

Research Article

Numerical Solution and Simulation of Second-Order Parabolic PDEs with Sinc-Galerkin Method Using Maple

Aydin Secer

Department of Mathematical Engineering, Yildiz Technical University, Davutpasa, 34210 Istanbul, Turkey

Correspondence should be addressed to Aydin Secer; asecer@yildiz.edu.tr

Received 4 March 2013; Accepted 12 June 2013

Academic Editor: Adem Kılıçman

Copyright © 2013 Aydin Secer. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An efficient solution algorithm for sinc-Galerkin method has been presented for obtaining numerical solution of PDEs with Dirichlet-type boundary conditions by using Maple Computer Algebra System. The method is based on Whittaker cardinal function and uses approximating basis functions and their appropriate derivatives. In this work, PDEs have been converted to algebraic equation systems with new accurate explicit approximations of inner products without the need to calculate any numeric integrals. The solution of this system of algebraic equations has been reduced to the solution of a matrix equation system via Maple. The accuracy of the solutions has been compared with the exact solutions of the test problem. Computational results indicate that the technique presented in this study is valid for linear partial differential equations with various types of boundary conditions.

1. Introduction

Sinc methods for differential equations were originally introduced by Stenger in [1–3]. The sinc functions were first analyzed in [4, 5] and a detailed research of the method for two-point boundary-value problems can be found in [6, 7]. In [8], parabolic and hyperbolic problems are presented in detail. To solve a problem arising from chemical reactor theory, the properties of the sinc-Galerkin method are used to reduce the computation of nonlinear two-point boundary-value problems to some algebraic equations in [9]. A computer algorithm for sinc method to solve numerically the linear and nonlinear ODEs and their simulations has been presented in [7, 10], respectively. The full sinc-Galerkin method is developed for a family of complex-valued partial differential equations with time-dependent boundary conditions [9]. A study of the performance of the Galerkin method using sinc basis functions for solving Bratu's problem is presented in [11]. In [12] a numerical algorithm has been presented for recovering the unknown function and obtaining a solution to the inverse ill-posed problem. They have presented a Galerkin method with the sinc basis functions in both space and time domains for solving the direct problem. A sinc-collocation method has been developed for solving linear systems of

integrodifferential equations of Fredholm and Volterra type with homogeneous boundary conditions in [13].

2. Sinc-Approximation Formula for PDEs

We use the sinc-Galerkin method as mentioned in [1] to derive an approximate solution of the following:

$$\begin{aligned} u_t - u_{xx} &= F(x, t), \\ u(0, t) &= u(1, t) = 0, \quad 0 < x < 1, \\ u(x, 0) &= f(x), \quad t > 0. \end{aligned} \quad (1)$$

For the equation given above, the sinc-Galerkin scheme can be developed in both space and time directions as follows.

In general, approximations can be constructed for infinite, semi-infinite, and infinite intervals and both spatial and time spaces will be introduced. Define the function

$$\phi(z) = \ln\left(\frac{z}{1-z}\right) \quad (2)$$

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

$$D_E = \left\{ z = x + iy : \left| \arg\left(\frac{z}{1-z}\right) \right| < d \leq \frac{\pi}{2} \right\}. \quad (3)$$

A more general form of sinc basis according to intervals can be given as follows:

$$S(m, h_x) \circ \phi(x) = \text{Sinc}\left(\frac{\phi(x) - mh_x}{h_x}\right), \quad m = -N_x, \dots, N_x,$$

$$S(k, h_t) \circ \gamma(t) = \text{Sinc}\left(\frac{\gamma(t) - kh_t}{h_t}\right), \quad k = -N_t, \dots, N_t, \quad (4)$$

where

$$\text{Sinc}(z) = \begin{cases} \frac{\sin(\pi z)}{\pi z}, & z \neq 0 \\ 1, & z = 0, \end{cases}$$

$$\text{Sinc}(k, h)(z)$$

$$= \text{Sinc}\left(\frac{z - kh}{h}\right)$$

$$= \begin{cases} \frac{\sin(\pi((z - kh)/h))}{\pi((z - kh)/h)}, & z \neq kh \\ 1, & z = kh, \end{cases} \quad k = 0, \pm 1, \pm 2, \pm 3, \dots \quad (5)$$

and the conformal maps for both directions

$$\phi(x) = \ln\left(\frac{x}{l-x}\right), \quad x \in (0, l), \quad (6)$$

$$\gamma(t) = \ln(t), \quad t \in (0, \infty)$$

are used to define the basis functions on the intervals $(0, l)$ and $(0, \infty)$, respectively. $h_x, h_t > 0$ represents the mesh sizes in the space direction and the time direction, respectively. The sinc nodes x_i and t_j are chosen so that $x_i = \phi^{-1}(ih_x)$, $t_j = \gamma^{-1}(jh_t)$.

