

MQ拟插值算子求解Sine-Gordon方程的拉格朗日乘子方法

岳媛^{1*}, 丁洁玉^{2#}

¹青岛大学数学与统计学院, 山东 青岛

²青岛大学计算机科学技术学院, 山东 青岛

收稿日期: 2026年2月3日; 录用日期: 2026年2月26日; 发布日期: 2026年3月4日

摘要

为提高算法的长时间仿真能力, 基于三次MQ拟插值算子构造求解Sine-Gordon方程的一种拉格朗日乘子方法。首先使用拉格朗日乘子法处理边界条件, 然后用三次MQ拟插值算子及其导数逼近函数本身及其空间导数, 最后用四阶龙格-库塔法进行时间离散。数值实验结果表明拉格朗日乘子方法更能还原问题的非线性特征、稳定性更高和更适合长时间仿真。

关键词

径向基函数, MQ拟插值算子, Sine-Gordon方程, 拉格朗日乘子法

Lagrange Multiplier Method Based on MQ Quasi-Interpolation Operator for Solving Sine-Gordon Equation

Yuan Yue^{1*}, Jieyu Ding^{2#}

¹School of Mathematics and Statistics, Qingdao University, Qingdao Shandong

²College of Computer Science and Technology, Qingdao University, Qingdao Shandong

Received: February 3, 2026; accepted: February 26, 2026; published: March 4, 2026

Abstract

To improve the long-term simulation capability of existing numerical algorithms, the method of

*第一作者。

#通讯作者。

文章引用: 岳媛, 丁洁玉. MQ拟插值算子求解 Sine-Gordon 方程的拉格朗日乘子方法[J]. 应用数学进展, 2026, 15(3): 59-69. DOI: 10.12677/aam.2026.153087

Lagrange multipliers based on cubic MQ quasi-interpolation operator is proposed for solving the Sine-Gordon equation. In this approach, the boundary conditions are first handled using the method of Lagrange multipliers. Subsequently, the cubic MQ quasi-interpolation operator and its derivatives are employed to approximate the solution and its spatial derivatives. Temporal discretization is then performed using the fourth-order Runge-Kutta scheme. The numerical results demonstrate that the method of Lagrange multipliers more effectively captures the nonlinear characteristics of the equation, achieves higher stability, and is particularly well suited for long-term simulations.

Keywords

Radial Basis Function, MQ Quasi-Interpolation Operator, Sine-Gordon Equation, The Method of Lagrange Multipliers

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

在函数逼近方法中, 插值法因结构相对简单、数值稳定性高、精度高等特点得到广泛研究和应用。常用的代数多项式和三角多项式函数空间在处理多元问题时刚性太强, 计算不稳定, 因此, 稳定性高和易于实现的径向基函数空间受到广泛关注。

1982年, Franke [1] 得出 MQ 径向基函数插值方法的精度和稳定性较其他方法更优。但是在应用径向基函数插值时需要求解线性方程组, 随着插值节点的增多, 系数矩阵趋于病态, 此时径向基函数插值方法不具有数值稳定性。因此需要使用拟插值方法, 其中 MQ 拟插值方法不仅可以避免求解线性方程组, 并且具有多项式再生性、保形性和令人满意的收敛速度。1992年, Beatson 和 Powell [2] 提出了三种一次 MQ 拟插值算子 L_A, L_B, L_C 。1994年, Wu 和 Schaback [3] 构造了精度不低于上述三种算子且不需要端点处导数值的一次新算子 L_D 。一次 MQ 拟插值算子在处理大规模、高维度的问题时难以提供足够的精度, 有时还会出现数值结果不稳定的现象。

相比于一次 MQ 拟插值算子, 三次 MQ 拟插值算子能更好地还原问题的非线性特征, 具有更高的数值稳定性。2009年, Feng 和 Li [4] 基于三次 MQ-B 样条构造了具有二次多项式再生性和三阶保形性的拟插值算子 L_d 。应用 MQ 拟插值算子数值求解偏微分方程时, 常使用 MQ 拟插值算子进行空间离散, 再结合向前欧拉法进行时间离散[5]-[7], 得到的数值结果的精度有限, 且随着求解时间步长的增加, 误差会累积。因此, 本文提出一种高精度的数值求解方法, 使用拉格朗日乘法处理边界条件、三次 MQ 拟插值算子进行空间离散, 最后使用四阶龙格-库塔法进行时间离散。

