Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 190 (2003) 1–21

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A fast algorithm for the solution of the time-independent Gross–Pitaevskii equation

Yung-Sze Choi [a], Juha Javanainen [b], Israel Koltracht [a,*], Marijan Koštrun [b,*], Patrick J. McKenna [a], Nataliya Savytska [a]

[a] Department of Mathematics, University of Connecticut, Storrs, CT 06269, USA
[b] Department of Physics, University of Connecticut, Storrs, CT 06269, USA

Received 22 May 2002; received in revised form 27 January 2003; accepted 17 April 2003

Abstract

A new efficient numerical method for the solution of the time-independent Gross–Pitaevskii partial differential equation in three spatial variables is introduced. This equation is converted into an equivalent fixed-point form and is discretized using the collocation method at zeros of Legendre polynomials. Numerical comparisons with a state-of-the-art method based on propagating the solution of the time-dependent Gross–Pitaevskii equation in imaginary time are presented.

Keywords: Partial differential equations; Numerical methods; Non-linear Hamiltonians; Gross–Pitaevskii equation; Ground state

1. Introduction

The subject of this paper is a numerical solution of the time-independent Gross–Pitaevskii equation [9]

$$-\Delta u + V(x,y,z)u + ku^3 = \lambda u, \tag{1}$$

$$u > 0, \quad \lim_{|(x,y,z)|\to\infty} u = 0, \quad \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u(x,y,z)^2\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = 1 \tag{2}$$

written here in dimensionless form, with $\Delta u = u_{xx} + u_{yy} + u_{zz}$. Both the function u and the eigenvalue λ are unknown. The parameter of non-linearity, k, is proportional to the number of atoms in the condensate, and therefore can be very large. $V(x,y,z)$ is a given potential function, which in our numerical experiments is taken to be the a-harmonic potential, $V(x,y,z) = ax^2 + by^2 + cz^2$.

[*] Corresponding authors.
E-mail addresses: israel.koltracht@uconn.edu (I. Koltracht), kostrun@phys.uconn.edu (M. Koštrun).

0021-9991/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0021-9991(03)00235-3

As pointed out in [6,7,15], the lowest lying eigenstate (ground state) of this equation is important in the numerical treatment of the Bose–Einstein condensates in the mean-field approximation. For large values of k and for completely anisotropic potentials, which require a fully three-dimensional formulation, the solution of the Gross–Pitaevskii equation presents serious numerical difficulties.

A standard numerical approach for finding the ground state is to integrate a time-dependent version of (1),

$$\mathrm{i}\frac{\partial u}{\partial t} = -\Delta u + V(x,y,z)u + ku^3, \tag{3}$$

in imaginary time. The ground state wavefunction is then a solution in the limit of large times, while the energy λ is proportional to the norm loss per integration step. A state-of-the-art algorithm of this type, the Dual or Dirac Stepping method, was recently published by two of the co-authors of the present paper, [11]. A summary of this algorithm including the details of the complex time integration is presented in Section 2 and serves as an introduction to the main topic of the paper.

We devote the rest of the paper to a novel direct approach for the time-independent equation. It implements the direct method of [15] and the expansion in the orthogonal polynomial basis, as in a discrete variable approximation (DVR) of [2]. However, here we are able to avoid the biggest shortcoming of the direct methods, that is, we manage to avoid the calculation of large ($10^5 \times 10^5$ or larger) and sparse discrete eigenvalue problem. For the sake of simplicity we outline our approach here in the introduction, for the one-dimensional prototype only. Consider the following boundary value problem:

$$-u''(x) + ax^2u(x) + ku^3(x) = \lambda u(x), \tag{4}$$

$$u > 0, \quad u(\pm T) = 0 \quad \int_{-T}^{T} u^2(x)\,\mathrm{d}x = 1. \tag{5}$$

Its discretization leads to a non-linear system of equations in R^n,

$$Au + k\begin{bmatrix} u_1^3 \\ \vdots \\ u_n^3 \end{bmatrix} = \lambda u. \tag{6}$$

This system of equations is converted into an equivalent fixed-point problem

$$u = S(u) = (cI + A)^{-1} \cdot \left((c+\lambda)\begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} - k\begin{bmatrix} u_1^3 \\ \vdots \\ u_n^3 \end{bmatrix} \right). \tag{7}$$

It is shown in [4] that for the finite differences discretization, and for the appropriate choice of the parameter $c > 0$ and the initial iterate x_0, the fixed-point (FP) iteration $u_{k+1} = S(u_k)$ converges to the positive solution of (6). A simple iteration on λ then produces specific value of λ for which the corresponding u satisfies the normalization condition in (5). In this paper we apply the above fixed-point iteration to discrete equations in R^n obtained via collocation at zeros of Legendre polynomials in three spatial variables. This spectral technique preserves the symmetry present in the differential equation. This symmetry, combined with the separable structure of the a-harmonic potential, allows us to work efficiently with the three-dimensional array representing the solution, u, without stretching it into a linear array. In fact, if N is the number of mesh points per each spatial variable, then it takes $\mathrm{O}(N^4)$ flops (floating point operations) to update the N^3 array. The number of mesh points, N, is modest. In our experiments reported in Section 3, it was enough to use just up to 128 mesh points per variable, for values of k up to 50,000, to achieve accuracy close to double precision. The number of iterations is also not too large (see examples in Section 3), which, combined with the efficiency of each iteration step, leads to a memory efficient and fast algorithm for the solution of the time-independent Gross–Pitaevskii equation in three spatial variables.

The paper is organized as follows. In Section 2 the Dirac or Dual Stepping algorithm (DS-method) of [11] is outlined. In Section 3 we describe in detail the new fixed-point type algorithm. In Section 4 we compare the numerical results of the fixed-point algorithm to the DS-method and Thomas–Fermi approximation. In Appendix A we give in more detail the collocation technique at Legendre points and in Appendix B we explain the Thomas–Fermi approximation.

2. Dual or Dirac Stepping method

The DS-method was suggested by Koštrun and Javanainen [11], as a possible replacement for the split-step class of methods, see [12,17,18] and references therein, for integration of a time-dependent Schrödinger equation or the systems thereof.

The idea is to use a Hamiltonian formulation of the quantum physics and conveniently choose a linear time-independent operator, that plays the role of the operator H_0 in, so-called, interaction picture of quantum physics, [13]. Using this operator, a time domain of the initial value problem is translated to the interval $[0, \Delta t]$, and solved there using an ODE solver. Upon the completion of integration, the solution is translated back to the absolute time. The method was developed for tackling the Gross–Pitaevskii equation and the systems thereof. We describe the method using the more general formulation and notation of [11] which we then modify in order to obtain the solution of the dimensionless Gross–Pitaevskii equation (3).

2.1. Interaction picture

For simplicity, consider a one-dimensional initial value problem, involving a linear time-independent Hamiltonian H_0 acting on wave function $\varphi(t,x)$, together with some, not necessary linear, operator w, that may depend on the wave function $\varphi(t,x)$:

$$\begin{aligned} &\mathrm{i}\frac{\partial\varphi}{\partial t} = H_0\varphi + w[t,x;\varphi(t,x)], \\ &\varphi(0,x) = f(x), x \in [-L,L]. \end{aligned} \tag{8}$$

The Hamiltonian H_0 contains a kinetic energy operator, $-(\hbar^2/2m)\nabla^2$, and possibly a potential energy $V_0(x)$, neither of which explicitly depends on the wave function φ. For notational convenience we do not explicitly denote dependence on x, and write

$$W[t,\varphi] = w[t,x;\varphi(t,x)]. \tag{9}$$

Observe that the domain of the problem has already been truncated to $[-L, L]$ and that appropriate boundary conditions were imposed on the wave function. The interaction picture in quantum mechanics [13] is a linear transformation, which uses operator $u_0(t) = \mathrm{e}^{-\mathrm{i}H_0t}$, to formally eliminate H_0 from Eq. (8) by replacing the wave function $\varphi(t)$, with $\tilde{\varphi}$ using the identity

$$\varphi(t) = \mathrm{e}^{-\mathrm{i}H_0t}\tilde{\varphi}(t) = u_0(t)\tilde{\varphi}(t), \tag{10}$$

so that the equation for the new wave function $\tilde{\varphi}(t,x)$ becomes

$$\mathrm{i}\frac{\partial\tilde{\varphi}}{\partial t} = \mathrm{e}^{\mathrm{i}H_0t}W[t,\mathrm{e}^{-\mathrm{i}H_0t}\tilde{\varphi}]. \tag{11}$$

In quantum physics this transformation is used as the basis of perturbational approach, for weakly perturbed systems, when such makes sense.

