ISSN 2616-5805 Vol. 4, Issue 4: 27-35, DOI: 10.25236/AJMS.2023.040405

A Method for Solving Global Solutions to Partial Differential Equations Based on the Annealing Algorithm and Polynomial Regression for the Feynman-Kac Formulation

Xin Gui[1,*], Shirui Zheng[2]

[1]School of Civil Engineering, Shandong Jianzhu University, Jinan, 250101, China
[2]Zhizhen College, Beihang University, Beijing, 100191, China
[*]Corresponding author: xgui73830@gmail.com

Abstract: *At present, the global solution of PDE is usually obtained by the Feynman-Kac formula and then constructed by the interpolation method. In this paper, a polynomial fitting method based on the annealing algorithm is proposed to replace the interpolation method to construct the global solution. The simulated annealing algorithm is used to determine the coefficients of the fitting polynomials to obtain higher accuracy. The numerical results show that the improved model has less error than the model obtained by the interpolation method, and the annealing algorithm has a great contribution to improving the accuracy. Then the convergence analysis of the time step, simulation number, and polynomial order is carried out, and the result shows that the convergence is good. Finally, because the Feynman-Kac formula is used to determine the numerical solution, the method is expected to be applied to high-dimensional problems.*

Keywords: *Feynman-Kac formula, Polynomial regression, Simulated annealing algorithm*

1. Introduction

In recent years, partial differential equations have played an important role in scientific and engineering problems. To solve partial differential equations, researchers have been improving the solution methods. Traditional solution methods such as the difference method [1] and the finite element method [2] face difficulties in dealing with some problems. The difference method is simple to implement, but the accuracy is limited by the grid size and step size, and the calculation becomes difficult when complex regions and boundary conditions are encountered that require the use of irregular grids. The finite element method is computationally expensive and requires a lot of computational resources. There are also several methods for solving partial differential equations using the Feynman-Kac formulation, a numerical method for solving partial differential equations that relates stochastic differential equations to partial differential equations and is usually used to solve linear partial differential equations with probabilistic interpretations, such as the Black-Scholes equation in option pricing [3]. In addition, the Feynman-Kac formulation can be used in conjunction with reflected Brownian motion to solve numerical solutions of Laplace equations with Robin boundary conditions [4], and in addition, the hybrid probabilistic and deterministic BIE-WOS method proposed in the literature [5] provides a highly parallel algorithm for any hybrid type of BVP for elliptic equations.

Most of the current research do not construct global solutions, or use interpolation methods to construct global solutions. However, due to the randomness of the Feynman-Kac formula, the interpolation points obtained are not necessarily on the original function. If the noise is too large, it is likely to cause a large number of oscillations between the interpolation function and the data points, resulting in a large error. Therefore, a polynomial fitting method based on the annealing algorithm is proposed in this paper, which is used to construct global solutions instead of interpolation. By using the annealing algorithm to select the coefficients of fitting polynomials, higher precision, and stability can be obtained in global solutions. Finally, the numerical simulation results show that the improved model has a lower relative error and a better fit than the interpolation method.

2. Pre-requisite knowledge

2.1 Elliptic equation and Feynman-Kac formula

In this paper, elliptic equations are studied as the object of research. Elliptic equations are widely used in physics, engineering and economics, such as steady-state solutions of heat conduction equations, Laplace's equation in elasticity, Poisson's equation in electric fields, etc. The elliptic equations studied in this paper are of the following form:

$$\begin{cases}\mu(x)\frac{du}{dx}+\frac{1}{2}\sigma^2(x)\frac{d^2u}{dx^2}+f(x)=0 & x\in D\\ u(x)=g(x) & x\in \overline{D}\end{cases} \tag{1}$$