2. 拟插值算子

由给定数据 $\{x_i, f_i\}, f_i = f(x_i), i = 0, 1, \dots, n, x_0 < x_1 < \dots < x_n, h = \max_{1 \leq i \leq n} \{x_i - x_{i-1}\}$, c_1 为正的常数, 定义一次 MQ 拟插值算子 L_D 为:

$$L_D f(x) = \sum_{i=0}^n f_i \delta_i(x), x \in R,$$

其中

$$\begin{aligned}
\delta_0(x) &= \frac{1}{2} + \frac{\phi_1(x) - (x - x_0)}{2(x_1 - x_0)}, \\
\delta_1(x) &= \frac{\phi_2(x) - \phi_1(x)}{2(x_2 - x_1)} - \frac{\phi_1(x) - (x - x_0)}{2(x_1 - x_0)}, \\
\delta_{n-1}(x) &= \frac{(x_n - x) - \phi_{n-1}(x)}{2(x_n - x_{n-1})} - \frac{\phi_{n-1}(x) - \phi_{n-2}(x)}{2(x_{n-1} - x_{n-2})}, \\
\delta_n(x) &= \frac{1}{2} + \frac{\phi_{n-1}(x) - (x_n - x)}{2(x_n - x_{n-1})}, \\
\delta_i(x) &= \frac{\phi_{i+1}(x) - \phi_i(x)}{2(x_{i+1} - x_i)} - \frac{\phi_i(x) - \phi_{i-1}(x)}{2(x_i - x_{i-1})}, i = 2, \dots, n-2, \\
\phi_i(x) &= \sqrt{(x - x_i)^2 + c_i^2}.
\end{aligned} \tag{1}$$

由给定数据 $\{x_i, f_i\}$, $f_i = f(x_i)$, $i = 0, 1, \dots, n$, $h = \max_{1 \leq i \leq n} \{x_i - x_{i-1}\}$, c_2 为正的常数, 定义三次 MQ 拟插值算子 L_d [4] 为:

$$L_d f(x) = \sum_{i=0}^n f_i \varepsilon_i(x), x \in R,$$

其中

$$\begin{aligned}
\varepsilon_0(x) &= \frac{1}{2} + \frac{(x - x_0)^2 - \omega_1(x)}{2(x_2 - x_0)(x_1 - x_0)} - \frac{x - x_0}{2(x_2 - x_0)} - \frac{x - x_0}{2(x_1 - x_0)}, \\
\varepsilon_1(x) &= \frac{\omega_1(x) - \omega_2(x)}{2(x_3 - x_1)(x_2 - x_1)} - \frac{(x - x_0)^2 - (x_1 - x_0)(x - x_0) - \omega_1(x)}{2(x_2 - x_0)(x_2 - x_1)} \\
&\quad - \frac{(x - x_0)^2 - \omega_1(x)}{2(x_2 - x_0)(x_1 - x_0)} + \frac{x - x_0}{2(x_2 - x_0)} + \frac{x - x_0}{2(x_1 - x_0)}, \\
\varepsilon_2(x) &= \frac{\omega_2(x) - \omega_3(x)}{2(x_4 - x_2)(x_3 - x_2)} - \frac{\omega_1(x) - \omega_2(x)}{2(x_3 - x_2)(x_3 - x_1)} - \frac{\omega_1(x) - \omega_2(x)}{2(x_3 - x_1)(x_2 - x_1)} \\
&\quad + \frac{(x - x_0)^2 - (x_1 - x_0)(x - x_0) - \omega_1(x)}{2(x_2 - x_0)(x_2 - x_1)}, \\
\varepsilon_i(x) &= \frac{\omega_i(x) - \omega_{i+1}(x)}{2(x_{i+2} - x_i)(x_{i+1} - x_i)} - \frac{\omega_{i-1}(x) - \omega_i(x)}{2(x_{i+1} - x_i)(x_{i+1} - x_{i-1})} \\
&\quad - \frac{\omega_{i-1}(x) - \omega_i(x)}{2(x_{i+1} - x_{i-1})(x_i - x_{i-1})} + \frac{\omega_{i-2}(x) - \omega_{i-1}(x)}{2(x_i - x_{i-1})(x_i - x_{i-2})}, 3 \leq i \leq n-3, \\
\varepsilon_{n-2}(x) &= \frac{(x_n - x)^2 + \omega_{n-2}(x) - (x_n - x_{n-1})(x_n - x)}{2(x_n - x_{n-2})(x_{n-1} - x_{n-2})} - \frac{\omega_{n-3}(x) - \omega_{n-2}(x)}{2(x_{n-1} - x_{n-2})(x_{n-1} - x_{n-3})} \\
&\quad - \frac{\omega_{n-3}(x) - \omega_{n-2}(x)}{2(x_{n-1} - x_{n-3})(x_{n-2} - x_{n-3})} + \frac{\omega_{n-4}(x) - \omega_{n-3}(x)}{2(x_{n-2} - x_{n-3})(x_{n-2} - x_{n-4})}, \\
\varepsilon_{n-1}(x) &= \frac{x_n - x}{2(x_n - x_{n-1})} + \frac{x_n - x}{2(x_n - x_{n-2})} - \frac{(x_n - x)^2 + \omega_{n-2}(x)}{2(x_n - x_{n-1})(x_n - x_{n-2})} \\
&\quad - \frac{(x_n - x)^2 + \omega_{n-2}(x) - (x_n - x_{n-1})(x_n - x)}{2(x_n - x_{n-2})(x_{n-1} - x_{n-2})} + \frac{\omega_{n-3}(x) - \omega_{n-2}(x)}{2(x_{n-1} - x_{n-3})(x_{n-1} - x_{n-2})},
\end{aligned} \tag{2}$$

