Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **14**, No. 6, pp. 1302-1332

DOI: 10.4208/aamm.OA-2021-0189
December 2022

Sinc Collocation Numerical Methods for Solving Two-Dimensional Gross-Pitaevskii Equations with Non-Homogeneous Dirichlet Boundary Conditions

Shengnan Kang[1], Kenzu Abdella[2,*], Macro Pollanen[2], Shuhua Zhang[3,1] and Liang Wang[1]

[1] *Coordinated Innovation Center for Computable Modeling in Management Science, Tianjin University of Finance and Economics, Tianjin 300222, China*
[2] *Department of Mathematics, Trent University, Peterborough, Ontario K9L 7B8, Canada*
[3] *Coordinated Innovation Center for Computable Modeling in Management Science, Yango University, Fujian 350015, China*

Received 1 June 2021; Accepted (in revised version) 1 December 2021

Abstract. This paper presents the numerical solution of the time-dependent Gross-Pitaevskii Equation describing the movement of quantum mechanics particles under non-homogeneous boundary conditions. Due to their inherent non-linearity, the equation generally can not be solved analytically. Instead, a highly accurate approximation to the solutions defined in a finite domain is proposed, using the Crank-Nicolson difference method and Sinc Collocation numerical methods to discretize separately in time and space. Two Sinc numerical approaches, involving the Sinc Collocation Method (SCM) and the Sinc Derivative Collocation Method (SDCM), are easy to implement. The results demonstrate that Sinc numerical methods are highly efficient and yield accurate results. Mainly, the SDCM decays errors faster than the SCM. Future work suggests that the SDCM can be extensively applied to approximate solutions under other boundary conditions to demonstrate its broad applicability further.

AMS subject classifications: 65N35, 81Q05

Key words: Quantum mechanics, spectral method, time-dependent partial differential equation, boundary value problem.

1 Introduction

1.1 Gross-Pitaevskii equation

In modern physics, the Gross-Pitaevskii Equation (GPE) as an important partial differential equation (PDE) has earned central importance due to its applications. For instance,

*Corresponding author.
Email: kabdella@trentu.ca (K. Abdella)

it is remarkable to describe the phenomenon of a Bose-Einstein Condensate [1,2] representing the phase transition where a macroscopic number of particles all go into the same quantum state while cooling them below the critical transition temperature [3]. In this paper, we consider the two-dimensional generalized Gross-Pitaevskii Equation given by:

$$iu_t+\alpha u_{xx}+\alpha u_{yy}+f(x,y)u+\rho|u|^2u=0, \quad (x,y)\in\Omega, \quad t\geq 0, \tag{1.1}$$

with the initial condition and Dirichlet boundary conditions:

$$\begin{aligned}&u(x,y,t=0)=g(x,y,0),\\&u(x,y,t)=\Theta(x,y,t), \quad (x,y)\in\Gamma, \quad t>0,\end{aligned}$$

where $i=\sqrt{-1}$ is the imaginary unit, α is a real constant, t is the time variable with $t\geq 0$, $(x,y)\in\mathbb{R}^2$, Ω denotes a bounded and open domain in $\mathbb{R}^2$, Γ is the simple closed curve bounding the region Ω, $u=u(x,y,t)$ is a complex-valued wave function, $|u|^2$ is named as the atomic density, $|u|^2u$ is the so-called cubic nonlinearity, and ρ is a given dimensionless constant describing the strength of interaction (negative for the repulsive or defocusing interaction and positive for the attractive or focusing interaction). In this case, the interactions are weak enough that the predictions made by this equation are very reliable [4]. The external potential function $f(x,y)$, the function $g(x,y,t)$ and $\Theta(x,y,t)$ are all known real-valued functions. The special case of $\rho=0$ in Eq. (1.1) corresponds to the well-known Schrödinger Equation (SCE). Also, when $f(x,y)=0$, it becomes the standard Non-Linear Schrödinger Equation (NLSE). Therefore, the GPE incorporates the properties of the SCE and the NLSE.

1.2 Literature review

Even though Gross, E. P. [5] and Pitaevskii, L. P. [6] separately presented the GPE in 1961, the academic research on Eq. (1.1) has not been paused, especially in its numerical solutions. That results from the truth that analytical solutions are hard to figure out. Even if the GPE given by Eq. (1.1) has an exact resolution, it usually exhibits large temporal and spatial gradients, including solition solutions, breather solutions, and bound states with multiple modes [7]. Thus, employing efficient and effective numerical methods is vital to identify solutions' qualitative and quantitative characteristics.

An extensive study in theoretical analyses and numerical simulations for the GPE has existed in the literature. As one of the typical techniques, Eq. (1.1) is solved in two time-splitting steps. The process relies on identical small time steps and eliminates the nonlinear mechanisms influenced in the spatial domain for numerical solutions [2,8,9]. Due to the limitation of the technique, wide-ranging researches are intended to discretize the GPE in time and space, respectively, for acquiring numerical solutions. With this regard, the Finite Difference Method (FDM) is a sweeping way to discretize in the temporal dimension. Especially, the Crank-Nicolson Method (CNM) is a prominent numerical treatment with the second-order accuracy for discretizing the time derivative [10]. In fact, one needs to solve a fully nonlinear system during every step, and thus it might be super time-consuming. A straightforward way adopted by a few researchers is to estimate the atomic density $|u^{(n+1)}|^2$ as $|u^n|^2$, which is the latest available approximation for the atomic density [11, 12], for saving the consumption cost. This study will follow the assumption and is contributed to analyzing the stability condition of the iterative time discretization.

Furthermore, the extant studies have exhibited various typically numerical approaches for spatial discretization, including the FDM [13, 14] and the Finite Element Method (FEM) [15–17] as the classical mesh-based methods. However, since they are not appropriate for all types of domains [18], some additional truncated treatments, such as absorbing boundary conditions and the perfectly matched layer, are applied to cope with the infinite domain [19]. As a result, Meshless (or Meshfree) Methods [20, 21] and Spectral Methods [22], dedicated to approximating unknown variables' derivatives by a linear combination of function values at interpolating nodes, have become prevalent in numerical investigations due to the uncomplicated treatment for the unlimited domain.

In particular, among Meshless Methods and Spectral Methods, the collocation technique is emblematic for solving the GPE, such as Spectral Collocation Methods [7, 23, 24] and Meshfree Collocation Methods [11, 18, 25–27]. Indeed, the two approaches have distinct advantages. Meshless Collation Methods are suitable for large-scale computation with high computational efficiency [28–30]. On the other hand, spectral Collocation Methods with outstanding error properties potentially involve Fourier Series expansions [31]. For instance, Dehghan and Emami-Naeini [23] demonstrated that the Sinc Collocation Method (SCM) is successfully implemented to obtain accurate solutions characterized by exponentially decaying errors. The SCM also has the well-known advantage of being highly efficient and adaptable in dealing with singularity problems. Moreover, the Sinc Derivative Collocation Method (SDCM) proposed in [32–34] inherits the advantages from the SCM. It processes the merits of solving PDEs with different boundary conditions, less sensitive to numerical errors than the SCM. Thus, this study is prone to fulfil the gap of solving the GPE using the SCM and the SDCM.

1.3 Objectives and outline

This study approximates the wave solution under the Dirichlet boundary conditions by discretizing the two-dimensional GPE in time through the CNM and in space via Sinc numerical methods. Especially, the SDCM based on the development of the SCM with a unique boundary treatment is proposed in this study to compare with the numerical results obtained by the SCM and other methods from recent studies. Furthermore, this research is devoted to investing the effect of two transformations onto the Sinc numerical methods, which convert the infinite spatial domain into the finite field.

In the following content, a brief introduction concerning the time discretizing scheme is demonstrated in Section 2. Next, we present the full discretizing system in time and space through two Sinc numerical methods for the two-dimensional GPE under non-homogeneous Dirichlet boundary conditions in Section 3. Then, the accuracy and efficiency of the proposed techniques are demonstrated by several examples in Section 4. Finally, a summary and highlights of this work are presented in Section 5.

2 Discretization in time by the Crank–Nicolson method

We first briefly introduce some functional spaces endowed with standard norms and inner products that will be used hereafter.

Definition 2.1. *The function $f(\mathbf{x})$ defined on Ω at most everywhere satisfying*

$$\|f\|_p \equiv \left\{ \int_\Omega |f(\mathbf{x})|^p d\mathbf{x} \right\}^{1/p} < \infty, \quad p \geq 1,$$

form a linear space denoted by $L^p(\Omega)$ [35]. The non-negative quantity in the above equation is called the norm of f. Furthermore, the inner product on $L^2(\Omega)$ is defined as:

$$(u,v) = \int_\Omega u(\mathbf{x})\overline{v(\mathbf{x})} d\mathbf{x}.$$

Theorem 2.1. *If f is in $L^p(\Omega)$, and g is in $L^q(\Omega)$, where $p+q=pq$, the Hölder inequality reads [35, p. 10],*

$$\int_\Omega |f(\mathbf{x})g(\mathbf{x})| d\mathbf{x} \leq \|f\|_p \|g\|_q. \tag{2.1}$$

In particular, if $p=q=2$, the inequality is called the Cauchy-Schwarz inequality

$$|(f,g)| \leq \|f\|_2 \|g\|_2. \tag{2.2}$$

2.1 Time discretization system

The Crank-Nicolson method is the average of the forward Euler method and the backward Euler method, which discretizes the time derivative in Eq. (1.1). To be specific, the time interval $[0,T]$ is divided into N equal sub-intervals, with the time step size $dt = \frac{T}{N}$ and $t^n = n \times dt$, for $n = 0,1,\cdots,N$, where n is the non-negative real number. Then the time discretization treatment for Eq. (1.1) can be expressed as follows

$$\begin{aligned} -i\frac{u^{n+1}-u^n}{dt} =& \frac{1}{2}(\alpha u_{xx}^{n+1} + \alpha u_{yy}^{n+1} + f(x,y)u^{n+1} + \rho\vartheta(u^{n+1})) \\ &+ \frac{1}{2}(\alpha u_{xx}^n + \alpha u_{yy}^n + f(x,y)u^n + \rho\vartheta(u^n)) + iR^{n+1}, \end{aligned} \tag{2.3}$$

where u^n is the exact solution for $u(x,y,t=n)$, $\vartheta(u^{n+1}) = |u^{n+1}|^2 u^{n+1}$, and R^{n+1} is the truncation error with the boundary $||R^{n+1}||_2 < K_0 dt^2$ (K_0 is a positive constant) [27,36].

To minimize computational costs, we assume that the atomic density value at time $t = t^{n+1}$ is substituted by its latest value, i.e., $|u^{n+1}|^2 \approx |u^n|^2$ [11,12] if the time step dt is small enough. In addition, the system used to solve the wave solution is omitting the truncation error part in Eq. (2.3) in practice. Then, based on the two mentioned aspects, rearranging the scheme Eq. (2.3) yields a linear system:

$$\begin{aligned}&iU^{n+1}+\lambda(\alpha U^{n+1}_{xx}+\alpha U^{n+1}_{yy}+f(x,y)U^{n+1}+\rho g(U^n)U^{n+1})\\=&iU^n-\lambda(\alpha U^n_{xx}+\alpha U^n_{yy}+f(x,y)U^n+\rho\vartheta(U^n)),\end{aligned}\tag{2.4}$$

or

$$\begin{aligned}&U^{n+1}-i\lambda(\alpha U^{n+1}_{xx}+\alpha U^{n+1}_{yy}+f(x,y)U^{n+1}+\rho g(U^n)U^{n+1})\\=&U^n+i\lambda(\alpha U^n_{xx}+\alpha U^n_{yy}+f(x,y)U^n+\rho\vartheta(U^n)),\end{aligned}\tag{2.5}$$

where $\lambda=\frac{dt}{2}$, U^n is the exact solution for Eq. (2.4) or Eq. (2.5), and $g(U^n)=|U^n|^2$. Accordingly, the linear time discretizing system is constructed, where the initial condition is employed to obtain the approximated wave solution at time t^1. Through iterating the system N times, the approximated wave solution at time T is generated.

2.2 The stability analysis

We examine the stability for the proposed time-discrete scheme Eq. (2.4) or Eq. (2.5) to ensure its accuracy and efficiency even for long-term evolution problems.