Here the function $x = \phi^{-1}(x) = e^x/(1 + e^x)$ is an inverse mapping of $\phi = \phi(x)$. We may define the range of ϕ^{-1} on the real line as

$$\Gamma_1 = \{\phi^{-1}(u) \in D_E : -\infty < u < \infty\}. \quad (7)$$

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

$$x_k = \phi^{-1}(kh) = \frac{e^{kh}}{1 + e^{kh}}, \quad (8)$$

where $0 < x_k < 1$, for all k .

The sinc basis functions in (4) do not have a derivative when x tends to 0 or 1. We modify the sinc basis functions as

$$\frac{S(m, h_x) \circ \phi(x)}{\phi'(x)} = \frac{\text{Sinc}((\phi(x) - mh_x)/h_x)}{\phi'(x)}, \quad (9)$$

where

$$\frac{1}{\phi'(x)} = x(1-x). \quad (10)$$

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

For the temporal space, we construct an approximation by defining the function

$$w = \gamma(r) = \ln(r) \quad (11)$$

which is a conformal mapping from D_W , the wedge-shaped domain in the r -plane, onto the infinite strip, D_S , where

$$D_W = \left\{r = t + is : |\arg(r)| < d < \frac{\pi}{2}\right\}, \quad (12)$$

derived from composite translated functions

$$S(k, h_t) \circ \gamma(t) = \text{Sinc}\left(\frac{\gamma(t) - kh_t}{h_t}\right), \quad k = -N_t, \dots, N_t, \quad (13)$$

for $r \in D_W$.

Here $w = \gamma(r)$ and $\gamma^{-1}(w) = r = e^w$. We may define γ^{-1} on the real line as

$$\Gamma_2 = \{\gamma^{-1}(u) \in D_w : -\infty < u < \infty\}. \quad (14)$$

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

$$t_k = \gamma^{-1}(kh) = e^{kh}, \quad (15)$$

where $0 < t_k < \infty$, for all k .

A list of conformal mappings may be found in Table 1 [14].

Definition 1. Let $B(D_E)$ be the class of functions F that are analytic in D_E and satisfy

$$\int_{\psi(L+u)} |F(z)| dz \rightarrow 0, \quad \text{as } u = \mp\infty, \quad (16)$$

where

$$L = \{iy : |y| < d \leq \frac{\pi}{2}\} \quad (17)$$

and on the boundary of D_E satisfy

$$T(F) = \int_{\partial D_E} |F(z)| dz < \infty. \quad (18)$$

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

Theorem 2. Let Γ be $(0, 1)$, $F \in B(D_E)$, and then for $h > 0$ sufficiently small

$$\int_{\Gamma} F(z) dz - h \sum_{j=-\infty}^{\infty} \frac{F(z_j)}{\phi'(z_j)} = \frac{i}{2} \int_{\partial D} \frac{F(z) k(\phi, h)(z)}{\sin(\pi \phi(z)/h)} dz \equiv I_F, \quad (19)$$

where

$$|k(\phi, h)|_{z \in \partial D} = |e^{[i\pi\phi(z)/h] \operatorname{sgn}(\operatorname{Im} \phi(z))}]|_{z \in \partial D} = e^{-\pi d/h}. \quad (20)$$

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

Theorem 3. If there exist positive constants α, β , and C such that

$$\left| \frac{F(x)}{\phi'(x)} \right| \leq C \begin{cases} e^{-\alpha|\phi(x)|}, & x \in \psi(-\infty, \infty) \\ e^{-\beta|\phi(x)|}, & x \in \psi(0, \infty), \end{cases} \quad (21)$$

then the error bound for the quadrature rule (19) is

$$\left| \int_{\Gamma} F(x) dx - h \sum_{j=-N}^N \frac{F(x_j)}{\phi'(x_j)} \right| \leq C \left(\frac{e^{-\alpha Nh}}{\alpha} + \frac{e^{-\beta Nh}}{\beta} \right) + |I_F|. \quad (22)$$

The infinite sum in (19) is truncated with the use of (20) to arrive at (22).

Making the selections

$$h = \sqrt{\frac{\pi d}{\alpha N}}, \quad (23)$$

$$N \equiv \left\| \frac{\alpha N}{\beta} + 1 \right\|,$$

where $\|\cdot\|$ is integer part of statement,

$$\int_{\Gamma} F(x) dx = h \sum_{j=-N}^N \frac{F(x_j)}{\phi'(x_j)} + O(e^{-(\pi \alpha d N)^{1/2}}). \quad (24)$$

Theorems 2 and 3 can be used to approximate the integrals that arise in the formulation of the discrete systems corresponding to two-point BVPs.

