International Journal of Informatics and Applied Mathematics
e-ISSN:2667-6990 Vol. 7, No. 1, 57-72

Nonlinear Reduced Order Modelling for Korteweg-de Vries Equation

Yusuf Çakır and Murat Uzunca

Sinop University, 57100 Sinop, Türkiye.
{ycakir,muzunca}@sinop.edu.tr

Abstract. Efficient computational techniques that maintain the accuracy and invariant preservation property of the Korteweg-de Vries (KdV) equations have been studied by a wide range of researchers. In this paper, we introduce a reduced order model technique utilizing kernel principle component analysis (KPCA), a nonlinear version of the classical principle component analysis, in a non-intrusive way. The KPCA is applied to the data matrix, which is formed by the discrete solution vectors of KdV equation. In order to obtain the discrete solutions, the finite differences are used for spatial discretization, and linearly implicit Kahan's method for the temporal one. The back-mapping from the reduced dimensional space, is handled by a non-iterative formula based on the idea of multidimensional scaling (MDS) method. Through KPCA, we illustrate that the reduced order approximations conserve the invariants, i.e., Hamiltonian, momentum and mass structure of the KdV equation. The accuracy of reduced solutions, conservation of invariants, and computational speed enhancements facilitated by classical (linear) PCA and KPCA are exemplified through one-dimensional KdV equation.

Keywords: Kernel Principle Component Analysis · Multi Dimensional Scaling · Energy Preservation.

Received: 19 March 2024 | Revised: 07 May 2024 | Accepted: 16 May 2024
DOI: 10.53508/ijiam.1455321

1 Introduction

Most of the physical phenomena of real life problems in several sciences are mathematically modeled by differential equations, especially by partial differential equations (PDEs). Among them, the Korteweg-de Vries (KdV) equation is a nonlinear partial differential equation that describes the propagation of waves in certain physical systems. It was first derived by physicists Diederik Korteweg and Gustav de Vries in 1895 to model shallow water waves in a canal. The KdV equation is widely studied in the field of nonlinear dynamics and has applications in various branches of physics, including fluid dynamics, plasma physics, and nonlinear optics. On a one dimensional spatial interval, the KdV equation is given by

$$\begin{aligned}
&\frac{\partial y}{\partial t} = -\alpha y \frac{\partial y}{\partial x} - \mu \frac{\partial^3 y}{\partial x^3}, \quad x \in (a,b),\ t \in (0,T], \\
&y(x,0) = u^0(x), \qquad\qquad\qquad\qquad x \in [a,b],
\end{aligned} \tag{1}$$

with real parameters α and μ, and for a positive terminal time $T > 0$. The function $u^0(x)$ is a prescribed initial wave and, in general case, periodic boundary conditions are to be satisfied. The KdV equation is an integrable Hamiltonian PDE with a constant Poisson structure

$$\frac{\partial y}{\partial t} = \mathcal{S} \frac{\delta \mathcal{H}}{\delta y}, \tag{2}$$

where δ denotes the variational derivative, $\mathcal{S}$ is the constant Poisson tensor, and $\mathcal{H}$ is the Hamiltonian (energy) functional, given by

$$\mathcal{S} = \frac{\partial}{\partial x}, \quad \mathcal{H}(y) = \int_a^b \left(-\frac{\alpha}{6} y^3 + \frac{\mu}{2} \left(\frac{\partial y}{\partial x} \right)^2 \right) dx.$$

The cubic Hamiltonian functional $\mathcal{H}$ is the major conserved quantity of the KdV equation. On the other hand, the KdV equation (1) is completely integrable, meaning that it has infinitely many invariants to be preserved. Here, we consider the most frequently used invariants: the quadratic momentum and the linear mass, given respectively by

$$\mathcal{I}_1(y) = \int_a^b y^2 dx, \quad \mathcal{I}_2(y) = \int_a^b y dx.$$

Numerical solutions for large-scale dynamic systems are demanding in terms of computational time and often necessitate extensive computer memory, particularly for real-time applications. Hamiltonian systems, like the KdV equation, belong to a category that conserves the system's energy when solved accurately to machine precision. However, solving Hamiltonian PDEs can be exceedingly time-consuming, especially in spatial dimensions of two and three. Consequently, reduced order modeling (ROM) has gained significant importance as they allow for the resolution of such challenges with reduced computational data. There are not many documents addressing the ROM of the KdV equation. ROMs are formulated using Lax pairs in [1], and a greedy proper orthogonal decomposition (POD) algorithm is devised in [2], incorporating the discrete empirical interpolation method (DEIM) rooted in the Poisson structure. To maintain the first integrals of the KdV equation, structure-preserving POD and DEIM methods are established in [3], and in [4] it is extended to one-dimensional conservative PDEs in Wasserstein space to construct ROMs that encompass the KdV equation. In [5], structure-preserving POD is used for the Hamiltonian PDEs in a tensorial framework, where the authors take into account the linear-quadratic structure of the arising dynamical system. All the aforementioned works use the ROM techniques in an intrusive way, i.e., a projection-based reduced dimensional system of equations has to be solved. In contrast, in this paper, utilizing mainly the principle component analysis (PCA), we consider a non-intrusive (data-driven) version of ROM, which only encounters a data set and do not need to solve a reduced system of equations.

The PCA is a powerful statistical technique used for dimensionality reduction and data visualization in various fields such as image processing, genetics, finance, and social sciences to simplify complex data sets, identify patterns, and facilitate interpretation [6,7]. It works by transforming the original variables into a new set of uncorrelated variables, called principal components, which capture the maximum variance in the data. PCA aims to find a lower-dimensional representation of the data while preserving as much of the original variability as possible. Besides, in the PCA, data belonging to different targets must be distributed with a very large variance, and when the distribution has very small variance, the performance of PCA decreases. In such a case, PCA cannot perform subspace decomposition with a linear transition. Recently, as an extension of PCA, kernel principal component analysis (KPCA) has become a favorable technique in place of classical PCA for dimensionality reduction. The KPCA allows for nonlinear dimensionality reduction and feature extraction in high-dimensional data spaces. Unlike the PCA, the KPCA employs a kernel function to implicitly map the input data into a higher-dimensional feature space where nonlinear relationships can be better captured. However, KPCA has a drawback when finding the pre-image of the projection in the feature space. A variety of approaches exist in the literature, among them, the multidimensional scaling (MDS) is a favorable method [8,9,10]. By MDS, one can iteratively compute the pre-image utilizing some distance metrics of the data vectors. In this paper, following the idea in [9], we apply a non-iterative version of the MDS in order to find the pre-image vectors in the feature space. The KPCA is particularly useful when the underlying structure of the data is nonlinear or when linear methods fail to adequately represent the data distribution. Thus, KPCA serves as a powerful tool for exploratory data analysis, pattern recognition, and manifold learning in various domains, including computer vision, bio-informatics, and signal processing [11,12,9,13,14,15].

