
A Meshfree splitting method for soliton dynamics in nonlinear Schrödinger equations

Marco Caliari¹, Alexander Ostermann² and Stefan Rainer^{2*}

¹ Dipartimento di Informatica, Università di Verona, Verona
marco.caliari@univr.it

² Institut für Mathematik, Universität Innsbruck, Innsbruck
{alexander.ostermann,stefan.rainer}@uibk.ac.at

Summary. A new method for the numerical simulation of the so-called soliton dynamics arising in a nonlinear Schrödinger equation in the semi-classical regime is proposed. For the time discretization a classical fourth-order splitting method is used. For the spatial discretization, however, a meshfree method is employed in contrast to the usual choice of (pseudo) spectral methods. This approach allows to keep the degrees of freedom almost constant as the semi-classical parameter ε becomes small. This behavior is confirmed by numerical experiments.

Key words: meshfree discretization, splitting methods, nonlinear Schrödinger equations, soliton dynamics, semi-classical regime

1 Introduction

In this paper, we provide a new numerical method for the solution of the nonlinear Schrödinger equation in the semi-classical regime (see [14])

$$\begin{cases} i\varepsilon\partial_t\phi_\varepsilon = -\frac{\varepsilon^2}{2}\Delta\phi_\varepsilon + V(x)\phi_\varepsilon - |\phi_\varepsilon|^{2p}\phi_\varepsilon, & x \in \mathbb{R}^d, t > 0 \\ \phi_\varepsilon(0, x) = \phi_{\varepsilon,0}(x), \end{cases} \quad (1)$$

where $V(x)$ is a smooth external potential and $0 < p < 2/d$. We are in particular interested in the so-called *soliton dynamics*, in which the shape of the time-dependent solution remains close to the initial value. Moreover such a solution has a small essential support and travels according to Newton's law. By a small essential support of a function we mean that the closure of the set of points for which the magnitude of the function is greater than some prescribed threshold value is small in comparison to the computational

* The work of this author was partially supported by the Tiroler Wissenschaftsfond grant UNI-0404/880.

domain, which can be even unbounded. A similar behavior can be observed for solutions of other important classes of partial differential equations, such as wave equations or transport equations. For nonlinear Schrödinger equations, splitting methods (see, e.g. [3, 4, 6, 10, 17]) have shown to be quite efficient and accurate, also in the conservation of the geometric properties, such as mass and energy.

For the problem we have in mind, the resolution of the spatial discretization has to be high enough to give a good approximation of the solution whose support is concentrated in a small region. This support depends on the semi-classical parameter ε . The smaller this parameter becomes the smaller the support of the solution gets. On the other hand, in order to keep the degrees of freedom at a reasonable level, it is not possible to have the same high resolution everywhere, in particular, along the trajectory of the soliton dynamics. One should rather consider an adaptive mesh, where most of the discretization points are contained in the moving essential support of the solution. Much more flexible, however, are the so-called meshfree (or meshless) methods, whose advantage is the possibility to easily add or delete discretization points. Nowadays there are many ways to construct meshfree approximations, for instance, by moving least squares or radial basis functions (RBFs). Since the essential support of the solution of problem (1) is compact, we choose compactly supported radial basis functions, developed around 1995 (see [18, 19, 21]). The overall goal of the proposed approach is to present a method whose degrees of freedom are independent of ε . In addition we introduce a smart procedure to control the spatial accuracy during time evolution, a feature often missing in the method of lines, even if implemented with an adaptive mesh or a meshless approach.

The outline of the paper is as follows. In Section 2 we give more details of our equation and state some theoretical facts that we use to solve problem (1). In Section 3 we shortly describe the method of RBF discretization. Sections 4 and 5 deal with the numerical solution of problem (1). First, in Section 4 we concentrate on the numerical computation of the initial value $\phi_{\varepsilon,0}(x)$. In Section 5 we explain how the time evolution is carried out. In particular, we give an introduction to splitting methods and show how the involved semiflows can be computed in the context of meshfree approximations. In the final Section 6 we perform some numerical experiments that demonstrate the independence of the degrees of freedom from ε .