Lemma 2.1. *Let the solution $U^n(\mathbf{x})\in L^2(\Omega)$, and we assume that the solution of Eq.* (1.1) *is analytical over $\overline{\Omega}\times[0,T]$, where $\overline{\Omega}$ is a closed domain including the boundary points in Ω. Suppose that $g(\cdot)$ is effortlessly satisfied the following Lipschitz condition, i.e.,*

$$|g(v_1)-g(v_2)|\le\ell|v_1-v_2|,\quad\forall v_1,v_2\in L^2(\Omega),\tag{2.6}$$

where ℓ is a positive constant, then the cubic operator in Eq. (1.1) *is bounded by:*

$$|g(v_1)v_3-g(v_2)v_4|\le K_1|v_1-v_2|+K_2|v_3-v_4|,\quad\forall v_3,v_4\in L^2(\Omega),\tag{2.7}$$

where $K_1=\ell|v_4|$ and $K_2=|g(v_1)|$ are positive constants. Especially, if $v_1=v_3$ and $v_2=v_4$, then the cubic operator is conveniently inclined to:

$$|g(v_1)v_3-g(v_2)v_4|=|\vartheta(v_1)-\vartheta(v_2)|\le(K_1+K_2)|v_1-v_2|\le K_3|v_1-v_2|,\tag{2.8}$$

where $K_3=K_1+K_2$.

Proof. To gain Eq. (2.7), making the following identity transformation and applying the triangle inequality under the condition Eq. (2.6):

$$\begin{aligned}|g(v_1)v_3-g(v_2)v_4|=&|g(v_1)v_3-g(v_1)v_4+g(v_1)v_4-g(v_2)v_4|\\\le&|g(v_1)(v_3-v_4)|+|v_4(g(v_1)-g(v_2))|\\\le&|g(v_1)||(v_3-v_4)|+|v_4||(g(v_1)-g(v_2)|\\\le&|g(v_1)||v_3-v_4|+\ell|v_4||v_1-v_2|.\end{aligned}$$

After all, let $K_1=\ell|v_4|$ and $K_2=|g(v_1)|$ generate Eq. (2.7). In addition, it is obvious to acquire Eq. (2.8) by given $v_1=v_3$ and $v_2=v_4$. □

Theorem 2.2. *Let the solution $U^n(\mathbf{x})\in L^2(\Omega)$, and we assume that the solution of Eq.* (1.1) *is analytical over $\overline{\Omega}\times[0,T]$, then the scheme defined by Eq.* (2.4) *or Eq.* (2.5) *is conditionally stable in $L^2(\Omega)$.*

Proof. The proof procedure is referred from [27]. The roundoff error $e^{n+1}(\mathbf{x})$ is as the following simplified pattern:

$$\begin{aligned}&e^{n+1}-i\lambda(\alpha\Delta e^{n+1}+f(\mathbf{x})e^{n+1}+\rho(g(U^n)U^{n+1}-g(\widehat{U}^n)\widehat{U}^{n+1}))\\=&e^n+i\lambda(\alpha\Delta e^n+f(\mathbf{x})e^n+\rho(\vartheta(U^n)-\vartheta(\widehat{U}^n))),\end{aligned}\tag{2.9}$$

where $e^{n+1}=U^{n+1}-\widehat{U}^{n+1}$ whereas $\widehat{U}^{n+1}$ is the approximate solution of Eq. (2.4) or Eq. (2.5). Multiplying Eq. (2.9) by e^{n+1} and integrating on Ω, we obtain:

$$\begin{aligned}&(e^{n+1},e^{n+1})-i\lambda\alpha(\Delta e^{n+1},e^{n+1})-i\lambda(f(\mathbf{x})e^{n+1},e^{n+1})\\&\quad-i\lambda\rho(g(U^n)U^{n+1}-g(\widehat{U}^n)\widehat{U}^{n+1},e^{n+1})\\=&(e^n,e^{n+1})+i\lambda\alpha(\Delta e^n,e^{n+1})+i\lambda(f(\mathbf{x})e^n,e^{n+1})\\&\quad+i\lambda\rho(\vartheta(U^n)-\vartheta(\widehat{U}^n),e^{n+1}).\end{aligned}\tag{2.10}$$

Based on Definition 2.1 and using the Green's formula, for $e^{n+1}\in L_0^2(\Omega)=\{u\in L^2(\Omega): u|_{\partial\Omega}=0\}$, Eq. (2.10) becomes:

$$\begin{aligned}&\|e^{n+1}\|_2^2+i\lambda\alpha\|\nabla e^{n+1}\|_2^2-i\lambda(f(\mathbf{x})e^{n+1},e^{n+1})\\=&(e^n,e^{n+1})+i\lambda\alpha(\Delta e^n,e^{n+1})+i\lambda(f(\mathbf{x})e^n,e^{n+1})\\&+i\lambda\rho(\vartheta(U^n)-\vartheta(\widehat{U}^n),e^{n+1})+i\lambda\rho(g(U^n)U^{n+1}-g(\widehat{U}^n)\widehat{U}^{n+1},e^{n+1}).\end{aligned}\tag{2.11}$$

By taking the absolute value on both sides of Eq. (2.11), according to the definition of absolute value in complex numbers and the triangle inequality, we have:

$$\begin{aligned}\|e^{n+1}\|_2^2\le&|(e^n,e^{n+1})|+\lambda|\alpha||(\Delta e^n,e^{n+1})|+\lambda|(f(\mathbf{x})e^n,e^{n+1})|\\&+\lambda|\rho||(\vartheta(U^n)-\vartheta(\widehat{U}^n),e^{n+1})|+\lambda|\rho||g(U^n)U^{n+1}-g(\widehat{U}^n)\widehat{U}^{n+1},e^{n+1})|.\end{aligned}\tag{2.12}$$

By using the Caucy-Schwarz inequality Eq. (2.2) and the boundedness Eq. (2.7) and Eq. (2.8), Eq. (2.12) can be rewritten as:

$$\begin{aligned}&\|e^{n+1}\|_2-\lambda|\rho|(K_1\|e^n\|_2+K_2\|e^{n+1}\|_2)\\\le&\|e^n\|_2+\lambda(|\alpha|\|\Delta e^n\|_2+\mathcal{M}_1\|e^n\|_2+|\rho|K_3\|e^n\|_2),\end{aligned}\tag{2.13}$$

where $|f(\mathbf{x})|\le\mathcal{M}_1$. According to the maximum modulus principle [37] for analytic functions (as cited in [27]), it is assuming $|\Delta e^n|\le\mathcal{M}_2|e^n|_2$ for a positive number $\mathcal{M}_2$. Finally, if $1-\lambda|\rho|K_2>0$ (i.e., $dt<\frac{2}{|\rho|K_2}$), then we have

$$\begin{aligned}\|e^{n+1}\|_2 &\leq \frac{1+\lambda|\alpha|\mathcal{M}_2+\lambda\mathcal{M}_1+\lambda|\rho|(K_1+K_3)}{1-\lambda|\rho|K_2}\|e^n\|_2 \\ &\leq \Big(\frac{1+\lambda|\alpha|\mathcal{M}_2+\lambda\mathcal{M}_1+\lambda|\rho|(K_1+K_3)}{1-\lambda|\rho|K_2}\Big)^2\|e^{n-1}\|_2\leq\cdots \\ &\leq \Big(\frac{1+\lambda|\alpha|\mathcal{M}_2+\lambda\mathcal{M}_1+\lambda|\rho|(K_1+K_3)}{1-\lambda|\rho|K_2}\Big)^{n+1}\|e^0\|_2. \end{aligned} \tag{2.14}$$

The series

$$K_{n+1}=\Big(\frac{1+\lambda|\alpha|\mathcal{M}_2+\lambda\mathcal{M}_1+\lambda|\rho|(K_1+K_3)}{1-\lambda|\rho|K_2}\Big)^{n+1}$$

is bounded while $n\to+\infty$ proved as follows. Since $dt=\frac{T}{n+1}$, where T is a constant, then

$$\begin{aligned}\lim_{n\to+\infty}K_{n+1}&=\frac{\lim_{n\to+\infty}(1+\frac{\frac{T}{2}(|\alpha|\mathcal{M}_2+\mathcal{M}_1+|\rho|(K_1+K_3))}{n+1})^{n+1}}{\lim_{n\to+\infty}(1-\frac{\frac{T}{2}|\rho|K_2}{n+1})^{n+1}} \\ &=\frac{e^{\frac{T}{2}(|\alpha|\mathcal{M}_2+\mathcal{M}_1+|\rho|(K_1+K_3))}}{e^{-\frac{T|\rho|K_2}{2}}} \\ &=e^{\frac{T}{2}(|\alpha|\mathcal{M}_2+\mathcal{M}_1+2|\rho|(K_1+K_2))}=K_4. \end{aligned} \tag{2.15}$$

To conclude, if $dt<\frac{2}{|\rho|K_1}$, then

$$\|e^{n+1}\|_2\leq K_4\|e^0\|_2.$$

That implies the conditional stability of the scheme Eq. (2.4) or Eq. (2.5). □

3 Full discretization system

3.1 Sinc function preliminary

Definition 3.1. *The translated Sinc function defined on the real line is as a part to interpolate x by given as [38]*

$$S(j,h)(x)=sinc\Big(\frac{x-jh}{h}\Big)=\begin{cases}\dfrac{\sin(\pi(x-jh))/h}{\pi(x-jh)/h}, & if\ \ x\neq jh, \\ 1, & if\ \ x=jh,\end{cases} \tag{3.1}$$

where $j=0,\pm1,\pm2,\cdots$, $h>0$, *and* $x\in\mathbb{R}$.

Definition 3.2. *Let* f *be a function defined on* $\mathbb{R}$ *and* $h>0$, *and define the series [35, p. 22],*

$$C(j,h)(x)\equiv\sum_{j=-\infty}^{\infty}f(jh)sinc\Big(\frac{x-jh}{h}\Big). \tag{3.2}$$

Whenever the series Eq. (3.2) *converges, it is called the cardinal function of f. In practice, the finite number of terms are used in Eq.* (3.2)*, such as $j=-M_x,\ldots,N_x$. Therefore, Eq.* (3.2) *evolves into the truncated cardinal series denoted as*

$$C_{M,N}(j,h)(x)\equiv\sum_{j=-M_x}^{N_x} f(jh)sinc\Big(\frac{x-jh}{h}\Big). \tag{3.3}$$

For approximating the problem within a finite domain (i.e., $x\in[a,b]$), a conformal map is used to transform the finite domain into an infinite set [35], then the composite truncated cardinal sinc series is defined as:

$$C_{M,N}(j,h,\phi)(x)=\sum_{j=-M_x}^{N_x} f(jh)S(j,h)\circ\phi(x)\equiv\sum_{j=-M_x}^{N_x} f(jh)sinc\Big(\frac{\phi(x)-jh}{h}\Big), \tag{3.4}$$

where $\phi(x)$ can be the single exponential (SE) transformation defined as:

$$\phi_S(x)=\ln\Big(\frac{x-a}{b-x}\Big), \tag{3.5}$$

or the double exponential (DE) transformation as:

$$\phi_D(x)=\text{arcsinh}\Big(\frac{2}{\pi}\text{arctanh}\Big(\frac{2x}{b-a}+\frac{a+b}{a-b}\Big)\Big). \tag{3.6}$$

It should notice that the functions $C_{M,N}(j,h,\phi)(x)$ vanishes to 0 when $x=a$ or b. After that, the sinc interpolating nodes in one dimension are acquired from

$$x_p=\psi(ph),\quad p=-M_x,\cdots,N_x, \tag{3.7}$$

where $\phi(x)$ is the inverse function of $\psi(ph)$.

In the complex plane, the conformal map can carry the eye-shaped domain [23]:

$$D=\Big\{z=x+iy\in\mathbb{C}:\Big|\arg\Big(\frac{z-a}{b-z}\Big)\Big|<d\Big\},$$

onto the infinite strip D_d which is the region for deriving the properties of the composite truncated cardinal Sinc series: $D_d=\{w=r+iv\in\mathbb{C}:|v|<d\}$.

Definition 3.3. *Let $B(D)$ denote the class of functions analytic in D which satisfy [23]:*

$$\int_{\psi(r+L)}|f(z)dz|\to 0,\quad t\to\pm\infty,$$

where $L=\{iv:|v|<d\}$, and for a simple closed contour D_1 in D:

$$N(F,D)=\lim_{D_1\to\partial D}\int_{D_1}|f(z)dz|<\infty.$$

The following theorems guarantee the composite truncated cardinal Sinc series decay errors exponentially for approximating the function $f(x)\in B(D)$.

