

RESEARCH

Open Access

An efficient computer application of the sinc-Galerkin approximation for nonlinear boundary value problems

Aydin Secer^{1*}, Muhammet Kurulay², Mustafa Bayram¹ and Mehmet Ali Akinlar³

*Correspondence:
asecer@yildiz.edu.tr

¹Department of Mathematical Engineering, Faculty of Chemical and Metallurgical Engineering, Yildiz Technical University, Davutpasa, İstanbul, 34210, Turkey
Full list of author information is available at the end of the article

Abstract

A powerful technique based on the sinc-Galerkin method is presented for obtaining numerical solutions of second-order nonlinear Dirichlet-type boundary value problems (BVPs). The method is based on approximating functions and their derivatives by using the Whittaker cardinal function. Without any numerical integration, the differential equation is reduced to a system of algebraic equations *via* new accurate explicit approximations of the inner products; therefore, the evaluation is based on solving a matrix system. The solution is obtained by constructing the nonlinear (or linear) matrix system using Maple and the accuracy is compared with the Newton method. The main aspect of the technique presented here is that the obtained solution is valid for various boundary conditions in both linear and nonlinear equations and it is not affected by any singularities that can occur in variable coefficients or a nonlinear part of the equation. This is a powerful side of the method when being compared to other models.

Keywords: Maple; sinc-Galerkin approximation; sinc basis function; nonlinear matrix system; Newton method

1 Introduction

We present here the sinc-Galerkin approximation technique using Maple to solve systems of nonlinear BVPs such as

$$\begin{aligned} P(x)y'' + Q(x)y' + R(x)NL(y) &= F(x), \\ y(a) = 0, \quad y(b) &= 0, \end{aligned} \tag{1.1}$$

where NL is the nonlinear part of Eq. (1.1) which can take any form of nonlinearity, and we investigate the approximate solution on some closed interval $[a, b]$ in R .

We start by casting a given linear or nonlinear BVP into a sinc-Galerkin form accurate to the order $O(N^{1/2}e^{-(\pi daN)^{1/2}})$ [1]. This discretization yields a set of linear or nonlinear algebraic equations that include all unknown coefficients. These equations are expressed in a nonlinear or linear matrix form depending on (1.1). If the equation is linear, the LU decomposition method can be used to find unknown coefficients. However, if it is not linear, the coefficients can be found by the Newton interpolation method for nonlinear equation systems by using Maple. The methodology is illustrated on nonlinear ordinary differential equations with Dirichlet-type boundaries. Once the solution is obtained, we

compare its accuracy with the Newton method as a graphical and numerical simulation by using Maple.

We start with some literature on the sinc-Galerkin methods. The sinc methods were introduced in [2] and expanded in [1] by Frank Stenger. Sinc functions were first analyzed in [3] and [4]. An extensive research of sinc methods for two-point boundary value problems can be found in [5, 6]. In [7, 8] parabolic and hyperbolic problems are discussed in detail. Some kind of singular elliptic problems are solved in [9], and the symmetric sinc-Galerkin method is introduced in [10]. The sinc domain decomposition is presented in [11–14]. Also, iterative methods for symmetric sinc-Galerkin systems are given in [15–17]. Sinc methods are studied thoroughly in [18]. Applications of sinc methods can also be found in [19–21]. The article [22] summarizes the results that are obtained by sinc numerical methods of computation. In [14] a numerical solution of the Volterra integro-differential equation by means of the sinc collocation method is considered. The paper [1] illustrates the application of a sinc-Galerkin method to the approximate solution of linear and nonlinear second-order ordinary differential equations and to the approximate solution of some linear elliptic and parabolic partial differential equations in the plane. The fully sinc-Galerkin method is developed for a family of complex-valued partial differential equations with time-dependent boundary conditions [19]. In [23] some novel procedures of using sinc methods to compute the solutions of three types of medical problems are illustrated. In [24], the sinc-based algorithm is used to solve a nonlinear set of partial differential equations. A new sinc-Galerkin method is developed for approximating the solution of convection diffusion equations with mixed boundary conditions on half-infinite intervals in [25]. The work which is presented in [26] deals with the sinc-Galerkin method for solving nonlinear fourth-order differential equations with homogeneous and nonhomogeneous boundary conditions. In [27], the sinc methods are used to solve second-order ordinary differential equations with homogeneous Dirichlet-type boundary conditions. In the paper given in [28], the sinc-Galerkin method is applied to solving Troesch's problem. The properties of the sinc procedure are utilized to reduce the computation of Troesch's equation to nonlinear equations with unknown coefficients.

2 Sinc basis functions

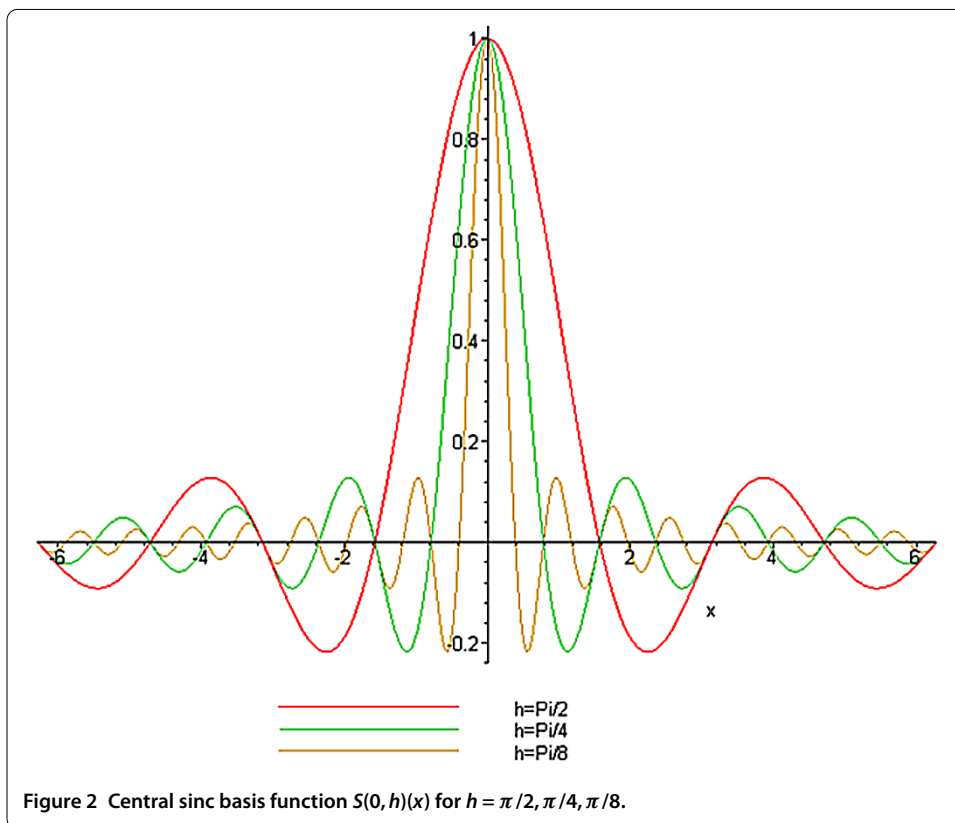
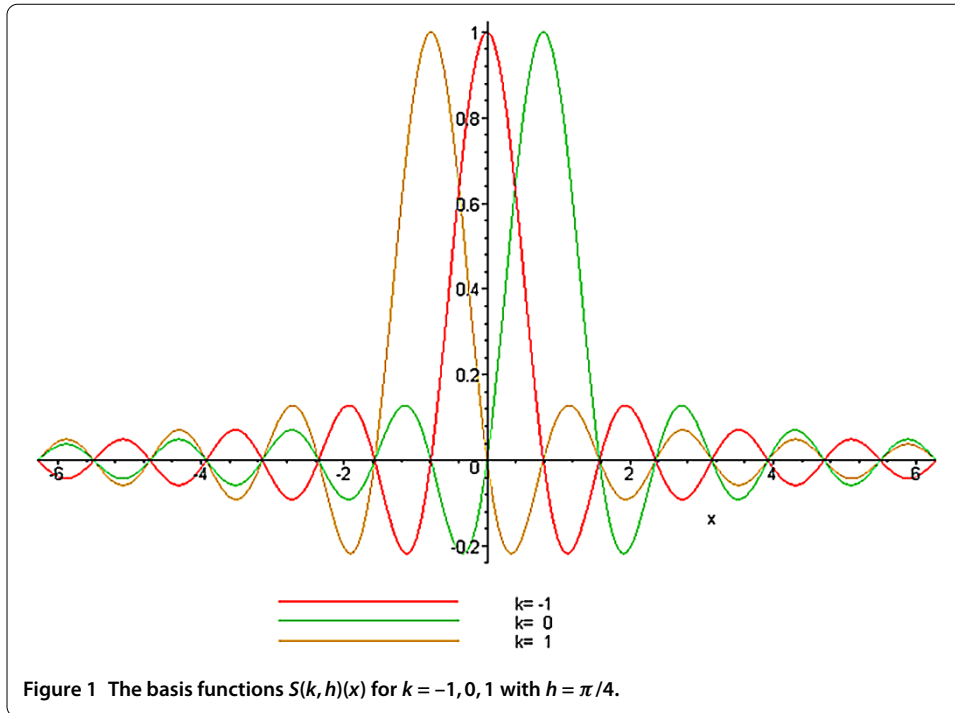
Let C denote the set of all complex numbers, and for all $z \in C$ define the sine cardinal or sinc function by

