Multiscale Extensions for Enhancing Coarse Grid Computations

Neta Rabin[a], Ángela Fernández[b], Dalia Fishelov[c]

[a] *Dept. of Industrial Engineering, Tel-Aviv University, Israel*
[b] *Dept. of Computer Engineering, Universidad Autonoma de Madrid, Spain*
[c] *Dept. of Mathematics, Afeka Tel-Aviv Academic College of Engineering, Israel*

Abstract

Many phenomena in Physics and Engineering are modeled by Partial Differential Equations. Typically, analytical solutions of such problems are unknown. Hence, numerical schemes are applied for approximating the exact solutions. In order to accurately approximate the exact solutions, very fine grids are required. However, the computations become too costly. In this work, a multiscale iterative approach is utilized for enhancing the accuracy of coarse grid computations which results from a high-order finite difference scheme. The main idea is to run the finite difference computations on a coarse grid until some intermediate time lever. Then, the coarse grid results are interpolated and extended to a fine grid using Laplacian Pyramids, a multiscale iterative approach. Finally, the finite difference scheme is employed on the finer grid until a prescribed final time. Comparing the results to those obtained on a fine grid, we see that the convergence rates obtained from the two methods are comparable, while the computational time is significantly reduced.

A modified multimodal Laplacian Pyramids algorithm for predicting future values of the solution is also suggested. The method approximates and extends a function based on two or more input modalities coded by a series of multiscale kernels, which are averaged as a convex combination. In this work, the modalities are the numerical model's approximations of the solution of the differential equation and its derivative at previous time steps, and the goal is to predict the solution at a proceeding time step. It can be seen that, by adapting the convex combination of the kernels to local regions of the solution with stronger or weaker gradients, the predicted results are improved.

Keywords: High-order finite difference schemes, Laplacian Pyramids, Forecasting

1. Introduction

Many phenomena in Physics and Engineering are modeled by Partial Differential Equations (PDEs). Typically, analytical solutions of such problems are unknown and, therefore, numerical schemes, such as finite difference [1] and finite element methods [2], are applied for approximating the exact solutions. In order to accurately approximate these exact solutions, in terms of the solution and its gradient, very fine grids are required. As a consequence, the computations become too costly and different approaches have been suggested for enhancing the speed of computations, while preserving highly accurate approximations, which would have been generated via finer grids.

Several recent approaches have been proposed for enhancing coarse grid computations. In [3], a Smoothness-Increasing Accuracy-Conserving filter was applied as a post-processing procedure for enhancing approximate solutions of PDEs. A neural network was proposed in [4] for resolving the gradients of the solution near the boundaries of the physical domain as, otherwise, one should apply a very fine grid or finite differences and finite elements methods in order to resolve the high gradients of the solution inside the physical domain and close to its boundary. In [5], a numerical method for approximating the solution of the one-dimensional acoustic wave problem was investigated, when violating the numerical stability condition. The authors of [5] used deep learning to create an explicit non-linear scheme that remains stable for larger time steps. The proposed spatio-temporal neural-network architecture is additionally enhanced during training with a physically-informed term, in order to adapt the approximate solution to the solution of the physical problem. Recovering a scatterer in an underwater medium was presented in [6]. There, the acoustic wave equation was solved numerically to create a dataset. Then, several neural networks with different architectures and parameters were trained, where the goal is to retrieve both, the location and shape of the unknown scatterer. After training the model, the inference takes milliseconds and the model can be used for real-time applications. A new approach called the Finite Volume Neural Network (FINN) was introduced in [7]. The FINN method adopts the numerical structure of the well-known Finite Volume Method for handling partial differential equations, so that each quantity of interest follows its own adaptable conservation law, while it concurrently accommodates learnable parameters.

In [8] kernel analog forecasting (KAF) methods were studied from the perspective of data generated by multiscale dynamical systems. The authors use multiscale systems to provide a framework in which the fundamental role of Markovianity in data-driven prediction can be studied. The setting of KAF is a methodology that has been successful in a number of applications, which were backed by a mature theory. The studies provide guidance for the interpretation of data-driven prediction methods when used in practice. The authors of [9] introduce and implement a framework for systematically extracting coarse-scale observables from microscopic/fine scale data and for discovering the underlying governing equations using machine learning techniques (e.g. Gaussian processes and artificial neural networks) enhanced by feature selection methods. Intrinsic representations of the coarse-scale behavior via manifold learning techniques (in particular, Diffusion Maps), generating alternative possible forms of the governing equations is also explored and discussed. See also [10] for a review on solving PDEs in Physics using machine learning.