Theorem 3.1. *If there are existing positive constants α, β, and C such that [23,35]:*

$$|f(x)|\leq C\begin{cases} \exp(-\alpha|\phi(x)|), & \text{if } x\in\Gamma_a\equiv\{x\in\Gamma:\phi(x)\in(-\infty,0)\},\\ \exp(-\beta|\phi(x)|), & \text{if } x\in\Gamma_b\equiv\{x\in\Gamma:\phi(x)\in[0,\infty)\},\end{cases} \tag{3.8}$$

and the selections

$$N_x=\left[\left|\frac{\alpha}{\beta}M_x+1\right|\right] \quad \text{and} \quad h_S=\left(\frac{\pi d}{\alpha M_x}\right)^{1/2}\leq\frac{2\pi d}{\ln(2)}$$

are made, then for all $x\in\Gamma$,

$$\|f(x)-C_{M,N}(f,h,\phi)(x)\|\leq K_5N_x^{1/2}\exp\Big(-(\pi d\alpha N_x)^{1/2}\Big),$$

where K_5 is a constant depending on f, d, ϕ, and D. The SE transformation is commonly used under above selections. Besides, if $\alpha/M_x\beta$ is an integer, then the choice for N_x equals to the value of M_x.

Theorem 3.2. *With positive constants δ, γ and C, for $f(x)\in B(D)$, if $f(x)$ decays double exponentially on the real line [39,40], i.e.,*

$$|f(x)|\leq C\exp\Big(-\delta\exp(\gamma|\phi(x)|)\Big), \quad x\in[a,b], \tag{3.9}$$

then we have

$$\sup_{a\leq x\leq b}\Big|f(x)-\sum_{j=-N_x}^{N_x}f(jh)sinc\Big(\frac{\phi(x)-jh}{h}\Big)\Big|\leq K_6\exp\Big(\frac{-\pi d\gamma N_x}{\log(\pi d\gamma N_x/\beta)}\Big), \tag{3.10}$$

where K_6 is a constant depending on f, d, ϕ, and D. In addition, the value of M_x is normally equal to N_x, and the mesh size h is taken as

$$h_D=\frac{\ln(\pi d\gamma N_x/\delta)}{\gamma N_x}. \tag{3.11}$$

Theorem 3.3. *The derivatives of the composite truncated Sinc function in Eq.* (3.4) *evaluated at the Sinc node $x_p=\phi^{-1}(ph)$ by the technical calculations is calculated as follows [23]:*

$$\delta_{pj}^{(0)}\equiv[S(j,h)\circ\phi(x)]|_{x=x_p}=\begin{cases}1, & \text{if } p=j,\\ 0, & \text{if } p\neq j,\end{cases} \tag{3.12a}$$

$$\delta_{pj}^{(1)}\equiv h\frac{d[S(j,h)\circ\phi(x)]}{d\phi(x)}\Big|_{x=x_p}=\begin{cases}0, & \text{if } p=j,\\ \dfrac{(-1)^{p-j}}{p-j}, & \text{if } p\neq j,\end{cases} \tag{3.12b}$$

$$\delta_{pj}^{(2)}\equiv h^2\frac{d^2[S(j,h)\circ\phi(x)]}{d\phi(x)^2}\Big|_{x=x_p}=\begin{cases}\dfrac{-\pi^2}{3}, & \text{if } p=j,\\ \dfrac{(-2)(-1)^{p-j}}{(p-j)^2}, & \text{if } p\neq j.\end{cases} \tag{3.12c}$$

Theorem 3.4. *Let $f(x)$ have a second order derivative defined on $[a,b]$, the integral of the composite truncated cardinal Sinc series is as follows:*

$$f(x)=\sum_{j=-M_x}^{N_x} f_x(x_j)\int_a^x S(j,h)\circ\phi(s)ds, \tag{3.13}$$

$$=\sum_{j=-M_x}^{N_x} h\frac{f_x(x_j)}{\phi'(x_j)}\times\begin{cases} 0, & \text{if } x=a,\\ \delta_{pj}^{(-1)}, & \text{if } x=x_p\in(a,b),\\ 1, & \text{if } x=b, \end{cases} \tag{3.14}$$

where

$$\delta_{pj}^{(-1)}=\begin{cases} \dfrac{1}{2}, & \text{if } p=j,\\ \dfrac{1}{2}+\displaystyle\int_0^{p-j}\frac{\sin(\pi w)}{\pi w}dw, & \text{if } p\neq j. \end{cases} \tag{3.15}$$

Proof. As $f(x)$ has a second order derivative, the first derivative $f_x(x)$ is continuous on $[a,b]$. Hence, the function

$$f(x)=\int_a^x f_x(s)ds$$

exists. If we approximate the function $f_x(s)$ by the composite truncated cardinal Sinc series Eq. (3.4), then we originate Eq. (3.13). To prove Eq. (3.14), we designate $s=\psi(t)$, where t is defined over $\mathbb{R}$, and replace in the function of $f(x)$:

$$\begin{aligned} f(x)&=\int_{\psi^{-1}(a)}^{\psi^{-1}(x)} f_x(\psi(t))\psi'(t)dt\\ &=\int_{-\infty}^{\psi^{-1}(x)}\frac{df(\psi(t))}{dt}dt. \end{aligned} \tag{3.16}$$

Next, approximating the function $\frac{df(\psi(t))}{dt}$ by the composite truncated cardinal Sinc series Eq. (3.4), we obtain the definite integral of it whose upper and lower limit are concerning the variable x and the negative infinity, respectively:

$$\begin{aligned} f(x)&=\int_{-\infty}^{\psi^{-1}(x)}\sum_{j=-M_x}^{N_x}\frac{df(\psi(t))}{dt}\Big|_{t=jh}S(j,h)(t)dt\\ &=\sum_{j=-M_x}^{N_x} f_x(\psi(jh))\psi'(jh)\int_{-\infty}^{\psi^{-1}(x)} sinc\Big(\frac{t-jh}{h}\Big)dt\\ &=\sum_{j=-M_x}^{N_x}\frac{f_x(x_j)}{\phi'(x_j)}\int_{-\infty}^{\psi^{-1}(x)} sinc\Big(\frac{t-jh}{h}\Big)dt. \end{aligned} \tag{3.17}$$

We redefine

$$z=\pi\Big(\frac{t}{h}-j\Big),$$

i.e., $\frac{dz}{dt}=\frac{\pi}{h}$ and replace it in Eq. (3.17):

$$f(x)=\sum_{j=-M_x}^{N_x}\frac{f_x(x_j)}{\phi'(x_j)}\frac{h}{\pi}\int_{-\infty}^{\pi(\frac{\psi^{-1}(x)}{h}-j)}\frac{\sin(z)}{z}dz. \tag{3.18}$$

(i). If $x=a$, then Eq. (3.18) becomes:

$$f(a)=\sum_{j=-M_x}^{N_x}\frac{f_x(x_j)}{\phi'(x_j)}\frac{h}{\pi}\int_{-\infty}^{-\infty}\frac{\sin(z)}{z}dz=0. \tag{3.19}$$

(ii). Besides, while evaluating the Sinc nodes $x_p\in(a,b)$

$$\begin{aligned}f(x_p)&=\sum_{j=-M_x}^{N_x}\frac{f_x(x_j)}{\phi'(x_j)}\frac{h}{\pi}\Big(\int_{-\infty}^{0}\frac{\sin(z)}{z}dz+\int_{0}^{\pi(\frac{\psi^{-1}(x_p)}{h}-j)}\frac{\sin(z)}{z}dz\Big)\\&=\sum_{j=-M_x}^{N_x}\frac{f_x(x_j)}{\phi'(x_j)}\frac{h}{\pi}\Big(\frac{\pi}{2}+\int_{0}^{p-j}\frac{\sin(w\pi)}{w\pi}\pi dw\Big)\\&=\sum_{j=-M_x}^{N_x}h\frac{f_x(x_j)}{\phi'(x_j)}\Big(\frac{1}{2}+\int_{0}^{p-j}\frac{\sin(w\pi)}{w\pi}dw\Big),\end{aligned} \tag{3.20}$$

where $w=\frac{z}{\pi}$, i.e., $dw=\frac{dz}{\pi}$. Thus, we obtain

$$\delta_{pj}^{(-1)}=\begin{cases}\dfrac{1}{2}, & \text{if } p=j,\\ \dfrac{1}{2}+\displaystyle\int_0^{p-j}\frac{\sin(\pi w)}{\pi w}dw, & \text{if } p\neq j.\end{cases} \tag{3.21}$$

(iii). Moreover, when $x=b$, based on the transformed function concerning z in Eq. (3.18), $f(b)$ can be expressed as:

$$\begin{aligned}f(b)&=\sum_{j=-M_x}^{N_x}\frac{f_x(x_j)}{\phi'(x_j)}\frac{h}{\pi}\int_{-\infty}^{\infty}\frac{\sin(z)}{z}dz\\&=\sum_{j=-M_x}^{N_x}\frac{f_x(x_j)}{\phi'(x_j)}\frac{h}{\pi}\pi=h\sum_{j=-M_x}^{N_x}\frac{f_x(x_j)}{\phi'(x_j)}.\end{aligned} \tag{3.22}$$

Therefore, the integral of the composite truncated cardinal Sinc series is defined as Eq. (3.13) to Eq. (3.15). In particular, for $j=-M_x,\cdots,N_x$, we have:

$$\int_a^x S(j,h)\circ\phi(s)ds=\frac{h}{\phi'(x_j)}\times\begin{cases}0, & \text{if } x=a,\\ \delta_{pj}^{(-1)}, & \text{if } x=x_p\in(a,b),\\ 1, & \text{if } x=b.\end{cases} \tag{3.23}$$

This completes the proof. □

3.2 Numerical treatment by the Sinc Collocation Method (SCM)

By the SCM, the approximate solution of $U^{n+1}(x,y)$ for $\overline{\Omega}=[a,b]\times[a,b]$ at time t^{n+1} is defined as

$$U_1^{n+1}(x,y)=\sum_{j=-M_x-1}^{N_x+1}\sum_{i=-M_y-1}^{N_y+1}C_{j,i}^{n+1}\varkappa_j(x)\varkappa_i(y), \tag{3.24}$$

where

$$\varkappa_j(x)=\begin{cases}\zeta_1(x)=\dfrac{b-x}{b-a}, & \text{if } j=-M_x-1,\\ S(j,h)\circ\phi(x), & \text{if } j=-M_x,\cdots,N_x,\\ \zeta_2(x)=\dfrac{x-a}{b-a}, & \text{if } j=N_x+1,\end{cases} \tag{3.25a}$$

$$\varkappa_i(y)=\begin{cases}\zeta_1(y)=\dfrac{b-y}{b-a}, & \text{if } i=-M_y-1,\\ S(i,h)\circ\phi(y), & \text{if } i=-M_y,\cdots,N_y,\\ \zeta_2(y)=\dfrac{y-a}{b-a}, & \text{if } i=N_y+1,\end{cases} \tag{3.25b}$$

as well as denoting $m_x=M_x+N_x+3$ and $m_y=M_y+N_y+3$. In Eq. (3.24), $C_{j,i}^{n+1}$ needs to be solved by interpolating the non-homogeneous Dirichlet boundary conditions and the selected interior Sinc nodes through the difference scheme Eq. (2.4) at every time level t^{n+1}.

First, to apply boundary conditions for points (a,a), (a,b), (b,a), (b,b), (a,y_q), (b,y_q), (x_p,a) and (x_p,b), respectively, the series x_p is originated from Eq. (3.7), and the sequence y_q is generated by:

$$y_q=\psi(qh),\quad q=-M_y,\cdots,N_y. \tag{3.26}$$

The following equations are satisfied:

$$\sum_{j=-M_x-1}^{N_x+1}\sum_{i=-M_y-1}^{N_y+1}C_{j,i}^{n+1}\varkappa_j(a)\varkappa_i(a)=\Theta(a,a,t^{n+1}), \tag{3.27a}$$

$$\sum_{j=-M_x-1}^{N_x+1}\sum_{i=-M_y-1}^{N_y+1}C_{j,i}^{n+1}\varkappa_j(a)\varkappa_i(b)=\Theta(a,b,t^{n+1}), \tag{3.27b}$$

$$\sum_{j=-M_x-1}^{N_x+1}\sum_{i=-M_y-1}^{N_y+1}C_{j,i}^{n+1}\varkappa_j(b)\varkappa_i(a)=\Theta(b,a,t^{n+1}), \tag{3.27c}$$

$$\sum_{j=-M_x-1}^{N_x+1}\sum_{i=-M_y-1}^{N_y+1}C_{j,i}^{n+1}\varkappa_j(b)\varkappa_i(b)=\Theta(b,b,t^{n+1}), \tag{3.27d}$$

$$\sum_{j=-M_x-1}^{N_x+1}\sum_{i=-M_y-1}^{N_y+1} C_{j,i}^{n+1}\varkappa_j(a)\varkappa_i(y_q)=\Theta(a,y_q,t^{n+1}), \tag{3.27e}$$

$$\sum_{j=-M_x-1}^{N_x+1}\sum_{i=-M_y-1}^{N_y+1} C_{j,i}^{n+1}\varkappa_j(b)\varkappa_i(y_q)=\Theta(b,y_q,t^{n+1}), \tag{3.27f}$$

$$\sum_{j=-M_x-1}^{N_x+1}\sum_{i=-M_y-1}^{N_y+1} C_{j,i}^{n+1}\varkappa_j(x_p)\varkappa_i(a)=\Theta(x_p,a,t^{n+1}), \tag{3.27g}$$

$$\sum_{j=-M_x-1}^{N_x+1}\sum_{i=-M_y-1}^{N_y+1} C_{j,i}^{n+1}\varkappa_j(x_p)\varkappa_i(b)=\Theta(x_p,b,t^{n+1}). \tag{3.27h}$$