Solving elliptic equations usually involves boundary conditions that describe the physical constraints of the problem. The boundary conditions are the Dirichlet boundary condition (i.e., the value of the function on a given boundary) and the Neumann boundary condition (i.e., the value of the normal guide on a given boundary). It was shown in [10] that in bounded domains with Dirichlet or Neumann boundary conditions, we can use more complex versions of the equations based on Brownian motion to reduce the errors in the equations. Some common methods for solving high-dimensional elliptic equations are, for example, the Calder′ on-Zygmund type estimate [6], and the separation of variables method [7] for specific cases in high-dimensional elliptic equations, where the hypothetical solution can be expressed by decomposing the multidimensional problem into the form of a product of a series of one-dimensional problems, thus transforming the high-dimensional elliptic equations into a series of one-dimensional ordinary differential equations. By solving these one-dimensional equations, the solution of the original high-dimensional elliptic equation can be obtained.

This paper uses the Feynman-Kac formula to solve elliptic equations. The Feynman-Kac formula named after Richard Feynman and Mark Kac, establishes the connection between partial differential equations and stochastic differential equations. The Feynman-Kac formula is a well-known tool for implementing many partial differential equations (such as diffusion or transport equations) with a stochastic representation of the point solution. It first appeared in the potential theory of the Schrödinger equation, which profoundly reformed quantum mechanics utilizing path integrals. Later, the formula also found applications in mathematical finance, interconnecting the probabilistic and partial differential equation representations of derivatives pricing. The Feynman-Kac formula corresponding to formula (1) is shown in (2).

$$\begin{cases}u(x)=E\left[\int_0^{\tau_D}f(X_t)dt+g(x_\tau)\right]\\ dX_t=\mu(x)dx+\sigma(x)dB_t,\end{cases} \tag{2}$$

Where τ is the time at which the stochastic process exits the boundary D is the time at which the stochastic process exits the boundary, and x_τ is the location of the exit boundary.

2.2 Monte Carlo simulation

Monte Carlo simulation [8] is a mathematical method based on probabilistic statistics to simulate, analyze and predict the behavior of complex systems by performing a large number of random samples under some probability distribution. Monte Carlo methods are widely used in various fields, such as finance, engineering, and statistical physics [9]. The key to the Monte Carlo method is the large random sampling. First, the probability distribution of the problem needs to be determined, and then a large number of samples are drawn from these distributions using a random number generator. Finally, these samples are statistically analyzed to understand the possible outcomes and risks.

In addition, Monte Carlo simulation leads to some errors, and the number of simulations$N$ related to the standard errorE_M satisfies the following equation (3) [11]:

$$E_M=\frac{1}{\sqrt{N}} \tag{3}$$

The advantages of Monte Carlo simulation [12][13] include: (1) Wide applicability. It can be applied to a wide variety of problems, especially those complex problems that are difficult to solve analytically. (2) Easy to implement. Only random sampling and statistical analysis are required to complete it. (3) Reliable results. The accuracy of the results will keep improving as the number of samples increases. However, the Monte Carlo method also has some disadvantages: (1) It is computationally intensive. A large number of random samples are needed, which requires high computational resources and sometimes slow convergence. (2) In some cases, the effect of random sampling is not ideal and more samples may be needed to get accurate results. Despite the limitations, Monte Carlo simulation is still an important tool for solving complex mathematical problems.

3. Constructing global solutions to differential equations using the Feynman-Kac formula

This section provides a methodological discussion with the following equations:

$$\begin{cases} -0.5\Delta u = -0.5\exp(x) & x \in (0,1) \\ u(x) = \exp(x) & x = 0 \text{ or } 1 \end{cases} \tag{4}$$

Its corresponding Feynman-Kac formula is

$$\begin{cases} u(x) = \exp(X_\tau) + \int_0^\tau -0.5\exp(X_s)ds \\ dX_t = dB_t, X_0 = x \end{cases} \tag{5}$$

3.1 Interpolation method

Interpolation is a numerical analysis method used to predict or estimate unknown values between known discrete data points based on those data points. The goal of the interpolation method is to find a suitable function such that the value of the function at known data points matches the actual value so that the value of other unknown points can be inferred. The interpolation method has a wide range of applications in scientific computing, engineering, and data analysis.