3. Discrete Solutions Scheme for Two-Point BVPs

In ordinary differential equations

$$Lu = f \quad (25)$$

on Γ_1 , sinc solution is assumed as an approximate solution u_m in the form of series with $m = 2N + 1$ terms

$$u_m(z) = \sum_{j=-N}^N c_j S(j, h) \circ \phi(z). \quad (26)$$

The coefficients $\{c_j\}_{j=-N}^N$ are determined by orthogonalizing the residual $Lu - f$ with respect to the sinc basis functions $\{S_k\}_{k=-N}^N$ where $S_k(z) = S(k, h) \circ \phi(z)$. An inner product for two continuous functions such as f_1 and f_2 can be given by the following formula

$$\langle f_1, f_2 \rangle = \int_{\Gamma} f_1 f_2 w dz, \quad (27)$$

where w is the weight function and is chosen depending on boundary conditions. If we implement the above inner product rule in orthogonalization, this yields the discrete sinc-Galerkin system:

$$\int_{\Gamma} (Lu_m - f)(z) S(k, h) \circ \phi(z) \cdot w(z) dz = 0, \quad (28)$$

$$-N \leq k \leq N.$$

Now, we are going to derive discrete sinc-Galerkin system for PDEs. Assume u_{m_z, m_t} is the approximate solution of (1). Then, the discrete system takes the following form:

$$u_{m_z, m_t}(z, t) = \sum_{j=-N}^N \sum_{k=-N}^N c_{jk} S(j, h) \circ \phi(z) \cdot S(k, h) \circ \gamma(t). \quad (29)$$

The coefficients $\{c_{jk}\}_{j,k=-N}^N$ are determined by orthogonalizing the residual $Lu_{m_z, m_t} - f$ with respect to the sinc basis functions $\{S_k S_h\}_{k,h=-N}^N$ where $S_j S_h(z, t) = S(j, h) \circ \phi(z) S(k, h) \circ \gamma(t)$ for $-N \leq j, k \leq N$. In this case the inner product takes the following form:

$$\langle f_1, f_2 \rangle = \int_{\Gamma_t} \int_{\Gamma_z} f_1(z, t) f_2(z, t) w(z, t) dz dt. \quad (30)$$

The choice of the weight function $w(z, t)$ in the double integrand depends on the boundary conditions, the domain, and the partial differential equation. Therefore, the discrete Galerkin system is

$$\int_{\Gamma_t} \int_{\Gamma_z} (Lu_{m_z, m_t} - f)(z, t) \cdot S(j, h) \circ \phi(z) \cdot S(k, h) \circ \gamma(t) \cdot w(z, t) dz dt = 0. \quad (31)$$

4. Matrix Representation of the Derivatives of Sinc Basis Functions at Nodal Points

The sinc-Galerkin method actually requires the evaluated derivatives of sinc basis functions at the sinc nodes, $z = z_j$. The r th derivative of $S_k(z) = S(k, h) \circ \phi(z)$ with respect to ϕ , evaluated at the nodal point z_j , is denoted by

$$\frac{1}{h^r} \delta_{pj}^{(r)} = \frac{d^r}{d\phi^r} (S(p, h) \circ \phi(z)) \Big|_{z=z_j}. \quad (32)$$

```

restart:
with(linalg):
with(LinearAlgebra):
N:=16:
d:=Pi/2:
h:=0.75/sqrt(N);
s:=.5/sqrt(N);
SIZE:=2*N+1;
delta0:=(i,j)->piecewise(j=i,1,j<>i,0):
delta1:=(i,j)->piecewise(i=j,0,i<>j,((-1)^(i-j))/(i-j)):
delta2:=(i,j)->piecewise(i=j,(-Pi^2)/3,i<>j,-2*(-1)^(i-j)/(i-j)^2):
I_0:=Matrix(SIZE,delta0):
I_1:=Matrix(SIZE,delta1):
I_2:=Matrix(SIZE,delta2):
xk:=1/2+1/2*tanh(k*h/2);
xk_func:=unapply(xk,k);
phi:=unapply(log((x)/(1-x)),x);
Dphi:=unapply(simplify(diff(phi(x),x)),x);
g:=unapply(simplify(1/diff(phi(x),x)),x);
Dg:=unapply(diff(g(x),x),x);
gk:=unapply(subs(x=xk,g(x)),k);
Dgk:=unapply(subs(x=xk,Dg(x)),k);
g_Div_Dphi:=unapply(g(x)/Dphi(x),x);
g_Div_Dphi_k:=unapply(subs(x=xk,g_Div_Dphi(x)),k);
#Temporal Spaces;
t1:=exp(1*s);
t1_func:=unapply(t1,l);
gamm:=unapply(log(t),t);
Dgam:=unapply(diff(gamm(t),t),t);
g_gamm:=unapply(1/diff(gamm(t),t),t);
g_gamm_l:=unapply(subs(t=t1,g_gamm(t)),l);
gamm_Div_Dgam:=unapply(g_gamm(t)/Dgam(t),t);
gamm_Div_Dgam:= t-> t^2;
gamm_Div_Dgam_l:=unapply(subs(t=t1,gamm_Div_Dgam(t)),l);
GenerateDiagonalAm := proc( x )
    local i:=1:
    local A:=Matrix(SIZE):
    for i from 1 by 1 to SIZE
        do
            A[i,i]:=evalf(x(-N+i-1)):
        end do:
    return A;
end proc;
B:=-2*h*MatrixMatrixMultiply(I_0,GenerateDiagonalAm(gk))+
    MatrixMatrixMultiply(I_1,GenerateDiagonalAm(Dgk))+1/h*I_2:
DD:=h*GenerateDiagonalAm(g_Div_Dphi_k):
C:=MatrixMatrixMultiply(GenerateDiagonalAm(g_gamm_l),s*(I_0-I_1)):
E:=s*GenerateDiagonalAm(gamm_Div_Dgam_l):
Fkl:=unapply((Pi^2-4)*sin(Pi*xk_func(k-N-1))*exp(-t1_func(1-N-1)),k,l);
F:=evalf(Matrix(SIZE,Fkl)):
V:=Matrix(SIZE,v):
EQN_SYS:=evalf(
    MatrixMatrixMultiply(
        MatrixMatrixMultiply(
            Matrix(inverse(DD)),B
        ),V