Then, for interpolating the Sinc collocation nodes (x_p,y_q), we construct an $m_x\times m_y$ meshgrid matrix with the vector sequence x_p as the column and vector sequence y_q as the row, where the two series are consistent with Eq. (3.7) and Eq. (3.26), respectively. After that, using the following system interpolates all Sinc interpolating nodes (x_p,y_q) in the defined matrix:

$$\begin{aligned}&\sum_{j=-M_x-1}^{N_x+1}\sum_{i=-M_y-1}^{N_y+1} C_{j,i}^{n+1}\big(i\varkappa_j(x_p)\varkappa_i(y_q)+\lambda[\alpha(\varkappa_j^{''}(x_p)\varkappa_i(y_q)+\varkappa_j(x_p)\varkappa_i^{''}(y_q))\\ &\qquad +f(x_p,y_q)\varkappa_j(x_p)\varkappa_i(y_q)+\rho|U_1^n(x_p,y_q)|^2\varkappa_j(x_p)\varkappa_i(y_q)]\big)\\ &=iU_1^n(x_p,y_q)-\lambda[\alpha(U_{1_{xx}}^n(x_p,y_q)+U_{1_{yy}}^n(x_p,y_q)+f(x_p,y_q)U_1^n(x_p,y_q)\\ &\qquad +\rho\vartheta(U_1^n(x_p,y_q))].\end{aligned} \tag{3.28}$$

The right side in the above equation is denoted as $H_1^n(x_p,y_q)$. From above, the approximated wave function $U_1^n(x_p,y_q)$ at the time level t^n is as a known value to calculate all coefficients at the time level t^{n+1}.

All unknown variables involved in Eq. (3.27) and Eq. (3.28) can be solved as a vector in the system of linear matrix equation at every time level using Python 3.7. The series C_{xy}^{n+1} is as the following order:

$$C_{xy}^{n+1}=\big[C_{-M_x-1,-M_y-1}^{n+1},\cdots,C_{-M_x-1,N_y+1}^{n+1},\cdots,C_{N_x+1,-M_y-1}^{n+1},\cdots,C_{N_x+1,N_y+1}^{n+1}\big]^T.$$

To transform the linear equations into a matrix format, given any series $g(x_k)$ for $k=1,\cdots,n$, defining the $n\times n$ diagonal matrix as

$$D(g(x_k))=diag(g(x_1),\cdots,g(x_n)),$$

and the column vector as

$$V(g(x_k))=[g(x_1),\cdots,g(x_n)]^T.$$

Besides, for transforming part components in the system into the matrix equations, we define the following $1\times m_x$ horizontal vectors by calculating the series function $\varkappa_j(x)$ at the valves of a and b, separately:

$$I_{x_a}^{(0)}=\big[\varkappa_j(a)\quad \text{for } j=-M_x-1,\cdots,N_x+1\big], \tag{3.29a}$$

$$I_{x_b}^{(0)}=\big[\varkappa_j(b)\quad \text{for } j=-M_x-1,\cdots,N_x+1\big]. \tag{3.29b}$$

Likewise, $I_{y_a}^{(0)}$ and $I_{y_b}^{(0)}$, as $1\times m_y$ vectors, are generated by calculating the series function $\varkappa_i(y)$ at a and b, respectively. Moreover, the matrices $I_{x_p\ (m_x-2)\times m_x}^{(0)}$ and $I_{y_q\ (m_y-2)\times m_y}^{(0)}$ are constructed by replacing the selected Sinc series values x_p and y_q in the two functions $\varkappa_j(x)$ and $\varkappa_i(y)$ according to the definition Eq. (3.12a), respectively. For instance,

$$I_{x_p}^{(0)}=\begin{bmatrix} \dfrac{b-x_{-M_x}}{b-a} & 1 & 0 & \cdots & 0 & 0 & \dfrac{x_{-M_x}-a}{b-a} \\ \vdots & \vdots & \ddots & \cdots & \vdots & \vdots & \vdots \\ \dfrac{b-x_0}{b-a} & 0 & \cdots & 1 & \cdots & 0 & \dfrac{x_0-a}{b-a} \\ \vdots & \vdots & \vdots & \cdots & \ddots & \vdots & \vdots \\ \dfrac{b-x_{N_x}}{b-a} & 0 & 0 & \cdots & 0 & 1 & \dfrac{x_{N_x}-a}{b-a} \end{bmatrix}. \tag{3.30}$$

The vertical direction is following the sequence of changing the values of p or q; also, the changing order of j or i is in the horizontal line. Moreover, based on definitions Eq. (3.12b) and Eq. (3.12c), $I_{x_p}^{(2)}$ and $I_{y_q}^{(2)}$ are obtained from the second-order derivative of the following two functions, separately:

$$\varkappa_j''(x_p)=\begin{cases} 0, & \text{if } j=-M_x-1, \\ \dfrac{\delta_{pj}^{(2)}}{h^2}(\phi'(x_p))^2+\dfrac{\delta_{pj}^{(1)}}{h}\phi''(x_p), & \text{if } j=-M_x,\cdots,N_x, \\ 0, & \text{if } j=N_x+1, \end{cases} \tag{3.31a}$$

$$\varkappa_i''(y_q)=\begin{cases} 0, & \text{if } i=-M_y-1, \\ \dfrac{\delta_{qi}^{(2)}}{h^2}(\phi'(y_q))^2+\dfrac{\delta_{qi}^{(1)}}{h}\phi''(y_q), & \text{if } i=-M_y,\cdots,N_y, \\ 0, & \text{if } i=N_y+1. \end{cases} \tag{3.31b}$$

Hence $I_{x_p}^{(2)}$ can be expressed as:

$$I_{x_p}^{(2)}=\begin{bmatrix} 0 & \frac{-\pi^2(\phi'(x_{-M_x}))^2}{3h^2} & \cdots & -\frac{2(\phi'(x_p))^2+\phi''(x_p)h(M_x+N_x)}{(-1)^{(M_x+N_x)}h^2(M_x+N_x)^2} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \frac{(-1)^{M_x}(-2(\phi'(x_0))^2+\phi''(x_0)hM_x)}{h^2(M_x)^2} & \cdots & \frac{-2(\phi'(x_0))^2-\phi''(x_0)hN_x}{(-1)^{N_x}(N_x)^2h^2} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \frac{\phi''(x_{N_x})h(N_x+M_x)-2(\phi'(x_{N_x}))^2}{(-1)^{-N_x-M_x}h^2(N_x+M_x)^2} & \cdots & \frac{-\pi^2(\phi'(x_{N_x}))^2}{3h^2} & 0 \end{bmatrix}. \tag{3.32}$$

Also, $I_{y_q}^{(2)}$ can be expressed similarly. Finally, the unknown variables are solved in the following matrix equation through iterating fixed $n+1$ times :

$$\begin{bmatrix} I_{x_a}^{(0)}\otimes I_{y_a}^{(0)} \\ I_{x_a}^{(0)}\otimes I_{y_b}^{(0)} \\ I_{x_b}^{(0)}\otimes I_{y_a}^{(0)} \\ I_{x_b}^{(0)}\otimes I_{y_b}^{(0)} \\ I_{x_a}^{(0)}\otimes I_{y_q}^{(0)} \\ I_{x_b}^{(0)}\otimes I_{y_q}^{(0)} \\ I_{x_p}^{(0)}\otimes I_{y_a}^{(0)} \\ I_{x_p}^{(0)}\otimes I_{y_b}^{(0)} \\ D(i+\lambda(f(x_p,y_q)+\rho|U_1^n(x_p,y_q)|^2) \\ \cdot(I_{x_p}^{(0)}\otimes I_{y_q}^{(0)})+\alpha\lambda(I_{x_p}^{(2)}\otimes I_{y_q}^{(0)}+I_{x_p}^{(0)}\otimes I_{y_q}^{(2)}) \end{bmatrix} C_{xy}^{n+1}=\begin{bmatrix} \Theta(a,a,t^{n+1}) \\ \Theta(a,b,t^{n+1}) \\ \Theta(b,a,t^{n+1}) \\ \Theta(b,b,t^{n+1}) \\ \Theta(a,y_q,t^{n+1}) \\ \Theta(b,y_q,t^{n+1}) \\ \Theta(x_p,a,t^{n+1}) \\ \Theta(x_p,b,t^{n+1}) \\ V(H_1^n(x_p,y_q)) \end{bmatrix}, \tag{3.33}$$

where $\cdot$ represents the matrix multiplication, and $\otimes$ denotes the Kronecker product.

Thus, all unknown variables can be solved from the above matrix equation at every time level, and then the approximated wave function is determined. In particular, the initial conditions are used to calculate the unknown variables at the time level t^1.

3.3 Numerical treatment by the Sinc-Derivative Collocation method

Based on Theorem 3.4, the new approximation including the Sinc Derivative for two-dimensional GPE is defined as follows:

$$U_2^{n+1}(x,y)=\sum_{j=-M_x-1}^{N_x+1}\sum_{i=-M_y-1}^{N_y+1} c_{j,i}^{n+1}\varsigma_j(x)\varsigma_i(y), \tag{3.34}$$

where

$$\varsigma_j(x)=\begin{cases} \zeta_3(x)=\dfrac{(x-b)^2(2x-3a+b)}{(b-a)^3}, & \text{if } j=-M_x-1,\\ \zeta_4(x)=\dfrac{(x-b)^2(x-a)}{(b-a)^2}, & \text{if } j=-M_x,\\ \displaystyle\int_a^x S(j,h)\circ\phi(s)ds, & \text{if } j=-M_x+1,\cdots,N_x,\\ \zeta_5(x)=\dfrac{(x-b)(x-a)^2}{(b-a)^2}, & \text{if } j=N_x+1, \end{cases} \tag{3.35a}$$

$$\varsigma_i(y)=\begin{cases} \zeta_3(y)=\dfrac{(y-b)^2(2y-3a+b)}{(b-a)^3}, & \text{if } i=-M_y-1,\\ \zeta_4(y)=\dfrac{(y-b)^2(y-a)}{(b-a)^2}, & \text{if } i=-M_y,\\ \displaystyle\int_a^y S(i,h)\circ\phi(s)ds, & \text{if } i=-M_y+1,\cdots,N_y,\\ \zeta_5(y)=\dfrac{(y-b)(y-a)^2}{(b-a)^2}, & \text{if } i=N_y+1. \end{cases} \tag{3.35b}$$

Especially, Eq. (3.15) and Eq. (3.23) are utilized to enumerate the function $\int_a^x S(j,h)\circ\phi(s)ds$. Besides, since the second-order derivative of U_2^{n+1} is involved, the second-order derivative of the above two functions Eq. (3.35a) and Eq. (3.35b) are as follows:

$$\varsigma_j''(x)=\begin{cases} \zeta_3''(x)=\dfrac{-12x+6b+6a}{(a-b)^3}, & \text{if } j=-M_x-1,\\ \zeta_4''(x)=\dfrac{-2a-4b+6x}{(a-b)^2}, & \text{if } j=-M_x,\\ \dfrac{d[S(j,h)\circ\phi(x)]}{d\phi(x)}, & \text{if } j=-M_x+1,\cdots,N_x,\\ \zeta_5''(x)=\dfrac{-4a-2b+6x}{(a-b)^2}, & \text{if } j=N_x+1, \end{cases} \tag{3.36a}$$

$$\varsigma_i''(y)=\begin{cases} \zeta_3''(x)=\dfrac{-12y+6b+6a}{(a-b)^3}, & \text{if } i=-M_y-1,\\ \zeta_4''(x)=\dfrac{-2a-4b+6y}{(a-b)^2}, & \text{if } i=-M_y,\\ \dfrac{d[S(j,h)\circ\phi(y)]}{d\phi(y)}, & \text{if } i=-M_y+1,\cdots,N_y,\\ \zeta_5''(x)=\dfrac{-4a-2b+6y}{(a-b)^2}, & \text{if } j=N_y+1. \end{cases} \tag{3.36b}$$