$$\operatorname{sinc}(z) = \begin{cases} \frac{\sin(\pi z)}{\pi z}, & y \neq 0, \\ 1, & y = 0. \end{cases} \tag{2.1}$$

For $h > 0$, the translated sinc function with evenly spaced nodes is given by

$$\operatorname{sinc}(k, h)(z) = \begin{cases} \frac{\sin(\pi \frac{z-kh}{h})}{\pi \frac{z-kh}{h}}, & z \neq kh, \\ 1, & z = kh. \end{cases} \tag{2.2}$$

For various values of k , the sinc basis function $S(k, \pi/4)(x)$ on the whole real line, $-\infty < x < \infty$, is illustrated in Figure 1. For various values of h , the central function $S(0, h)(x)$ is illustrated in Figure 2.



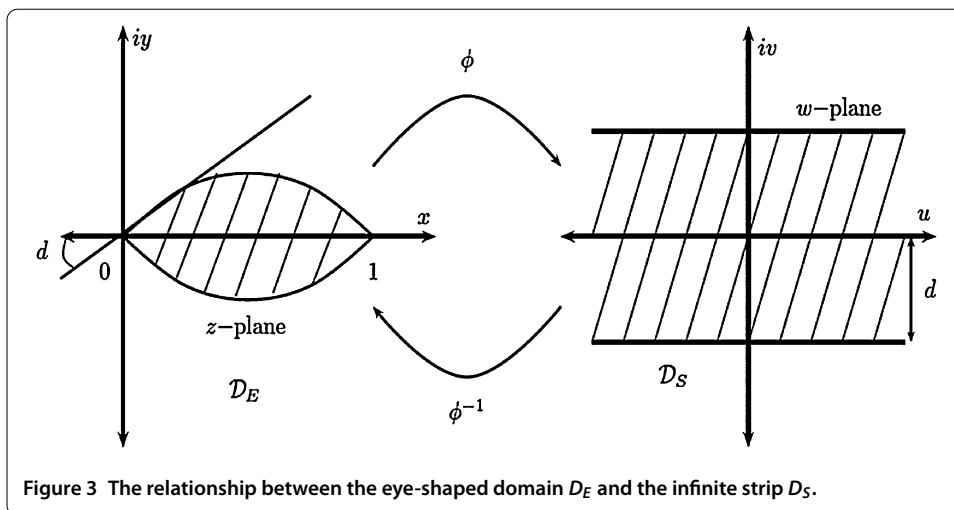


Figure 3 The relationship between the eye-shaped domain D_E and the infinite strip D_S .

If a function $f(x)$ is defined over the real line, then for $h > 0$ the series

$$C(f, h)(x) = \sum_{k=-\infty}^{\infty} f(kh) \operatorname{sinc}\left(\frac{x - kh}{h}\right) \tag{2.3}$$

is called the Whittaker cardinal expansion of f whenever this series converges. The infinite strip D_S of the complex w plane, where $d > 0$, is given by

$$D_S \equiv \left\{ w = u + iv : |v| < d \leq \frac{\pi}{2} \right\}. \tag{2.4}$$

In general, approximations can be constructed for infinite, semi-infinite, and finite intervals. Define the function

$$w = \phi(z) = \ln\left(\frac{z}{1 - z}\right) \tag{2.5}$$

which is a conformal mapping from D_E , the eye-shaped domain in the z -plane, onto the infinite strip D_S , where

$$D_E = z = \left\{ x + iy : \left| \arg\left(\frac{z}{1 - z}\right) \right| < d \leq \frac{\pi}{2} \right\}. \tag{2.6}$$

