

# Numeričko rješavanje Gross-Pitaevskiijeve jednadžbe u dvije dimenzije

---

Poparić, Ivan

Undergraduate thesis / Završni rad

2022

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Split, University of Split, Faculty of science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:166:596218>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-02-26**

*Repository / Repozitorij:*

[Repository of Faculty of Science](#)



Sveučilište u Splitu  
Prirodoslovno – matematički fakultet

**Numeričko rješavanje Gross-Pitaevskiijeve  
jednadžbe u dvije dimenzije**

Završni rad

Ivan Poparić

Split, rujan 2022.

## Temeljna dokumentacijska kartica

Sveučilište u Splitu  
Prirodoslovno – matematički fakultet  
Odjel za fiziku  
Ruđera Boškovića 33, 21000 Split, Hrvatska

Završni rad

### Numeričko rješavanje Gross-Pitaevskiijeve jednadžbe u dvije dimenzije

Ivan Poparić

Sveučilišni preddiplomski studij Fizika

#### Sažetak:

Gross-Pitaevskiijeva jednadžba opisuje ponašanje bozona u Bose–Einsteinovom kondenzatu pri apsolutnoj nuli. Jednadžba je nelinearna i nije analitički rješiva, zbog čega se služimo numeričkim metodama. Cilj ovoga rada je riješiti vremenski ovisnu Gross-Pitaevskiijevu jednadžbu za kondenzat u sferno simetričnoj harmonijskoj magnetskoj zamci u dvije dimenzije i dobiti uvid u oblik valne funkcije stacionarnog stanja za privlačne i odbojne interakcije između bozona. Za potrebe numeričkog rješavanja diskretizira se domena valne funkcije i pronalazi algoritam za rješavanje jednadžbe. Rješenja su uspoređena s postojećom literaturom.

**Ključne riječi:** Gross-Pitaevskiijeva jednadžba, Crank-Nicolson shema, Bose-Einsteinov kondenzat, numeričko rješenje

**Rad sadrži:** 21 stranicu, 6 slika, 0 tablica, 4 literaturna navoda. Izvornik je na hrvatskom jeziku.

**Mentor:** doc. dr. sc. Petar Stipanović

**Ocjenjivači:** doc. dr. sc. Petar Stipanović,  
prof. dr. sc. Leandra Vranješ Markić,  
doc. dr. sc. Ivana Weber

**Rad prihvaćen:** 9. rujna 2022.

Rad je pohranjen u Knjižnici Prirodoslovno – matematičkog fakulteta, Sveučilišta u Splitu.

<b>Basic documentation card</b>
---------------------------------

University of Split  
Faculty of Science  
Department of Physics  
Ruđera Boškovića 33, 21000 Split, Croatia

Bachelor thesis

**Numerical solutions to the Gross-Pitaevskii equation in two dimensions**

Ivan Poparić

University undergraduate study programme Physics

**Abstract:**

The Gross-Pitaevskii equation describes the behavior of bosons in the Bose–Einstein condensate at absolute zero. The equation is non-linear and cannot be solved analytically, which is why we use numerical methods. The aim of this thesis is to solve the time-dependent Gross-Pitaevskii equation for a condensate in a spherically symmetric harmonic magnetic trap in two dimensions and to gain insight into the shape of the stationary state wave function for attractive and repulsive interactions between bosons. For the purposes of finding a numerical solution, the domain of the wave function is discretized and an algorithm for solving the equation is found. The solutions are compared with existing literature.

**Keywords:** Gross-Pitaevskii equation, Crank-Nicolson scheme, Bose-Einstein condensate, numerical solution

**Thesis consists of:** 21 pages, 6 figures, 0 tables, 4 references. Original language: Croatian.

**Supervisor:** Assist. Prof. Dr. Petar Stipanović

**Reviewers:** Assist. Prof. Dr. Petar Stipanović,  
Prof. Dr. Leandra Vranješ Markić,  
Assist. Prof. Dr. Ivana Weber

**Thesis accepted:** September 9, 2022.

This thesis is deposited in the library of the Faculty of Science, University of Split.

# Sadržaj

<b>1</b>	<b>Uvod</b> . . . . .	<b>1</b>
1.1	Gross-Pitaevskiijeva jednađba . . . . .	2
<b>2</b>	<b>Metoda</b> . . . . .	<b>4</b>
<b>3</b>	<b>Rješavanje jednađbe</b> . . . . .	<b>7</b>
3.1	Algoritam . . . . .	7
<b>4</b>	<b>Rješenja</b> . . . . .	<b>10</b>
<b>5</b>	<b>Zaključak</b> . . . . .	<b>14</b>
<b>A</b>	<b>Priloženi kod</b> . . . . .	<b>16</b>
<b>B</b>	<b>Priložena gnuplot skripta</b> . . . . .	<b>20</b>

# 1 Uvod

U kvantnoj fizici ne razlikujemo identične čestice, a s obzirom na ponašanje valne funkcije kada zamijenimo čestice, čestice možemo podijeliti na fermione i bozone. Bozoni su čestice s cjelobrojnim spinom i opisuju ih Bose-Einsteinova statistika, dok fermioni imaju polucjelobrojan spin i opisuju ih Fermi-Diracovom statistikom. Valna je funkcija bozonskog sustava simetrična, pa bozoni ne podliježu Paulijevom principu, odnosno proizvoljno mnogo bozona može biti u istom kvantnom stanju. Broj bozona u  $i$ -tom stanju dan je Bose-Einsteinovom funkcijom razdiobe

$$n_i = \frac{1}{\exp\left(\frac{E_i - \mu}{k_B T}\right) - 1}, \quad (1.1)$$

gdje je  $E_i$  energija  $i$ -tog stanja,  $\mu$  kemijski potencijal,  $k_B$  Boltzmannova konstanta, a  $T$  temperatura. Snižavanjem temperature bozonskog plina ispod kritične temperature  $T_c$  bozoni prelaze u osnovno stanje te dolazi do Bose-Einsteinove kondenzacije. Kritična temperatura za idealni bozonski plin dana je s [1]

$$T_c = \frac{h^2}{2\pi m k_B} \left( \frac{N}{V \zeta\left(\frac{3}{2}\right)} \right)^{\frac{2}{3}}, \quad (1.2)$$

gdje je  $h$  Planckova konstanta,  $m$  masa bozona,  $N$  broj bozona,  $V$  volumen u kojem se nalaze i  $\zeta\left(\frac{3}{2}\right)$  vrijednost Riemannove zeta funkcije. Na temperaturama nižim od  $T_c$  bozonski se sustav nalazi u dvije faze, plin i Bose-Einsteinov kondenzat (BEC). Udio bozona koji su u kondenzatu raste snižavanjem temperature. Na apsolutnoj nuli,  $T = 0$  K, svi su bozoni u Bose-Einsteinovom kondenzatu. Svojstva BEC znatno su drugačija od običnih plinova. Gustoća kondenzata reda je veličine  $10^{13} - 10^{15} \text{ cm}^{-3}$  što je puno manje u odnosu na gustoću zraka koja je reda veličine  $10^{19} \text{ cm}^{-3}$ . Temperatura potrebna za postizanje BEC za plinove toliko niskih gustoća reda je veličine  $< 10^{-5}$  K. [2]

Einstein je teorijski predvidio postojanje BEC 1925., a Cornell i Wieman su 1995. eksperimentalno postigli kondenzaciju  $^{87}\text{Rb}$  [1, 2]. Kasnije je postignuta kondenzacija s  $^{23}\text{Na}$ ,  $^7\text{Li}$ ,  $^{41}\text{K}$  i drugim atomima, naročito alkalijskim, a prva kondenzacija molekule postignuta je 2003. [1] Kondenzacija alkalijskih atoma se postiže zarobljavanjem atoma u magnetskim zamkama, u kojima se potom hlade laserima i evaporativno. Pri evaporativnom hlađenju atomi viših energija uklanjaju se iz zamke. [2]