In this work, a multiscale iterative approach is utilized for enhancing the accuracy of high-order finite difference results, which are applied in coarse grids. The main idea is to begin the computation of the evolution in time on a coarse grid, using a finite difference scheme, while switching from a coarse to a fine grid towards the final time steps of the computations. The coarse grid results are interpolated and extended to a fine grid using Laplacian Pyramids (LP), which is a multiscale iterative approach [11, 12].

We will show that comparing the results to those obtained on a fine grid throughout the computations, the errors and the rates of convergence obtained from the two methods are comparable, while the computational time is significantly reduced with this proposal.

A second part of this paper uses a modified LP model for predicting future values of the solution. The modified LP method approximates and extends a function based on two or more input modalities. Each modality is coded by a series of multiscale kernels, which are averaged as a linear combination. In this work, the modalities are the numerical model's approximations of the problems solution and its derivative at past time points, and the goal is to predict or forecast the solution at a future time. We will show that, by adapting the linear combination of the kernels to the local regions of the solution that have stronger or weaker gradients, the prediction results are improved.

The outline of this paper is the following. In Section 2 the Laplacian Pyramids for approximation and extension of functions are introduced. In addition, a procedure for Forecasting, called the multimodal Laplacian Pyramids is described. In Section 3, we describe the two algorithms used in this paper. The first is for switching from a coarse to a fine grid in order to save computational time, while keeping the same convergence rates. The second is the algorithm for forecasting. In Section 4 the numerical method is described in more detail for both algorithms. Finally, numerical results are presented in Section 5. They assess the fourth-order accuracy of the proposed algorithm and the accuracy of the forecasting algorithm. Conclusions are provided in Section 6.

2. The Laplacian Pyramids Scheme

Laplacian Pyramids (LP) is a method for approximation and extension of functions that are defined over a grid type or a scattered dataset. By convolving the data with Gaussian kernels of decreasing widths, a multiscale representation is constructed. In statistics, this approach is known as the Nadaraya-Watson estimator [13]. The regression model is achieved by stopping the iterations when the difference between the functions and its approximation is smaller than a predefined error. When the function is smooth, such as in the case we are dealing with in this work, the LP iterations are evoked until the residual is small enough thus, the LP model is an interpolation (rather than a regression) scheme [14]. In what follows, we present two versions of the LP scheme. The first, is the standard version, which will be utilized for extending the information from the coarse grid to the fine grid. In the second version, the LP model is computed based on two input modalities.

2.1. Laplacian Pyramids for Interpolation and Extension

Let $X = \{x_0, \dots, x_n\}$ be a set of data points. Here, this set contains the grid points that belong to a coarse grid. Let f be a function that is defined on X. In the present application, f is a numerical approximation to the physical phenomena that is generated by a high-order numerical scheme. An interpolation of f at the grid points X is constructed as follows. A series of

Gaussian kernels of decreasing scales, $G_l = (g_l(x_i, x_j)))$, are defined by

$$G_l = g_l(x_i, x_j) = e^{\frac{-\|x_i - x_j\|^2}{\sigma_l}}, \tag{1}$$

where $x_i, x_j \in X$. The initial kernel scale σ_0 is set to be a large number. The associated row-normalized kernels of G_l are denoted by $K_l = k_l(x_i, x_j)$. Then, a representation of f is computed using an iterative procedure, by convolving f with the series of smoothing operators K_l. A first course representation of f is constructed by

$$s_0(x_i) = \sum_{x \in X} k_0(x_i, x) f(x). \tag{2}$$

We denote the first convolution as $f_0 = s_0$. The difference $d_1 = f - f_0$ is averaged by a finer kernel $K_1 = k_1(x_i, x_j)$, that has $\sigma_1 = \frac{\sigma_0}{2}$. This results with a finer representation of f, $f_1 = f_0 + s_1$, where $s_1(x_i) = \sum_{x \in X} k_1(x_i, x) d_1(x)$. In general, for $l = 1, 2, 3 \ldots$, we have $d_l = f - f_{l-1}$, and

$$f_l(x_i) = f_{l-1}(x_i) + s_l(x_i) = f_{l-1}(x_i) + \sum_{x \in X} k_l(x_i, x) d_l(x). \tag{3}$$

The iterations are stopped when the difference $\|f - f_l\|$ is smaller than a prescribed ϵ.

