Vol 18 No 8, August 2009
1674-1056/2009/18(08)/3099-05

Chinese Physics B

© 2009 Chin. Phys. Soc.
and IOP Publishing Ltd

A numerical method for one-dimensional nonlinear sine-Gordon equation using multiquadric quasi-interpolation*

Ma Li-Min(马利敏)† and Wu Zong-Min(吴宗敏)

Shanghai Key Laboratory for Contemporary Applied Mathematics, School of Mathematical Sciences, Fudan University, Shanghai 200433, China

(Received 11 November 2008; revised manuscript received 2 December 2008)

In this paper, we use a univariate multiquadric quasi-interpolation scheme to solve the one-dimensional nonlinear sine-Gordon equation that is related to many physical phenomena. We obtain a numerical scheme by using the derivative of the quasi-interpolation to approximate the spatial derivative and a difference scheme to approximate the temporal derivative. The advantage of the obtained scheme is that the algorithm is very simple so that it is very easy to implement. The results of numerical experiments are presented and compared with analytical solutions to confirm the good accuracy of the presented scheme.

Keywords: quasi-interpolation, Hardy Multiquadric (MQ) interpolation methods, sine-Gordon equations, scattered data approximation, meshless method

PACC: 0200

1. Introduction

As is well known, the investigation of nonlinear partial differential equations that describe many important phenomena and dynamical processes in many different fields, such as particle vibrations in lattices, currents in electrical networks, pulses in biological chains and so on, plays an important role in studying modern physics. In the past decades, many numerical methods have been developed successfully to solve virtually most well-posed partial differential equations (PDEs) that are related to many real-life physical problems.[1−3] They include the effective and broadly applicable techniques like the finite difference (FD) method, finite elements (FE) method, boundary elements (BE) method, and finite volume (FV) method. Those methods, however, depend on a suitable generation of meshes, which is difficult for problems with very complicated and irregular geometries. To avoid the generation of meshes, recently many researchers have switched to using meshless techniques.[4,5] The essence of meshless methods is to avoid the generation of meshes by constructing the approximation only in terms of nodes. That is to say, in a meshless method a set of nodes are used instead of meshing the domain of the problem. Multiquadric quasi-interpolation, one of meshless methods, possesses some advantages compared with other approaches, such as less computation complexity, better shape-preserving properties, etc. With so many good properties, the basic theoretical study of multiquadric quasi-interpolation methods has been extensively studied[6,7] and a lot of applications can be found in Ref.[8] and the references therein. The present paper is devoted to the numerical computation of a one-dimensional time-dependent nonlinear undamped sine-Gordon equation. In our numerical scheme, we use the derivative of the quasi-interpolation to approximate the spatial derivative and a low order forward difference scheme to approximate the temporal derivative. The one-dimensional sine-Gordon equation is given by

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} - \sin(u) \tag{1}$$

with $x \in [a, b]$, for $t > t_0$. This equation is related to many scientific research topics such as the propagation of fluxion in Josephson junctions between two superconductors, the motion of a rigid pendulum attached to a stretched wire, solid state physics, nonlinear optics, and dislocations in metals.[9,10] The one-dimensional sine-Gordon equation first arose in a

*Project supported by the State Key Development Program for Basic Research of China (Grant No 2006CB303102), and Science and Technology Commission of Shanghai Municipality, China (Grant No 09DZ2272900).

†Corresponding author. E-mail: malimin@fudan.edu.cn

strict mathematical context of differential geometry about the theory of surfaces of constant curvature.[11]

The remainder of the present paper is organized as follows. In Section 2 the multiquadric quasi-interpolation is introduced, and the numerical techniques are developed, where multiquadric quasi-interpolation is used to solve the sine-Gordon equation. To demonstrate the efficiency and the accuracy of this method, a numerical example is given in Section 3. Finally, brief conclusions drawn from the present study are presented in Section 4.