## 1.1 Gross-Pitaevskiijeva jednadžba

Kada proučavamo svojstva realnih plinova potrebno je imati opis međučestične interakcije. Međuatomske interakcije općenito su komplicirane za opisati, ali u razrijeđenom BEC efektivnu interakciju između dva atoma možemo opisati konstantom i Diracovom delta funkcijom

$$V_{1,2} = U_0 \delta(\vec{r}_1 - \vec{r}_2). \quad (1.3)$$

U gornjoj su jednadžbi  $\vec{r}_1$  i  $\vec{r}_2$  položaji dva atoma, a  $U_0$  je konstanta. Kada promatramo cijeli kondenzat, za opis potencijala međuatomske interakcije služimo se teorijom srednjeg polja tako da je proporcionalan gustoći čestica [2],

$$V_{\text{međuatomska}}(\vec{r}, \tau) = gN |\Psi(\vec{r}, \tau)|^2, \quad (1.4)$$

gdje je  $\Psi(\vec{r}, \tau)$  valna funkcija Bose-Einsteinova kondenzata na položaju  $\vec{r}$  i u trenutku  $\tau$ ,  $N$  broj bozona, a  $g$  je jakost međuatomske interakcije. Pretpostavljamo da je valna funkcija sustava produkt jednočestičnih valnih funkcija  $\phi_i(\vec{r}_i, \tau)$

$$\Psi(\vec{r}, \tau) = \prod_i \phi_i(\vec{r}_i, \tau). \quad (1.5)$$

Za  $g < 0$  interakcija je privlačna, a za  $g > 0$  odbojna. Kada je duljina raspršenja<sup>1</sup> puno manja od srednje međuatomske udaljenosti navedene pretpostavke vrijede, pa valna funkcija BEC na apsolutnoj nuli zadovoljava Gross-Pitaevskiijevu jednadžbu (GPE)[2]

$$\left[ -\frac{\hbar^2}{2m} \nabla^2 + V(\vec{r}, \tau) + gN |\Psi(\vec{r}, \tau)|^2 \right] \Psi(\vec{r}, \tau) = i\hbar \frac{\partial}{\partial \tau} \Psi(\vec{r}, \tau), \quad (1.6)$$

gdje prvi član predstavlja kinetičku energiju, a  $V(\vec{r}, \tau)$  je vanjski potencijal.

Za BEC u harmonijskoj magnetskoj zamci frekvencije  $\omega$  GPE poprima oblik

$$\left[ -\frac{\hbar^2}{2m} \nabla^2 + \frac{1}{2} m \omega^2 r^2 + gN |\Psi(\vec{r}, \tau)|^2 - i\hbar \frac{\partial}{\partial \tau} \right] \Psi(\vec{r}, \tau) = 0, \quad (1.7)$$

gdje je  $m$  masa atoma. U dvije dimenzije za sferno simetrično rješenje  $\Psi(\vec{r}, \tau) \equiv \varphi(r, \tau)$  jednadžba (1.7) postaje

$$\left[ -\frac{\hbar^2}{2m} \frac{1}{r} \frac{\partial}{\partial r} r \frac{\partial}{\partial r} + \frac{1}{2} m \omega^2 r^2 + gN |\varphi(r, \tau)|^2 - i\hbar \frac{\partial}{\partial \tau} \right] \varphi(r, \tau) = 0. \quad (1.8)$$

Takav sustav možemo postići ako je termalna energija bozona puno manja od energije oscilatora u jednome smjeru, tj.  $\hbar\omega_z \gg k_B T_c > \hbar\omega_x = \hbar\omega_y$ . [3] Radi lakšeg rješavanja GPE prelazimo na

<sup>1</sup>Pri niskim energijama dovoljno je promatrati s-valno raspršenje. U granici s-valnog raspršenja valna funkcija ima oblik  $\psi = 1 - \frac{a}{r}$ , gdje konstantu  $a$  zovemo duljinom raspršenja [2].

bezdimezionalne varijable:  $x \equiv \frac{r}{a_{\text{HO}}}$ ,  $t \equiv \tau \frac{\omega}{2}$ ,  $\psi(x, t) = a_{\text{HO}} \sqrt{2\pi} \varphi(r, \tau)$ , gdje je  $a_{\text{HO}} \equiv \sqrt{\frac{\hbar}{m\omega}}$  karakteristična duljina zamke. Tada jednađžba (1.8) postaje

$$\left[ -\frac{1}{x} \frac{\partial}{\partial x} x \frac{\partial}{\partial x} + x^2 + \frac{mgN}{\pi^2 \hbar^2} |\psi(x, t)|^2 - i \frac{\partial}{\partial t} \right] \psi(x, t) = 0, \quad (1.9)$$

$\frac{mgN}{\pi^2 \hbar^2} \equiv cn$ , gdje je  $n$  reducirani broj čestica, dok  $c = \text{sgn}(g)$  predstavlja vrstu interakcije. Za pozitivan  $c$  interakcija je odbojna, a za negativan privlačna.

GPE općenito je analitički nerješiva, stoga ćemo je riješiti numerički Crank-Nicolsonovom shemom, koja je kombinacija implicitne i eksplicitne sheme rješavanja parcijalnih diferencijalnih jednađžbi. Metoda je detaljnije opisana u literaturi [4]. Za rješavanje GPE služimo se supstitucijom [3]

$$\phi(x, t) = x\psi(x, t), \quad (1.10)$$

pa jednađžba (1.9) poprima oblik

$$\left[ -\frac{\partial^2}{\partial x^2} + \frac{1}{x} \frac{\partial}{\partial x} - \frac{1}{x^2} + x^2 + cn \frac{|\phi(x, t)|^2}{x^2} - i \frac{\partial}{\partial t} \right] \phi(x, t) = 0. \quad (1.11)$$

Za pronalazak rješenja potrebni su rubni uvjeti. Rubni uvjeti za  $\psi(x, t)$  su [3]

$$\psi(0, t) = \text{konst.}, \quad \psi'(0, t) = 0, \quad (1.12)$$

$$\lim_{x \rightarrow \infty} \psi(x, t) = 0. \quad (1.13)$$

Za valnu funkciju također vrijedi i uvjet normiranosti

$$\int_0^\infty d\vec{r} |\Psi(\vec{r}, \tau)|^2 = 1, \quad (1.14)$$

koji prelaskom na  $\psi(x, t)$ , uz  $d\vec{r} = 2\pi r dr$ , postaje

$$\int_0^\infty dx |\psi(x, t)|^2 x = 1. \quad (1.15)$$

Fokusirat ćemo se na stacionarni problem, kojeg ćemo riješiti primjenom Crank-Nicolsonove sheme na vremenski ovisnu jednađžbu (1.11). Problem se također može riješiti primjenom Runge-Kutta 4 metode za rješavanje običnih diferencijalnih jednađžbi na vremenski neovisnu Gross-Pitaevskiijevu jednađžbu, analogno vremenski neovisnoj Schrödingerovoj valnoj jednađžbi, pretpostavljajući separabilno rješenje  $\Psi(\vec{r}, \tau) = \Psi(\vec{r}) e^{-i\mu\tau/\hbar}$ . U drugom poglavlju opisana je metoda rješavanja, u trećem poglavlju raspravljen je algoritam, rezultati su prikazani u četvrtom poglavlju, a kritički osvrt i zaključci sažeti su u zadnjem poglavlju.



