




Data Driven Regional Weather Forecasting

Randall Clark, Luke C. Fairbanks ¹, Ramon E. Sanchez, Pacharadech
Wacharanan,
Department of Physics
University of California San Diego
La Jolla, CA 92093, USA

Henry D. I. Abarbanel
Department of Physics
and
Marine Physical Laboratory (Scripps Institution of Oceanography)
University of California San Diego
La Jolla, CA 92093, USA
habarbanel@ucsd.edu
 <https://orcid.org/0000-0002-4690-6081>

November 7, 2022

¹Corresponding Author: lfairban@ucsd.edu



16	Contents	
17	1 Introduction	5
18	1.1 Recent Work in Using Machine Learning Methods in Weather and	
19	Climate Modelling	6
20	1.2 Plan for this Paper	7
21	2 Shallow Water Flow on a β plane used to illustrate DDF	8
22	2.1 Fluid Equations on a Grid; The Global Model	9
23	2.2 Dynamics on a Subgrid; The Regional Model	9
24	2.3 Proceeding in Two Steps	11
25	2.3.1 First Step, $D_R = D_G$; Introducing Radial Basis Functions .	11
26	2.3.2 Second Step, $D_R < D_G$; Utilizing Time Delay Embedding .	14
27	3 Estimating χ by Minimizing $C(\chi)$	15
28	4 Results from the Example of the SWE on a β Plane	18
29	4.1 Clustered Sensor Region; 3x3 Corner	18
30	4.2 Region With Sparse, Distributed Sensors	26
31	4.3 Off Center; 2x2 Region	26
32	5 Addressing Noisy Data	36
33	6 Summary and Discussion	43
34	6.1 General Remarks	43
35	6.2 The Specific Results in this Paper	45
36	6.3 Looking Forward	45
37	7 Appendix	47
38	7.1 Performing DDF on Observed Data	47
39	7.2 Designing a Representation of the Discrete Time Flow Vector Field	47
40	7.2.1 Choosing a Radial Basis Function	48
41	7.2.2 Including Polynomial Terms in Eq. (24)	49
42	7.2.3 How to choose Centers	50
43	7.2.4 Time Delay Embedding (TDE)	50
44	7.2.5 Finding the hyper-parameters R and B in χ	51
45	7.3 Programming DDF	52
46	7.3.1 PreBuilt DDF Code in Python	52
47	7.3.2 Memory Management	52
48	7.3.3 Parallelizing DDF	52
49	7.4 Future Improvements to DDF	53



50	7.4.1 Centers and RBF's	53
51	7.4.2 Differential Evolution	53



52 Abstract

53 Using data alone, without knowledge of underlying physical models, nonlin-
54 ear discrete time regional forecasting dynamical rules are constructed employ-
55 ing well tested methods from applied mathematics and nonlinear dynamics.
56 Observations of environmental variables such as wind velocity, temperature,
57 pressure, etc allow the development of forecasting rules that predict the fu-
58 ture of these variables only. A regional set of observations with appropriate
59 sensors allows one to forgo standard considerations of spatial resolution and
60 uncertainties in the properties of detailed physical models. Present global
61 or regional models require specification of details of physical processes glob-
62 ally or regionally, and the ensuing, often heavy, computational requirements
63 provide information of the time variation of many quantities not of interest
64 locally. In this paper we formulate the construction of data driven forecast-
65 ing (DDF) models of geophysical processes and demonstrate how this works
66 within the familiar example of a ‘global’ model of shallow water flow on a
67 mid-latitude β plane. A sub-region, where observations are made, of the
68 global flow is selected. A discrete time dynamical forecasting system is con-
69 structed from these observations. DDF forecasting accurately predicts the
70 future of observed variables.



1 Introduction

In geophysical models for global and regional weather or climate forecasting, solutions of the Navier-Stokes equations, expressing conservation of mass, and momentum, are accompanied by a thermodynamic equation describing energy conservation. An equation of state relating the thermodynamic state variables to each other is required as are further parameterizations to represent unresolved Physics below the model grid scale. Cloud moisture dynamics is a critical example of the latter.

Numerical solutions for these partial differential equations (PDEs) using, for example, a finite difference method Press et al. (2007); Olver (2020) lead to formulations with a very large number of ordinary differential equations (ODEs) at a global set of spatial grid points. In global models the number of these degrees of freedom (ODEs) may range from 10^8 to 10^{10} while for regional models this may still be large, say 10^6 to 10^7 . Even then the resolution of the largest operational GCMs is today about 9 km in the horizontal Stevens et al. (2019). Models with scales down to 1.4 km are being developed and tested Wedi et al. (2020), and the computational challenges grow as these are realized.