Extension of the model to new points is straightforward. Given a new point $\tilde{x}$, which is in the present application a grid point that belongs to the fine grid, the multiscale representations $f_0, f_1, \ldots, f_l$ are extended to evaluate $f_l(\tilde{x})$. First, s_0 is extended by

$$s_0(\tilde{x}) = \sum_{x \in X} k_0(\tilde{x}, x) f(x). \tag{4}$$

Similarly, the kernels that form the finer resolutions $s_1, \ldots, s_l$ are extended, resulting with

$$f_l(\tilde{x}) = f_{l-1}(\tilde{x}) + s_l(\tilde{x}) = f_{l-1}(\tilde{x}) + \sum_{x \in X} k_l(\tilde{x}, x) d_l(x). \tag{5}$$

2.2. Multimodal Laplacian Pyramids

We describe a general modification of LP for multimodal input. Later, this model is adapted for forecasting by considering input points that are time-trajectories and a function that is the value of these trajectories at a future time.

Let $\mathbf{X}^{(A)} = \{x_0^A, \ldots, x_n^A\}$ and $\mathbf{X}^{(B)} = \{x_0^B, \ldots, x_n^B\}$ be two aligned datasets, here two datasets that are defined on the $n+1$ grid points. The function f is defined over $\mathbf{X}^{(A)}$ and $\mathbf{X}^{(B)}$, on each grid point, such that $f(x_i^A) = f(x_i^B) = f(i)$.

We construct a series of coarse kernels $G_l^{(A)}$ and $G_l^{(B)}$ based on $\mathbf{X}^{(A)}$ and $\mathbf{X}^{(B)}$ with initial coarse scales $\sigma_0^{(A)}$ and $\sigma_0^{(B)}$. Denote the associated row-normalised kernels by $K_l^{(A)}$ and $K_l^{(B)}$. Consider a new series of kernels, which is a convex combination of two kernels at a given scale, defined by

$$\alpha K_l^{(A)} + (1-\alpha) K_l^{(B)}. \tag{6}$$

A first coarse approximation of f, $f_0 = s_0$, is then defined by

$$s_0(i) = \alpha \left(\sum_{x_j^A \in \mathbf{X}^{(A)}} k_0^{(A)}(x_i^A, x_j^A) f(j) \right) + (1-\alpha) \left(\sum_{x_j^B \in X^{(B)}} k_0^{(B)}(x_i^B, x_j^B) f(j) \right). \tag{7}$$

The parameter α, $0 \leq \alpha \leq 1$, defines the weight assigned to each input modality. Then, the residual $d_1 = f - f_0$ is smoothed by the linear combinations of the kernels at level $l = 1$, as defined in Eq. (6). The iterations are defined by

$$f_l(i) = f_{l-1}(i) + s_l(i), \tag{8}$$

where,

$$s_l(i) = \alpha \left(\sum_{x_j^A \in \mathbf{X}^{(A)}} k_l^{(A)}(x_i^A, x_j^A) d_l(j) \right) + (1-\alpha) \left(\sum_{x_j^B \in \mathbf{X}^{(B)}} k_l^{(B)}(x_i^B, x_j^B) d_l(j) \right). \tag{9}$$

Here, $d_l(i) = f(i) - f_{l-1}(i)$.

Given a new pair of input points that belong to $\mathbf{X}^{(A)}$ and $\mathbf{X}^{(B)}$, extension of the kernels $K_l^{(A)}$ and $K_l^{(B)}$, is carried out in a similar manner as in Equations (4) and (5).

2.3. Error Analysis for the LP Scheme

For analyzing the LP error, the previously defined kernel is considered. First, notice that, when working in the continuous kernel setting, the summation becomes an integral. Therefore, we have $k_l(x, x')$ for a Gaussian function.