In this paper, we take equations (4), (5) as an example to solve the global solution on the boundaries $[a, b]$. The global solution is solved on the boundary as follows.

(1) At the defined boundary $[a, b]$ interval based on Gaussian points are selected M. The points to be found are given as $\{x_1, x_2, \cdots x_M\}$.

(2) An arbitrary point x_m is selected, and according to formula (6), this point is taken as the starting point of Brownian motion to simulate P Brownian motions. Note that the motion position of the point at time t of the path p is $B_t^p$and the time step is set to δ, where R_n is generated by the standard normal distribution N (0,1). B_t^p changes every time step δ, until B_t^p falls outside the motion boundary and stops moving, record the Brownian motion position B_t^p at this time, record the time as the termination time τ, $\{B_{t_1}^p, B_{t_2}^p, \cdots B_{t_n}^p\}$ and all time node$\{t_1^p, t_2^p, t_3^p, \cdots, t_n^p\}$ recorded on trajectory list X^p and movement time T^p in the list.

$$\begin{cases} B_{t_1}^p = x_m \\ B_{t_{n+1}} = B_{t_n} + \sqrt{\delta}R_n \end{cases} \tag{6}$$

(3) The values obtained from step (2) are brought into the Feynman-Kac formula (5), and for an integral part, the trapezoidal product is taken as shown in equation (7), which can be implemented with the trapz command in the numpy library. For the non-integral part directly the x_τ^p is brought in. The two parts are summed to solve for a $\hat{u}^p(x_i)$ value.

$$\int_0^\tau -0.5 \cdot exp(X_s)ds = (-0.5)\frac{\delta\left(\exp(B_{t_1}^p)+\exp(B_{t_2}^p)\right)+\cdots\delta\left(\exp(B_{t_{n-1}}^p)+\exp(B_{t_n}^p)\right)}{2} \tag{7}$$

(4) We get P values of $\hat{u}^p(x_m)$. Taking the expectation gives the numerical solution $\hat{u}(x_m)$ at x_m, as shown in equation (8).

$$\hat{u}(x_m) = \frac{\sum_{p=1}^P \hat{u}^p(x_m)}{P} \tag{8}$$

(5) Iterate through all the points to be solved and repeat the above four steps to obtain M numerical solutions on the sampling points$\{\hat{u}(x_1), \hat{u}(x_2) \cdots \hat{u}(x_M)\}$.

(6) For this M numerical solutions are subjected to Lagrangian interpolation according to the following equation to obtain the global solution $\tilde{u}(x)$ as shown in equation (9).

$$\tilde{u}(x) = \sum_{i=1}^M \hat{u}(x_m) \prod_{j=1, j\neq i}^M \frac{x - x_j}{x_i - x_j} \tag{9}$$

3.2 Polynomial fitting based on simulated annealing algorithm

The conventional method of constructing global solutions of partial differential equations, although easy, has several problems as follows:

(1) Because the Brownian motion has a certain randomness, that is, the points used for interpolation may have a certain gap with the real value, and the interpolation method requires that the fitted curve must pass through the interpolation points, so a large systematic error may arise from the randomness of the motion.

(2) Boundary condition handling: Interpolation methods may encounter difficulties when dealing with complex boundary conditions. For example, when partial differential equations have Robin bounds with nonlinear or nonuniform boundary conditions, interpolation methods may require more sophisticated processing techniques to accurately describe these conditions.

Given the above shortcomings, this paper uses a new method to construct the global solution by replacing the original interpolation method with a polynomial fitting method based on the simulated annealing algorithm, and the specific solution steps are as follows:

(1) On the determined boundary [a, b] interval selected by uniform distribution M. The points to be solved, let them be $\{x_1, x_2, \cdots x_M\}$.

(2) Without loss of generality, a point to be found is chosen $x_m, (m \in \{1,2,\cdots,M\})$and simulate the Brownian motion similar to step (2) of the interpolation method in this paper, but the number of simulations per point P is fixed to 1.

