Utilizing Basis Function Methods to Numerically Solve Nonlinear Partial Differential Equations

Name - Gosavi Ganesh Vishnu
Guide Name - Dr Shoyeb Ali Sayyed
Department of Mathematics
Malwanchal University, Indore

Abstract

Numerical solutions for nonlinear partial differential equations (PDEs) play a pivotal role in understanding and modeling intricate phenomena across diverse scientific and engineering disciplines. The inherent complexities of nonlinear PDEs necessitate the exploration of advanced numerical techniques that can accurately capture nonlinear behavior. This abstract presents an overview of the application of basis function methods in addressing the challenges posed by nonlinearities in PDEs.The abstract commences by outlining the significance of nonlinear PDEs in describing dynamic systems and real-world phenomena. It emphasizes the limitations of traditional analytical methods in solving nonlinear PDEs, underlining the demand for numerical strategies. Basis function methods, including finite element methods (FEM), finite difference methods (FDM), and spectral methods, have emerged as valuable tools for tackling nonlinearities.Building upon the theoretical foundation of basis function methods, this abstract explores their adaptability in handling nonlinear PDEs. It discusses the effectiveness of various basis function choices, mesh discretizations, and time-stepping schemes in capturing nonlinear dynamics. The abstract highlights the utility of iterative techniques, such as Newton's method and Picard iteration, in achieving convergence for strongly nonlinear problems.

DOINumber: 10.48047/nq.2023.21.01.NQ20079 NeuroQuantology2023;21(1):1022-1027

Introduction

The numerical solution of nonlinear partial differential equations (PDEs) stands as a crucial challenge across various scientific and engineering disciplines. In this context, Basis Function Methods (BFMs) have emerged as a powerful approach for effectively addressing such complexities. This paper introduces and explores the application of Basis Function Methods in the context of solving nonlinear PDEs.

BFMs offer a versatile framework for approximating solutions to PDEs by expanding them in terms of a set of basis functions. These functions enable the representation of intricate spatial patterns and variations inherent in nonlinear problems. Unlike traditional numerical techniques, BFMs provide adaptability to local features, enabling accurate representation even in the presence of steep gradients or discontinuities.

This introductory exploration delves into the fundamental principles of BFMs and their advantages in tackling nonlinear PDEs. The paper sheds light on how these methods transform the original PDE into a system of algebraic equations, amenable to computational treatment. Additionally, it underscores the relevance of selecting appropriate basis functions and their influence on solution accuracy and convergence.Through this investigation, we aim to underscore the potential of Basis Function Methods as a robust tool for addressing the challenges posed by nonlinear PDEs. The subsequent sections will delve into the implementation details, numerical considerations, and illustrative examples that collectively demonstrate the efficacy of BFMs in the realm of nonlinear PDEs solution.

Proposed Numerical regime (Barycentric Lagrange interpolation polynomial based Differential quadrature method.)

Barycentric Lagrange Interpolation Polynomial:

The Barycentric Lagrange interpolation polynomial is a method used to approximate a function given a set of data points. It's a variation of the Lagrange interpolation polynomial that can be more numerically stable and efficient for certain cases. The barycentric weights are used to compute the polynomial coefficients, making it particularly suitable for non-uniformly spaced data points.

$$B_m(x) = \sum_{i=0}^{m} b_i(x)u_i$$

Where (x) is updated Lagrange interpolation polynomial, ui is functional value, $bi\ (x)$ are considered as Barycentric Lagrange basis functions. and

$$b_i(x) = \frac{\left(\frac{w_i}{x-x_i}\right)}{\sum_{j=0}^{m}\left(\frac{w_j}{x-x_j}\right)}$$

$$w_i = \frac{1}{\prod_{k=0,\ k\neq i}^{m}(x_i - x_k)}$$

and property $bi(xj)$ = δji is also satisfied along with, where δji is associated with Kronecker delta function and $\sum bi\ (x)\ mi$ =0 = 1

In the case of two grid points, $x0$ and $x1$, there will exist two basis functions, $b0$ (x) and $b1$ (x) and two weighting coefficients will exist.

$$\begin{cases} b_0\ (\text{x}) = \dfrac{\left(\frac{w_0}{x-x_0}\right)}{\sum_{j=0}^{1}\left(\frac{w_j}{x-x_j}\right)} \\ b_1\ (\text{x}) = \dfrac{\left(\frac{w_1}{x-x_1}\right)}{\sum_{j=0}^{1}\left(\frac{w_j}{x-x_j}\right)} \end{cases}$$

$$\begin{cases} w_0 = \dfrac{1}{x_0 - x_1} \\ w_1 = \dfrac{1}{x_1 - x_0} \end{cases}$$