2.2. Time translation and integration

Let us first examine the equation of propagation of a wave function over the time interval $[t_k, t_{k+1}]$, for some $k \in N$. For $0 \leqslant \tau \leqslant \Delta t$, we have

$$\mathrm{i}\frac{\partial\tilde{\varphi}(t_k+\tau)}{\partial\tau} = \mathrm{e}^{\mathrm{i}H_0(t_k+\tau)}W\left[t_k+\tau, \mathrm{e}^{-\mathrm{i}H_0(t_k+\tau)}\tilde{\varphi}(t_k+\tau)\right]. \tag{12}$$

We introduce a new wave function $\hat{\varphi}(\tau)$ which serves as a dummy variable [1] for the time integration. It is defined by

$$\hat{\varphi}(\tau) = \mathrm{e}^{-\mathrm{i}H_0t_k}\tilde{\varphi}(t_k+\tau), \tag{13}$$

whose evolution is described by

$$\mathrm{i}\frac{\partial\hat{\varphi}(\tau)}{\partial\tau} = \mathrm{e}^{\mathrm{i}H_0\tau}W\left[t_k+\tau, \mathrm{e}^{-\mathrm{i}H_0\tau}\hat{\varphi}\right], \tag{14}$$

with the initial conditions

$$\hat{\varphi}(0) = \mathrm{e}^{-\mathrm{i}H_0t_k}\tilde{\varphi}(t_k) = \varphi(t_k). \tag{15}$$

The operator W is a contact operator, as described in Eq. (9), so the time integration of Eq. (14) can be performed using a standard ODE integration method [16].

As a result of the integration the array $\hat{\varphi}(\delta t)$ is obtained. It can be shown that its relationship to the function $\varphi(t_\kappa + \delta t) = \varphi(t_{k+1})$ is given by

$$\varphi(t_{k+1}) = \varphi(t_k+\delta t) = \mathrm{e}^{-\mathrm{i}H_0\delta t}\hat{\varphi}(\delta t). \tag{16}$$

Application of Runge–Kutta fourth order integration formulae to Eq. (14) introduces the quantities $\{Y_i'\}_{i=1,\ldots,4}$, defined by

$$\begin{aligned}
Y_1' &= -\mathrm{i}W[t_k, \varphi(t_k)],\\
Y_2' &= -\mathrm{i}W\left[t_k+\frac{\delta t}{2}, u_0\left(\frac{\delta t}{2}\right)\left(\varphi(t_k)+\frac{\delta t}{2}Y_1'\right)\right],\\
Y_3' &= -\mathrm{i}W\left[t_k+\frac{\delta t}{2}, u_0\left(\frac{\delta t}{2}\right)\varphi(t_k)+\frac{\delta t}{2}Y_2'\right],\\
Y_4' &= -\mathrm{i}W\left[t_k+\delta t, u_0\left(\frac{\delta t}{2}\right)\left(u_0\left(\frac{\delta t}{2}\right)\varphi(t_k)+\delta tY_3'\right)\right].
\end{aligned} \tag{17}$$

Using this in Eq. (16) yields the final integration formula

$$\varphi(t_k+\delta t) = u_0(\delta t)\left(\varphi(t_k)+\frac{\delta t}{6}Y_1'\right)+\frac{\delta t}{3}u_0\left(\frac{\delta t}{2}\right)\left(Y_2'+Y_3'\right)+\frac{\delta t}{6}Y_4'. \tag{18}$$

[1] Using this function as the dummy variable renders its indexing with respect to the absolute time $t_j,\ j=1,\ldots,M$ unnecessary. As it follows from the scheme, its values are discarded at the end of the evolution by δt once the wave function $\varphi(t_k+\delta t)$ is calculated.

We identify two standard versions of the DS-integration formula regarding the way the operator $u_0(\tau)$ acts on the wave function:

- *DS2*, where Cayley approximation [10,14] is used in calculation of effects of $u_0(\tau)$ on φ
$$\left(1 + \mathrm{i}\frac{\tau}{2}H_0\right)\varphi(t+\tau) = \left(1 - \mathrm{i}\frac{\tau}{2}H_0\right)\varphi(t), \tag{19}$$
yielding the second order accuracy in time, and
- *DS4*, which consists of taking $H_0 = -\Delta$ in expression for $u_0(\tau)$, and using the Fast Fourier Transform to evaluate the evolution of wave function due to it. This replacement yields the fourth order accuracy in time, and is thus called DS4-formula.

Numerical results regarding the DS-method presented in this manuscript are obtained using the DS2-formula with $H_0 = -\Delta$.

2.3. The Gross–Pitaevskii equation

We present the results for calculation of the lowest eigenvalue of the system described by the Gross–Pitaevskii equation, see e.g. [6],

$$\mathrm{i}\partial_\tau\varphi = (-\Delta + ax^2 + by^2)\varphi + k|\varphi|^2\varphi \tag{20}$$

using the method. The method was modified so that the propagation is done in imaginary instead of in real time. Under those circumstances the wave function loses its norm in each iteration. Renormalization of the solution forces the system to its lowest lying stationery state (if such exists) in the limit of infinite evolution times. The lowest lying eigenvalue λ is given as the limit

$$\lambda = \lim_{\mathrm{i}t\to\infty}\frac{1}{\mathrm{i}\delta t}\left(\frac{1}{|\varphi(\mathrm{i}t + \mathrm{i}\delta t)|} - 1\right). \tag{21}$$

For our calculation the domain was $[-10, 10] \times [-10, 10]$ with a square grid of 129×129 points. We used the finite differences for the operator $u_0 = u_{0,x}u_{0,y}$ with the Cayley expansion. Time integration was performed using the Runge–Kutta fourth order method.

3. Fixed-point iteration

The fixed-point iteration described below is, in principle, applicable to any discretization of the Gross–Pitaevskii equation. In this paper we suggest collocation at zeros of Legendre polynomials. Besides the usual advantages of using a spectral-type method, such as high accuracy at low cost [8], this particular method leads to a symmetric discretization, which becomes critical for two and three spatial variables. The discretization technique is described in some detail in Appendix A. Here it is gradually applied to the case of one, two, and three spatial variables.

3.1. The one variable case

The discretization of the differential equation (4) is described in Appendix A. It leads to a non-linear system of equations in R^n,

$$AV + \frac{k}{T}W^{-2}V^3 = \lambda V. \tag{22}$$

To formulate a fixed-point iteration, we add the matrix cI, where I is an identity matrix of size $N \times N$ and $c > 0$, on both sides of Eq. (22)

$$(A + cI)V = (c + \lambda)V - \frac{k}{T} W^{-2} V^3. \tag{23}$$

The reason for adding a positive multiple of I to the both sides of (22) is that (for the finite differences approximation) with the appropriate choice of $c > 0$ and the initial iterate V_0, the iteration below becomes monotone and is guaranteed to converge. For details, see [4,5].

For the present discretization we do not have a convergence proof as yet, but in all of our numerical experiments the iteration converged for an appropriate choice of c. Next, since A is symmetric, we can decompose it as $A = Q_A D_A Q_A^{\mathrm{T}}$, where D_A is the diagonal matrix of eigenvalues of A, and Q_A is the orthogonal matrix of the corresponding eigenvectors. We now have

$$Q_A(cI + D_A)Q_A^{\mathrm{T}} V = (c + \lambda)V - \frac{k}{T} W^{-2} V^3, \tag{24}$$

or

$$V = Q_A(cI + D_A)^{-1} Q_A^{\mathrm{T}} \left[(c + \lambda)V - \frac{k}{T} W^{-2} V^3 \right], \tag{25}$$

where

$$D_A = \begin{bmatrix} d_1^A & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & d_N^A \end{bmatrix}, \quad (cI + D_A)^{-1} = \begin{bmatrix} \frac{1}{c+d_1^A} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1}{c+d_N^A} \end{bmatrix},$$

and the iteration is performed as follows:

$$V_{\mathrm{next}} = Q_A(cI + D_A)^{-1} Q_A^{\mathrm{T}} \left[(c + \lambda)V_{\mathrm{prev}} - \frac{k}{T} W^{-2} V_{\mathrm{prev}}^3 \right]. \tag{26}$$

We remark that the eigenvalue decomposition of A is not needed in the case of one spatial variable, since one could simply solve the equation

$$(cI + A)V_{\mathrm{next}} = (c + \lambda)V_{\mathrm{prev}} - \frac{k}{T} W^{-2} V_{\mathrm{prev}}^3. \tag{27}$$

This decomposition becomes critical, however, for two and three spatial dimensions and therefore is introduced here as well.

