Computer Physics Communications 286 (2023) 108669

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

OpenMP Fortran programs for solving the time-dependent dipolar Gross-Pitaevskii equation ☆

Luis E. Young-S. [a], Paulsamy Muruganandam [b], Antun Balaž [c], Sadhan K. Adhikari [d,*]

[a] Grupo de Modelado Computacional, Facultad de Ciencias Exactas y Naturales, Universidad de Cartagena, 130014 Cartagena, Bolivar, Colombia
[b] Department of Physics, Bharathidasan University, Palkalaiperur Campus, Tiruchirappalli 620024, Tamilnadu, India
[c] Institute of Physics Belgrade, University of Belgrade, Pregrevica 118, 11080 Belgrade, Serbia
[d] Instituto de Física Teórica, UNESP – Universidade Estadual Paulista, 01.140-70 São Paulo, São Paulo, Brazil

ARTICLE INFO

Article history:
Received 28 December 2022
Accepted 13 January 2023
Available online 24 January 2023

Keywords:
Dipolar Bose-Einstein condensate
Contact and dipolar interaction
Gross-Pitaevskii equation
Split-step Crank-Nicolson scheme
Fortran programs
Partial differential equation

ABSTRACT

In this paper we present Open Multi-Processing (OpenMP) Fortran 90/95 versions of previously published numerical programs for solving the dipolar Gross-Pitaevskii (GP) equation including the contact interaction in one, two and three spatial dimensions. The atoms are considered to be polarized along the z axis and we consider different cases, e.g., stationary and non-stationary solutions of the GP equation for a dipolar Bose-Einstein condensate (BEC) in one dimension (along x and z axes), two dimensions (in x-y and x-z planes), and three dimensions. The algorithm used is the split-step semi-implicit Crank-Nicolson scheme for imaginary- and real-time propagation to obtain stationary states and BEC dynamics, respectively, as in the previous version (Kishor Kumar et al., 2015 [3]). These OpenMP versions have significantly reduced execution time in multicore processors.

New version program summary

Program title: DBEC-GP-OMP, a program package containing programs imag3d-th.f90, real3d-th.f90, imag2dXY-th.f90, real2dXY-th.f90, imag2dXZ-th.f90, real2dXZ-th.f90, imag1dX-th.f90, real1dX-th.f90, imag1dZ-th.f90, real1dZ-th.f90, with fftw3.f03 and fftw3.mod.
CPC Library link to program files: https://doi.org/10.17632/sp3wvbtnmh.1
Licensing provisions: Apache License 2.0
Programming language: Open Multi-Processing (OpenMP) Fortran 90/95. The program is tested with the GNU, Intel, and Oracle (former Sun) compilers.
Journal reference of previous version: Comput. Phys. Commun. 195 (2015) 117. https://doi.org/10.1016/j.cpc.2015.03.024
Does the new version supersede the previous version?: Yes, Fortran programs
Nature of problem: The present OpenMP Fortran 90/95 programs solve the time-dependent nonlinear partial differential Gross-Pitaevskii (GP) equation for a trapped dipolar Bose-Einstein condensate (BEC) in one (1D), two (2D), and three (3D) spatial dimensions.
Solution method: We employ the split-step Crank-Nicolson scheme to discretize the time-dependent GP equation in space and time. The discretized equation is then solved by imaginary- or real-time propagation, employing adequately small space and time steps, to yield the solution of stationary and non-stationary problems, respectively.
Reason for new version: Previously published Fortran programs [1] for solving the GP equation for a BEC have become useful tools. These programs have been translated to the C programming language [2] and later extended to the more complex scenario of dipolar atoms [3], spinor condensates [4], and rotating condensates [5]. Now virtually all computers have multi-core processors and some have motherboards with more than one physical computer processing unit (CPU), which may increase the number of available CPU cores on a single computer to several tens. The Fortran [6] and C [7] programs for a nondipolar BEC have been adopted to be very fast on such multi-core modern computers. The C programs for a dipolar BEC have been adopted to multicore processors to yield OpenMP, OpenMP/MPI, and CUDA/MPI versions [8, 9].

☆ The review of this paper was arranged by Prof. J. Ballantyne.

* Corresponding author.
E-mail addresses: lyoung@unicartagena.edu.co (L.E. Young-S.), anand@bdu.ac.in (P. Muruganandam), antun@ipb.ac.rs (A. Balaž), sk.adhikari@unesp.br (S.K. Adhikari).

https://doi.org/10.1016/j.cpc.2023.108669

The available Fortran 90/95 and C programs for the solution of the GP equation for a dipolar BEC in 1D, 2D, and 3D are enjoying widespread use [1–3]. Later, OpenMP/MPI, and CUDA/MPI [8, 9] versions of these C programs were published, which highly reduce the CPU times in multicore processors. Although, Fortran 90/95 is a popular programming language, the previous Fortran 90/95 programs for a dipolar BEC [3], lacking an efficient parallelization, suffer from a drawback of requiring large CPU time. In this paper we present the OpenMP version of the Fortran 90/95 programs for the solution of the GP equation for a dipolar BEC in 1D, 2D, and 3D, ideal for an efficient execution in multicore processors, which highly reduce the CPU time.