If one's interest is in forecasting the weather or climate only in a selected region, another point of view may be employed. This builds the relevant dynamics using only observations of the state variables (pressure, velocities, temperatures, ...) one wishes to forecast. Knowledge of the forcing of the system at the location of the observations is required, but this is already estimated in the formulation of the global models Roeckner et al. (2003); Staff (2021).

Working from data alone avoids uncertainties in initial conditions for the ODEs, uncertain physical features of the models and their boundary conditions, and the like. It also circumvents the growing computational complexity as the spatial resolution of big models is increased.

Not surprisingly, one loses something in a formulation that bypasses knowledge of the fundamental physical dynamical equations of the problem. At the same time the computations for forecasting observables only is enormously simplified compared to calculating a full set of physical properties in a region or globally.

The subject of this paper is a method, which we call Data Driven Forecasting (DDF). It is a combination of applied mathematical tools that were well developed some time ago Hardy (1971); Micchelli (1986); Broomhead



108 and Lowe (1988); Schaback (1995); M. D. J. (2002); Buhmann (2009) with
 109 the essential Physics of how one can learn properties of nonlinear dynamical
 110 systems through observations of a subset of the dynamical variables of
 111 that system Takens (1981); Eckmann and Ruelle (1985); Sauer et al. (1991);
 112 Abarbanel (1996); Kantz and Schreiber (2004).

113 **1.1 Recent Work in Using Machine Learning Methods** 114 **in Weather and Climate Modeling**

115 The interest in using data driven modeling in earth systems problems is
 116 both intense and productive. It is well documented in a recent theme issue
 117 of the Proceedings of the Royal Society A entitled “Machine Learning for
 118 weather and climate modeling” (The articles can be accessed directly at
 119 www.bit.ly/TransA-2194) Chantry et al. (2021).

120 Efforts are directed toward replacing large numerical weather forecast-
 121 ing GCMs with data trained networks (machine learning or ML) Shi et al.
 122 (2015) Comrie (1997) Krasnopolsky et al. (2002) Geer (2021) Dueben and
 123 Bauer (2018). While there is success in these investigations, there remains
 124 hesitancy in the climate and weather forecasting community to openly ac-
 125 cept these methods as they are viewed as black boxes that have tenuous
 126 relation to the physics that they model other than the data that is provided
 127 to them Schultz et al. (2021). It is for reasons like this that there have been ef-
 128 forts in ML research to investigate methods with a large focus on the efficacy
 129 and trustworthiness of ML tools Mackowiak et al. (2021) Huang et al. (2020).
 130 The ML community in weather forecasting have gone an additional step fur-
 131 ther by implementing hybrid models that are machine learning devices that
 132 include physical constraints in some form Wandel et al. (2020) Grover et al.
 133 (2015) Daw et al. (2017). There have been studies showing that weather
 134 forecasting tools that account for physical constraints (typically in their loss
 135 functions) show improvement over ML tools that utilize restricted knowl-
 136 edge of the underlying physics Kashinath et al. (2021). In summary, there
 137 is both an interest and benefit to the hybridization of physics and ML mod-
 138 els Watson-Parris (2021); Brajard et al. (2021).

139 In this paper we follow a complementary but different path in realizing
 140 nonlinear, dynamical forecasting rules based on observed data alone. We
 141 call it data driven forecasting or DDF. We argue that the DDF modeling
 142 strategy we describe in this paper embodies the hybridization of physics and



143 ML in a way not typically performed to reap the forecasting benefits while
 144 maintaining a high degree of transparency in how it actually operates.

145 DDF as will be shown in its formulation and explicitly in a simple, fa-
 146 miliar geophysical model incorporates both the physics of the underlying
 147 model and ML tools to create an update rule for forecasting. Unlike many
 148 hybridization ML models, it does not enforce a physics constraint in its train-
 149 ing function. The training of the ML model proceeds as standard regularized
 150 ridge regression. The update rule for DDF is a sum of radial basis functions
 151 and physically inspired terms which include, but are not limited to, forcing
 152 terms and polynomial variables in the original model. The separation of the
 153 forcing of the intrinsic geophysical fluid dynamics is both a feature of the
 154 underlying fluid dynamical equations and makes testing of the generalization
 155 to innovative forcing straightforward.

156 As we use DDF in this paper, a subset of the degrees of freedom of a
 157 ‘global’ model, which for purposes of illustration and simplicity is taken as
 158 a one layer shallow water flow on a β -plane, are considered observed. Then
 159 using the familiar method of time delay embedding Takens (1981); Abarbanel
 160 (1996), we introduce the physics of the global flow into the regional forecast.
 161 It is clear that this paper represents a proof of principle in using DDF and
 162 addressing how the method scales to large, increasingly realistic models is
 163 still to be accomplished. In using the DDF forecasting formulation on field
 164 data there is a requirement for comparison of computational complexity to
 165 mainstream GCMs that is an important part of continuing work.