2 Soliton dynamics

We are concerned with the numerical solution of nonlinear Schrödinger equations in the semi-classical limit. The problem has the form (1) with $0 < p < 2/d$. The initial value $\phi_{\varepsilon,0}(x)$ is a bump-like function

$$\phi_{\varepsilon,0}(x) = r \left(\frac{x-\bar{x}}{\varepsilon} \right) e^{\frac{i}{\varepsilon} x \cdot \bar{v}}, \quad \bar{x}, \bar{v} \in \mathbb{R}^d. \quad (2)$$

Here $r(x)e^{-i\kappa t}$ is the ground state solution of

$$\begin{cases} i\partial_t\phi = -\frac{1}{2}\Delta\phi - |\phi|^{2p}\phi \\ \|\phi\|_{L^2}^2 = m, \quad m > 0 \text{ given,} \end{cases} \quad (3)$$

that is the solution of the form $\phi(t, x) = u(x)e^{-i\kappa t}$ minimizing the energy functional

$$E(\phi) = E(u) = \frac{1}{2} \int_{\mathbb{R}^d} |\nabla u|^2 dx - \frac{1}{p+1} \int_{\mathbb{R}^d} |u|^{2p+2} dx.$$

In this setting the following assertion holds.

Proposition 1 (see [8, Thm. 1]). *Let $\phi_\varepsilon(t, x)$ be the solution of (1) with initial value (2). Then, there exists a family of shifts $\theta_\varepsilon: \mathbb{R}^+ \rightarrow [0, 2\pi)$ such that, as ε tends to zero,*

$$\left\| \phi_\varepsilon(t, x) - r\left(\frac{x-x(t)}{\varepsilon}\right) \exp\left(\frac{i}{\varepsilon}(x \cdot \dot{x}(t) + \theta_\varepsilon(t))\right) \right\|_{\mathbb{H}_\varepsilon} = \mathcal{O}(\varepsilon),$$

locally uniformly in the variable t .

Here,

$$\|\phi\|_{\mathbb{H}_\varepsilon}^2 = \varepsilon^{2-d} \|\nabla\phi\|_{L^2}^2 + \varepsilon^{-d} \|\phi\|_{L^2}^2,$$

and $x(t)$ is the solution of Newton's law

$$\begin{cases} \ddot{x}(t) = -\nabla V(x(t)), \\ x(0) = \bar{x}, \\ \dot{x}(0) = \bar{v}. \end{cases} \quad (4)$$

This dynamical behavior, where the shape of $|\phi_\varepsilon(t, x)|^2$ remains close to $|\phi_{\varepsilon,0}(x)|^2$, is typically known as *soliton dynamics*. The case where the initial value is a multibump, say

$$\phi_{\varepsilon,0}(x) = \sum_{j=1}^{\ell} r_j \left(\frac{x-\bar{x}_j}{\varepsilon}\right) e^{i\frac{1}{\varepsilon}x \cdot \bar{v}_j}, \quad (5)$$

where $r_j(x)$ minimizes $E(u)$ under the restriction $\|u\|_{L^2}^2 = m_j$, is studied, e.g., in [12, 14].

3 Meshfree approximation

For the numerical representation of a function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ we use the concept of RBF interpolation. Given a set of interpolation points $\Xi = \{\xi_1, \dots, \xi_n\}$ and a radial function

$$\Psi^\alpha(x) = \psi(\alpha\|x\|), \quad \alpha > 0$$

we construct an interpolant

$$p(x) = \sum_{\xi \in \Xi} \lambda_\xi \Psi_\xi^\alpha(x), \quad \Psi_\xi^\alpha(x) = \Psi^\alpha(x - \xi)$$

satisfying

$$p(\xi_i) = f(\xi_i), \quad i = 1, \dots, n.$$

For more details on RBF interpolation, in particular for the computation of the coefficients $[\lambda_\xi]_{\xi \in \Xi}$, we refer to [9, 15, 20].

In order to save memory one aims to minimize the required number of interpolation points, under the condition $|p(x) - f(x)| \leq \text{tol}$ for a given tolerance $\text{tol} > 0$ and all $x \in \mathbb{R}^d$. This can be done using a residual subsampling approach as described in [11]. For the convenience of the reader, a summary of this approach is given in the appendix. Moreover with the help of the shape parameter α we can adapt the form of the basis function in such a way that the error is minimized for a fixed set of interpolation points.

For the practical implementation of the method we take the compactly supported RBF

$$\psi(r) = (1 - r)_+^6 (35r^2 + 18r + 3),$$

also known as Wendland's function $\varphi_{3,2}$. For more details on compactly supported RBFs we refer to [9, 15, 19, 20].