The definition in Eq. (3.34) includes $m_x\times m_y$ unknown variables resolved as the procedure by the SCM. That means as many as equations are required to solve these unknown variables $c_{j,i}^{n+1}$ by interpolating boundary points and the interior Sinc collocation nodes (x_p,y_q). All interpolating collocation nodes are equivalent to the nodes selected in the SCM. Through interpolating the same points by $U_2(x,y)$, we can also obtain the equations, like Eq. (3.27) and Eq. (3.28), in which the unknown coefficients $C_{j,i}^{n+1}$ and functions $\varkappa_j(x)$, $\varkappa_i(y)$, $U_1^{n+1}(x,y)$ become $c_{j,i}^{n+1}$ and functions $\varsigma_j(x)$, $\varsigma_i(y)$, $U_2^{n+1}(x,y)$, respectively. Also, $H_2^n(x_p,y_q)$ is denoted as the right side of Eq. (3.28) where $U_1^n(x,y)$ is replaced by $U_2^n(x,y)$. To employ the same scheme Eq. (3.33) solving unknown coefficients in Eq. (3.34) at every time level, it is necessary to compute series functions $\varsigma_j(x)$ at values a and b, respectively:

$$I_{x_a}^{(-1)} = \big[\varsigma_j(a) \quad \text{for } j=-M_x-1,\cdots,N_x+1\big], \tag{3.37a}$$

$$I_{x_b}^{(-1)} = \big[0 \quad 0 \quad h\psi'((-M_x-1)h) \quad \cdots \quad h\psi'(N_xh) \quad 0\big]. \tag{3.37b}$$

We can also obtain vectors $I_{y_a}^{(-1)}$ and $I_{y_b}^{(-1)}$ by calculating the series function $\varsigma_i(y)$ when y equals to a and b, separately. Furthermore, we construct $(m_x-2)\times m_x$ matrices $I_{x_p}^{(-1)}$ and $I_{x_p}^{(1)}$ to demonstrate the series functions $\varsigma_j(x)$ and $\varsigma_j^{''}(x)$ computed at $x-$interpolating nodes, respectively. As the function $\varsigma_j(x)$ includes the first-order derivative of the Sinc function, we build the following $(m_x-2)\times(m_x-3)$ matrix $I_{S_d}^{(-1)}$ to illustrate the function values at $x-$nodes,

$$I_{S_d}^{(-1)} = h\begin{bmatrix} \dfrac{\frac{1}{2}+\int_0^{-1}\frac{\sin(\pi w)}{\pi w}dw}{\phi'(x_{-M_x+1})} & \cdots & \dfrac{\frac{1}{2}+\int_0^{-M_x}\frac{\sin(\pi w)}{\pi w}dw}{\phi'(x_0)} & \cdots & \dfrac{\frac{1}{2}+\int_0^{-M_x-N_x}\frac{\sin(\pi w)}{\pi w}dw}{\phi'(x_{N_x})} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \dfrac{\frac{1}{2}+\int_0^{M_x-1}\frac{\sin(\pi w)}{\pi w}dw}{\phi'(x_{-M_x+1})} & \cdots & \dfrac{1}{2\phi'(x_0)} & \cdots & \dfrac{\frac{1}{2}+\int_0^{-N_x}\frac{\sin(\pi w)}{\pi w}dw}{\phi'(x_{N_x})} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \dfrac{\frac{1}{2}+\int_0^{M_x+N_x-1}\frac{\sin(\pi w)}{\pi w}dw}{\phi'(x_{-M_x+1})} & \cdots & \dfrac{\frac{1}{2}+\int_0^{N_x}\frac{\sin(\pi w)}{\pi w}dw}{\phi'(x_0)} & \cdots & \dfrac{1}{2\phi'(x_{N_x})} \end{bmatrix}. \tag{3.38}$$

After that, the matrix $I_{x_p}^{(-1)}$ concatenated by the entries in $I_{S_d}^{(-1)}$ can be expressed as follows:

$$I_{x_p}^{(-1)} = \left[\begin{array}{c|c|c|c} \dfrac{(x_{-M_x}-b)^2(2x_{-M_x}-3a+b)}{(b-a)^3} & \dfrac{(x_{-M_x}-b)^2(x_{-M_x}-a)}{(b-a)^2} & & \dfrac{(x_{-M_x}-b)(x_{-M_x}-a)^2}{(b-a)^2} \\ \vdots & \vdots & & \vdots \\ \dfrac{(x_0-b)^2(2x_0-3a+b)}{(b-a)^3} & \dfrac{(x_0-b)^2(x_0-a)}{(b-a)^2} & I_{S_d}^{(-1)} & \dfrac{(x_0-b)(x_0-a)^2}{(b-a)^2} \\ \vdots & \vdots & & \vdots \\ \dfrac{(x_{N_x}-b)^2(2x_{N_x}-3a+b)}{(b-a)^3} & \dfrac{(x_{N_x}-b)^2(x_{N_x}-a)}{(b-a)^2} & & \dfrac{(x_{N_x}-b)(x_{N_x}-a)^2}{(b-a)^2} \end{array}\right]. \tag{3.39}$$

In addition, the function $\varsigma_j^{''}(x)$ for $j=-M_x+1,\cdots,N_x$ enumerated at $x-$nodes can be constructed in the following matrix based on Eq. (3.12b):

$$I_{S_d}^{(1)}=\frac{1}{h}\begin{bmatrix} 1 & \cdots & \dfrac{(-1)^{-M_x}}{-M_x} & \cdots & \dfrac{(-1)^{-M_x-N_x}}{-M_x-N_x} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \dfrac{(-1)^{M_x-1}}{M_x-1} & \cdots & 0 & \cdots & \dfrac{(-1)^{-N_x}}{-N_x} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \dfrac{(-1)^{N_x+M_x-1}}{N_x+M_x-1} & \cdots & \dfrac{(-1)^{N_x}}{N_x} & \cdots & 0 \end{bmatrix}.$$

Hence, the matrix $I_{x_p}^{(1)}$ can be expressed as follows:

$$I_{x_p}^{(1)}=\left[\begin{array}{c|c|c|c} \dfrac{-12x_{-M_x}+6b+6a}{(a-b)^3} & \dfrac{-2a-4b+6x_{-M_x}}{(a-b)^2} & & \dfrac{-4a-2b+6x_{-M_x}}{(a-b)^2} \\ \vdots & \cdots & & \cdots \\ \dfrac{-12x_0+6b+6a}{(a-b)^3} & \dfrac{-2a-4b+6x_0}{(a-b)^2} & I_{S_d}^{(1)} & \dfrac{-4a-2b+6x_0}{(a-b)^2} \\ \vdots & \cdots & & \cdots \\ \dfrac{-12x_{N_x}+6b+6a}{(a-b)^3} & \dfrac{-2a-4b+6x_{N_x}}{(a-b)^2} & & \dfrac{-4a-2b+6x_{N_x}}{(a-b)^2} \end{array}\right]. \tag{3.40}$$

In like wise, we obtain $(m_y-2)\times m_y$ matrices $I_{y_q}^{(-1)}$ and $I_{y_q}^{(1)}$ while computing $\varsigma_i(y)$ and $\varsigma_i^{''}(y)$ at $y-$interpolating nodes, separately.

Thus, the unknown variables are solved as a vector c_{xy}^{n+1} in the following matrix equation through iterating fixed $n+1$ times, where the vector c_{xy}^{n+1} is following the order of C_{xy}^{n+1}:

$$\begin{bmatrix} I_{x_a}^{(-1)}\otimes I_{y_a}^{(-1)} \\ I_{x_a}^{(-1)}\otimes I_{y_b}^{(-1)} \\ I_{x_b}^{(-1)}\otimes I_{y_a}^{(-1)} \\ I_{x_b}^{(-1)}\otimes I_{y_b}^{(-1)} \\ I_{x_a}^{(-1)}\otimes I_{y_q}^{(-1)} \\ I_{x_b}^{(-1)}\otimes I_{y_q}^{(-1)} \\ I_{x_p}^{(-1)}\otimes I_{y_a}^{(-1)} \\ I_{x_p}^{(-1)}\otimes I_{y_b}^{(-1)} \\ D(i+\lambda(f(x_p,y_q)+\rho|U_2^n(x_p,y_q)|^2) \\ \cdot(I_{x_p}^{(-1)}\otimes I_{y_q}^{(-1)})+\alpha\lambda(I_{x_p}^{(1)}\otimes I_{y_q}^{(-1)}+I_{x_p}^{(-1)}\otimes I_{y_q}^{(1)}) \end{bmatrix} c_{xy}^{n+1}=\begin{bmatrix} \Theta(a,a,t^{n+1}) \\ \Theta(a,b,t^{n+1}) \\ \Theta(b,a,t^{n+1}) \\ \Theta(b,b,t^{n+1}) \\ \Theta(a,y_q,t^{n+1}) \\ \Theta(b,y_q,t^{n+1}) \\ \Theta(x_p,a,t^{n+1}) \\ \Theta(x_p,b,t^{n+1}) \\ V(H_2^n(x_p,y_q)) \end{bmatrix}. \tag{3.41}$$

3.4 Least square solution

From the above discussion, the unknown coefficients involved in the approximated functions $U_l(x,y)$ for $l=1,2$ are solved by the linear matrix equation systems Eq. (3.33) and Eq. (3.41), respectively. In theory, the square coefficient matrices in the two systems are full rank, which guarantees the existence of the solution. However, it may or may not be possible to compute the solution accurately. It commonly displays the uncertainty that a small perturbation in the system's right side can give large errors in the solution vector of unknown variables [41].

To circumvent the issue, we are concerned with the solution of least-squares problems, which is a prevailing and profound treatment for linear matrix equations in academic [42]:

$$\min_{\mathbf{c}} ||\mathbf{Ac}-\mathbf{B}||_2, \tag{3.42}$$

where $\mathbf{A}$ represents the coefficient matrix, $\mathbf{c}$ is the solution, and $\mathbf{B}$ is the right side of the equation system in Eq. (3.33) or Eq. (3.41). With this regard, we adopt the function code *numpy.linalg.solve*() in Python to obtain the results. If there are multiple minimizing solutions, the one with the smallest $2-$norm $||\mathbf{c}||$ is returned [41].

4 Numerical results

To test the accuracy and verify the performance of the proposed two numerical approaches, five examples are examined in this section. The maximum absolute error (MAE) between the exact and the approximate solution $||u_\varepsilon||_\infty$ defined as:

$$||u_\varepsilon||_\infty = \mathbf{Max}|u_{ex}(x,y,t^n)-u_{app}(x,y,t^n)|, \tag{4.1}$$

is used to demonstrate the precision. Also, the maximum absolute errors on the real and the imaginary parts are denoted by $||R_\varepsilon||_\infty$ and $||I_\varepsilon||_\infty$, respectively. The relative MAE defined as below is also employed:

$$R_\infty(u_\varepsilon)=\frac{||u_\varepsilon||_\infty}{\mathbf{Max}|u_{ex}(x,y,t^n)|}. \tag{4.2}$$

The error estimation is for approximating not only non-boundary interpolation nodes but also the test points distributed within the considerable domain. The two transformations Eq. (3.5) and Eq. (3.6) are utilized in all examples. Besides, Sinc numerical methods can maintain the convergence even if selecting $M_x=N_x=M_y=N_y$ in two dimensions [23]. According to Theorem 3.1, it is practically setting $\alpha=\beta=1$ and generating $h_S=\sqrt{\frac{\pi}{2M_x}}$ with $d=\frac{1}{2}$ by the SE transformation. Also, based on Theorem 3.2, we assign $\delta=1$ and $\gamma=2$ and yield $h_D=\frac{\ln(\pi N_x)}{2N_x}$ through the DE transformation.

Example 4.1. Considering a linear SCE equation:

$$i\frac{\partial u}{\partial t}+u_{xx}+u_{yy}+(3-2\tanh^2 x-2\tanh^2 y)u=0, \quad x,y\in[0,1]^2, \quad t\geq 0, \tag{4.3}$$

the exact solution of this example is as the following form [23,26,27,43]

$$u(x,y,t)=\frac{i\exp(it)}{\cosh x\cosh y}. \tag{4.4}$$

Also, the initial condition and boundary conditions are obtained from the exact solution.

Table 1: Accuracy comparison of sinc-based numerical methods for approximating interpolation nodes with different values of N_x and $dt=0.001$ at $t_{final}=1$.