(3)

$$\begin{aligned}\varepsilon_n(x) &= \frac{1}{2} + \frac{(x_n - x)^2 + \omega_{n-2}(x)}{2(x_n - x_{n-1})(x_n - x_{n-2})} - \frac{x_n - x}{2(x_n - x_{n-1})} - \frac{x_n - x}{2(x_n - x_{n-2})}, \\ \omega_i(x) &= \frac{\varphi_i(x) - \varphi_{i+1}(x)}{x_{i+1} - x_i}, 1 \leq i \leq n-2, \\ \varphi_i(x) &= \frac{1}{3} \phi_{i,2}(x) = \frac{1}{3} \left((x - x_i)^2 + c_2^2 \right)^{3/2}.\end{aligned}$$

3. Sine-Gordon 方程

Sine-Gordon 方程是 19 世纪被发现的一种非线性偏微分方程, 可以用于描述多种物理现象, 如约瑟夫森结的量子隧穿[8]、附着在可拉伸线上的刚性摆的运动和金属中的位错现象[9]。Sine-Gordon 方程的精确解只有在特殊情况可以获得, 通常使用数值解法, 如有限差分法、有限元法[10]和 B-样条配置法[11]等。Sine-Gordon 方程的形式如下

$$u_{tt} = u_{xx} - \sin(u), x \in [a, b] \subset \mathbb{R}, t > 0, \quad (4)$$

初始条件为

$$\begin{aligned}u(x, t) &= u_0(x), t = 0, \\ u_t(x, t) &= u_{t0}(x), t = 0,\end{aligned}$$

边界条件为

$$\begin{aligned}u(x, t) &= f(t), x = a, t > 0, \\ u(x, t) &= g(t), x = b, t > 0,\end{aligned}$$

其中 $u_0(x), u_{t0}(x), f(t), g(t)$ 是已知的函数。

4. 拉格朗日乘子法

首先按空间步长 $h = (b - a)/n$ 对(4)进行离散, 可以得到

$$u_{tt}(x_i, t) = u_{xx}(x_i, t) - \sin(u(x_i, t)), \quad (5)$$

其中 $x_i = a + ih, i = 0, 1, \dots, n$ 。

边界条件可以用以下约束矩阵近似

$$\Phi = \begin{pmatrix} u(x_0, t) - f(t) \\ u(x_n, t) - g(t) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

将此约束矩阵和(5)联立, 引入拉格朗日乘子, 化为如下方程组

$$\begin{cases} U_{tt} + \Phi_u^T \lambda = U_{xx} - \sin(U), \\ \Phi = 0 \end{cases},$$

其中

$$U_{tt} = \begin{pmatrix} u_{tt}(x_0, t) \\ u_{tt}(x_1, t) \\ \vdots \\ u_{tt}(x_n, t) \end{pmatrix}, U_{xx} = \begin{pmatrix} u_{xx}(x_0, t) \\ u_{xx}(x_1, t) \\ \vdots \\ u_{xx}(x_n, t) \end{pmatrix}, U = \begin{pmatrix} u(x_0, t) \\ u(x_1, t) \\ \vdots \\ u(x_n, t) \end{pmatrix}, \Phi_u = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}, \lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix}.$$

为联立求解, 约束矩阵对时间 t 求二阶导

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

计算中, 选择 $\tau = 0.01, h = 0.01$ 。

使用下述三种范数评估数值结果:

$$L_{\infty} = \max_{1 \leq i \leq n} |u_{\text{exact}}(i) - u_{\text{num}}(i)|,$$