## 2 Metoda

Za numeričko rješavanje jednađžbe (1.11) potrebno je diskretizirati domenu funkcije  $\phi(x, t)$ . Rješenje tražimo na prostornom intervalu  $[x_0, x_N]$  u vremenskom periodu  $[t_0, t_M]$ . Uzimamo  $x_0 = 0$  i  $t_0 = 0$ . Interval  $[0, x_N]$  dijelimo na  $N$  segmenata sa  $N + 1$  točkom tako da je koordinata svake točke  $x_j, j = 0, 1, 2, \dots, N$ , dana sa

$$x_j = j \cdot \Delta x. \quad (2.1)$$

Prostorni korak  $\Delta x$  iznosi  $\Delta x = \frac{x_N}{N}$ . Vremenski interval dijelimo na isti način,  $M + 1$  trenutak  $t_k, k = 0, 1, \dots, M, \Delta t = \frac{t_M}{M}$ ,

$$t_k = k \cdot \Delta t. \quad (2.2)$$

Skup svih točaka  $(x_j, t_k)$  definira mrežu u čijim čvorovima tražimo vrijednost funkcije  $\phi(x_j, t_k) \equiv \phi_j^k$ .

Derivacije funkcije  $\phi(x_j, t_k)$  po prostornoj koordinati izražavamo metodom konačnih razlika [4]:

$$\frac{\partial}{\partial x} \phi(x_j, t_k) \approx \frac{\phi_{j+1}^k - \phi_{j-1}^k}{2\Delta x}, \quad (2.3)$$

i

$$\frac{\partial^2}{\partial x^2} \phi(x_j, t_k) \approx \frac{\phi_{j+1}^k - 2\phi_j^k + \phi_{j-1}^k}{\Delta x^2}. \quad (2.4)$$

Derivacije funkcije  $\phi(x_j, t_k)$  po prostornoj koordinati možemo napisati kao derivacije u trenutku  $t_k$ , kao u izrazima gore, što rezultira eksplicitnom shemom, ili kao derivacije u trenutku  $t_{k+1}$ , što rezultira implicitnom shemom. Jednađžbu (1.11) riješavamo Crank-Nicolsonovom shemom, koja kombinira obje sheme. Derivacija funkcije  $\phi(x_j, t_k)$  po vremenu glasi

$$\frac{\partial}{\partial t} \phi(x_j, t_k) \approx \frac{\phi_j^{k+1} - \phi_j^k}{\Delta t}. \quad (2.5)$$

Primjenom Crank-Nicolson sheme na jednađžbu (1.11) dobije se [3]

$$\begin{aligned} \frac{i(\phi_j^{k+1} - \phi_j^k)}{\Delta t} &= -\frac{1}{2(\Delta x)^2} [(\phi_{j+1}^{k+1} - 2\phi_j^{k+1} + \phi_{j-1}^{k+1}) + (\phi_{j+1}^k - 2\phi_j^k + \phi_{j-1}^k)] \\ &+ \frac{1}{4x_j\Delta x} [(\phi_{j+1}^{k+1} - \phi_{j-1}^{k+1}) + (\phi_{j+1}^k - \phi_{j-1}^k)] \\ &+ \frac{1}{2} \left[ x_j^2 - \frac{1}{x_j^2} + cn \frac{|\phi_j^k|^2}{x_j^2} \right] (\phi_j^{k+1} + \phi_j^k). \end{aligned} \quad (2.6)$$

Uvodimo pokratu

$$\xi_j = \left[ x_j^2 - \frac{1}{x_j^2} + cn \frac{|\phi_j^k|^2}{x_j^2} \right]. \quad (2.7)$$

U izrazu (2.6) grupiramo sve članove ovisne o  $t_{k+1}$  na lijevu stranu, a one ovisne o  $t_k$  na desnu,

$$\begin{aligned} \frac{i}{\Delta t} \phi_j^{k+1} + \frac{1}{2(\Delta x)^2} (\phi_{j+1}^{k+1} - 2\phi_j^{k+1} + \phi_{j-1}^{k+1}) - \frac{1}{4x_j \Delta x} (\phi_{j+1}^{k+1} - \phi_{j-1}^{k+1}) - \frac{1}{2} \xi_j \phi_j^{k+1} = \\ \frac{i}{\Delta t} \phi_j^k - \frac{1}{2(\Delta x)^2} (\phi_{j+1}^k - 2\phi_j^k + \phi_{j-1}^k) + \frac{1}{4x_j \Delta x} (\phi_{j+1}^k - \phi_{j-1}^k) + \frac{1}{2} \xi_j \phi_j^k. \end{aligned} \quad (2.8)$$

Dalje preoblikujemo (2.8):

$$\begin{aligned} \left( \frac{1}{2(\Delta x)^2} + \frac{1}{4x_j \Delta x} \right) \phi_{j-1}^{k+1} + \left( \frac{i}{\Delta t} - \frac{1}{(\Delta x)^2} - \frac{1}{2} \xi_j \right) \phi_j^{k+1} + \left( \frac{1}{2(\Delta x)^2} - \frac{1}{4x_j \Delta x} \right) \phi_{j+1}^{k+1} = \\ \left( -\frac{1}{2(\Delta x)^2} - \frac{1}{4x_j \Delta x} \right) \phi_{j-1}^k + \left( \frac{i}{\Delta t} + \frac{1}{(\Delta x)^2} + \frac{1}{2} \xi_j \right) \phi_j^k + \left( -\frac{1}{2(\Delta x)^2} + \frac{1}{4x_j \Delta x} \right) \phi_{j+1}^k. \end{aligned} \quad (2.9)$$

U izrazu (2.9) možemo uvesti supstitucije za izraze u zagradama uz  $\phi$

$$a_j^{k+1} \phi_{j-1}^{k+1} + b_j^{k+1} \phi_j^{k+1} + c_j^{k+1} \phi_{j+1}^{k+1} = a_j^k \phi_{j-1}^k + b_j^k \phi_j^k + c_j^k \phi_{j+1}^k, \quad (2.10)$$

i uočiti da vrijedi:

$$a_j^k = -a_j^{k+1} = -a_j, \quad (2.11)$$

$$b_j^k = \frac{2i}{\Delta t} - b_j^{k+1} = \frac{2i}{\Delta t} - b_j, \quad (2.12)$$

$$c_j^k = -c_j^{k+1} = -c_j. \quad (2.13)$$

Jednačžbu (2.10) možemo zapisati u matričnom obliku

$$\begin{bmatrix} b_1 & c_1 & 0 & 0 & \dots \\ a_2 & b_2 & c_2 & 0 & \dots \\ 0 & a_3 & b_3 & c_3 & \dots \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix} \cdot \begin{bmatrix} \phi_1^{k+1} \\ \phi_2^{k+1} \\ \phi_3^{k+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \frac{2i}{\Delta t} - b_1 & -c_1 & 0 & 0 & \dots \\ -a_2 & \frac{2i}{\Delta t} - b_2 & -c_2 & 0 & \dots \\ 0 & -a_3 & \frac{2i}{\Delta t} - b_3 & -c_3 & \dots \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix} \cdot \begin{bmatrix} \phi_1^k \\ \phi_2^k \\ \phi_3^k \\ \vdots \end{bmatrix}. \quad (2.14)$$

