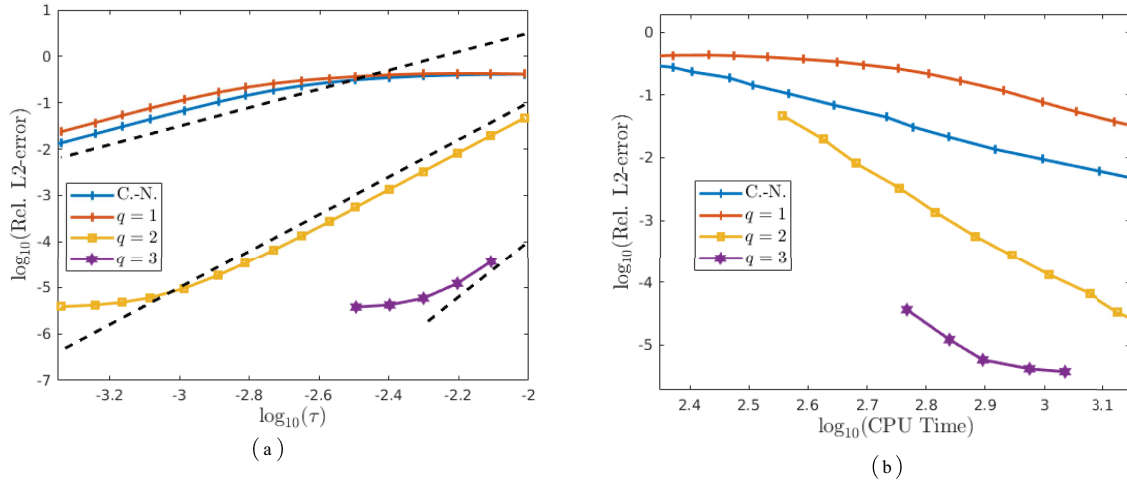


FIGURE 6. Comparison in time of the cG-LOD scheme with $q = 1, 2, 3$ and the Crank–Nicolson scheme. The LOD discretization in space is based on the coarse mesh size $H = 40/2^{11}$ and $\ell = 12$. (a) Convergence plot of rel. L^2 -error versus time-step size τ . (b) Convergence plot of rel. L^2 -error versus CPU-time.

TABLE 4. Error table over varying τ for final time $T = 200$, truncation parameter for the LOD is $\ell = 12$ and the coarse mesh size is $H = 40/2^{11}$. The estimated order of convergence is computed as $\text{EOC}_i = \log(E_{i-1}/E_i)/\log(\tau_{i-1}/\tau_i)$, where E_i is the L^2 -error.

τ	$\ u - u_{\text{LOD}}\ /\ u\ $	EOC	$\ \nabla(u - u_{\text{LOD}})\ /\ \nabla u\ $	CPU [h]
200/22500	0.03841	—	0.01619	3.2
200/23000	0.03401	5.54	0.01436	3.3
200/23500	0.03021	5.51	0.01276	3.3
200/24000	0.02691	5.50	0.01138	3.2
200/ 2^{15}	0.00567	5.00	0.00241	3.9
200/ $(2^{15} \cdot 17/16)$	0.00446	3.96	0.00189	4.1
200/ $(2^{15} \cdot (17/16)^2)$	0.00362	3.44	0.00154	4.2
200/ $(2^{15} \cdot (17/16)^3)$	0.00303	2.93	0.00129	6.1
200/ 2^{16}	0.00175	1.07	0.00075	6.4
200/ 2^{17}	0.00170	0.05	0.00072	11.0

We conclude this one dimensional example with a comparison of previous results from [54]. For that purpose, we now compute the solution on a larger time scale, *i.e.*, we choose $T = 200$. Further, we fix $q = 3$ and hence solve with the 6th order time-stepping method. The relative errors for different time-step sizes (same as in [54]) are tabulated in Table 4. The values demonstrate an initial order of convergence close to 6. However this rate quickly wears off due to the aforementioned spatial pollution. However, compared to the Crank–Nicolson discretization in [54], which otherwise uses the same approach, we find that a mere 3 h computation gives the same relative error as the earlier 100 h computation (which in turn was compared by extrapolation to a hypothetical 4 year computation using a standard $P1$ finite element approach). This significantly improves previous results obtained by Crank–Nicolson time discretizations. Hence, the cG-LOD schemes turns out to be a promising candidate for an efficient solver of the time-dependent GPE in order to simulate dynamics of BECs on long time scales.