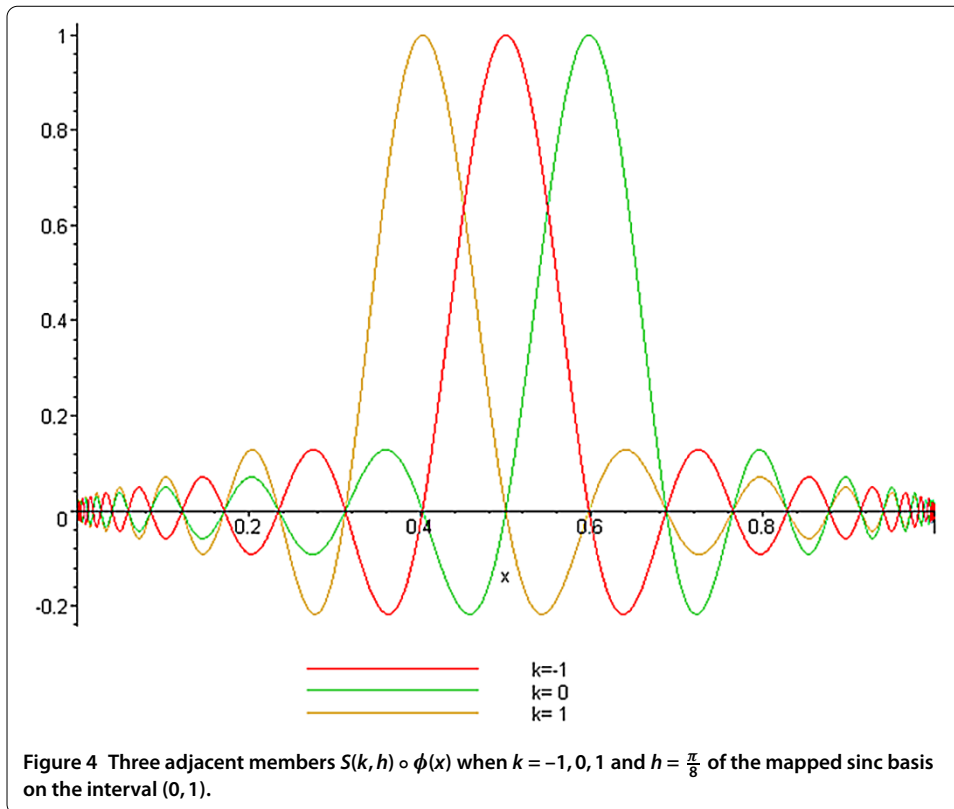
This is shown in Figure 3.

For the sinc-Galerkin method, the basis functions are derived from the composite translated sinc functions:

$$S_h(z) = S(k, h)(z) = \operatorname{sinc}\left(\frac{\phi(z) - kh}{h}\right) \tag{2.7}$$

for $z \in D_E$. These are shown in Figure 4 for real values of x . The function $z = \phi^{-1}(w) = \frac{e^w}{1 + e^w}$ is an inverse mapping of $w = \phi(z)$. We may define the range of ϕ^{-1} on the real line as

$$\Gamma = \{ \phi^{-1}(u) \in D_E : -\infty < u < \infty \}. \tag{2.8}$$



For the evenly spaced nodes $\{kh\}_{k=-\infty}^{\infty}$ on the real line, the image which corresponds to these nodes is denoted by

$$x_k = \phi^{-1}(kh) = \frac{e^{kh}}{1 + e^{kh}}. \tag{2.9}$$

A list of conformal mappings may be found in Table 2.1 [6].

Definition 2.1 Let D_E be a simply connected domain in the complex plane C , and let ∂D_E denote the boundary of D_E . Let a, b be points on ∂D_E and ϕ be a conformal map D_E onto D_S such that $\phi(a) = -\infty$ and $\phi(b) = \infty$. If the inverse map of ϕ is denoted by φ , define

$$\Gamma = \{\phi^{-1}(u) \in D_E : -\infty < u < \infty\} \tag{2.10}$$

and $z_k = \varphi(kh), k = \mp 1, \mp 2, \dots$

We can use Table 1 to choose convenient conformal map according to boundary conditions.

Definition 2.2 Let $B(D_E)$ be the class of functions F that are analytic in D_E and satisfy

$$\int_{\psi(L+u)} |F(z)| dz \rightarrow 0, \quad \text{as } u = \mp \infty, \tag{2.11}$$

Table 1 Conformal mappings and nodes for several subintervals of R

(a, b)	$\phi(z)$	z_k	
a	b	$\ln\left(\frac{z-a}{b-z}\right)$	$\frac{a+be^{kh}}{1+e^{kh}}$
0	1	$\ln\left(\frac{z}{1-z}\right)$	$\frac{e^{kh}}{1+e^{kh}}$
0	∞	$\ln(z)$	e^{kh}
0	∞	$\ln(\sinh(z))$	$\ln(e^{kh} + \sqrt{e^{2kh} + 1})$
$-\infty$	∞	z	kh
$-\infty$	∞	$\sinh^{-1}(z)$	kh

where

$$L = \left\{ iy : |y| < d \leq \frac{\pi}{2} \right\} \tag{2.12}$$

and on the boundary of D_E it satisfies

$$T(F) = \int_{\partial D_E} |F(z) dz| < \infty. \tag{2.13}$$

The proof of following theorems can be found in [1].

Theorem 2.1 *Let Γ be $(0, 1)$, $F \in B(D_E)$, then for $h > 0$ sufficiently small,*

$$\int_{\Gamma} F(z) dz - h \sum_{j=-\infty}^{\infty} \frac{F(z_j)}{\phi'(z_j)} = \frac{i}{2} \int_{\partial D} \frac{F(z)k(\phi, h)(z)}{\sin(\pi\phi(z)/h)} dz \equiv I_F, \tag{2.14}$$

where

$$|k(\phi, h)|_{z \in \partial D} = \left| e^{\left[\frac{i\pi\phi(z)}{h} \operatorname{sgn}(\operatorname{Im}\phi(z)) \right]} \right|_{z \in \partial D} = e^{-\frac{\pi d}{h}}. \tag{2.15}$$

For the sinc-Galerkin method, the infinite quadrature rule must be truncated to a finite sum. The following theorem indicates the conditions under which an exponential convergence results.

Theorem 2.2 *If there exist positive constants α, β and C such that*

$$\left| \frac{F(x)}{\phi'(x)} \right| \leq C \begin{cases} e^{-\alpha|\phi(x)|}, & x \in \psi((-\infty, \infty)), \\ e^{-\beta|\phi(x)|}, & x \in \psi((0, \infty)), \end{cases} \tag{2.16}$$

then the error bound for the quadrature rule (2.14) is given by

$$\left| \int_{\Gamma} F(x) dx - h \sum_{j=-N}^N \frac{F(x_j)}{\phi'(x_j)} \right| \leq C \left(\frac{e^{-\alpha N h}}{\alpha} + \frac{e^{-\beta N h}}{\beta} \right) + |I_F|. \tag{2.17}$$

The infinite sum in (2.14) is truncated with the use of (2.16) to arrive at (2.17).

Making the selections

$$h = \sqrt{\frac{\pi d}{\alpha N}}, \tag{2.18}$$

$$N \equiv \left\lceil \left\lfloor \frac{\alpha N}{\beta} + 1 \right\rfloor \right\rceil, \tag{2.19}$$

where $\lceil \cdot \rceil$ is an integer part of the statement and N is the integer value which specifies the grid size, then

$$\int_{\Gamma} F(x) dx = h \sum_{j=-N}^N \frac{F(x_j)}{\phi'(x_j)} + O(e^{-\pi\alpha dN^{1/2}}). \tag{2.20}$$

We used Theorems 2.1 and 2.2 to approximate the integrals that arise in the formulation of the discrete systems corresponding to the second-order boundary value problem.