Uvedemo oznake:

$$\mathbf{A}^{k+1} = \begin{bmatrix} b_1 & c_1 & 0 & 0 & \dots \\ a_2 & b_2 & c_2 & 0 & \dots \\ 0 & a_3 & b_3 & c_3 & \dots \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}, \quad (2.15)$$

$$\tilde{\mathbf{A}}^k = \begin{bmatrix} \frac{2i}{\Delta t} - b_1 & -c_1 & 0 & 0 & \dots \\ -a_2 & \frac{2i}{\Delta t} - b_2 & -c_2 & 0 & \dots \\ 0 & -a_3 & \frac{2i}{\Delta t} - b_3 & -c_3 & \dots \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}, \quad (2.16)$$

$$\Phi^{k+1} = \begin{bmatrix} \phi_1^{k+1} \\ \phi_2^{k+1} \\ \phi_3^{k+1} \\ \vdots \end{bmatrix}, \quad (2.17)$$

i

$$\Phi^k = \begin{bmatrix} \phi_1^k \\ \phi_2^k \\ \phi_3^k \\ \vdots \end{bmatrix}. \quad (2.18)$$

S novim oznakama jednačnja (2.14) postaje

$$\mathbf{A}^{k+1} \Phi^{k+1} = \tilde{\mathbf{A}}^k \Phi^k. \quad (2.19)$$

Desna je strana (2.19) eksplicitni dio rješenja

$$\tilde{\Phi}^k = \begin{bmatrix} (\frac{2i}{\Delta t} - b_1^k) \phi_1^k - c_1 \phi_2^k \\ -a_2 \phi_1^k + (\frac{2i}{\Delta t} - b_2^k) \phi_2^k - c_2 \phi_3^k \\ \vdots \end{bmatrix}. \quad (2.20)$$

Jednačnja

$$\mathbf{A}^{k+1} \Phi^{k+1} = \tilde{\Phi}^k \quad (2.21)$$

je tridijagonalna matična jednačnja oblika  $\mathbf{A}\mathbf{X} = \mathbf{D}$ , čije se rješenje lako pronade pojednostavljenim oblikom Gaussove eliminacije [4].

### 3 Rješavanje jednačbe

Za rješavanje jednačbe (2.9) potrebno je diskretizirati domenu, odnosno definirati prostorni i vremenski interval na kojem tražimo rješenje,  $[x_0, x_N]$  i  $[t_0, t_M]$  te  $\Delta x$  i  $\Delta t$ . Nakon diskretizacije prostora, definiramo glavni i pomoćni niz u koje ćemo pohranjivati vrijednosti  $\phi_j^k$ . Također definiramo  $c$  i  $\Delta n$ .  $\Delta n$  je iznos za koji ćemo iterativno povećavati reducirani broj čestica  $n$ , od 0 do željene vrijednosti, nakon čega  $n$  držimo konstantnim. Konstanta  $c = \pm 1$  je karakteristika interakcije, za  $c = -1$  interakcija je privlačna, a za  $c = +1$  odbojna. Postupno povećanje  $n$  odgovara postupnom jačanju interakcije, ili dodavanju čestica u kondenzat. Ako bismo na početku postavili  $n$  na željenu vrijednost, utjecaj nelinearnog člana bio bi prevelik i izgubili bismo stabilnost rješenja. Prevelik  $\Delta n$  iz istog razloga dovodi do nestabilnosti.

Potom definiramo početne i rubne uvjete. Za rubne uvjete uzimamo  $\phi_0 = 0$ , što je opravdano prema definiciji (1.10), i  $\phi_N = 0$ . Potonji uvjet slijedi iz norme (1.15) koja zahtijeva da valna funkcija iščezava kada  $x \rightarrow \infty$ . Za dovoljno velik  $x$ , valna funkcija  $\psi(x, t)$  prostorno ograničenog kondenzata mora asimptotski iščezavati. [3] Promatramo stacionarni problem. Za početni uvjet koristimo analitičko rješenje osnovnog stanja valne funkcije problema harmonijskog oscilatora

$$\phi(x, 0) = x\psi(x, 0) = \sqrt{2}xe^{-x^2/2}, \quad (3.1)$$

jer početna vrijednost  $n = 0$  daje neinteragirajući sustav u harmonijskom oscilatoru. Vrijednosti početnih i rubnih uvjeta pohranjujemo u glavni niz. Početne i rubne uvjete u kodu postavljamo funkcijama.

```

10 void RU(double complex *u, int N) { // Rubni uvjet, za svaki t
11     u[0] = 0 + 0*I;
12     u[N] = 0 + 0*I;
13 }
14
15 double complex PU(double x) { // pocetni uvjeti, za t0
16     return sqrt(2) * x * exp(-x * x * 0.5) + 0*I;
17 }

```

#### 3.1 Algoritam

Algoritam kojim rješavamo jednačbu (2.9) sastoji se od tri glavna dijela koji se ponavljaju dok rješenje ne konvergira:

1. računanje koeficijenata  $a_j$ ,  $b_j$  i  $c_j$  definiranih izrazima (2.11), (2.12) i (2.13),
2. računanje eksplicitnog dijela, odnosno  $\tilde{\Phi}^k$ , koristeći (2.20),

## 3. računanje implicitnog dijela, odnosno rješavanje sustava (2.21).

Pri računanju koeficijenta  $b_j$  treba paziti na njegovu ovisnost o  $\phi_j^k$ . Dakle, koeficijent  $b_j$  treba računati u svakoj iteraciji petlje po vremenu. Kako su elementi tridijagonalne matrice i koeficijenti eksplicitnog dijela jednaki do na konstantu, prije računanja eksplicitnog dijela računamo  $a_j$ ,  $b_j$  i  $c_j$ . Za računanje eksplicitnog dijela,  $\tilde{\Phi}^k$ , služimo se pomoćnim nizom u kojeg pohranjujemo vrijednosti  $\tilde{\Phi}^k$ .

```

99     for (i = 1; i < N; i++) { // 1 je prva ne rubna, N-1 zadnja
100         // ELEMENTI TRIDIJAGONALNE MATRICE
101         xdx = x[i] * dx;
102         x2 = x[i] * x[i];
103         ksi = x2 - 1.0 / x2 + cn * pow(cabs(u[i]),2) / x2;
104         a[i] = 0.5 / dx2 + 0.25 / xdx;
105         b[i] = I / dt - 1.0 / dx2 - 0.5 * ksi;
106         c[i] = 0.5 / dx2 - 0.25 / xdx;
107         if (i == 1) a[i] = 0;
108         if (i == N - 1) c[i] = 0;
109         // 1. EKSPPLICITNI DIO
110         up[i] = -a[i] * u[i-1] + (2 * I / dt - b[i]) * u[i] - c[i] * u[i←
            +1];
111     }

```

Za računanje implicitnog dijela potrebno je riješiti tridijagonalni sustav (2.21), što se lako postigne Thomasovim algoritmom [4]. Thomasov algoritam implementiran je u funkciji tridijagonal(), kao što je prikazano dolje.

```

32 void tridijagonal(double *a, double complex *b, double *c, double complex ←
    *d, double complex *x, int N) {
33     int i;
34     double complex cn[N], dn[N];
35
36     cn[1] = c[1] / b[1];
37     dn[1] = d[1] / b[1];
38     for (i = 2; i < N - 1; i++) {
39         cn[i] = c[i] / (b[i] - a[i] * cn[i-1]);
40         dn[i] = (d[i] - a[i] * dn[i-1]) / (b[i] - a[i] * cn[i-1]);
41     }
42     dn[N-1] = (d[N-1] - a[N-1] * dn[N-1]) / (b[N-1] - a[N-1] * cn[N-1]);
43
44     x[N-1] = dn[N-1];
45     for (i = N - 2; i > 0; i--)
46         x[i] = dn[i] - cn[i] * x[i+1];
47 }

