



Research article**A distanced-network-based numerical approximation method for second-order differential equations****Ray-Ming Chen**^{1,2,*}¹ Department of Mathematical Sciences, College of Science, Mathematics and Technology, Wenzhou-Kean University, Wenzhou 325060, China² Department of Mathematical Sciences, College of Science, Mathematics and Technology, Kean University, 1000 Morris Avenue, Union, New Jersey 07083, USA*** Correspondence:** Email: crayming@wku.edu.cn, crayming@kean.edu.

Abstract: This research incorporates a distanced network into the calculus of variations to yield a distanced-network-based method (DNBM), which approximates and interpolates the solutions, or $\hat{y}(x)$, for a boundary value problem (BVP) of second-order differential equations. Given an input vector X and boundary conditions $y(x_0)$ and $y(x_{N+1})$, we identify the distance matrix D for X and choose a function ρ to reveal the relationship between the inputs, or D_ρ . We perform calculus of variations to find the internal relationships between the variables and parameters. We also demonstrate how to apply the theory when D_ρ is singular by adding one weighted matrix to avert the singularity. A final experimental comparison between our DNBM and the Ritz finite-element method is also conducted. Our network method should provide another comprehensive and insightful approximation.

Keywords: metricized interpolation; calculus of variation; BVP; distanced network**Mathematics Subject Classification:** 65L10, 65L60, 46B70

1. Introduction

In comparison with the analytic approach, numerical methods for Ordinary Differential Equations (ODEs) [1] or Partial Differential Equations (PDEs) [2] have much more practical merit in application. Some methods are simply based on Euler's method [3] or Taylor series [4] or the finite-volume method [5], and several applications are related to Runge–Kutta methods [6, 7] and numerous variants [8]. Some interesting methods, such as collocation methods [9, 10] and finite-difference methods [11], are also frequently used in numerical approximation. Much more sophisticated methods involving variational calculus [12, 13] are also widely used in solving differential equations, such as the Ritz finite-element method [14]. In this research, we devise and

incorporate the concept of a distance function into the framework of variational calculus to largely enhance the practical applicability of calculus of variations. Let $M = (\mathbb{R}, d)$ denote a metric space, where $\mathbb{R}$ is the set of all the real numbers and d is a metric on $\mathbb{R}$. Let $X \equiv \{\alpha \equiv x_0, x_1, x_2, \dots, x_{N-1}, x_N, x_{N+1} \equiv \beta\} \subseteq \mathbb{R}$, with each $x_i < x_{i+1}$, and let $D \equiv [d(x_i, x_j)]_{i,j=0}^{N+1}$ be arbitrary such that its determinant $|D| \neq 0$. We propose a numerical approach based on distance functions to numerically approximate $y(x)$, the solutions of the following boundary value problem (BVP) for the typical second-order differential equations [15, 16]:

$$c_1(x) \cdot y''(x) + c_2(x) \cdot y'(x) + c_3(x) \cdot y(x) = c_4(x), \quad (1.1)$$

$$y(\alpha) = a, \quad (1.2)$$

$$y(\beta) = b, \quad (1.3)$$

where $c_1(x) \neq 0$ for all $x \in [\alpha, \beta]$. Let ρ denote an arbitrary twice differentiable function from $\mathbb{R}_0^+$, the set of all the non-negative real numbers, to $\mathbb{R}$. Let D_ρ denote the $(N+2)$ -by- $(N+2)$ matrix $[\rho(d(x_i, x_j))]_{i,j=0}^{N+1}$. Define $\rho_d^i : \mathbb{R} \rightarrow \mathbb{R}$ by $\rho_d^i(x) := \rho(d(x, x_i))$. For any $x \in [\alpha, \beta]$, let x_X denote a $(N+2)$ -by-1 distance-induced matrix $[\rho_d^i(x)]_{i=0}^{N+1} = [\rho(d(x, x_i))]_{i=0}^{N+1}$. Let x_{D_ρ} denote its $(N+2)$ -by-1 coordinate matrix represented by D_ρ , that is $x_X = [\rho(d(x, x_i))]_{i=0}^N = D_\rho x_{D_\rho}$, that is, $x_{D_\rho} = D_\rho^{-1} x_X = D_\rho^{-1} [\rho(d(x, x_i))]_{i=0}^{N+1} \equiv$

$$\begin{bmatrix} \delta_{0,0} & \delta_{0,1} & \cdots & \delta_{0,N+1} \\ \delta_{1,0} & \delta_{1,1} & \cdots & \delta_{1,N+1} \\ \vdots & \vdots & \vdots & \vdots \\ \delta_{N,0} & \delta_{N,1} & \cdots & \delta_{N,N+1} \\ \delta_{N+1,0} & \delta_{N+1,1} & \cdots & \delta_{N+1,N+1} \end{bmatrix} \begin{bmatrix} \rho \circ d(x, x_0) \\ \rho \circ d(x, x_1) \\ \vdots \\ \rho \circ d(x, x_N) \\ \rho \circ d(x, x_{N+1}) \end{bmatrix} = \begin{bmatrix} \sum_{j=0}^{N+1} \delta_{0,j} \cdot \rho \circ d(x, x_j) \\ \sum_{j=0}^{N+1} \delta_{1,j} \cdot \rho \circ d(x, x_j) \\ \vdots \\ \sum_{j=0}^{N+1} \delta_{N,j} \cdot \rho \circ d(x, x_j) \\ \sum_{j=0}^{N+1} \delta_{N+1,j} \cdot \rho \circ d(x, x_j) \end{bmatrix}, \quad (1.4)$$

where

$$\begin{bmatrix} \sum_{j=0}^{N+1} \delta_{0,j} \cdot \rho \circ d(x, x_j) \\ \sum_{j=0}^{N+1} \delta_{1,j} \cdot \rho \circ d(x, x_j) \\ \vdots \\ \sum_{j=0}^{N+1} \delta_{N,j} \cdot \rho \circ d(x, x_j) \\ \sum_{j=0}^{N+1} \delta_{N+1,j} \cdot \rho \circ d(x, x_j) \end{bmatrix} \equiv \begin{bmatrix} \alpha_0(x) \\ \alpha_1(x) \\ \vdots \\ \alpha_N(x) \\ \alpha_{N+1}(x) \end{bmatrix}, \text{ and } D_\rho^{-1} = [\delta_{i,j}]_{i,j=0}^{N+1}.$$

The rationale for setting a function ρ is to further regulate distance functions to fit real applications. Let $\hat{y}(x)$ denote an approximation function of $y(x)$. Let $\hat{Y}_X$ denote the $(N+2)$ -by-1 approximation matrix $[\hat{y}_i]_{i=0}^{N+1}$ of $[y_i]_{i=0}^{N+1}$, where y_i denotes $y(x_i)$, nodal value $\hat{y}_i$ denotes $\hat{y}(x_i)$ and $\hat{y}_0 \equiv y_0 = a$; $\hat{y}_{N+1} \equiv y_{N+1} = b$; and $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N$ are the parameters, or in a vector form $\hat{Y} = [\hat{y}_i]_{i=1}^N$, to be estimated by our proposed approach. Set

$$\hat{y}(x) \equiv \hat{y}(x; \hat{Y}_X) := \hat{Y}_X \bullet x_{D_\rho} = \sum_{i=0}^{N+1} \hat{y}(x_i) \cdot \alpha_i(x),$$

where $\bullet$ is the Euclidean inner product. The (numerical) approximation version of Eq (1.1) becomes

$$c_1(x) \cdot \sum_{j=0}^{N+1} \hat{y}(x_j) \cdot \alpha_j''(x) + c_2(x) \cdot \sum_{i=0}^{N+1} \hat{y}(x_i) \cdot \alpha_i'(x) + c_3(x) \sum_{i=0}^{N+1} \hat{y}(x_i) \cdot \alpha_i(x) \approx c_4(x),$$

where

$$\begin{aligned} \alpha_i(x) &= \sum_{j=0}^{N+1} \delta_{ij} \cdot \rho(d(x; x_j)); \quad \alpha_i'(x) = \sum_{j=0}^{N+1} \delta_{ij} \cdot \rho'(d(x; x_j)) \cdot d'(x; x_j), \\ \alpha_i''(x) &= \sum_{s=0}^{N+1} \delta_{is} \cdot [\rho''(d(x; x_s)) \cdot [d'(x; x_s)]^2 + \rho'(d(x; x_s)) \cdot d''(x; x_s)]. \end{aligned}$$

Let us detail the theories and the complete procedures in the next section.

2. Theories and derivations

In this section, we develop the mathematical foundations of the distance-network-based method (DNBM). We begin by defining the interpolation framework using distance matrices, then derive the Euler–Lagrange equations that govern the variational formulation, and finally establish the linear system that determines the unknown nodal values. If M is a square matrix, we use $|M|$ to denote its determinant. Let $[\alpha, \beta] \rightarrow \mathbb{R}$ be the set of all twice differentiable real functions whose domains are $[\alpha, \beta]$. Recall that $\hat{Y}_X = [\hat{y}_0 = a, \hat{y}_1, \dots, \hat{y}_N, \hat{y}_{N+1} = b]$ and $\hat{Y} = [\hat{y}_1, \dots, \hat{y}_N]$, where $\hat{y}_0 = y(x_0) = y(\alpha) = a$ and $\hat{y}_{N+1} = y(x_{N+1}) = y(\beta) = b$ are the boundary conditions. Assume the inner product

$$\langle f, g \rangle_\mu = \int_\alpha^\beta e^{\int_{c_1(x)}^{c_2(x)} dx} f(x) \cdot g(x) dx$$

for any pair of real functions f and g whose domains are $[\alpha, \beta]$, where $c_1(x)$ and $c_2(x)$ are the coefficients defined in Eq (1.1) for the BVP.

Definition 2.1. (Interpolation) Define $\hat{y} : [\alpha, \beta] \rightarrow \mathbb{R}$ by

$$\hat{y}(x; \hat{Y}_X) := \hat{Y}_X \bullet x_{D_\rho} = \sum_{i=0}^{N+1} \hat{y}_i \cdot \left[\sum_{j=0}^{N+1} \delta_{ij} \cdot \rho(d(x, x_j)) \right] = \sum_{i=0}^{N+1} \hat{y}_i \cdot \alpha_i(x), \quad (2.1)$$

where $\bullet$ is a Euclidean inner product, and the DNBM basis functions $\alpha_i(x) = \sum_{j=0}^{N+1} \delta_{ij} \cdot \rho(d(x, x_j))$. $\alpha_i(x)$ are adaptive interpolation weights that distribute the nodal information $\hat{y}_i$ to reconstruct $y(x)$

everywhere. When D_ρ is nonsingular, D_ρ^{-1} is used as defined above. In the numerical implementation, however, if D_ρ is singular or numerically rank-deficient, its Moore–Penrose pseudoinverse $D_\rho^\dagger$ is used in place of D_ρ^{-1} .

Figure 1 shows the transformation from one-dimensional numerical data into high-dimensional ones based on distance perspectives.

$$\vec{d}_X(x_j) = [\rho \circ d(x_j, x_0), \rho \circ d(x_j, x_1), \dots, \rho \circ d(x_j, x_N), \rho \circ d(x_j, x_{N+1})].$$

$$\vec{d}_X(x) = \sum_{k=0}^{N+1} c_k \cdot \vec{d}_X(x_k), \hat{y}(x) := \sum_{k=0}^{N+1} c_k \cdot \hat{y}_k; \underline{x} = x_k,$$

where $k = \underset{m}{\operatorname{argmin}}\{x \geq x_m : 0 \leq m \leq N+1\}; x \in [\alpha, \beta]$ and $\rho : \mathbb{R}_0^+ \rightarrow \mathbb{R}$.