$$L_2 = \sqrt{\sum_{i=1}^n (u_{\text{exact}}(i) - u_{\text{num}}(i))^2},$$

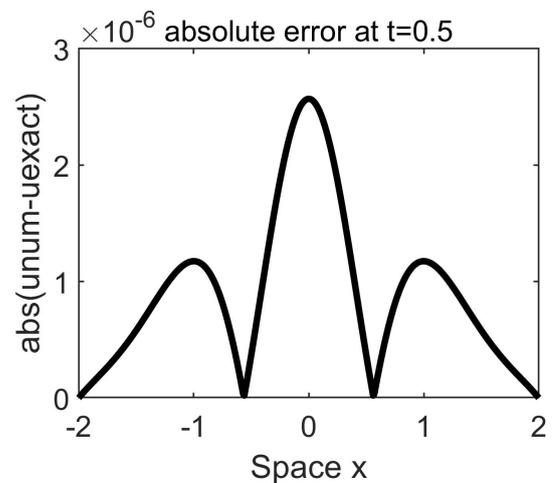
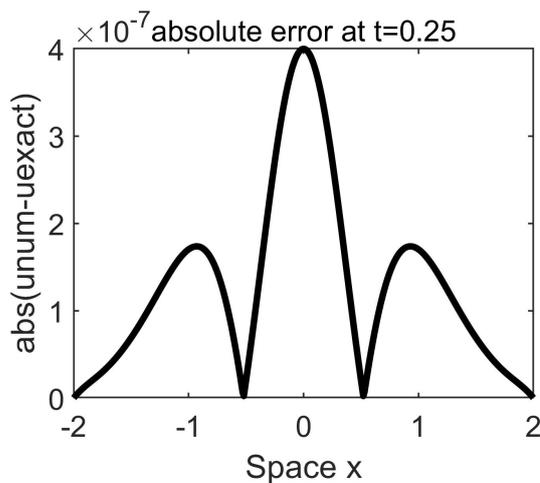
$$\text{RMS} = \sqrt{\left(\sum_{i=1}^n (u_{\text{exact}}(i) - u_{\text{num}}(i))^2 \right) / n}.$$

其中 u_{exact} 是精确解, u_{num} 是数值解。

Table 1. The error between the numerical solution and the exact solution of Example 1 obtained by LLdMQ method, where $c_2 = 1 \times 10^{-8}, \tau = 0.01, h = 0.01$

表 1. 使用 LLdMQ 方法得到的例 1 的数值解与精确解之间的误差, 其中 $c_2 = 1 \times 10^{-8}, \tau = 0.01, h = 0.01$

t	L_{∞}	L_2	RMS
0.1	2.70763×10^{-8}	2.23954×10^{-7}	1.11837×10^{-8}
0.2	2.10389×10^{-7}	1.75245×10^{-6}	8.75132×10^{-8}
0.3	6.68035×10^{-7}	5.64439×10^{-6}	2.81868×10^{-7}
0.4	1.45660×10^{-6}	1.25514×10^{-5}	6.26789×10^{-7}
0.5	2.56886×10^{-6}	2.26791×10^{-5}	1.13254×10^{-6}
0.6	3.94854×10^{-6}	3.58525×10^{-5}	1.79039×10^{-6}
0.7	5.51261×10^{-6}	5.16344×10^{-5}	2.57850×10^{-6}
0.8	7.17236×10^{-6}	6.94448×10^{-5}	3.46791×10^{-6}
0.9	8.84828×10^{-6}	8.86471×10^{-5}	4.42683×10^{-6}
1.0	1.04778×10^{-5}	1.08592×10^{-4}	5.42284×10^{-6}



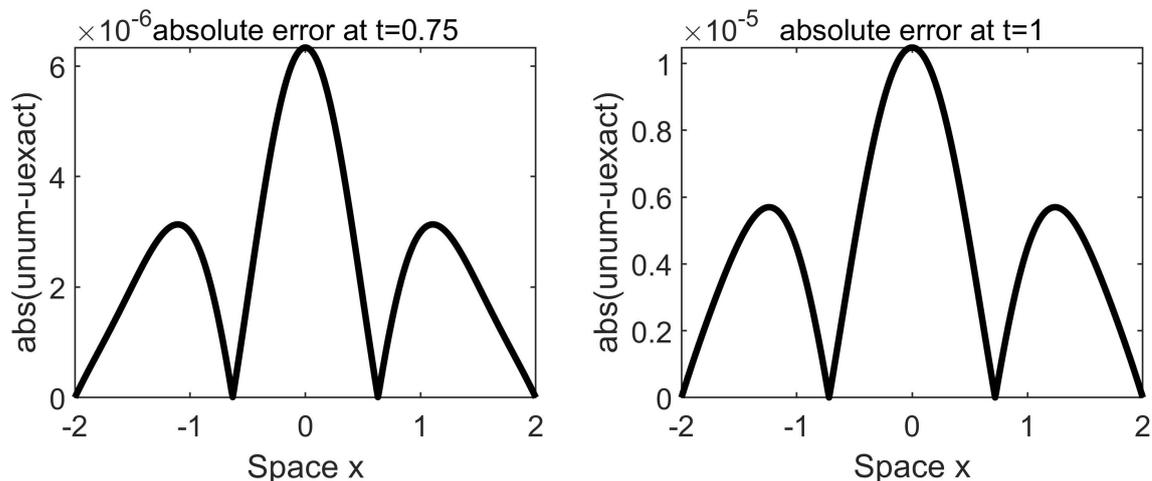


Figure 1. The absolute error between the numerical solution and the exact solution of Example 1 obtained by LLdMQ method, where $c_2 = 1 \times 10^{-8}$, $\tau = 0.01$, $h = 0.01$, $t = 0.25, 0.5, 0.75, 1$