Furthermore, for all l, writing now $p_l\,(x) = k_l(x, 0)$, is an approximation to a delta function satisfying

$$\begin{aligned} &\int p_l\,(x)\,dx = 1, \qquad \int x p_l\,(x)\,dx \;\; = 0, \\ &\int \|x\|_2^2\; p_l\,(x)\,dx \leq 2C, \end{aligned} \tag{10}$$

where C is a constant. Assume that f is in L_2, then (see [12, 15])

$$\|f_l - f\|_{L^2} \leq C\sigma^2 \left(\tfrac{\sigma^2}{\mu^{(l+1)}}\right)^l \|f\|_{2l+2,2}, \tag{11}$$

where $\|f\|_{m,2}$ denotes the Sobolev norm of a function with up to m derivatives in L_2. Thus, the L_2 norm of the LP error decays at a very fast rate.

Applying the previous result to the kernel $\alpha k_l^{(A)} + (1-\alpha)k_l^{(B)}$ and defining $f - f_l = d_{l-1}$ (see Equation (9)), we have for the convex combinations of the two kernels the same bound for the error as in Equation (11).

3. The Proposed Algorithms

In this work the LP is utilized in two algorithms. The first algorithm is the extension of the numerical results from a coarse to a fine grid, for which the results are invoked for the application of the scheme in the proceeding time interval. The second algorithm for which the LP algorithm is invoked is for forecasting future values of the solution of the differential equation, using regression of the resulting numerical results for previous time steps.

3.1. Extending coarse grid approximations

The LP is utilized for enhancing the accuracy of high-order finite difference results, which are applied to the resulting approximation of the differential equation on a coarse grid. In more detail, we begin the computation of the evolution in time of the differential equation on a coarse grid, using a high order finite difference scheme, while switching from a coarse to a fine grid in the proceeding time interval.

In particular, we aim to reach an approximation at time $t = T_0$. The simulation is executed on a coarse grid until $t^\star = \beta T_0$, where $0.5 \leq \beta \leq 0.75$. Here, we set $\beta = 0.5$. Then, the coarse grid results are interpolated and extended to a finer grid, using the LP scheme (Section 2.1). The simulation is executed on the fine grid until we reach $t = T_0$.

3.2. Forecasting future values with the multimodal LP scheme

In this part, the multimodal LP scheme is applied for prediction of future values of the solution based on historic time trajectories that were generated by the numerical scheme. Concretely, given the numerical approximations for $f(i)$ and $f_x(i)$ at times $t-k$ to t on all grid points, $0 \leq i \leq n$, we forecast the values of the numeric approximation at a future time $\tilde{t}$, where $\tilde{t} > t$.

The input dataset $\mathbf{X}^{(A)}$ holds trajectories of length k of the numerical approximation of $f(i)$ at each grid point. The second dataset $\mathbf{X}^{(B)}$ holds concatenated trajectories of $f(i)$ and $f_x(i)$ at each grid point (the concatenated trajectories are of length $2k$). We note that the concatenation of the trajectories of f to f_x improves the results compared to taking just the trajectories of f_x.

4. Numerical Methods

Our test problem is the Kuramoto-Sivashinsky equation

$$\begin{aligned} &\tfrac{\partial u}{\partial t} + \partial_x^4 u + \partial_x^2 u + u\ \partial_x u = f(x,t), \quad -30 < x < 30, \quad t > 0, \\ &u(-30,t) = \partial_x u(30,t) = 0, \quad t \geq 0 \\ &u(30,t) = \partial_x u(30,t) = 0, \quad t \geq 0 \\ &u(x,0) = u_0(x), \quad -30 \leq x \leq 30. \end{aligned} \tag{12}$$

The right-hand side $f(x,t)$ is chosen so that the exact solution for the problem is

$$\begin{aligned} u(x,t) = c + (15/19)\sqrt{11/19}[-9\tanh(k(x-ct-x_0)) \\ +11\tanh^3(k(x-ct-x_0))], \end{aligned} \tag{13}$$

where $x \in [-30,30]$. In this case one may show that $f(x,t)=0$. We picked $c=-0.1, k=0.5\sqrt{11/19}$ and $x_0=-10$.

4.1. The numerical finite-difference scheme

We equip the interval $\Omega = [0,1]$ with a uniform grid

$$x_j = jh, \quad 0 \leq j \leq N, \quad h = \frac{1}{N}.$$

The approximation is carried out by grid functions v defined on $\{x_j,\ 0 \leq j \leq N\}$. The space of these grid functions is denoted by l_h^2.