The main focus of this paper is the application of the KPCA to the data set obtained by discrete solution vectors of the KdV equation (1). To the best of our knowledge, the utilization of KPCA in ROM has not been previously employed for the KdV equation, maintaining both solution accuracy and the preservation of the invariant conservation structure. The paper is organized as follows: in the next Section, we introduce shortly the mathematical formulation of full discrete system of the KdV equation (1), i.e., spatial and temporal discretization methods. In Section 3, the process of KPCA applied to the data set obtained by the discrete solution vectors of the KdV equation, is explained. The accuracy of reduced solutions, conservation of invariants, and computational speed enhancements are demonstrated through the one-dimensional KdV equation in Section 4. The paper ends with some conclusion.

2 Discrete data set of KdV equation

In this section, we give the full discrete formulation of the KdV equation over its Hamiltonian form (2), whose solution vectors are used for formation of the data set to be considered in the sequel. We begin by the spatial discretization of the KdV equation. To this end, we partition the interval $[a, b]$ into m equidistant sub-intervals of length $\Delta x = (b - a)/m$, forming the discrete grid nodes $x_i = a + (i - 1)\Delta x$, $i = 1, \dots, m + 1$. On this mesh, we define time-dependent semi-discrete solution vectors $\boldsymbol{y}(t) : [0, T] \mapsto \mathbb{R}^m$ as

$$\boldsymbol{y}(t) = (y_1(t), \dots, y_m(t))^T,$$

where $y_i(t) = y(x_i, t)$, and we omit the solution at the right boundary node x_{m+1} because of the periodic boundary condition $y(x_1, t) = y(x_{m+1}, t)$. For the first and second order derivative terms, we use the centered finite difference approximations, which yields the m-dimensional semi-discrete (dynamical) system

$$\dot{\boldsymbol{y}} = D_1 \nabla H(\boldsymbol{y}),$$

or, inserting the discrete gradient $\nabla H(\boldsymbol{y}) = -\mu D_2 \boldsymbol{y} - (\alpha/2)\boldsymbol{y}^2$, we get the explicit form

$$\dot{\boldsymbol{y}} = -\mu D_3 \boldsymbol{y} - \frac{\alpha}{2} D_1 \boldsymbol{y}^2, \tag{3}$$

where $D_3 = D_1 D_2$, with $D_1 \in \mathbb{R}^{m\times m}$ and $D_2 \in \mathbb{R}^{m\times m}$ denoting the matrices of first order and second order finite difference operators, respectively, under periodic boundary

$$D_1 := \frac{1}{2\Delta x}\begin{pmatrix} 0 & 1 & & & -1 \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ 1 & & & -1 & 0 \end{pmatrix}, \quad D_2 := \frac{1}{\Delta x^2}\begin{pmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{pmatrix}.$$

We note that the square operation in (3) hold component-wise, i.e., $\boldsymbol{y}^2 \in \mathbb{R}^m$ and $(\boldsymbol{y}^2)_i = y_i^2$, $i = 1, \ldots, m$.

The full discrete system of the KdV equation is obtained by application of a temporal integration method to the semi-discrete system (3). Here, we apply Kahan's method [16,17] which is a second order, time-reversal, and linearly implicit method for ODEs with quadratic vector fields [18], like the semi-discrete KdV equation (3). Divide the temporal interval $[0, T]$ into $n-1$ equidistant steps with the time step-size $\Delta t = T/(n-1)$ to form the discrete time instances $t_j = j\Delta t$, $j = 0, \ldots, n-1$. We denote by $\boldsymbol{y}^j = \boldsymbol{y}(t_j) \in \mathbb{R}^m$ the solution vector at time t_j. Then, application of the Kahan's method results in the full discrete problem: given $\boldsymbol{y}^0 \in \mathbb{R}^m$ with $(\boldsymbol{y}^0)_i = u^0(x_i)$, find $\boldsymbol{y}^j \in \mathbb{R}^m$ from the system

$$\frac{\boldsymbol{y}^{j+1} - \boldsymbol{y}^j}{\Delta t} = -\frac{\mu}{2} D_3(\boldsymbol{y}^j + \boldsymbol{y}^{j+1}) - \frac{\alpha}{2} D_1 \boldsymbol{y}^j \boldsymbol{y}^{j+1}, \qquad j = 0, \ldots, n-1. \tag{4}$$

Under the given discretization setting, the discrete Hamiltonian $H(t_j)$, the discrete momentum $I_1(t_j)$, and the discrete mass $I_2(t_j)$ at time t_j are defined by

$$\begin{aligned} H(t_j) &= \Delta x \sum_{i=1}^{m} \left(-\frac{\alpha}{6} y_i^3(t_j) + \frac{\mu}{2} \left(\frac{y_{i+1}(t_j) - y_i(t_j)}{\Delta x} \right)^2 \right), \\ I_1(t_j) &= \Delta x \sum_{i=1}^{m} y_i^2(t_j) \,, \qquad I_2(t_j) = \Delta x \sum_{i=1}^{m} y_i(t_j). \end{aligned}$$

3 Reduced order Model

In this section, we discuss on the linear/nonlinear ROM formulation for the KdV equation. Firstly, we give shortly the traditional PCA approach, then we explain the KPCA which is the nonlinear version of PCA. In either case, the procedure is two folds: beginning with the vectors from the column space of the data matrix, referred to as the input space, transitioning to the reduced space through projection, and then reversing the mapping from the reduced space back to the input space to obtain the reduced approximation. Since the ROM technique we consider is data-driven, we consider in the upcoming sections the data matrix

$$Y = [\boldsymbol{y}^1 \boldsymbol{y}^2 \cdots \boldsymbol{y}^n] \in \mathbb{R}^{m \times n}, \tag{5}$$

where the columns of the data matrix are the solution vectors of the KdV equation obtained by the full discrete system (4). We note that, for easy notation, we start the starting superscript of the snapshot vectors in (5) from 1 in place of 0, i.e., the column vector $\boldsymbol{y}^i \in \mathbb{R}^m$ in (5) is the solution vector $\boldsymbol{y}^{i-1} \in \mathbb{R}^m$ of the KdV equation obtained by the system (4), $i = 1, \ldots, n$.