```

ALGORITHM 1: Continued.

```

        )
    )
    +evalf(
        MatrixMatrixMultiply(
            MatrixMatrixMultiply(
                V,C),Matrix(inverse(E)
            )
        )
    ):
SYS:=[]:
    for i from 1 by 1 to SIZE
        do
            for j from 1 by 1 to SIZE
                do
                    SYS:=[op(SYS),EQN_SYS(i,j)=F(i,j)];
                end do:
            end do:
        vars:=seq(seq(v(i,j),i=1..2*N+1),j=1..2*N+1):
        A,b:LinearAlgebra[GenerateMatrix](evalf(SYS),[vars]):
        c:=linsolve(A,b):
        CoeffMatrix=Matrix(SIZE):
        cnt:=1;
        for i to SIZE do
            for j to SIZE do
                CoeffMatrix[j,i]:=c[cnt]:
                cnt:=cnt+1
            end do;
        end do;
        CoeffMatrix:=Matrix(CoeffMatrix,SIZE):
        ApproximateSol:=unapply(
            evalf(
                sum(
                    sum("CoeffMatrix"[m+N+1,n+N+1]
                        *sin(Pi*(phi(x)-m*h)/h)/(Pi*(phi(x)-m*h)/h)
                        *sin(Pi*(gamm(t)-n*s)/s)/(Pi*(gamm(t)-n*s)/s)
                        ,m=-N..N),
                    n=-N..N)
                )
            ,x,t):
        plot3d(ApproximateSol(x,t),x=0..1,t=0..1):
        Exact:=unapply(exp(-Pi^2*t)*sin(Pi*x),x,t);
        plot3d(Exact(x,t),x=0..1,t=0..1);
        plot3d({Exact(x,t),ApproximateSol(x,t)},x=0..1,t=0..1);
        XX:=.6;
        #Numerical Comparision EXACT
        for j from 0.1 to 10 by 1
            do
                evalf(Exact(XX,j)):
            od;
        #Numerical Comparision APPROX
        for j from 0.1 to 10 by 1
            do
                evalf(ApproximateSol(XX,j)):
            od;
        #Numerical Comparision ERROR
        for j from 0.1 to 10 by 1
            do
                abs(evalf(evalf(ApproximateSol(XX,j)-Exact(XX,j)))):
            od;

```

The expressions in (14) for each p and j can be stored in a matrix $I^{(r)} = [\delta_{pj}^{(r)}]$. For $r = 0, 1, 2, \dots$

$$\begin{aligned}
 I^{(0)} &= \delta_{jk}^{(0)} = [S(j, h) \circ \phi(x)]|_{x=x_k} = \begin{cases} 1, & k = j \\ 0, & k \neq j, \end{cases} \\
 I^{(1)} &= \delta_{jk}^{(1)} = h \frac{d}{d\phi} [S(j, h) \circ \phi(x)]|_{x=x_k} = \begin{cases} 0, & k = j \\ \frac{(-1)^{k-j}}{(k-j)}, & k \neq j, \end{cases} \\
 I^{(2)} &= \delta_{jk}^{(2)} = h \frac{d^2}{d\phi^2} [S(j, h) \circ \phi(x)]|_{x=x_k} \\
 &= \begin{cases} \frac{-\pi^2}{3}, & k = j \\ \frac{-2(-1)^{k-j}}{(k-j)^2}, & k \neq j, \end{cases}
 \end{aligned} \quad (33)$$

where

$$\begin{aligned}
 I_m^{(0)} &= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} = [\delta_{jk}^{(0)}], \\
 I_m^{(1)} &= \begin{bmatrix} 0 & -1 & \frac{1}{2} & \cdots & \frac{1}{2N} \\ 1 & 0 & -1 & \cdots & -\frac{1}{2N-1} \\ -\frac{1}{2} & 1 & 0 & \cdots & \frac{1}{2N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{2N} & \frac{1}{2N-1} & \frac{1}{2N-2} & \cdots & 0 \end{bmatrix} = [\delta_{jk}^{(1)}], \\
 I_m^{(2)} &= \begin{bmatrix} -\frac{\pi^2}{3} & \frac{2}{1^2} & -\frac{2}{2^2} & \cdots & -\frac{2}{(2N)^2} \\ \frac{2}{1^2} & -\frac{\pi^2}{3} & \frac{2}{1^2} & \cdots & \frac{2}{(2N-1)^2} \\ -\frac{2}{2^2} & \frac{2}{1^2} & -\frac{\pi^2}{3} & \cdots & -\frac{2}{(2N-2)^2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{2}{(2N)^2} & \frac{2}{(2N-1)^2} & -\frac{2}{(2N-2)^2} & \cdots & -\frac{\pi^2}{3} \end{bmatrix} \\
 &= [\delta_{jk}^{(2)}].
 \end{aligned} \quad (34)$$