Theorem 2.3 *Let ϕ be a conformal one-to-one map of the simply connected domain D_E onto D_S . Then*

$$\delta_{jk}^{(0)} = [S(j, h) \circ \phi(x)]|_{x=x_k} = \begin{cases} 1, & k = j, \\ 0, & k \neq j, \end{cases} \tag{2.21}$$

$$\delta_{jk}^{(1)} = h \frac{d}{d\phi} [S(j, h) \circ \phi(x)]|_{x=x_k} = \begin{cases} 0, & k = j, \\ \frac{(-1)^{k-j}}{(k-j)}, & k \neq j, \end{cases} \tag{2.22}$$

$$\delta_{jk}^{(2)} = h \frac{d^2}{d\phi^2} [S(j, h) \circ \phi(x)]|_{x=x_k} = \begin{cases} \frac{-\pi^2}{3}, & k = j, \\ \frac{-2(-1)^{k-j}}{(k-j)^2}, & k \neq j. \end{cases} \tag{2.23}$$

3 Convergence analysis

Consider the following problem:

$$P(x)y'' + Q(x)y' + R(x)NL(y) = F(x) \tag{3.1}$$

with Dirichlet-type boundary condition

$$y(a) = 0, \quad y(b) = 0, \tag{3.2}$$

where $P, Q, R,$ and F are analytic on D . We consider sinc approximation by the formula

$$y(x) \approx y_N(x) = \sum_{k=-N}^N c_k S(k, h) \circ \phi(x), \tag{3.3}$$

$$S(k, h) = \frac{\sin[\frac{\pi}{h}(x - kh)]}{\frac{\pi}{h}(x - kh)}. \tag{3.4}$$

The unknown coefficients c_k in Eq. (3.3) are determined by orthogonalizing the residual with respect to the sinc basis functions. The Galerkin method enables us to determine the c_k coefficients by solving the nonlinear system of equations

$$(NLy_N - F, S(k, h) \circ \phi(x)) = 0, \quad k = -N, -N + 1, \dots, N - 1, N. \tag{3.5}$$

Let f_1 and f_2 be analytic functions on D . The inner product in (3.5) is defined as follows:

$$\langle f_1, f_2 \rangle = \int_{\Gamma} w(x) f_1(x) f_2(x) dx, \tag{3.6}$$

where w is the weight function. For the second-order problems, it is convenient to take [1]

$$w(x) = \frac{1}{\phi'(x)}. \tag{3.7}$$

For Eq. (3.1), we use the notations (2.21)-(2.23) together with the inner product given in (3.5) [1] to get the following approximation formulas:

$$\langle F(x), S(k, h) \circ \phi(x) \rangle = \int_{\Gamma} w(x) F(x) S(k, h) \circ \phi(x) dx \cong \frac{hw_k F_k}{\phi'_k}, \tag{3.8}$$

$$\begin{aligned} \langle R(x)NL(y(x)), S(k, h) \circ \phi(x) \rangle &= \int_{\Gamma} w(x) R(x)NL(y(x)) S(k, h) \circ \phi(x) dx \\ &\cong h \left(\frac{w_k R_k}{\phi'_k} \right) NL(c_k), \end{aligned} \tag{3.9}$$

$$\begin{aligned} \langle (Q(x)y'(x)), S(k, h) \circ \phi(x) \rangle &= \int_{\Gamma} w(x) Q(x)y'(x) S(k, h) \circ \phi(x) dx \cong h \left(\frac{w_k Q_k}{\phi'_k} \right) c'_k \\ &\cong -h \sum_{j=-N}^N c_j \left[\frac{(Qw)'_j}{\phi'_j} \delta_{kj}^{(0)} + (Qw)_j \frac{\delta_{kj}^{(0)}}{h} \right], \end{aligned} \tag{3.10}$$

$$\begin{aligned} \langle P(x)y''(x), S(k, h) \circ \phi(x) \rangle &= \int_{\Gamma} w(x) P(x)y''(x) S(k, h) \circ \phi(x) dx \cong h \left(\frac{w_k P_k}{\phi'_k} \right) c''_k \\ &\cong -h \sum_{j=-N}^N c_j \left[\frac{(Pw)''_j}{\phi'_j} \delta_{kj}^{(0)} + \left(2(Pw)'_j + \frac{(Pw)_j \phi''_j}{\phi'_j} \right) \frac{\delta_{kj}^{(1)}}{h} + (Pw)_j \phi'_j \frac{\delta_{kj}^{(2)}}{h^2} \right], \end{aligned} \tag{3.11}$$

where $w_k = w(x_k)$ etc. The choices $h = (\pi d/\alpha N)^{1/2}$ and $w(x) = 1/\phi'(x)$ yield $O(N^{1/2} \times e^{-(\pi d\alpha N)^{1/2}})$ [1] accuracy for each of the approximations in (3.8)-(3.11).

Using (3.5), (3.8)-(3.11), we obtain a nonlinear system of equations for $2N + 1$ numbers c_k .

The nonlinear system with $2N + 1$ unknowns given in (3.5) can be expressed by means of matrices. Let $m = 2N + 1$ and let $S_m, c_m, NL(c_m)$ be column vectors defined by

$$S_m(x) = \begin{pmatrix} S_{-N} \\ S_{-N+1} \\ \vdots \\ S_N \end{pmatrix}, \quad c_m = \begin{pmatrix} c_{-N} \\ c_{-N+1} \\ \vdots \\ c_N \end{pmatrix}, \quad NL(c_m) = \begin{pmatrix} NL(c_{-N}) \\ NL(c_{-N+1}) \\ \vdots \\ NL(c_N) \end{pmatrix}. \tag{3.12}$$

Let $A_m(y)$ denote a diagonal matrix whose diagonal elements are $y(x_{-N}), y(x_{-N+1}), \dots, y(x_N)$ and non-diagonal elements are zero, and also let $I_m^{(0)}, I_m^{(1)}$ and $I_m^{(2)}$ denote the matrices

$$I_m^{(0)} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} = \left[\delta_{jk}^{(0)} \right], \tag{3.13}$$

$$I_m^{(1)} = \begin{bmatrix} 0 & -1 & \frac{1}{2} & \dots & \frac{1}{2N} \\ 1 & 0 & -1 & \dots & -\frac{1}{2N-1} \\ -\frac{1}{2} & 1 & 0 & \dots & \frac{1}{2N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{2N} & \frac{1}{2N-1} & \frac{1}{2N-2} & \dots & 0 \end{bmatrix} = \left[\delta_{jk}^{(1)} \right], \tag{3.14}$$

$$I_m^{(2)} = \begin{bmatrix} -\frac{\pi^2}{3} & \frac{2}{1^2} & -\frac{2}{2^2} & \dots & -\frac{2}{(2N)^2} \\ \frac{2}{1^2} & -\frac{\pi^2}{3} & \frac{2}{1^2} & \dots & \frac{2}{(2N-1)^2} \\ -\frac{2}{2^2} & \frac{2}{1^2} & -\frac{\pi^2}{3} & \dots & -\frac{2}{(2N-2)^2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{2}{(2N)^2} & \frac{2}{(2N-1)^2} & -\frac{2}{(2N-2)^2} & \dots & -\frac{\pi^2}{3} \end{bmatrix} = \left[\delta_{jk}^{(2)} \right]. \tag{3.15}$$