The Hermitian derivative of $v \in l^2_{h,0}$, approximates $\partial_x u$. It will serve not only in approximating (to fourth-order of accuracy) first-order derivatives, but also as a fundamental building block in the construction of finite difference approximations to higher-order derivatives. First, we introduce the "Simpson operator"

$$(\sigma_x v)_j = \frac{1}{6}v_{j-1} + \frac{2}{3}v_j + \frac{1}{6}v_{j+1}, \quad 1 \le j \le N-1. \tag{14}$$

The Hermitian derivative v_x is now defined by

$$(\sigma_x v_x)_j = (\delta_x v)_j, \quad 1 \le j \le N-1, \tag{15}$$

where $\delta_x v = \frac{v_{j+1}-v_{j-1}}{2h}$.

Remark. *In definition* (15)*, the values of* $(v_x)_j$*,* $j = 0, N$*, need to be provided, so that the left-hand side of the equation is well defined for* $j = 1, N-1$*. If not otherwise specified, we shall henceforth assume that these values are given by the boundary conditions* $v_x \in l^2_{h,0}$*, namely*

$$(v_x)_0 = (v_x)_N = 0.$$

In particular, the linear correspondence $l^2_{h,0} \ni v \to v_x \in l^2_{h,0}$ *is well defined, but not onto, since* δ_x *has a non-trivial kernel.*

We define now δ_x^2, which is a second-order approximation to $\partial_x^2 u$:

$$(\delta_x^2 v)_j = \frac{v_{j+1} - 2v_j + v_{j-1}}{h^2}, \quad 1 \le j \le N-1. \tag{16}$$

The biharmonic discrete operator is given by (for v, $v_x \in l^2_{h,0}$)

$$\delta_x^4 v = \frac{12}{h^2}[\delta_x v_x - \delta_x^2 v]. \tag{17}$$

We next introduce a fourth-order replacement to the operator δ_x^2:

$$(\widetilde{\delta_x^2} v)_j = 2(\delta_x^2 v)_j - (\delta_x v_x)_j, \quad 1 \le j \le N-1. \tag{18}$$

All the approximations in space for $\partial_x^4 u$ and $\partial_x^2 u$ are fourth-order and compact (it uses only nearest grid neighbors). To step the differential equation in time we have used a Crank-Nicolson scheme. It is an implicit trapezoidal rule for the linear terms $\partial_x^4 u$ and $\partial_x^2 u$ and it is an explicit scheme for the non-linear term $u\ \partial_x u$. Since this is a second order scheme in time, we have picked the time step Δt to be $O(h^2)$:

$$\begin{cases} \dfrac{v_j^{n+1/2} - v_j^n}{\Delta t/2} + \dfrac{1}{2}(\delta_x^4 v_j^{n+1/2} + \tilde{\delta}_x^2 v_j^{n+1/2}) = -\dfrac{1}{2}(\delta_x^4 v_j^n + \tilde{\delta}_x^2 v_j^n) - v_j^n v_{x,j}^n + f_j^{*,n}, \\ \dfrac{v_j^{n+1} - v_j^n}{\Delta t} + \dfrac{1}{2}(\delta_x^4 v_j^{n+1} + \tilde{\delta}_x^2 v_j^{n+1}) = -\dfrac{1}{2}(\delta_x^4 v_j^n + \tilde{\delta}_x^2 v_j^n) + v_j^{n+1/2} v_{x,j}^{n+1/2} + f_j^{*,n+1/2}, \end{cases} \tag{19}$$

5. Numerical results

Recall that our test problem is the Kuramoto-Sivashinsky equation

$$\begin{aligned} &\tfrac{\partial u}{\partial t} + \partial_x^4 u + \partial_x^2 u + u\ \partial_x u = w(x,t), \quad x \in (-30,30), t > 0, \\ &u(-30,t) = \partial_x u(30,t) = 0, \quad t \geq 0, \\ &u(30,t) = \partial_x u(30,t) = 0, \quad t \geq 0, \\ &u(x,0) = u_0(x), \quad x \in [-30,30]. \end{aligned} \tag{20}$$

The right-hand side $w(x,t)$ is chosen so that the exact solution for the problem is

$$\begin{aligned} u(x,t) = c + (15/19)\sqrt{11/19}[&-9\tanh(k(x-ct-x_0)) \\ &+11\tanh^3(k(x-ct-x_0))], \end{aligned} \tag{21}$$

where $x \in [-30,30]$.