166 1.2 Plan for this Paper

- 167 1. To begin we describe the DDF method and its forecasting goals in
 168 the context of a simple fluid dynamics problem of geophysical interest:
 169 shallow water flow (SWE) Jiang et al. (1994); Pedlosky (1986); Vallis
 170 (2017).
- 171 2. The discussion of the SWE example illustrates the key ideas in a con-
 172 crete example and provides us the basis for a general consideration of
 173 the problem of learning from observations alone how to forecast the
 174 future development of those observations.
- 175 3. We then return to the SWE example to show explicitly how the DDF
 176 method is implemented and performs.



- 177 4. A Summary and Discussion completes the main body of the paper.
- 178 5. An Appendix contains many of the details describing how one estimates
179 the parameters in a DDF model. The discussion is more general than
180 the simple geophysical examples in the body of the paper and may be
181 widely useful beyond this paper.

182 2 Shallow Water Flow on a β plane used to 183 illustrate DDF

184 To illustrate the main ideas in this paper we begin with a familiar example
185 which, though simple, illustrates the ideas in a useful context. The example
186 is shallow water flow on a β plane, and this is introduced as the *global* model
187 of a geophysical flow. This is two dimensional flow with state or dynamical
188 variables of velocities in the x and y direction and the height of the fluid:
189 $\{v_1(\mathbf{r}, t), v_2(\mathbf{r}, t), \zeta(\mathbf{r}, t)\} = \{u(\mathbf{r}, t), v(\mathbf{r}, t), \zeta(\mathbf{r}, t)\}$. $\mathbf{r} = [r_1, r_2] = [x, y]$. The
190 partial differential equations of shallow water flow are solved on a two dimen-
191 sional global grid in a finite difference approximation for the state variables
192 $\{v_1(i, j, t), v_2(i, j, t), \zeta(i, j, t)\}$ with the integers (i, j) denoting locations on
193 the grid.

194 We then select a subgrid as the region where we record observations of
195 the state variables $\{v_1(I, J, t), v_2(I, J, t), \zeta(I, J, t)\}$ with the integers (I, J)
196 denoting locations on the regional subgrid.

197 In this paper we address how data from these regional measurements
198 alone, without knowledge of the underlying dynamical equations or knowl-
199 edge of the states of the global system outside the subgrid, can allow us to
200 forecast the states of the regional variables $\{v_1(I, J, t), v_2(I, J, t), \zeta(I, J, t)\}$.

201 In applying the methods developed here one needs only measurements
202 of the state variables we wish to forecast at selected spatial locations in a
203 sub-region. There is no ‘grid’ where we must place the sensors for the desired
204 observations.

205 We do not require information about the state of the system outside the
206 region. The forcing of the fluid in the selected region is required. That is
207 already in the global formulation of the problem Roeckner et al. (2003); Staff
208 (2021).



2.1 Fluid Equations on a Grid; The Global Model

We initiate our considerations with the one layer Shallow Water Equations (SWE) on a mid-latitude β plane Jiang et al. (1994); Pedlosky (1986); Vallis (2017). The plane has coordinates $\mathbf{r} = \{r_1, r_2\} = \{x, y\}$, and the dynamical variables of the flow are the velocities in the x and the y directions $\{v_1(\mathbf{r}, t) = u(\mathbf{r}, t), v_2(\mathbf{r}, t) = v(\mathbf{r}, t)\}$ and the fluid height $\zeta(\mathbf{r}, t)$. $\nabla_\perp = (\frac{\partial}{\partial r_1}, \frac{\partial}{\partial r_2})$. The SWE take the form

$$\begin{aligned} \frac{\partial \zeta(\mathbf{r}, t)}{\partial t} + \nabla_\perp(\zeta(\mathbf{r}, t)\mathbf{v}(\mathbf{r}, t)) &= 0 \\ \frac{\partial \mathbf{v}(\mathbf{r}, t)}{\partial t} + \mathbf{v}(\mathbf{r}, t) \cdot \nabla_\perp \mathbf{v}(\mathbf{r}, t) + f(\mathbf{r})\hat{z} \times \mathbf{v}(\mathbf{r}, t) &= -g\nabla_\perp \zeta(\mathbf{r}, t) \\ + A\nabla_\perp^2 \mathbf{v}(\mathbf{r}, t) - \epsilon \mathbf{v}(\mathbf{r}, t) + \mathcal{F}(\mathbf{r}, t). \end{aligned} \quad (1)$$

The pressure is given by the hydrostatic relation $p(\mathbf{r}, z, t) = g\rho(\zeta(\mathbf{r}, t) - z)$; $p(\mathbf{r}, \zeta(\mathbf{r}, t), t) = 0$. Body forces driving the fluid are $\mathcal{F}(\mathbf{r}, t) = [\mathcal{F}_1(\mathbf{r}, t), \mathcal{F}_2(\mathbf{r}, t)]$. We solve the SWE on a $n_x \times n_y$ grid $\{r_1 = r_{10} + i\Delta x, r_2 = r_{20} + j\Delta y\}$; $i = 0, 1, \dots, n_x - 1$; $j = 0, 1, \dots, n_y - 1$ using, for example, periodic boundary conditions

$$v(0, y, t) = v(L_x, y, t); u(x, 0, t) = u(x, L_y, t); L_x = n_x \Delta x; L_y = n_y \Delta y. \quad (2)$$

The fluid density is constant and chosen to be $\rho = 1\text{kg/m}^3$. $f(\mathbf{r}) = f_0 + \beta r_2$ is the local rotation of the earth, A is an effective kinematic viscosity, ϵ is a Rayleigh friction coefficient.