```

Ispis u datoteku vrši se posebnom funkcijom koja računa  $\psi(x, t)$  po relaciji (1.10). Obzirom

da su valne funkcije kompleksne, ispisujemo apsolutne vrijednosti, kao i  $\text{Re}(\psi(x,t))$  i  $\text{Im}(\psi(x,t))$ . Također računamo i ispisujemo korisne veličine poput norme i srednje vrijednosti kvadrata položaja  $\langle x^2 \rangle$ .

```

49 double ispis(FILE *dat, double *t, double *x, double complex *u, double *cn,
50   double *dx, int *N, double *S, double *x2, double *psi0) {
51   int i, j, Ni;
52   double psi, re, im;
53   *S=0.0; *x2=0.0;
54   psi = 0.025 / (*dx); // ispis nakon x-pomaka 0.025
55   Ni = psi;
56
57   for (i = 0; i <= *N; i++) {
58     if(i==0) j=1; // izbjegni dijeljenje s 0
59     else j=i;
60     psi = cabs(u[j])/x[j];
61     if(i==1) *psi0 = psi*sqrt(n);
62     re = creal(u[j])/x[j];
63     im = cimag(u[j])/x[j];
64     *S = *S + x[j] * psi * psi * (*dx);
65     *x2 = *x2 + pow(x[j],3) * psi * psi * (*dx);
66     if(i % Ni == 0)
67       fprintf(dat, "%lf\t%lf\t%e\t%e\t%e\t%lf\t%lf\t%lf\n", *t, x[i], ←
68         psi, re, im, *cn, *S, *x2);
69   }
70   fprintf(dat, "\n\n");

```

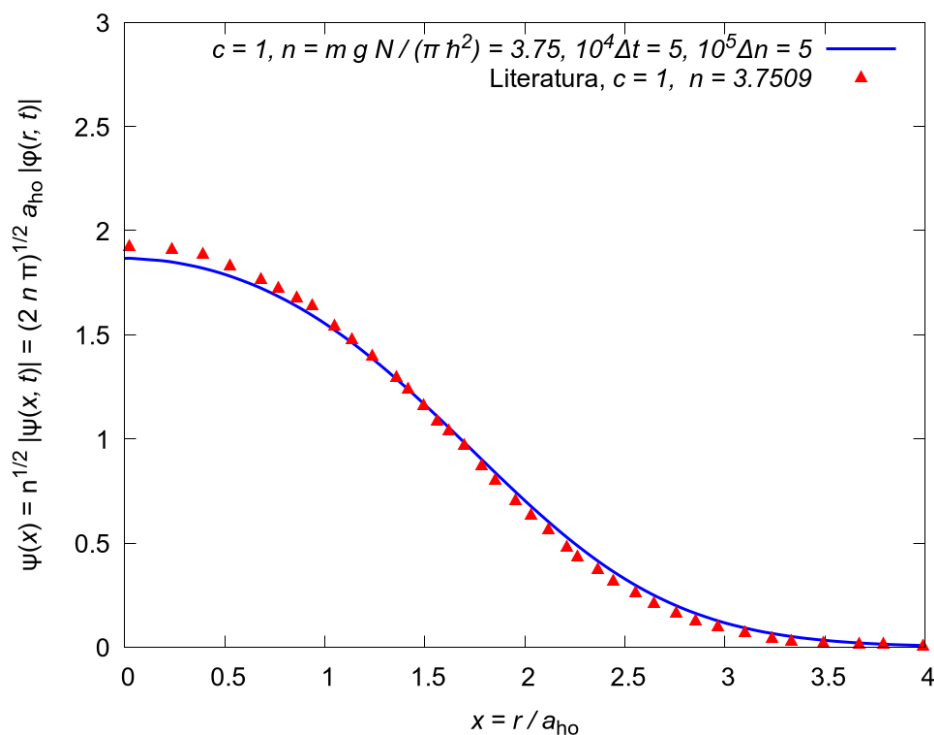
Kod je napisan u jeziku C koristeći biblioteku `<complex.h>` za rad s kompleksnim brojevima i funkcijama, dok su grafovi crtani u programu `gnuplot`. Cijeli kod priložen je u dodatku A, a primjer `gnuplot` skripte nalazi se u dodatku B.

## 4 Rješenja

Rješenje stacionarnog problema podrazumijeva pronalazak valne funkcije  $\psi(x, t) \equiv \sqrt{n}\psi(x)$  u promatranoj domeni. Rješenja tražimo postupkom opisanom u prethodnoj sekciji. Nakon što se postigne željeni  $n$ , rješenje se propagira u vremenu, dok se ne postigne stabilno rješenje. Jednačica (1.7) je nelinearna, pa je osjetljiva na izbor koraka  $\Delta t$  i  $\Delta n$ . Veća se preciznost postiže sa manjim vrijednostima  $\Delta t$  i  $\Delta n$ , ali je i dulje vrijeme izvršavanja. Ako je  $\Delta n$  prevelik, dolazi do gubitka stabilnosti, te konačno rješenje nema fizikalnog smisla.

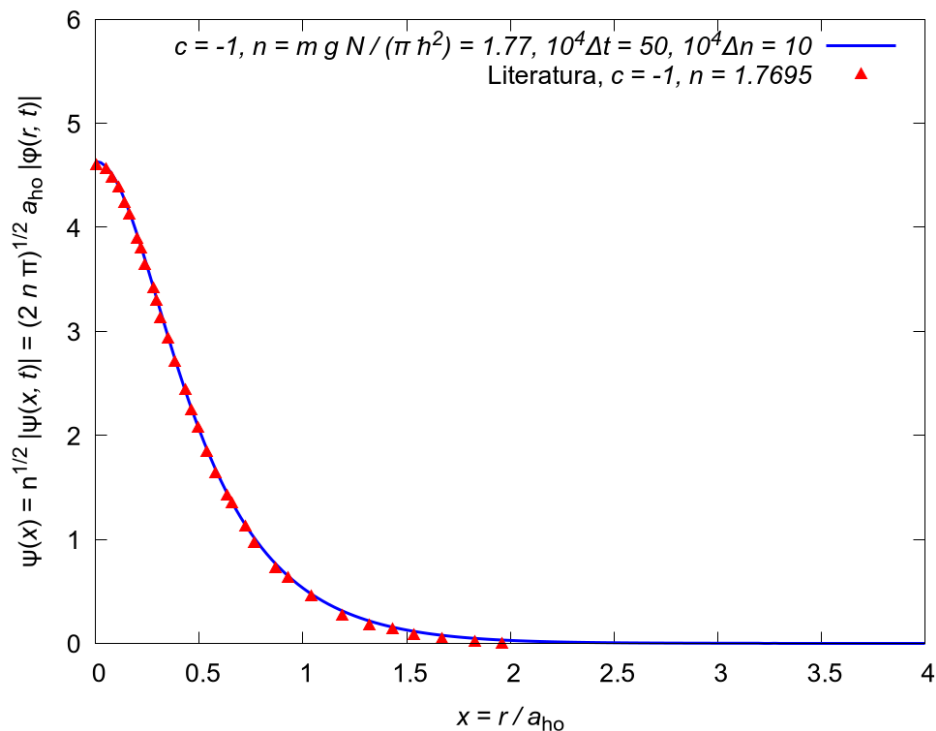
Rješenja su tražena na prostornom intervalu  $x \in [0, 8]$ , sa korakom  $\Delta x = 0.001$  i na vremenskom intervalu  $t \in [0, 100]$ . Nakon doseganja željene vrijednosti  $n$ , rješenja bi brzo konvergirala. Konvergenciju je lakše postići za privlačnu interakciju.