In this case $f(x,t) = 0$. We picked $c = -0.1, k = 0.5\sqrt{11/19}$ and $x_0 = -10$. The discrete solution v is again obtained by the scheme (19).

In Figure 1 we have a plot of the solution and its numerical approximation at $t = 1$ and $N = 121$ (left) and for $N = 961$ (right) in order to show the gradients in the exact and numerical solution cases.

5.1. Extending Coarse Grid Approximations for the Kuramoto-Sivashinsky Equation

We display results when we combine the finite-difference scheme with kernel extensions, as explained in Section 3.1. First, the fourth-order compact

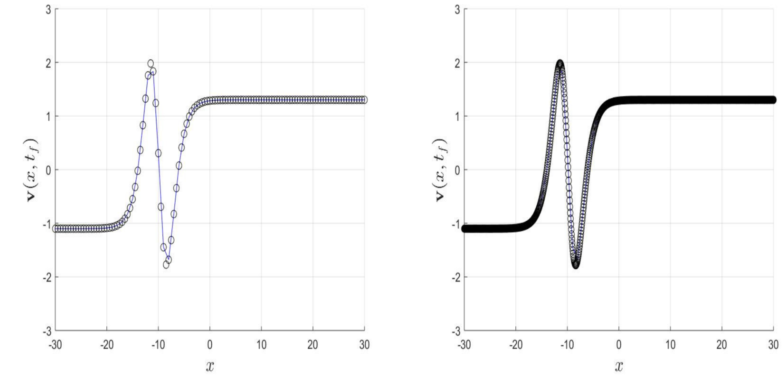

Figure 1: KS numerical example: Exact solution (solid line) and computed solution (circles) for $N = 121$ (left) and $N = 961$ (right).

scheme was ran to final times $T = 1$ and $T = 2$ with meshes of $N = 241$, $N = 481$ and $N = 961$. In Table 1 we display numerical results for the equation (20) for the fourth-order scheme. We have picked $\Delta t = h^2$, where $h = 1/N$, and the final time is $t = 1$. Observe in the right panel that fourth-order accuracy is achieved for u and $\frac{\partial u}{\partial x}$.

Mesh	$N = 241$	Rate	$N = 481$	Rate	$N = 961$
$T = 1,\ \lvert e\rvert_h$	3.2873(-4)	3.99	2.0752(-5)	4.00	1.2984(-6)
$T = 1,\ \lvert e_x\rvert_h$	2.9822(-4)	3.95	1.9332(-5)	3.98	1.2246(-6)
$T = 2,\ \lvert e\rvert_h$	3.8205(-4)	3.97	2.4315(-5)	3.99	1.5265(-6)
$T = 2,\ \lvert e_x\rvert_h$	3.6001(-4)	3.92	2.3719(-5)	3.98	1.5048(-6)

Table 1: Compact scheme for KS equation (12) with exact solution $u = u(x,t) = c + (15/19)\sqrt{(11/19)}(-9\tanh(k(x-ct-x_0)) + 11\tanh^3(k(x-ct-x_0))$. We display $|e|_h$ and $|e_x|_h$, the errors in u, and u_x, respectively at $t = 1$. The time step is $\Delta t = h^2$.

On the other hand, the scheme was ran with $N = 241$ up to $T = 1$, then extended the discrete solution to $N = 481$. Subsequently, the scheme was ran with $N = 481$ from $T = 1$ to $T = 2$. In a similar test we first ran the scheme with $N = 481$ up to $T = 1$, then extended the function to $N = 961$ and then ran the scheme with $N = 961$ from $T = 1$ to $T = 2$. The results are displayed in Table 2.