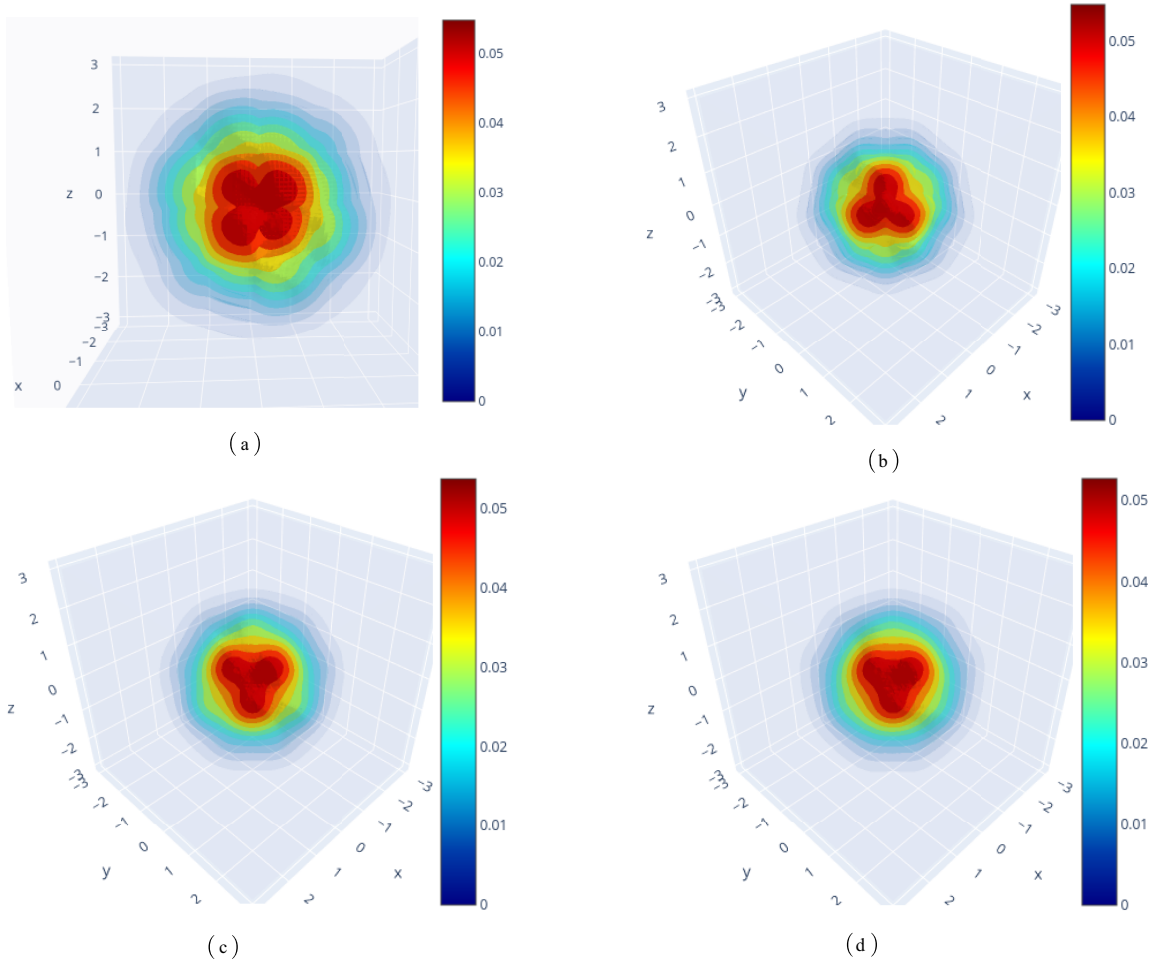


FIGURE 7. Isosurfaces of the density for model problem in Section 6.2.2 at different time instances. (a) $t = 0$. (b) $t = 0$. (c) $t = 0.273$. (d) $t = 9.180$.

6.2.2. Coupled GPEV-GPE problem for 3d dynamics

We conclude this paper with a combined 3-dimensional GPEV-GPE problem on $\mathcal{D} = (-3, 3)^3$, namely:

$$\begin{aligned} i\partial_t u &= -\Delta u + Vu + \beta|u|^2 u && \text{in } \mathcal{D} \times (0, T], \\ u &= 0 && \text{on } \partial\mathcal{D} \times (0, T], \\ u &= u_{\text{gs}} && \text{on } \mathcal{D} \times \{t = 0\}, \end{aligned}$$

where $V(x) = \frac{1}{2}|x|^2$ and the initial value is given by the ground state w.r.t. to a different trapping potential, i.e.,

$$u_{\text{gs}} = \arg \min_{u \in \mathbb{S}} \int_{\mathcal{D}} \frac{1}{2} |\nabla v|^2 + (V + V_d)|v|^2 + \frac{\beta}{2} |v|^4 dx,$$

with $V_d = 2 \bmod(\sum_{i=1}^3 \lfloor x_i \rfloor, 2)$. The coarse mesh size is set to $H = 0.2$, the fine mesh size is $h = H/3$, corresponding to a grid of $90 \times 90 \times 90$. The truncation parameter is chosen to be $\ell = 2$. With this, the LOD-space is computed w.r.t. the inner product defined by $\frac{1}{2}(\nabla u, \nabla v) + (V_d u, v)$. Final computational time is set to

$T = 10$ using a time-step size of $T/2^7 = 0.078125$. We note that in this setting, we were unable to compute a reference solution on the fine mesh. The problem became only computationally traceable in the LOD space, although we mention that the truncation parameter $\ell = 2$ is chosen to be small to keep the computations feasible. However, in Section 7 we discuss further potential improvements of the methods that may serve in future for even better approximations in 3d.