With these notations, the discrete system in (3.5) takes the form:

$$\begin{aligned} & (NLy_N - F, S_m(k, h) \circ \phi(x)) \\ &= h \left[I_m^{(0)} A_m \left(\frac{(Pw)''}{\phi'} \right) + \frac{1}{h} I_m^{(1)} A_m (2(Pw)' + (Pw)\phi''/\phi') + \frac{1}{h^2} I_m^{(2)} A_m ((Pw)\phi') \right] c_m \\ & - h \left[I_m^{(0)} A_m \left(\frac{(Qw)''}{\phi'} \right) + \frac{1}{h} I_m^{(1)} A_m (Qw) \right] c_m \\ & + h \left[I_m^{(0)} A_m \left(\frac{Rw}{\phi'} \right) \right] NL(c_m) \\ & - h A_m \frac{Fw}{\phi'}. \end{aligned} \tag{3.16}$$

Theorem 3.1 *Let $c, NL(c)$ be an m -vector whose j th component is c_j and $NL(c_j)$ then the system (3.16) yields the following matrix system whose dimensions are $(2N + 1) \times (2N + 1)$:*

$$\Phi_1 \cdot c + \Phi_2 \cdot NL(c) = A_m \frac{Fw}{\phi'}. \tag{3.17}$$

Now we have a nonlinear system with $(2N + 1)$ equations in the $(2N + 1)$ unknown coefficients. If we solve (3.17) with the Newton method (for nonlinear equation systems) by using Maple, we can obtain c_j coefficients for the approximate sinc-Galerkin solution

$$y(x) \approx y_N(x) = \sum_{k=-N}^N c_k S(k, h) \circ \phi(x). \tag{3.18}$$

4 Examples

In this section, three examples are given to illustrate the performance of the sinc-Galerkin method by solving nonlinear Dirichlet-type boundary value problems. Each of these problems have been chosen to simulate how the solutions change in different zero boundary intervals. In the following examples, the discrete sinc system defined by (3.18) is used to compute the coefficients c_j ; $j = -N, \dots, N$. The computations are done by the algorithm which we developed for sinc-Galerkin method by using Maple. The algorithm automatically compares the sinc-method to the Newton method. The following examples show that the sinc-Galerkin method is a very efficient and powerful tool for nonlinear Dirichlet-type boundary value problems.

Example 4.1 Consider the following nonlinear Dirichlet-type boundary value problem on the interval $[-1, 1]$:

$$\frac{d^2}{dx^2}y(x) + 5\frac{d}{dx}y(x) + xy(x)\sin(y(x)) = -e^{x^3}\tan(x^2) + 2x^2, \tag{4.1}$$

$$y(-1) = 0, \quad y(1) = 0.$$

We choose the weight function according to [1], $\phi(x) = \ln\left(\frac{x+1}{1-x}\right)$, $w(x) = \frac{1}{\phi'(x)}$ and by taking $d = \pi/2$, $h = \frac{2}{\sqrt{N}}$, $x_k = \frac{-1+e^{kh}}{1+e^{kh}}$ for $N = 2, 4, 8, 48$, the solutions presented in Figure 5 and Table 2.

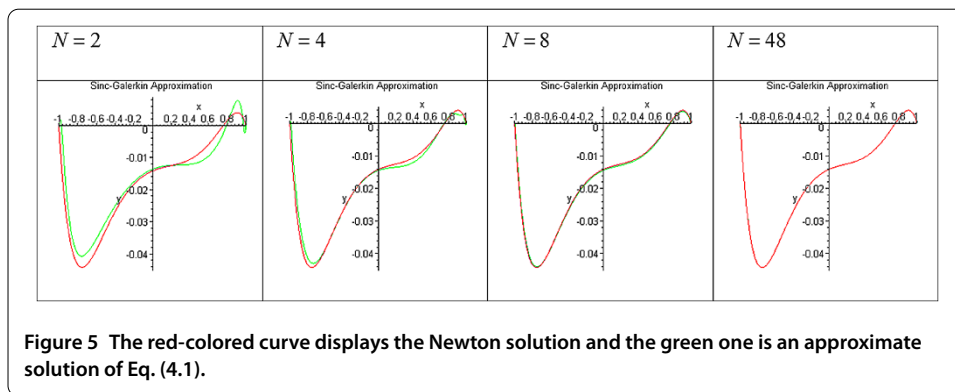


Figure 5 The red-colored curve displays the Newton solution and the green one is an approximate solution of Eq. (4.1).

Table 2 The numerical results for the approximate solutions obtained by sinc-Galerkin in comparison with the Newton solutions of Eq. (4.1) for $N = 48$

x	Newton Solution	Sinc-Galerkin	Relative Error
-0.79	-0.04367296498889709080	-0.04367277695575780	1.88033138931166478E-7
-0.59	-0.03819740808174877660	-0.03819740640312920	1.67861962497104996E-9
-0.39	-0.02633354081076150210	-0.02633360882474840	6.80139867594367620E-8
-0.19	-0.01804242984461256610	-0.01804252401568270	9.41710703577436960E-8
0.01	-0.01405035691404465420	-0.01405046008732300	1.03173278485183830E-7
0.21	-0.01240001647814625340	-0.01240012227610340	1.05797957363189950E-7
0.41	-0.01037407556626012560	-0.01037418276112410	1.07194863974841016E-7
0.61	-0.00547925725715721866	-0.00547936685525044	1.09598093266566966E-7

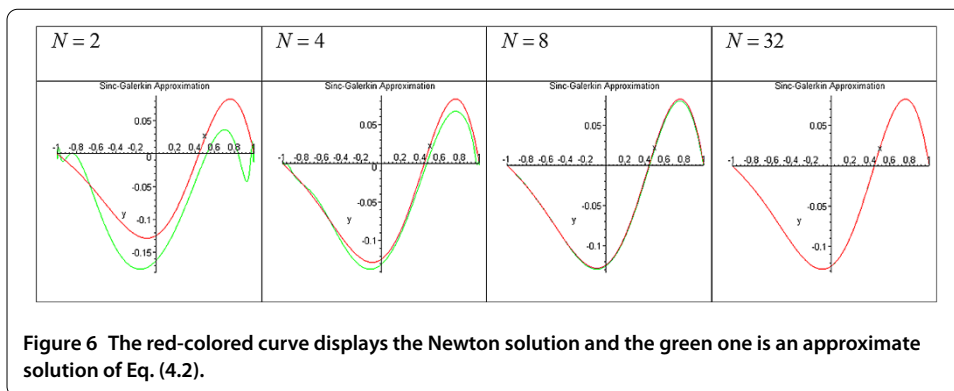


Figure 6 The red-colored curve displays the Newton solution and the green one is an approximate solution of Eq. (4.2).