Summary of revisions: The program package DBEC-GP-OMP contains 3D programs imag3d-th.f90, real3d-th.f90, 2D programs imag2dXY-th.f90, real2dXY-th.f90, imag2dXZ-th.f90, real2dXZ-th.f90, and 1D programs imag1dX-th.f90, real1dX-th.f90, imag1dZ-th.f90, real1dZ-th.f90, in the directory src, as well as the files makefile, readme.txt, and readme-fftw.txt. The 3D and 2D programs are OpenMP versions leading to a significant reduction in execution time in a multicore processor. However, in 1D the supplied programs are autoparallel versions as there is no further reduction in the execution time in the OpenMP versions. The makefile allows an automated compilation of the programs using different supported compilers (GNU, Intel, Sun Oracle) by a simple make command, as in Ref. [3]. However, the 1D and 2D dipolar programs in previous publications [3, 8, 9] erroneously used the 3D nonlinearities, in place of corresponding 1D and 2D nonlinearities, in numerical calculations. The corresponding arXiv archive versions do not have these errors. The file readme.txt contains instructions on how to compile and run the programs. The file readme-fftw.txt provides instructions for installing the fast Fourier transformation (FFT) routine in Linux/Unix operating systems unless this routine is preinstalled. All input parameters are placed in the beginning of each program in MODULEs GPE_DATA and COMM_DATA. The directory output contains examples of matching outputs of imaginary- and real-time propagation programs in subdirectories with a generic name, e.g. imag3d-th, real2dXY-th, etc. The reader is advised to consult Ref. [3] for details. The imaginary-time programs implement the calculation starting from an initial analytic Gaussian wave function using a non-zero value of the parameter NSTP, whereas the real-time programs use the converged solution of the imaginary-time programs as the initial state of calculation employing NSTP = 0. The FFT algorithm works faster and more efficiently when we take the number of space grid points − N in 1D and NX, NY, NZ in 2D and 3D − in powers of 2, e.g. 2^n with n an integer, and should be so chosen for reducing the execution time. The operational scheme for running the codes is identical to that in Ref. [3].

References

[1] P. Muruganandam, S.K. Adhikari, Comput. Phys. Commun. 180 (2009) 1888, arXiv:0904.3131.
[2] D. Vudragović, I. Vidanović, A. Balaž, P. Muruganandam, S.K. Adhikari, Comput. Phys. Commun. 183 (2012) 2021, arXiv:1206.1361.
[3] R. Kishor Kumar, L.E. Young-S., D. Vudragović, A. Balaž, P. Muruganandam, S.K. Adhikari, Comput. Phys. Commun. 195 (2015) 117, arXiv:1506.03283.
[4] R. Ravisankar, D. Vudragović, P. Muruganandam, A. Balaz, S.K. Adhikari, Comput. Phys. Commun. 259 (2021) 107657, arXiv:2009.13507.
[5] R. K. Kumar, V. Lončar, P. Muruganandam, S. K. Adhikari, A. Balaž, Comput. Phys. Commun. 240 (2019) 74, arXiv:1906.06327.
[6] L.E. Young-S., P. Muruganandam, S.K. Adhikari, V. Loncar, D. Vudragović, Antun Balaž, Comput. Phys. Commun. 220 (2017) 503, arXiv:1709.04423.
[7] B. Satarić, V. Slavnić, A. Belić, A. Balaž, P. Muruganandam, S.K. Adhikari, Comput. Phys. Commun. 200 (2016) 411, arXiv:1601.04641.
[8] V. Lončar, L.E. Young-S., S. Škrbić, P. Muruganandam, S.K. Adhikari, A. Balaž, Comput. Phys. Commun. 209 (2016) 190, arXiv:1610.05329.
[9] V. Lončar, A. Balaž, A. Bogojević, S. Škrbić, P. Muruganandam, S.K. Adhikari, Comput. Phys. Commun. 200 (2016) 406, arXiv:1601.04640.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

L.E.Y.-S. acknowledges the financial support by the Vicerrectoria de Investigaciones, Universidad de Cartagena, Colombia through Project No. 019-2021. P.M. acknowledges the financial support by the Council of Scientific and Industrial Research (CSIR), India under Grant No. 03(1422)/18/EMR-II, and Science and Engineering Research Board (SERB), India under Grant No. CRG/2019/004059. A.B. acknowledges funding provided by the Institute of Physics Belgrade, through a grant by the Ministry of Science, Technological Development, and Innovations of the Republic of Serbia. S.K.A. acknowledges support by the CNPq, Brazil through Grant No. 301324/2019-0.