2. Multiquadric quasi-interpolation and the construction of a numerical scheme

Multiquadric functions were first proposed by Hardy,[12] and they have been performed very well in many calculations including the numerical experiments that were reported by Franke in Ref.[13]. For the good performance of multiquadric function, multiquadric quasi-interpolation is one of the most frequently applied approaches in modern approximation theory when the task is to approximate scattered data in one-dimensional problems. The multiquadric quasi-interpolation of a function $u : [x_0, x_N] \to R$ with multiquadrics at the scattered points

$$a = x_0 < x_1 < \cdots < x_N = b$$

and

$$h := \max_{0\le j\le N}(x_j - x_{j-1})$$

usually takes the form

$$(Lu)(x) = \sum_{j=0}^{N} u(x_j)\psi_j(x),$$

where $\psi_j(x)$ is linear combinations of the multiquadrics

$$\phi_j(x) = \sqrt{c^2 + (x - x_j)^2},$$

in which c is a positive constant. Beatson and Powell[6] proposed three univariate multiquadric quasi-interpolations, namely, L_A, L_B and L_C, to approximate a function $\{u(x), x_0 \le x \le x_N\}$ from the space that is spanned by the multiquadrics $\phi_j(x) = \sqrt{c^2 + (x - x_j)^2}$ and a linear function, where c is a positive constant and the centres $\{x_j : j = 0, \ldots, N\}$ are the given distinct points in the interval $[x_0, x_N]$. Subsequently, Wu and Schaback[7] proposed the univariate multiquadric quasi-interpolation L_D in the interval $[x_0, x_N]$ and proved that the scheme is shape-preserving and convergent. However, in these papers, multiquadric quasi-interpolation is discussed only on the function itself but not on its derivatives. To solve PDEs numerically, the present paper is devoted to constructing a new multiquadric quasi-interpolation scheme, named L_{E}, which possesses the property

$$u^{(j)}(x) \approx (L_E u)^{(j)}(x), \quad j = 0, 1, 2, \ldots .$$

Since $\sqrt{c^2 + x^2}$ is close to $|x|$ as c approaches to zero and multiquadric quasi-interpolation (as well as multiquadric interpolation) based on $|x|$ is piecewise linear interpolation, the convergence property to derivatives can be expected to hold for quasi-interpolation with multiquadric function, too. As done in Refs.[6] and [7], in this paper the normalized second divided difference is defined as

$$\psi_j(x) = \frac{\phi_{j+1}(x) - \phi_j(x)}{2(x_{j+1} - x_j)} - \frac{\phi_j(x) - \phi_{j-1}(x)}{2(x_j - x_{j-1})}, \quad x \in R,$$

for every integer j. Given that $\{(x_j, u(x_j))\}_{j=0}^{N}$, where $a = x_0 < x_1 < \ldots < x_N = b$, we construct the multiquadric quasi-interpolation L_{E}. We start by considering the following multiquadric quasi-interpolation scheme:

$$(Lu)(x) = \sum_{j=0}^{N} u(x_j)\psi_j(x), \quad x \in R. \tag{2}$$

Buhmann[14] studied the accuracy of the quasi-interpolation operator L of a function $u : R \mapsto R$ at infinite regular points in R and showed that this scheme reproduces linear polynomials. However, the power to reproduce linear polynomials does not require the centres $\{x_j\}$ to be equally spaced when there are infinitely many of them, say $\{x_j, j \in Z\}$ that extend to both ends of the real line. With the centres being a strictly increasing sequence, Powell[15] proved that the quasi-interpolation operator L satisfied the identity $(Lu)(x) = u(x)$ when $u(x)$ was a linear polynomial. The estimating of convergence, however, needs a more difficult analysis than the familiar one from the spline theory for instance, because the compact support of the basis functions makes the proof techniques much simpler. Note the asymptotic decay at an algebraic rate of the basis function, which is not a radial basis function but a linear combination of its translations, so it is important to distinguish carefully

between those. The basis function we use in the approximation is our $\psi_j(x)$. Using this decay in tandem with polynomial recovery of a nontrivial order, one can prove the convergence. For more details, readers may refer to Ref.[14].

In application, the data always come from an interval $[x_0, x_N]$ but not from the infinite domain R. To handle this, we introduce a polynomial $p(x)$ satisfying

$$\begin{aligned}
&p(x_0) = u(x_0), \quad p(x_N) = u(x_N),\\
&p(x_1) = u(x_1), \quad p(x_{N-1}) = u(x_{N-1}),\\
&p(x_2) = u(x_2), \quad p(x_{N-2}) = u(x_{N-2}),\\
&p(x_3) = u(x_3), \quad p(x_{N-3}) = u(x_{N-3}),\\
&p(x_4) = u(x_4), \quad p(x_{N-4}) = u(x_{N-4}).
\end{aligned}$$