Table 3 The numerical results for the approximate solutions obtained by sinc-Galerkin in comparison with the exact solutions of Eq. (4.2) for $N = 32$

x	Newton Solution	Sinc-Galerkin	Relative Error
-0.79	-0.0279198537590887030	-0.0279190996665941	7.54092494600466520E-7
-0.59	-0.0611657628394463374	-0.0611664872171184	7.24377672134392476E-7
-0.39	-0.0973239208356010965	-0.0973261954947172	2.27465911560142520E-6
-0.19	-0.1238852239083363670	-0.1238891553354690	3.93142713083890409E-6
0.01	-0.1237120026631739060	-0.1237176238268980	5.62116372328485409E-6
0.21	-0.0847168119395961262	-0.0847241308368652	7.31889726906482055E-6
0.41	-0.0126466852046787422	-0.012655978765954	8.91267191663590533E-6
0.61	0.0601106492319938846	0.0601002769211166	1.03723108773924407E-5

Example 4.2 Let us have the following form of nonlinear Dirichlet-type boundary value problem on the interval $[-1, 1]$:

$$e^{-x} \frac{d^2}{dx^2} y(x) + x \frac{d}{dx} y(x) + x^3 e^{-y(x)} = \cos(\pi x), \tag{4.2}$$

$$y(-1) = 0, \quad y(1) = 0,$$

where $\phi(x) = \ln(\frac{x+1}{1-x})$, $w(x) = \frac{1}{\phi'(x)}$ and by taking $d = \pi/2$, $h = \frac{2}{\sqrt{N}}$, $x_k = \frac{-1+e^{kh}}{1+e^{kh}}$ for $N = 2, 4, 8, 32$ we get the solutions presented in Figure 6 and Table 3.

Example 4.3 In this case, we take the problem to be given on the interval $[4, 5]$

$$\frac{d^2}{dx^2} y(x) + \frac{d}{dx} y(x) - \frac{e^{-\sin(y(x))} (y(x))^2}{1 + y(x)} = \cos(\pi x^2) x, \tag{4.3}$$

$$y(4) = 0, \quad y(5) = 0,$$

where we chose $\phi(x) = \ln(\frac{x-4}{5-x})$, $w(x) = \frac{1}{\phi'(x)}$ and by taking $d = \pi/2$, $h = \frac{2}{\sqrt{N}}$, $x_k = \frac{4+5e^{kh}}{1+e^{kh}}$ for $N = 16, 24, 32, 48$ we get the results presented in Figure 7 and Table 4.

5 Discussion

A new efficient computer application of sinc-Galerkin method has been presented for nonlinear BVPs. The main advantage of our technique compared to other methods (e.g., Newton’s method) is that the solution is independent of the singularity conditions and

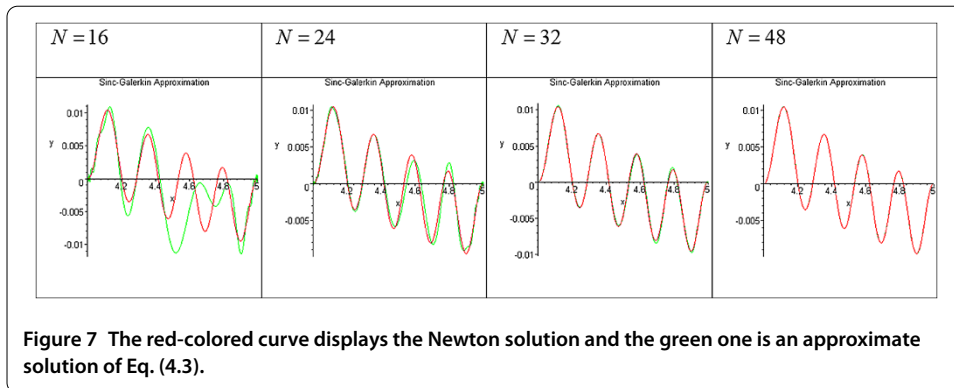


Figure 7 The red-colored curve displays the Newton solution and the green one is an approximate solution of Eq. (4.3).

Table 4 The numerical results for the approximate solutions obtained by sinc-Galerkin in comparison with the Newton solutions of Eq. (4.3) for $N = 48$

x	Newton Solution	Sinc-Galerkin	Relative Error
4.01	0.0000819807184737612502	0.0000855970440948756	3.616325621114482100E-6
4.11	0.0103179812858067658000	0.0103251970537185000	7.215767911801503840E-6
4.21	-0.0011116378549017658700	-0.0011221143338274400	1.047647892566533000E-5
4.31	0.0030747524654397085400	0.0030712556170523700	3.496848387328578060E-6
4.41	0.0010377464582858838400	0.0010372904397433000	4.560185426059223020E-7
4.51	-0.0034415082789204937200	-0.0034511540521152400	9.645773194724025980E-6
4.61	0.0018479414405554888800	0.0018722669507933500	2.432551023786393470E-5
4.71	-0.0072577349999568422200	-0.0072485697536300800	9.165246326795330540E-6
4.81	0.0010544514520660040800	0.0010554094005589100	9.579484929011514780E-7
4.91	-0.0092573478518260567200	-0.0092581760507376400	8.281989115607653900E-7

valid for Dirichlet-type boundary conditions. The order of accuracy used in this paper is $O(N^{1/2}e^{-(\pi d\alpha N)^{1/2}})$. We have used different N node points for all figures presented in this paper. Even though the numerical solution looks complex for even $N > 30$ node points, Maple handles it very well. In the Appendix, a useful Maple program is given to explain the technique and to show how the same solution can be used for different boundary conditions. By using the same program, substituting N and other parameters (like equations, boundaries), different solutions and graphics can be produced. The total time taken on a 3.5 GHz Pentium I7 processor with 8 Core and 8 GB RAM for producing figures and numerical results is less than 20 seconds.

6 Conclusion

In this study, the sinc-Galerkin method has been employed to find the solutions of second-order nonlinear Dirichlet-type boundary value problems on some closed real interval and the method has been compared to the Newton method. Our main purpose is to find the solution of boundary value problems which arise from the singular problems for which the Newton method does not converge at singular points. The powerful side of our method is that it can easily compute solutions even if the equation has singularities. The Newton method can fail when computing some complicated forms of governing equations; on the other hand, our method can easily handle this situation. The examples show that the accuracy improves by increasing the number of sinc grid points N . The method presented here is simple and gives a numerical solution, which is valid for various boundary conditions. We have developed a very efficient algorithm to solve second-order nonlinear Dirichlet-type boundary value problems with sinc-Galerkin method in Maple Computer Algebra

System. Several nonlinear BVPs have been solved by using our technique in less than 20 seconds. All computations and graphical representations have been prepared automatically by our algorithm.

Appendix: A computer application of numeric solutions for nonlinear boundary value problems (NBVPs)

We demonstrate below how to solve and simulate for a nonlinear BVP. For example, the following Maple code computes and simulates Example 4.3.

Set all parameters as default values

> **restart:**

For drawing approximation graphics, we must type the following line

> **with(plots):**

A user has to specify with (linalg) for linear algebra operations in Maple

> **with(linalg):**

A user can define the grid point size N for sinc-Galerkin approximation

> **N:=48:**

The boundary conditions are given as Eq. (4.3).