3.2. The two variable case

In two spatial variables, we consider the equation

$$-u_{xx}(x,y) - u_{yy}(x,y) + V(x,y)u(x,y) + ku^3(x,y) = \lambda u(x,y) \tag{28}$$

$$u|_\Gamma = 0, \quad \int_{-T}^{T} \int_{-T}^{T} u^2(x,y)\,\mathrm{d}x\,\mathrm{d}y = 1. \tag{29}$$

Here Γ is the boundary of the square $\{(x,y) | -T \leqslant x, y \leqslant T\}$, and $V(x,y) = ax^2 + by^2$. In this case we approximate the function $u(x,y)$ as follows:

$$u(x,y) = \sum_{i,j=0}^{N-1} \alpha_{ij} p_i(x) p_j(y). \tag{30}$$

We derive the discretization of this equation using the technique described in Appendix A. We assume the discretization mesh that employs the Legendre points in both spatial directions and therefore has the form

$$\{(x_k, y_m) \mid k, m = 1, \dots, N+1\}, \tag{31}$$

where $x_k = y_k$, $k = 1, \dots, N$ are the zeros of $P_N(x)$ multiplied by T.

Let

$$u(x_k, y_m) = \sum_{k,m=0}^{N-1} \alpha_{ij} p_i(x_k) p_j(y_m) \tag{32}$$

and let $U = [u_{km}]_{k,m=1}^{N+1}$ be the matrix of the values of the function u at the Legendre points: $u_{km} = u(x_k, y_m)$. If $\mathscr{A} = [\alpha_{ij}]_{i,j=0}^{N-1}$, then

$$U = P\mathscr{A}P^{\mathrm{T}}, \tag{33}$$

where P is the same as in the one-dimensional case. It follows from (A.14) that

$$\mathscr{A} = P^{\mathrm{T}} W^2 U W^2 P, \tag{34}$$

where W is the matrix of Legendre weights. Following the same approach as in the one-dimensional case, we obtain the following discretization of Eq. (28)

$$-\frac{1}{T^2}(PDP^{\mathrm{T}}W^2)U - U\left(\frac{1}{T^2}PDP^{\mathrm{T}}W^2\right)^{\mathrm{T}} + a\begin{bmatrix} x_1^2 & & 0 \\ & \ddots & \\ 0 & & x_N^2 \end{bmatrix} U + bU \begin{bmatrix} y_1^2 & & 0 \\ & \ddots & \\ 0 & & y_N^2 \end{bmatrix} + kU^3 = \lambda U. \tag{35}$$

Here $-(1/T^2)(PDP^{\mathrm{T}}W^2)U$ represents the discretization of u_{xx}, and is identical to the one variable case for each fixed y. The second term corresponds to u_{yy}.

Multiplying (35) by $\sqrt{T}W$ on both sides and introducing the notations

$$V = T\begin{bmatrix} \omega_1 & & 0 \\ & \ddots & \\ 0 & & \omega_N \end{bmatrix} U \begin{bmatrix} \omega_1 & & 0 \\ & \ddots & \\ 0 & & \omega_N \end{bmatrix} = TWUW, \tag{36}$$

$$A = -\frac{1}{T^2}(WPDP^{\mathrm{T}}W) + a\begin{bmatrix} x_1^2 & & 0 \\ & \ddots & \\ 0 & & x_N^2 \end{bmatrix} \tag{37}$$

and

$$B = -\frac{1}{T^2}(WPDP^{\mathrm{T}}W) + b\begin{bmatrix} y_1^2 & & 0 \\ & \ddots & \\ 0 & & y_N^2 \end{bmatrix}, \tag{38}$$

we get

$$AV + VB + \frac{k}{T^2} W^{-2} V^3 W^{-2} = \lambda V, \tag{39}$$

where both A and B are symmetric matrices. As before, the monotone fixed-point iteration is obtained by adding a multiple of the matrix V, cV $(c > 0)$ to both sides of (39) and by the orthogonal decomposition of matrices A and B, $A = Q_A D_A Q_A^{\mathrm{T}}$, $B = Q_B D_B Q_B^{\mathrm{T}}$. Here D_A, D_B are the diagonal matrices of eigenvalues of A and B correspondingly, and Q_A,Q_B are the orthogonal matrices of the corresponding eigenvectors. Thus

$$cV + Q_A D_A Q_A^{\mathrm{T}} V + V Q_B D_B Q_B^{\mathrm{T}} = (c + \lambda) V - \frac{k}{T^2} W^{-2} V^3 W^{-2}. \tag{40}$$

Multiplying this equation by Q_A^{T} on the left and by Q_B on the right, we obtain

$$c\underline{Q_A^{\mathrm{T}} V Q_B} + D_A \underline{Q_A^{\mathrm{T}} V Q_B} + \underline{Q_A^{\mathrm{T}} V Q_B} D_B = Q_A^{\mathrm{T}} \left[(c + \lambda) V - \frac{k}{T^2} W^{-2} V^3 W^{-2} \right] Q_B. \tag{41}$$

Letting $\widetilde{V} = Q_A^{\mathrm{T}} V Q_B$, we can write (41) as

$$c\widetilde{V} + D_A \widetilde{V} + \widetilde{V} D_B = Q_A^{\mathrm{T}} \left((c + \lambda) V - \frac{k}{T^2} W^{-2} V^3 W^{-2} \right) Q_B. \tag{42}$$

If the entries of the matrix $\widetilde{V}$ are denoted by $\widetilde{v}_{ij}$ and

$$D_A = \begin{bmatrix} d_1^A & & 0 \\ & \ddots & \\ 0 & & d_N^A \end{bmatrix}, \quad D_B = \begin{bmatrix} d_1^B & & 0 \\ & \ddots & \\ 0 & & d_N^B \end{bmatrix}, \tag{43}$$

then for each $i, j = 1, \ldots, N$,

$$c\widetilde{v}_{ij} + d_i^A \widetilde{v}_{ij} + d_j^B \widetilde{v}_{ij} = \left[Q_A^{\mathrm{T}} \left((c + \lambda) V - \frac{k}{T^2} W^{-2} V^3 W^{-2} \right) Q_B \right]_{ij} \tag{44}$$

or

$$\widetilde{v}_{ij} = \left(\frac{1}{c + d_i^A + d_j^B} \right) \left[Q_A^{\mathrm{T}} \left((c + \lambda) V - \frac{k}{T^2} W^{-2} V^3 W^{-2} \right) Q_B \right]_{ij}. \tag{45}$$

In the matrix form we write

$$\widetilde{V} = D_{AB} \bigcirc \left[Q_A^{\mathrm{T}} \left((c + \lambda) V - \frac{k}{T^2} W^{-2} V^3 W^{-2} \right) Q_B \right]. \tag{46}$$

In this formula $[D_{AB}]_{ij} = 1/(c + d_i^A + d_j^B)$ and $\bigcirc$ denotes the componentwise or Schur product of the matrices. Therefore in the two-dimensional case the fixed-point iteration has the form

$$V_{\mathrm{next}} = Q_A \left[D_{AB} \bigcirc \left(Q_A^{\mathrm{T}} \left((c + \lambda) V_{\mathrm{prev}} - \frac{k}{T^2} W^{-2} V_{\mathrm{prev}}^3 W^{-2} \right) Q_B \right) \right] Q_B^{\mathrm{T}}. \tag{47}$$

3.3. The three variable case

Finally, let us explain how this technique works in the three-dimensional case, in the simplest case of a spherically symmetric potential V (so that all the scales are equal, $T_x = T_y = T_z = T$),

$$-u_{xx} - u_{yy} - u_{zz} + (ax^2 + by^2 + cz^2)u(x,y,z) + ku^3(x,y,z) = \lambda u(x,y,z), \tag{48}$$

$$u|_{\Gamma} = 0, \quad \int_{-T}^{\mathrm{T}} \int_{-T}^{T} \int_{-T}^{T} u^2(x,y,z)\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = 1, \tag{49}$$

where Γ is the boundary of the cube $-T \leqslant x,y,z \leqslant T$.