(3) The values obtained from step (2) are brought into the Feynman-Kac formula, and the algorithm is similar to step (2) of the interpolation method, and the trapezoidal product algorithm is adopted for the integral part as shown in Equation (7), and the integral and non-integral parts are added to solve for the $\hat{u}(x_m)$.

(4) Through all the points to be solved, we get M numerical solutions $\{\hat{u}(x_1), \hat{u}(x_2), \cdots, \hat{u}(x_M)\}$.

(5) The polynomial order is set to o , construct the polynomial coefficients by the least squares method, to obtain the polynomial $\hat{u}(x)$as shown in equation (10), the approximate analytical expression of $u(x)$.

$$\hat{u}(x) = a'_o x^o + a'_{o-1} x^{o-1} + \cdots a'_1 x + a'_0 \tag{10}$$

(6) The combination of relative and absolute errors of the fit is used as the energy function E. As shown in Equation (11), the final global solution is obtained by optimizing the polynomial coefficients with the simulated annealing algorithm. The initial temperature of the simulated annealing algorithm is T_0, the annealing rate is α, and the termination temperature is set to T_{end} . The final solution $\tilde{u}(x)$ is obtained in the form of Eq. (12).

$$E = 0.6 * \left|\frac{y_{predict} - y_{true}}{y_{true}}\right| + 0.4 * \left|y_{predict} - y_{true}\right| \tag{11}$$

$$\tilde{u}(x) = a_o x^o + a_{o-1} x^{o-1} + \cdots a_1 x + a_0 \tag{12}$$

4. Numerical experimental results

In this section, numerical experiments are conducted with equations (4), (5), and global solutions are constructed by interpolation and polynomial methods, respectively, with the true solution of the equation as e^x for comparison.

Numerical experiments of the interpolation method are performed first. The points to be found on [0,1] were selected as [0,0.3,0.6,0.9]. The number of simulations for each point. Experiment according to the operation in the interpolation method in 3.1 of this paper. The fitting results are shown in Figure 1.

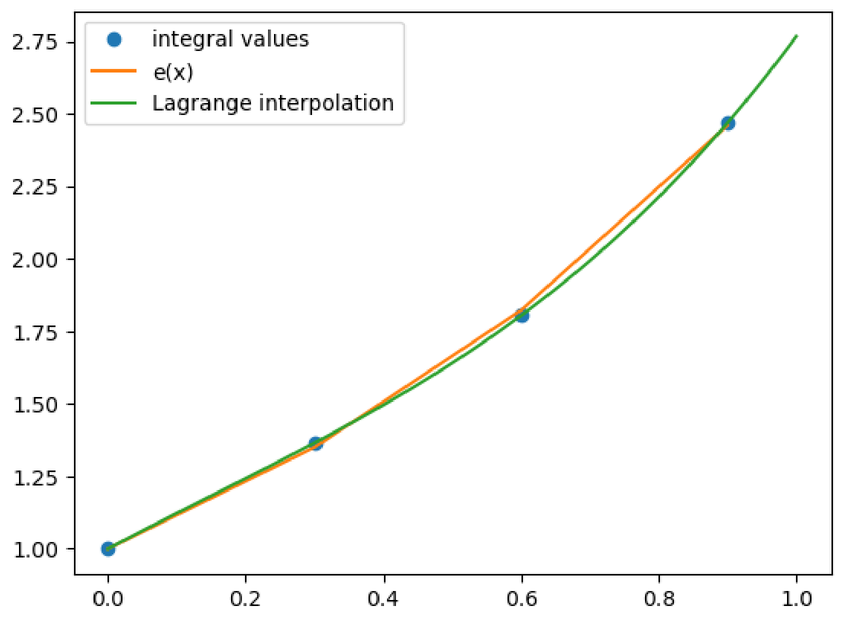

Figure 1: Lagrangian interpolation image

Then the numerical experiment of the improved method is carried out. M points are generated by a uniform distribution. M takes 20000 and the boundary D is set to [0,1]. The polynomial order o is taken as 6, and the parameters of simulated annealing set as $T_0 = 1000$, T_{end} = 1e-5. $\alpha = 0.999$. The global solution equations (13)(14) before and after the optimization of the simulated annealing algorithm are obtained as follows.