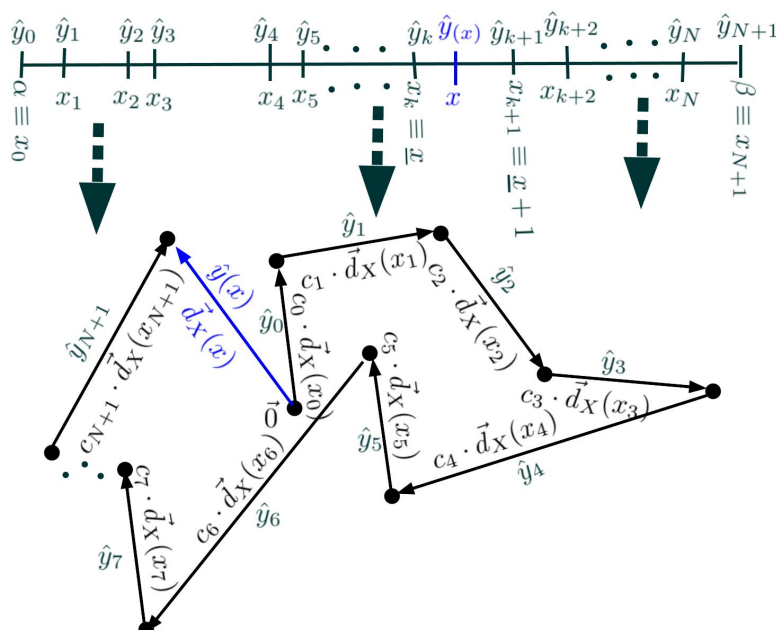


Figure 1. Distance-network-based numerical approximation of differential equations.

Before we proceed with our arguments and derivations, let us find some basic identities: For all $1 \leq i \leq N$,

$$\frac{\partial}{\partial \hat{y}_i} \hat{y}(x; \hat{Y}_X) = \sum_{s=0}^{N+1} \delta_{i,s} \cdot \rho(d(x, x_s)) = \alpha_i(x); \quad (2.2)$$

$$\hat{y}'(x; \hat{Y}_X) \equiv \frac{d\hat{y}(x; \hat{Y}_X)}{dx} = \sum_{i=0}^{N+1} \hat{y}_i \cdot \left[\sum_{t=0}^{N+1} \delta_{i,t} \cdot \rho'(d(x, x_t)) \cdot d'(x, x_t) \right] = \sum_{i=0}^{N+1} \hat{y}_i \cdot \beta_i(x), \quad (2.3)$$

where

$$\beta_i(x) = \sum_{t=0}^{N+1} \delta_{i,t} \cdot \rho'(d(x, x_t)) \cdot d'(x, x_t). \quad (2.4)$$

Furthermore,

$$\frac{\partial \hat{y}'(x; \hat{Y}_X)}{\partial \hat{y}_i} = \sum_{t=0}^{N+1} \delta_{i,t} \cdot \rho'(d(x; x_t)) \cdot d'(x; x_t) = \beta_i(x). \quad (2.5)$$

Claim 1. (Integrand L) If

$$L(x, y, y') := e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \left\{ [y'(x)]^2 - \frac{c_3(x)}{c_1(x)} \cdot [y(x)]^2 + 2 \cdot \frac{c_4(x)}{c_1(x)} \cdot y(x) \right\}, \quad (2.6)$$

then the Euler–Lagrange equation

$$\frac{dL(x, y, y')}{dy} - \frac{d}{dx} \left[\frac{dL(x, y, y')}{dy'} \right] = 0$$

is equivalent to Eq (1.1).

Proof. It follows from the direct computation of the Euler–Lagrange equation. $\square$

Definition 2.2. Define a functional $\Phi_{\alpha, \beta}^L : ([\alpha, \beta] \rightarrow \mathbb{R}) \rightarrow \mathbb{R}$ by

$$\Phi_{\alpha, \beta}^L(y) := \int_{\alpha}^{\beta} L(x, y, y') dx. \quad (2.7)$$

The parameterized approximation version could be hence represented by

$$L(x, \hat{y}, \hat{y}') := e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \left\{ [\hat{y}'(x)]^2 - \frac{c_3(x)}{c_1(x)} \cdot [\hat{y}(x)]^2 + 2 \cdot \frac{c_4(x)}{c_1(x)} \cdot \hat{y}(x) \right\}, \quad (2.8)$$

$$\Phi_{\alpha, \beta}^L(\hat{y}; \hat{Y}_X) := \int_{\alpha}^{\beta} L(x, \hat{y}(x; \hat{Y}_X), \hat{y}'(x; \hat{Y}_X)) dx. \quad (2.9)$$

In order for $\hat{y}(x; \hat{Y}_X)$ to attain an extremum, a necessary condition is

$$\nabla_{\hat{Y}_X} [\Phi_{\alpha, \beta}^L(\hat{y}(x; \hat{Y}_X))] \equiv \left[\frac{\partial}{\partial \hat{y}_j} \Phi_{\alpha, \beta}^L(\hat{y}(x; \hat{Y}_X)) \right]_{j=1}^N = [0]_{j=1}^N = 0.$$

Claim 2. If $\hat{y}(x; \hat{Y}_X)$ attains an extremum, then

$$\int_{\alpha}^{\beta} \frac{\partial}{\partial \hat{y}_i} L(x, \hat{y}(x; \hat{Y}_X), \hat{y}'(x; \hat{Y}_X)) dx = 0, \quad (2.10)$$

for all $1 \leq i \leq N$.

Because $\hat{y}_0$ and $\hat{y}_{N+1}$ are given, we only have to reveal the necessary conditions for $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N$. The results are detailed in the subsequent Lemma 1 and Theorem 1.

Lemma 1. If $\hat{y}(x; \hat{Y}_X)$ attains an extremum, then $\frac{\partial \Phi_{\alpha, \beta}^L(\hat{y}; \hat{Y}_X)}{\partial \hat{y}_i} = 0$; that is

$$\int_{\alpha}^{\beta} e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \left\{ \hat{y}'(x; \hat{Y}_X) \cdot \frac{\partial \hat{y}'(x; \hat{Y}_X)}{\partial \hat{y}_i} + \left[\frac{c_4(x) - c_3(x) \cdot \hat{y}(x; \hat{Y}_X)}{c_1(x)} \right] \cdot \frac{\partial \hat{y}(x; \hat{Y}_X)}{\partial \hat{y}_i} \right\} = 0 \quad (2.11)$$

for all $1 \leq i \leq N$.

Proof. By Claims 1 and 2, one has

$$\int_{\alpha}^{\beta} e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \frac{\partial}{\partial \hat{y}_i} \left\{ [\hat{y}'(x; \hat{Y}_X)]^2 - \frac{c_3(x)}{c_1(x)} \cdot [\hat{y}(x; \hat{Y}_X)]^2 + 2 \cdot \frac{c_4(x)}{c_1(x)} \cdot \hat{y}(x; \hat{Y}_X) \right\} = 0;$$

that is

$$2 \cdot \int_{\alpha}^{\beta} e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \left\{ \hat{y}'(x; \hat{Y}_X) \cdot \frac{\partial \hat{y}'(x; \hat{Y}_X)}{\partial \hat{y}_i} + \frac{c_4(x) - c_3(x) \cdot \hat{y}(x; \hat{Y}_X)}{c_1(x)} \cdot \frac{\partial \hat{y}(x; \hat{Y}_X)}{\partial \hat{y}_i} \right\} = 0.$$

□

Theorem 1. If $\hat{y}(x; \hat{Y}_X)$ attains an extremum, then

$$\begin{aligned} & \begin{bmatrix} \tilde{\gamma}_{11} & \tilde{\gamma}_{12} & \cdots & \tilde{\gamma}_{1,N-1} & \tilde{\gamma}_{1,N} \\ \tilde{\gamma}_{21} & \tilde{\gamma}_{22} & \cdots & \tilde{\gamma}_{2,N-1} & \tilde{\gamma}_{2,N} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \tilde{\gamma}_{N-1,1} & \tilde{\gamma}_{N-1,2} & \cdots & \tilde{\gamma}_{N-1,N-1} & \tilde{\gamma}_{N-1,N} \\ \tilde{\gamma}_{N,1} & \tilde{\gamma}_{N,2} & \cdots & \tilde{\gamma}_{N,N-1} & \tilde{\gamma}_{N,N} \end{bmatrix} \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_{N-1} \\ \hat{y}_N \end{bmatrix} \\ &= \begin{bmatrix} \tilde{\theta}_1 - \tilde{\gamma}_{1,0} \cdot \hat{y}_0 - \tilde{\gamma}_{1,N+1} \cdot \hat{y}_{N+1} \\ \tilde{\theta}_2 - \tilde{\gamma}_{2,0} \cdot \hat{y}_0 - \tilde{\gamma}_{2,N+1} \cdot \hat{y}_{N+1} \\ \vdots \\ \tilde{\theta}_{N-1} - \tilde{\gamma}_{N-1,0} \cdot \hat{y}_0 - \tilde{\gamma}_{N-1,N+1} \cdot \hat{y}_{N+1} \\ \tilde{\theta}_N - \tilde{\gamma}_{N,0} \cdot \hat{y}_0 - \tilde{\gamma}_{N,N+1} \cdot \hat{y}_{N+1} \end{bmatrix} \\ &= - \int_{\alpha}^{\beta} e^{\int \frac{c_2(x)}{c_1(x)} dx} \frac{c_4(x)}{c_1(x)} \begin{bmatrix} \alpha_1(x) \\ \alpha_2(x) \\ \vdots \\ \alpha_{N-1}(x) \\ \alpha_N(x) \end{bmatrix} dx - \begin{bmatrix} \tilde{\gamma}_{1,0} \\ \tilde{\gamma}_{2,0} \\ \vdots \\ \tilde{\gamma}_{N-1,0} \\ \tilde{\gamma}_{N,0} \end{bmatrix} \hat{y}_0 - \begin{bmatrix} \tilde{\gamma}_{1,N+1} \\ \tilde{\gamma}_{2,N+1} \\ \vdots \\ \tilde{\gamma}_{N-1,N+1} \\ \tilde{\gamma}_{N,N+1} \end{bmatrix} \hat{y}_{N+1}, \end{aligned}$$

where

$$\tilde{\gamma}_{ij} = \int_{\alpha}^{\beta} e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \left[\beta_i(x) \cdot \beta_j(x) - \frac{c_3(x)}{c_1(x)} \cdot \alpha_i(x) \cdot \alpha_j(x) \right] dx,$$

$$\tilde{\theta}_i = - \int_{\alpha}^{\beta} \left[e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \frac{c_4(x)}{c_1(x)} \cdot \alpha_i(x) \right] dx,$$

$$\alpha_i(x) = \sum_{s=0}^{N+1} \delta_{i,s} \cdot \rho(d(x, x_s)); \beta_i(x) = \sum_{t=0}^{N+1} \delta_{i,t} \cdot \rho'(d(x, x_t)) \cdot d'(x, x_t).$$

Proof. By Lemma 1 and Eqs (2.2), (2.3), and (2.5), one has, for each $1 \leq i \leq N$,

$$\begin{aligned} & \int_{\alpha}^{\beta} e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \left\{ \hat{y}'(x; \hat{Y}_X) \cdot \frac{\partial \hat{y}'(x; \hat{Y}_X)}{\partial \hat{y}_i} + \left[\frac{c_4(x) - c_3(x) \cdot \hat{y}(x; \hat{Y}_X)}{c_1(x)} \right] \cdot \frac{\partial \hat{y}(x; \hat{Y}_X)}{\partial \hat{y}_i} \right\} dx \\ &= \int_{\alpha}^{\beta} e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \left\{ \left[\sum_{j=0}^{N+1} \hat{y}_j \cdot \beta_j(x) \right] \cdot \beta_i(x) + \left[\frac{c_4(x) - c_3(x) \cdot \hat{y}(x; \hat{Y}_X)}{c_1(x)} \right] \cdot \alpha_i(x) \right\} dx = 0 \end{aligned}$$

$$\begin{aligned} \Rightarrow \sum_{j=0}^{N+1} \left\{ \int_{\alpha}^{\beta} e^{\int_{c_1(x)}^{c_2(x)} dx} \cdot \left[\beta_i(x) \cdot \beta_j(x) - \frac{c_3(x)}{c_1(x)} \cdot \alpha_i(x) \cdot \alpha_j(x) \right] dx \right\} \cdot \hat{y}_j &= - \int_{\alpha}^{\beta} \left[e^{\int_{c_1(x)}^{c_2(x)} dx} \cdot \frac{c_4(x)}{c_1(x)} \cdot \alpha_i(x) \right] dx \\ \Rightarrow \tilde{\gamma}_{i,0} \cdot \hat{y}_0 + \tilde{\gamma}_{i,N+1} \cdot \hat{y}_{N+1} + \sum_{j=1}^N \tilde{\gamma}_{ij} \cdot \hat{y}_j &= \tilde{\theta}_i \Rightarrow \sum_{j=1}^N \tilde{\gamma}_{ij} \cdot \hat{y}_j = \tilde{\theta}_i - \tilde{\gamma}_{i,0} \cdot \hat{y}_0 - \tilde{\gamma}_{i,N+1} \cdot \hat{y}_{N+1}, \end{aligned}$$

where $\tilde{\gamma}_{i,j}$ and $\tilde{\theta}_i$ are defined in the theorem. $\square$

This representation will be extended to piecewise functions for some practical reasons. Let $\chi_{k \geq i}$ denote the indicator function given by

$$\chi_{k \geq i} = \begin{cases} 1, & k \geq i, \\ 0, & k < i. \end{cases}$$

We begin by introducing the piecewise definitions. In practical applications, the functions under consideration may fail to be differentiable at finitely many points. To accommodate such cases, we partition their domains into finitely many subdomains such that the functions are twice differentiable on each subdomain. The following definitions, claims, and corollaries are formulated accordingly for this piecewise setting.