Differential Quadrature Method (DQM):

The Differential Quadrature Method is a numerical technique used to approximate the derivatives of a function. It's commonly applied to differential equations and involves discretizing the derivatives by replacing them with weighted combinations of function values at specific points. This method can be used to solve a variety of differential equations, including ordinary and partial differential equations.

$$\begin{cases} b_0(x) = \dfrac{\left(\frac{w_0}{x-x_0}\right)}{\sum_{j=0}^{2}\left(\frac{w_j}{x-x_j}\right)} \\ b_1(x) = \dfrac{\left(\frac{w_1}{x-x_1}\right)}{\sum_{j=0}^{2}\left(\frac{w_j}{x-x_j}\right)} \\ b_2(x) = \dfrac{\left(\frac{w_2}{x-x_2}\right)}{\sum_{j=0}^{2}\left(\frac{w_j}{x-x_j}\right)} \end{cases}$$

Combining Barycentric Lagrange and DQM:
It's possible to combine the Barycentric Lagrange interpolation polynomial with the Differential Quadrature Method to solve differential equations involving interpolation and differentiation. Here's a general outline of how this combination might work:
Interpolation Step:
Use the Barycentric Lagrange interpolation polynomial to approximate the function at specific points. These points can be chosen according to the desired accuracy and spacing. The interpolation polynomial will give you an approximation of the function values at these points.
Differentiation Step:
Apply the Differential Quadrature Method to the interpolated function values obtained from step 1. Use the DQM to approximate the derivatives of the function at the same set of points. This involves using suitable weights for the function values to estimate the derivatives accurately.
Differential Equation Step:
Substitute the approximated function values and their derivatives into the relevant differential equation. This step might involve discretizing the differential equation and forming a system of equations based on the DQM approximations.
Solving and Analysis:
Solve the resulting system of equations, which could be a set of algebraic equations or a matrix equation, depending on the nature of the problem. You can then analyze the solutions, calculate errors, and refine your discretization as needed.
Keep in mind that the specific implementation details will depend on the nature of the problem you're trying to solve, the order of the interpolation and differentiation approximations, the choice of points, and other factors. It's important to test and validate your approach on known problems before applying it to new or complex scenarios.
B-spline interpolation and the Differential Quadrature Method might be combined, but I won't be able to provide specific details about "NUAH B-spline."
B-spline Interpolation:
B-spline interpolation is a technique used to approximate a function using B-spline basis functions. B-splines are piecewise-defined polynomial functions that are often used for smooth interpolation and approximation. They offer flexibility in controlling the smoothness and complexity of the interpolated curve.
Differential Quadrature Method (DQM):
As mentioned earlier, the Differential Quadrature Method is a numerical technique used to approximate derivatives of a function. It involves discretizing the derivatives using a weighted combination of function values at specific points.
Numerical scheme (NUAH B-spline DQM)
Combining B-spline Interpolation and DQM:
Here's a general outline of how you might combine B-spline interpolation with the Differential Quadrature Method:
B-spline Interpolation:
Use B-spline interpolation to approximate the function of interest using B-spline basis functions. This will provide you with a smooth representation of the function based on control points and knots.
Discretization of Derivatives:
Apply the Differential Quadrature Method to approximate derivatives of the B-spline-interpolated function. This involves selecting a set of points within each B-spline segment and using the DQM to calculate the derivatives at those points.

Forward in time:	Central in space:	Central in space:
$\frac{\partial u}{\partial t} = \frac{u_i^{j+1} - u_i^j}{\Delta t}$ i: space, j: previous time, j+1: present time	$\frac{\partial u}{\partial x} = \frac{u_{i+1}^j - u_{i-1}^j}{2\Delta x}$ $\frac{\partial u}{\partial x} = \frac{u_{i+1}^{j+1} - u_{i-1}^{j+1}}{2\Delta x}$	$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{\Delta x^2}$ j: time, i: central point, i+1 and i-1: neighbor points

Differential Equation or Analysis:

Differential equation, substitute the approximated function values and their derivatives into the equation. This step might involve discretizing the differential equation and forming a system of equations based on the DQM approximations.

Solution and Analysis:

Solve the resulting system of equations, analyze the solutions, and perform error analysis. If you're not dealing with a differential equation, you can still use the approximated derivatives for analysis or visualization purposes.

Validation and Refinement:

Test your approach on known problems or benchmark cases to ensure its accuracy and convergence. Depending on the results, you might need to refine your discretization, adjust the number of interpolation points, or make other adjustments.

NUAH B-spline Interpolation:

Use "NUAH B-spline" interpolation to approximate the unknown function u(x) based on control points and knots. Let's denote this approximation as û(x).