Na slikama 1 i 2 grafički su prikazana rješenja za slučaj odbojne i privlačne interakcije te njihova usporedba s vrijednostima iz literature [3]. Vidljivo je iznimno slaganje s Adhikarijevim rješenjem. Za potrebe usporedbe dobiveno rješenje  $\psi(x, t)$  skalira se faktorom  $\sqrt{n}$ .



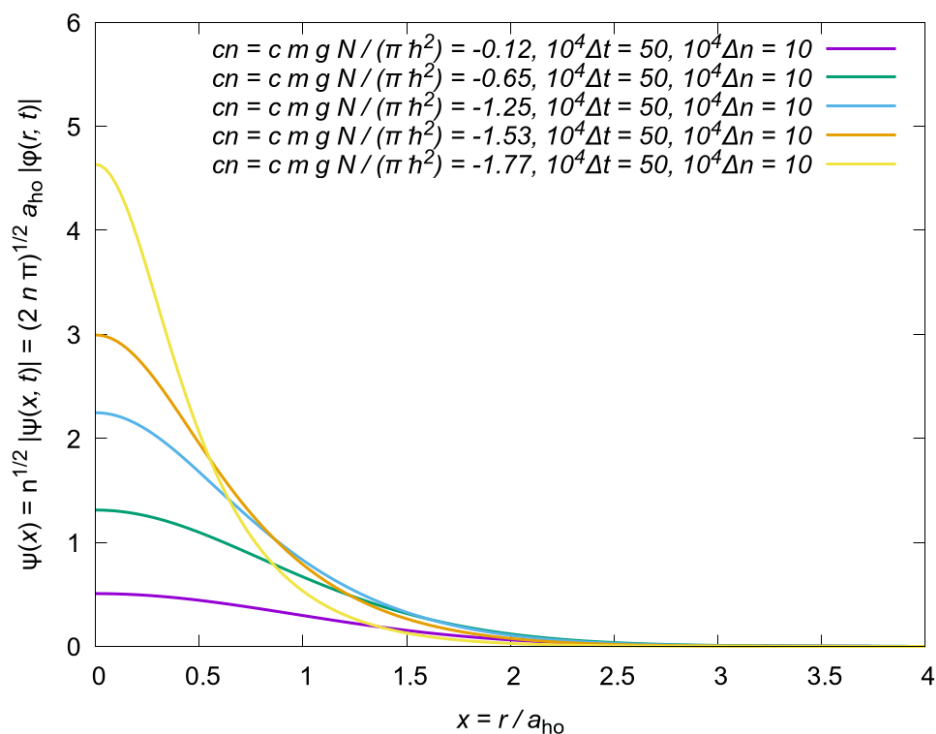
**Slika 1:** Grafički prikaz rješenja za slučaj odbojne interakcije i usporedba s literaturom [3].  $n = 3.75$

Slika 3 prikazuje nekoliko rješenja kada je međuatomska interakcija privlačna, te se jasno vidi da valna funkcija ima šiljak u ishodištu, što odgovara središtu zamke. Dakle, bozoni se skupljaju u ishodištu, što je i očekivano. Što je  $n$  veći to je interakcija jača i shodno tomu šiljak je viši. Slika 4 prikazuje rješenja za slučaj odbojne interakcije te je nagib u blizini ishodišta znatno manji u odnosu na privlačni slučaj. Tu "razmazanost" valne funkcije očekujemo, s obzirom na



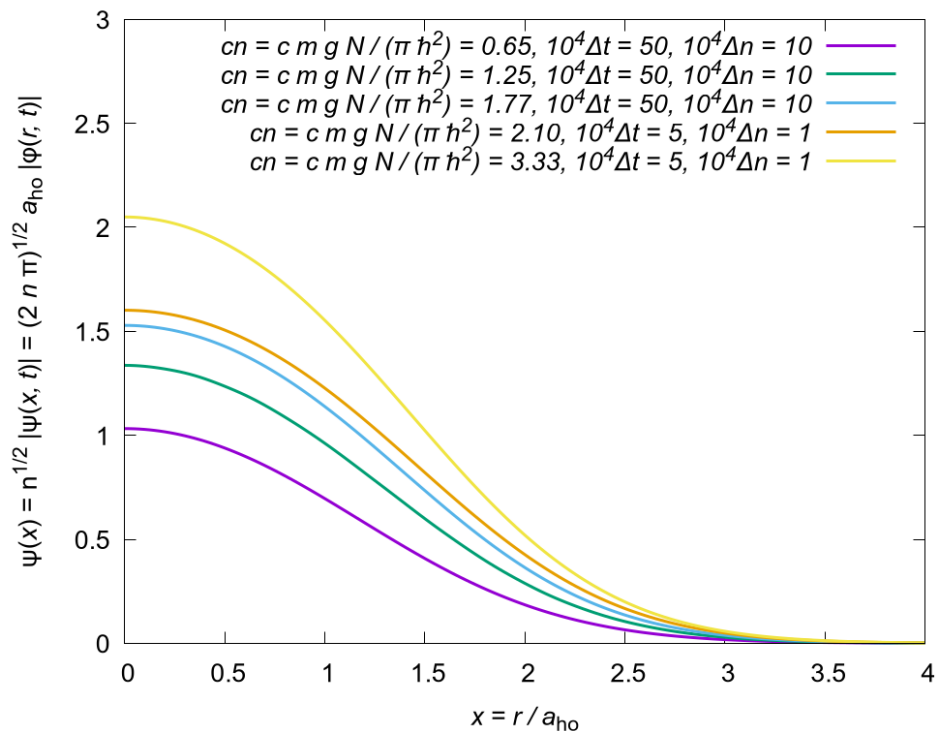
**Slika 2:** Grafički prikaz rješenja za slučaj privlačne interakcije i usporedba s literaturom [3].  $n = 1.77$

to da se bozoni odbijaju. Različito je ponašanje uočljivije kada usporedimo oba slučaja za isti  $n$ , što je prikazano na slici 5.