Definition 2.3. For all $1 \leq i, j \leq N$,

$$\langle \beta_i^k, \beta_j^k \rangle_{\mu} = \int_{x_k}^{x_{k+1}} e^{\int_{c_1(x)}^{c_2(x)} dx} \cdot [\beta_i^k(x) \cdot \beta_j^k(x)] dx,$$

where $\beta_i^k(x) = \sum_{s=0}^{N+1} \delta_{i,s} \cdot \rho'(d_k(x; x_s)) \cdot d'_k(x; x_s)$, where $d_k(\cdot; x_s) : (x_k, x_{k+1}) \rightarrow \mathbb{R}_0^+$.

Definition 2.4. For all $1 \leq i, j \leq N$,

$$\left\langle \frac{c_3(x)}{c_1(x)}, \alpha_i^k \cdot \alpha_j^k \right\rangle_{\mu} = \int_{x_k}^{x_{k+1}} e^{\int_{c_1(x)}^{c_2(x)} dx} \cdot \left[\frac{c_3(x)}{c_1(x)} \cdot \alpha_i^k(x) \cdot \alpha_j^k(x) \right] dx,$$

where $\alpha_i^k(x) = \sum_{j=0}^{N+1} \delta_{i,j} \cdot \rho(d_k(x, x_j))$ and where $d_k(\cdot; x_j) : (x_k, x_{k+1}) \rightarrow \mathbb{R}_0^+$.

Remark 1. $d'_k(x; x_j) = \begin{cases} 1, & \text{if } k \geq j, \\ -1, & \text{if } k < j. \end{cases}$; thus $d'_k(\cdot; x_j) = \chi_{k \geq j}$ and $d_k(x; x_j) = \chi_{k \geq j} \cdot (x - x_j)$.

Recall that

$$\langle \beta_i, \beta_j \rangle_{\mu} := \int_{\alpha}^{\beta} e^{\int_{c_1(x)}^{c_2(x)} dx} \cdot [\beta_i(x) \cdot \beta_j(x)] dx.$$

Thus, one has $\langle \beta_i, \beta_j \rangle_{\mu} = \sum_{k=0}^N \langle \beta_i^k, \beta_j^k \rangle_{\mu}$.

Claim 3. Fix $1 \leq i, j \leq N$. If $c_1(x) = x, c_2(x) = 1, \rho(z) = z^4$ and $d(x, y) = |x - y|$, then for all $0 \leq k \leq N, \langle \beta_i^k, \beta_j^k \rangle_\mu^k = 8$.

$$\sum_{s=0}^{N+1} \sum_{t=0}^{N+1} \delta_{is} \cdot \delta_{jt} \cdot \left\{ [x_{k+1}^2 \cdot (x_{k+1} - x_s)^3 \cdot (x_{k+1} - x_t)^3 - x_k^2 \cdot (x_k - x_s)^3 \cdot (x_k - x_t)^3] \right. \\ \left. - 3 \cdot \int_{x_k}^{x_{k+1}} x^2 \cdot (x - x_s)^2 \cdot (x - x_t)^2 \cdot [2\dot{x} - x_s - x_t] dx \right\},$$

where $\beta_i^k(x) = \sum_{s=0}^{N+1} \delta_{is} \cdot \rho'(d_k(x; x_s)) \cdot d'_k(x; x_s) \equiv 8 \cdot \langle \Delta_{ij}, \Theta_k \rangle$, where $\Delta_{ij} = [\delta_{is} \cdot \delta_{jt}]_{s,t=0}^{N+1}$ and $\Theta_k = [v_{st}]_{s,t=0}^{N+1}$, in which each $v_{st} = [x_{k+1}^2 \cdot (x_{k+1} - x_s)^3 \cdot (x_{k+1} - x_t)^3 - x_k^2 \cdot (x_k - x_s)^3 \cdot (x_k - x_t)^3] - 3 \cdot \int_{x_k}^{x_{k+1}} x^2 \cdot (x - x_s)^2 \cdot (x - x_t)^2 \cdot [2\dot{x} - x_s - x_t] dx$.

Proof. By Definition 2.3, one has $\langle \beta_i^k, \beta_j^k \rangle_\mu^k = \int_{x_k}^{x_{k+1}} e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot [\beta_i^k(x) \cdot \beta_j^k(x)] dx$. By setting $f'(x) = e^{\int \frac{c_2(x)}{c_1(x)} dx}$ and $g_{ij}^k(x) = \sum_{s=0}^{N+1} \sum_{t=0}^{N+1} \delta_{is} \cdot \delta_{jt} \cdot \rho'(d_k(x; x_s)) \cdot d'_k(x; x_s) \cdot \rho'(d_k(x; x_t)) \cdot d'_k(x; x_t) = 16 \cdot \sum_{s=0}^{N+1} \sum_{t=0}^{N+1} \delta_{is} \cdot \delta_{jt} \cdot \chi_{k \geq s}^3(x - x_s)^3 \cdot \chi_{k \geq t}^3(x - x_t)^3 \cdot \chi_{k \geq t}^3(x - x_t)^3 = 16 \cdot \sum_{s=0}^{N+1} \sum_{t=0}^{N+1} \delta_{is} \cdot \delta_{jt} \cdot (x - x_s)^3 \cdot (x - x_t)^3$, which is independent of k , then we could take $f(x) = \frac{x^2}{2}$ and

$$\int_{x_k}^{x_{k+1}} f'(x) \cdot g_{ij}^k(x) dx = \lim_{\epsilon \rightarrow 0^+} \int_{x_k + \epsilon}^{x_{k+1} - \epsilon} f'(x) \cdot g_{ij}^k(x) dx \\ = \lim_{\epsilon \rightarrow 0^+} [f(x) \cdot g_{ij}^k(x)]_{x_k + \epsilon}^{x_{k+1} - \epsilon} - \lim_{\epsilon \rightarrow 0^+} \int_{x_k + \epsilon}^{x_{k+1} - \epsilon} f(x) \cdot g_{ij}^k(x) dx \\ = [f(x_{k+1}) \cdot g_{ij}^k(x_{k+1}) - f(x_k) \cdot g_{ij}^k(x_k)] \\ - 24 \cdot \sum_{s=0}^{N+1} \sum_{t=0}^{N+1} \delta_{is} \cdot \delta_{jt} \cdot \int_{x_k}^{x_{k+1}} x^2 \cdot (x - x_s)^2 \cdot (x - x_t)^2 \cdot [2\dot{x} - x_s - x_t] dx \\ = 8 \cdot \sum_{s=0}^{N+1} \sum_{t=0}^{N+1} \delta_{is} \cdot \delta_{jt} \cdot [x_{k+1}^2 \cdot (x_{k+1} - x_s)^3 \cdot (x_{k+1} - x_t)^3 - x_k^2 \cdot (x_k - x_s)^3 \cdot (x_k - x_t)^3] \\ - 24 \cdot \sum_{s=0}^{N+1} \sum_{t=0}^{N+1} \delta_{is} \cdot \delta_{jt} \cdot \int_{x_k}^{x_{k+1}} x^2 \cdot (x - x_s)^2 \cdot (x - x_t)^2 \cdot [2\dot{x} - x_s - x_t] dx.$$

□

Claim 4. For all $1 \leq i, j \leq N$,

$$\langle \beta_i, \beta_j \rangle_\mu = 8 \cdot [x_{N+1}^2 \cdot (x_{N+1} - x_i)^3 \cdot (x_{N+1} - x_j)^3 - x_N^2 \cdot (x_N - x_i)^3 \cdot (x_N - x_j)^3] \\ - 24 \cdot \int_{x_0}^{x_{N+1}} x^2 \cdot (x - x_i)^2 \cdot (x - x_j)^2 \cdot [2\dot{x} - x_i - x_j] dx.$$

Proof. By the fact that $\langle \beta_i, \beta_j \rangle_\mu = \sum_{k=0}^N \langle \beta_i, \beta_j \rangle_\mu^k$ and Claim 3, the result follows. $\square$

Claim 5. If $c_1(x) = x$, $c_3(x) = 2x^2$, $\rho(z) = z^4$ and $d(x, y) = |x - y|$, then for all $1 \leq i, j \leq N$, $\langle \frac{c_3(x)}{c_1(x)}$,

$$\begin{aligned} \alpha_i^k \cdot \alpha_j^k \rangle_\mu &= \frac{2}{3} \cdot \sum_{s=0}^{N+1} \sum_{t=0}^{N+1} \delta_{is} \cdot \delta_{jt} \\ &\cdot \left\{ [x^3 \cdot (x - x_i)^4 \cdot (x - x_j)^4]_{x_k}^{x_{k+1}} - 4 \cdot \int_{x_k}^{x_{k+1}} x^3 \cdot (x - x_i)^3 \cdot (x - x_j)^3 \cdot (2x - x_i - x_j) dx \right\} \\ &= \frac{2}{3} \cdot \langle \Delta_{ij}, \tilde{\Theta}_k \rangle, \text{ where } \Delta_{ij} = [\delta_{is} \cdot \delta_{jt}]_{s,t=0}^{N+1}, \text{ and } \tilde{\Theta}_k = [\tilde{v}_{st}]_{s,t=0}^{N+1}, \end{aligned}$$

where

$$\tilde{v}_{st} = [x^3 \cdot (x - x_i)^4 \cdot (x - x_j)^4]_{x_k}^{x_{k+1}} - 4 \cdot \int_{x_k}^{x_{k+1}} x^3 \cdot (x - x_i)^3 \cdot (x - x_j)^3 \cdot (2x - x_i - x_j) dx.$$

Proof. By Definition 2.4,

$$\langle \frac{c_3(x)}{c_1(x)}, \alpha_i^k \cdot \alpha_j^k \rangle_\mu = 2 \cdot \int_{x_k}^{x_{k+1}} x^2 \cdot [\alpha_i^k(x) \cdot \alpha_j^k(x)] dx,$$

where $\alpha_i^k(x) = \sum_{s=0}^{N+1} \delta_{is} \cdot (x - x_s)^4 = 2 \cdot \int_{x_k}^{x_{k+1}} \phi'(x) \cdot q_{ij}^k(x) dx$, in which $\phi'(x) = x^2$, and

$$\begin{aligned} q_{ij}^k(x) &= \alpha_i^k(x) \cdot \alpha_j^k(x) = 2 \cdot [\phi(x) \cdot q_{ij}^k(x)]_{x_k}^{x_{k+1}} - 2 \cdot \int_{x_k}^{x_{k+1}} \phi(x) \cdot \frac{d}{dx} q_{ij}^k(x) dx \\ &= \frac{2}{3} \cdot \sum_{s=0}^{N+1} \sum_{t=0}^{N+1} \delta_{is} \cdot \delta_{jt} \cdot \left\{ [x^3 \cdot (x - x_i)^4 \cdot (x - x_j)^4]_{x_k}^{x_{k+1}} - 4 \cdot \int_{x_k}^{x_{k+1}} x^3 \cdot (x - x_i)^3 \cdot (x - x_j)^3 \cdot (2x - x_i - x_j) dx \right\}. \end{aligned}$$