The solution on this grid comprises our *global* dynamics. There are $D_G = 3(n_x \times n_y)$ ODEs for the states $\mathbf{S}(i, j, t) = \{v_1(i, j, t), v_2(i, j, t), \zeta(i, j, t)\}$:

$$\frac{d\mathbf{S}(i, j, t)}{dt} = \mathbf{F}_{i,j}(\mathbf{S}(i, j, t), \theta) + [\mathcal{F}(i, j, t), 0]. \quad (3)$$

θ are fixed parameters in the global system. $\mathbf{F}_{i,j}(\mathbf{S}, \theta)$ is the vector field of the D_G (global) nonlinear differential equations for the shallow water flow.

2.2 Dynamics on a Subgrid; The Regional Model

Next select a subgrid, **our region**, where we make measurements. The locations in the observation region are denoted by $\mathbf{R} = \mathbf{R}_0 + \{R_{10} + I\Delta r_1, R_{20} + J\Delta r_2\}$; $I = 0, 1, 2, \dots, N_x - 1, J = 0, 1, 2, \dots, N_y - 1$; $N_x \leq n_x, N_y \leq n_y$. It is



in this region that we collect observations at a subset of the full complement
 of state variables $\mathbf{S}(\mathbf{r}, t)$.
 These observations are $\mathbf{O}(\mathbf{R}, t) = \mathbf{O}(I, J, t) = \{v_1(I, J, t), v_2(I, J, t), \zeta(I, J, t)\}$
 which satisfy

$$\begin{aligned}\frac{d\mathbf{O}(\mathbf{R}, t)}{dt} &= F_{\mathbf{R}}(\mathbf{S}(\mathbf{r}, t), \theta) + [\mathcal{F}(\mathbf{R}, t), 0], \\ \frac{d\mathbf{O}(I, J, t)}{dt} &= F_{I,J}(\mathbf{S}(i, j, t), \theta) + [\mathcal{F}(I, J, t), 0]\end{aligned}\quad (4)$$

The number of regional observed variables is $D_R = 3(N_x \times N_y) \leq D_G$.
 $F_{\mathbf{R}}(\mathbf{S}(\mathbf{r}, t), \theta) = F_{I,J}(\mathbf{S}, \theta)$ is the (global) vector field restricted to the region
 \mathbf{R} . It is a function of the states of the global dynamics $\mathbf{S}(i, j, t)$.

Using data from observations on the $\mathbf{O}(\mathbf{R}, t)$, without knowledge of the
 vector field $F_{I,J}(\mathbf{S}(i, j, t), \theta)$, we want to construct a discrete time dynamical
 rule which takes $\mathbf{O}(\mathbf{R}, t)$ forward in time; $\mathbf{O}(\mathbf{R}, t) \rightarrow \mathbf{O}(\mathbf{R}, t + \Delta t) = \mathbf{O}(\mathbf{R}, t +$
 $h)$. This discrete time dynamical map is our forecasting system for the region.

Observations are made at N_O times $t_n = t_0 + nh; n = 0, 1, \dots, N_O - 1$
 giving us $\mathbf{O}(\mathbf{R}, t_n) = \mathbf{O}(\mathbf{R}, n)$ at all those times. These observations form a
 trajectory in $D_R \leq D_G$ dimensional space. We are interested in the situation
 where $D_R < D_G$, giving us observations in a subregion of the global dynam-
 ics. For purposes of explaining the steps in the construction of a regional
 forecasting system we will first consider the case $D_R = D_G$. A return to
 $D_R < D_G$ will follow that discussion.