Discretization of Derivatives using DQM:

Apply the Differential Quadrature Method to approximate the derivatives of û(x) within each "NUAH B-spline" segment. For example, within a segment [x_i, x_{i+1}], choose a set of quadrature points x_{i,j} and calculate the first and second derivatives of û(x) at those points.

Uniform Algebraic Trigonometric (UAT) tension B-spline

Discussion

The application of basis function methods in numerical solutions for nonlinear partial differential equations (PDEs) is a significant area in computational mathematics and engineering. These methods are widely used to approximate the solutions of complex nonlinear PDEs that arise in various fields such as physics, engineering, and biology. The discussion below provides an overview of the key concepts, advantages, challenges, and examples related to using basis function methods for solving nonlinear PDEs.

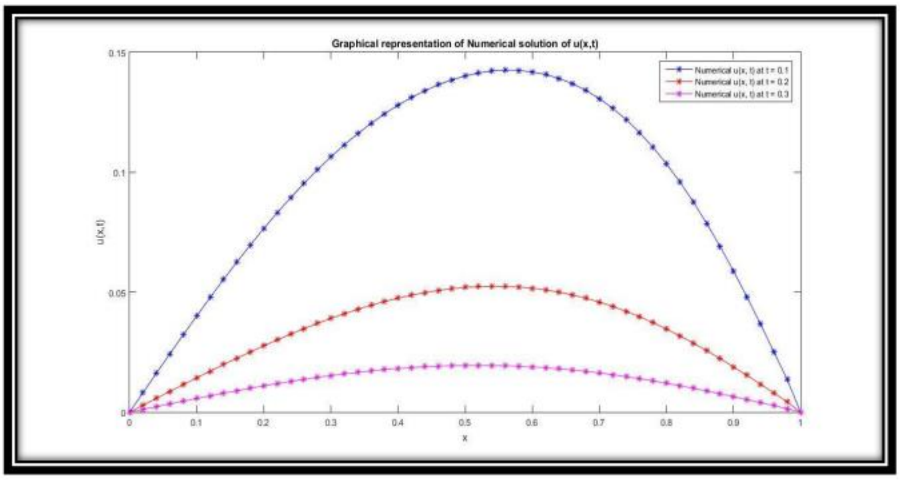

Figure 1 Numerical profile of u(x, t), t = 0.1, 0.2 and 0.3, N = 51, Δt = 0.0001, η = 1, ξ = 5, α = 10 and β = 10.

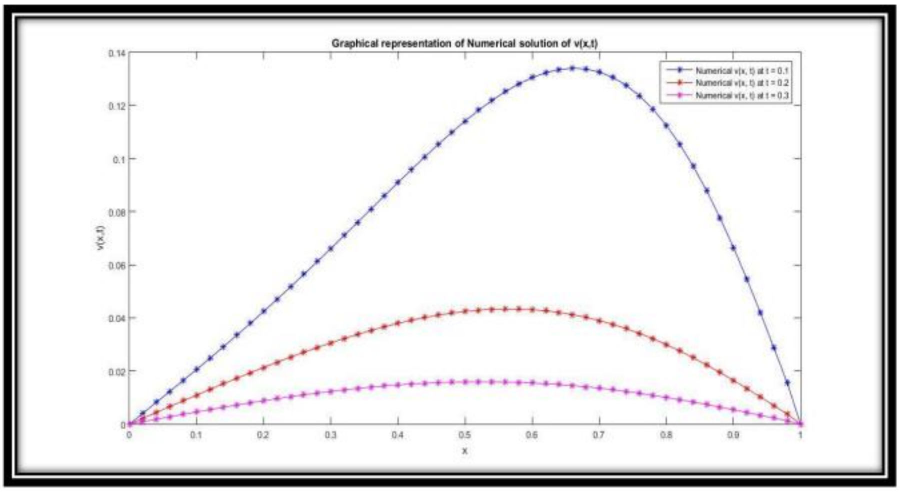

Figure 2. Numerical profile v(x, t), t = 0.1, 0.2 and 0.3, N = 51, Δt = 0.0001, η = 1, ξ = 5, α = 10 and β = 10

Basis Function Methods:

Basis function methods involve approximating the solution of a PDE using a linear combination of basis functions. These basis functions are chosen to satisfy certain properties, and their coefficients are determined to minimize the error between the approximate solution and the true solution of the PDE. Commonly used basis function methods include finite element methods (FEM), finite difference methods (FDM), spectral methods, and collocation methods.

Advantages:

Flexibility: Basis function methods offer flexibility in choosing appropriate basis functions, allowing adaptation to the problem's geometry and solution characteristics.