$\square$

Corollary 1. If $c_1(x) = x$, $c_3(x) = 2x^2$, $\rho(z) = z^4$, and $d(x, y) = |x - y|$, then for all $1 \leq i, j \leq N$, $\langle \frac{c_3(x)}{c_1(x)}$,

$$\begin{aligned} \alpha_i \cdot \alpha_j \rangle_\mu &= \frac{2}{3} \cdot \sum_{s=0}^{N+1} \sum_{t=0}^{N+1} \delta_{is} \cdot \delta_{jt} \\ &\cdot \left\{ [x^3 \cdot (x - x_i)^4 \cdot (x - x_j)^4]_{x_0}^{x_{N+1}} - 4 \cdot \int_{x_0}^{x_{N+1}} x^3 \cdot (x - x_i)^3 \cdot (x - x_j)^3 \cdot (2x - x_i - x_j) dx \right\}. \end{aligned}$$

Corollary 2. $[\langle \beta_i, \beta_j \rangle_\mu - \langle \frac{c_3}{c_1}, \alpha_i \cdot \alpha_j \rangle_\mu]_{i,j=1}^N \hat{Y} = [-\langle \frac{c_4}{c_1}, \alpha_i \rangle_\mu - \langle \beta_i, \beta_0 \rangle_\mu \cdot \hat{y}_0 - \langle \beta_i, \beta_{N+1} \rangle_\mu \cdot \hat{y}_{N+1}]_{i=1}^N$.

Let us use $\tilde{R}\hat{Y} = \tilde{Q}$ to abbreviate the above corollary.

Theorem 2.

$$\hat{Y} \equiv [\hat{y}_j]_{j=1}^N = \tilde{R}^{-1} \tilde{Q}, \quad (2.12)$$

if $\tilde{R}^{-1}$ exists.

In order to make it this theory applicable to some special cases, we have to temper the original definitions by splitting the domain $[\alpha, \beta]$ into various subdomains. Then, the counterparts with respect to the subdomains could be rewritten as follows:

Claim 6. For all $x_k < x < x_{k+1}$, if $\rho(z) = z^4$, and $d(x, y) = |x - y|$, then $\alpha_i^k(x) = \sum_{s=0}^{N+1} \delta_{i,s} \cdot (x - x_s)^4$, $\beta_i^k(x) = 4 \cdot \sum_{t=0}^{N+1} \delta_{i,t} \cdot (x - x_t)^3$, and $\frac{d\alpha_i^k(x)}{dx} = \beta_i^k(x)$.

Proof. Because

$$\alpha_i^k(x) = \sum_{j=0}^{N+1} \delta_{i,j} \cdot \rho(d_k(x, x_j)); \beta_i^k(x) = \sum_{j=0}^{N+1} \delta_{i,j} \cdot \rho'(d_k(x, x_j)) \cdot d'_k(x, x_j),$$

the results follow. $\square$

Obviously, the result is independent of k . Similarly, we have the following piecewise definitions.

Definition 2.5.

$$\tilde{\gamma}_{ij}^k = \int_{x_k}^{x_{k+1}} e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \left[\beta_i^k(x) \cdot \beta_j^k(x) - \frac{c_3(x)}{c_1(x)} \cdot \alpha_i^k(x) \cdot \alpha_j^k(x) \right] dx;$$

$$\tilde{\theta}_i^k = - \int_{x_k}^{x_{k+1}} \left[e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \frac{c_4(x)}{c_1(x)} \cdot \alpha_i^k(x) \right] dx.$$

Definition 2.6. (Residual measure) Define a residual function $R(x)$ by $R(x) = c_1(x) \cdot y''(x) + c_2(x) \cdot y'(x) + c_3(x) \cdot y(x) - c_4(x)$, which is defined in Definition 1.1.

Example 1. Suppose that $\alpha = x_0 = 0.01$ and $\beta = x_4 = 4$. Assume that $y(0.01) = 2$ and that $y(4) = 5$. Suppose that

$$X = [\alpha, \alpha + \frac{1}{4} \cdot (\beta - \alpha), \alpha + \frac{2}{4} \cdot (\beta - \alpha), \alpha + \frac{3}{4} \cdot (\beta - \alpha), \beta] = [0.0100, 1.0075, 2.0050, 3.0025, 4.0000].$$

Suppose $c_1(x) = x$, $c_2(x) = 1$, $c_3(x) = 2x^2$, $c_4(x) = x^2$. Moreover, $N = 3$. If $\rho(z) := z^4$ and $d(x, y) = |x - y|$, then $D_\rho \equiv [\rho(d(x_i, x_j))]_{i,j=0}^4 := [(d(x_i, x_j))^4]_{i,j=0}^4 := [(x_i - x_j)^4]_{i,j=0}^4$. Hence,

$$D_\rho^{-1} = \begin{bmatrix} 0.0039 & -0.0226 & 0.0515 & 0.0129 & -0.0032 \\ -0.0226 & 0.1314 & -0.3083 & 0.0183 & 0.0128 \\ 0.0515 & -0.3083 & 0.7635 & -0.3058 & 0.0504 \\ 0.0129 & 0.0183 & -0.3058 & 0.1293 & -0.0219 \\ -0.0032 & 0.0128 & 0.0504 & -0.0219 & 0.0037 \end{bmatrix}.$$

Let us adopt $d_k(x; x_i)$ to mean $|x - x_i|$, where $x_k < x < x_{k+1}$. Then one has $f'(x) = e^{\int \frac{c_2}{c_1} dx} = e^{\int \frac{1}{x} dx} = e^{x+c} = e^{\ln(x)+c} = e^c \cdot x$. Hence, by integration by parts, we could take $f(x) = \frac{1}{2}x^2$. Then, $f(x) = \frac{1}{2}x^2$.

$$\begin{aligned} \langle \beta_i, \beta_j \rangle_\mu|_{x_k}^{x_{k+1}} &= c_{ij}^k - \int_{x_k}^{x_{k+1}} f(x) \cdot \{[\rho'(d(x; x_i)) \cdot d'(x; x_i)]' \cdot [\rho'(d(x; x_j)) \cdot d'(x; x_j)] \\ &\quad + [\rho'(d(x; x_i)) \cdot d'(x; x_i)] \cdot [\rho'(d(x; x_j)) \cdot d'(x; x_j)]'\} dx, \end{aligned}$$

where

$$c_{ij}^k = \lim_{\epsilon \rightarrow 0} f(x_{k+1} - \epsilon) \cdot g_{ij}^k(x_{k+1} - \epsilon) - f(x_k + \epsilon) \cdot g_{ij}^k(x_k + \epsilon)$$

$$= 8 \cdot \chi_{k \geq i} \cdot \chi_{k \geq j} \cdot [x_{k+1}^2 \cdot (x_{k+1} - x_i)^3 \cdot (x_{k+1} - x_j)^3 - x_k^2 \cdot (x_k - x_i)^3 \cdot (x_k - x_j)^3]$$

and

$$\begin{aligned} g_{ij}^k(x) &= \rho'(d(x; x_i)) \cdot d'(x; x_i) \cdot \rho'(d(x; x_j)) \cdot d'(x; x_j) \\ &= 16 \cdot \chi_{k \geq i} \cdot \chi_{k \geq j} \cdot (x - x_i)^3 \cdot (x - x_j)^3, \end{aligned}$$

by Remark 1. Furthermore, after a sequence of derivations, one has

$$\begin{aligned} \langle \beta_i, \beta_j \rangle_\mu &= \sum_{k=0}^N \langle \beta_i, \beta_j \rangle_\mu|_{x_k}^{x_{k+1}} \\ &= \sum_{k=0}^N c_{ij}^k - 48 \cdot \int_{x_0}^{x_{N+1}} x \cdot [(x - x_i)^2 \cdot (x - x_j)^3 + (x - x_i)^3 \cdot (x - x_j)^2] dx, \end{aligned}$$

where

$$\sum_{k=0}^N c_{ij}^k = 8 \cdot \sum_{k=0}^N \{ \chi_{k \geq i} \cdot \chi_{k \geq j} \cdot [x_{k+1}^2 \cdot (x_{k+1} - x_i)^3 \cdot (x_{k+1} - x_j)^3 - x_k^2 \cdot (x_k - x_i)^3 \cdot (x_k - x_j)^3] \}.$$

Similarly,

$$\begin{aligned} \langle \frac{c_3}{c_1}, \alpha_i \rangle_\mu &= \langle \frac{c_3}{c_1}, \sum_{k=0}^N \alpha_i^k \rangle_\mu = \sum_{j=0}^{N+1} \delta_{ij} \cdot \int_{x_0}^{x_{N+1}} e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \frac{c_3(x)}{c_1(x)} \cdot (x - x_j)^4 dx \\ &= \sum_{j=0}^{N+1} \delta_{ij} \cdot \int_{x_0}^{x_{N+1}} 2x^2 \cdot (x - x_j)^4 dx, \\ \langle \frac{c_4}{c_1}, \alpha_i \rangle_\mu &= \langle \frac{c_4}{c_1}, \sum_{k=0}^N \alpha_i^k \rangle_\mu = \sum_{j=0}^{N+1} \delta_{ij} \cdot \int_{x_0}^{x_{N+1}} e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \frac{c_4(x)}{c_1(x)} \cdot (x - x_j)^4 dx \\ &= \sum_{j=0}^{N+1} \delta_{ij} \cdot \int_{x_0}^{x_{N+1}} x^2 \cdot (x - x_j)^4 dx. \end{aligned}$$

By Claims 4 and 5, Corollaries 1-2, and above-mentioned formulas, one has

$$\tilde{R} = \begin{bmatrix} 3.4280 & -4.9090 & 3.1410 \\ -4.9090 & 11.8400 & -10.1800 \\ 3.1410 & -10.1800 & 14.2300 \end{bmatrix} - \begin{bmatrix} 2.6360 & -1.7890 & 2.6480 \\ -1.7890 & 12.4300 & -6.1260 \\ 2.6480 & -6.1260 & 25.7600 \end{bmatrix}$$

$$= \begin{bmatrix} 0.7922 & -3.1199 & 0.4931 \\ -3.1199 & -0.5957 & -4.0521 \\ 0.4931 & -4.0521 & -11.5220 \end{bmatrix}; \quad \tilde{Q} = \begin{bmatrix} 3.0905 \\ -12.7193 \\ 14.0254 \end{bmatrix}.$$

The residual defined in Definition 2.6 is calculated and then presented in Table 1 and Figure 2.

Table 1. Residuals calculated based on DNBm: In this example, there are only five nodal points being considered. All the terms are defined earlier.

x_k	$\hat{y}(x_k)$	$\hat{y}'(x_k) = \sum_i \hat{y}_i \alpha'_i(x_k)$	$\hat{y}''(x_k)$	$R(x_k)$
0.0100	2.000000	14.422700	-30.559200	50.733560
1.0075	5.414400	-3.786700	-7.903300	-7.471981
2.0050	0.216400	-4.496200	4.774700	12.013040
3.0025	-1.061700	2.343300	7.194600	-19.570090
4.0000	5.000000	8.343600	2.097500	32.572530

In the table, as one can see, $R(x_4)$ is high. This persists even for the linear Ritz FEM or the quadratic Ritz FEM. Here we apply it just for demonstrative purposes. For a serious approximation and interpolation, we need to increase the sampled points and the sampling methods.