3.1 Linear dimension reduction (PCA)

For a data matrix $Y = [\boldsymbol{y}^1 \boldsymbol{y}^2 \cdots \boldsymbol{y}^n] \in \mathbb{R}^{m\times n}$, like the matrix (5) of the solution vectors for KdV equation, PCA seeks to identify a linear model of some dimension $k \ll m$ that effectively captures the variability present in the vectors $\boldsymbol{y}^i$. Assume that the matrix Y has zero column sum, otherwise it can be done by subtracting the mean $\bar{y} = (1/n)\sum \boldsymbol{y}^i$ of the columns from each column. The covariance matrix of Y is defined by

$$C = \sum_{i=1}^{n} \boldsymbol{y}^i (\boldsymbol{y}^i)^T = YY^T \in \mathbb{R}^{m\times m}.$$

The diagonalization of the covariance matrix reads as

$$C = U \Lambda U^T,$$

where the column vectors of the orthogonal matrix $U = [\boldsymbol{u}^1 \boldsymbol{u}^2 \cdots \boldsymbol{u}^m] \in \mathbb{R}^{m\times m}$ are the eigenvectors corresponding to the eigenvalues $\lambda_1 > \ldots > \lambda_m$ located at the diagonal entries of the diagonal matrix $\Lambda \in \mathbb{R}^{m\times m}$. Then, in its simplest way, the basis of the k-dimensional linear subspace is taken as the eigenvectors $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k\} \subset \mathbb{R}^m$ of the covariance matrix C corresponding to the k largest eigenvalues $\lambda_1 > \ldots > \lambda_k$ [6,7,15]. Finally, any arbitrary vector $\boldsymbol{y}^* \in \mathbb{R}^m$ from the input space can be approximately represented by the pre-image $\widehat{\boldsymbol{y}}^* \in \mathbb{R}^m$ as the linear combination of the eigenvectors

$$\boldsymbol{y}^* \approx \widehat{\boldsymbol{y}}^* = \sum_{i=1}^{k} z_i^* \boldsymbol{u}_i = U_k \boldsymbol{z}^*, \tag{6}$$

where the matrix $U_k = [\boldsymbol{u}^1 \boldsymbol{u}^2 \cdots \boldsymbol{u}^k] \in \mathbb{R}^{m\times k}$ consists of the first k columns of the orthogonal matrix U. The coefficients $z_i^* = (\boldsymbol{y}^*)^T \boldsymbol{u}_i$ are the entries of the projection vector $\boldsymbol{z}^* = [z_1^*, z_2^*, \ldots z_k^*]^T \in \mathbb{R}^k$ of $\boldsymbol{y}^*$ onto the reduced linear subspace, in other words, the $k \ll m$ dimensional vector $\boldsymbol{z}^* = U_k^T \boldsymbol{y}^*$ lies in the reduced space, which is the projection of the m dimensional vector $\boldsymbol{y}^*$ from the input space.

3.2 Nonlinear dimension reduction (KPCA)

The PCA is limited to linear dimensionality reduction. Yet, when dealing with data characterized by complex structures that cannot be effectively captured within a linear subspace, standard PCA may be ineffective. However, KPCA offers a solution by enabling us to extend the linear PCA to nonlinear dimensionality reduction [11,12,9,14,15].

Forward map By KPCA, the vectors from the m dimensional input space are first transformed into a $M \gg m$ dimensional (possibly infinite dimensional) space, named feature space, through an arbitrary nonlinear map $\Phi(\cdot) : \mathbb{R}^m \mapsto$

$\mathbb{R}^M$, and then traditional PCA is applied to the vectors in this feature space. To this end, let us denote by $\widetilde{Y}$ the transformed data matrix defined by

$$\widetilde{Y} = [\Phi(\boldsymbol{y}^1)\Phi(\boldsymbol{y}^2)\cdots\Phi(\boldsymbol{y}^n)] \in \mathbb{R}^{M\times n},$$

where the columns are the transformed vectors $\Phi(\boldsymbol{y}^l) \in \mathbb{R}^M$ related to the input space vectors $\boldsymbol{y}^l$. For the arbitrary map $\Phi(\cdot)$, the matrix $\widetilde{Y}$, usually, does not have zero column sum which is a necessity for the PCA process. We may obtain a data matrix with zero column sum by simply subtracting the mean $\bar{\Phi}(\boldsymbol{y}) = (1/n)\sum \Phi(\boldsymbol{y}^i)$ of the columns from each column, yielding the following data matrix with zero column sum

$$\widetilde{\widetilde{Y}} = [\widetilde{\Phi}(\boldsymbol{y}^1)\widetilde{\Phi}(\boldsymbol{y}^2)\cdots\widetilde{\Phi}(\boldsymbol{y}^n)] = \widetilde{Y}H \in \mathbb{R}^{M\times n},$$

where the columns are given by $\widetilde{\Phi}(\boldsymbol{y}^l) = \Phi(\boldsymbol{y}^l) - \bar{\Phi}(\boldsymbol{y})$, and H is called the centering matrix given by

$$H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T \in \mathbb{R}^{n\times n}, \tag{7}$$

with I is the n-dimensional identity matrix and $\mathbf{1} = (1,\ldots,1)^T \in \mathbb{R}^n$ is the vector of ones. Then, it follows the traditional PCA procedure explained in the previous section for the data matrix $\widetilde{\widetilde{Y}}$ whose columns span the feature space, i.e., we need to find the eigenvectors of the covariance matrix

$$\widetilde{C} = \sum_{l=1}^{n} \widetilde{\Phi}(\boldsymbol{y}^l)\widetilde{\Phi}(\boldsymbol{y}^l)^T = \widetilde{\widetilde{Y}}\widetilde{\widetilde{Y}}^T \in \mathbb{R}^{M\times M}. \tag{8}$$