Methods	$N_x\times N_y$	$\|\|R_\varepsilon\|\|_\infty$	$\|\|I_\varepsilon\|\|_\infty$	$R_\infty(u_\varepsilon)$
SCM(SE) [23]	8×8	3.0186E-05	2.2355E-05	3.7563E-05
	16×16	3.1856E-06	2.3591E-06	3.6880E-06
	32×32	6.1041E-08	7.3105E-08	8.6784E-08
SGM(SE) [23]	8×8	2.3733E-05	2.2993E-05	2.7234E-05
	16×16	3.0212E-06	1.9844E-06	3.5527E-06
	32×32	6.0660E-08	6.7787E-08	9.0965E-08
SSGM(SE) [23]	8×8	2.8190E-06	3.8209E-06	4.1021E-06
	16×16	1.1353E-07	1.5892E-07	1.7513E-07
	32×32	2.2225E-09	1.7258E-09	2.2403E-09
SDCM(SE)	8×8	3.0337E-05	2.1965E-05	3.7454E-05
	16×16	4.8016E-07	4.0400E-07	5.0977E-07
	32×32	1.0783E-08	8.9310E-09	1.4001E-08
SDCM(DE)	8×8	1.5618E-05	6.7860E-06	1.5734E-05
	16×16	2.1325E-07	1.5064E-07	2.1440E-07
	32×32	1.3752E-09	1.5032E-09	1.7925E-09

Table 1 compares the numerical results by the SDCM through SE and DE transformations, denoted by SDCM(SE) and SDCM(DE), receptively, with other Sinc numerical methods in [23], such as the SCM, the Sinc-Galerkin method (SGM), and the symmetric Sinc-Galerkin method (SSGM) by the SE transformation. With interpolating more Sinc nodes, our proposed SDCM(SE) is more accurate than the SCM(SE) and SGM(SE). Fig. 1 depicts the natural logarithm of relative MAEs towards different valves of $N_x(N_y)$ while approximating interpolation nodes by the SDCM with $d_t=0.001$ at $t_{final}=1$. It indicates the SDCM with the feature of quasi-exponentially decaying errors and the advantage of the DE transformation over the SE one as well. Finally, the surface plots for approximating uniform grids by the SDCM(DE) are shown in Fig. 2. Comparing with the analytical solution demonstrates the efficiency and accuracy of the SDCM.

Example 4.2. Similarly, considering a linear SCE equation,

$$iu_t+u_{xx}+u_{yy}-\frac{4x^2+4y^2-4x-4y-2}{4}u=0, \quad x,y\in[0,1]^2, \quad t\geq 0, \tag{4.5}$$

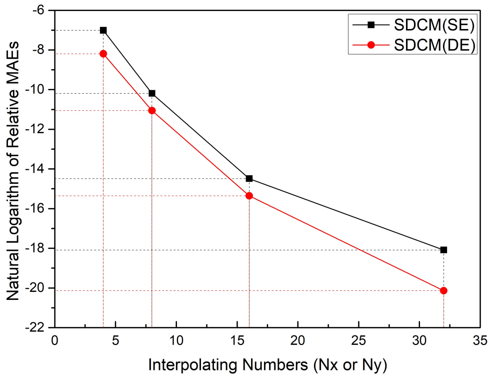
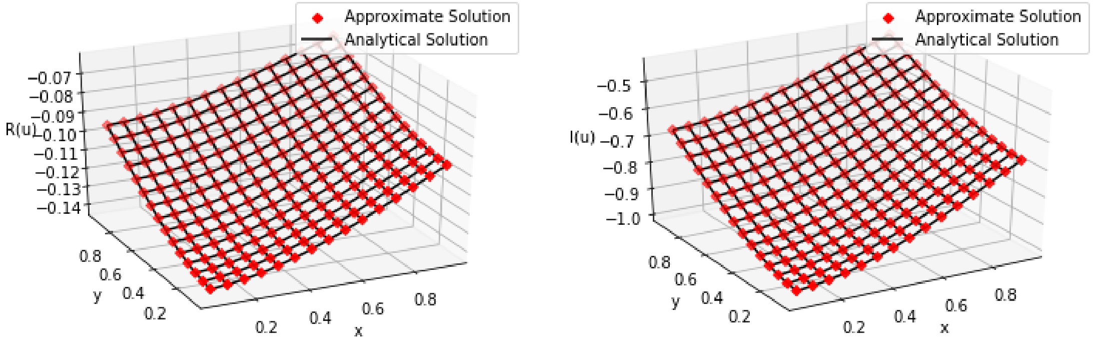


Figure 1: Natural logarithm of Relative MAEs concerning different values of N_x for approximating interpolation nodes with $dt=0.001$ at $t_{final}=1$ (Example 4.1).



(a) Real parts

(b) Imaginary parts

Figure 2: Surface plots of real parts and imaginary parts between the approximation and the exact solution under $N_x=16$ and $dt=1/100$ for uniform grids with the step $ds=1/16$ by the SDCM(DE) at the final time $t_{final}=3$.

where the exact solution is

$$u(x,y,t)=\exp\Big(-\frac{(x-0.5)^2}{2}-\frac{(y-0.5)^2}{2}-it\Big),$$

see [23]. The initial and the boundary conditions are obtained from the exact solution.

In this example, we set $h_S=\frac{\pi}{\sqrt{3N_x}}$ to mesh with the parameter selection in [23], whereas the parameters for the DE transformation are not changed. Table 2 lists the numerical errors obtained by Sinc numerical methods through the SE transformation. After comparing the numerical errors, the SDCM(SE) can lead to a more accurate result than the SCM(SE) and SGM(SE). It is also as accurate as the SSGM(SE) when interpolating more

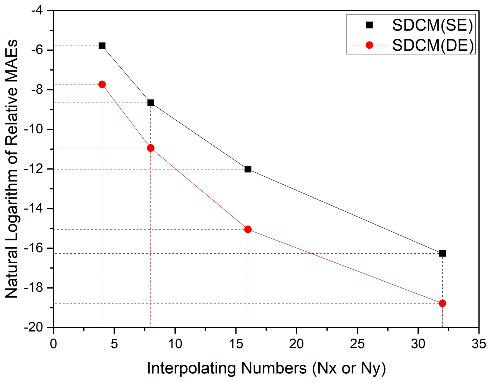
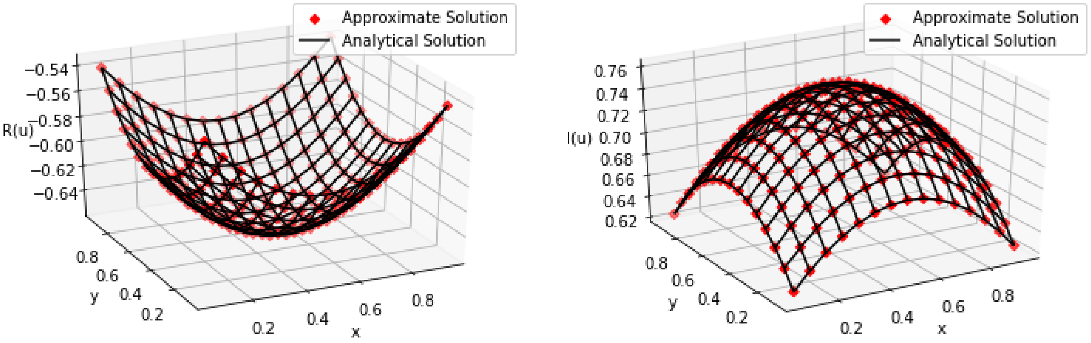


Figure 3: Natural logarithm of Relative MAEs concerning different values of N_x for approximating interpolation nodes with $dt=0.001$ at $t_{final}=1$ (Example 4.2).



(a) Real parts

(b) Imaginary parts

Figure 4: Surface plots of real parts and imaginary parts of approximate and exact solutions with $N_x=16$ and $dt=1/100$ for uniform grids with the step $ds=1/16$ by the SDCM(DE) at the final time $t_{final}=4$.

nodes. Fig. 3 plots the natural logarithm of relative MAEs concerning different numbers of interpolating nodes with $dt=0.001$ at $t_{final}=1$. It confirms the quasi-exponential convergence of the SDCM as well, and the DE transformation triggers the SDCM to be more accurate. Finally, Fig. 4 plots the surface plots for uniform grids approximated by the SDCM(DE) at $t_{final}=4$, which demonstrates the accuracy of our proposed SDCM through comparing with the analytical solution.

Example 4.3. Considering the GPE with the non-linear item:

$$iu_t+u_{xx}+u_{yy}+|u|^2u=0, \quad x,y\in[0,2\pi]^2, \quad t\geq 0, \tag{4.6}$$

where the exact solution is $u(x,y,t)=\exp(i(x+y-t))$ [25]. The initial and the boundary conditions are obtained from the exact solution.

Table 2: Accuracy comparison for approximating sinc nodes with different values of N_x and dt=0.01 at t_{final}=1.

Methods	$N_x\times N_y$	$\|\|R_\varepsilon\|\|_\infty$	$\|\|I_\varepsilon\|\|_\infty$	$R_\infty(u_\varepsilon)$
SCM(SE) [23]	8×8	2.0425E-04	1.9411E-04	2.3726E-04
	16×16	5.6107E-05	3.3497E-05	6.2899E-05
	32×32	2.7525E-06	1.5927E-06	2.7579E-06
SGM(SE) [23]	8×8	2.0174E-04	2.0489E-04	2.5279E-04
	16×16	6.1466E-05	3.3550E-05	6.7182E-05
	32×32	2.7072E-06	1.6009E-06	2.7085E-06
SSGM(SE) [23]	8×8	8.3694E-05	4.8119E-05	9.3360E-05
	16×16	6.4202E-06	6.9574E-06	8.3271E-06
	32×32	2.5099E-07	5.1128E-07	5.3137E-07
SDCM(SE)	8×8	1.2490E-04	1.3410E-04	1.6525E-04
	16×16	3.7073E-06	3.9853E-06	5.3079E-06
	32×32	4.5994E-07	6.3546E-07	7.8445E-07

For approximating the interpolation points, Fig. 5 depicts the natural logarithm of relative MAEs concerning different numbers of interpolation nodes by the SCM and the SDCM through SE and DE transformations, respectively. It indicates that the SDCM decays errors faster than the SCM no matter what transformation is applied. Then, considering approximating the uniform grids, Table 3 lists the numerical results obtained by estimating different uniform numbers P by the Meshless Symplectic Method (MSM) [25] and the two Sinc numerical approaches. The number of interpolation nodes and time costs are also recorded in this Table when the two Sinc approaches attain the same decimal place of numerical errors as the MSM. To generate a more accurate simulating result, the SDCM does not need to interpolate more nodes as the SCM, which saves more computational costs. That convinces the SDCM has the quality of decaying errors fast. Finally,

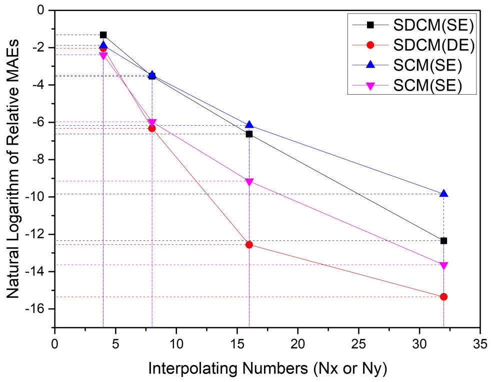
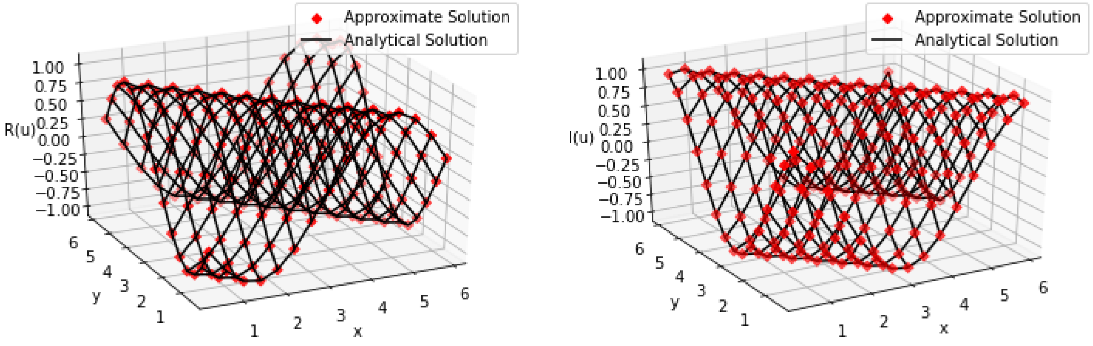


Figure 5: Natural logarithm of MAEs concerning different values of N_x for approximating interpolation nodes with dt=0.001 at t_{final}=1.



(a) Real parts

(b) Imaginary parts

Figure 6: Surface plots of real parts and imaginary parts of approximate and exact solutions with $N_x = 10$ and $dt = 1/1000$ for uniform grids with the step $ds = 1/16$ by the SDCM(DE) at the final time $t_{final} = 5$.

Table 3: Numerical results for uniform points with the time step $dt = 0.001$ at $t_{final} = 1$.