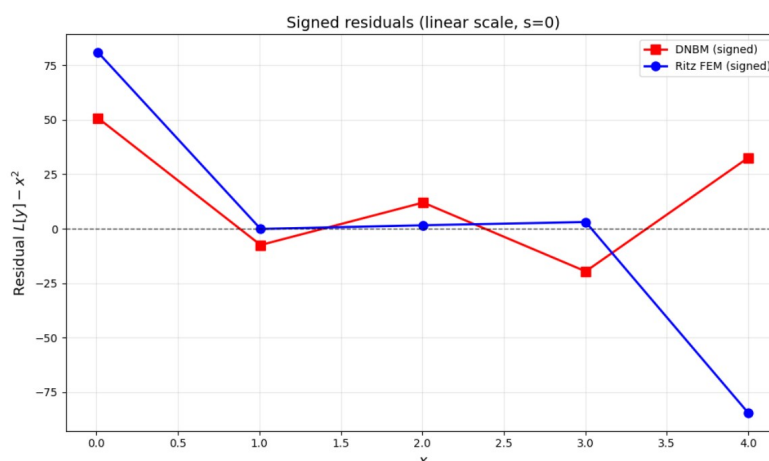


Figure 2. Signed residuals for uniformly-spaced X values: The DNBm part corresponds to Table 1. For this example, both DNBm and Ritz FEM could not perform well at the boundary points. For the related code, please refer to <https://github.com/raymingchen/ODE/blob/main/Eg1.ipynb>.

If $\tilde{R}$ is not always invertible, we then extend all the inferences to another optimization problem (for a given N -by- N matrix W and a N -by-1 matrix V):

$$\hat{Y}_* \equiv \underset{\hat{Y}}{\operatorname{argmin}} \{ \|(W \odot \tilde{R})\hat{Y} - V \odot \tilde{Q}\| : \hat{Y} \in \mathbb{R}^N \},$$

where $\odot$ stands for the Hadamard product.

Theorem 3. *If $W \odot \tilde{R}$ is invertible, then*

$$\hat{Y}_* = [W \odot \tilde{R}]^{-1} [V \odot \tilde{Q}]. \quad (2.13)$$

Proof. The norm with value 0 could be attained when $W \odot \tilde{R}$ is invertible. $\square$

Let

$$w_{ij} = \frac{e^{-d(x_i, x_j)}}{\sum_{s=1}^N e^{-d(x_i, x_s)}}; v_i = \frac{\sum_{s=1}^N e^{-d(x_i, x_s)}}{\sum_{s,t=1}^N e^{-d(x_s, x_t)}} \quad (2.14)$$

for all $1 \leq i, j \leq N$. Then, the natural candidates for weight matrices are $W = [w_{ij}]_{i,j=1}^N$ and $V = [v_i]_{i=1}^N$. Observe that W is not necessarily a symmetric matrix. Note that the Hadamard weighting in Theorem 3 regularizes the variational matrix R , not the distance matrix D_ρ . The singularity of D_ρ is handled separately via pseudoinverse construction of the basis functions.

3. Application

In the theory, D_ρ^{-1} appears only in the symbolic definition of the basis functions. In the numerical implementation, the Moore–Penrose pseudoinverse of D_ρ is used solely to construct these basis functions. The actual ODE solution is computed using the variational matrix R , which is full-rank. Thus, the rank-deficiency of D_ρ does not affect the numerical method. In order to show the validity of our method, we use some known second-order ODE problems with exact solutions to check our method. We know that

$$y(x) = x + e^x \sin(\pi x), \quad (3.1)$$

is an exact solution of the following ODE problem:

$$\begin{cases} (1 + x^2)y''(x) + (\sin x + x)y'(x) + (2 + \cos(2\pi x))y(x) = c_4(x), \\ y(0.01) = 0.0417, \quad y(4) = 4, \end{cases} \quad (3.2)$$

where

$$\begin{aligned} c_4(x) = & (1 + x^2)e^x (2\pi \cos(\pi x) + (1 - \pi^2) \sin(\pi x)) \\ & + (\sin x + x) (1 + e^x (\sin(\pi x) + \pi \cos(\pi x))) \\ & + (2 + \cos(2\pi x)) (x + e^x \sin(\pi x)). \end{aligned} \quad (3.3)$$

In Example 1, we set $\rho(z) = z^4$ and $d(x, y) = |x - y|$. If we look into the setting of ρ , then we observe that the setting is arbitrary. To counter the side effect from this choice, the choice of X could play an important role. Here, we apply a clustering function, mainly a *tanh* function in our case, to serve as the sampling technique. Then, one could observe the effect of the combination X and ρ . The parameterized clustering function we apply goes as follows:

$$x_k = \alpha + (\beta - \alpha) \cdot \frac{\frac{\tanh(s \cdot (k/N - 0.5))}{\tanh(s/2)} + 1}{2}, \text{ where } k \in \{1, 2, \dots, N\}.$$

Observe that the clustering strength s controls the sampling distribution $X = \{x_k\}$: from uniform distributions (or $s \rightarrow 0$) to polarized/boundary distributions ($s \rightarrow \infty$). An illustrative visualization of X distributions is shown in Figure 3.

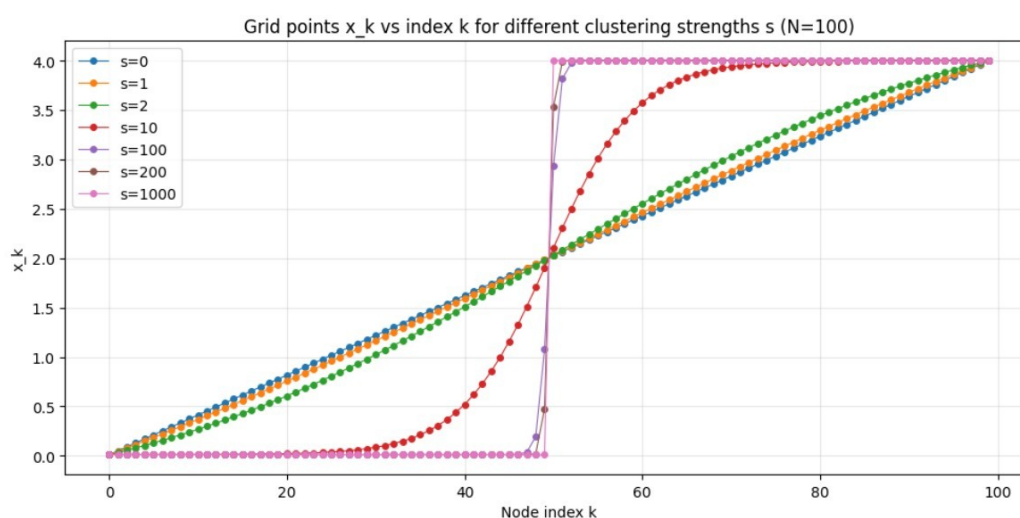


Figure 3. Sampled points x_k from $x_k = \alpha + (\beta - \alpha) \cdot \frac{\frac{\tanh(s \cdot (k/N - 0.5))}{\tanh(s/2)} + 1}{2}$, where $k \in \{1, 2, \dots, N\}$ under various clustering strengths s . If s approaches 0, the distribution tends to be uniform; otherwise, it tends to be polarized. For a complete code, please refer to https://github.com/raymingchen/ODE/blob/main/x_k_for_various_s.ipynb.

Now, based on this sampling technique/function, we could expand our knowledge to investigate the numerical solutions and the distribution of X values. Such knowledge will reveal the sensitivity and stability of one method. Here we compare the DNBM and Ritz FEM methods together. The calculations are based on the standard procedures for each method. We divide the scenarios into two: one with sample size 10, and the other with sample size 500, with various clustering strengths for various values $s = 0, 1, 10, 100$. The results are shown in Figures 4 and 5. In this example, one observes that both methods (DNBM and Ritz FEM) do not perform as one expects, as the exact solution is highly nonlinear. The degree of nonlinearity of both methods cannot capture the exact solution well. Now, let us look at some solutions that might have lower nonlinearity. Despite that, Ritz FEM has better approximations in both residuals and actual errors. The new ODE problem is $x y''(x) + y'(x) + 2x^2 y(x) = x^2$, in which the boundary conditions are $y(0.01) = 2$ and $y(4) = 5$. The numerical solutions are given in Figure 6. The residuals are all managed by both methods, although DNBM has a better approximation in this case.

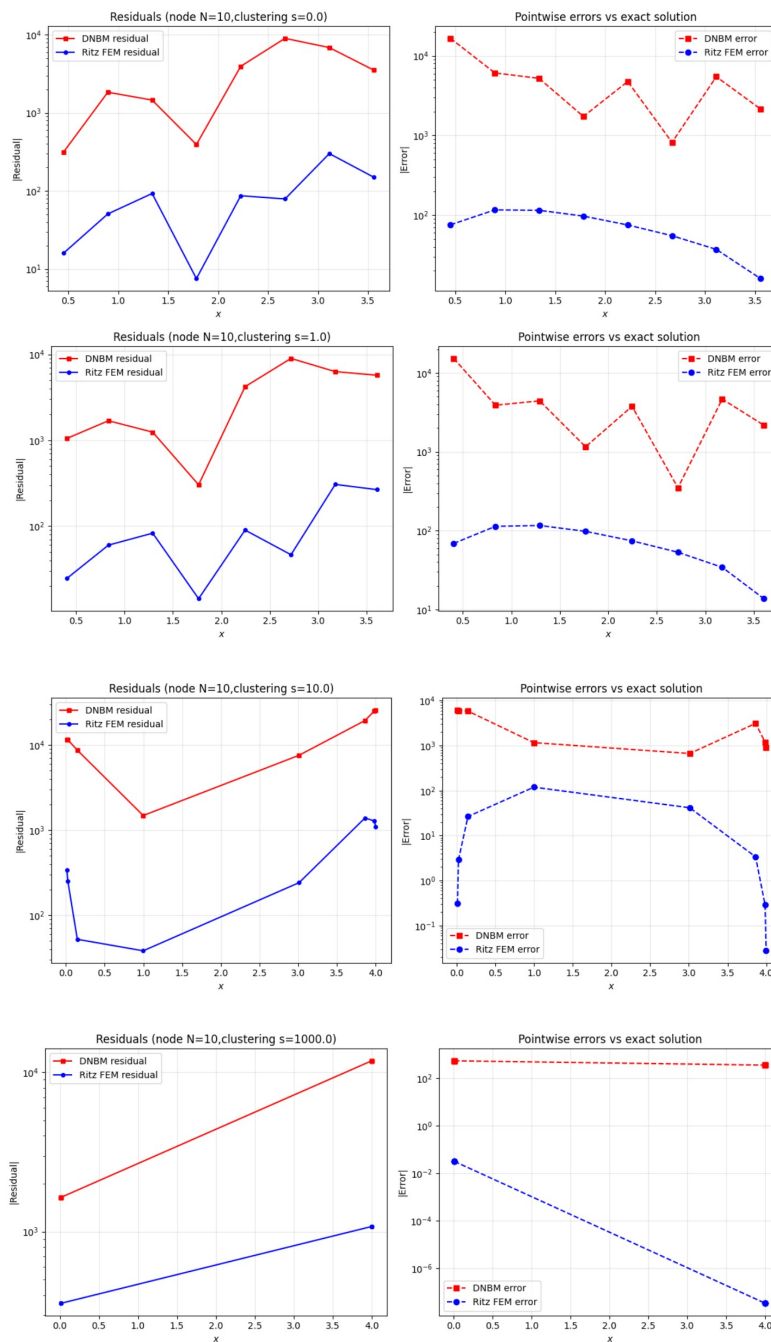


Figure 4. These figures show how DNBM and Ritz FEM perform in a given ODE problem: $(1 + x^2)y''(x) + (\sin x + x)y'(x) + (2 + \cos(2\pi x))y(x) = c_4(x)$; $y(0.01) = 0.0417$; $y(4) = 4$, with an exact solution $y(x) = x + e^x \sin(\pi x)$. In this experiment, N is fixed to be 10 and the clustering strength $s \in \{0, 1, 10, 1000\}$. On the y-axis, we use absolute value symbols to indicate that the computed values are represented by their positive counterparts. The experimental code can be found at <https://github.com/raymingchen/ODE/blob/main/2ndODEproblem.ipynb>.