4 Approximation of the initial value

The considered initial value (2) is the product of the ground state of (3) at time $t = 0$ and a phase factor. This ground state, however, does not depend on ε . Hence we can use standard methods to compute it.

4.1 Approximation of the ground state solution

There are several approaches to compute the ground state solution of (3). One possibility consists in directly minimizing the Euler–Lagrange function

$$\mathcal{E}(u, \mu) = E(u) + \mu(\|u\|_{L^2}^2 - m)$$

by a Newton-like method with approximate line-search strategy, as done, for instance, in [13]. Alternatively, it is possible to use the continuous normalized gradient flow (see [2, 14]), i.e. the continuous version of the propagation of the Schrödinger equation (3) along imaginary time $-it$ with projection to the L^2 sphere of radius \sqrt{m} . This requires the solution of the parabolic PDE

$$\begin{cases} \partial_t u = \frac{1}{2} \Delta u + u^{2p+1} + \mu(u)u, & t > 0 \\ u(0, x) = u_0(x), \quad \|u_0\|_{L^2}^2 = m \end{cases} \quad (6)$$

with

$$\mu(u) = \frac{\frac{1}{2} \int_{\mathbb{R}^d} |\nabla u|^2 dx - \int_{\mathbb{R}^d} |u|^{2p+2} dx}{\|u\|_{L^2}^2}.$$

It can be shown that $u(t, x)$ has constant mass m and its energy decreases in time. Moreover, the steady-state solution $r(x) = \lim_{t \rightarrow \infty} u(t, x)$ together with $\kappa = \mu(r(x))$ gives the desired ground state solution $r(x)e^{-i\kappa t}$ of (3), see [14].

In both cases, a spatial discretization of the unknown function is necessary. For this, spectral methods, based on Hermite or Fourier decompositions are quite standard. Hermite functions are to be preferred when an external harmonic (or quasi-harmonic) potential, that is a term of the form $H(x)\phi$ with $H(x) \approx \frac{|x|^2}{2}$ is present in (3), see, for instance, [13]. In our case of the free-potential equation, the use of Fourier modes is suggested. In fact, the ground state solution of (3) exponentially decays to zero and is symmetric (see [14]). This means that the computational domain can be safely truncated to a reasonable size, corresponding to a modest number of Fourier modes to be used. We further note that the computational domain required at this stage needs just to contain the essential support of the ground state solution and not the whole trajectory of the soliton dynamics.

For our method we use the imaginary time method and approximate the solution of the ODE arising from the discretization of (6), namely

$$\begin{cases} w'(t) = Aw(t) + g(w(t)), & t > 0 \\ w(0) = w_0, \end{cases}$$

with an exponential integrator. Here $w(t)$ denotes the vector of Fourier coefficients of $u(t, x)$, A is the diagonal matrix of the eigenvalues of the Laplace operator (divided by two) with respect to the Fourier eigenfunctions and $g(w(t))$ is the truncated discrete Fourier expansion of the nonlinear part of (6). Exponential integrators (see the survey paper [16]) are particularly suited in this situation, because they are explicit and can manage stiff problems without any restriction on the time step size. Moreover, in the case of a diagonal matrix, the computation of matrix exponential-like functions is straightforward. We use the exponential Runge–Kutta method of order two with tableau

$$\begin{array}{c|cc} 0 & & \\ 1 & \varphi_1 & \\ \hline & \varphi_1 - \varphi_2 & \varphi_2 \end{array}$$

and the embedded exponential Euler method as error estimate. This is particularly useful when computing a steady-state solution. In this situation the

time step sizes can be chosen larger and larger when approaching the steady-state. We stop the integration at time $t = \bar{t}$ and take $w(\bar{t})$ as “steady-state” solution if the condition

$$\|Aw(\bar{t}) + g(w(\bar{t}))\| \leq \delta$$

is satisfied for a given tolerance δ . The initial value $u(0, x)$ in (6) can be arbitrary, but it appears reasonable to take a solution with small energy. If we restrict ourselves to the Gaussian family of mass m

$$G_\sigma(x) = \frac{\sqrt{m\sigma^d}}{\sqrt{\pi^d}} e^{-\sigma^2|x|^2/2}, \quad \sigma > 0$$

it is possible to analytically compute the energy $E(G_\sigma)$ and to minimize it, for $d = 1, 2, 3$ (see [14]). The exponential decay σ for which the energy of the corresponding Gaussian function is minimal gives also an indication of the size of the required computational domain.