With polynomial $p(x)$ in hand, we extend the function $u(x)$ from $[x_0, x_N]$ to domain $(-\infty, \infty)$:

$$u^*(x) = \begin{cases} p(x), & \text{if} \quad x \in (-\infty, x_0),\\ u(x), & \text{if} \quad x \in [x_0, x_N],\\ p(x), & \text{if} \quad x \in (x_N, \infty). \end{cases} \tag{3}$$

We apply the presented operator L to function $u^*(x) - p(x)$. Note that function $u^*(x) - p(x)$ is a compactly supported function with $[x_4, x_{N-4}]$ as its support. Moreover, $u^*(x) - p(x) = 0$ when $x = x_4, x_{N-4}$. Now we present our multiquadric quasi-interpolation scheme $(L_{\mathrm{E}}u)(x)$:

$$(L_E u)(x) = \sum_{j=5}^{N-5} (u(x_j) - p(x_j)) \cdot \psi_j(x) + p(x),$$

in which $x \in [x_0, x_N]$. The scheme L_{E} possesses the property that $L_{\mathrm{E}}u$ converges to u in a satisfactory approximation degree. We should mention that, with the introduced polynomial $p(x)$ terms $u(x_0)$, $u(x_1)$, $u(x_2)$, $u(x_3)$, $u(x_4)$, $u(x_N)$,$u(x_{N-1})$, $u(x_{N-2})$, $u(x_{N-3})$ and $u(x_{N-4})$ disappearing, their roles are represented impliticly by $p(x)$. Apparently, in application one can use the second derivative of $(L_{\mathrm{E}}u)(x)$

$$(L_E u)''(x) = \sum_{j=5}^{N-5} (u(x_j) - p(x_j)) \cdot \psi_j''(x) + p''(x),$$

where $x \in [x_0, x_N]$, to approximate $u''(x)$.

Now, we present the numerical scheme for solving the sine-Gordon equation by using the multiquadric quasi-interpolation scheme. In our method, we use the derivative of the multiquadric quasi-interpolation to approximate the spatial derivative of the differential equations and employ a first order accurate forward difference to approximate the temporal derivative.

Discretizing the sine-Gordon equation

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} - \sin(u)$$

in the time domain yields

$$u_k^{n+1} = 2u_k^n - u_k^{n-1} + (\Delta t)^2 (u_{xx})_k^n - (\Delta t)^2 \sin(u_k^n),$$

where u_k^n is the approximation of the value of $u(x,t)$ at point (x_k, t_n), $t_n = n\Delta t$, and Δt is the time step. Because equation (1) is a developmental equation, it is reasonable to employ the finite difference method in the time domain. Then, we use the derivatives of the multiquadric quasi-interpolation to approximate u_{xx}. Thus, the resulting numerical scheme is

$$\begin{aligned}
u_k^{n+1} = {} & 2u_k^n - u_k^{n-1}\\
& + (\Delta t)^2 \Big[\sum_{j=5}^{N-5} (u_j^n - p_n(x)) \cdot \psi_k''(x) + p_n''(x_k) \Big]\\
& - (\Delta t)^2 \sin(u_k^n).
\end{aligned} \tag{4}$$

We iterate this scheme to achieve the numerical solution.

3. Numerical example

To test the accuracy and the efficiency of the multiquadric quasi-interpolation scheme, it is applied to an example having an analytical solution. We consider Eq.(1) in the domain $-2 \le x \le 2$. The initial conditions are given by

$$\begin{aligned}
&u(x,0) = f(x) = 0,\\
&u_t(x,0) = g(x) = 4\,\mathrm{sech}(x).
\end{aligned}$$

The solitary wave solution of the sine-Gordon equation is

$$u(x,t) = 4\arctan(\mathrm{sech}(x)t).$$

In our computation, we choose $\Delta x = 0.01$ and $\Delta t = 0.01$. Figures 1–4 show the error function $\widetilde{u}(x,t) - u(x,t)$, where $u(x,t)$ is the exact solution and $\widetilde{u}(x,t)$ is the numerical solution obtained by using multiquadric quasi-interpolation method with $\Delta x = 0.01$ and $c = 0.1\sqrt[3]{\Delta x}$, at $t = 0.25$, 0.50, 0.75, 1.00 separately.