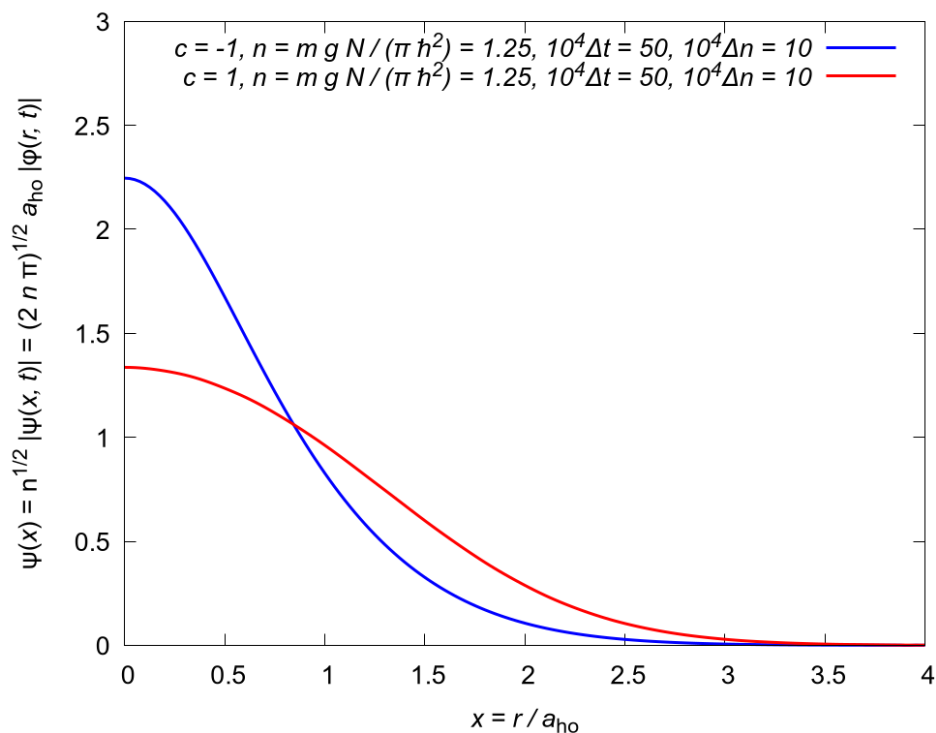


**Slika 3:** Grafički prikaz različitih rješenja za slučaj privlačne interakcije.





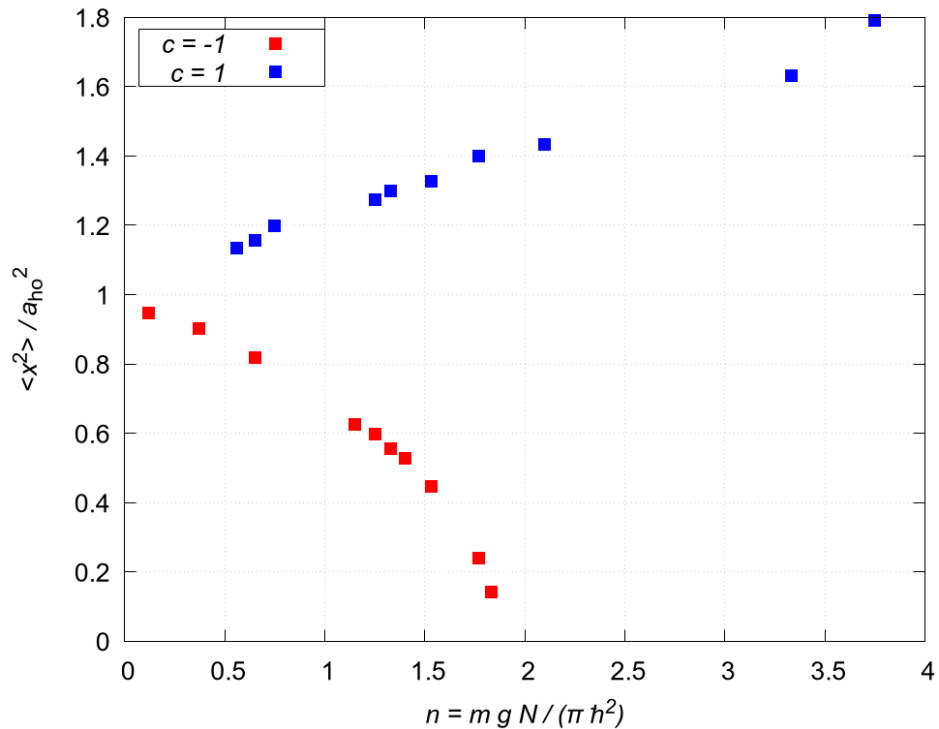
Slika 4: Grafički prikaz različitih rješenja za slučaj odbojne interakcije.



Slika 5: Grafički prikaz rješenja za slučaj odbojne ( $c = 1$ ) i privlačne ( $c = -1$ ) interakcije.  $n = 1.25$

Grafički prikaz ovisnosti srednje vrijednosti kvadrata položaja  $\langle x^2 \rangle$  o  $n$  nalazi se na slici 6. Povećanjem  $n$  interakcija je jača, pa bi za odbojni slučaj  $\langle x^2 \rangle$  trebao rasti s  $n$ , dok bi u

privlačnom trebao padati, što i vidimo na slici.



**Slika 6:** Grafički prikaz srednje vrijednosti kvadrata položaja  $\langle x^2 \rangle$  za odbojnu ( $c = 1$ ) i privlačnu ( $c = -1$ ) interakciju u ovisnosti o reduciranom broju bozona  $n$ .

Lakše je postići konvergenciju za male vrijednosti  $n$ , naročito u privlačnom slučaju. Bolja se konvergencija mogla postići odabirom manjih koraka  $\Delta t$  i  $\Delta n$ , ali tada znatno raste vrijeme izvršavanja koda. U privlačnom slučaju koncentracija čestica u središtu veća je za jaču interakciju (slika 3). Kinetička energija održava stabilnost kapljice kondenzata. Za velike vrijednosti  $n$  energija interakcije je prevelika pa GPE za te vrijednosti nema rješenja [3].

## 5 Zaključak

Gross-Pitaevskiijeva jednađba je nelinearna jednađba te je analitički nerješiva, stoga smo je riješili numerički Crank-Nicolsonovom shemom. Jednađba je riješena za stacionarno stanje Bose-Einsteinovog kondenzata u magnetskoj zamci. Iz izraza za numeričke derivacije dobivena je matrična jednađba i napisan algoritam koji propagira valnu funkciju u vremenu. Prilikom izvršavanja broj čestica postupno je povećavan, kako bi se izbjegla nestabilnost koju nelinearni član uvodi. Nakon doseganja konačnog broja čestica, funkcija je propagirana u vremenu dok nije postignuto konačno rješenje.

Kada je interakcija privlačna, bozoni se grupiraju oko središta zamke gdje je dobiven šiljak valne funkcije. Srednji kvadrat radijusa se smanjuje s povećanjem broja bozona, što je u skladu s očekivanjima, jer je interakcija proporcionalna koncentraciji. Za odbojnu interakciju, valna je funkcija šira i maksimum u ishodištu je niži. Srednji kvadrat radijusa raste s povećanjem broja bozona. Slaganje rješenja s literaturom [3] je vrlo dobro.

Preciznost je ograničena koracima  $\Delta t$  i  $\Delta n$ . Zbog prirode metode postoje oscilacije rješenja u vremenu koje su jako male za rješenje koje konvergira, a za funkcije za koje nije postignuta konvergencija su vrlo uočljive. Postoje numeričke greške koje se javljaju pri svakom numeričkom rješavanju problema. Birane vrijednosti  $\Delta x$ ,  $\Delta t$  i  $\Delta n$  su reda veličine  $10^{-3}$  -  $10^{-5}$ , čime se postiže dobra preciznost.

## 6 Literatura

- [1] Pathria, R. K., & Beale, P. D. (2011). *Statistical Mechanics*, treće izdanje, Elsevier publishing.
- [2] Pethick, C. J., & Smith, H. (2008). *Bose–Einstein condensation in dilute gases*. Cambridge university press.
- [3] Adhikari, S. K. (2000). Numerical study of the spherically symmetric Gross-Pitaevskii equation in two space dimensions. *Physical Review E*, 62(2), 2937.
- [4] Hjorth-Jensen, M. (2015). *Computational physics*. Lecture Notes Fall 2015.