$$\hat{u}(x) = 10.042x^6 + -26.239x^5 + 24.391x^4 + -9.455x^3 + 2.139x^2 + 0.844x + 1.018 \quad (13)$$

$$\tilde{u}(x) = 10.044x^6 + -26.223x^5 + 24.389x^4 + -9.478x^3 + 2.132x^2 + 0.853x + 1.009 \quad (14)$$

The images of the fitted global solution compared with the real solution are shown in Figure 2.

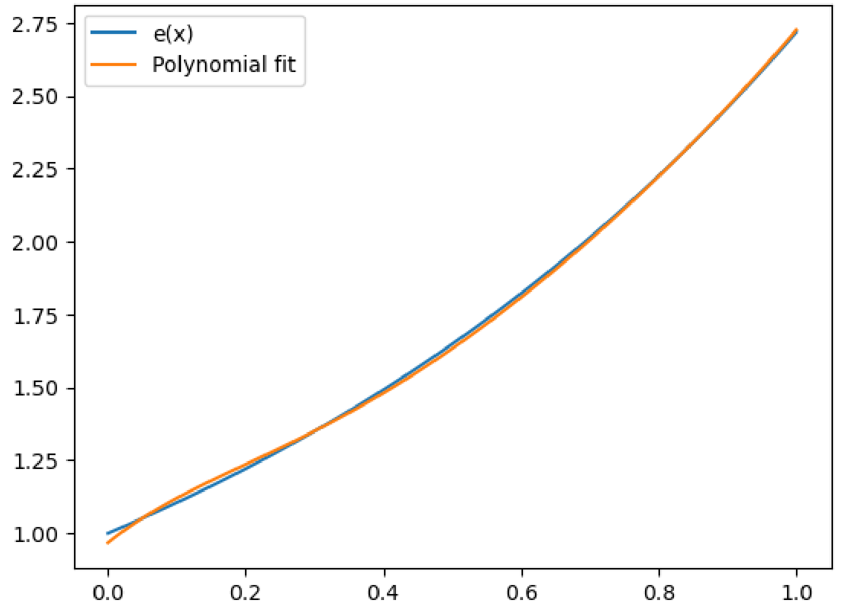

Figure 2: Polynomial fitting image based on simulated annealing algorithm

Finally, the advantages and disadvantages of these two methods are compared. In this paper, the relative error is chosen as the evaluation index to compare the fitting of the traditional method, the polynomial fit by least squares, and the polynomial fit method optimized by the annealing algorithm. The specific steps are to select 100 points uniformly in [0,1] as the test set, calculate the predicted values using the traditional method, the polynomial obtained by least squares and the polynomial optimized by annealing algorithm, respectively, and calculate the relative error $E_{rel1}, E_{rel2}, E_{rel3}$, and the results obtained from the calculation are shown in Table 1.

Table 1: Error comparison

Method	$\boldsymbol{E_{rel}}$
Interpolation method	1.838e-03
Polynomial fit	4.000e-04
Polynomial fitting based on simulated annealing algorithm	4.855e-06

According to the table, it can be seen that $E_{rel1} > E_{rel2} \gg E_{rel3}$. The improved method clearly outperforms the traditional method in the case of the same number of simulated Brownian motions and almost the same arithmetic power consumption.

5. Convergence analysis

Three aspects of this polynomial fitting approach may lead to errors, such as the degree of discretization represented by the time interval between recorded Brownian motions, the number of Monte Carlo simulations, and the order of the polynomial, for which convergence analysis, as well as validation experiments, which are carried out in this section.