The chain rule has been used for the z -derivative of product sinc functions. For example, when $S_k(z) = S(k, h) \circ \phi(z)$,

$$\begin{aligned}
 \frac{d(S_j(z) w(z))}{dz} &= \left(\frac{dS_j(z)}{d\phi(z)} \cdot \frac{d\phi(z)}{dz} \right) w(z) \\
 &\quad + S_j(z) \frac{dw(z)}{dz} \\
 &= \frac{dS_j(z)}{d\phi} \phi'(z) w(z) + S_j(z) w'(z), \\
 \frac{d^2(S_j(z) w(z))}{dz^2} &= \frac{d}{dz} \left(\frac{dS_j(z)}{d\phi} \phi'(z) w(z) + S_j(z) w'(z) \right) \\
 &= \frac{d^2 S_j(z)}{d\phi^2} (\phi'(z))^2 w(z) \\
 &\quad + \frac{dS_j(z)}{d\phi} \phi''(z) w(z) \\
 &\quad + 2 \cdot \frac{dS_j(z)}{d\phi} \phi'(z) w'(z) + S_j(z) w''(z).
 \end{aligned} \quad (35)$$

Now, we are going to develop discrete form for (1). We choose for special case the parameters as follows for the spatial dimension:

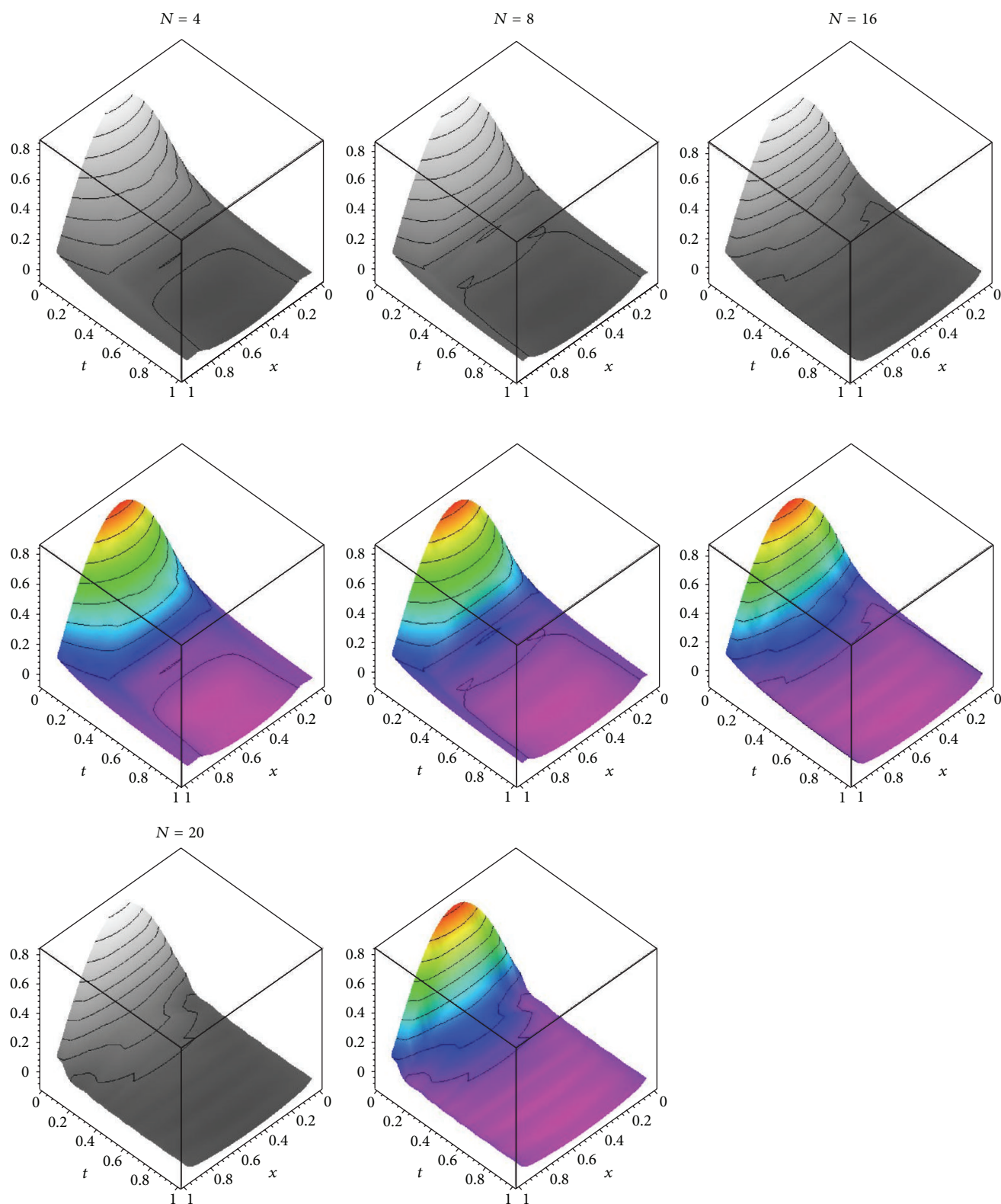
$$\begin{aligned}
 \phi(z) &= \ln \left(\frac{z}{1-z} \right), \\
 w_X(z) &= \frac{1}{\phi'(z)}, \\
 \frac{1}{\phi'(z)} &= z(1-z),
 \end{aligned} \quad (36)$$