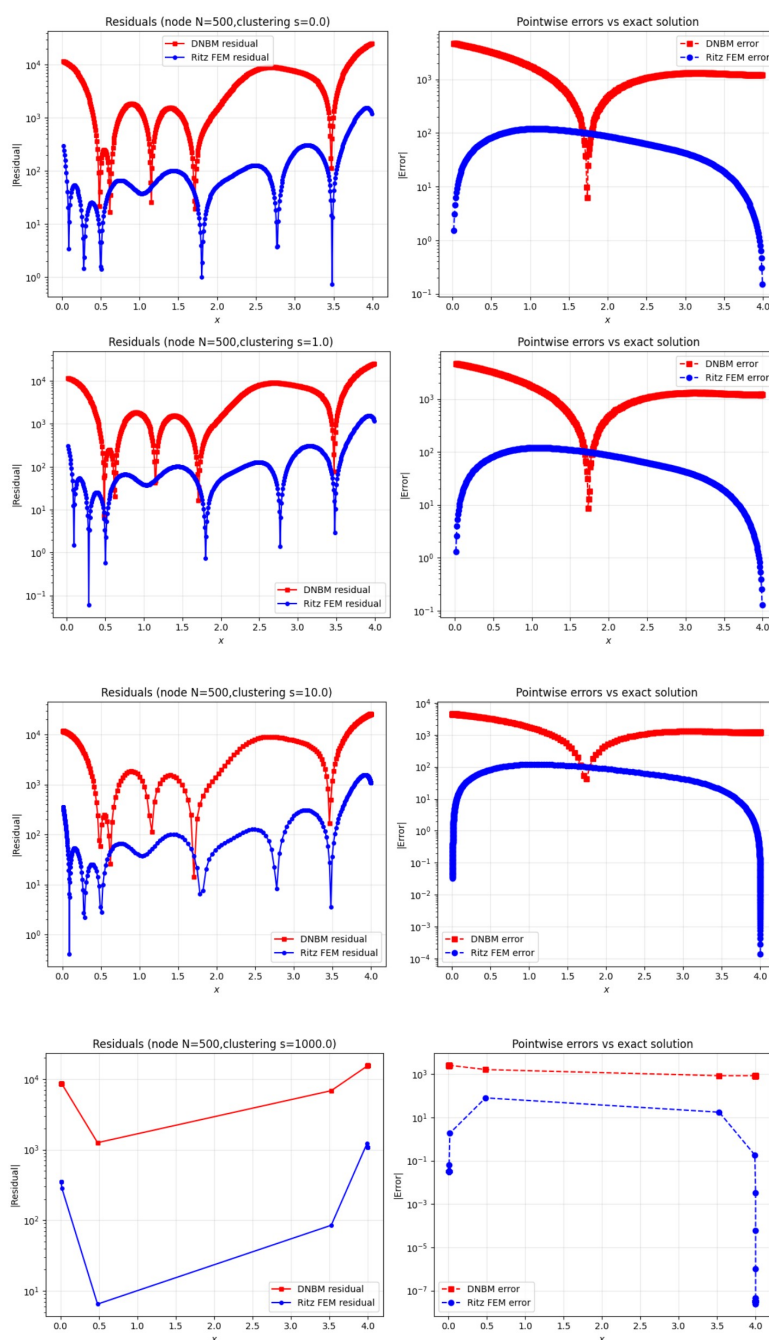
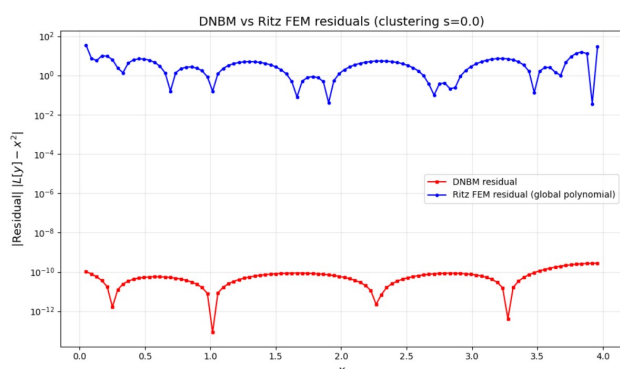
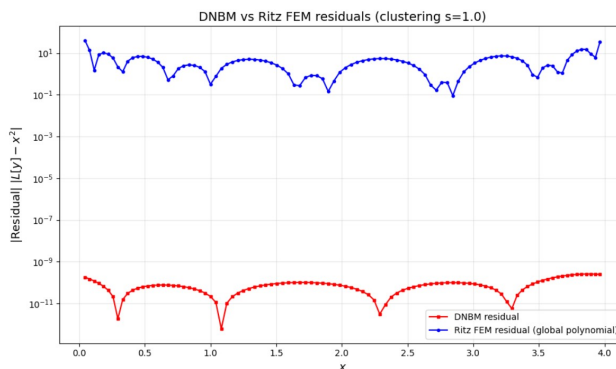


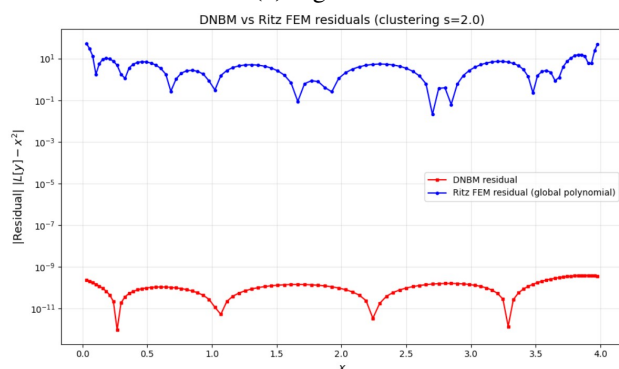
Figure 5. These figures show how DNBM and Ritz FEM perform in a given ODE problem: $(1 + x^2)y''(x) + (\sin x + x)y'(x) + (2 + \cos(2\pi x))y(x) = c_4(x)$; $y(0.01) = 0.0417$; $y(4) = 4$, with an exact solution $y(x) = x + e^x \sin(\pi x)$. In this experiment, N is fixed to be 500 and the clustering strength $s \in \{0, 1, 10, 1000\}$. On the y-axis, we use absolute value symbols to indicate that the computed values are represented by their positive counterparts. The experimental code can be found at <https://github.com/raymingchen/ODE/blob/main/2ndODEproblem.ipynb>.



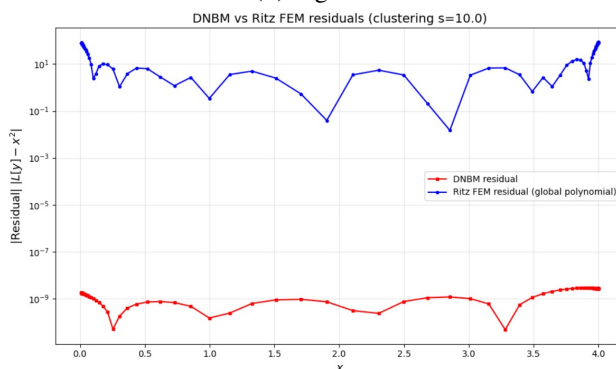
(a) Figure 1



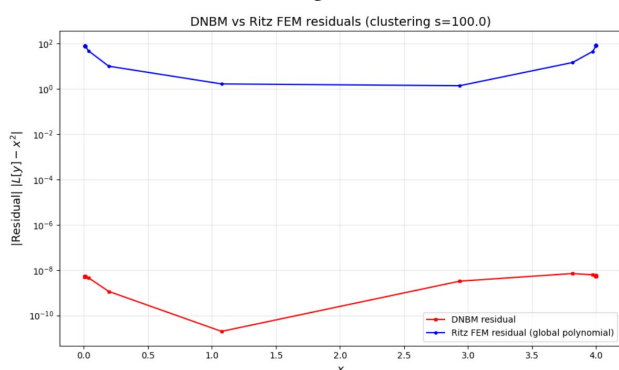
(b) Figure 2



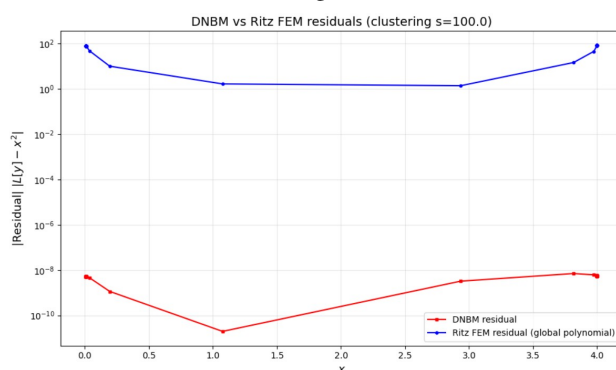
(c) Figure 3



(d) Figure 4



(e) Figure 5



(f) Figure 6

Figure 6. The comparison between DNBm and Ritz FEM under various clustering strengths s . These numerical approaches solve the ODE problem: $xy''(x) + y'(x) + 2x^2y(x) = x^2$, in which the boundary conditions are $y(0.01) = 2$ and $y(4) = 5$. Please refer to the code (by changing the values of s) at https://github.com/raymingchen/ODE/blob/main/DNBm_FEM.ipynb.

4. Analysis and comparison

4.1. Residual evaluation

The residual of the numerical approximation $\hat{y}(x)$ is defined as

$$R(x_k) = x_k \hat{y}''(x_k) + \hat{y}'(x_k) + 2x_k^2 \hat{y}(x_k) - x_k^2,$$

for each grid point $x_k, k = 0, 1, \dots, 99$. This is simply the ODE

$$xy'' + y' + 2x^2y = x^2$$

rearranged, with the exact solution y replaced by the numerical approximation $\hat{y}$.

4.1.1. Ritz FEM (Piecewise linear)

On each interval $[x_j, x_{j+1}]$, the approximation $\hat{y}(x)$ is linear:

$$\hat{y}(x) = \hat{y}_j + \frac{\hat{y}_{j+1} - \hat{y}_j}{h_j}(x - x_j), \quad h_j = x_{j+1} - x_j.$$

Therefore,

$$\hat{y}'(x) = \frac{\hat{y}_{j+1} - \hat{y}_j}{h_j}, \quad \hat{y}''(x) = 0.$$

At each node, x_k (except the last), the residual is computed using the slope from the interval to the right:

$$R(x_k) = \frac{\hat{y}_{k+1} - \hat{y}_k}{h_k} + 2x_k^2 \hat{y}_k - x_k^2.$$

At the last node $x_{99} = 4$, the slope from the interval to the left is used:

$$R(x_{99}) = \frac{\hat{y}_{99} - \hat{y}_{98}}{h_{98}} + 2x_{99}^2 \hat{y}_{99} - x_{99}^2.$$

This is exactly what the `residual_fem` function implements.

4.1.2. DNBM (Global quartic interpolation)

The approximation is given by

$$\hat{y}(x) = \sum_{i=0}^{99} \hat{y}_i \alpha_i(x),$$

where

$$\alpha_i(x) = \sum_{j=0}^{99} \delta_{ij}(x - x_j)^4.$$

Its derivatives are

$$\hat{y}'(x) = \sum_{i=0}^{99} \hat{y}_i \beta_i(x), \quad \hat{y}''(x) = \sum_{i=0}^{99} \hat{y}_i \gamma_i(x),$$

with

$$\beta_i(x) = \sum_{j=0}^{99} \delta_{ij} \cdot 4(x - x_j)^3, \quad \gamma_i(x) = \sum_{j=0}^{99} \delta_{ij} \cdot 12(x - x_j)^2.$$

At each point x_k , we evaluate these sums to obtain $\hat{y}(x_k)$, $\hat{y}'(x_k)$, and $\hat{y}''(x_k)$, and then compute

$$R(x_k) = x_k \hat{y}''(x_k) + \hat{y}'(x_k) + 2x_k^2 \hat{y}(x_k) - x_k^2.$$

This is exactly what the `residual_dnbm` function implements.

4.2. Summary

Both methods evaluate the same residual equation:

$$R(x_k) = x_k \hat{y}''(x_k) + \hat{y}'(x_k) + 2x_k^2 \hat{y}(x_k) - x_k^2.$$

The only difference lies in how $\hat{y}$, $\hat{y}'$, and $\hat{y}''$ are obtained. This completes the description of the residual sampling equation and its numerical implementation. The difference between them could be summed up in Table 2.