P	Methods	$\|U_\epsilon\|_\infty$	$N_x(N_y)$	CPU Times(s)
	MSM [25]	5.8054E-03		
10×10	SCM(DE)	4.6018E-03	7	8.02
	SDCM(DE)	4.8858E-03	7	11.52
	MSM [25]	1.2745E-03		
15×15	SCM(DE)	4.7933E-03	7	6.95
	SDCM(DE)	4.9144E-03	7	10.66
	MSM [25]	3.2816E-04		
20×20	SCM(DE)	6.7712E-04	11	56.17
	SDCM(DE)	4.9011E-04	9	28.11
	MSM [25]	8.9240E-05		
25×25	SCM(DE)	8.5849E-05	17	397.39
	SDCM(DE)	7.2551E-05	11	60.34

the surface plots for real and imaginary parts by the SDCM(DE) at $t_{final} = 5$ are shown in Fig. 6.

Example 4.4. In this example, considering for $x,y \in [0,1]^2$ and $t \geq 0$:

$$iu_t + u_{xx} + u_{yy} + (2\pi^2 - 1)(1 - \cos^2 \pi x \cos^2 \pi y) + (2\pi^2 - 1)|u|^2 u = 0, \tag{4.7}$$

where the exact solution of this example is [27]:

$$u(x,y,t) = \exp(-it)\cos\pi x \cos\pi y.$$

The initial and boundary conditions are from the exact solution.

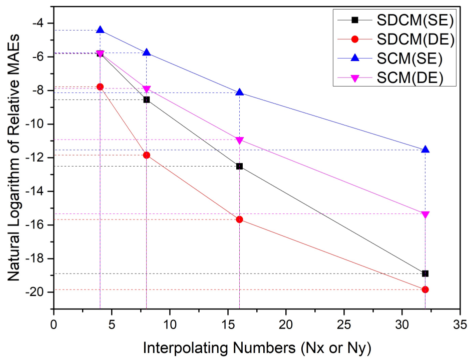
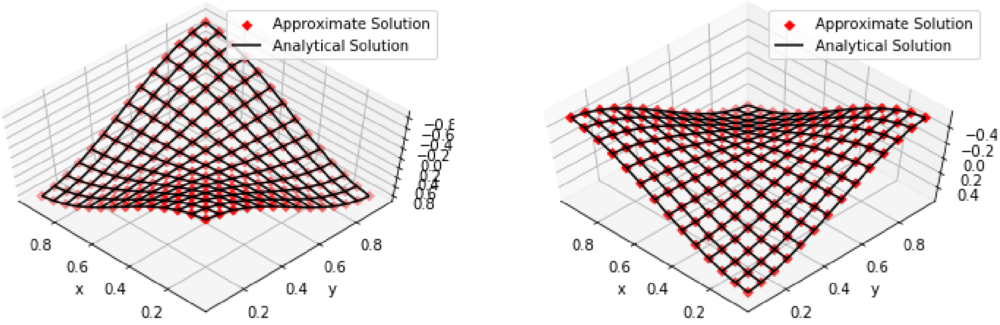


Figure 7: Natural logarithm of MAEs concerning different values of N_x for approximating interpolation nodes with $dt=0.001$ at $t_{final}=1$.



(a) Real parts

(b) Imaginary parts

Figure 8: Surface plots of real parts and imaginary parts of approximate and exact solutions with $N_x=10$ and $dt=1/100$ for uniform grids with the step $ds=1/15$ by the SDCM(DE) at the final time $t_{final}=10$.

As previous examples, Fig. 7 demonstrates that the natural logarithm of MAEs for all approaches decreases with interpolating more nodes at $t_{final}=1$. It is consistent with the results from the above examples in the quasi-exponential convergence rate and the best performance of the SDCM(DE). Moreover, in order to compare with the numerical errors by the Spectral Meshless Radial Point Interpolation Sheme (SMRPIS) in [27], the relative error defined as follows is used to demonstrate the accuracy, as well as for the real and imaginary parts (denoted as $R_2(R_\varepsilon)$ and $R_2(I_\varepsilon)$, receptively):

$$R_2(u_\varepsilon)=\sqrt{\frac{\sum(u_{ex}(\mathbf{x}_i,t^n)-u_{ap}(\mathbf{x}_i,t^n))^2}{\sum(u_{ex}(\mathbf{x}_i,t^n))^2}}.$$

Table 4 compares the numerical errors for approximating the uniform grids with different space and time steps. The numbers of interpolating nodes and time costs are also

Table 4: Numerical results of maximum absolute errors and relative errors for real and imaginary parts with different dt, ds at $t_{final}=1$ on $[0,1]^2$ for test points.

(ds,dt)	Methods	Real Parts		Imaginary Parts		N_x/CPU(s)
		$\|R_\varepsilon\|_\infty$	$R_2(R_\varepsilon)$	$\|I_\varepsilon\|_\infty$	$R_2(I_\varepsilon)$	
$(\frac{1}{14},\frac{1}{14})$	SMRPIS [27]	9.1304E-06	1.6899E-05	2.4102E-05	2.8643E-05	
	SCM(DE)	6.9940E-06	9.6897E-06	1.1884E-05	9.3205E-06	15/3.44
	SDCM(DE)	3.9993E-06	3.6645E-06	7.5715E-06	6.6854E-06	10/0.63
$(\frac{1}{20},\frac{1}{100})$	SMRPIS [27]	6.3766E-06	1.1802E-05	5.5303E-06	6.5722E-06	
	SCM(DE)	4.8949E-06	1.1339E-05	6.6798E-06	8.8761E-06	17/37.63
	SDCM(DE)	2.0870E-06	1.6374E-06	3.2033E-06	2.5025E-06	10/4.67
$(\frac{1}{26},\frac{1}{1000})$	SMPRIS [27]	1.0660E-06	1.9730E-06	3.1469E-06	3.7397E-06	
	SCM(DE)	4.1444E-06	7.2002E-06	7.2666E-06	1.0741E-05	18/612.56
	SDCM(DE)	2.1343E-06	1.8037E-06	2.8807E-06	2.2695E-06	10/50.13
$(\frac{1}{32},\frac{1}{2000})$	SMPRIS [27]	4.1518E-07	7.6843E-07	1.1395E-06	1.3542E-06	
	SCM(DE)	4.7138E-06	7.6143E-06	7.7735E-06	1.1124E-05	18/1233.52
	SDCM(DE)	6.7301E-07	5.9128E-07	1.0670E-06	8.3205E-07	12/176.84

given for Sinc numerical methods when they attain the same precision with the SMRPIS. Again, the SDCM is not necessary to interpolate as many nodes as the SCM, which economizes computational costs. Finally, the surface plots for simulating uniform grids at $t_{final}=10$ are plotted in Fig. 8.

Example 4.5. We consider the 2D-GPE concerned on $[-1,1]\times[-1,1]$:

$$iU_t-U_{xx}-U_{yy}+(1-4\cos^2 2x\cos^2 2y)u+4|u|^2u=0. \tag{4.8}$$

It admits a breather solution $u(x,y,t)=e^{9it}\cos 2x\cos 2y$ [24] from which the initial and boundary conditions are obtained.

Table 5 records the numerical errors and CPU times for approximating interior Sinc collocation nodes. It similarly demonstrates that Sinc numerical methods are decaying with interpolating more nodes. Additionally, the developed SDCM decays errors faster than the SCM, and the DE induces the collocation methods more accurately. This example also tests the accuracy for approximating 50 irregular nodes following the uniform distribution in the considered domain. Table 6 lists the numerical errors for simulating 50 irregular nodes in the considering domain at different time steps. In every test experiment, the irregular nodes are randomly selected. That means the numerical result is not affected by the selection of approximating points. Our proposed Sinc numerical methods avoid the limitation of regular grids.

Further, in Fig. 9, the simulation for irregular nodes in 3D plots is consistent with the surface wireframe from the analytical solutions at $t_{final}=10$. Finally, for the accuracy test, the convergence order is calculated by the formula $\frac{\ln(Err_1/Err_2)}{\ln(step_1/step_2)}$ [25], where $Err_j(k=1,2)$ is the error with the step size $step_j$. To better capture the error convergence characteristics

Table 5: Accuracy comparison for approximating Sinc nodes with different values of N_x and $dt=0.0001$ at $t_{final}=0.01$.

Methods	$N_x\times N_y$	$\|\|R_\varepsilon\|\|_\infty$	$\|\|I_\varepsilon\|\|_\infty$	$\|\|u_\varepsilon\|\|_\infty$	CPU(s)
SCM(SE)	8×8	4.0190E-03	2.7322E-03	4.3688E-03	3.27
	16×16	3.0989E-04	2.1738E-04	3.1083E-04	54.05
	32×32	1.1857E-05	4.6723E-06	1.1927E-05	1815.69
SCM(DE)	8×8	3.7598E-04	1.7887E-04	3.7610E-04	7.61
	16×16	2.9241E-05	9.9998E-06	2.9798E-05	48.77
	32×32	2.1281E-07	1.4574E-07	2.1698E-07	1789.59
SDCM(SE)	8×8	1.3027E-03	2.2843E-03	2.2844E-03	7.27
	16×16	3.0242E-05	3.9038E-05	3.9210E-05	58.63
	32×32	1.2851E-07	1.1471E-07	1.3340E-07	2066.67
SDCM(DE)	8×8	5.2350E-05	9.1057E-05	9.7224E-05	2.67
	16×16	4.4273E-07	3.6694E-07	4.4421E-07	68.47
	32×32	2.8725E-09	6.0148E-09	6.0401E-09	1876.78

Table 6: Accuracy comparison for approximating irregular nodes with $N_x=16$ and $dt=0.01$ at different final times.

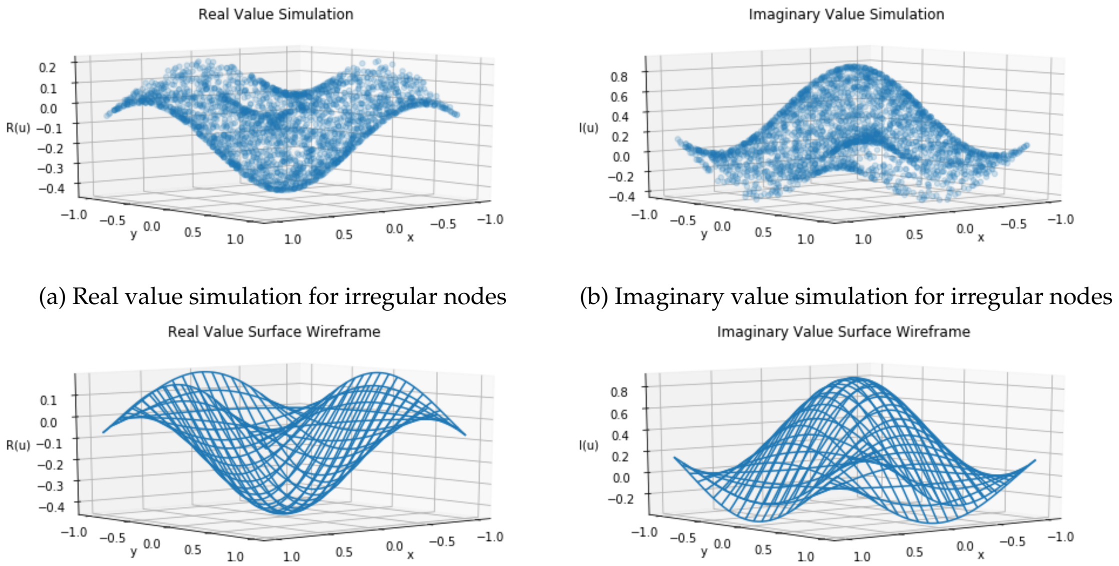
t_{final}	Methods	$\|\|R_\varepsilon\|\|_\infty$	$\|\|I_\varepsilon\|\|_\infty$	$\|\|u_\varepsilon\|\|_\infty$	CPU(s)
0.1	SDCM(SE)	4.8615E-04	3.1784E-04	5.8082E-04	18.73
	SDCM(DE)	4.9046E-04	2.8186E-04	5.6556E-04	23.84
1	SDCM(SE)	6.5792E-03	1.6676E-03	6.7873E-03	91.53
	SDCM(DE)	6.5533E-03	1.6937E-03	6.7686E-03	50.67
10	SDCM(SE)	2.2780E-02	3.0504E-02	3.8071E-02	365.20
	SDCM(DE)	2.2337E-02	3.0400E-02	3.7724E-02	349.16

in spatial discretization, a small time step $dt=0.00005$ is chosen. The L^∞ errors defined in Eq. (4.1) and the convergence orders by all Sinc numerical methods are presented in Table 7, where the step size $step_k$ equals the step length in one dimension divided by the number of N_x. Since Examples 4.1 and 4.2 belong to the SCE without the cubic nonlinearity, the convergence order is only evaluated for the last examples as the GPE. It demonstrates that Sinc numerical methods have good accuracy in spatial discretization.