and for the temporal space as

$$\begin{aligned}
 \gamma(t) &= \ln(t), \\
 w_T(t) &= \frac{1}{\gamma'(t)}, \\
 \frac{1}{\gamma'(t)} &= t.
 \end{aligned} \quad (37)$$

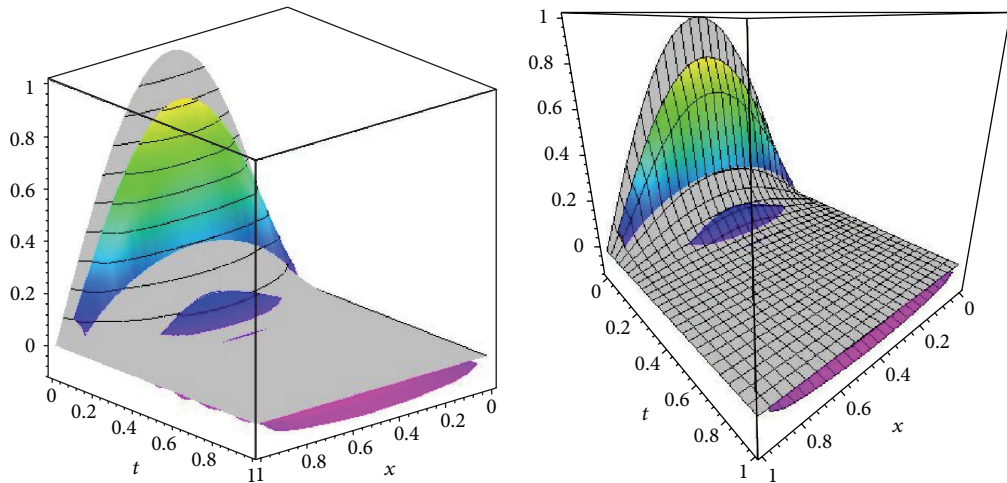
The discrete form of (1) can be given the following form:

$$\begin{aligned}
 &\langle Lu - F, S_k \cdot S_l \rangle \\
 &= \int_{\Gamma_t} \int_{\Gamma_z} (Lu - F) S(k, h) \circ \phi(z) \\
 &\quad \cdot w_X(x) S(l, s) \circ \gamma(t) \cdot w_T(t) dz dt \\
 &= \int_{\Gamma_t} \int_{\Gamma_z} (u_t - u_{xx} - F) S(k, h) \circ \phi(z) \\
 &\quad \cdot w_X(x) S(l, s) \circ \gamma(t) \cdot w_T(t) dz dt.
 \end{aligned} \quad (38)$$

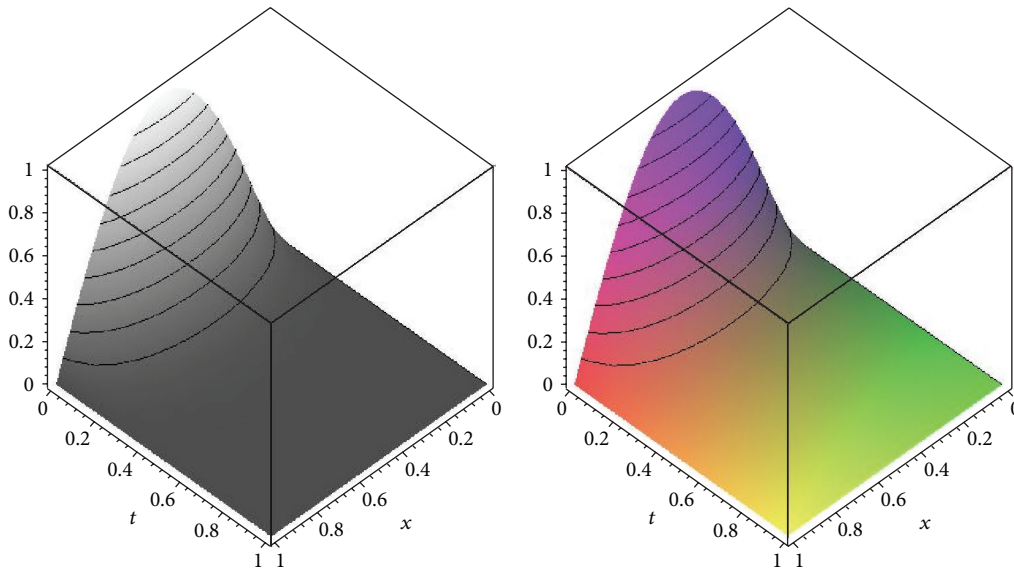


(a) The sinc-Galerkin solutions according to the grid points size N

FIGURE 1: Continued.