Now we integrate the regional dynamical equation Eq. (4) over the in-
 terval $[t_n, t_n + h] = [t_n, t_{n+1}]$. This gives us the *flow* of the D_R dimensional
 dynamical system which we find to be

$$\begin{aligned}\mathbf{O}(\mathbf{R}, t_n + h) &= \mathbf{O}(\mathbf{R}, n + 1) = \mathbf{O}(\mathbf{R}, n) \\ &+ \int_{t_n}^{t_n + h} dt' \left\{ F_{\mathbf{R}}(\mathbf{S}(\mathbf{r}, t'), \theta) + [\mathcal{F}(\mathbf{R}, t'), 0] \right\},\end{aligned}\quad (5)$$

As we do not know the vector field $F_{I,J}(\mathbf{S}(\mathbf{r}, t), \theta)$, we must **represent**
 the integral over it in some manner. The integral over the external forces on
 the fluid in the subregion, \mathbf{R} , $[\mathcal{F}(\mathbf{R}, t), 0]$ we can approximate, because, to
 solve the original global problem, we were required to specify how the fluid
 was driven both regionally and globally Roeckner et al. (2003); Staff (2021).
 The forcing of the fluid is an aspect of the flow that is not intrinsic to the



fluid properties themselves, and it is important to observe that this external forcing is additive. Essentially these are just Newton's equations of motion. Using whatever approximation to the integral over $\mathbf{F}_{I,J}(\mathbf{S}(\mathbf{r}, t'), \theta)$ one wishes Olver (2017) that establishes it as dependent on $\mathbf{S}(\mathbf{r}, t)$, so the resulting dynamical rule for moving forward in time is explicit, we arrive at a representation of the flow as the discrete time map ($1 \leq i \leq n_x, 1 \leq j \leq n_y : 1 \leq I \leq N_x \leq n_x, 1 \leq J \leq N_y \leq n_y$.)

$$\begin{aligned} \mathbf{O}(I, J, n+1) &= \mathbf{O}(I, J, n) + \mathbf{f}_{I,J}(\mathbf{S}(i, j, n), \boldsymbol{\chi}) \\ &+ \frac{h}{2} \left[[\mathcal{F}(I, J, n), 0] + [\mathcal{F}(I, J, n+1), 0] \right]. \end{aligned} \quad (6)$$

Although we do not use it in our construction of DDF models, we could have chosen Equation (6) to have $\mathbf{S}(i, j, n+1)$ on the right hand side as we represent the integral over the vector field $\mathbf{F}_{I,J}(\mathbf{S}(i, j, t))$ in Equation (5) Olver (2017). The dynamical discrete time map would then be *implicit* and using it would require the solution of a nonlinear problem at each step.

The trapezoidal rule was used to approximate the integral over the known forces which should be quite adequate as the measurements are known only at intervals of size h . The $\boldsymbol{\chi}$ are constants to be estimated in the representation of the flow of the discrete time dynamics. We call $\mathbf{f}_{I,J}(\mathbf{S}(i, j, n), \boldsymbol{\chi})$ the vector field of the discrete time flow restricted to the regional variables.

2.3 Proceeding in Two Steps

As noted earlier, we proceed now in two steps:

1. First we select $D_R = D_G$ putting us in the situation where we observe all components of the global dynamics, but in which we do not know the equations of these global dynamics but only have the data.
2. Second we move to the case of interest in this paper when $D_R < D_G$ where we are also required to find a way to introduce information about the global dynamical flow into our rule $\mathbf{O}(\mathbf{R}, t) \rightarrow \mathbf{O}(\mathbf{R}, t + \Delta t)$, Equation (6).

2.3.1 First Step, $D_R = D_G$; Introducing Radial Basis Functions

This is the case where we observe all of the global set of state variables $\mathbf{S}(\mathbf{r}, n)$. This is not of central interest to us, but it is nonetheless very instructive for



Parameter	Symbol	Value
Fluid Density	ρ	1 kg/m ³
Coriolis Parameter	$f_0 + \beta y$	$f_0 = 10^{-5} 1/s$ $\beta = 10^{-12} 1/m s$
Effective Viscosity	A	100 m ² /s
Rayleigh friction	ϵ	10 ⁻⁸ 1/s
Gravity	g	9.8 m/s ²
Time Step	h	6 min
Resting Fluid Depth	H_0	50 m
Grid Resolution	$\Delta x = \Delta y$	100 km
Domain Extent	$L_x = n_x \Delta x; L_y = n_y \Delta y$	
Body Forcing	$[\mathcal{F}_1(\mathbf{r}, t), \mathcal{F}_2(\mathbf{r}, t)]$ $F_1(\mathbf{r}, t) = -F_0 \cos(\frac{2\pi y}{L_y})$	$[F_1(\mathbf{r}, t), 0] m/s^2$ $F_0 = 10^{-5} m/s^2$
Number of Centers	N_c	1000

Table 1: Table of Parameters and Symbols in the shallow water flow. For the RBF representation we have additional parameters χ used in the solution to the DDF flow vector field. These depend on the particular problem, e.g. which subregion one selects. The χ include the RBF Shape Factor, σ , the time Time Delay $\tau = T_h h$; ($T_h = \text{integer}$), the Embedding Dimension, D_E , and the Tikhonov Regularization Parameter, B . We have used $N_c = 1000$.



288 making our next step.