> **a:=4:**

> **b:=5:**

> **Boundaries:=y(a)=0,y(b)=0;**

$$\text{Boundaries} := y(a) = 0, y(b) = 0.$$

P , Q and R are the variable coefficients of Eq. (1.1). In Maple for Eq. (4.3) they are defined as follows:

> **P(x):=1;**

$$P(x) := 1.$$

> **Q(x):=1;**

$$Q(x) := 1.$$

> **R(x):=1;**

$$R(x) := 1.$$

F is right side of Eq. (4.3)

> **F(x):=cos(Pi*x^2)*x;**

$$F(x) := \cos(\pi x^2)x.$$

We can write a nonlinear part of Eq. (1.1) as follows. User can define any form of nonlinearity in this section.

> **NLPart:=-exp(-sin(y(x)))*y(x)^2/(1+y(x));**

$$NL(y(x)) = NLPart := -\frac{e^{-\sin(y(x))}(y(x))^2}{1+y(x)}.$$

The main form of Eq. (1.1)

> **Equation:=P(x)*diff(y(x),x\$2)+Q(x)*diff(y(x),x\$1)+R(x)*NLPart=F(x);**

$$\frac{d^2}{dx^2}y(x) + \frac{d}{dx}y(x) - \frac{e^{-\sin(y(x))}(y(x))^2}{1+y(x)} = \cos(\pi x^2)x.$$

If the user needs, the main equation can be written in the latex format

> **latex(Equation);**

$$\left\{ \frac{d^2}{dx^2} y \left(x \right) + \frac{d}{dx} y \left(x \right) - \frac{e^{-\sin \left(y \left(x \right) \right)} \left(y \left(x \right) \right)^2}{1+y \left(x \right)} \right\} = \cos \left(\pi x^2 \right) x$$

In order to compare our method with the Newton interpolation (for nonlinear ODE) method, we first solve Eq. (4.3) numerically as follows:

> **NewtonSolution:=dsolve({Equation,Boundaries},y(x),
 type=numeric,method=bvp);**

Prepare the plot of the Newton solution

> **PlotNewtonSolution:=odeplot(NewtonSolution,a...b):**

To define $I_m^{(0)} = [\delta_{jk}^{(0)}]$, $I_m^{(1)} = [\delta_{jk}^{(1)}]$ and $I_m^{(2)} = [\delta_{jk}^{(2)}]$ matrices given in Eqs. (3.13)-(3.15), we use piecewise functions in Maple in the following way:

> **delta[0]:=unapply(piecewise(j=k,1,j<>k,0),j,k):**
 > **delta[1]:=unapply(piecewise(j=k,0,j<>k,((-1)^(k-j))/(k-j)),j,k):**
 > **delta[2]:=unapply(piecewise(j=k,(-Pi^2)/3,j<>k,-2*(-1)^(k-j)/(k-j)^2),j,k):**

The parameters for sinc-approximation given [1]

> **d:=Pi/2:**
 > **h:=2/sqrt(N):**

The evenly spaced nodes given (2.9) and Table 1 are defined as follows:

> **xk:=unapply((a+b*exp(k*h))/(1+exp(k*h)),k);**

$$x_k := k \mapsto \frac{4 + 5e^{kh}}{1 + e^{kh}}.$$

The conformal map in Table 1 for sinc-Galerkin method and its derivatives is computed as follows:

> **phi:=unapply(log((x-a)/(b-x)),x);**

$$\phi(x) := x \mapsto \ln \left(\frac{x-4}{5-x} \right).$$

> **Dphi:=unapply(simplify(diff(phi(x),x)),x):**
 > **D2phi:=unapply(simplify(diff(phi(x),x\$2)),x):**

The weight function and its derivatives are computed for using an inner product to discretization Eq. (4.3)

```

> w:=unapply(1/Dphi(x),x):
> Dw:=unapply(simplify(diff(w(x),x$1)),x):
> D2w:=unapply(simplify(diff(w(x),x$2)),x):
By using sinc-discretization in (3.16), the matrix system with  $(2N + 1) \times (2N + 1)$  dimensions defined in (3.17) is obtained by the following iteration:
> MatrixSystem:=[]:
  for p from -N to N
  do
    MatrixSystem:=[op(sys),
      h*(
        sum(c[j]*(1/h^2)*delta[2](p,j)*(
          Dphi(xk(j))*subs(x=xk(j),P(x)*w(x))+ (1/h^1)*delta[1](p,j)*(
            (D2phi(xk(j))/Dphi(xk(j)))*subs(x=xk(j),P(x)*w(x))+2*subs(x=xk(j),
              diff(P(x)*w(x),x)) + (1/h^0)*delta[0](p,j)*((subs(x=xk(j),
                diff(P(x)*w(x),x$2))/Dphi(xk(j))))),j=-N..N)
        -sum(c[j]*(1/h^1)*delta[1](p,j)*(subs(x=xk(j),Q(x)*w(x)))+
          (1/h^0)*delta[0](p,j)*(subs(x=xk(j),diff(Q(x)*w(x),x)))/Dphi(xk(j))
          ),j=-N..N)
        +subs(y(x)=c[p],NLPart)*subs(x=xk(p),w(x)*R(x))/Dphi(xk(p))
        -subs(x=xk(p),w(x)*F(x))/Dphi(xk(p))=0]:
    od:

```

If we want to obtain solutions of linear BVPs, we can use the following lines. They can reduce time complexity. Here, the linear solution is given as a comment (“#”).

```

> #for Linear system
> #vars:=seq(c[i],i=-N..N):
> #A,b:=LinearAlgebra[GenerateMatrix](evalf(MatrixSystem),[vars]):
> #c:=linsolve(A,b);

```

In this paper, we want to solve nonlinear problems. Then we use **fsolve** function given by Maple to find unknown c_j coefficients (3.17)-(3.18) from nonlinear matrix systems. This function can solve any nonlinear systems by using the Newton method (for nonlinear equation systems).

```

> c:=fsolve(evalf(MatrixSystem)):
Finally, we have unknown  $c_j$  coefficients for the approximate sinc-Galerkin solution (3.18)
> ApproximateSol:=unapply(sum('rhs'(c[j+N+1])*sin(Pi*(phi(x)-j*h)/h)/(Pi*(phi(x)-
  j*h)/h),j=-N..N),x):

```

We define plot of Eq. (4.3) obtained by the sinc-Galerkin solution

```

> Sinc-GalerkinPlot:=plot({ApproximateSol(x)},x=a..b,color=green,thickness=1):
Simulation: Figure 5, Figure 6, and Figure 7 are obtained as
> display({Sinc-GalerkinPlot, PlotNewtonSolution }, title =
  "Sinc-GalerkinApproximation",
  labels=["x","y"]);

```

Enter the number of digits here

```

> Digits := 15:
Tables 2, 3, and 4 are obtained by the following code:
> Exact:=[]:Apprx:=[]:Err:=[]:XPoint:=[]:
> for s from a+0.01 by 0.1 to b-0.01