4.2 Choice of the interpolation points and the shape parameter

From the above construction, we have the ground state r available on a fine grid \mathcal{G} . Next we replace r by its RBF interpolant p on an reasonable small set of interpolation points Ξ . To construct this set, we use a multilevel iteration (see [15]). We select successively interpolation points from \mathcal{G} and add them to Ξ until the accuracy requirement

$$|p(x) - \phi_{\varepsilon,0}(x)| \leq \text{tol}$$

is satisfied for all $x \in \mathcal{G}$. In order to be able to use the resulting set Ξ for the solution of (1), we have to rescale the spatial variable. For $y := \frac{x}{\varepsilon}$, we get

$$\chi(y) = \phi_{\varepsilon,0}(x) = r(y - \bar{y})e^{i\bar{y}v}$$

which can be represented on the set of interpolation points $\Xi_\varepsilon := \varepsilon\Xi$. Note that the radial functions have to be rescaled as well. For arbitrary radial functions this can be done by adjusting the shape parameter α , using a standard one-dimensional minimization procedure. We take the golden section search method starting from an admissible interval $[0, \alpha^*]$.

5 Approximation of the time dependent solution

For the numerical solution of (1), splitting methods are well established (see, e.g., [17, Chap. 3]). In this section we give a short introduction to splitting methods, and we describe how the involved semiflows can be computed in the context of meshfree approximations. Based on this we explain how the proposed numerical method is constructed.

5.1 Splitting methods

The idea behind splitting is to decompose the right-hand side of a differential equation into two parts and to split the problem into two subproblems. The proper combination of their solutions gives an approximation to the solution of the original problem of desired order.

In our case we split equation (1) into

$$i\varepsilon\partial_t\phi_\varepsilon^{[1]}(t, x) = -\frac{\varepsilon^2}{2}\Delta\phi_\varepsilon^{[1]}(t, x) \quad (7a)$$

and

$$i\varepsilon\partial_t\phi_\varepsilon^{[2]}(t, x) = V(x)\phi_\varepsilon^{[2]}(t, x) - |\phi_\varepsilon^{[2]}(t, x)|^{2p}\phi_\varepsilon^{[2]}(t, x). \quad (7b)$$

The linear problem (7a) is usually solved by (pseudo) spectral methods, relying on Hermite or Fourier basis functions (see, e.g., [10, 17]). However, even if the mass of the solution is supposed to be mainly concentrated in small regions, the bump moves according to $x(t)$, the solution of (4). Hence, the computational domain has to be large enough to cover at least this trajectory. This might lead to an unreasonable number of spectral modes, if one requires the solution to have the same accuracy as the initial value. In order to keep the degrees of freedom small, we use a meshfree approximation based on compactly supported radial basis functions instead.

For a real potential V , the modulus of the solution of (7b), $|\phi_\varepsilon^{[2]}|$, stays constant in time. Thus problem (7b) is a linear differential equation with time independent coefficients depending on a parameter $x \in \mathbb{R}^d$, and its solution can be computed exactly.

For the numerical approximation of $\phi_\varepsilon(t_{n+1}, x)$ we have to combine the exact flows of the split problem, namely

$$\begin{aligned} \partial_t\phi_\varepsilon^{[1]} &= \frac{i\varepsilon}{2}\Delta\phi_\varepsilon^{[1]} = L\phi_\varepsilon^{[1]}, & \phi_\varepsilon^{[1]}(t_n, x) &= w_1(x), \\ \partial_t\phi_\varepsilon^{[2]} &= B\phi_\varepsilon^{[2]}, & \phi_\varepsilon^{[2]}(t_n, x) &= w_2(x) \end{aligned}$$

with the multiplication operator $B(x) = V(x) - |w_2(x)|^{2p}$ and appropriately chosen initial values w_1 and w_2 . As time integration we use the fourth-order symmetric Runge–Kutta–Nyström method SRKN₆^b which has the form

$$\prod_{i=1}^7 e^{a_i k L} e^{b_i k B} \phi_\varepsilon(t_n, x).$$

Here $k = t_{n+1} - t_n$ denotes the time step size. The coefficients a_i, b_i are given in Table 1. For more details about the method we refer to [7].