图 1. 使用 LLdMQ 方法得到的例 1 的数值解与精确解之间的绝对误差, 其中 $c_2 = 1 \times 10^{-8}$, $\tau = 0.01$, $h = 0.01$, $t = 0.25, 0.5, 0.75, 1$

从表 1 和图 1 可以得出, LLdMQ 方法在 $t = 0.1$ 到 $t = 1$ 中任意时刻的 L_∞, L_2 和 RMS 误差均较小, 整体精度较高。

Table 2. The L_∞ error obtained by LLdMQ, LdMQ, LDMQ method in Example 1 is compared with that obtained by LEMQ method in Ma's paper, where $\tau = 0.01$, $h = 0.01$

表 2. 使用 LLdMQ, LdMQ, LDMQ 方法得到的例 1 的 L_∞ 误差与 Ma 文中 LEMQ 方法得到的 L_∞ 误差对比, 其中 $\tau = 0.01$, $h = 0.01$

t	LLdMQ	LdMQ	LEMQ [12]	LEMQ
0.1	2.70763×10^{-8}	1.31430×10^{-5}	1.5378×10^{-5}	1.53836×10^{-5}
0.2	2.10389×10^{-7}	2.52045×10^{-5}	4.2499×10^{-5}	4.26424×10^{-5}
0.3	6.68035×10^{-7}	3.54048×10^{-5}	9.0176×10^{-5}	9.10860×10^{-5}
0.4	1.45660×10^{-6}	4.34241×10^{-5}	1.6248×10^{-4}	1.65798×10^{-4}
0.5	2.56886×10^{-6}	4.93755×10^{-5}	2.5840×10^{-4}	2.67298×10^{-4}
0.6	3.94854×10^{-6}	5.36551×10^{-5}	3.7271×10^{-4}	3.92352×10^{-4}
0.7	5.51261×10^{-6}	5.67684×10^{-5}	4.9759×10^{-4}	5.35366×10^{-4}
0.8	7.17236×10^{-6}	5.91988×10^{-5}	6.2418×10^{-4}	6.89835×10^{-4}
0.9	8.84828×10^{-6}	6.13392×10^{-5}	7.4389×10^{-4}	8.49459×10^{-4}
1.0	1.04778×10^{-5}	6.34724×10^{-5}	7.4389×10^{-4}	1.00883×10^{-3}

从表 2 可以得出, LLdMQ 方法在 $t = 0.1$ 到 $t = 1$ 中任意时刻的 L_∞ 误差均优于 LdMQ, LEMQ [12], LDMQ 方法, 其中 LLdMQ 方法在 $t = 0.1$ 到 $t = 0.9$ 中任意时刻的 L_∞ 误差均比 LdMQ 方法至少小一个数量级。

Table 3. The error between the numerical solution and the exact solution of Example 1 obtained by LLdMQ method, where $c_2 = 1 \times 10^{-8}, \tau = 0.01, h = 0.01$

表 3. 使用 LLdMQ 方法得到的例 1 的数值解与精确解之间的误差, 其中 $c_2 = 1 \times 10^{-8}, \tau = 0.01, h = 0.01$

t	L_∞	L_2	RMS
1	1.04778×10^{-5}	1.08592×10^{-4}	5.42284×10^{-6}
2	2.03397×10^{-5}	2.47381×10^{-4}	1.23536×10^{-5}
3	3.92021×10^{-5}	5.98542×10^{-4}	2.98897×10^{-5}
4	5.15423×10^{-5}	7.28496×10^{-4}	3.63794×10^{-5}
5	2.46843×10^{-5}	3.65894×10^{-4}	1.82719×10^{-5}
6	3.07180×10^{-5}	3.54078×10^{-4}	1.76818×10^{-5}
7	3.92121×10^{-5}	6.46678×10^{-4}	3.22936×10^{-5}
8	2.26343×10^{-5}	2.27981×10^{-4}	1.13848×10^{-5}
9	3.36228×10^{-5}	4.49253×10^{-4}	2.24346×10^{-5}
10	3.79370×10^{-5}	4.91271×10^{-4}	2.45329×10^{-5}