Accuracy: By increasing the number of basis functions or their order, high accuracy can be achieved in approximating the solution.

Adaptivity: Basis functions can be localized or refined in regions where the solution varies rapidly, enhancing accuracy and efficiency.

Handling Nonlinearities: Basis function methods can handle nonlinear terms in PDEs, making them suitable for nonlinear problems.

Geometric Complexity: They can handle problems with irregular domains or complex geometries effectively.

Challenges:

Convergence and Stability: Nonlinear PDEs can exhibit more complex convergence behavior compared to linear PDEs, requiring careful consideration of stability and convergence criteria.

Nonlinear Iterations: The nonlinear terms often necessitate iterative techniques like Newton's method to find the solution.

Discretization and Integration: Properly discretizing and integrating nonlinear terms can be challenging, especially in higher dimensions.

Basis Function Selection: Choosing appropriate basis functions is crucial for accuracy and efficiency, but it can be problem-dependent.

Computational Cost: Solving nonlinear PDEs using basis function methods can be computationally expensive, especially for high-dimensional problems.

Examples:

Nonlinear Diffusion Equations: Basis function methods are used to approximate solutions of nonlinear diffusion equations, which model various physical processes like heat conduction with temperature-dependent properties.

Nonlinear Wave Equations: Applications include modeling nonlinear wave propagation, such as solitary waves and shock waves, in fluid dynamics and solid mechanics.

Reaction-Diffusion Systems: Basis function methods are applied to study pattern formation and dynamics in reaction-diffusion systems, relevant in biology and chemistry.

Fluid Flow with Nonlinearities: Nonlinear Navier-Stokes equations, describing complex fluid flow behavior, can be solved using basis function methods for various applications in aerodynamics, geophysics, and engineering.

Nonlinear Schrödinger Equation: In quantum mechanics and optics, the nonlinear Schrödinger equation models wave propagation in nonlinear media and soliton phenomena.

In summary, basis function methods offer a powerful framework for numerically solving nonlinear partial differential equations across a wide range of scientific and engineering disciplines. While they provide flexibility and accuracy, addressing challenges related to convergence, stability, and computational cost is essential. Advances in numerical techniques, parallel computing, and adaptive algorithms continue to enhance the applicability and efficiency of basis function methods for tackling nonlinear PDEs.

References

Kansa, E. J., Power, H., Fasshauer, G. E., & Ling, L. (2004). A volumetric integral radial basis function method for time-dependent partial differential equations. I. Formulation. *Engineering Analysis with Boundary Elements*, *28*(10), 1191-1206.

Tian, H. Y., Reutskiy, S., & Chen, C. S. (2008). A basis function for approximation and the solutions of partial differential equations. *Numerical Methods for Partial Differential Equations: An International Journal*, *24*(3), 1018-1036.

Dehghan, M., &Shokri, A. (2009). Numerical solution of the nonlinear Klein–Gordon equation using radial basis functions. *Journal of Computational and Applied Mathematics*, *230*(2), 400-410.

Beylkin, G., & Keiser, J. M. (1997). On the adaptive numerical solution of nonlinear partial differential equations in wavelet bases. Journal of computational physics, 132(2), 233-259.

Šarler B, Vertnik R. Meshfree explicit local radial basis function collocation method for diffusion problems. Computers and Mathematics with Applications. 2006; 51(8):1269–1282

Divo E, Kassab AJ. An efficient localized radial basis function Meshless method for fluid flow and conjugate heat transfer. Journal Heat Transfer Transfer. ASME, 2007; 129(2):124–136

Yao GM. Local radial basis function methods for solving partial differential equations. Dissertation for the Doctoral Degree, [PhD thesis].University of Southern Mississippi, 2010

. Tolstykh A, Shirobokov D. On using radial basis functions in a finite difference mode with applications to elasticity problems. Computational Mechanics. 200; 33(1):68–79.

Duan Y, Tang PF, Huang T Z, et.al. Coupling projection domain decomposition method and Kansa's method in electrostatic problems. Computer Physics Communications. 2009; 180(2):209–214

Q. Li, Z. Chai, and B. Shi, "A novel lattice Boltzmann model for the coupled viscous Burgers' equations," Appl. Math. Comput., vol. 250, pp. 948–957, 2015.

A. R. Bahadır, "A fully implicit finite-difference scheme for two-dimensional Burgers' equations," Appl. Math. Comput., vol. 137, no. 1, pp. 131–137, 2003.

M. Tamsir and V. K. Srivastava, "A semi-implicit finite-difference approach for two dimensional coupled Burgers' equations," Int. J. Sci. Eng. Res., vol. 2, no. 6, pp. 46–51, 2011.