We compute the ground state energy to be 3.354636 and the energy for $t > 0$ to be 2.425574. Some isosurfaces of the density of the ground state are plotted in Figure 7a in which a tetrahedral structure clearly appears. Solving the time-dependent equation, we find that the solution exhibits a periodic behaviour whereby the tetrahedral formation quite abruptly changes orientation. These changes are interlaced with longer periods without such clear structure. This is illustrated in Figure 7a through Figure 7d. Comparing for example Figures 7a–7d we see that this abrupt change takes place in around 0.3 units of time. As illustrated in Figure 7d the density at $t = 9.180$ exhibits the same structure as roughly 9 units of time before.

Finally, we give some notes on the CPU times and memory usage: The minimizer in the LOD-space was computed in 690s and required 14 iterations. The solution at final time $T = 10$ was computed in about 14 hours, corresponding to one time step every 400s. As for the offline phase, the LOD-space was computed in about half an hour and the computation of the tensor ω required 3h. We stress that this was without using any of the periodicity of the discontinuous potential, *i.e.*, many similar basis functions were recomputed and likewise for the entries of ω . The tensor ω required 22GB of memory. The representation of the LOD-basis functions on the fine mesh required a matrix of 1GB.

7. OUTLOOK

We believe the above numerical examples to clearly illustrate the strong capabilities and promising outlook of the proposed method of combining a two level discretization based on the method of LOD with a high order continuous Galerkin time integrator. It achieves high order accuracy and computational efficiency in settings of both low and high regularity. As a final remark, we mention the possibly tremendous improvement of the method by employing a novel construction of the same LOD-space introduced by M. Hauck and D. Peterseim in [42]. This novel construction achieves a much more localized basis. Here we recall that the computational complexity of our method depends heavily on the localization parameter, ℓ , of the LOD-basis functions. For $\ell = 0$, the resulting operators contain neighbor-to-neighbor interactions, *e.g.*, tridiagonal matrices in 1d. For example, for our benchmark problem in 1d, the localization parameter was $\ell = 12$. As shown by Hauck and Peterseim, in 1d, the very same LOD-space is spanned by a set of basis functions for which $\ell = 1$. In addition, they show that in 2d and 3d the decay of the LOD-basis functions can in fact be made to be super exponential, more precisely $\exp(-c\ell^2)$ in 2d and $\exp(-c\ell^{1.5})$ in 3d, as compared to the purely exponential decay of the construction we use. This may well be game changing and shall naturally be considered in future work.

APPENDIX A. NOTES ON THE ASSEMBLY OF THE 3-VALENCE TENSOR ω

The 3-valence tensor introduced in Section 3.4, is extremely sparse. Therefore, some care is required when implementing it. In this section, we briefly describe an efficient digital representation of ω and implementation of pertaining algorithms. First we recall the definition of the 3-valence tensor from (19):

$$\omega_{ijk} := (\varphi_i^{\text{LOD}} \varphi_j^{\text{LOD}}, \varphi_k^{\text{LOD}}), \quad 1 \leq i, j, k \leq N_H.$$

A.1. Compressed sparse row representation of ω

The sparse representation of ω is similar to the CSR format of sparse matrices. Thus it consists of four vectors: $\mathbf{I}_{\text{ptr}}$, $\mathbf{J}_{\omega}$, $\mathbf{K}_{\omega}$ and $\mathbf{V}_{\omega}$. The information about the non-zero elements of ω with first index i is found at indices $\mathbf{I}_{\text{ptr}}[i]$ to $\mathbf{I}_{\text{ptr}}[i+1] - 1$, in the remaining vectors. That is, defining $\text{sub}_i = \mathbf{I}_{\text{ptr}}[i] : \mathbf{I}_{\text{ptr}}[i+1] - 1$, the non-zero values with first index i have second index $\mathbf{J}_{\omega}[\text{sub}_i]$, third index $\mathbf{K}_{\omega}[\text{sub}_i]$ and take values $\mathbf{V}_{\omega}[\text{sub}_i]$. Moreover, the list $\mathbf{J}_{\omega}[\text{sub}_i]$ is sorted. Searching the list $\mathbf{J}_{\omega}[\text{sub}_i]$ for values equal to j , one is left with a subset

of indices indicating elements of ω that have first index i and second index j , let us call this list sub_{ij} . Again, these elements have third index according to the list $K_\omega[\text{sub}_{ij}]$, which is, in turn, sorted. This allows for the quick access to, and manipulation of, entries by means of successive binary searches. For example, consider the following snippet of code which returns the value of ω_{ijk} .