At this point, the KPCA has two serious drawbacks. Firstly, the dimension M may be a very large number that makes almost impossible to compute the eigenvectors of the M-dimensional covariance matrix $\widetilde{C}$. Second, the nonlinear map $\Phi(\cdot)$ is arbitrary and that most of the time it is not available. To overcome all these issues, a kernel trick is applied [14,15]. In order to understand this trick, let consider the eigenvalue problem

$$\widetilde{C}\boldsymbol{v}^i = \widetilde{\lambda}_i\boldsymbol{v}^i, \qquad i = 1,\ldots,M, \tag{9}$$

where $\{\widetilde{\lambda}_i, \boldsymbol{v}^i\}$ are the eigen-pair of the covariance matrix $\widetilde{C}$. Note that by definition the eigenvectors span the feature space, i.e., the column space of the transformed data matrix $\widetilde{\widetilde{Y}} = [\widetilde{\Phi}(\boldsymbol{y}^1)\widetilde{\Phi}(\boldsymbol{y}^2)\cdots\widetilde{\Phi}(\boldsymbol{y}^n)]$. As a result, for each eigenvector $\boldsymbol{v}^i$, there exist some coefficients a_{ij} satisfying the linear combination

$$\boldsymbol{v}^i = \sum_{j=1}^{n} a_{ij}\widetilde{\Phi}(\boldsymbol{y}^j), \qquad i = 1,\ldots,M. \tag{10}$$

Now, substituting the relation (10) together with the identity (8) into the equation (9), we obtain that

$$\sum_{l=1}^{n} \widetilde{\Phi}(\boldsymbol{y}^l) \sum_{j=1}^{n} a_{ij} \widetilde{\Phi}(\boldsymbol{y}^l)^T \widetilde{\Phi}(\boldsymbol{y}^j) = \widetilde{\lambda}_i \sum_{j=1}^{n} a_{ij} \widetilde{\Phi}(\boldsymbol{y}^j),$$

or, since all the eigenvectors lie in the span of the transformed vectors $\{\widetilde{\Phi}(\boldsymbol{y}^s)\}_{s=1}^n$, we can consider for $s = 1, \ldots, n$ the equivalent equations after projection onto the vectors $\widetilde{\Phi}(\boldsymbol{y}^s)$, yielding

$$\sum_{l=1}^{n} \widetilde{\Phi}(\boldsymbol{y}^s)^T \widetilde{\Phi}(\boldsymbol{y}^l) \sum_{j=1}^{n} a_{ij} \widetilde{\Phi}(\boldsymbol{y}^l)^T \widetilde{\Phi}(\boldsymbol{y}^j) = \widetilde{\lambda}_i \sum_{j=1}^{n} a_{ij} \widetilde{\Phi}(\boldsymbol{y}^s)^T \widetilde{\Phi}(\boldsymbol{y}^j). \tag{11}$$

At this point, a kernel function $\kappa(\cdot,\cdot) : \mathbb{R}^m \times \mathbb{R}^m \mapsto \mathbb{R}$ which aims to represent the Euclidean inner products $\Phi(\cdot)^T \Phi(\cdot)$ of non-centered transformed vectors in the feature space in terms of the input space vectors, is defined as follows

$$\kappa(\boldsymbol{y}^s, \boldsymbol{y}^l) = \Phi(\boldsymbol{y}^s)^T \Phi(\boldsymbol{y}^l), \qquad\qquad s, l = 1, \ldots, n.$$

In order to represent the Euclidean inner products $\widetilde{\Phi}(\cdot)^T \widetilde{\Phi}(\cdot)$ of centered transformed vectors in the feature space, we use the setting [9]

$$\tilde{\kappa}(\boldsymbol{y}^s, \boldsymbol{y}^l) = \kappa(\boldsymbol{y}^s, \boldsymbol{y}^l) - \frac{1}{n} \mathbf{1}^T \boldsymbol{k}_{\boldsymbol{y}^s} - \frac{1}{n} \mathbf{1}^T \boldsymbol{k}_{\boldsymbol{y}^l} + \frac{1}{n^2} \mathbf{1}^T K \mathbf{1},$$

where the kernel matrix $K \in \mathbb{R}^{n \times n}$ and the vector $\boldsymbol{k}_{\boldsymbol{y}} \in \mathbb{R}^n$ are defined respectively by

$$K_{ij} = \kappa(\boldsymbol{y}^i, \boldsymbol{y}^j), \qquad \boldsymbol{k}_{\boldsymbol{y}} = (\kappa(\boldsymbol{y}, \boldsymbol{y}^1), \ldots, \kappa(\boldsymbol{y}, \boldsymbol{y}^n))^T.$$

We note that the vectors $\boldsymbol{k}_{\boldsymbol{y}^i}$ need not be calculated, as they are nothing but the ith columns of the symmetric kernel matrix K. Then, we get an equivalent equation to (11) as

$$\sum_{l=1}^{n} \tilde{\kappa}(\boldsymbol{y}^s, \boldsymbol{y}^l) \sum_{j=1}^{n} a_{ij} \tilde{\kappa}(\boldsymbol{y}^l, \boldsymbol{y}^j) = \widetilde{\lambda}_i \sum_{j=1}^{n} a_{ij} \tilde{\kappa}(\boldsymbol{y}^s, \boldsymbol{y}^j). \tag{12}$$

There are a variety of kernel functions used in the literature, such as linear, polynomial and Gaussian kernels [13,14]. Here, we use the Gaussian kernel given by