Suppose that

$$u(x,y,z) = \sum_{i,j,k=0}^{N-1} \alpha_{ijk} p_i(x) p_j(y) p_k(z). \tag{50}$$

and let

$$U = [u_{ijk}]_{i,j,k=1}^{N}, \quad u_{ijk} = u(x_i, y_j, z_k), \tag{51}$$

where the meshpoints, $x_i, y_j, z_k,\ i,j,k = 1,\ldots,N, x_i = y_i = z_i$, are the zeros of the Nth Legendre polynomial, $P_N(x)$, multiplied by the factor T. Since U is now a three-dimensional array of the size $N \times N \times N$, we need to define the action of an arbitrary matrix A of size $N \times N$ on U. By $A_x(U)$ we denote the three-dimensional array given for $j,k = 1,\ldots,N$ by

$$\begin{bmatrix} A_x(U)_{1,j,k} \\ \vdots \\ A_x(U)_{N,j,k} \end{bmatrix} = A \begin{bmatrix} u_{1,j,k} \\ \vdots \\ u_{N,j,k} \end{bmatrix}, \tag{52}$$

where the subscript x stands for the direction of this action, i.e., in this case a matrix A is applied in the direction of the x-variable. The action of the matrix A in two other directions, $A_y(U)$ and $A_z(U)$, is defined similarly. Notice that this operation is commutative, meaning that if B is another $N \times N$ matrix, then

$$A_{\bigstar}(B_{\bar{\bigstar}}(U)) = B_{\bar{\bigstar}}(A_{\bigstar}(U)). \tag{53}$$

Here $\bigstar$ and $\bar{\bigstar}$ stand for any choice of the directions x, y or z. This is just the discrete analog of the fact that a mixed partial derivative of a smooth function does not depend on the ordering of variables in which partial derivatives are taken. For the notational simplicity from now on we omit the repeated use of parentheses. After the discretization of Eq. (48) we have

$$-\frac{1}{T^2}[PDP^{\mathrm{T}}W^2]_x(U) - \frac{1}{T^2}[PDP^{\mathrm{T}}W^2]_y(U) - \frac{1}{T^2}[PDP^{\mathrm{T}}W^2]_z(U) + aW_x(U) + bW_y(U) + cW_z(U) + kU^3 = \lambda U. \tag{54}$$

Here again, $(1/T^2)[PDP^{\mathrm{T}}W^2]_x(U)$ represents the discretization of u_{xx} and similarly for u_{yy} and u_{zz}. To symmetrize this equation we apply $\sqrt{T}W_z(\sqrt{T}W_y(\sqrt{T}W_x(\cdot)))$ to Eq. (54) and use the substitution $V = T^{3/2}W_zW_yW_x(U)$ to obtain

$$A_x(V) + B_y(V) + C_z(V) + \frac{k}{T^3}W_z^{-2}W_y^{-2}W_x^{-2}(V^3) = \lambda V. \tag{55}$$

In this formula, the matrices A and B are the same as in the two-dimensional case and the matrix C is

$$C = -\frac{1}{T^2}(WPDP^{\mathrm{T}}W) + c\begin{bmatrix} z_1^2 & & 0 \\ & \ddots & \\ 0 & & z_N^2 \end{bmatrix}. \tag{56}$$

As before, we add a multiple of V, cV $(c > 0)$ to both sides of Eq. (55)

$$cV + A_x(V) + B_y(V) + C_z(V) = (c+\lambda)V - \frac{k}{T^3}W_z^{-2}W_y^{-2}W_x^{-2}(V). \tag{57}$$

With $A = Q_AD_AQ_A^{\mathrm{T}}$, $B = Q_BD_BQ_B^{\mathrm{T}}$ and $C = Q_CD_CQ_C^{\mathrm{T}}$ being the orthogonal decompositions of the matrices A, B and C respectively, we can rewrite the above equation as follows,

$$cV + \left(Q_AD_AQ_A^{\mathrm{T}}\right)_x(V) + \left(Q_BD_BQ_B^{\mathrm{T}}\right)_y(V) + \left(Q_CD_CQ_C^{\mathrm{T}}\right)_z(V) = (c+\lambda)V - \frac{k}{T^3}W_z^{-2}W_y^{-2}W_x^{-2}(V) \tag{58}$$

or

$$cV + Q_{Ax}D_{Ax}Q_{Ax}^{\mathrm{T}}(V) + Q_{By}D_{By}Q_{By}^{\mathrm{T}}(V) + Q_{Cz}D_{Cz}Q_{Cz}^{\mathrm{T}}(V) = (c+\lambda)V - \frac{k}{T^3}W_z^{-2}W_y^{-2}W_x^{-2}(V). \tag{59}$$

We apply now consecutively the matrix Q_A^{T} in x-direction, Q_B^{T} in the y-direction, Q_C^{T} in the z-direction to (59) and introduce the notation $\widetilde{V} = Q_{Cz}^{\mathrm{T}}Q_{By}^{\mathrm{T}}Q_{Ax}^{\mathrm{T}}(V)$. Thus

$$c\widetilde{V} + D_{Ax}\widetilde{V} + D_{By}\widetilde{V} + D_{Cz}\widetilde{V} = Q_{Cz}^{\mathrm{T}}Q_{By}^{\mathrm{T}}Q_{Ax}^{\mathrm{T}}\left((c+\lambda)V - \frac{k}{T^3}W_z^{-2}W_y^{-2}W_x^{-2}(V)\right). \tag{60}$$

Comparing (i,j,k) entries on both sides we find that

$$\left(c + d_i^A + d_j^B + d_k^C\right)\widetilde{V}_{ijk} = \left[Q_{Cz}^{\mathrm{T}}Q_{By}^{\mathrm{T}}Q_{Ax}^{\mathrm{T}}\left((c+\lambda)V - \frac{k}{T^3}W_z^{-2}W_y^{-2}W_x^{-2}(V)\right)\right]_{ijk}. \tag{61}$$

Hence, in three spatial variables the fixed-point iteration can be written as

$$V_{\text{next}} = Q_{Ax}Q_{By}Q_{Cz}\left(\mathscr{D}\bigcirc\left[Q_{Cz}^{\mathrm{T}}Q_{By}^{\mathrm{T}}Q_{Ax}^{\mathrm{T}}\left((c+\lambda)V_{\text{prev}} - \frac{k}{T^3}W_z^{-2}W_y^{-2}W_x^{-2}(V_{\text{prev}})\right)\right]\right), \tag{62}$$

where $\bigcirc$ is the componentwise product and where

$$[\mathscr{D}]_{ijk} = \frac{1}{c + d_i^A + d_j^B + d_k^C}.$$

Despite the fact that the formula given above has a complicated structure, actual programming of the three-dimensional iteration is quite similar to the programming of the two-dimensional model due to the eigenvalue decomposition technique used.

The fact that A, B, and C are symmetric is very important here. It enables us to perform the the iteration (62) in $\mathrm{O}(N^4)$ floating point operations. Here N^4 is the number of flops needed to multiply an $N \times N$ matrix by N^2 columns of an $N \times N \times N$ array. Since in particular computing environments the cost of a matrix-vector multiplication is often less than N^2, the cost of performing the iteration (62) is reduced accordingly. This is to be compared with the cost of solving (57) directly by stretching V into a linear array of length N^3 and using Gaussian elimination. The cost would become a staggering $\mathrm{O}(N^9)$ flops, unless some use is made of the special structure of the $N^3 \times N^3$ matrix representing the sum of linear transformations A_x, B_y and C_z.

Algorithm 1 summarizes our previous discussion:

- Legendre points are found as the eigenvalues of the matrix (A.8) and then translated onto the interval $[-T, T]$.
- The matrix P consisting of the values of the first N normalized Legendre polynomials at the Legendre points is found. For this the recursive formula (A.5) is used.
- The matrix of The Gauss–Legendre weights is obtained using the relation $(W^{-1})^2 = Pp^{\mathrm{T}}$.
- Formulas (A.17)–(A.27) are used to calculate matrix D that transforms the coefficients $\alpha_0, \dots, \alpha_N$ of the function $U(X)$ into the coefficients of its second derivative.
- Matrices A, B and C are constructed as described in (37), (38) and (56).
- Orthogonal decomposition of the matrices A, B and $C : A = Q_A D_A Q_A^{\mathrm{T}}$, $B = Q_B D_B Q_B^{\mathrm{T}}$ and $C = Q_C D_C Q_C^{\mathrm{T}}$ is obtained.
- The iteration (62) is set up.
- The solution (and its norm) is found.

For a given value of λ and sufficiently large choice of the constant $c > 0$, this algorithm generates a sequence of three-dimensional arrays that converges to the solution of the eigenproblem (48).

Note that larger values of k require larger values of the constant c for the algorithm to converge. For example, in two-dimensional situation for $k = 10$ and $V(x, y) = ax^2 + by^2$ the algorithm converges if we take $c = 10$. But for $k = 100$ the algorithm diverges if $c = 10$. We need to increase the value of c to at least $c = 30$ to make the procedure work. Increasing c, however, affects the rate of convergence of the algorithm. For example, in two-dimensional case for $k = 1$ and $V(x, y) = 0.2x^2 + 0.7y^2$, it takes about 60 iterations to obtain a single precision result with $c = 0.1$. The number of iterations approximately doubles for $c = 1$. The same result is obtained in about 700 and 6000 iterations if $c = 10$ and $c = 100$, respectively. Typical graphs of solutions are shown in Figs. 1 and 2. Fig. 3 gives some insight into why the choice of Legendre mesh points is so effective. One can see that these points are clustered towards the boundaries where the function has steepest slopes, and there are far fewer points in the middle, where the function is more flat. Therefore a relatively small number of points is needed for the discretization. To achieve a double precision accuracy, we typically use 32–128 points depending on the size of the domain. While advantageous at zero temperature, the choice of Legendre mesh may not be the most appropriate at non-zero temperatures. There, as pointed out by [1], the presence of thermal atoms lengthens the tail at the edge of the condensate [3].