(b) Intersection of surfaces for $N = 20$. The left-side figure is normal perspective and the second one has medium perspective view



(c) The exact solution (or particular solution)

FIGURE 1: Simulation of approximate solution.

We solve this by taking our approximating basis functions to be

$$\begin{aligned}
 S_k(x) &= w_X S(k, h) \circ \phi(x), & w_X &= \frac{1}{\phi'(x)} = x(1-x), \\
 \phi(x) &= \ln\left(\frac{x}{1-x}\right), \\
 S_l(t) &= w_T S(l, s) \circ \gamma(t), & w_T &= \frac{1}{\gamma'(t)} = t, \\
 \gamma(t) &= \ln(t).
 \end{aligned}
 \tag{39}$$

If we apply sinc-quadrature rules with the help of (32)–(37) on the definite integral given (38) by using (39), we can get the following matrix system.

Let $A_m(u)$ denote a diagonal matrix, whose diagonal elements are $u(x_{-N}), u(x_{-N+1}), \dots, u(x_N)$ and nondiagonal elements are zero. Then (38) reproduces the following matrixes accordingly.

Firstly we set the coefficient matrix as follows:

$$C = \begin{pmatrix} c_{-N,-N} & c_{-N,-N+1} & c_{-N,-N+2} & \cdots & c_{-N,N} \\ c_{-N+1,-N} & c_{-N+1,-N+1} & c_{-N+1,-N+2} & \cdots & c_{-N+1,N} \\ c_{-N+2,-N} & c_{-N+2,-N+1} & c_{-N+2,-N+2} & \cdots & c_{-N+2,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{N,-N} & c_{N,-N+1} & c_{N,-N+2} & \cdots & c_{N,N} \end{pmatrix},$$

$$B = -2hI_m^{(0)}(A_m(w_X)) + I_m^{(1)}(A_m(w_X')) + \frac{I_m^{(2)}}{h},$$

$$G = A_m(w_T) [sI_m^{(0)} - I_m^{(1)}],$$

TABLE 2: Numerical results.

	t	Exact solution	Sinc-Galerkin solution	Error
$N = 16, x = 0.6$	0.1	0.35446621870000	0.59102026517764600	0.23655404647764600
	0.11	0.00001833412226	-0.08327827240548060	0.08329660652774060
	0.21	$0.9482992112 \times 10^{-9}$	-0.03068110846399060	0.03068110941228980
	0.31	$0.4904905672 \times 10^{-13}$	-0.01083286679129120	0.01083286679134030
	0.41	$0.2536973471 \times 10^{-17}$	-0.000420168677914081	0.000420168677914083
	0.51	$0.1312203514 \times 10^{-21}$	-0.00108069489771883	0.00108069489771883
	0.61	$0.6787134677 \times 10^{-26}$	-0.00492511086695295	0.00492511086695295
	0.71	$0.3510522275 \times 10^{-30}$	0.00331706758955819	0.00331706758955819
	0.81	$0.1815753976 \times 10^{34}$	-0.00303031084399243	0.00303031084399243
	0.91	$0.9391658013 \times 10^{-39}$	0.00342519240062265	0.00342519240062265

$$\begin{aligned} D &= hA_m \left(\frac{w_X}{\phi'} \right), \\ E &= sA_m \left(\frac{w_T}{\gamma'} \right). \end{aligned} \quad (40)$$

Finally, for the right side function F given (1) can be written in the following matrix form:

$$F = \begin{pmatrix} F_{-N,-N} & F_{-N,-N+1} & F_{-N,-N+2} & \cdots & F_{-N,N} \\ F_{-N+1,-N} & F_{-N+1,-N+1} & F_{-N+1,-N+2} & \cdots & F_{-N+1,N} \\ F_{-N+2,-N} & F_{-N+2,-N+1} & F_{-N+2,-N+2} & \cdots & F_{-N+2,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ F_{N,-N} & F_{N,-N+1} & F_{N,-N+2} & \cdots & F_{N,N} \end{pmatrix}. \quad (41)$$

Using (32)–(37) we arrive at a matrix system given in [1] as follows:

$$D^{-1}BC + CGE^{-1} = F. \quad (42)$$

Finally, by using Maple Computer Algebra Software, the matrix system (42) can be solved by using LU or QR decomposition method and unknown coefficients can be found. After calculation of C we get approximate solution as follows:

$$u_{x,t} = \sum_{j=-N}^N \sum_{k=-N}^N c_{jk} S(j, h) \circ \phi(x) \cdot S(k, h) \circ \gamma(t). \quad (43)$$

5. Numerical Simulation

The example in this section will illustrate the sinc method.