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{2\sigma^2}\right),$$

where $\|\cdot\|$ denotes the standard Euclidean norm, and σ is a parameter. For the coefficient vector $\boldsymbol{a}^i = (a_{i1}, \ldots, a_{in})^T \in \mathbb{R}^n$, the equation (12) can be written in the following matrix-vector form

$$\widetilde{K}^2\boldsymbol{a}^i = \widetilde{\lambda}_i\widetilde{K}\boldsymbol{a}^i \qquad \text{or} \qquad \widetilde{K}\boldsymbol{a}^i = \widetilde{\lambda}_i\boldsymbol{a}^i, \qquad\qquad i = 1,\ldots,M, \tag{13}$$

where $\widetilde{K} = HKH$, with the centering matrix H in (7). Finally, for an arbitrary vector $\boldsymbol{y}^* \in \mathbb{R}^m$ from the input space, with its centered nonlinear transformed vector $\widetilde{\Phi}(\boldsymbol{y}^*) \in \mathbb{R}^M$ in the feature space, let $\boldsymbol{z}^* = [z_1^*, z_2^*, \ldots z_k^*]^T \in \mathbb{R}^k$ denotes the projection vector of $\widetilde{\Phi}(\boldsymbol{y}^*) \in \mathbb{R}^M$ onto the k dimensional ($k \ll m \ll M$) reduced space spanned by the eigenvectors $\{\boldsymbol{v}^1, \ldots, \boldsymbol{v}^k\}$ corresponding to the first k largest eigenvalues $\widetilde{\lambda}_i$, $i = 1,\ldots,k$. Once the coefficients $\boldsymbol{a}^i$ are computed from the eigenvalue problem (13), using the identity (10) of the eigenvectors $\boldsymbol{v}^i$, the entries z_i^* of the projection vectors (principle components) can be found in terms of the kernel function as follows

$$\begin{aligned} z_i^* &= \widetilde{\Phi}(\boldsymbol{y}^*)^T\boldsymbol{v}^i = \sum_{j=1}^n a_{ij}\widetilde{\Phi}(\boldsymbol{y}^*)^T\widetilde{\Phi}(\boldsymbol{y}^j) \\ &= \sum_{j=1}^n a_{ij}\tilde{\kappa}(\boldsymbol{y}^*, \boldsymbol{y}^j), \qquad\qquad i = 1,\ldots,k. \end{aligned}$$

Reconstruction of pre-image For any arbitrary vector $\boldsymbol{y}^* \in \mathbb{R}^m$ from the input space, its pre-image $\widehat{\boldsymbol{y}}^* \in \mathbb{R}^m$ can simply be approximated as in (6) by the standard PCA. However, this is not the case for KPCA. Let us denote by $P_k\Phi(\boldsymbol{y}^*) \in \mathbb{R}^k$ the projection of the feature space vector $\Phi(\boldsymbol{y}^*)$ onto the reduced space spanned by the eigenvectors $\{\boldsymbol{v}^1, \ldots, \boldsymbol{v}^k\}$, i.e., we have that

$$P_k\Phi(\boldsymbol{y}^*) = \sum_{i=1}^k z_i^*\boldsymbol{v}^i + \bar{\Phi}(\boldsymbol{y}).$$

Then, an approximate pre-image $\widehat{\boldsymbol{y}}^*$ can be obtained so that the transformed vector $\Phi(\widehat{\boldsymbol{y}}^*)$ is the closest vector to the projection vector $P_k\Phi(\boldsymbol{y}^*)$. This requires finding, in least-squares sense, the minimum of the objective functional

$$\rho(\widehat{\boldsymbol{y}}^*) = \|\Phi(\widehat{\boldsymbol{y}}^*) - P_k\Phi(\boldsymbol{y}^*)\|.$$

Setting $\nabla_{\widehat{\boldsymbol{y}}^*}\rho = 0$ gives the solution equation

$$\widehat{\boldsymbol{y}}^* = \frac{\sum_{i=1}^k \tilde{\gamma}_i \exp(-\|\widehat{\boldsymbol{y}}^* - \boldsymbol{y}^i\|^2/2\sigma^2)\boldsymbol{y}^i}{\sum_{i=1}^k \tilde{\gamma}_i \exp(-\|\widehat{\boldsymbol{y}}^* - \boldsymbol{y}^i\|^2/2\sigma^2)}, \tag{14}$$

where

$$\tilde{\gamma}_i = \gamma_i + \frac{1}{n}\left(1 - \sum_{j=1}^n \gamma_j\right), \qquad \gamma_i = \sum_{l=1}^k z_l^* a_{li}.$$

The equation (14) can be solved by fixed point iteration. On the other hand, the iteration method is numerically unstable and relies on the initial guess [12]. In KPCA proposed here, we find the pre-image through a non-iterative scheme [9], which is based on the relationship between the distance of vectors in input space and the distance of transformed vectors in feature space, utilizing the Gaussian kernel function which can be considered as a function of distance metric $\| \cdot \|$. Using the idea of MDS [8,9], we can approximately find a pre-image $\widehat{\boldsymbol{y}}^*$ for which dissimilarities between the distance of $\boldsymbol{y}^*$ to each input space vector $\boldsymbol{y}^i$ and the distance of projected vector $P_k\Phi(\boldsymbol{y}^*)$ to each transformed feature space vectors $\Phi(\boldsymbol{y}^i)$ are preserved, see Fig. 1. To this end, let

$$d^2(\boldsymbol{y}^i,\boldsymbol{y}^j) = d_{ij}^2 = \|\boldsymbol{y}^i - \boldsymbol{y}^j\|^2 \quad \text{and} \quad \widetilde{d}^2(\Phi(\boldsymbol{y}^i),\Phi(\boldsymbol{y}^j)) = \widetilde{d}_{ij}^2 = \|\Phi(\boldsymbol{y}^i) - \Phi(\boldsymbol{y}^j)\|^2,$$

denote the (Euclidean) distance metrics of input space vectors and their transformed feature space vectors, respectively. Since the Gaussian kernel function can be taken as a function of distance metric with $f(d_{ij}^2) = \exp(-d_{ij}^2/2\sigma^2)$, we can get after some manipulations the following relation between the input space distance and feature space distance [9]

$$f(d_{ij}^2) = \frac{1}{2}(K_{ii} + K_{jj} - \widetilde{d}_{ij}^2),$$

and, for the Gaussian kernel, the inverse map yields

$$d_{ij}^2 = -2\sigma^2 \ln\left(\left(K_{ii} + K_{jj} - \|\Phi(\boldsymbol{y}^i) - \Phi(\boldsymbol{y}^j)\|^2\right)/2\right). \tag{15}$$

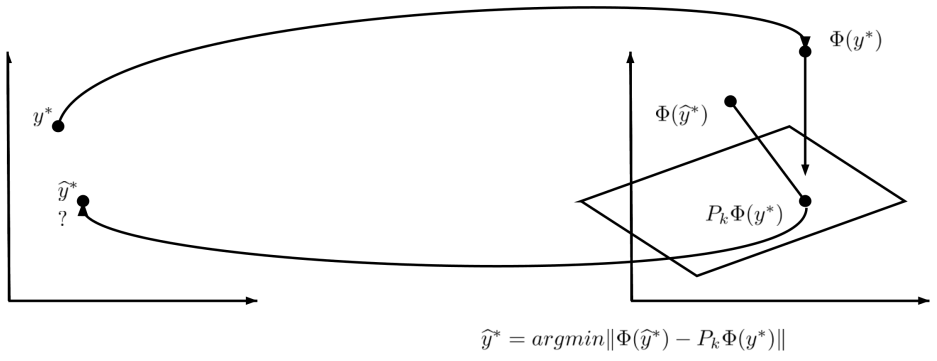

Fig. 1: Reconstruction with KPCA.