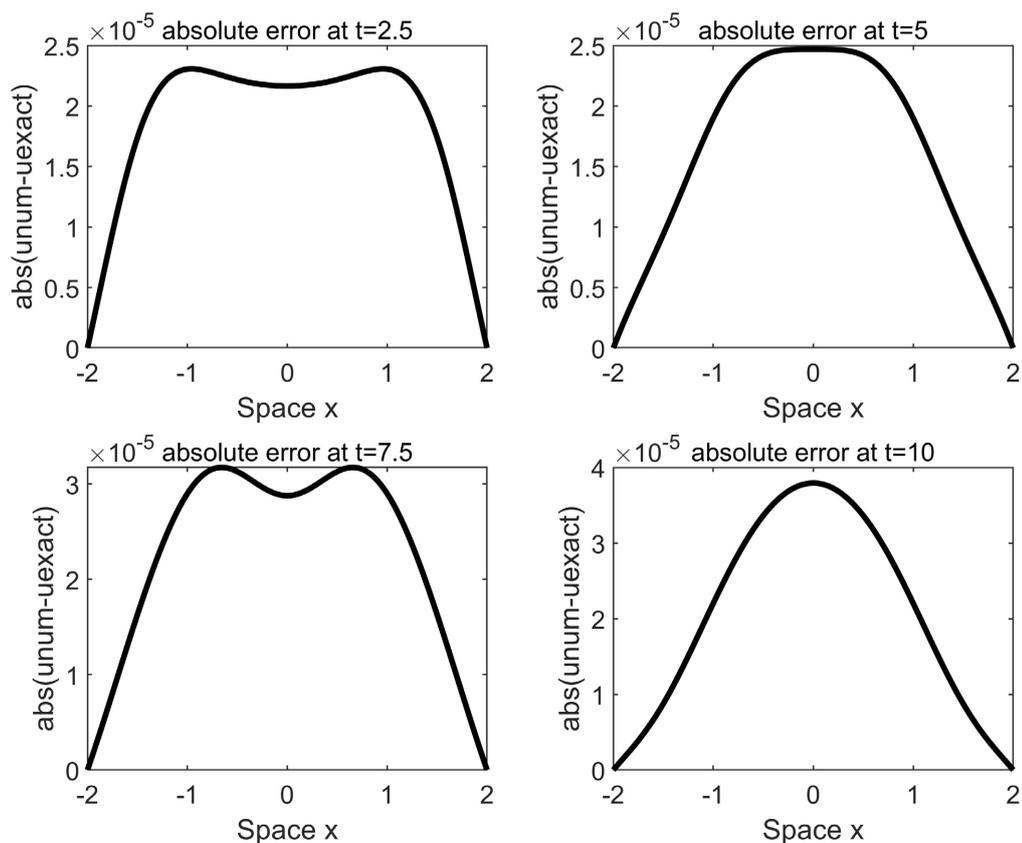


Figure 2. The absolute error between the numerical solution and the exact solution of Example 1 obtained by LLdMQ method, where $c_2 = 1 \times 10^{-8}, \tau = 0.01, h = 0.01, t = 2.5, 5, 7.5, 10$

图 2. 使用 LLdMQ 方法得到的例 1 的数值解与精确解之间的绝对误差, 其中 $c_2 = 1 \times 10^{-8}, \tau = 0.01, h = 0.01, t = 2.5, 5, 7.5, 10$

从表 3 和图 2 可以得出, LLdMQ 方法在 $t = 0.1$ 到 $t = 10$ 中任意时刻的 L_∞, L_2 和 RMS 误差与 LdMQ 方法在 $t = 0.1$ 到 $t = 10$ 中任意时刻的 L_∞, L_2 和 RMS 误差是同一个数量级, 能还原非线性特征, 适合非线性偏微分方程的长时间仿真。

Table 4. The error between the numerical solution and the exact solution of Example 1 obtained by LLdMQ method, where $c_2 = 1 \times 10^{-8}, \tau = 0.01, h = 0.01$

表 4. 使用 LLdMQ 方法得到的例 1 的数值解与精确解之间的误差, 其中 $c_2 = 1 \times 10^{-8}, \tau = 0.01, h = 0.01$

t	L_∞	L_2	RMS
10	3.79370×10^{-5}	4.91271×10^{-4}	2.45329×10^{-5}
20	4.12375×10^{-5}	5.22671×10^{-4}	2.61010×10^{-5}
30	3.41203×10^{-5}	5.11947×10^{-4}	2.55654×10^{-5}
40	3.11899×10^{-5}	4.89222×10^{-4}	2.44306×10^{-5}
50	2.92117×10^{-5}	4.58141×10^{-4}	2.28785×10^{-5}
60	2.56762×10^{-5}	4.15502×10^{-4}	2.07492×10^{-5}
70	2.34366×10^{-5}	3.57192×10^{-4}	1.78373×10^{-5}
80	1.82100×10^{-5}	2.81794×10^{-4}	1.40721×10^{-5}
90	1.16020×10^{-5}	1.93533×10^{-4}	9.66457×10^{-6}
100	7.35848×10^{-6}	1.06276×10^{-4}	5.30717×10^{-6}