Algorithm 1. Accessing values of ω .

```
function get_val(W,i,j,k)
    sub_i = W.Iptr[i]:W.Iptr[i+1]-1;
    sub_ij = (sub_i[1]-1).+searchsorted(W.J[sub_i],j) #binary search
    sub_ijk = (sub_ij[1]-1).+searchsorted(W.K[sub_ij],k) #binary search
    return W.V[sub_ijk]
end
```

A.2. Preallocating ω

How large should the vectors I_{ptr} , J_ω , K_ω and V_ω be? Here we describe an efficient way of determining their size. First, it is noted that the mass matrix in the LOD-space, M_{LOD} , tells us which pairs of basis functions interact. Here, “interact” means their L^2 inner product is non-zero. Now consider basis function i , which interacts pairwise with other basis functions according to the list of non-zero elements of the column i of M_{LOD} . Let us call this list s and its size n_s . It is now straightforward to find all basis functions such that $\int \varphi_i \varphi_j \varphi_k dx \neq 0$, by looping over the list s .

Algorithm 2. Preallocating ω .

```
INPUT: N #Dimension of LOD-space
       M_LOD #Mass matrix in LOD-space
N_omega_count = 0
W.Iptr = zeros(N+1)
W.Iptr[1] = 1;
for i = 1:N #N dimension of V_LOD
    s = M_LOD[:,i].nzind;
    n_s = length(s)
    start = searchsorted(s,i)[1]
    for j = start:n_s
        for k = j:n_s
            if(abs(M_LOD[s[j],s[k]]))>10^-16) N_omega_count += 1; end
        end
    end
    W.Iptr[i+1] = N_omega_count+1;
end
```

The above code snippet can then be modified to fill in the values of J_ω and K_ω by setting $J[N_omega_count]=s[j]$ and likewise for K_ω , in the inner loop.

A.3. Computing elements of ω

In the spirit of FEM, the values of ω are computed in a loop over the coarse simplices. For each simplex, quadrature points of a sufficiently high order rule are computed. Each such point is located in a fine simplex.

Using this fine simplex' nodes it is straightforward to see, *via* the LOD-matrix, which basis functions have support in it. These are then listed in `active_basis` and their respective values, in the quadrature point multiplied by the appropriate quadrature weight, are stored in `quad_vals`. These values are then added to ω in Algorithm 3.

Algorithm 3. Computing elements of ω .

The following is done for each quadrature point on each coarse simplex.

```

INPUT: active_basis  #list of LOD-basis functions with support on simplex
       n_basis_loc   #length of "active_basis"
       quad_vals     #eval. of "active_basis" in quadrature point on simplex
       W             # CSR lists of w
       w_H          #quadrature weight

```

```

function Add_to_w(quad_vals,w_H,n_basis_loc,active_basis,W)
  for i = 1:n_basis_loc
    i_glob = active_basis[i];
    sub_i = W.Iptr[i_glob]:( W.Iptr[i_glob+1]-1);

    #Evaluate first basis function in quadrature points
    W1 = quad_vals[i]*w_H;
    for j = i:n_basis_loc
      j_glob = active_basis[j]
      sub_ijk = (sub_i[1]-1).+searchsorted(W.J[sub_i],j_glob)

      #Evaluate second basis function in quadrature points
      W2 = W1*quad_vals[j]
      for k = j:n_basis_loc
        k_glob = active_basis[k];
        sub_ijk = (sub_ijk[1]-1).+searchsorted(W.K[sub_ijk],k_glob)[1]
        W3 = W2*quad_vals[k]
        W.V[sub_ijk]+=W3;
      end
    end
  end
end