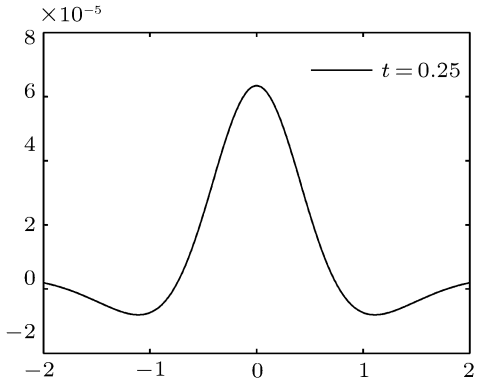


Fig.1. Error function $\widetilde{u}(x,t)-u(x,t)$ where $u(x,t)$ is an exact solution and solution $\widetilde{u}(x,t)$ is obtained by using the multiquadric quasi-interpolation method with $\Delta x = 0.01$ and $c = 0.1\sqrt[3]{\Delta x}$, at $t = 0.25$.

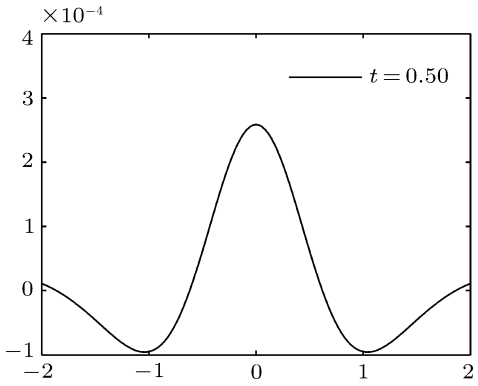


Fig.2. Error function $\widetilde{u}(x,t)-u(x,t)$, where $u(x,t)$ is an exact solution and solution $\widetilde{u}(x,t)$ is obtained by using the multiquadric quasi-interpolation method with $\Delta x = 0.01$ and $c = 0.1\sqrt[3]{\Delta x}$, at $t = 0.50$.

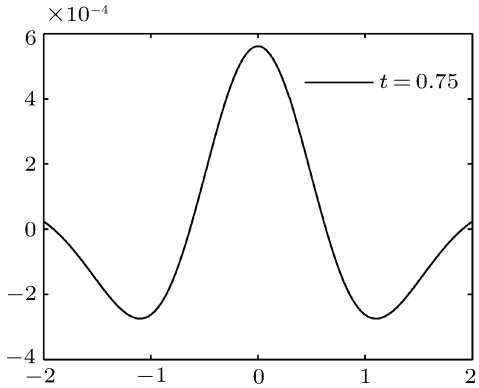


Fig.3. Plots error function $\widetilde{u}(x,t)-u(x,t)$, where $u(x,t)$ is an exact solution and solution $\widetilde{u}(x,t)$ is obtained by using the multiquadric quasi-interpolation method with $\Delta x = 0.01$ and $c = 0.1\sqrt[3]{\Delta x}$, at $t = 0.75$.

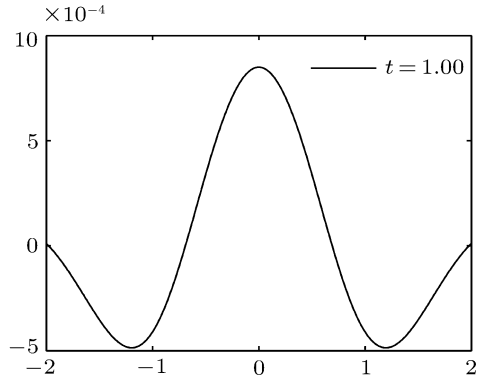


Fig.4. Error function $\widetilde{u}(x,t)-u(x,t)$, where $u(x,t)$ is an exact solution and solution $\widetilde{u}(x,t)$ is obtained by using the multiquadric quasi-interpolation method with $\Delta x = 0.01$ and $c = 0.1\sqrt[3]{\Delta x}$, at $t = 1.00$.

The root-mean-square error $E_2(t_n)$ and the max error $E_\infty(t_n)$ are usually defined as

$$E_2(t_n) = \frac{1}{N+1}\sqrt{\sum_{j=0}^{N}(u(x_j,t_n)-\widetilde{u}(x_j,t_n))^2}$$

and

$$E_\infty(t_n) = \max_{0\le j\le N}|u(x_j,t_n)-\widetilde{u}(x_j,t_n)|.$$

where $u(x_k,t_n)$ is an exact solution and u_k^n is a numerical solution obtained by using the multiquadric quasi-interpolation scheme. The values of E_2 and E_∞ at $t_n = 0.05, 0.10, 0.15, \ldots, 1.00$ are given in Table 1.