5.2 Computation of the semiflows

Since the multiplication operator B acts pointwise, the numerical computation of $\phi_\varepsilon^{[2]}(t_n + b_i k) = e^{b_i k B(x)} w_2(x)$ is trivial for every $x \in \mathbb{R}^d$.

Table 1. Coefficients of SRKN_6^b , see [7].

i	a_i	b_i
1	0.245298957184271	0.0829844064174052
2	0.604872665711080	0.396309801498368
3	$0.5 - (a_1 + a_2)$	-0.0390563049223486
4	a_3	$1 - 2(b_1 + b_2 + b_3)$
5	a_2	b_3
6	a_1	b_2
7	0	b_1

To approximate $e^{a_i k L} w_1(\cdot)|_{\xi \in \Xi_\varepsilon}$ we use polynomial interpolation at conjugate pairs of Leja points (see [5, 11]). Similar as direct Krylov methods (see [16] and references therein) and truncated Taylor series approximations (see [1]), Newton interpolation is based on successive applications of L to a given vector $v = [v_\xi]_{\xi \in \Xi_\varepsilon}$. In case of meshfree methods this can be realized as follows: we construct an interpolant

$$p(x) = \sum_{\xi \in \Xi_\varepsilon} \lambda_\xi \Psi_\xi^\alpha(x), \quad p(\xi) = v_\xi, \quad \xi \in \Xi_\varepsilon \quad (8)$$

on the given set of interpolation points Ξ_ε . The coefficients $[\lambda_\xi]_{\xi \in \Xi_\varepsilon}$ are computed from the collocation conditions. With A denoting the interpolation matrix $\Psi_\xi^\alpha(\eta)$, $\xi, \eta \in \Xi_\varepsilon$, we approximate $\frac{i\varepsilon}{2} \Delta v(x)$ by

$$Lp(x) = \frac{i\varepsilon}{2} \Delta p(x) = \sum_{\xi \in \Xi_\varepsilon} [A^{-1}v]_\xi \left(\frac{i\varepsilon}{2} \Delta \Psi_\xi^\alpha \right) (x),$$

hence

$$Lp(\cdot)|_{\xi \in \Xi_\varepsilon} = A_L A^{-1} p(\cdot)|_{\xi \in \Xi_\varepsilon} = A_L A^{-1} v$$

with the matrix $A_L = \frac{i\varepsilon}{2} \Delta \Psi_\xi^\alpha(\eta)$, $\xi, \eta \in \Xi_\varepsilon$.

5.3 Time evolution of the solution

The essential support of the solution of (1) for a fixed time becomes smaller with decreasing ε . The trajectory, however, stays close to the solution of (4) on compact time intervals. Since the modulus of the solution does not change much during time evolution we want to use a similar number of interpolation points for any ε and arbitrary time intervals. This will be done in the following way. We start with the set of interpolation points for the ground state which is independent of ε . This set is scaled according to the procedure described

in Section 4.2. This gives an appropriate set of interpolation points for the initial value $\phi_{\varepsilon,0}(x)$. Next we combine the splitting method described before and the concept of a meshfree integrator as in [11] to advance the solution in time.

Starting from an approximation at time t_n we perform a time step and get a set of interpolation points for the current solution. Depending on the step size this set of interpolation points might be too large to represent the actual solution (in order to perform a time step the set of interpolation points has to cover the essential support of the solution for the whole step). To reduce this set we use the concept of residual subsampling to find a reliable set of interpolation points for the actual solution. As described in the appendix, the subsampling procedure requires a set of candidate interpolation points. This set is obtained by propagating the set of interpolation points of the previous time according to (4). So we have found an approximation at time t_{n+1} . This procedure is repeated until we reach the final time.

6 Numerical experiments

In this section we give some numerical experiments that demonstrate the power of the method. Throughout this section we consider equation (1) in two dimensions with $p = 0.2$ and the harmonic potential

$$V(x, y) = \frac{(1.5x)^2 + y^2}{2}.$$

We integrate from $t = 0$ to 2π . The initial value is the bump-like function (2) with $m = 1$, $\bar{x} = [-2.5, -2.5]^T$ and $\bar{v} = [0, 0]^T$. The trajectory for this potential is a periodic Lissajous curve, starting at $[-2.5, -2.5]^T$ and ending at $[2.5, -2.5]^T$. In Figure 1 the time evolutions for $\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-3}$ are shown. The spatial tolerance is $2.5 \cdot 10^{-4}$ and the time step size is $k = 10^{-3}$.