5.1 Impact of Discretization

The smaller the time step, the higher the accuracy of the numerical computation, but the longer the time required for the computation will be. When the time step is smaller, the calculation results will be closer to the real solution because the simulation is closer to the real physical process at a smaller time step. However, when the time step is too small, the errors in the numerical computation also become more sensitive because the errors can accumulate over more time steps and thus affect the accuracy of the computation. On the other hand, when the time step is larger, the time required for the calculation will be shorter, but the accuracy of the numerical calculation will be reduced. When the time step is too large, problems such as numerical instability and divergence may occur, which can lead to inaccurate calculation results. Therefore, it is necessary to balance the computational accuracy and computational efficiency when choosing the time step to obtain optimal numerical calculation results. In conclusion, the time step is an important parameter in numerical computation, which determines the accuracy and computational efficiency of numerical computation. The selection of the appropriate time step requires comprehensive consideration of simulation accuracy, computational efficiency, computational stability, and other factors, and adjustment according to the specific problem to obtain the optimal numerical calculation results.

To investigate the effect of the degree of discretization on the accuracy of the results, the number of simulated Brownian motions is fixed as M = 20000 and the polynomial order o is fixed to 6, and the transform the time step δ. To carry out the above numerical simulation experiments, the transverse coordinates are transformed as follows to make the data more intuitive.

$$x = -\lg(\delta) \tag{15}$$

The obtained results are shown in Fig. 3, where it can be seen that the error shows a tendency to converge to 0 as the time step is shortened.

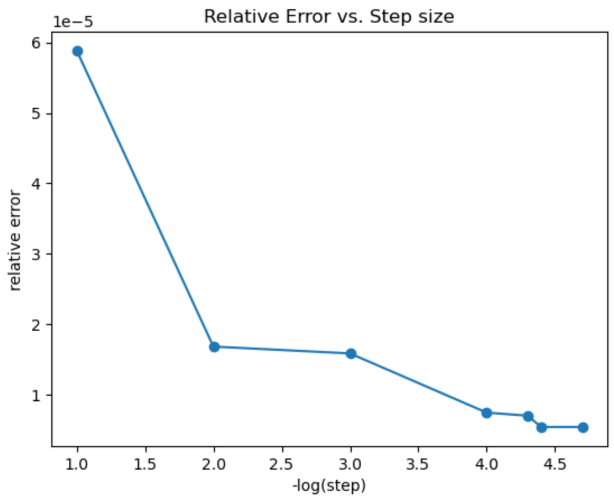

Figure 3: Relative error vs. Step size

5.2 Effect of the Number of Monte Carlo Simulations

Monte Carlo simulation is a random number-based simulation method that estimates the solution of a problem by generating a large number of random samples. As the number of simulations increases, the error of Monte Carlo simulation gradually decreases because more samples can provide more accurate statistical information, thus improving the accuracy of the estimates. The error of a Monte Carlo simulation can usually be measured by the standard error, which is the standard deviation of the estimate divided by the square root of the number of samples. As the number of simulations increases, the standard error decreases and approaches zero. This indicates that the more the number of simulations, the smaller the error of the Monte Carlo simulation. On the other hand, the error of Monte Carlo simulation does not decrease linearly but shows a decreasing trend. At the beginning of increasing the number of simulations, error decreases rapidly as the number of simulations increases, but as the number of simulations increases, the error decreases gradually and slows down until it stabilizes. Therefore, when choosing the number of simulations, it is necessary to balance the computational accuracy and computational efficiency to obtain optimal simulation results. In conclusion, there is a decreasing relationship between the error of Monte Carlo simulation and the number of simulations, and the error decreases gradually and tends to be stable with the increase of the number of simulations. Choosing a suitable number of simulations can obtain more accurate simulation results, but factors such as computational efficiency need to be considered.