Table 2. Summary of FEM and DNBM: This table compares the difference between FEM and DNBM from original function, first order and second order functions.

Method	$\hat{y}(x)$	$\hat{y}'(x)$	$\hat{y}''(x)$
FEM	piecewise linear	constant on each element	0
DNBM	$\sum_{i=0}^{99} \hat{y}_i \alpha_i(x)$	$\sum_{i=0}^{99} \hat{y}_i \beta_i(x)$	$\sum_{i=0}^{99} \hat{y}_i \gamma_i(x)$

4.2.1. Global versus local basis functions

FEM uses piecewise linear (or polynomial) basis functions that are *local*: Each basis function is nonzero only on a small interval (one or two elements). On a highly clustered grid, some elements become extremely small (e.g., the last interval is $\approx 10^{-6}$ in our case). The derivative $\hat{y}'(x)$ is approximated by the finite difference

$$\frac{\hat{y}_{j+1} - \hat{y}_j}{h},$$

where $h = x_{j+1} - x_j$. If the solution varies smoothly, the slope can become enormous when h is tiny, even if $\hat{y}_{j+1} - \hat{y}_j$ is moderate. This leads to artificially amplified residuals at the clustered boundary, as observed ($\sim 10^6$ at $x = 4$). DNBM uses global basis functions derived from distance matrices:

$$\alpha_i(x) = \sum_j \delta_{ij} \rho(d(x, x_j)),$$

which are typically smooth functions (e.g., quartic polynomials when $\rho(z) = z^4$). These basis functions are *global*: Each $\alpha_i(x)$ is defined over the entire domain and is smooth, regardless of the distribution of grid points. The derivatives $\hat{y}'(x)$ and $\hat{y}''(x)$ are computed analytically from these smooth polynomials, not from finite differences. Therefore, even when grid points cluster severely, the approximation remains stable, and the residuals do not blow up.

4.2.2. Conditioning and stability

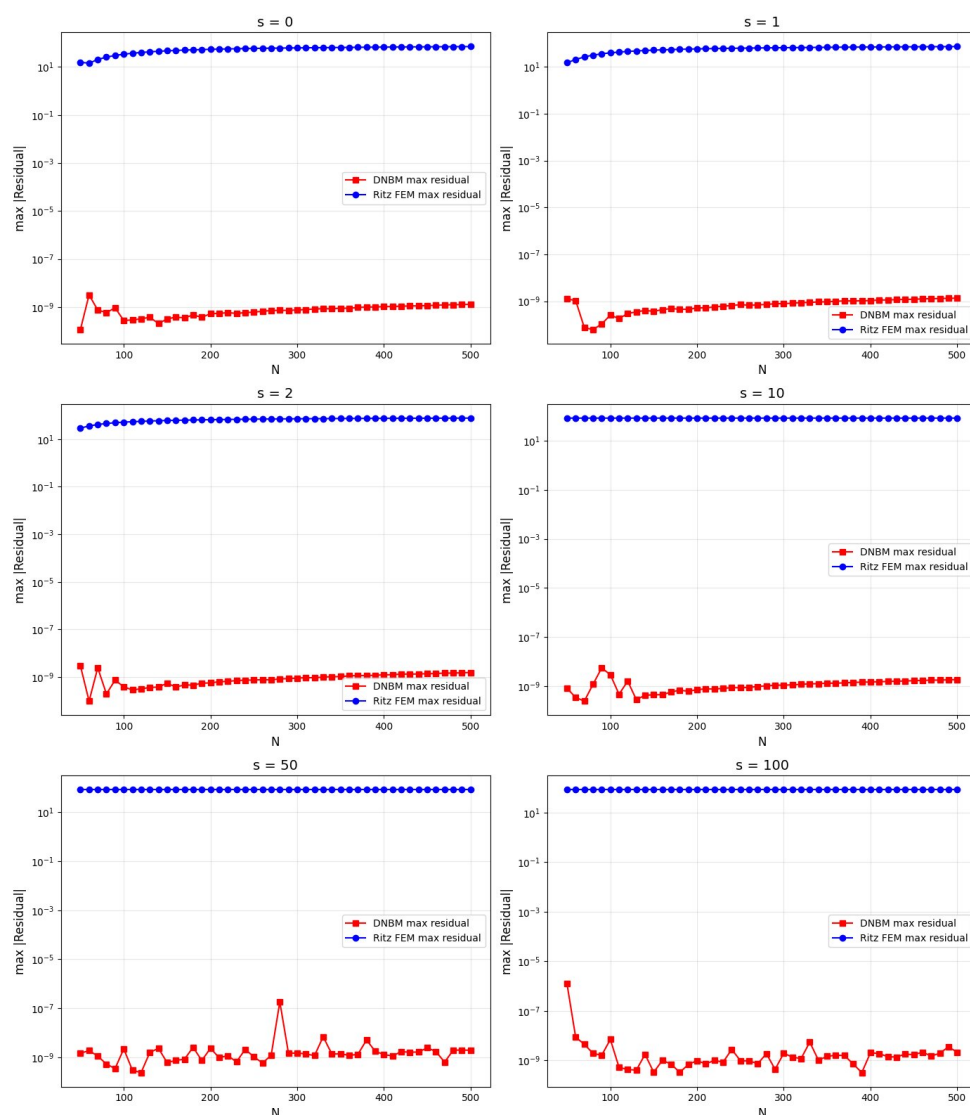


Figure 7. Stability of maximum residuals for DNBM (red squares) and Ritz FEM (blue circles) running from $N = 50, 60, 70, \dots, 500$ under different clustering strengths $s = 0, 1, 2, 10, 50, 100$. These numerical approaches solve the ODE problem: $xy''(x) + y'(x) + 2x^2y(x) = x^2$, in which the boundary conditions are $y(0.01) = 2$ and $y(4) = 5$. Please refer to the code at <https://github.com/raymingchen/ODE/blob/main/ConvergenceAnalysis.ipynb>.

The FEM linear system on a highly clustered mesh becomes ill-conditioned because the local stiffness matrix entries scale with $1/h$. As $h \rightarrow 0$, the condition number grows as $O(h^{-2})$, leading to numerical instability and large errors in the solution (and hence in the residual). This is why the residual jumped to 4.8×10^6 at the last points; the solution $\hat{y}$ itself may still be reasonable, but its derivative is computed from an ill-conditioned system. DNBM does not depend on mesh spacing for its derivatives; the derivatives are computed directly from the smooth basis functions. The linear system for DNBM (the R matrix) can also be ill-conditioned when the distance matrix D is nearly singular (e.g., for extreme clustering), but this can be mitigated by *regularization* (e.g., adding a small diagonal term) or by using pseudoinverses. In our case, the DNBM residuals remained stable and consistent with the exact solution, whereas the FEM residuals exploded. As for the stability test, one could refer to Figure 7. The experiment shows that both methods are fairly stable with respect to N under various clustering strengths s . We report the observed numerical behavior on uniform grids. For the ODE $x y''(x) + y'(x) + 2x^2 y(x) = x^2$, the maximum residual decreases from approximately 10^{-2} at $N = 50$ to 10^{-3} at $N = 500$. This empirical trend indicates that the DNBM exhibits stable improvement in accuracy as N increases.

4.2.3. Comparison with the exact solution

The true solution of the BVP

$$xy'' + y' + 2x^2y = x^2, \quad y(0.01) = 2, \quad y(4) = 5,$$

is smooth and well-behaved (it exists uniquely). The exact residual, by definition, is zero at all points. Any numerical method approximates this residual with an error that should be small and bounded independently of the mesh size, provided the method is consistent and stable.

FEM, due to its local nature, fails to deliver a bounded residual on the clustered mesh because the derivative approximation becomes unreliable. DNBM, with its global smooth basis, yields residuals that are uniformly small and do not blow up, even at the extreme boundary. This demonstrates superior stability of DNBM for problems with boundary layers or extreme mesh grading.

DNBM is not tied to a particular mesh topology; it works with an arbitrary set of points X . The distance matrix D captures the geometry of the points, and the interpolation is achieved by solving a linear system that is independent of mesh quality. This makes DNBM particularly attractive for:

- Problems where the solution exhibits boundary layers or steep gradients.
- *Adaptive refinement*: You can add more points near boundaries without harming the stability of the method.
- *Multivariate problems*: The distance-based interpolation naturally extends to higher dimensions, whereas FEM requires complex mesh generation.

Ritz FEM, when applied to a highly clustered grid, suffers from numerical instability due to the extreme variation in element sizes, leading to artificially large residuals near the boundary. In contrast, the proposed DNBM, with its globally smooth basis functions, remains stable and yields residuals that are uniformly small and consistent with the exact solution. This demonstrates the practical advantage of DNBM for problems requiring strong local refinement, such as boundary layer phenomena.

5. Conclusions

Differential equations play vital roles in modeling the real dynamical world and behaviors. Finding the numerical solutions for the implicit expression becomes of paramount importance. In this research, we combine the calculus of variations and distance-network to yield a DNBm. There are some characteristics of DNBm: First, this method keeps track of the global properties of the dynamical system, that is the change of output is influenced by the network. Second, it designs some weighted functions, such as ρ , W , V , and so on, to counteract the potential singularity of a distance matrix D and to serve as the regulators for enhancing the roles of distanced-network-based variables. Third, it offers high flexibility to extend the theory to partial differential equations or multivariate differential equations via the concept of distance functions. Because there are various numerical methods for finding the explicit solutions for the implicit expression, the ultimate evaluation or choice of them will mostly depend on their modeling capability for real data. The method we devise should be easily incorporated and enrich the trend of data mining or machine learning. However, there is still some room for improvement in future studies, such as the convergence analysis for DNBm [17], singularly perturbed BVPs for DNBm [18], the choice of other distance functions [19], and the combination of DNBm with other machine learning techniques [20].

Use of Generative-AI tools declaration

AI tools were used only for limited editorial assistance and code verification. Specifically, AI support was employed to check spelling and grammar consistency in the manuscript and to validate the syntactic correctness and basic functionality of Python code snippets. No AI-generated text, analysis, data, or scientific content contributed to the research results, interpretations, or conclusions. The author takes full responsibility for the accuracy, integrity, and originality of all content presented in this work.

Acknowledgments

The research is funded by the Internal (Faculty/Staff) Start-Up Research Grant of Wenzhou-Kean University (Project No. ISRG2023029) and the Wenzhou Science and Technology Bureau (Project No. GC20250023).