Moreover, the feature space distance between the projected vector $P_k\Phi(\boldsymbol{y}^*)$ and some transformed vector $\Phi(\boldsymbol{y}^i)$ can be shown, in terms of the kernel matrix, to be [9]

$$\widetilde{d}^2(\Phi(\boldsymbol{y}^i), P_k\Phi(\boldsymbol{y}^*)) = \|\Phi(\boldsymbol{y}^i)\|^2 + \|P_k\Phi(\boldsymbol{y}^*)\|^2 - 2P_k\Phi(\boldsymbol{y}^*)^T\Phi(\boldsymbol{y}^i)$$
$$= \left(\boldsymbol{k}_{\boldsymbol{y}^*} + \frac{1}{n}K\mathbf{1} - 2\boldsymbol{k}_{\boldsymbol{y}^i}\right)^T H^T C_a H \left(\boldsymbol{k}_{\boldsymbol{y}^*} - \frac{1}{n}K\mathbf{1}\right) + \frac{1}{n^2}\mathbf{1}^T K\mathbf{1} + K_{ii} - \frac{2}{n}\mathbf{1}^T\boldsymbol{k}_{\boldsymbol{y}^*}, \tag{16}$$

for the matrix $C_a = \sum_{j=1}^{k}(1/\widetilde{\lambda}_j)\boldsymbol{a}^j(\boldsymbol{a}^j)^T$.

Finally, using the natural assumption $P_k\Phi(\boldsymbol{y}^*) \approx \Phi(\widehat{\boldsymbol{y}}^*)$ and that the metric relation (15) inside the equation (14), we reach the non-iterative solution formula [9]

$$\widehat{\boldsymbol{y}}^* = \frac{\sum_{i=1}^{k}\gamma_i\left((2 - \|P_k\Phi(\boldsymbol{y}^*) - \Phi(\boldsymbol{y}^i)\|^2)/2\right)\boldsymbol{y}^i}{\sum_{i=1}^{k}\gamma_i\left((2 - \|P_k\Phi(\boldsymbol{y}^*) - \Phi(\boldsymbol{y}^i)\|^2)/2\right)}, \tag{17}$$

where the distances $\|P_k\Phi(\boldsymbol{y}^*) - \Phi(\boldsymbol{y}^i)\|^2$ inside the formula can be calculated by the relation (16). Additionally, the k vectors $\boldsymbol{y}^i$ referenced within the equation (17) are selected from among the n input space vectors that is closest to $\boldsymbol{y}^*$. The overall KPCA algorithm is summarized in Algorithm 1.

Algorithm 1 KPCA Algorithm

```
Input: Vector $\boldsymbol{y}^* \in \mathbb{R}^m$, matrix $Y = [\boldsymbol{y}^1\boldsymbol{y}^2\cdots\boldsymbol{y}^n] \in \mathbb{R}^{m\times n}$ , reduced dimension $k$.
Output: Pre-image vector $\widehat{\boldsymbol{y}}^* \in \mathbb{R}^m$.

1: Compute kernel matrix $K \in \mathbb{R}^{n\times n}$ and vector $\boldsymbol{k}_{\boldsymbol{y}^*} \in \mathbb{R}^n$.
2: Compute eigenvalues $\widetilde{\lambda}_i$ and eigenvectors $\boldsymbol{a}^i \in \mathbb{R}^n$ from (13), $i = 1, \ldots, k$.
3: Find $k$ closest $\boldsymbol{y}^i$ by looking at $\|\boldsymbol{y}^* - \boldsymbol{y}^i\|$, $i = 1, \ldots, k$.
4: Calculate distances $\widetilde{d}^2(\Phi(\boldsymbol{y}^i), P_k\Phi(\boldsymbol{y}^*))$ from (16), $i = 1, \ldots, k$.
5: Calculate pre-image vector $\widehat{\boldsymbol{y}}^*$ by (17).
```

4 Numerical tests

In this section, we investigate the performance of the proposed non-intrusive nonlinear dimensionality reduction technique over the KdV equation. Firstly, we construct the data matrix Y to which we apply dimensionality reduction. We consider the KdV equation (1) on the spatio-temporal domain $(x, t) \in [-10, 10]\times [0, 8]$ with the initial wave

$$u^0(x) = \frac{\beta}{2}\operatorname{sech}^2\left(\frac{\sqrt{\beta}}{2}x_0\right).$$

The problem data and mesh sizes are set as [5]: $\alpha = 6$, $\mu = 1$, $\beta = 1.5$, $\Delta x = 0.04$ and $\Delta t = 0.02$. The full order solution vectors are obtained from the full discrete system (4). By the given setting, the discrete solution vectors form the matrix Y of dimension 500×400, i.e., $m = 500$ and $n = 400$. The solution waves at some time instances are given in Fig. 2, where the initial wave moves to the right as the time progresses with the same amplitude which is expected from a numerical scheme for reliable approximations.

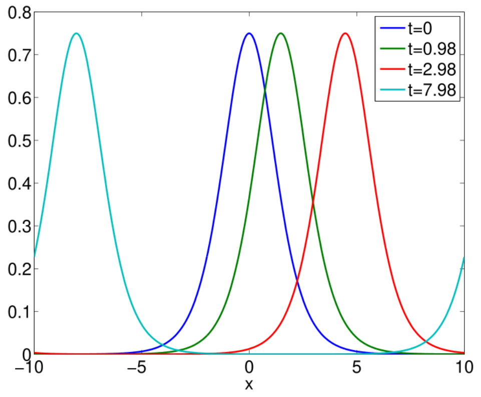

Fig. 2: Full order solution profiles at different time instances.