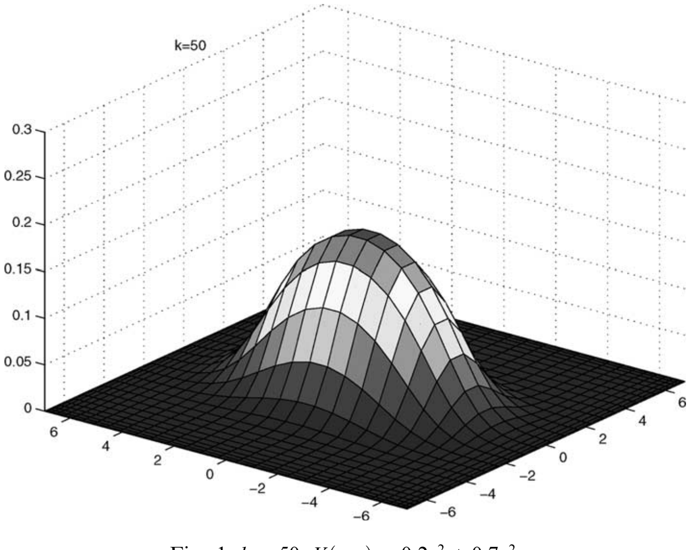

Fig. 1. $k = 50$, $V(x, y) = 0.2x^2 + 0.7y^2$.

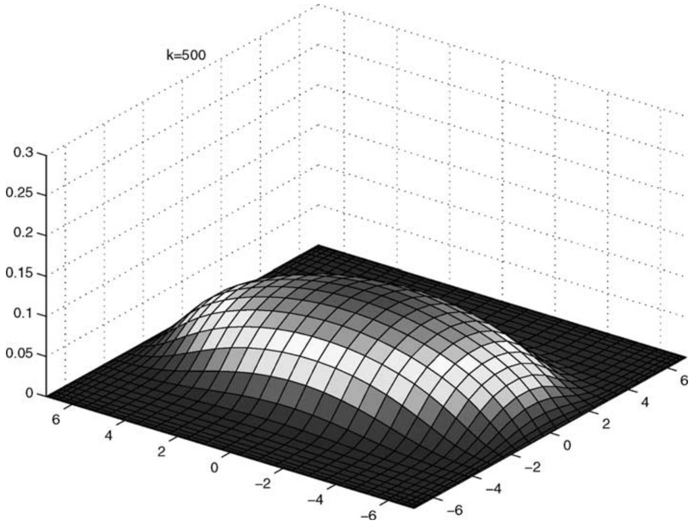

Fig. 2. $k = 500$, $V(x,y) = 0.2x^2 + 0.7y^2$.

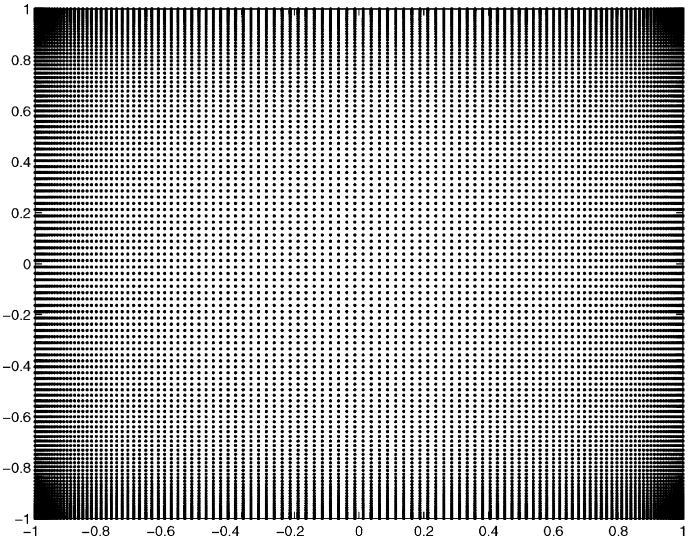

Fig. 3. Distribution of Legendre points in a square domain.

So far we have described how to compute the positive eigenfunction corresponding to a given eigenvalue λ. Our goal, however, is to obtain a positive solution of a particular magnitude. Thus we seek $\lambda^\star > 0$ such that the corresponding solution, $u(\lambda^\star)$, is of the unit norm

$$\|u(\lambda^\star)\|^2 = \int_{-T}^{T}\int_{-T}^{T}\int_{-T}^{T} u^2(x,y,z)\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = 1. \tag{63}$$

Our numerical experiments show that norm of solution $\|u(\lambda)\|$ is a monotone function of λ: if $\lambda_1 \leqslant \lambda_2$, then $\|u(\lambda_1)\| \leqslant \|u(\lambda_2)\|$. To approximate the value of $\lambda^\star$ that corresponds to the solution of the unitary

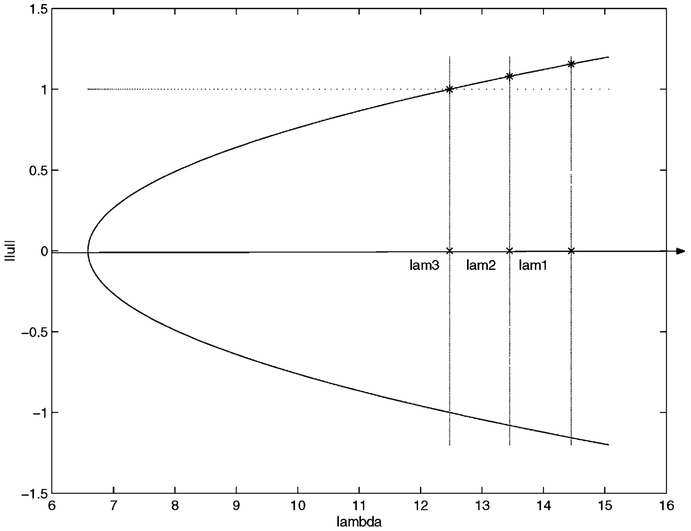

Fig. 4. Normalization strategy.

norm, we apply the following normalization strategy. Using the Thomas–Fermi estimates (see Appendix B), we find initial values of λ_1 and λ_2 $(\lambda_1 > \lambda_2)$, large enough so that both $h_1 = \|u(\lambda_1)\| > 1$ and $h_2 = \|u(\lambda_2)\| > 1$. Using Algorithm 1 we compute $h_1 = \|u(\lambda_1)\|$ and $h_2 = \|u(\lambda_2)\|$ and then construct the unique parabola with a given axis of symmetry through (h_1, λ_1) and (h_2, λ_2), see Fig. 4. We find the intersection of this parabola with the horizontal line $h = 1$, which gives a new approximation λ_3 to the value of $\lambda^\star$. We use algorithm 1 to compute $u(\lambda_3)$ and repeat. A more formal description is given in terms of Algorithm 2.

- Using the Thomas–Fermi approximation, chose $\lambda_2 < \lambda_1$ large enough such that the norms of the corresponding solutions, $\|u(\lambda_2)\|$ and $\|u(\lambda_1)\|$, are greater than 1,
- while $|h_2 - 1| > eps$ **do**
  - ○ construct a parabola that goes through the points with coordinates (λ_1, h_1^2) and (λ_2, h_2^2): $\lambda = \lambda_2 + ((\lambda_1 - \lambda_2)/(h_1^2 - h_2^2))(h^2 - h_2^2)$;
  - ○ find the intersection of this parabola with the horizontal line $h = 1$, and find the corresponding $\lambda : \lambda_3 = \lambda_2 + ((\lambda_1 - \lambda_2)/(h_1^2 - h_2^2))(1 - h_2^2)$;
  - ○ use Algorithm 1 to find the corresponding solution $u_{(\lambda_3)}$ and its norm h_3, and set $\lambda_1 = \lambda_2$; $h_1 = h_2$; $\lambda_2 = \lambda_3$; $h_2 = h_3$;
- **done**

Algorithm 2 is quite efficient. It typically takes 7–10 iterations only to converge.

4. Numerical experiments

Since no analytic solutions of the Gross–Pitaevskii equation are known to us, we use the Dual or Dirac Stepping (DS) method in two dimension as well as the Thomas–Fermi (TF) approximation for numerical comparisons. In three dimensions we can use TF only, because the results of the DS method are no longer available. For the stopping criterion of the fixed-point iteration we use the size of the residual, which is found by substituting the computed eigenvalue λ and the corresponding eigenfunction into the Gross–Pitaevskii equation.