289 The discrete time dynamics appears in this instance as

$$\begin{aligned} \mathbf{S}(i, j, n+1) &= \mathbf{S}(i, j, n) + \mathbf{f}_{i,j}(\mathbf{S}(r, n), \boldsymbol{\chi}) \\ &+ \frac{h}{2} \left[[\mathcal{F}(\mathbf{r}, n), 0] + [\mathcal{F}(\mathbf{r}, n+1), 0] \right], \end{aligned} \quad (7)$$

290 and we need a representation of the discrete time vector field $\mathbf{f}_{i,j}(\mathbf{S}(i, j, n), \boldsymbol{\chi})$
 291 of the flow.

292 This flow vector field $\mathbf{f}(\mathbf{S}, \boldsymbol{\chi})$ is a set of D_G functions on the $\mathbf{S} \in \mathcal{R}^{D_G}$.
 293 There are many ways to represent such functions of many variables. We can
 294 think of the observed samples as points of information about a distribution
 295 $\mathbf{f}(\mathbf{S}, \boldsymbol{\chi})$ and ask that the representation give us an interpolating function
 296 among the observed point locations $\mathbf{S}(\mathbf{r}, t_n) = \mathbf{S}(\mathbf{r}, n)$.

297 We select a method of representing this function using radial basis func-
 298 tions (RBFs) Hardy (1971); Micchelli (1986); Broomhead and Lowe (1988);
 299 Schaback (1995); M. D. J. (2002); Buhmann (2009). In using this method
 300 we use a K-means clustering algorithm Du and Swamy (2006) to select N_c of
 301 all the N_O observed locations in \mathbf{S} space. These are called centers. The flow
 302 vector field is then given in components by

$$f_a(\mathbf{S}, \boldsymbol{\chi}) = \sum_{k=1}^K c_{ak} p_k(\mathbf{S}) + \sum_{q=1}^{N_c} w_{aq} \psi((\mathbf{S} - \mathbf{S}(q))^2, \sigma), a = 1, 2, \dots, D_G \quad (8)$$

303 In this $p_k(\mathbf{S})$ is a polynomial of degree k . The radial basis functions
 304 $\psi((\mathbf{S} - \mathbf{S}(q))^2, \sigma)$ are sensitive to the distribution samples, the centers $\mathbf{S}(q)$,
 305 over a range σ .

306 There are a multitude of choices for the RBF $\psi((\mathbf{S} - \mathbf{S}(q))^2, \sigma)$, and we
 307 have investigated the use of two. The Gaussian

$$\psi_G((\mathbf{S} - \mathbf{S}(q))^2, \sigma) = \exp[-R(\mathbf{S} - \mathbf{S}(q))^2]; R = \frac{1}{2\sigma^2}, \quad (9)$$

308 and the multiquadric of Hardy Hardy (1971)

$$\psi_{MQ}((\mathbf{S} - \mathbf{S}(q))^2, \sigma) = \sqrt{(\mathbf{S} - \mathbf{S}(q))^2 + \sigma^2}. \quad (10)$$

309 The constants $\boldsymbol{\chi} = \{c_{ak}, w_{aq}\}$ are determined by the linear algebra prob-



310 lem

$$\begin{aligned} \mathbf{S}(n+1) = \mathbf{S}(n) + \sum_{k=1}^K c_{ak} p_k(\mathbf{S}(n)) + \sum_{q=1}^{N_c} w_{aq} \psi((\mathbf{S}(n) - \mathbf{S}(q))^2, \sigma) \\ + \frac{h}{2} \left[[\mathcal{F}(\mathbf{r}, n), 0] + [\mathcal{F}(\mathbf{r}, n+1), 0] \right], \end{aligned} \quad (11)$$

311 for $n = 0, 1, 2, \dots, N_O - 1$ and $a = 1, 2, \dots, D_G$.

312 2.3.2 Second Step, $D_R < D_G$; Utilizing Time Delay Embedding

313 Turning back to the question of developing a dynamical map for the regional
 314 subset $\mathbf{O}(\mathbf{R}, n)$ of our dynamical variables, we see that the observations are a
 315 projection from dimension $D_G \rightarrow D_R < D_G$. To proceed we require a space
 316 of state variables equivalent to the full state space of the $\mathbf{S}(\mathbf{r}, t)$, so we must
 317 ‘unproject’ the $\mathbf{O}(\mathbf{R}, t)$ to a space equivalent to $\mathbf{S}(\mathbf{r}, t)$.

318 A dynamical method for accomplishing this is well analyzed in the non-
 319 linear dynamics literature. It rests on the fact that as the observed quantities
 320 move from some time $t - \tau$ to time t , they depend on all of the state variables
 321 $\mathbf{S}(\mathbf{r}, t)$ as seen in Equation (6). Using time delays of the observed regional
 322 variables provides us the desired information on the unobserved state vari-
 323 ables.