Preserving the numerical integrity of an invariant entails gauging the variance between the measure of the invariant at various times and its initial measure. Therefore, we expect oscillations within a narrow range, in order to indicate the preservation of an invariant. In Fig. 3, we illustrate the variance of the discrete momentum I_1, discrete mass I_2 and discrete Hamiltonian H at time instances from their initial values, obtained by full order solutions. The oscillations observed suggest a well-preservation for either invariants, with the discrete mass maintaining precision comparable to machine accuracy.

Next, we compare the traditional PCA and the KPCA in the sense of solution accuracy and computational efficiency. For this simulation, we fix the input space solution vector $\boldsymbol{y}^* = \boldsymbol{y}^{100}$ which corresponds to the wave at time $t = 2$. In terms of accuracy, we present the results in Table 1 for the reduced dimensions $k = 1, \ldots, 6$, where we compute the relative absolute error

$$\|\boldsymbol{y}^* - \widehat{\boldsymbol{y}}^*\|_R = \frac{|\boldsymbol{y}^* - \widehat{\boldsymbol{y}}^*|}{|\boldsymbol{y}^*|},$$

between the full order solution $\boldsymbol{y}^*$ and the reduced order approximation $\widehat{\boldsymbol{y}}^*$. We see that for the same reduced dimension k, the pre-images obtained by KPCA lead to more accurate approximations compared to the PCA. The results also suggest that with $k = 2$, the approximations by KPCA are satisfactory, while it needs reduced dimension greater than $k = 6$ for the PCA in order to be able to obtain the same accuracy.

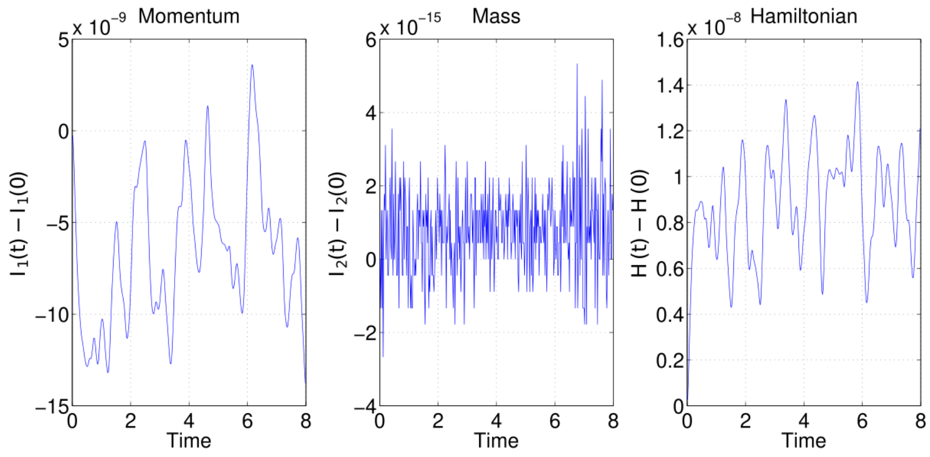

Fig. 3: Preservation of conserved quantities by full order solutions

Table 1: Relative absolute errors $\|\boldsymbol{y}^* - \widehat{\boldsymbol{y}}^*\|_R$ for different values of k

k	1	2	3	4	5	6
PCA	7.31e-01	3.84e-01	3.44e-01	2.87e-01	1.35e-01	7.78e-02
KPCA	1.41e-02	6.79e-03	1.34e-02	1.16e-02	9.87e-03	2.05e-03

In terms of computational efficiency, on the other hand, we give the CPU times needed by the KPCA scheme using the formula (14) which is solved by fixed point iteration, and by the one using the formula (17) which is an algebraic equation. Table 2 shows the results, where the first two columns are for the errors between the full and reduced order solutions with either formula, and the last two are cpu times needed to construct the pre-images. It can be seen that the errors through the nonlinear equation (14) solved with iteration and the errors through the algebraic equation (17) solved without iteration are almost the same, which claims that the solutions obtained without iteration are agreeable in the sense of accuracy. The main contribution of the KPCA method proposed in this paper is seen in the last two columns, where the CPU times needed by the scheme without iteration, are much less than the one with iteration, that is our scheme is highly fast.

As the final simulation, we test both the accuracy/reliability of the reduced order solutions and the preservation of the discrete conserved invariants of the KdV equation by the proposed KPCA algorithm. We fix the reduced dimension $k = 2$ suggested by the errors in Table 1. In order to compute the invariants by reduced order approximations, we apply the KPCA algorithm to the whole trajectory, that is, we find the pre-images $\boldsymbol{y}^* = \boldsymbol{y}^i$ for $i = 1, \dots, n$.

Fig. 4 shows the full order solutions together with the reduced order approximations at times $t = 0.5$ and $t = 3.82$. We see even for the reduced dimension $k = 2$ that at either time instance, the full and reduced order waves coincide to

Table 2: Relative absolute errors and CPU times for different values of k

	$\lVert \boldsymbol{y}^* - \widehat{\boldsymbol{y}}^* \rVert_R$		CPU Time (Sec.)	
k	With Iteration	Without Iteration	With Iteration	Without Iteration
1	1.41e-02	1.41e-02	1.9634	0.1026
2	6.79e-03	6.83e-03	1.9047	0.1630
3	1.34e-02	1.35e-02	2.1734	0.1660
4	1.16e-02	1.17e-02	1.8595	0.1932
5	9.87e-03	1.01e-02	2.2519	0.1721
6	2.05e-03	2.51e-03	2.3998	0.1872

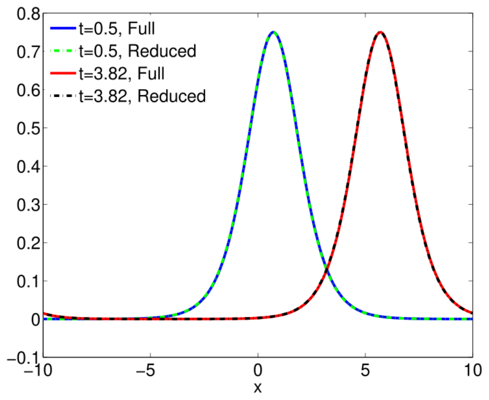

Fig. 4: Full/reduced order solution profiles at $t = 0.5, 3.82$ for $k = 2$.

each other. This means that the reduced order solutions yield a reliable wave propagation with the same wave length and wave amplitude as for the full order one, which is crucial for a physically meaningful simulation.