```

Regarding the choice of quadrature we remark that in the case of the canonical LOD-space in $1d$, a 9th order quadrature rule computes the elements of ω exactly. To see this we let $\Delta^{-1} : L^2((0, 1)) \mapsto H_0^1((0, 1))$ denote the solution operator such that $a_c(\Delta^{-1}f, v) = (f, v) \forall v \in H_0^1((0, 1))$ and let $W = \ker(P_H)$. Consequently, for $v_H \in V_H$, the following holds true $a_c(\Delta^{-1}v_H, w) = (v_H, w) = 0$. The set $\{v \in H_0^1((0, 1)) : v = \Delta^{-1}v_H, \text{ for some } v_H \in V_H\}$ is thus the a -orthogonal complement of the kernel of P_H , in other words it is the LOD-space as defined in equation (10). Lastly, since v''_{LOD} is piecewise linear, any function $v_{\text{LOD}} \in V_{\text{LOD}}$ must be a piecewise polynomial of order 3. This observation is due to Hauck and Peterseim [42]. However, we stress that, in higher dimensions it no longer holds that the canonical basis is a piecewise polynomial of order 3. Inspired by this, we therefore employ, in $2d$, the 19 point quadrature rule of 9th order as described in [37], so as to isolate the 6th order convergence in H . This quadrature rule achieves the optimal number of points required for 9th order convergence and has positive weights in the interior of the simplex. Since we are not aware of any adequate quadrature rule of 9th degree in $3d$, we instead employ a 8th degree tetrahedral quadrature in the $3d$ setting. In this case, the minimal

number of quadrature points is known to be 40 (cf. [76]) which however has not been achieved yet, to the best of our knowledge. We therefore employ the quadrature rule described in [58] requiring 45 quadrature points. Here we remark an error in paper [58], namely a minus sign is missing in Table 1, column W , row $N = 8$, $M = 1$ and the entry should read $-0.393270066412926145d - 1$.

ACKNOWLEDGEMENTS

J. Wärnegård acknowledges the support of the Knut and Alice Wallenberg Foundation KAW 2021.0266.

DEDICATION

This article is dedicated in memory of Professor Assyr Abdulle whose research did not only have huge impact on the field of numerical analysis, but who was also an inspiration to his students and colleagues. Among various other topics, he was especially fascinated by multiscale problems and their numerical treatment, where he made many significant contributions that formed and steered the direction of the research field over the last years [1–3, 7–11]. The second author (PH) had the chance to join his working group at the EPFL Lausanne in 2014 as a postdoctoral researcher and to enjoy his highly motivating and educative mentorship. During that time, the shared interest in numerical multiscale methods lead to three joined works [4–6] on the Localized Orthogonal Decomposition, the same methodology used here for the Gross–Pitaevskii equation, showing how his wide research on multiscale methods also had an influence on this article.

REFERENCES

- [1] A. Abdulle, On a priori error analysis of fully discrete heterogeneous multiscale FEM. *Multiscale Model. Simul.* **4** (2005) 447–459.
- [2] A. Abdulle, The finite element heterogeneous multiscale method: a computational strategy for multiscale PDEs, in Multiple Scales Problems in Biomathematics, Mechanics, Physics and Numerics. Vol. 31 of *GAKUTO Internat. Ser. Math. Sci. Appl.* Gakkotosho, Tokyo (2009) 133–181.
- [3] A. Abdulle and Y. Bai, Reduced basis finite element heterogeneous multiscale method for high-order discretizations of elliptic homogenization problems. *J. Comput. Phys.* **231** (2012) 7014–7036.
- [4] A. Abdulle and P. Henning, A reduced basis localized orthogonal decomposition. *J. Comput. Phys.* **295** (2015) 379–401.
- [5] A. Abdulle and P. Henning, Localized orthogonal decomposition method for the wave equation with a continuum of scales. *Math. Comput.* **86** (2017) 549–587.
- [6] A. Abdulle and P. Henning, Multiscale methods for wave problems in heterogeneous media, in Handbook of Numerical Methods for Hyperbolic Problems. Vol. 18 of *Handb. Numer. Anal.* Elsevier/North-Holland, Amsterdam (2017) 545–576.
- [7] A. Abdulle and T. Pouchon, A priori error analysis of the finite element heterogeneous multiscale method for the wave equation over long time. *SIAM J. Numer. Anal.* **54** (2016) 1507–1534.
- [8] A. Abdulle and C. Schwab, Heterogeneous multiscale FEM for diffusion problems on rough surfaces. *Multiscale Model. Simul.* **3** (2004/2005) 195–220.
- [9] A. Abdulle and E. Weinan, Finite difference heterogeneous multi-scale method for homogenization problems. *J. Comput. Phys.* **191** (2003) 18–39.
- [10] A. Abdulle, E. Weinan, B. Engquist and E. Vanden-Eijnden, The heterogeneous multiscale method. *Acta Numer.* **21** (2012) 1–87.
- [11] A. Abdulle, D. Arjmand and E. Paganoni, Exponential decay of the resonance error in numerical homogenization via parabolic and elliptic cell problems. *C. R. Math. Acad. Sci. Paris* **357** (2019) 545–551.
- [12] 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) 31–53.
- [13] T. Aktosun, T. Busse, F. Demontis and C. van der Mee, Exact solutions to the nonlinear Schrödinger equation, in Topics in Operator Theory. Volume 2. Systems and Mathematical Physics. Vol. 203 of *Oper. Theory Adv. Appl.* Birkhäuser Verlag, Basel (2010) 1–12.
- [14] R. Altmann, P. Henning and D. Peterseim, The J -method for the Gross–Pitaevskii eigenvalue problem. *Numer. Math.* **148** (2021) 575–610.