Table 1
Two-dimensional data

k	DS	FP	TF
1	1.03	1.036706	0.488058
10	1.82	1.819286	1.543377
100	5.00	4.999394	4.880587
1000	15.49	15.524270	15.433771
3000	26.79	26.744547	26.732076
10,000	48.89	48.993929	48.805871

Table 2
Three-dimensional data

k	FP	TF	—FP–TF—
100	3.752857	3.170947	0.5819
1000	8.263033	7.965060	0.2980
3000	12.574191	12.360543	0.2136
10,000	20.154451	20.007326	0.1471
30,000	31.150280	31.048281	0.1020
50,000	38.177264	38.087026	0.0902

The computation was performed on a DELL Workstation with the operating system Red Hat Linux 5.2 in double precision, and used 64–128 points in each spatial dimension, depending on the size of the domain. The domain to which the equation is truncated is chosen according to the Thomas–Fermi approximation, that is, its size increases with increasing value of k.

In two dimensions we use the potential $V(x,y) = ax^2 + by^2$ with $a = 0.2$ and $b = 0.7$. Numerical results are summarized in Table 1. Although the two numerical approaches are quite different, the results show good agreement. To calculate the FP-column we performed the calculation for a number of of Legendre points. The values given in the table have all of the figures significant, as determined by increasing the number of Legendre points. In three dimensions we use $V(x,y,z) = 0.3x^2 + 0.6y^2 + 0.5z^2$. The results are shown in Table 2. As expected, the bigger is the coefficient k, the better is the agreement between the Thomas–Fermi estimates and our numerical results.

Appendix A. Discretization of the Gross–Pitaevski equation using Legendre polynomials

We consider the discretization of the truncated Gross–Pitaevskii equation

$$-u''(x) + ax^2u(x) + ku^3(x) = \lambda u(x), \tag{A.1}$$

$$u(\pm T) = 0, \quad \int_{-T}^{T} u^2(x)\,\mathrm{d}x = 1 \tag{A.2}$$

using Legendre polynomials. Let $P_n(x)$, $n = 0, 1, \ldots,$ denote Legendre polynomials of degree n. These polynomials are well documented in the literature. Properties which are relevant for our purpose can be found in [8]. Legendre polynomials are orthogonal on the interval $[-1, 1]$ and satisfy the recursion

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x), \tag{A.3}$$

where $P_0(x) \equiv 1$ and $P_1(x) = x$. Also $P_n(\pm 1) = (\pm 1)^n$. To normalize these polynomials scaling by the factor of $\sqrt{(2n+1)/2}$ is needed: $p_n(x) = \sqrt{(2n+1)/2}P_n(x)$. Now $p_n(x)$, $n = 1, \ldots, N+1$, are such that

$$\int_{-1}^{1} p_n^2(t)\,\mathrm{d}t = 1, \quad \int_{-1}^{1} p_n(t)p_k(t)\,\mathrm{d}t = 0, \quad k \neq n,\ k,n = 0,1,\ldots, \tag{A.4}$$

and

$$(n+1)\sqrt{\frac{2}{2n+3}}p_{n+1}(x) = (2n+1)\sqrt{\frac{2}{2n+1}}p_n - n\sqrt{\frac{2}{2n-1}}p_{n-1}. \tag{A.5}$$

Let

$$u(x) = \sum_{i=0}^{N+1} \alpha_i^{(N+2)} p_i(x), \quad -1 \leqslant x \leqslant 1, \tag{A.6}$$

and let

$$u(x_k^{(N+2)}) = \sum_{i=0}^{N+1} \alpha_i^{(N+2)} p_i(x_k^{(N+2)}), \qquad k = 1,\ldots,N+2, \tag{A.7}$$

where $x_1^{(N+2)}, x_2^{(N+2)}, \ldots, x_{N+2}^{(N+2)}$ are zeros of $p_{N+2}(x)$, which lie in $(-1,1)$. These points are a highly effective choice of collocation points [8]. They can be found as eigenvalues of a known tridiagonal matrix [16]

$$\begin{bmatrix} 0 & c_2 & 0 & \ldots & \ldots & 0 \\ c_2 & 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & c_i & \ddots & \vdots \\ \vdots & \ddots & c_i & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & c_{N+2} \\ 0 & \ldots & \ldots & 0 & c_{N+2} & 0 \end{bmatrix}, \quad c_i = \frac{\mathrm{i}-1}{\sqrt{(2\mathrm{i}-3)(2\mathrm{i}-1)}}. \tag{A.8}$$

In the matrix form (A.7) becomes

$$\begin{bmatrix} u(x_1^{(N+2)}) \\ \vdots \\ u(x_{N+2}^{(N+2)}) \end{bmatrix} = \begin{bmatrix} p_0(x_1^{(N+2)}) & \ldots & p_{N+1}(x_1^{(N+2)}) \\ \vdots & \ddots & \vdots \\ p_0(x_{N+2}^{(N+2)}) & \ldots & p_{N+1}(x_{N+2}^{(N+2)}) \end{bmatrix} \begin{bmatrix} \alpha_0^{(N+2)} \\ \vdots \\ \alpha_{N+1}^{(N+2)} \end{bmatrix} = P_{N+2} \begin{bmatrix} \alpha_0^{(N+2)} \\ \vdots \\ \alpha_{N+1}^{(N+2)} \end{bmatrix}. \tag{A.9}$$

Hence,

$$\begin{bmatrix} \alpha_0^{(N+2)} \\ \vdots \\ \alpha_{N+1}^{(N+2)} \end{bmatrix} = P_{N+2}^{-1} \begin{bmatrix} u(x_1^{(N+2)}) \\ \vdots \\ u(x_{N+2}^{(N+2)}) \end{bmatrix}. \tag{A.10}$$

To avoid the inversion of the matrix P_{N+2} in (A.10) we show that P_{N+2} has orthogonal rows. First notice that

$$\int_{-1}^{1} u^2(t)\,\mathrm{d}t = \sum_{i=0}^{N+1} (\alpha_i^{(N+2)})^2 = [u(x_1^{(N+2)}),\ldots,u(x_{N+2}^{(N+2)})] P_{N+2}^{-\mathrm{T}} P_{N+2}^{-1} \begin{bmatrix} u\left(x_1^{(N+2)}\right) \\ \vdots \\ u\left(x_{N+2}^{(N+2)}\right) \end{bmatrix}, \tag{A.11}$$

where $P_{N+2}^{-\mathrm{T}} = (P_{N+2}^{-1})^{\mathrm{T}}$. Alternatively, employing Gauss–Legendre quadrature gives us

$$\int_{-1}^{1} u^2(t)\,\mathrm{d}t = \sum_{i=1}^{N+2} \omega_i^2 u^2(x_i^{(N+2)}) = [u(x_1^{(N+2)}),\ldots,u(x_{N+2}^{(N+2)})] \begin{bmatrix} \omega_1^2 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \omega_{N+2}^2 \end{bmatrix} \begin{bmatrix} u(x_1^{(N+2)}) \\ \vdots \\ u(x_{N+2}^{(N+2)}) \end{bmatrix}, \tag{A.12}$$

where $\omega_1,\ldots,\omega_{N+2}$ are the Gauss–Legendre weights. Since the vector $[u(x_1^{(N+2)}),\ldots,u(x_{N+2}^{(N+2)})]$ is arbitrary, it follows by matching Eqs. (A.11) and (A.12) that

$$P_{N+2}^{-\mathrm{T}} P_{N+2}^{-1} = \begin{bmatrix} \omega_1^2 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \omega_{N+2}^2 \end{bmatrix} = W^2, \tag{A.13}$$

and hence,

$$P_{N+2}^{-1} = P_{N+2}^{\mathrm{T}} W^2. \tag{A.14}$$

The matrix P_{N+2} is computed using the recursion (A.5), the matrix of the Legendre weights, W, is found as $(W^{-1})^2 = P_{N+2}P_{N+2}^{\mathrm{T}}$, and (A.10) becomes

$$\begin{bmatrix} \alpha_0^{(N+2)} \\ \vdots \\ \alpha_{N+1}^{(N+2)} \end{bmatrix} = P_{N+2}^{\mathrm{T}} W^2 \begin{bmatrix} u(x_1^{(N+2)}) \\ \vdots \\ u(x_{N+2}^{(N+2)}) \end{bmatrix}. \tag{A.15}$$