Note that for moderate values like $\varepsilon = 10^{-2}$, standard grid-based methods have no problems, too. However, for decreasing values of ε , the computational cost increases considerably. In fact, the trajectories lie in the same computational domain, but the essential support of the solution becomes smaller. Therefore, the number of grid points, or, equivalently, the number of spectral modes, has to be increased in order to have the same spatial resolution during time evolution. The growth of the degrees of freedom can be dramatic, especially in the three dimensional case.

On the other hand, for the proposed method the number of interpolation points is in practice independent of ε , as can be seen in Figure 2, on the left. For the same problem as described above and a range of ε from 2^{-6} to 2^{-16} , we state the number of interpolation points needed to meet the prescribed spatial tolerance tol .

In the right plot in Figure 2, we see that the shape parameter α of the compactly supported radial basis functions $\Psi^\alpha(x)$ scales as ε^{-1} , as expected from Section 4.2.

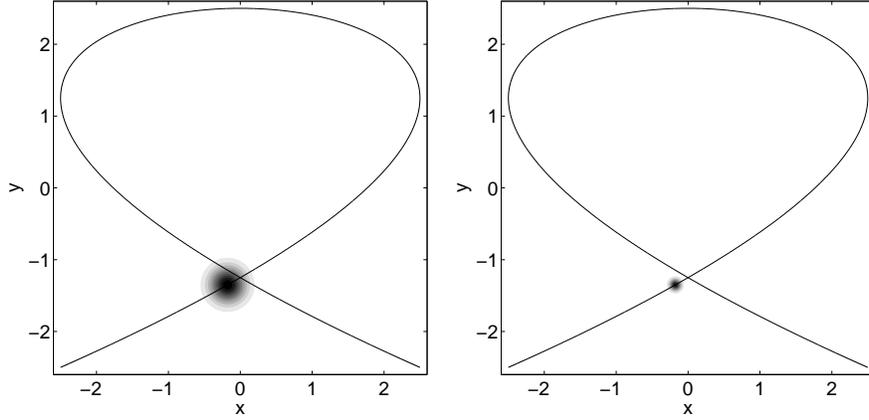


Figure 1. Solution of (1) at $t = 1$ with the whole trajectory of (4) for $\varepsilon = 10^{-2}$ (left) and $\varepsilon = 10^{-3}$ (right) for spatial tolerance $\text{tol} = 2.5 \cdot 10^{-4}$ and time step size $k = 10^{-3}$.

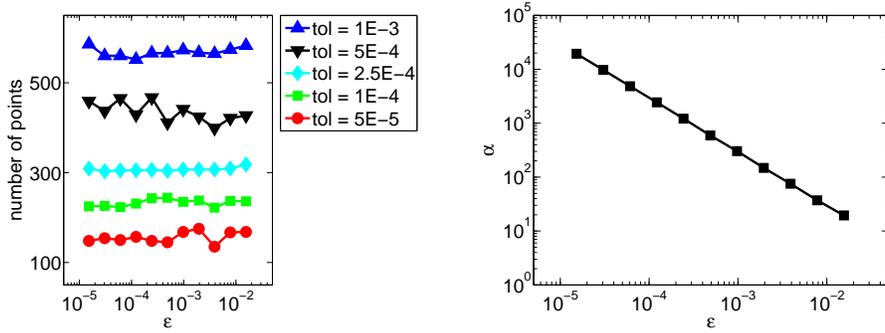


Figure 2. Number of interpolation points for $\varepsilon = 2^{-j}$, $j = 6, \dots, 16$ and various prescribed spatial tolerances tol (left); value of the shape parameter α for different prescribed values of $\varepsilon = 2^{-j}$, $j = 6, \dots, 16$ (right).

Still an open problem is the time step size. The error in time is inverse proportional to ε . This means that a small ε will lead to a large number of time steps. This requires a new time integrator which is work in progress.