To investigate the effect of the number of Monte Carlo simulations on the accuracy of the results, varying the number of simulations N is taken as $\{100, 1000, 10000, 25000, 50000, 100000, 150000\}$ fixed polynomial order 6, fixed time step δ as 0.001, the sub numerical experiments were conducted, and the experimental results are shown in Figure 4, where it can be seen that the error shows a tendency to converge to 0 as the number of simulations increases.

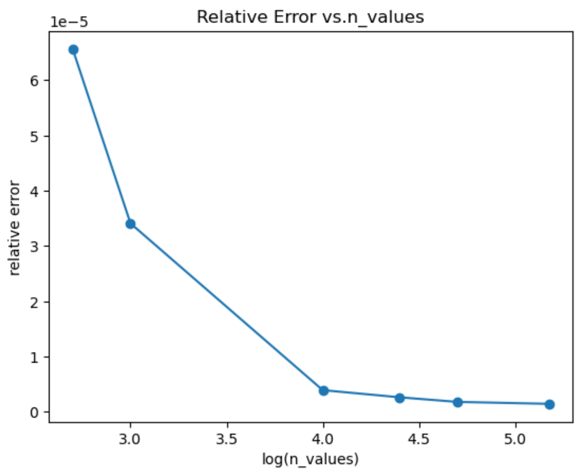

Figure 4: Relative error vs. number of simulations

5.3 Polynomial order effect

In this subsection, we explore the relationship between polynomial order and polynomial fitting accuracy. Polynomial fitting is a widely used mathematical method for data analysis, modeling, and prediction. By choosing the right polynomial order, we can achieve high accuracy in the fitting process. However, the appropriate choice of polynomial order is crucial to avoid overfitting and underfitting.

The basic principle of the polynomial fitting is to use an o polynomial of order to approximate the objective function of the form, As shown in formula (16).

$$P(x) = a_0 + a_1x + a_2x^2 + \dots + a_ox^o \tag{16}$$

where $a_0, a_1, \dots, a_o$ are the polynomial coefficients, and x denotes the input variable, and $P(x)$ denotes the output variable, and o denotes the order of the polynomial. The goal of the fitting process is to find the best coefficient value that minimizes the error of the polynomial at a given data point.

There is a complex relationship between polynomial order and fitting accuracy. In general, increasing the order of a polynomial can improve the fitting accuracy. This is because polynomials of higher order have more degrees of freedom and can capture more complex data features. However, when the order is too high, it may lead to an overfitting phenomenon. Overfitting means that the polynomial performs well on training data, but has poor predictive performance on new, unseen data. This is because higher-order polynomials may focus too much on the noise in the training data and fail to capture the true patterns in the data. On the other hand, if the polynomial order is too low, it may lead to an underfitting phenomenon. Underfitting refers to the inability of the polynomial to capture the key features in the data, which leads to poor prediction performance. To avoid overfitting and underfitting, a balance between model complexity and fitting accuracy is needed.

To investigate the effect of polynomial order on the results, varying the polynomial order is shown in the horizontal coordinate of Fig. 5, fixed as the number of simulations M is 20000, the time step of Brownian motion δ = 0.001. The numerical experiments were conducted, and the convergence of the experiments is shown in Fig. 5, where it can be seen that the error shows a tendency to converge to 0 as the polynomial order increases.