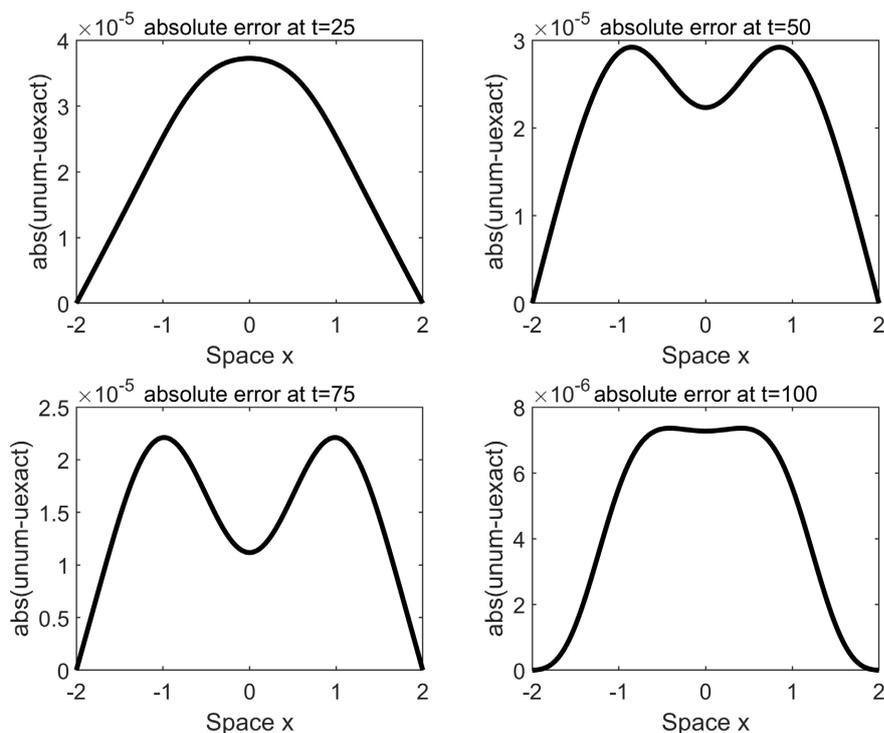


Figure 3. The absolute error between the numerical solution and the exact solution of Example 1 obtained by LLdMQ method, where $c_2 = 1 \times 10^{-8}, \tau = 0.01, h = 0.01, t = 25, 50, 75, 100$

图 3. 使用 LLdMQ 方法得到的例 1 的数值解与精确解之间的绝对误差, 其中 $c_2 = 1 \times 10^{-8}, \tau = 0.01, h = 0.01, t = 25, 50, 75, 100$

从表 4 和图 3 可以得出, LLdMQ 在计算到 $t = 100$, 即 10000 个时间步长后依旧可以保持较高精度, 误差数量级与计算到 $t = 10$ 时基本保持一致, 能还原非线性特征, 适合非线性偏微分方程的长时间仿真。

Table 5. The maximum values of L_∞, L_2 and RMS error of Example 1 obtained by LLdMQ method, where $t = 1, \tau = 0.01, h = 0.01$

表 5. 使用 LLdMQ 方法得到的例 1 的 L_∞, L_2 和 RMS 误差的最大值, 其中 $t = 1, \tau = 0.01, h = 0.01$

c_2	L_∞	L_2	RMS
0.5	3.04090×10^{60}	1.51748×10^{61}	7.57792×10^{59}
0.2	1.19739×10^{-2}	1.21534×10^{-1}	6.06914×10^{-3}
0.1	3.10346×10^{-3}	3.19692×10^{-2}	1.59647×10^{-3}
0.01	5.22194×10^{-5}	5.41140×10^{-4}	2.70232×10^{-5}
1×10^{-4}	1.07921×10^{-5}	1.11849×10^{-4}	5.58549×10^{-6}
1×10^{-8}	1.04778×10^{-5}	1.08592×10^{-4}	5.42284×10^{-6}

从表 5 可以得出, 在 $c_2 = 1 \times 10^{-8}$ 时, LLdMQ 方法的 L_∞, L_2 和 RMS 误差的最大值均最小。本文使用并行计算的方式寻找最优的形状参数 c 。

Table 6. The minimum values of L_∞, L_2 and RMS error of Example 1 obtained by LLdMQ method, where $c_2 = 1 \times 10^{-8}, t = 10, h = 0.01$

表 6. 使用 LLdMQ 方法得到的例 1 的 L_∞, L_2 和 RMS 误差的最小值, 其中 $c_2 = 1 \times 10^{-8}, t = 10, h = 0.01$

τ	L_∞	L_2	RMS	$t_{\text{CPU/s}}$
0.01	1.25531×10^{-11}	1.60005×10^{-10}	7.99026×10^{-12}	3
0.001	2.74208×10^{-14}	2.26144×10^{-13}	1.12931×10^{-14}	27
0.0001	2.78640×10^{-17}	2.28440×10^{-16}	1.14078×10^{-17}	378
0.00001	3.38813×10^{-20}	2.28617×10^{-19}	1.14166×10^{-20}	7888

从表 6 可以得出, 随着 τ 的减小, LLdMQ 方法的 L_∞, L_2 和 RMS 误差的最小值均减小, 但求解所需的时间 t_{CPU} 显著增大。

Table 7. The minimum values of L_∞, L_2 and RMS error of Example 1 obtained by LLdMQ method, where $c_2 = 1 \times 10^{-8}, t = 1$