Example 4. This problem has been addressed in [1]. The following equation is given in Dirichlet-type boundary condition:

$$\begin{aligned} \frac{\partial u(x, t)}{\partial t} - \frac{\partial^2 u(x, t)}{\partial x^2} &= 0, \quad 0 < x < 1, \quad 0 < t, \\ u(0, t) &= u(1, t) = 0, \\ u(x, 0) &= \sin(\pi x). \end{aligned} \quad (44)$$

The particular solution of (44) can be calculated via separation variables rules and can be given as follows:

$$u(x, t) = e^{-\pi^2 t} \sin(\pi x). \quad (45)$$

For (44) we choose sinc components here in the following:

$$\begin{aligned} h &= s = \frac{0.75}{\sqrt{N}}, & x_k &= \frac{e^{kh}}{1 + e^{kh}}, & t_l &= e^{sl}, \\ \phi(x) &= \ln\left(\frac{x}{1-x}\right), & \gamma(t) &= \ln(t), & w_X &= \frac{1}{\phi'(x)}, \\ w_T &= \frac{1}{\gamma'(t)}. \end{aligned} \quad (46)$$

According to the above parameters, the approximate solution simulation of (44) has been given in Figure 1 and numerical results also can be found in Table 2.

6. Conclusions

We have developed a Maple algorithm to solve and simulate second-order parabolic PDEs with Dirichlet-type boundary conditions based on sinc-Galerkin approximation on some closed real intervals and the method has been compared with the exact solutions. When compared with other computational approaches, this method turns out to be more efficient in the sense that selection parameters and changing boundary conditions and also giving different problems to the algorithms. The accuracy of the solutions improves by increasing the number of sinc grid points N . The method presented here is simple and uses sinc-Galerkin method that gives a numerical solution, which is valid for various boundary conditions. Several PDEs 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

See Algorithm 1.

Conflict of Interests

The author declares that he has no conflict of interests.

References

- [1] F. Stenger, "A Sinc-Galerkin method of solution of boundary value problems," *Mathematics of Computation*, vol. 33, no. 145, pp. 85–109, 1979.
- [2] F. Stenger, "Summary of Sinc numerical methods," *Journal of Computational and Applied Mathematics*, vol. 121, no. 1-2, pp. 379–420, 2000.
- [3] F. Stenger, "Approximations via Whittaker's cardinal function," *Journal of Approximation Theory*, vol. 17, no. 3, pp. 222–240, 1976.
- [4] E. T. Whittaker, "On the functions which are represented by the expansions of the interpolation theory," *Proceedings of the Royal Society of Edinburgh*, vol. 35, pp. 181–194, 1915.
- [5] J. M. Whittaker, *Interpolation Function Theory*, *Cambridge Tracts in Mathematics and Mathematical Physics*, no. 33, Cambridge University Press, London, UK, 1935, by E. F. Beckenbach, McGraw-Hill, New York, NY, USA, 1961.
- [6] J. Lund, "Symmetrization of the sinc-Galerkin method for boundary value problems," *Mathematics of Computation*, vol. 47, no. 176, pp. 571–588, 1986.
- [7] A. Secer and M. Kurulay, "The Sinc-Galerkin method and its applications on singular Dirichlet-type boundary value problems," *Boundary Value Problems*, vol. 2012, article 126, 2012.
- [8] K. M. McArthur, K. L. Bowers, and J. Lund, "Numerical implementation of the Sinc-Galerkin method for second-order hyperbolic equations," *Numerical Methods for Partial Differential Equations*, vol. 3, no. 3, pp. 169–185, 1987.
- [9] A. Saadatmandi, M. Razzaghi, and M. Dehghan, "Sinc-Galerkin solution for nonlinear two-point boundary value problems with applications to chemical reactor theory," *Mathematical and Computer Modelling*, vol. 42, no. 11-12, pp. 1237–1244, 2005.
- [10] A. Secer, M. Kurulay, M. Bayram, and M. A. Akinlar, "An efficient computer application of the Sinc-Galerkin approximation for nonlinear boundary value problems," *Boundary Value Problems*, vol. 2012, article 117, 2012.
- [11] J. Rashidinia, K. Maleknejad, and N. Taheri, "Sinc-Galerkin method for numerical solution of the Bratu's problems," *Numerical Algorithms*, vol. 62, no. 1, pp. 1–11, 2013.
- [12] A. Shidfar and A. Babaei, "The sinc-Galerkin method for solving an inverse parabolic problem with unknown source term," *Numerical Methods for Partial Differential Equations*, vol. 29, no. 1, pp. 64–78, 2013.
- [13] M. Zarebnia and M. G. A. Abadi, "A numerical sinc method for systems of integro-differential equations," *Physica Scripta*, vol. 82, no. 5, Article ID 055011, 2010.
- [14] J. Lund and K. L. Bowers, *Sinc Methods for Quadrature and Differential Equations*, SIAM, Philadelphia, Pa, USA, 1992.