Now if $u(x)$ is given by (A.7), then its second derivative $u''(x)$ can be written as

$$u''(x) = \sum_{j=0}^{N-1} \beta_j p_j(x), \tag{A.16}$$

where the coefficients β_j of the expansion (A.16) can be expressed in terms of the coefficients $\alpha_i^{(N+2)}$ using the following formula, see for instance [8]:

$$\beta_j = \left(j + \frac{1}{2}\right) \sum_{\substack{i=j+2,\\ j+i \text{ even}}}^{N+1} [i(i+1) - j(j+1)]\alpha_i^{(N+2)}. \tag{A.17}$$

Clearly only the first N Legendre polynomials are represented in this case and therefore

$$\begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{N-1} \end{pmatrix} = \bar{\mathscr{D}} \begin{pmatrix} \alpha_0^{(N+2)} \\ \alpha_1^{(N+2)} \\ \vdots \\ \alpha_{N-1}^{(N+2)} \\ \alpha_N^{(N+2)} \\ \alpha_{N+1}^{(N+2)} \end{pmatrix}, \tag{A.18}$$

where $\bar{\mathscr{D}}$ is $N \times (N+2)$. It is convenient to partition matrix $\bar{\mathscr{D}}$ as follows:

$$\bar{\mathscr{D}} = [\bar{\mathscr{D}}_1, \Phi], \tag{A.19}$$

where $\bar{\mathscr{D}}_1$ is an $N \times N$ matrix and Φ is an $N \times 2$ matrix consisting of the last two columns of $\bar{\mathscr{D}}$. Using boundary conditions, $u(\pm 1) = 0$, and the fact that $p_i(\pm 1) = (\pm 1)^i \sqrt{(2\mathrm{i}+1)/2}$, we obtain that

$$\sum_{i=0}^{N+1} \sqrt{\frac{2\mathrm{i}+1}{2}} \alpha_i^{(N+2)} = 0, \tag{A.20}$$

and

$$\sum_{i=0}^{N+1} (-1)^i \sqrt{\frac{2\mathrm{i}+1}{2}} \alpha_i^{(N+2)} = 0. \tag{A.21}$$

Solving for $\alpha_N^{(N+2)}$, $\alpha_{N+1}^{(N+2)}$ we have

$$\alpha_N^{(N+2)} = -\sum_{i=0}^{N/2-1} \sqrt{\frac{4\mathrm{i}+1}{2N+1}} \alpha_{2i}^{(N+2)}, \tag{A.22}$$

$$\alpha_{N+1}^{(N+2)} = -\sum_{i=0}^{N/2-1} \sqrt{\frac{4\mathrm{i}+3}{2N+3}} \alpha_{2i+1}^{(N+2)} \tag{A.23}$$

or in the matrix form

$$\begin{pmatrix} \alpha_N^{(N+2)} \\ \alpha_{N+1}^{(N+2)} \end{pmatrix} = \Psi \begin{pmatrix} \alpha_0^{(N+2)} \\ \vdots \\ \alpha_{N-1}^{(N+2)} \end{pmatrix}, \tag{A.24}$$

where

$$\Psi = -\begin{bmatrix} \sqrt{\frac{1}{2N+1}} & \sqrt{\frac{5}{2N+1}} & \cdots & \sqrt{\frac{2N-3}{2N+1}} \\ \sqrt{\frac{3}{2N+3}} & \sqrt{\frac{7}{2N+3}} & \cdots & \sqrt{\frac{2N-1}{2N+3}} \end{bmatrix}. \tag{A.25}$$

Now (A.19) becomes

$$\begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{N-1} \end{pmatrix} = \bar{\mathscr{D}}_1 \begin{pmatrix} \alpha_0^{(N+2)} \\ \alpha_1^{(N+2)} \\ \vdots \\ \alpha_{N-1}^{(N+2)} \end{pmatrix} + \Phi\Psi \begin{pmatrix} \alpha_0^{(N+2)} \\ \alpha_1^{(N+2)} \\ \vdots \\ \alpha_{N-1}^{(N+2)} \end{pmatrix} = \left(\bar{\mathscr{D}}_1 + \Phi\Psi\right) \begin{pmatrix} \alpha_0^{(N+2)} \\ \alpha_1^{(N+2)} \\ \vdots \\ \alpha_{N-1}^{(N+2)} \end{pmatrix}. \tag{A.26}$$

With the notation $D = \bar{\mathscr{D}}_1 + \Phi\Psi$, we write

$$\begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{N-1} \end{pmatrix} = D \begin{pmatrix} \alpha_0^{(N+2)} \\ \alpha_1^{(N+2)} \\ \vdots \\ \alpha_{N-1}^{(N+2)} \end{pmatrix}. \tag{A.27}$$

Here D is a square $N \times N$ matrix which transforms the first N coefficients $\alpha_0^{(N+2)}, \ldots, \alpha_{N-1}^{(N+2)}$ of the function $u(x)$, $u(\pm 1) = 0$, into the coefficients of the second derivative of the function $u(x)$. Notice also that

$$\begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{N-1} \end{pmatrix} = P_N^{-1} \begin{pmatrix} u''(x_1^{(N)}) \\ u''(x_2^{(N)}) \\ \vdots \\ u''(x_N^{(N)}) \end{pmatrix}, \tag{A.28}$$

where $x_i^{(N)}$, $i = 1, \ldots, N$, are now zeros of $P_N(x)$.

Before using the described technique to discretize (4) we make some remarks. In general $u(x)$ does not equal to $\sum_{i=0}^{N+1} \alpha_i^{(N+2)} p_i(x)$, but rather is approximated by this sum. For smooth $u(x)$ this approximation is very accurate, in fact, see [8], if $u(x) \in C^{p+1}_{[-1,1]}$, then

$$\max_{x\in[-1,1]} \left| u(x) - \sum_{i=0}^{N+1} \alpha_i^{(N+2)} p_i(x) \right| = \mathrm{O}\left(\frac{1}{N^p}\right).$$

If $u(x)$ is analytic, as in our case, the approximation is super-linear, or spectral in N (cf. [8]). This means that the approximation error eventually goes to zero faster than any power of N. This, in turn, implies that a relatively small N is needed for a highly accurate approximation, which becomes very important for the discretization in three spatial variables. This also implies that for sufficiently large N the difference between $\sum_{i=0}^{N-1} \alpha_i^{(N)} p_i(x)$ and $\sum_{i=0}^{N+1} \alpha_i^{(N+2)} p_i(x)$ is negligible, where α_i^N are defined via

$$\begin{bmatrix} u(x_1^{(N)}) \\ \vdots \\ u(x_N^{(N)}) \end{bmatrix} = \begin{bmatrix} p_0(x_1^{(N)}) & \ldots & p_{N-1}(x_1^{(N)}) \\ \vdots & \ddots & \vdots \\ p_0(x_N^{(N)}) & \ldots & p_{N-1}(x_N^{(N)}) \end{bmatrix} \begin{bmatrix} \alpha_0^N \\ \vdots \\ \alpha_{N-1}^N \end{bmatrix}, \tag{A.29}$$

and where $x_1^{(N)}, \ldots, x_N^{(N)}$ are now the zeros of $P_N(x)$. Therefore, without affecting the overall accuracy we replace $\alpha_0^{(N+2)}, \ldots, \alpha_{N-1}^{(N+2)}$ in (A.26) with $\alpha_0^N, \ldots, \alpha_{N-1}^N$.

For this reason, from now on, we drop the subscripts and write $\alpha_0, \ldots, \alpha_{N-1}$ in (A.10) given approximately by

$$\begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{N-1} \end{bmatrix} = \begin{bmatrix} p_0(x_1) & \ldots & p_{N-1}(x_1) \\ \vdots & \ddots & \vdots \\ p_0(x_N) & \ldots & p_{N-1}(x_N) \end{bmatrix}^{-1} \begin{bmatrix} u(x_1) \\ \vdots \\ u(x_N) \end{bmatrix} = P^{-1} \begin{bmatrix} u(x_1) \\ \vdots \\ u(x_N) \end{bmatrix}, \tag{A.30}$$

where $x_1, \ldots, x_N$ are the zeros of $P_N(x)$. With this understanding, it follows from (A.13), (A.29) and (A.30) that, approximately,

$$\begin{bmatrix} u''(x_1) \\ \vdots \\ u''(x_N) \end{bmatrix} = PDP^{\mathrm{T}} W^2 \begin{bmatrix} u(x_1) \\ \vdots \\ u(x_N) \end{bmatrix}, \tag{A.31}$$

where P is the matrix of (A.30). We also remark that the above discretization technique can be extended to the interval $[-T, T]$ by the simple change of variable, $x_i \to Tx_i$, $i = 1, \ldots, N$. For notational convenience, we use the same notation, x_i, for the points mapped into $(-T, T)$. Taking everything into account gives us the following discretization of (A.1):

$$-\frac{1}{T^2}[PDP^{\mathrm{T}} W^2]U + a \begin{bmatrix} x_1^2 & & 0 \\ & \ddots & \\ 0 & & x_N^2 \end{bmatrix} U + kU^3 = \lambda U, \tag{A.32}$$

where $U = [u(x_1), \ldots, u(x_N)]^{\mathrm{T}}$ and $U^3 = [u^3(x_1), \ldots, u^3(x_N)]^{\mathrm{T}}$. Again, for the sake of simplicity, we use the same notation $u(x_i)$ for the exact and the approximate values of the solution.