- [15] R. Altmann, P. Henning and D. Peterseim, Numerical homogenization beyond scale separation. *Acta Numer.* **30** (2021) 1–86.
- [16] R. Altmann, D. Peterseim and T. Stykel, Energy-adaptive Riemannian optimization on the Stiefel manifold. *ESAIM Math. Model. Numer. Anal.* **56** (2022) 1629–1653.
- [17] X. Antoine and R. Duboscq, GPELab, a Matlab toolbox to solve Gross–Pitaevskii equations I: computation of stationary solutions. *Comput. Phys. Commun.* **185** (2014) 2969–2991.
- [18] 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.
- [19] X. Antoine, W. Bao and C. Besse, Computational methods for the dynamics of the nonlinear Schrödinger/Gross–Pitaevskii equations. *Comput. Phys. Commun.* **184** (2013) 2621–2633.
- [20] X. Antoine, A. Levitt and Q. Tang, Efficient spectral computation of the stationary states of rotating Bose–Einstein condensates by preconditioned nonlinear conjugate gradient methods. *J. Comput. Phys.* **343** (2017) 92–109.
- [21] W. Bao and Y. Cai, Uniform error estimates of finite difference methods for the nonlinear Schrödinger equation with wave operator. *SIAM J. Numer. Anal.* **50** (2012) 492–521.
- [22] W. Bao and Y. Cai, Mathematical theory and numerical methods for Bose–Einstein condensation. *Kinet. Relat. Models* **6** (2013) 1–135.
- [23] W. Bao and Q. Du, Computing the ground state solution of Bose–Einstein condensates by a normalized gradient flow. *SIAM J. Sci. Comput.* **25** (2004) 1674–1697.
- [24] W. Bao, S. Jin and P.A. Markowich, Numerical study of time-splitting spectral discretizations of nonlinear Schrödinger equations in the semiclassical regimes. *SIAM J. Sci. Comput.* **25** (2003) 27–64.
- [25] C. Besse, A relaxation scheme for the nonlinear Schrödinger equation. *SIAM J. Numer. Anal.* **42** (2004) 934–952.
- [26] C. Besse, S. Descombes, G. Dujardin and I. Lacroix-Violet, Energy-preserving methods for nonlinear Schrödinger equations. *IMA J. Numer. Anal.* **41** (2021) 618–653.
- [27] J. Bunder and A. Roberts, High-order homogenisation by learning spatial discretisations of PDEs that provably preserve self-adjointness. *Part. Differ. Equ. Appl. Math.* **6** (2022) 1–16.
- [28] E. Cancès, R. Chakir and Y. Maday, Numerical analysis of nonlinear eigenvalue problems. *J. Sci. Comput.* **45** (2010) 90–117.
- [29] T. Cazenave, Semilinear Schrödinger Equations. Vol. 10 of *Courant Lecture Notes in Mathematics*. New York University, Courant Institute of Mathematical Sciences, New York; American Mathematical Society, Providence, RI (2003).
- [30] E.A. Cornell and C.E. Wieman, Nobel lecture: Bose–Einstein condensation in a dilute gas, the first 70 years and some recent experiments. *Rev. Mod. Phys.* **74** (2002) 875–893.
- [31] J. Cui, W. Cai and Y. Wang, A linearly-implicit and conservative Fourier pseudo-spectral method for the 3D Gross–Pitaevskii equation with angular momentum rotation. *Comput. Phys. Commun.* **253** (2020) 107160, 26.
- [32] I. Danaila and P. Kazemi, A new Sobolev gradient method for direct minimization of the Gross–Pitaevskii energy with rotation. *SIAM J. Sci. Comput.* **32** (2010) 2447–2467.
- [33] I. Danaila and B. Protas, Computation of ground states of the Gross–Pitaevskii functional via Riemannian optimization. *SIAM J. Sci. Comput.* **39** (2017) B1102–B1129.
- [34] K. Dekker and J.G. Verwer, Stability of Runge–Kutta Methods for Stiff Nonlinear Differential Equations. Vol. 2 of *CWI Monographs*. North-Holland Publishing Co., Amsterdam (1984).
- [35] C.M. Dion and E. Cancès, Ground state of the time-independent Gross–Pitaevskii equation. *Comput. Phys. Comm.* **177** (2007) 787–798.
- [36] C. Döding and P. Henning, Uniform L^∞ -bounds for energy-conserving higher-order time integrators for the Gross–Pitaevskii equation with rotation. *IMA J. Numer. Anal.* (2023).
- [37] D.A. Dunavant, High degree efficient symmetrical Gaussian quadrature rules for the triangle. *Int. J. Numer. Methods Eng.* **21** (1985) 1129–1148.
- [38] C. Engwer, P. Henning, A. Målqvist and D. Peterseim, Efficient implementation of the localized orthogonal decomposition method. *Comput. Methods Appl. Mech. Eng.* **350** (2019) 123–153.
- [39] L. Erdős, B. Schlein and H.-T. Yau, Derivation of the Gross–Pitaevskii equation for the dynamics of Bose–Einstein condensate. *Ann. of Math.* **172** (2010) 291–370.
- [40] M. Greiner, O. Mandel, T. Esslinger, T.W. Hänsch and I. Bloch, Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms. *Nature* **415** (2002) 39–41.
- [41] E.P. Gross, Structure of a quantized vortex in boson systems. *Nuovo Cimento* **20** (1961) 454–477.
- [42] M. Hauck and D. Peterseim, Super-localization of elliptic multiscale problems. *Math. Comput.* **92** (2023) 981–1003.