Table 1. Errors between the exact solution and the numerical solution at $t_n = 0.05, 0.10, 0.15, \ldots, 1.00$ with $\Delta x = 0.01$ and $c = 0.1\sqrt[3]{\Delta x}$.

t	E_2	E_∞
0.05	3.4189×10^{-6}	6.9168×10^{-6}
0.10	7.4296×10^{-6}	1.5378×10^{-5}
0.15	1.1755×10^{-5}	2.6823×10^{-5}
0.20	1.7617×10^{-5}	4.2499×10^{-5}
0.25	2.5517×10^{-5}	6.3391×10^{-5}
0.30	3.6039×10^{-5}	9.0176×10^{-5}
0.35	4.9583×10^{-5}	1.2320×10^{-4}
0.40	6.6359×10^{-5}	1.6248×10^{-4}
0.45	8.6411×10^{-5}	2.0773×10^{-4}
0.50	1.0965×10^{-4}	2.5840×10^{-4}
0.55	1.3585×10^{-4}	3.1370×10^{-4}
0.60	1.6473×10^{-4}	3.7271×10^{-4}
0.65	1.9589×10^{-4}	4.3438×10^{-4}
0.70	2.2890×10^{-4}	4.9759×10^{-4}
0.75	2.6327×10^{-4}	5.6123×10^{-4}
0.80	2.9849×10^{-4}	602418×10^{-4}
0.85	3.3405×10^{-4}	6.8539×10^{-4}
0.90	3.6941×10^{-4}	7.4389×10^{-4}
0.95	4.0407×10^{-4}	7.9879×10^{-4}
1.00	4.3753×10^{-4}	8.4930×10^{-4}

From figures 1–4 and table 1, we conclude that the methods are feasible and valid.

4. Concluding remarks

In this paper, the multiquadric quasi-interpolation method is used for solving a one-dimensional nonlinear sine-Gordon equation. The meshless feature of the multiquadric quasi-interpolation method is the most important advantage of this scheme over the traditional mesh-dependent techniques such as finite difference, finite element and boundary element methods. At the same time, it still possesses high accuracy. Compared with the traditional meshless collocation method in which the collocation nodes are interpolated, the quasi-interpolation method does not require to solve a large-scale linear system of equations.

For comparison, our technique is used for the non-equidistant grids though we use equidistant grids in our numerical experiments. We see that the present technique requires calculating

$$\psi_j(x) = \frac{\phi_{j+1}(x) - \phi_j(x)}{2(x_{j+1} - x_j)} - \frac{\phi_j(x) - \phi_{j-1}(x)}{2(x_j - x_{j-1})}$$

and

$$\psi_j''(x) = \frac{\phi_{j+1}''(x) - \phi_j''(x)}{2(x_{j+1} - x_j)} - \frac{\phi_j''(x) - \phi_{j-1}''(x)}{2(x_j - x_{j-1})},$$

$(j = 1, \ldots, N)$ only once.

References

[1] Xie Y X and Tang J S 2005 *Chin. Phys.* **14** 1303

[2] Yang Y N, Yang B, Zhu J R, Shen Z H, Lu J and Ni X W 2008 *Chin. Phys.* B **17** 1318

[3] Yao X H, Han Q and Xin H 2008 *Acta Phys. Sin.* **57** 329 (in Chinese)

[4] Wu Z M 1992 *Anal. Theor. Appl.* **8** 1

[5] Dehghan M and Shokri Ali 2007 *Numer. Methods Partial Differential Equations* **24** 687

[6] Beatson R K and Powell M J D 1992 *Constru. Approx.* **8** 275

[7] Wu Z M and Schaback R 1994 *Acta Math. Appl. Sin.* **10** 441

[8] Fasshauer G E 2007 *Meshfree Approximation Methods with MATLAB* (Singapore, Hong Kong: World Scientific)

[9] Perring J K and Skyrme T H A 1962 *Nucl Phys.* **31** 550

[10] Wazwaz A M 2005 *Appl. Math. Comput.* **167** 1196

[11] Hu H C and Lou S Y 2005 *Phys. Lett.* **341** 422

[12] Hardy R L 1971 *J. Geophy. Res.* **76** 1905

[13] Franke R 1982 *Math. Comp.* **38** 181

[14] Buhmann M D 1988 *IMA J. Numer. Anal.* **8** 365

[15] Powell M J D 1990 *Multivariate Approximation and Interpolation* (Basel: Birkhäuser Verlag) p227