Let us consider now Eq. (A.32). The matrix $PDP^{\mathrm{T}}W^2$ is not symmetric in general. Symmetrization of the equation is desirable. Multiplying Eq. (A.32) by $\sqrt{T}W$ on the left and introducing the notation

$$A = -\frac{1}{T^2}[WPDP^{\mathrm{T}}W] + a\begin{bmatrix} x_1^2 & & 0 \\ & \ddots & \\ 0 & & x_N^2 \end{bmatrix}, \tag{A.33}$$

we obtain

$$AV + \frac{k}{T}W^{-2}V^3 = \lambda V, \tag{A.34}$$

where $V = \sqrt{T}WU$. Notice that

$$\int_{-T}^{\mathrm{T}} u^2(t)\,\mathrm{d}t = T\sum_{i=1}^{N} u(x_i)^2 w_i^2$$

and hence,

$$\int_{-T}^{\mathrm{T}} u^2(t)\,\mathrm{d}t = \sum_{i=1}^{N} v_i^2.$$

Appendix B. Thomas–Fermi estimates

In (4) a parameter of nonlinearity k can range from 1 to 10^6. Large coefficients k correspond to the Thomas–Fermi regime. In this regime $\Delta^2 u$ becomes negligible in the Gross–Pitaevskii equation

$$-\Delta^2 u + (ax^2 + by^2 + cz^2)u + ku^3 = \lambda u, \tag{B.1}$$

and hence approximately

$$ku^2 = \lambda - (ax^2 + by^2 + cz^2)$$

or

$$u^2 = \begin{cases} \frac{1}{k}(\lambda - (ax^2 + by^2 + cz^2)) & \text{if } ax^2 + by^2 + cz^2 \leqslant \lambda, \\ 0 & \text{otherwise.} \end{cases}$$

Since the norm of the solution is one we need λ such that

$$1 = \frac{1}{k}\int\int\int_{ax^2+by^2+cz^2 \leqslant \lambda} (\lambda - (ax^2 + by^2 + cz^2))\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z. \tag{B.2}$$

To find λ in the above equation first consider the double integral

$$\int\int_{ax^2+by^2 \leqslant \lambda} (ax^2 + by^2)\,\mathrm{d}x\,\mathrm{d}y, \tag{B.3}$$

which corresponds to the case of two spatial variables. Geometrically, this means the volume of the elliptic cylinder, V_{c}, of height λ minus the volume of the elliptic paraboloid, V_{p}, truncated at the level $ax^2 + by^2 = \lambda$.

The area of the base ellipse,

$$ax^2 + by^2 = \lambda$$

or

$$\frac{x^2}{\left(\sqrt{\frac{1}{a}}\right)^2} + \frac{y^2}{\left(\sqrt{\frac{1}{b}}\right)^2} = 1,$$

is given by

$$A(\lambda) = \pi \cdot \sqrt{\frac{\lambda^2}{ab}} = \frac{\lambda\pi}{\sqrt{ab}}. \tag{B.4}$$

Thus

$$V_{\mathrm{c}} = \lambda \cdot \frac{\lambda\pi}{\sqrt{ab}} = \frac{\pi}{\sqrt{ab}}\lambda^2. \tag{B.5}$$

Also

$$V_{\mathrm{p}} = \int_0^\lambda A(\mu)\,\mathrm{d}\mu = \frac{\pi}{\sqrt{ab}}\int_0^\lambda \mu\,\mathrm{d}\mu = \frac{1}{2}\frac{\pi}{\sqrt{ab}}\lambda^2,$$

and therefore

$$\int\int_{ax^2+by^2 \leqslant \lambda} (ax^2 + by^2)\,\mathrm{d}x\,\mathrm{d}y = V_{\mathrm{c}} - V_{\mathrm{p}} = \frac{\pi}{2\sqrt{ab}}\lambda^2 = k, \tag{B.6}$$

from which λ is easily found. Consider now

$$I(\lambda) = \int\int\int_{ax^2+by^2+cz^2 \leqslant \lambda} (ax^2 + by^2 + cz^2)\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z. \tag{B.7}$$

The volume of the ellipsoid $ax^2 + by^2 + cz^2 = \lambda$ or

$$\frac{x^2}{\left(\sqrt{\frac{\lambda}{a}}\right)^2} + \frac{y^2}{\left(\sqrt{\frac{\lambda}{b}}\right)^2} + \frac{z^2}{\left(\sqrt{\frac{\lambda}{c}}\right)^2} = 1 \tag{B.8}$$

is given by

$$V(\lambda) = \frac{4\pi}{3}\sqrt{\frac{\lambda^3}{abc}}. \tag{B.9}$$

In a similar way we have

$$I(\lambda) = \lambda V(\lambda) - \int_0^\lambda V(\mu)\,\mathrm{d}\mu = \frac{4\pi}{3\sqrt{abc}}\lambda^{5/2}. \tag{B.10}$$

Thus

$$k = \frac{4\pi}{3\sqrt{abc}}\lambda^{5/2} - \frac{4\pi}{5\sqrt{abc}}\lambda^{5/2} = \frac{8\pi}{15\sqrt{abc}}\lambda^{5/2} \tag{B.11}$$

and the Thomas–Fermi approximation of λ is

$$\lambda = \left(\frac{15k}{8\pi}\right)^{2/5}(abc)^{1/5}. \tag{B.12}$$

Note that in the case of spherical symmetry, $a = b = c = \frac{1}{2}$,

$$\lambda = \frac{1}{2}\left(\frac{15k}{4\pi}\right)^{2/5}.$$

References

[1] Written communication from anonymous referee.

[2] D. Baye, P.H. Heenen, Generalized meshes for quantum mechanical problems, Journal of Physics A (Mathematical and General) 19 (1986) 2041.

[3] T. Bergeman, D.L. Feder, N.L. Balazs, B.I. Schneider, Bose condensates in a harmonic trap near the critical temperature, Physical Review A 61 (2000) 063605.

[4] Y.-S. Choi, I. Koltracht, P.J. McKenna, A generalization of the Perron–Frobenius theorem for non-linear perturbations of Stieltjes matrices, Contemporary Mathematics, AMS 281 (2001) 325.

[5] Y.-S. Choi, I. Koltracht, P.J. McKenna, N. Savytska, Global monotone convergence of newton iteration for a nonlinear eigenproblem, Linear Algebra and its Applications 357 (2002) 217.

[6] F. Dalfovo, S. Giorgini, L.P. Pitaevskii, S. Stringari, Theory of Bose–Einstein condensation in trapped gasses, Reviews of Modern Physics 71 (1999) 463.

[7] M. Edwards, K. Burnett, Numerical solution of the nonlinear Schrödinger equation for small samples of trapped neutral atoms, Physical Review A 51 (1995) 1382.

[8] D. Gottlieb, S.A. Orszag, Numerical Analysis of Spectral Methods: Theory and Applications, Society for Industrial and Applied Mathematics, Philadelphia, c1977.

[9] A. Griffin, D.V. Snoke, A. Stringari (Eds.), Bose–Einstein Condensation, Cambridge University Press, Cambridge, 1995.

[10] A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 1985.

[11] M. Koštrun, J. Javanainen, An alternative numerical method for initial value problems involving the contact nonlinear Hamiltonians, Journal of Computational Physics 172 (2001) 298.

[12] G.I. Marchuk, Splitting and alternating direction methods, in: P.G. Ciarlet, J.L. Lions (Eds.), Handbook of Numerical Analysis, vol. I, North-Holland, Amsterdam, 1990.

[13] A. Messiah, Quantum Mechanics, Vol. II, Ch. XVII, Dover Publications, Mineola, 1999.

[14] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recepies—The Art of Scientific Computing, Cambridge University Press, Cambridge, 1986.

[15] B.I. Schneider, D.L. Feder, Numerical approach to the ground and excited states of a Bose–Einstein condensed gas confined in a completely anysotropic trap, Physical Review A 59 (1999) 2232.

[16] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, TAM 12, second ed., Springer, New York, 1992.

[17] J.W. Thomas, Numerical Partial Differential Equations, TAM 22, Springer, New York, 1995.

[18] F.A. Williams, Numerical Methods for Partial Differential Equations, Academic Press, New York, 1992.