We see that although we ran the scheme on the finer mesh only from

Mesh	$N = 241$ ext. to $N = 481$	Rate	$N = 481$ ext. to $N = 961$
$T = 2,\ \lvert e\rvert_h$	2.0853(-4)	4.22	1.1136(-5)
$T = 2,\ \lvert e_x\rvert_h$	7.1261(-4)	4.40	3.3690(-5)

Table 2: Compact scheme for KS equation (20) with exact solution $u = u(x,t) = c + (15/19)\sqrt{(}11/19)(-9\tanh(k(x-ct-x_0)) + 11\tanh^3(k(x-ct-x_0))$. We display $|e|_h$ and $|e_x|_h$, the errors in u, u_x, respectively at $t = 2$. The time step is $\Delta t = h^2$. Here we first ran the scheme with a coarse mesh to $T = 1$, then extended it to a finer mesh, and then ran on the finer mesh to $T = 2$

$T = 1$ to $T = 2$, instead of running it through all the time-interval from $T = 0$ to $T = 2$, the rate of convergence in 2 is around 4, as it is in 1. So, we have saved computational time for half of the time interval, while keeping the same convergence rate.

5.2. Forecasting Future Values of the Kuramoto-Sivashinsky Equation with Multimodal LP

In this experiment, the input for the regression model is taken from the results of the numerical scheme executed on a fine grid, with 961 grid points. Here, the high-order numerical scheme, which was described in Section 4, was carried out starting with $T = 0$ until $T = 1$. The time step was picked as $\Delta t = (\Delta x)^2$, where $\Delta x = 1/960$ is the mesh size. The number of the time steps needed to reach $T = 1$ is therefore 256, and the intermediate temporal steps are $t_n = n\Delta t, n = 1, 2, ..., 256$. Thus, $f(x) = v(x, 1)$, where $v(x,t)$ is the finite-difference approximation to the solution $u(x,t)$ of the Kuramoto-Sivashinsky equation.

The model was trained on data from the temporal interval $t = 231$ to $t = 250$ in order to approximate $f(i) = f(x)$ at the time step $t = 255$, for all grid points $0 \leq i \leq n$. Then, by extending the multimodal LP to new time intervals that hold data from time steps $t = 232$ to $t = 251$, we forecast the values of $f(x)$ at time step $t = 256$.The data which we have used for $\mathbf{X^{(A)}}$ and $\mathbf{X^{(B)}}$ is the numerical approximations to the Kuramoto-Sivashinsky equation and its derivative, as described in Section 3.2.

Since the solution of the Kuramoto-Sivashinsky equation has strong gradients in some regions and smooth in others, the forecasting was split into 3 sub-regions, based on the smoothness of the solution. Figure 2 plots the 3 sub-regions (they were detected automatically by performing a k-means clustering on $f(x)$ at time step $t = 255$). Region 1 (in blue) spans the interval

$-30 \leq x \leq -10$, region 2 (in green) is the interval $-10 < x \leq 10$ and region 3 (in yellow) is the interval $10 < x \leq 30$.

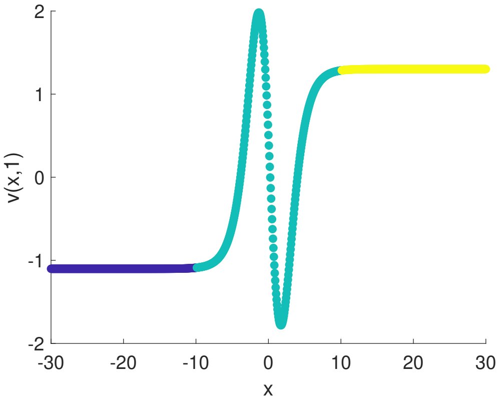

Figure 2: $f(x)$ split into 3 regions based on the gradients of the function.

Multimodal LP was used to forecast the values of the numeric approximation after $t = 256$ time steps ($v(x, 1)$) in each of the regions. Three values of the parameter α (see Eqs. (8) and (9)) were tested, specifically $\alpha = 1,\ 0.5,\ 0$. Note that as α decreases from 1 to 0, more weight is given to the kernels $K_l^{(B)}$, which hold information on $f_x = v_x$. When $\alpha = 1$, in practice, we evoke the regular single modality LP scheme.

Table 3 plots the forecasting errors in terms of Root Mean Square Error (RMSE). It may be seen that adding the information for $\mathbf{X}^{(\mathbf{B})}$ significantly reduces the errors in the second region. On the other hand, in the smooth regions, this information is not needed. Moreover, splitting the region to local sub-regions, improves the overall accuracy.