表 7. 使用 LLdMQ 方法得到的例 1 的 L_∞, L_2 和 RMS 误差的最小值, 其中 $c_2 = 1 \times 10^{-8}, t = 1$

(τ, h)	L_∞	L_2	RMS	$t_{\text{CPU/s}}$
(0.001, 0.01)	2.74208×10^{-14}	2.26144×10^{-13}	1.12931×10^{-14}	3
(0.001, 0.001)	4.38018×10^{-16}	1.04416×10^{-14}	1.65075×10^{-16}	425
(0.0001, 0.01)	2.78640×10^{-17}	2.28440×10^{-16}	1.14078×10^{-17}	31
(0.0001, 0.001)	5.96311×10^{-19}	1.17968×10^{-17}	1.86501×10^{-19}	5375

从表 7 可以得出, 随着 τ 和 h 的减小, LLdMQ 方法的 L_∞, L_2 和 RMS 误差的最小值均减小, 但求解所需的时间 t_{CPU} 显著增大。

6. 总结

本文使用三次 MQ 拟插值算子结合拉格朗日乘子方法对 Sine-Gordon 方程进行数值求解, 通过数值实验, 得出三次 MQ 拟插值算子较一次 MQ 拟插值算子具有更高的精度和数值稳定性, 拉格朗日乘子方法较三次 MQ 拟插值算子结合向前欧拉法更能还原问题的非线性特征、稳定性更高和更适合长时间仿真。后续将进一步研究使用本文提出的拉格朗日乘子方法求解更高维的非线性偏微分方程。

基金项目

国家自然科学基金(12172186, 11772166)。

参考文献

- [1] Franke, R. (1982) Scattered Data Interpolation: Tests of Some Methods. *Mathematics of Computation*, **38**, 181-200. <https://doi.org/10.1090/s0025-5718-1982-0637296-4>
- [2] Beatson, R.K. and Powell, M.J.D. (1992) Univariate Multiquadric Approximation: Quasi-Interpolation to Scattered Data. *Constructive Approximation*, **8**, 275-288. <https://doi.org/10.1007/bf01279020>
- [3] Wu, Z. and Robert, S. (1994) Shape Preserving Properties and Convergence of Univariate Multiquadric Quasi-interpolation. *Acta Mathematicae Applicatae Sinica*, **10**, 441-446. <https://doi.org/10.1007/bf02016334>
- [4] Feng, R. and Li, F. (2009) A Shape-Preserving Quasi-Interpolation Operator Satisfying Quadratic Polynomial Reproduction Property to Scattered Data. *Journal of Computational and Applied Mathematics*, **225**, 594-601. <https://doi.org/10.1016/j.cam.2008.08.024>
- [5] Zhang, J. and Yu, J. (2025) A Multi-Quadrics Quasi-Interpolation Scheme for Numerical Solution of Burgers' Equation. *Applied Numerical Mathematics*, **208**, 38-44. <https://doi.org/10.1016/j.apnum.2024.09.025>
- [6] 樊超凡. 非线性偏微分方程及其数值算法研究[D]: [硕士学位论文]. 大连: 大连交通大学, 2024.
- [7] 李姗姗. 采用拟插值技术模拟二维 Camassa-Holm 方程[D]: [硕士学位论文]. 成都: 电子科技大学, 2023.
- [8] Perring, J.K. and Skyrme, T.H.R. (1962) A Model Unified Field Equation. *Nuclear Physics*, **31**, 550-555. [https://doi.org/10.1016/0029-5582\(62\)90774-5](https://doi.org/10.1016/0029-5582(62)90774-5)
- [9] Wazwaz, A. (2005) The Tanh Method: Exact Solutions of the Sine-Gordon and the Sinh-Gordon Equations. *Applied Mathematics and Computation*, **167**, 1196-1210. <https://doi.org/10.1016/j.amc.2004.08.005>
- [10] Guo, P.F., Liew, K.M. and Zhu, P. (2015) Numerical Solution of Nonlinear Klein-Gordon Equation Using the Element-Free KP-Ritz Method. *Applied Mathematical Modelling*, **39**, 2917-2928. <https://doi.org/10.1016/j.apm.2014.11.025>
- [11] Mittal, R.C. and Bhatia, R. (2014) Numerical Solution of Nonlinear Sine-Gordon Equation by Modified Cubic B-Spline Collocation Method. *International Journal of Partial Differential Equations*, **2014**, Article ID: 343497. <https://doi.org/10.1155/2014/343497>
- [12] Ma, L.M. and Wu, Z.M. (2009) A Numerical Method for One-Dimensional Nonlinear Sine-Gordon Equation Using Multiquadric Quasi-Interpolation. *Chinese Physics B*, **18**, 3099-3103. <https://doi.org/10.1088/1674-1056/18/8/001>