324 This suggests creating a D_E -dimensional time delay embedding space Tak-
 325 ens (1981); Eckmann and Ruelle (1985); Sauer et al. (1991); Abarbanel
 326 (1996); Kantz and Schreiber (2004) with vectors of dimension $D_{TD} = D_R D_E$

$$\mathbf{TD}(t) = [\mathbf{O}(\mathbf{R}, t), \mathbf{O}(\mathbf{R}, t - \tau), \mathbf{O}(\mathbf{R}, t - 2\tau), \dots, \mathbf{O}(\mathbf{R}, t - (D_E - 1)\tau)]. \quad (12)$$

327 This vector of time delays depends only on the observed quantities in the
 328 region labeled by \mathbf{R} and their time delays. It is through those time delays
 329 that \mathbf{TD} inherits information about the dynamics outside the region \mathbf{R} .

330 Using this time delay vector in the observed dynamics gives us

$$\begin{aligned} \mathbf{O}(\mathbf{R}, n+1) = \mathbf{O}(\mathbf{R}, n) + \\ \mathbf{f}_{\mathbf{R}}(\mathbf{TD}(n), \chi) + \frac{h}{2} \left[[\mathcal{F}(\mathbf{R}, n), 0] + [\mathcal{F}(\mathbf{R}, n+1), 0] \right]. \end{aligned} \quad (13)$$

331 It is useful to write this in components. The $\mathbf{O}(\mathbf{R}, t)$ are D_R dimensional
 332 $\mathbf{O}(\mathbf{R}, t) = \{O_\alpha(t)\}; \alpha = 1, 2, \dots, D_R$. The dynamical map for the regional



333 observables becomes

$$O_\alpha(n+1) = O_\alpha(n) + f_\alpha(\mathbf{TD}(n), \boldsymbol{\chi}) + \frac{h}{2} \left[[\mathcal{F}(\mathbf{R}, n), 0] + [\mathcal{F}(\mathbf{R}, n+1), 0] \right]_\alpha. \quad (14)$$

334 This informs us that we will need a representation for each of the D_R
 335 components of the observed regional variables. The DDF parameters we need
 336 to estimate using the observed data now include $\boldsymbol{\chi} = \{w_{\alpha q}, c_{\alpha j}, \sigma, B, D_E, T_h\}$.

337 **3 Estimating $\boldsymbol{\chi}$ by Minimizing $C(\boldsymbol{\chi})$**

338 The $\mathbf{O}(\mathbf{R}, t)$ are D_R dimensional $\mathbf{O}(\mathbf{R}, t) = \{O_\alpha(t)\}; \alpha = 1, 2, \dots, D_R$. The
 339 dynamical map for the regional observables is

$$O_\alpha(n+1) = O_\alpha(n) + f_\alpha(\mathbf{TD}(n), \boldsymbol{\chi}) + \frac{h}{2} \left[[\mathcal{F}(\mathbf{R}, n), 0] + [\mathcal{F}(\mathbf{R}, n+1), 0] \right]_\alpha$$

$$\mathbf{TD}(t) = [\mathbf{O}(\mathbf{R}, t), \mathbf{O}(\mathbf{R}, t-\tau), \dots, \mathbf{O}(\mathbf{R}, t-(D_E-1)\tau)]. \quad (15)$$

340 To estimate the parameters $\boldsymbol{\chi}$ in the DDF forecasting function, we form
 341 the cost function $C(\boldsymbol{\chi})$ and minimize this objective function with respect to
 342 the elements of $\boldsymbol{\chi} = \{w_{\alpha q}, c_{\alpha j}, R, B, \tau = hT_h, D_E\}$.

$$C(\boldsymbol{\chi}) = \sum_{N_c+1}^{N_O} \left\{ [\mathbf{O}(n+1) - \mathbf{O}(n) - \mathbf{f}(\mathbf{TD}(n), \boldsymbol{\chi}) - \frac{h}{2} \left[[\mathcal{F}(\mathbf{R}, n), 0] + [\mathcal{F}(\mathbf{R}, n+1), 0] \right]] \right\}^2. \quad (16)$$

343 In Equation (16)

$$f_\alpha(\mathbf{TD}(n), \boldsymbol{\chi}) = \sum_{q=1}^{N_c} w_{\alpha q} \psi((\mathbf{TD}(n) - \mathbf{TD}^c(q))^2, \sigma) + \sum_{j=1}^{D_R} c_{\alpha j} \mathbf{O}_j(n) \quad (17)$$

344 Even though we can choose any RBF for ψ , the $C(\boldsymbol{\chi})$ is always linear in
 345 the weights $w_{\alpha q}, c_{\alpha j}$, enabling the use of the linear algebra of Ridge Regression
 346 or Tikhonov regularization in estimating them.