## A Priloženi kod

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4 #include <complex.h>
5
6 #define c11 1.0
7 #define n 3.75
8 #define deltan 0.0001
9
10 void RU(double complex *u, int N) { // Rubni uvjet, za svaki t
11     u[0] = 0 + 0*I;
12     u[N] = 0 + 0*I;
13 }
14
15 double complex PU(double x) { // pocetni uvjeti, za t0
16     return sqrt(2) * x * exp(-x * x * 0.5) + 0*I;
17 }
18
19 FILE* fprovjera(FILE *dat) {
20     float cn = c11*n;
21     char str[50];
22     printf("ispis: GP_cn%+4.2f.txt\n", cn);
23     sprintf(str, "GP_cn%+4.2f.txt", cn);
24     dat = fopen(str, "w");
25     if (dat == NULL) {
26         printf("Greska pri otvaranju datoteke.\n");
27         exit(EXIT_FAILURE);
28     }
29     return dat;
30 }
31
32 void tridijagonal(double *a, double complex *b, double *c, double complex *d, double complex *x, int N) {
33     int i;
34     double complex cn[N], dn[N];
35
36     cn[1] = c[1] / b[1];
37     dn[1] = d[1] / b[1];
38     for (i = 2; i < N - 1; i++) {
39         cn[i] = c[i] / (b[i] - a[i] * cn[i-1]);
40         dn[i] = (d[i] - a[i] * dn[i-1]) / (b[i] - a[i] * cn[i-1]);
41     }
42     dn[N-1] = (d[N-1] - a[N-1] * dn[N-1]) / (b[N-1] - a[N-1] * cn[N-1]);
43 }

```

```

44     x[N-1] = dn[N-1];
45     for (i = N - 2; i > 0; i--)
46         x[i] = dn[i] - cn[i] * x[i+1];
47 }
48
49 double ispis(FILE *dat, double *t, double *x, double complex *u, double *cn,
50             double *dx, int *N, double *S, double *x2, double *psi0) {
51     int i, j, Ni;
52     double psi, re, im;
53
54     *S=0.0; *x2=0.0;
55     psi = 0.025 / (*dx); // ispis nakon x-pomaka 0.025
56     Ni = psi;
57
58     for (i = 0; i <= *N; i++) {
59         if(i==0) j=1; // izbjegni dijeljenje s 0
60         else j=i;
61         psi = cabs(u[j])/x[j];
62         if(i==1) *psi0 = psi*sqrt(n);
63         re = creal(u[j])/x[j];
64         im = cimag(u[j])/x[j];
65         *S = *S + x[j] * psi * psi * (*dx);
66         *x2 = *x2 + pow(x[j],3) * psi * psi * (*dx);
67         if(i % Ni == 0)
68             fprintf(dat, "%lf\t%lf\t%e\t%e\t%e\t%lf\t%lf\t%lf\n", *t, x[i],
69                 psi, re, im, *cn, *S, *x2);
70     }
71     fprintf(dat, "\n\n");
72 }
73
74 void CNGP(double x0, double xn, double dx, double t0, double tn, double dt,
75          FILE* dat) {
76     int N = (xn - x0) / dx;
77     int M = (tn - t0) / dt;
78     int i, j, Nj = 200; // ispis nakon svakih Nj vremenskih koraka
79
80     double a[N], c[N];
81     double complex b[N];
82
83     double complex u[N+1], up[N+1], ksi; // pomocno
84     double t=t0, x[N+1], x2, xdx, dx2 = dx * dx, S, psi0, cn = 0.0; // c11*n
85     ;
86
87     // pocetni uvjeti
88     for (i = 0; i <= N; i++) {
89         x[i] = x0 + i * dx;
90         u[i] = PU(x[i]);

```

```

87     }
88     // rubni uvjeti
89     RU(u, N);
90     RU(up, N);
91     ispis(dat, &t, x, u, &cn, &dx, &N, &S, &x2, &psi0);
92
93     for (j = 1; j <= M; j++) { // petlja po vremenu
94         t = t0 + j * dt;
95         // POSTEPENO JACANJE INTERAKCIJE
96         if (c11 < 0){
97             if (-cn < n-1E-10) cn -= deltan;}
98         else if (cn < n-1E-10) cn += deltan;
99         for (i = 1; i < N; i++) { // 1 je prva ne rubna, N-1 zadnja
100            // ELEMENTI TRIDIJAGONALNE MATRICE
101            xdx = x[i] * dx;
102            x2 = x[i] * x[i];
103            ksi = x2 - 1.0 / x2 + cn * pow(cabs(u[i]),2) / x2;
104            a[i] = 0.5 / dx2 + 0.25 / xdx;
105            b[i] = I / dt - 1.0 / dx2 - 0.5 * ksi;
106            c[i] = 0.5 / dx2 - 0.25 / xdx;
107            if (i == 1) a[i] = 0;
108            if (i == N - 1) c[i] = 0;
109            // 1. EKSPlicitNI DIO
110            up[i] = -a[i] * u[i-1] + (2 * I / dt - b[i]) * u[i] - c[i] * u[i-1];
111        }
112
113        // 2. IMPLICITNI DIO
114        tridijagonal(a, b, c, up, u, N); // a, b, c, d=up, x=u
115
116        // POHRANA I INFO
117        if (j%Nj == 0){ //nakon Nj vremenskih koraka
118            ispis(dat, &t, x, u, &cn, &dx, &N, &S, &x2, &psi0);
119            // smanji informiranje nakon postignutog cn
120            if (fabs(cn) < n-1E-10)
121                printf("%d. cn=%.3lf, norma=%.3lf, <x2>=%.3lf, Psi(0)=%.3lf \n←",
122                    j, cn, S, x2, psi0);
123            else if (j%(10*Nj) == 0)
124                printf("%d. cn=%.3lf, norma=%.3lf, <x2>=%.3lf, Psi(0)=%.3lf \n←",
125                    j, cn, S, x2, psi0);
126        }
127    }
128
129 int main() {
130     double dx = 0.001;
131     double x0 = 0.0; // pocetna tocka

```

```
131  double xn = 8.0;    // konacna tocka
132
133  double dt = 0.0005; // vremenski korak
134  double t0 = 0.0;    // pocetno vrijeme
135  double tn = 100.0; // konacno vrijeme
136
137  FILE* dat;
138
139  dat=fprovjera(dat);
140  CNGP(x0, xn, dx, t0, tn, dt, dat);
141  fclose(dat);
142  return 0;
143 }
```



## B Priložena gnuplot skripta

```

1 # RESET POSTAVKI
2 unset multiplot
3 reset
4 set encoding utf8
5
6 # POSTAVKE GRAFA
7 set xrange [0:4]
8 set yrange [0:3]
9 set style line 1 lw 3 lc rgb "blue"
10
11 # NAZIVI
12 set xlabel '{/:Italic x = r / a}_{ho}'
13 set ylabel ' $\psi({/:Italic x}) = n^{1/2} |\psi({/:Italic x, t})| = (2 {/:Italic \leftarrow$ 
14      $n) \pi^{1/2} {/:Italic a}_{ho} |\varpi({/:Italic r, t})|'$ 
15
16 # POSTAVKE
17 set term pngcairo \
18     background "#ffffff" \
19     font "arial,20pt" \
20     size 1000,800
21
22 # PARAMETRI
23 c=+1.00
24 n=3.750000
25 cn=c*n
26 dn=0.000050
27 dt=0.000500
28 info = sprintf("{/:Italic c = %i, n = m g N / (π @^{-}{h}^2) = %4.2f, 10^4↔
29     Δt = %.0f, 10^5Δn = %.0f}", \
30     c, n, dt*1E4, dn*1E5)
31 data1 = sprintf("GP_cn%+5.2f.txt", cn)
32 data2 = sprintf("Adhikari.txt")
33 out = sprintf("Graf_GP_cn%+5.2f+literatura.png", cn)
34
35 set output out
36
37 plot data1 i 2000 u 2:($3*sqrt(n)) w l ls 1 ti info,\
38     data2 i 2 u 1:2 w p pt 9 ps 2 lc "red" ti 'Literatura, {/:Italic c = ↔
39     1, n = 3.7509}'
40
41 unset output

```

```
42 # RESET POSTAVKI
43 set terminal GNUTERM
```