Region	$\boldsymbol{\alpha = 1}$	$\boldsymbol{\alpha = 0.5}$	$\boldsymbol{\alpha = 0}$
Reg. 1 ($-30 \leq x \leq -10$)	**7.09(-7)**	8.11(-7)	8.89(-7)
Reg. 2 ($-10 < x \leq 10$)	3.98(-4)	2.61(-5)	**1.53(-6)**
Reg. 3 ($10 < x \leq 30$)	**7.08(-7)**	8.15(-7)	8.94(-7)
Regs. 1,2,3 ($-30 \leq x \leq 30$)	4.02(-4)	4.12(-5)	**2.7(-6)**

Table 3: Forecasting errors for $f(x) = v(x, 1)$ in terms of RMSE in three regions for different and modalities combinations (different values of α).

6. Conclusions

This work presents an efficient and easy-to-implement method for enhancing computation on a coarse grid to a finer grid, while keeping the fine grid's convergence rate. Extending the information from the coarse to fine grid is done via the Laplacian Pyramids algorithm, which interpolates and extends the data with high accuracy. Moreover, an adaptation of the Laplacian Pyramid scheme that takes several modalities as its input was introduced and applied for forecasting.

7. Acknowledgement

This research was supported by the Israel Science Foundation [Grant no. 1144/20].

References

[1] M. Ben-Artzi, J.-P. Croisille and D. Fishelov, *"Navier-Stokes Equations in Planar Domains"*, Imperial College Press, 2013.

[2] V. Girault, and P. A. Raviart, Finite element methods for Navier-Stokes equations: theory and algorithms (Vol. 5). Springer Science & Business Media, 2012.

[3] J. Docampo-Sánchez, G. B. Jacobs,X. Li, and J. K. Ryan, Enhancing accuracy with a convolution filter: What works and why!. Computers & Fluids, 213, p.104727, 2020.

[4] J. Nordström and O. Ålund, Neural network enhanced computations on coarse grids. Journal of Computational Physics, 425, 109821, 2021.

[5] O. Ovadia, A. Kahana, E. Turkel and S. Dekel, Beyond the Courant-Friedrichs-Lewy condition: Numerical methods for the wave problem using deep learning, J. Comput. Physics, Vol. 442, 110493.

[6] A. Kahana, E. Turkel, S. Dekel and D. Givoli, Obstacle segmentation based on the wave equation and deep learning, J. Comput. Physics, Vol. 413, 109458.

[7] T. Praditia, M. Karlbauer, S. Otte, S. Oladyshkin, M. Butz, W. Nowak, Finite Volume Neural Network: Modeling Subsurface Contaminant Transport, arXiv:2104.06010

[8] D. Burkov, D. Giannakisy, K. Manohar and A. Stuart, Kernel Analog Forecasting: Multiscale Test Pproblems, Multiscale Modeling and Simulation, Vol. 19, 19(2), pp. 1011–1040, 2021.

[9] S. Lee,M. Kooshkbaghi,K. Spiliotis,C. I. Siettos and I. G. Kevrekidis, Coarse-scale PDEs from fine-scale observations via machine learning, https://arxiv.org/pdf/1909.05707.pdf , 2021.

[10] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang and L. Yang, Physics-informed machine learning. Nature Reviews Physics, Vol. 3, pp. 422–440, 2021.

[11] N. Rabin, R. R. Coifman, Heterogeneous datasets representation and learning using diffusion maps and Laplacian pyramids. In Proceedings of the 2012 SIAM International Conference on Data Mining (pp. 189-199). Society for Industrial and Applied Mathematics, 2012.

[12] Á. Fernández, N. Rabin, D. Fishelov, and J. R. Dorronsoro, Auto-adaptive multi-scale Laplacian Pyramids for modeling non-uniform data. Engineering Applications of Artificial Intelligence, 93, p.103682, 2020.

[13] H. J. Bierens, The Nadaraya-Watson kernel regression function estimator, *Topics in Advanced Econometrics* New York: Cambridge University Press. 212–247, 1994.

[14] L. Kang and R. V. Joseph, Kernel approximation: From regression to interpolation, *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):112–129, 2016.

[15] D. Fishelov, A new vortex scheme. Journal of Computational Physics, 86, pp. 211-224, 1990.