On the other hand, the conserved invariants obtained by the pre-images are illustrated in Fig. 5, where, similar to the full order invariants in Fig. 3, we plot the difference between the measure of either invariant at discrete times and the initial one. It is clear that the discrete mass is again conserved with machine precision, while the errors in the invariants of discrete momentum and discrete Hamiltonian are not as precise as the ones obtained by full ordered solutions. It's indeed expected within reduced order modelling methodology, particularly in the context of conserved invariants. We can confidently affirm that these invariants remain conserved over the specified time period, since their oscillations persist within a small band without any drift.

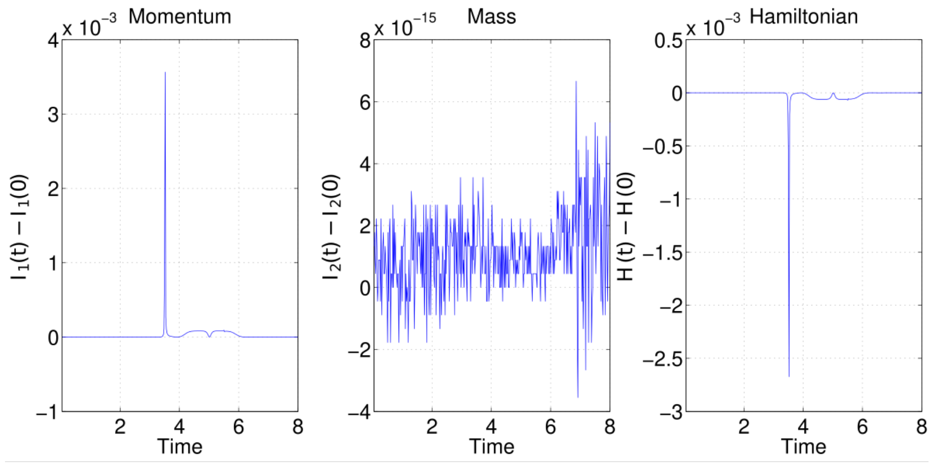

Fig. 5: Preservation of conserved quantities by reduced order approximations for $k = 2$.

5 Conclusion

In this paper, we propose a nonlinear dimensionality reduction technique that maintains the accuracy and preserves the conservation of the invariants of the KdV equation. The approach for the dimensionality reduction is non-intrusive and based on the KPCA. Within the KPCA, the idea of MDS method together with the relationship between the input space and feature space distance metrics are used in order to obtain a non-iterative algorithm to reconstruct the pre-images. The accuracy of solutions and preservation of discrete Hamiltonian, mass and momentum are demonstrated through a numerical test example, and comparison with traditional PCA and KPCA, as well as iterative scheme with non-iterative one are provided. As a future work, we plan to study on different kinds of kernel functions that may work more efficient comparing to the Gaussian kernel.

References

1. Jean-Frédéric Gerbeau and Damiano Lombardi. Approximated Lax pairs for the reduced order integration of nonlinear evolution equations. *Journal of Computational Physics*, 265:246–269, 2014.
2. Jan Hesthaven and Cecilia Pagliantini. Structure-preserving reduced basis methods for poisson systems. *Mathematics of Computation*, 90:1701–1740, 2021.
3. Yuto Miyatake. Structure-preserving model reduction for dynamical systems with a first integral. *Japan Journal of Industrial and Applied Mathematics*, 36(3):1021–1037, 2019.
4. V. Ehrlacher, D. Lombardi, O. Mula, and F. X. Vialárd. Nonlinear model reduction on metric spaces. application to one-dimensional conservative PDEs in Wasserstein spaces. *ESAIM: Mathematical Modelling and Numerical Analysis*, 2020.

5. M. Uzunca, B. Karasözen, and S. Yıldız. Structure-preserving reduced-order modeling of Korteweg-de Vries equation. *Mathematics and Computers in Simulation*, 188:193–211, 2021.
6. B. B. Jackson and Bund B. *Multivariate Data Analysis: An Introduction.* From ThriftBooks-Atlanta (AUSTELL, GA, U.S.A. Richard D. Irwin, 1983.
7. R.A. Johnson and D.W. Wichern. *Applied multivariate statistical analysis.* New Jersey: Prentice-Hall. Michael Bell, 2007.
8. T. F. Cox and M. A. A. Cox. *Multidimensional Scaling.* Monographs on Statistics and Applied Probability. Chapman & Hall, London, U.K., 2001.
9. Y. Rathi, S. Dambreville, and A. Tannenbaum. Statistical shape analysis using kernel PCA. *School of Electrical and Computer Engineering Georgia Institute of Technology*, 6064:1–8, 2006.
10. C. K. I. Williams. On a connection between kernel PCA and metric multidimensional scaling. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, Cambridge, MA, 2001. MIT Press.
11. A. G. González, A. Huerta, S. Zlotnik, and P. Diez. A kernel principal component analysis (kpca) digest with a new backward mapping (pre-image reconstruction) strategy. *Universitat Politecnica de Catalunya - Barcelona*, 2021.
12. S. Mika, B. Schölkopf, A. Smola, K. R. Müller, M. Scholz, and G. Rätsch. Kernel PCA and de-noising in feature spaces. *Advances in neural information processing systems*, 11:536–542, 1998.
13. B. Schölkopf and A. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 2002.
14. B. Schölkopf, A. Smola, and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
15. Q. Wang. Kernel principal component analysis and its applications in face recognition and active shape models. *Computer Vision and Pattern Recognition*, pages 1–8, 2012.
16. E. Celledoni, V. Grimm, R. I. McLachlan, D. I. McLaren, D. O'Neale, B. Owren, and G. R. W. Quispel. Preserving energy resp. dissipation in numerical pdes using the 'Average Vector Field' method. *Journal of Computational Physics*, 231(20):6770 – 6789, 2012.
17. William Kahan and Ren-Chang Li. Unconventional schemes for a class of ordinary differential equations with applications to the Korteweg-de Vries equation. *Journal of Computational Physics*, 134(2):316–331, 1997.
18. E. Celledoni, R.I. McLachlan, B. Owren, and G.R.W. Quispel. Geometric properties of Kahan's method. *Journal of Physics A: Mathematical and Theoretical*, 46(2):025201, 2013.