- [43] P. Heid, B. Stamm and T.P. Wihler, Gradient flow finite element discretizations with energy-based adaptivity for the Gross–Pitaevskii equation. *J. Comput. Phys.* **436** (2021) 110165.
- [44] F. Hellman, T. Keil and A. Målqvist, Numerical upscaling of perturbed diffusion problems. *SIAM J. Sci. Comput.* **42** (2020) A2014–A2036.
- [45] F. Hellman, A. Målqvist and S. Wang, Numerical upscaling for heterogeneous materials in fractured domains. *ESAIM Math. Model. Numer. Anal.* **55** (2021) S761–S784.
- [46] P. Henning and A. Målqvist, Localized orthogonal decomposition techniques for boundary value problems. *SIAM J. Sci. Comput.* **36** (2014) A1609–A1634.
- [47] P. Henning and A. Målqvist, The finite element method for the time-dependent Gross–Pitaevskii equation with angular momentum rotation. *SIAM J. Numer. Anal.* **55** (2017) 923–952.
- [48] P. Henning and A. Persson, On optimal convergence rates for discrete minimizers of the Gross–Pitaevskii energy in localized orthogonal decomposition spaces. *Multiscale Model. Simul.* **21** (2023) 993–1011.
- [49] P. Henning and D. Peterseim, Oversampling for the multiscale finite element method. *SIAM Multiscale Model. Simul.* **11** (2013) 1149–1175.
- [50] P. Henning and D. Peterseim, Crank–Nicolson Galerkin approximations to nonlinear Schrödinger equations with rough potentials. *Math. Models Methods Appl. Sci.* **27** (2017) 2147–2184.
- [51] P. Henning and D. Peterseim, Sobolev gradient flow for the Gross–Pitaevskii eigenvalue problem: global convergence and computational efficiency. *SIAM J. Numer. Anal.* **58** (2020) 1744–1772.
- [52] P. Henning and J. Wärnegård, Numerical comparison of mass-conservative schemes for the Gross–Pitaevskii equation. *Kinet. Relat. Models* **12** (2019) 1247–1271.
- [53] P. Henning and J. Wärnegård, A note on optimal H^1 -error estimates for Crank–Nicolson approximations to the nonlinear Schrödinger equation. *BIT Numer. Math.* **61** (2021) 37–59.
- [54] P. Henning and J. Wärnegård, Superconvergence of time invariants for the Gross–Pitaevskii equation. *Math. Comput.* **91** (2022) 509–555.
- [55] P. Henning, A. Målqvist and D. Peterseim, Two-level discretization techniques for ground state computations of Bose–Einstein condensates. *SIAM J. Numer. Anal.* **52** (2014) 1525–1550.
- [56] E. Jarlebring, S. Kvaal and W. Michiels, An inverse iteration method for eigenvalue problems with eigenvector nonlinearities. *SIAM J. Sci. Comput.* **36** (2014) A1978–A2001.
- [57] O. Karakashian and C. Makridakis, A space-time finite element method for the nonlinear Schrödinger equation: the continuous Galerkin method. *SIAM J. Numer. Anal.* **36** (1999) 1779–1807.
- [58] P. Keast, Moderate-degree tetrahedral quadrature formulas. *Comput. Methods Appl. Mech. Eng.* **55** (1986) 339–348.
- [59] W. Ketterle, Nobel lecture: when atoms behave as waves: Bose–Einstein condensation and the atom laser. *Rev. Mod. Phys.* **74** (2002) 1131–1151.
- [60] F. Kröppel, R. Maier and D. Peterseim, Operator compression with deep neural networks. *Adv. Contin. Discrete Models* **2022** (2022) 29.
- [61] E. Lieb, R. Seiringer and J. Yngvason, Bosons in a trap: a rigorous derivation of the Gross–Pitaevskii energy functional. *Phys. Rev. A* **61** (2000).
- [62] E.H. Lieb, R. Seiringer and J. Yngvason, A rigorous derivation of the Gross–Pitaevskii energy functional for a two-dimensional Bose gas. *Comm. Math. Phys.* **224** (2001) 17–31. Dedicated to Joel L. Lebowitz.
- [63] P. Ljung, A. Målqvist and A. Persson, A generalized finite element method for the strongly damped wave equation with rapidly varying data. *ESAIM Math. Model. Numer. Anal.* **55** (2021) 1375–1403.
- [64] C. Lubich, On splitting methods for Schrödinger–Poisson and cubic nonlinear Schrödinger equations. *Math. Comput.* **77** (2008) 2141–2153.
- [65] R. Maier, A high-order approach to elliptic multiscale problems with general unstructured coefficients. *SIAM J. Numer. Anal.* **59** (2021) 1067–1089.
- [66] A. Målqvist and D. Peterseim, Localization of elliptic multiscale problems. *Math. Comput.* **83** (2014) 2583–2603.
- [67] A. Målqvist and D. Peterseim, Numerical Homogenization by Localized Orthogonal Decomposition. Vol. 5 of *SIAM Spotlights*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2021).
- [68] A. Målqvist and B. Verfürth, An offline-online strategy for multiscale problems with random defects. *ESAIM Math. Model. Numer. Anal.* **56** (2022) 237–260.
- [69] M.R. Matthews, B.P. Anderson, P.C. Haljan, D.S. Hall, C.E. Wieman and E.A. Cornell, Vortices in a Bose–Einstein condensate. *Phys. Rev. Lett.* **83** (1999) 2498–2501.
- [70] J.M. Melenk and I. Babuška, The partition of unity finite element method: basic theory and applications. *Comput. Methods Appl. Mech. Eng.* **139** (1996) 289–314.