```

```
do
  XPoints:=[op(XPoints),s];
  NewtonOutputArray:=[op(Exact),rhs(NewtonSolution (s)[2])];
  ApprxmOutputArray:=[op(Apprx),evalf(ApproximateSol(s))];
  ErrOutputArray:=[op(Err),evalf(abs(ApproximateSol(s)-
  rhs(NewtonSolution(s)[2])))];
od:
> latex(XPoints);
[4.01, 4.11, 4.21, 4.31, 4.41, 4.51, 4.61, 4.71, 4.81, 4.91]
> latex(NewtonOutputArray);
[0.0000819807184737612502, 0.0103179812858067658, -0.00111163785490176587, 0.00307475246543970854,
0.00103774645828588384, -0.00344150827892049372, 0.00184794144055548888, -0.00725773499995684222,
0.00105445145206600408, -0.00925734785182605672]
> latex(ApprxmOutputArray);
[0.0000855970440948756, 0.0103251970537185, -0.00112211433382744, 0.00307125561705237,
0.00103729043974330, -0.00345115405211524, 0.00187226695079335, -0.00724856975363008,
0.00105540940055891, -0.00925817605073764]
> latex(ErrOutputArray);
[0.0000036163256211448210, 0.00000721576791180150384, 0.0000104764789256653300,
0.00000349684838732857806, 0.000000456018542605922302, 0.00000964577319472402598,
0.0000243255102378639347, 0.00000916524632679533054, 0.000000957948492901151478,
0.000000828198911560765390]
```

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

AS proposed main idea of the solution schema by using Sinc Method. He developed computer algorithm and worked on theoretical aspect of problem. MK searched the materials about study and compared with other techniques. MAA contributed us with his experience on Nonlinear Approximation methods. MB contributed us with his experience on Nonlinear Approximation methods, suggested us some valuable techniques.

Author details

¹Department of Mathematical Engineering, Faculty of Chemical and Metallurgical Engineering, Yildiz Technical University, Davutpasa, İstanbul, 34210, Turkey. ²Department of Mathematics, Faculty of Art and Sciences, Yildiz Technical University, Davutpasa, İstanbul, 34210, Turkey. ³Department of Mathematics, Bilecik University, Bilecik, 11210, Turkey.

Received: 3 August 2012 Accepted: 2 October 2012 Published: 24 October 2012

References

1. Stenger, F: A sinc-Galerkin method of solution of boundary value problems. *Math. Comput.* **33**, 85-109 (1979)
2. Stenger, F: Approximations via Whittaker's cardinal function. *J. Approx. Theory* **17**, 222-240 (1976)
3. Whittaker, ET: On the functions which are represented by the expansions of the interpolation theory. *Proc. R. Soc. Edinb.* **35**, 181-194 (1915)
4. Whittaker, JM: *Interpolation Function Theory*. Cambridge Tracts in Mathematics and Mathematical Physics, vol. 33. Cambridge University Press, London (1935)
5. Lund, J: Symmetrization of the sinc-Galerkin method for boundary value problems. *Math. Comput.* **47**, 571-588 (1986)
6. Lund, J, Bowers, KL: *Sinc Methods for Quadrature and Differential Equations*. SIAM, Philadelphia (1992)
7. Lewis, DL, Lund, J, Bowers, KL: The space-time sinc-Galerkin method for parabolic problems. *Int. J. Numer. Methods Eng.* **24**, 1629-1644 (1987)
8. McArthur, KM, Bowers, KL, Lund, J: Numerical implementation of the sinc-Galerkin method for second-order hyperbolic equations. *Numer. Methods Partial Differ. Equ.* **3**, 169-185 (1987)
9. Bowers, KL, Lund, J: Numerical solution of singular Poisson problems via the sinc-Galerkin method. *SIAM J. Numer. Anal.* **24**(1), 36-51 (1987)
10. Lund, J, Bowers, KL, McArthur, KM: Symmetrization of the sinc-Galerkin method with block techniques for elliptic equations. *IMA J. Numer. Anal.* **9**, 29-46 (1989)
11. Lybeck, NJ: *Sinc domain decomposition methods for elliptic problems*. PhD thesis, Montana State University, Bozeman (1994)

12. Lybeck, NJ, Bowers, KL: Domain decomposition in conjunction with sinc methods for Poisson's equation. *Numer. Methods Partial Differ. Equ.* **12**, 461-487 (1996)
13. Morlet, AC, Lybeck, NJ, Bowers, KL: The Schwarz alternating sinc domain decomposition method. *Appl. Numer. Math.* **25**, 461-483 (1997)
14. Morlet, AC, Lybeck, NJ, Bowers, KL: Convergence of the sinc overlapping domain decomposition method. *Appl. Math. Comput.* **98**, 209-227 (1999)
15. Alonso, N, Bowers, KL: An alternating-direction sinc-Galerkin method for elliptic problems. *J. Complex.* **25**, 237-252 (2009)
16. Ng, M: Fast iterative methods for symmetric sinc-Galerkin systems. *IMA J. Numer. Anal.* **19**, 357-373 (1999)
17. Ng, M, Bai, Z: A hybrid preconditioner of banded matrix approximation and alternating-direction implicit iteration for symmetric sinc-Galerkin linear systems. *Linear Algebra Appl.* **366**, 317-335 (2003)
18. Stenger, F: *Numerical Methods Based on Sinc and Analytic Functions*. Springer, New York (1993)
19. Koonprasert, S: *The sinc-Galerkin method for problems in oceanography*. PhD thesis, Montana State University, Bozeman (2003)
20. McArthur, KM, Bowers, KL, Lund, J: The sinc method in multiple space dimensions: model problems. *Numer. Math.* **56**, 789-816 (1990)
21. Stenger, F: Numerical methods based on Whittaker cardinal, or sinc functions. *SIAM Rev.* **23**, 165-224 (1981)
22. Stenger, F: Summary of sinc numerical methods. *J. Comput. Appl. Math.* **121**, 379-420 (2000)
23. Stenger, F, O'Reilly, MJ: Computing solutions to medical problems via sinc convolution. *IEEE Trans. Autom. Control* **43**, 843 (1998)
24. Narasimhan, S, Majdalani, J, Stenger, F: A first step in applying the sinc collocation method to the nonlinear Navier Stokes equations. *Numer. Heat Transf., Part B, Fundam.* **41**, 447-462 (2002)
25. Mueller, JL, Shores, TS: A new sinc-Galerkin method for convection-diffusion equations with mixed boundary conditions. *Comput. Math. Appl.* **47**, 803-822 (2004)
26. El-Gamel, M, Behiry, SH, Hashish, H: Numerical method for the solution of special nonlinear fourth-order boundary value problems. *Appl. Math. Comput.* **145**, 717-734 (2003)
27. Lybeck, NJ, Bowers, KL: Sinc methods for domain decomposition. *Appl. Math. Comput.* **75**, 4-13 (1996)
28. Zarebnia, M, Sajjadian, M: The sinc-Galerkin method for solving Troesch's problem. *Math. Comput. Model.* (2011). doi:10.1016/j.mcm.2011.11.071

doi:10.1186/1687-2770-2012-117

Cite this article as: Secer et al.: An efficient computer application of the sinc-Galerkin approximation for nonlinear boundary value problems. *Boundary Value Problems* 2012 **2012**:117.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