In all, the five examples demonstrate the effectiveness and efficiency of Sinc numerical methods with the high accuracy.

5 Conclusions

This study approximates the time-dependent Gross–Pitaevskii equation in two dimensions within a finite domain under the non-homogeneous Dirichlet boundary conditions by discretizing in time and space. A new Sinc numerical method, the Sinc-Derivative Collocation Method (SDCM), with a unique boundary treatment is developed in this study to compare with the Sinc Collocation Method (SCM) and other similar numerical tech-



(a) Real value simulation for irregular nodes

(b) Imaginary value simulation for irregular nodes

(c) Real value surface wireframe

(d) Imaginary value surface wireframe

Figure 9: Comparison between the approximation by the SDCM(DE) with the analytical solution under $N_x = 16$ and $dt = 0.01$ at $t_{final} = 10$.

Table 7: Numerical Results and Convergence Orders by the SCM(SE) and the SDCM(SE) with $dt = 0.00005$ at $t_{final} = 0.005$.

$N_x \times N_y$	SCM(SE)	Order	SDCM(SE)	Order
Example 4.3				
4×4	5.6032E-02		1.6223E-01	
8×8	7.4746E-03	2.91	1.8948E-02	3.10
16×16	8.9059E-04	3.07	6.9198E-04	4.78
32×32	2.5708E-05	5.11	3.8200E-06	7.50
Example 4.4				
4×4	1.3199E-02		2.8191E-03	
8×8	2.5434E-03	2.38	3.6227E-04	2.96
16×16	2.1151E-04	3.59	5.9282E-06	5.93
32×32	7.2916E-06	4.86	7.8308E-09	9.56
Example 4.5				
4×4	2.0582E-02		2.8216E-02	
8×8	4.1319E-03	2.32	2.3114E-03	3.61
16×16	3.7140E-04	3.48	4.6325E-05	5.64
32×32	1.0451E-05	5.15	1.2779E-07	8.50

niques. Two exponential transformations are applied for the Sinc numerical methods. As demonstrated in all examples, the double exponential transformation induces Sinc numerical methods faster decaying errors than the single one. Both of the Sinc numerical methods exhibit the quasi-exponential convergence rate. Moreover, the SDCM can interpolate fewer nodes than the SCM to attain the same numerical results, which saves the computational costs.

In the future, to demonstrate the wide range of applicability of our proposed methods, it is expected to solve boundary value problems with other types of general boundary conditions, such as the Neumann boundary condition, the Robin boundary condition, and the Mixed boundary condition. With this regard, the first-order derivative of the approximated function should be defined on the boundary to solve ordinary or partial differential equations under those three boundary conditions. The SDCM satisfies the requirement, while the first-order derivative of the approximated function Eq. (3.24) in the SCM is not defined on the boundary. That indicates that our developed SDCM overcomes the difficulty of the traditional SCM to solve differential equations only with the Dirichlet boundary condition. To achieve that, the procedure mentioned in Section 3.3, as the matrix equation system Eq. (3.41) combined by the boundary and the interior treatment equations, can be easily implemented to yield the numerical solution as well, whereas the boundary treatment equation follows different boundary conditions. Thus, the developed SCDM is available for solving boundary value problems no matter which situations, and it is significant to test whether the SDCM retains the high accuracy advantage under different boundary conditions.

Acknowledgements

The authors thank the editor and anonymous reviewers for their constructive comments on the manuscript. This research was supported in part by the Natural Sciences and Engineering Research Council of Canada, and the National Natural Science Foundation of China (Nos. 11771322, and 72071140).

References

[1] M. Ma, C. Dang, and Z. Huang, *Analytical expressions for dark soliton solution of a gross–pitaevskii equation,* Appl. Math. Comput., 273 (2016), pp. 383–389.

[2] H. Wang, X. Ma, J. Lu, and W. Gao, *An efficient time-splitting compact finite difference method for Gross–Pitaevskii equation,* Appl. Math. Comput., 297 (2017), pp. 131–144.

[3] P. G. Kevrekidis, D. J. Frantzeskakis, and R. Carretero-González, Emergent Nonlinear Phenomena in Bose-Einstein Condensates: Theory and Experiment, Vol. 45, Springer Science & Business Media, 2007.

[4] J. Rogel-Salazar, *The Gross–Pitaevskii equation and Bose–Einstein condensates,* Euro. J. Phys., 34(2) (2013), 247.

[5] E. P. Gross, *Structure of a quantized vortex in Boson systems,* Nuovo Cimento, 20(3) (1961), pp. 454–477.

[6] L. P. Pitaevskii, *Vortex lines in an imperfect Bose gas,* Soviet Phys. Jetp. Ussr, (1961).

[7] J. FANG, B. WU, AND W. LIU, *An explicit spectral collocation method using nonpolynomial basis functions for the time-dependent schrödinger equation,* Math. Methods Appl. Sci., 42(1) (2019), pp. 186–203.

[8] M. CALIARI, AND S. ZUCCHER, *Reliability of the time splitting Fourier method for singular solutions in quantum fluids,* Comput. Phys. Commun., 222 (2018), pp. 46–58.

[9] X. ANTOINE, J. SHEN, AND Q. TANG, *Scalar auxiliary variable/lagrange multiplier based pseudospectral schemes for the dynamics of nonlinear Schrödinger/Gross-Pitaevskii equations,* J. Comput. Phys., 437 (2021), 110328.

[10] X. ANTOINE, W. BAO, AND C. BESSE, *Computational methods for the dynamics of the nonlinear Schrödinger/Gross–Pitaevskii equations,* Comput. Phys. Commun., 184(12) (2013), pp. 2621–2633.

[11] Y. HONG, J. LU, J. LIN, AND W. CHEN, *Numerical simulation of non-linear Schrödinger equations in arbitrary domain by the localized method of approximate particular solution,* Adv. Appl. Math. Mech., 11(1) (2019), pp. 108–131.

[12] M. DEHGHAN, AND D. MIRZAEI, *The meshless local Petrov–Galerkin (MLPG) method for the generalized two-dimensional non-linear Schrödinger equation,* Eng. Anal. Boundary Elements, 32(9) (2008), pp. 747–756.

[13] M. DEHGHAN, *Finite difference procedures for solving a problem arising in modeling and design of certain optoelectronic devices,* Math. Comput. Simulation, 71(1) (2006), pp. 16–30.

[14] W. BAO, AND Y. CAI, *Uniform error estimates of finite difference methods for the nonlinear Schrödinger equation with wave operator,* SIAM J. Numer. Anal., 50(2) (2012), pp. 492–521.

[15] L. ZHEN, Y. BAI, Q. LI, AND K. WU, *Symplectic and multisymplectic schemes with the simple finite element method,* Phys. Lett. A, 314(5-6) (2003), pp. 443–455.

[16] H. LI, Z. MU, AND Y. WANG, *An energy-preserving Crank–Nicolson Galerkin spectral element method for the two dimensional nonlinear Schrödinger equation,* J. Comput. Appl. Math., 344 (2018), pp. 245–258.

[17] P. HEID, B. STAMM, AND T. P. WIHLER, *Gradient flow finite element discretizations with energy-based adaptivity for the Gross-Pitaevskii equation,* J. Comput. Phys., 436 (2021), 110165.

[18] N. İ. KARABAŞ, S. Ö. KORKUT, G. TANOĞLU, AND I. AZIZ, ET AL., *An efficient approach for solving nonlinear multidimensional Schrödinger equations,* Eng. Anal. Boundary Elements, 132 (2021), pp. 263–270.

[19] Z.-J. FU, Z.-Y. XIE, S.-Y. JI, C.-C. TSAI, AND A.-L. LI, *Meshless generalized finite difference method for water wave interactions with multiple-bottom-seated-cylinder-array structures,* Ocean Eng., 195 (2020), 106736.

[20] H. XIA, AND Y. GU, *Generalized finite difference method for electroelastic analysis of three-dimensional piezoelectric structures,* Appl. Math. Lett., 117 (2021), 107084.

[21] Y. GU, C.-M. FAN, AND Z. FU, *Localized method of fundamental solutions for three-dimensional elasticity problems: theory,* Adv. Appl. Math. Mech., 13(6) (2021), pp. 1520–1534.

[22] J. GAIDAMOUR, Q. TANG, AND X. ANTOINE, *Bec2hpc: A hpc spectral solver for nonlinear Schrödinger and rotating gross-pitaevskii equations. stationary states computation,* Comput. Phys. Commun., 265 (2021), 108007.

[23] M. DEHGHAN, AND F. EMAMI-NAEINI, *The sinc-collocation and sinc-galerkin methods for solving the two-dimensional Schrödinger equation with nonhomogeneous boundary conditions,* Appl. Math. Model., 37(22) (2013), pp. 9379–9397.

[24] X. LI, AND L. ZHANG, *An efficient spectral-collocation difference method for two-dimensional Schrödinger equation with Neumann boundary conditions,* Comput. Math. Appl., 79(8) (2020), pp. 2322–2335.

[25] Z. Sun, *A meshless symplectic method for two-dimensional nonlinear Schrödinger equations based on radial basis function approximation,* Eng. Anal. Boundary Elements, 104 (2019), pp. 1–7.

[26] S. Abbasbandy, H. R. Ghehsareh, and I. Hashim, *A meshfree method for the solution of two-dimensional cubic nonlinear Schrödinger equation,* Eng. Anal. Boundary Elements, 37(6) (2013), pp. 885–898.

[27] E. Shivanian, and A. Jafarabadi, *An efficient numerical technique for solution of two-dimensional cubic nonlinear Schrödinger equation with error analysis,* Eng. Anal. Boundary Elements, 83 (2017), pp. 74–86.

[28] Q. Liu, C. Fan, and B. Šarler, *Localized method of fundamental solutions for two-dimensional anisotropic elasticity problems,* Eng. Anal. Boundary Elements, 125 (2021), pp. 59–65.

[29] X. Yue, F. Wang, P.-W. Li, and C.-M. Fan, *Local non-singular knot method for large-scale computation of acoustic problems in complicated geometries,* Comput. Math. Appl., 84 (2021), pp. 128–143.

[30] Q. Xi, Z. Fu, C. Zhang, and D. Yin, *An efficient localized trefftz-based collocation scheme for heat conduction analysis in two kinds of heterogeneous materials under temperature loading,* Comput. Struct., 255 (2021), 106619.

[31] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, Spectral Methods: Fundamentals in Single Domains, Springer Science & Business Media, 2007.

[32] K. Abdella, *Numerical solution of two-point boundary value problems using sinc interpolation,* in: Proceedings of the American Conference on Applied Mathematics (American-Math'12): Applied Mathematics in Electrical and Computer Engineering, (2012), pp. 157–162.

[33] K. Abdella, *Solving differential equations using sinc-collocation methods with derivative interpolations,* J. Comput. Methods Sci. Eng., 15(3) (2015), pp. 305–315.

[34] Y. Mohseniahouei, K. Abdella, and M. Pollanen, *An application of the sinc-collocation method to a three-dimensional oceanography model,* World Academy of Science, Engineering and Technology, International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering 7(5) (2013), pp. 777–783.

[35] J. Lund, and K. L. Bowers, Sinc Methods for Quadrature and Differential Equations, Vol. 32, SIAM, 1992.

[36] W. Dos Passos, Numerical Methods, Algorithms and Tools in C, (2016).

[37] J. W. Brown, and R. V. Churchill, Complex Variables and Applications Eighth Edition, McGraw-Hill Book Company, 2009.

[38] A. Fahim, M. A. F. Araghi, J. Rashidinia, and M. Jalalvand, *Numerical solution of Volterra partial integro-differential equations based on sinc-collocation method,* Adv. Difference Equations, (1) (2017), 362.

[39] C. Li, and X. Wu, *Numerical solution of differential equations using sinc method based on the interpolation of the highest derivatives,* Appl. Math. Model., 31(1) (2007), pp. 1–9.

[40] M. Sugihara, and T. Matsuo, *Recent developments of the sinc numerical methods,* J. Comput. Appl. Math., 164 (2004), pp. 673–689.

[41] R. Johansson, Numerical Python: a Practical Techniques Approach for Industry, Apress, 2015.

[42] Z. Jia, *Approximation accuracy of the krylov subspaces for linear discrete ill-posed problems,* J. Comput. Appl. Math., 374 (2020) 112786.

[43] N. Pervaiz, and I. Aziz, *Haar wavelet approximation for the solution of cubic nonlinear schrodinger equations,* Phys. A Stat. Mech. Appl., 545 (2020), 123738.