Appendix

The idea of residual subsampling is the following. Given a set of candidate interpolation points and a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, find a reasonable set of interpolation points in order that the resulting interpolant p fulfills a certain accuracy requirement. Our aim is to construct an interpolant that satisfies

$$\theta(x) = |f(x) - p(x)| \leq \text{tol}, \quad x \in \mathbb{R}^d. \quad (9)$$

For this purpose we consider in addition a set of check points \mathcal{C} which is used to check (9). Every candidate interpolation point has at least one check point. Using the set of candidate interpolation points we construct an interpolant of f and evaluate it at the set of check points. If the value $\theta(c)$ for some $c \in \mathcal{C}$ is larger than tol we add this check point to the set candidate interpolation points. Else, if $\theta(c)$ is smaller than $0.1 \cdot \text{tol}$ for all check points c corresponding to a candidate interpolation point, we remove this point from the set of candidate interpolation points. If the actual set of candidate interpolation points has changed, we update the set of check points, interpolate again and repeat this procedure until no more points are added or removed. The final set of candidate interpolation points becomes the set of interpolation points.

For the choice of check points there are various possibilities. From (9) it is clear that the denser the set the better we can satisfy the accuracy requirement. On the other hand, more check points results in higher computational costs. Our choice is the following. We compute a Delaunay triangulation of the set of candidate interpolation points and choose the centers of the circumspheres of the resulting simplices as set of check points. For more details we refer to [11].

References

1. A. H. Al-Mohy and N. J. Higham, *Computing the action of the matrix exponential, with an application to exponential integrators*, SIAM J. Sci. Comput., **33** (2011), pp. 488–511.
2. 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), pp. 1674–1697.
3. W. Bao, D. Jaksch and P. Markowich, *Numerical solution of the Gross–Pitaevskii equation for Bose–Einstein condensation*, J. Comp. Phys., **187** (2003), pp. 318–342.
4. W. Bao and J. Shen, *A fourth-order time-splitting Laguerre–Hermite pseudospectral method for Bose–Einstein condensates*, SIAM J. Sci. Comput., **26** (2005), pp. 2010–2028.
5. L. Bergamaschi, M. Caliarì and M. Vianello, *Interpolating discrete advection–diffusion propagators at Leja sequences*, J. Comput. Appl. Math., **172**, pp. 79–99.
6. C. Besse, B. Bidégary and S. Descombes, *Order estimates in time of splitting methods for the nonlinear Schrödinger equation*, SIAM J. Numer. Anal., **40** (2002), pp. 26–40.
7. S. Blanes and P. C. Moan, *Practical symplectic partitioned Runge–Kutta and Runge–Kutta–Nyström methods*, J. Comput. Appl. Math., **142** (2002), pp. 313–330.
8. J. Bronski and R. Jerrard, *Soliton dynamics in a potential*, Math. Res. Letters **7** (2000), pp. 329–342.

9. M. D. Buhmann, *Radial basis functions: theory and implementations*, Cambridge University Press, 2003.
10. M. Caliari, C. Neuhauser and M. Thalhammer, *High-order time-splitting Hermite and Fourier spectral methods for the Gross–Pitaevskii equation*, J. Comput. Phys. **228** (2009), pp. 822–832.
11. M. Caliari, A. Ostermann and S. Rainer, *Meshfree exponential integrators*, (2011), submitted.
12. M. Caliari, A. Ostermann and S. Rainer, *Meshfree integrators*, Oberwolfach Reports **8** (2011), pp. 883–885.
13. M. Caliari, A. Ostermann, S. Rainer and M. Thalhammer, *A minimisation approach for computing the ground state of Gross–Pitaevskii systems*, J. Comput. Phys. **228** (2009), pp. 349–360.
14. M. Caliari and M. Squassina, *Numerical computation of soliton dynamics for NLS equations in a driving potential*, Electron. J. Diff. Eqns. **89** (2010), pp. 1–12.
15. G. E. Fasshauer, *Meshfree approximation methods with MATLAB*, World Scientific Publishing, Hackensack, 2007.
16. M. Hochbruck and A. Ostermann, *Exponential integrators*, Acta Numer., **19** (2010), 209–286.
17. C. Lubich, *From quantum to classical molecular dynamics: reduced models and numerical analysis*, European Mathematical Society, Zürich, 2008.
18. R. Schaback, *Creating surfaces from scattered data using radial basis functions*, in M. Dæhlen, T. Lyche and L. L. Schumaker, eds., *Mathematical methods for curves and surfaces*, Vanderbilt Univ. Press, Nashville, 1995, pp. 477–496.
19. H. Wendland, *Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree*, Adv. Comput. Math., **4** (1995), pp. 389–396.
20. H. Wendland, *Scattered data approximation*, Cambridge University Press, 2005.
21. Z. Wu, *Compactly supported positive definite radial functions*, Adv. Comput. Math., **4** (1995), pp. 283–292.