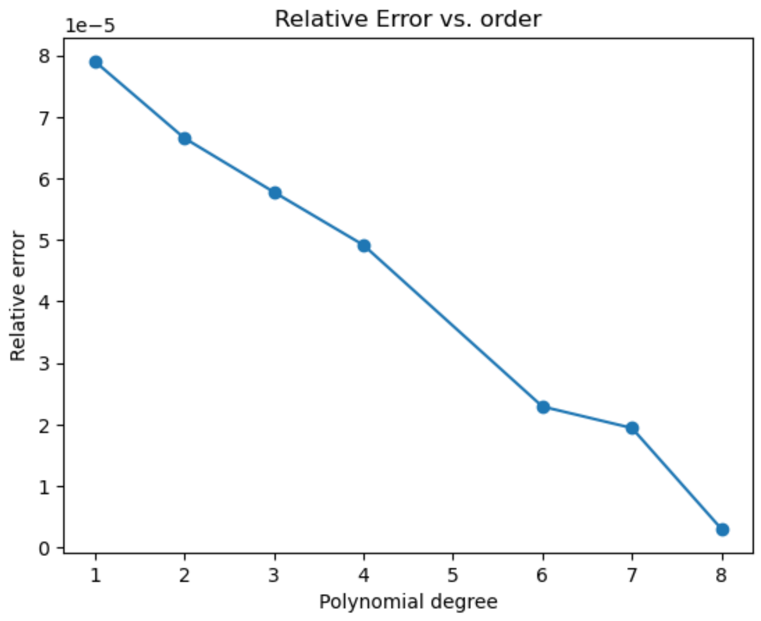

Figure 5: Relative error vs. polynomial order

6. Conclusion

This paper introduces the importance of partial differential equations in scientific and engineering problems and the difficulties encountered by traditional numerical methods such as the finite element method and the difference method in dealing with the problems, as well as the problems encountered by general interpolation methods in constructing global solutions. In this paper, we propose a method for constructing global solutions of partial differential equations by solving numerical solutions with the Feynman-Kac formula and replacing the interpolation method with a polynomial fitting method based on the annealing algorithm, which constructs global solutions with higher accuracy than the traditional methods. The relative error of the constructed global solution is much smaller than that of the traditional method. The subsequent numerical experiments of convergence analysis show that with the decrease of the time step, the number of simulations increases, the order of polynomials increases, and the error gradually approaches 0. On the other hand, because of the use of the Feynman-Kac formula, it can be extended to a certain extent to solve in higher dimensions.

References

[1] Morton K W. Revival: Numerical solution of convection-diffusion problems (1996) [J]. 2019.
[2] Reddy J N. Introduction to the finite element method [J]. 2019.
[3] Ma C, Ma Q, Yao H, et al. An accurate European option pricing model under Fractional Stable Process based on Feynman Path Integral [J]. Physica A: Statistical Mechanics and its Applications, 2018, 494: 87-117.
[4] Ali H, Kamrujjaman M D. Numerical solutions of nonlinear parabolic equations with Robin condition: Galerkin approach [J]. 2022.
[5] Ding C, Yan C, Zeng X, et al. A Parallel Iterative Probabilistic Method for Mixed Problems of ellipse Equations with the Feynman—Kac Formula of Killed Brownian Motions[J]. SIAM Journal on Scientific Computing, 2022, 44(5): A3413-A3435.
[6] Balci A K, Diening L, Weimar M. Higher order Calderón-Zygmund estimates for the p-ellipse equation [J]. Journal of Differential Equations, 2020, 268(2): 590-635.
[7] Polyanin A D, Zhurov A I. Separation of variables in PDEs using nonlinear transformations: Applications to reaction–diffusion type equations [J]. Applied Mathematics Letters, 2020, 100: 106055.
[8] Arend M G, Schäfer T. Statistical power in two-level models: A tutorial based on Monte Carlo simulation [J]. Psychological Methods, 2019, 24(1): 1.
[9] Landau D, Binder K. A guide to Monte Carlo simulations in statistical physics[M]. Cambridge university press, 2021.
[10] Nisar K, Sabir Z, Asif Zahoor Raja M, et al. Artificial neural networks to solve the singular model with neumann–robin, dirichlet and neumann boundary conditions[J]. Sensors, 2021, 21(19): 6498.
[11] Maire S, Tanré E. Some new simulations schemes for the evaluation of Feynman–Kac representations [J]. 2008.
[12] Harrison R L. Introduction to monte carlo simulation[C]//AIP conference proceedings. American Institute of Physics, 2010, 1204(1): 17-21.
[13] Landau D, Binder K. A guide to Monte Carlo simulations in statistical physics[M]. Cambridge university press, 2021.