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. A. K. Singh, *Numerical methods for ordinary differential equations with programs*, Alpha Science International, 2018.
2. L. Ambrosio, N. Dancer, *Calculus of variations and partial differential equations*, Berlin: Springer, 2000. <https://doi.org/10.1007/978-3-642-57186-2>

3. M. Kamruzzaman, M. C. Nath, A comparative study on numerical solution of initial value problem by using Euler's method, modified Euler's method and Runge-Kutta method, *J. Comput. Math. Sci.*, **9** (2018), 493–500.
4. G. Corliss, Y. F. Chang, Solving ordinary differential equations using Taylor series, *ACM Trans. Math. Softw.*, **8** (1982), 114–144. <https://doi.org/10.1145/355993.355995>
5. W. Zhang, Y. Zhuang, L. Zhang, A new high-order finite volume method for 3D elastic wave simulation on unstructured meshes, *J. Comput. Phys.*, **340** (2017), 534–555.
6. J. C. Butcher, *The numerical analysis of ordinary differential equations: Runge–Kutta and general linear methods*, Wiley-Interscience, 1987.
7. J. C. Butcher, A history of Runge–Kutta methods, *Appl. Numer. Math.*, **20** (1996), 247–260. [https://doi.org/10.1016/0168-9274\(95\)00108-5](https://doi.org/10.1016/0168-9274(95)00108-5)
8. T. Monovasilis, Z. Kalogiratos, T. E. Simos, Trigonometrical fitting conditions for two derivative Runge-Kutta methods, *Numer. Algor.*, **79** (2018), 787–800. <https://doi.org/10.1007/s11075-017-0461-3>
9. F. Costabile, A. Napoli, A class of collocation methods for numerical integration of initial value problems, *Comput. Math. Appl.*, **62** (2011), 3221–3235. <https://doi.org/10.1016/j.camwa.2011.08.036>
10. Y. Wang, S. Chen, X. Wu, A rational spectral collocation method for solving a class of parameterized singular perturbation problems, *J. Comput. Appl. Math.*, **233** (2010), 2652–2660. <https://doi.org/10.1016/j.cam.2009.11.011>
11. R. J. LeVeque, *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*, SIAM, Philadelphia, 2007.
12. L. P. Lebedev, M. J. Cloud, *The calculus of variations and functional analysis with optimal control and applications in mechanics*, World Scientific, 2003. <https://doi.org/10.1142/5374>
13. E. Kreyszig, On the calculus of variations and its major influences on the mathematics of the first half of our century. Part II, *Am. Math. Mon.*, **101** (1994), 902–908. <https://doi.org/10.2307/2975142>
14. B. S. Sarma, T. K. Varadan, Ritz finite element approach to nonlinear vibrations of beams, *Int. J. Numer. Methods Eng.*, **20** (1984), 353–367. <https://doi.org/10.1002/nme.1620200213>
15. S. N. Jator, J. Li, Boundary value methods via a multistep method with variable coefficients for second order initial and boundary value problems, *Int. J. Pure Appl. Math.*, **50** (2009), 403–420.
16. Y. Workineh, H. Mekonnen, B. Belew, Numerical methods for solving second-order initial value problems of ordinary differential equations with Euler and Runge-Kutta fourth-order methods, *Front. Appl. Math. Stat.*, **10** (2024), 1360628. <https://doi.org/10.3389/fams.2024.1360628>
17. Z. Liu, Q. Xu, L^2 error estimates of unsymmetric RBF collocation for second order elliptic boundary value problems, *Results Appl. Math.*, **23** (2024), 100495. <https://doi.org/10.1016/j.rinam.2024.100495>
18. I. Ali, S. Haq, R. Ullah, S. U. Arifeen, Approximate solution of second order singular perturbed and obstacle boundary value problems using meshless method based on radial basis functions, *J. Nonlinear Math. Phys.*, **30** (2023), 215–234. <https://doi.org/10.1007/s44198-022-00080-7>

19. M. Bucelli, F. Regazzoni, L. Dede', A. Quarteroni, Robust radial basis function interpolation based on geodesic distance for the numerical coupling of multiphysics problems, *SIAM J. Sci. Comput.*, **46** (2024), B981–B1002. <https://doi.org/10.1137/24M1643888>
20. C. Y. Liu, C. Y. Ku, A novel ANN-based radial basis function collocation method for solving elliptic boundary value problems, *Mathematics*, **11** (2023), 3935. <https://doi.org/10.3390/math11183935>

Appendix

The procedures of the Ritz finite element method (FEM) go as follows:

- (1) Input a second order BVP (1.1) with given $X = \{x_0, x_1, \dots, x_N, x_{N+1}\}$;
- (2) Associate a set of functions $\{\hat{y}_j : [x_j, x_{j+1}] \rightarrow \mathbb{R}\}_{j=0}^{N+1}$;
- (3) Define a set of functional $\{\Phi_{x_j, x_{j+1}}^{L_j}(y_j) := \int_{x_j}^{x_{j+1}} L_j(x, y_j, y'_j) dx\}_{j=0}^N$ (2.7), where $\{L_j\}_{j=0}^N$ is to be determined;
- (4) The explicit form of each L_j (2.6), $L_j(x, \hat{y}_j, \hat{y}'_j) := e^{\int \frac{c_2(x)}{c_1(x)} dx}$.

$$\left\{ [y'_j(x; \hat{y}_j, \hat{y}_{j+1})]^2 - \frac{c_3(x)}{c_1(x)} \cdot [y_j(x; \hat{y}_j, \hat{y}_{j+1})]^2 + 2 \cdot \frac{c_4(x)}{c_1(x)} \cdot \hat{y}_j(x; \hat{y}_j, \hat{y}_{j+1}) \right\},$$

is identified based on the Euler–Lagrange differential equation;

- (5) With the explicit form of each L_j , a necessary condition for finding optimal $\{\hat{y}_j\}_{j=0}^{N+1}$ is revealed:

$$\int_{x_j}^{x_{j+1}} \frac{\partial}{\partial \hat{y}_k} L_j(x, \hat{y}_j(x; \hat{y}_j, \hat{y}_{j+1}), \hat{y}'_j(x; \hat{y}_j, \hat{y}_{j+1})) dx = 0,$$

and calculated (refer to Lemma 1): $\int_{x_j}^{x_{j+1}} e^{\int \frac{c_2(x)}{c_1(x)} dx}$.

$$\left\{ \hat{y}'_j(x; \hat{y}_j, \hat{y}_{j+1}) \cdot \frac{\partial \hat{y}'_j(x; \hat{y}_j, \hat{y}_{j+1})}{\partial \hat{y}_j} + \left[\frac{c_4(x) - c_3(x) \cdot \hat{y}_j(x; \hat{y}_j, \hat{y}_{j+1})}{c_1(x)} \right] \cdot \frac{\partial \hat{y}_j(x; \hat{y}_j, \hat{y}_{j+1})}{\partial \hat{y}_j} \right\} = 0$$

where

$$\hat{y}_j(x; \hat{y}_j, \hat{y}_{j+1}) = \omega^-(x) \cdot \hat{y}(x_{j+1}) + \omega^+(x) \cdot \hat{y}(x_j);$$

$$\hat{y}'_j(x; \hat{y}_j, \hat{y}_{j+1}) \equiv \frac{d}{dx} \hat{y}_j(x; \hat{y}_j, \hat{y}_{j+1}) = \frac{\hat{y}_{j+1} - \hat{y}_j}{x_{j+1} - x_j};$$

$$\frac{\partial}{\partial \hat{y}_j} \hat{y}'_j(x; \hat{y}_j, \hat{y}_{j+1}) = \frac{-1}{x_{j+1} - x_j}; \quad \frac{\partial}{\partial \hat{y}_{j+1}} \hat{y}'_j(x; \hat{y}_j, \hat{y}_{j+1}) = \frac{1}{x_{j+1} - x_j};$$

$$\frac{\partial}{\partial \hat{y}_j} \hat{y}_j(x; \hat{y}_j, \hat{y}_{j+1}) = \omega^+(x) = \frac{x_{j+1} - x}{x_{j+1} - x_j}; \quad \frac{\partial}{\partial \hat{y}_{j+1}} \hat{y}_j(x; \hat{y}_j, \hat{y}_{j+1}) = \omega^-(x) = \frac{x - x_j}{x_{j+1} - x_j}.$$

By simplifying the above terms, one has the following result:

$$\begin{aligned}
& \left\{ \int_{x_j}^{x_{j+1}} e^{\int \frac{c_2(x)}{c_1(x)} dx} \left[1 - \frac{c_3(x)}{c_1(x)} \cdot (x - x_{j+1})^2 \right] \right\} \cdot \hat{y}_j \\
& + \left\{ \int_{x_j}^{x_{j+1}} e^{\int \frac{c_2(x)}{c_1(x)} dx} \left[-1 + \frac{c_3(x)}{c_1(x)} \cdot (x - x_j) \cdot (x - x_{j+1}) \right] \right\} \cdot \hat{y}_{j+1} \\
& = -(x_{j+1} - x_j)^2 \cdot \int_{x_j}^{x_{j+1}} \left[e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \frac{c_4(x)}{c_1(x)} \cdot w^+(x) \right] dx \\
& = (x_{j+1} - x_j) \cdot \int_{x_j}^{x_{j+1}} \left[e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \frac{c_4(x)}{c_1(x)} \cdot (x - x_{j+1}) \right] dx \\
& \left\{ \int_{x_j}^{x_{j+1}} e^{\int \frac{c_2(x)}{c_1(x)} dx} \left[-1 + \frac{c_3(x)}{c_1(x)} \cdot (x - x_j) \cdot (x - x_{j+1}) \right] \right\} \cdot \hat{y}_j \\
& + \left\{ \int_{x_j}^{x_{j+1}} e^{\int \frac{c_2(x)}{c_1(x)} dx} \left[1 - \frac{c_3(x)}{c_1(x)} \cdot (x - x_j)^2 \right] \right\} \cdot \hat{y}_{j+1} \\
& = -(x_{j+1} - x_j)^2 \cdot \int_{x_j}^{x_{j+1}} e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \frac{c_4(x)}{c_1(x)} \cdot w^-(x) dx \\
& = -(x_{j+1} - x_j) \cdot \int_{x_j}^{x_{j+1}} \left[e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \frac{c_4(x)}{c_1(x)} \cdot (x - x_j) \right] dx
\end{aligned}$$

i.e.,

$$\begin{bmatrix} p_j - q_j^{11} & -p_j + q_j^{01} \\ -p_j + q_j^{01} & p_j - q_j^{00} \end{bmatrix} \begin{bmatrix} \hat{y}_j \\ \hat{y}_{j+1} \end{bmatrix} = \begin{bmatrix} v_j^1 \\ -v_j^0 \end{bmatrix}, \quad (\text{A.1})$$

where

$$p_j = \int_{x_j}^{x_{j+1}} e^{\int \frac{c_2(x)}{c_1(x)} dx} dx; \quad (\text{A.2})$$

$$q_j^{st} = \int_{x_j}^{x_{j+1}} e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \frac{c_3(x)}{c_1(x)} \cdot (x - x_{j+s}) \cdot (x - x_{j+t}) dx; \quad (\text{A.3})$$

$$v_j^s = (x_{j+1} - x_j) \cdot \int_{x_j}^{x_{j+1}} e^{\int \frac{c_2(x)}{c_1(x)} dx} \cdot \frac{c_4(x)}{c_1(x)} \cdot (x - x_{j+s}) dx \quad (\text{A.4})$$

for all $s, t \in \{0, 1\}$. Furthermore, by the next finite element, one also has (by replacing j with $j + 1$ in Eq (A.1)):

$$\begin{bmatrix} p_{j+1} - q_{j+1}^{11} & -p_{j+1} + q_{j+1}^{01} \\ -p_{j+1} + q_{j+1}^{01} & p_{j+1} - q_{j+1}^{00} \end{bmatrix} \begin{bmatrix} \hat{y}_{j+1} \\ \hat{y}_{j+2} \end{bmatrix} = \begin{bmatrix} v_{j+1}^1 \\ -v_{j+1}^0 \end{bmatrix}. \quad (\text{A.5})$$

(1) Associate a parameterized form of $\hat{y}_j(x)$, or $\hat{y}_j(x; \hat{y}_j, \hat{y}_{j+1}) = \omega^-(x) \cdot \hat{y}_{j+1} + \omega^+(x) \cdot \hat{y}_j$. Then,

- Find whether D_ρ is invertible or not (1.4);
- If D_ρ is invertible, then calculate x_D (1.4); if D_ρ is not invertible, then add a weighted matrix (2.14), whose (i, j) -th element is calculated by

$$w_{ij} = \frac{e^{-d(x_i, x_j)}}{\sum_{j=1}^N e^{-d(x_i, x_j)}},$$

to counteract the singularity. In our real experiment, we adopt another method: The Moore–Penrose pseudoinverse $(D_\rho^\dagger)$ is used in constructing the basis functions when (D_ρ) is singular.

- (2) Calculate $\hat{y}(x_i)$ and $\hat{y}(x)$: The Moore–Penrose pseudoinverse $(D_\rho^\dagger)$ is used in constructing the basis functions when (D_ρ) is singular. The resulting variational system is then solved using the variational matrix $(\widetilde{R})$, with appropriate numerical regularization when necessary. The weighted formulation in Theorem 3 provides an alternative treatment of possible singularity of the variational matrix and is not used in the numerical experiments presented in this paper.
- (3) The explicit form of $\hat{y}(x) = \hat{Y}_X \bullet x_{D_\rho}$ is revealed.



AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)