- [71] L.P. Pitaevskii, Vortex lines in an imperfect Bose gas. Number 13. Soviet Physics JETP-USSR (1961).
- [72] L.P. Pitaevskii and S. Stringari, Bose–Einstein Condensation. Oxford University Press, Oxford (2003).
- [73] C. Raman, J.R. Abo-Shaeer, J.M. Vogels, K. Xu and W. Ketterle, Vortex nucleation in a stirred Bose–Einstein condensate. *Phys. Rev. Lett.* **87** (2001) 210402.
- [74] J.M. Sanz-Serna, Methods for the numerical solution of the nonlinear Schrödinger equation. *Math. Comput.* **43** (1984) 21–27.
- [75] J.M. Sanz-Serna and J.G. Verwer, Conservative and nonconservative schemes for the solution of the nonlinear Schrödinger equation. *IMA J. Numer. Anal.* **6** (1986) 25–42.
- [76] P. Solin, K. Segeth and I. Dolezel, Higher-Order Finite Element Methods, 1st edition. Chapman and Hall/CRC (2003).
- [77] M. Thalhammer, Convergence analysis of high-order time-splitting pseudospectral methods for nonlinear Schrödinger equations. *SIAM J. Numer. Anal.* **50** (2012) 3231–3258.
- [78] Y. Tourigny, Optimal H^1 estimates for two time-discrete Galerkin approximations of a nonlinear Schrödinger equation. *IMA J. Numer. Anal.* **11** (1991) 509–523.
- [79] P. Upadhyaya, E. Jarlebring and E.H. Rubensson, A density matrix approach to the convergence of the self-consistent field iteration. *Numer. Algebra Control Optim.* **11** (2021) 99–115.
- [80] J.G. Verwer and J.M. Sanz-Serna, Convergence of method of lines approximations to partial differential equations. *Computing* **33** (1984) 297–313.
- [81] J. Wang, A new error analysis of Crank–Nicolson Galerkin FEMs for a generalized nonlinear Schrödinger equation. *J. Sci. Comput.* **60** (2014) 390–407.
- [82] Z. Wu and Z. Zhang, Convergence analysis of the localized orthogonal decomposition method for the semiclassical Schrödinger equations with multiscale potentials. *J. Sci. Comput.* **93** (2022) 73.
- [83] G.E. Zouraris, On the convergence of a linear two-step finite element method for the nonlinear Schrödinger equation. *M2AN Math. Model. Numer. Anal.* **35** (2001) 389–405.



Please help to maintain this journal in open access!

This journal is currently published in open access under the Subscribe to Open model (S2O). We are thankful to our subscribers and supporters for making it possible to publish this journal in open access in the current year, free of charge for authors and readers.

Check with your library that it subscribes to the journal, or consider making a personal donation to the S2O programme by contacting subscribers@edpsciences.org.

More information, including a list of supporters and financial transparency reports, is available at <https://edpsciences.org/en/subscribe-to-open-s2o>.