347 The other elements of χ enter the minimization of $C(\chi)$ nonlinearly.
348 There are many excellent ways to search for values for these using ideas noted
349 in the Appendix. We utilized a rather coarse grained grid search to identify
350 good forecasts, then refined the search in those regions. This works well, as
351 the results we present below demonstrate. Nonetheless, we recommend more
352 efficient methods. Again, see the Appendix.

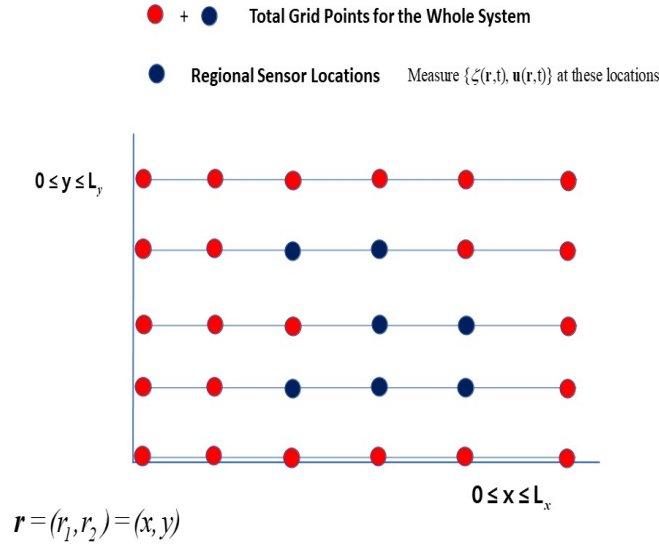


Figure 1: Overall Scheme of Regional DDF Forecasting in the context of the SWE. Red dots are grid points of the overall dynamical system, Eq. (3). Blue dots are the location of the “regional sensors” where fluid height $\zeta(\mathbf{r}, t)$ and fluid velocity $\mathbf{v}(\mathbf{r}, t) = \{v_1(\mathbf{r}, t), v_2(\mathbf{r}, t)\}$ are recorded. If there are N_R regional sensor points then there are $D_R = 3N_R$ regional measured time series. The spatial locations for the sensors are denoted \mathbf{R} ; these are the blue locations on the grid. In this Figure $n_x \times n_y = 30$; $N_R = 7$. In this example there would then be $D_R = 3N_R = 21$ observed regional time series out of $D_G = 3(n_x \times n_y) = 90$ global time series. $D_R < D_G$.

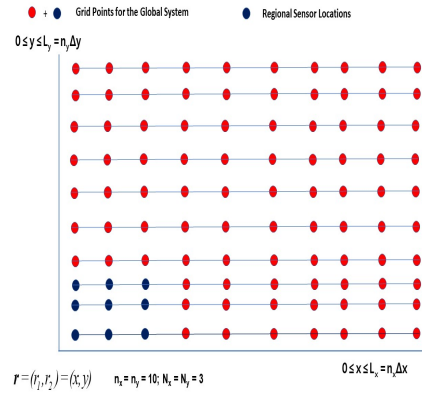


Figure 2: Shallow Water Equations on a $(n_x = 10) \times (n_y = 10)$ grid. $D_G = 3(n_x \times n_y)$. Red dots are grid points of the overall global dynamical system, Eq. (3), a set of Shallow Water Equations on a β plane. Blue dots are the location of the "regional sensors" where fluid height $\zeta(\mathbf{r}, t)$ and fluid velocity $\mathbf{v}(\mathbf{r}, t)$ are recorded. There are $(N_x = 3) \times (N_y = 3) = 9$ regional sensor locations in this example, and $D_R = 3(N_x \times N_y) = 27$ measured time series. The spatial locations of the sensors are denoted \mathbf{R} ; these are the blue locations on the grid. In this Figure $n_x \times n_y = 100$, and $D_G = 3(n_x \times n_y) = 300$. We have 27 observed regional time series out of 300 global time series. $D_R < D_G$.

353 4 Results from the Example of the SWE on 354 a β Plane

355 4.1 Clustered Sensor Region; 3x3 Corner

356 The shallow water equations were solved using the method of Sadourny Sadourny
 357 (1975) with forcing, viscous dissipation, Coriolis forces, and Rayleigh friction
 358 as indicated in Equation (1). We used periodic boundary conditions.

359 The global grid is $n_x = n_y = 10$, so we solved 300 ODEs to generate the
 360 time series for the state variables $\mathbf{S}(\mathbf{r}, t)$ on this grid. A regional sub grid
 361 with $N_x = N_y = 3$ was selected, and on this regional subgrid we 'measured'
 362 $\{u(\mathbf{R}, t), v(\mathbf{R}, t), \zeta(\mathbf{R}, t)\}$ to form the $D_R = 27$ dimensional observation vec-
 363 tors $\mathbf{O}(\mathbf{R}, t)$. The regional sensor locations are shown in Figure (2). We

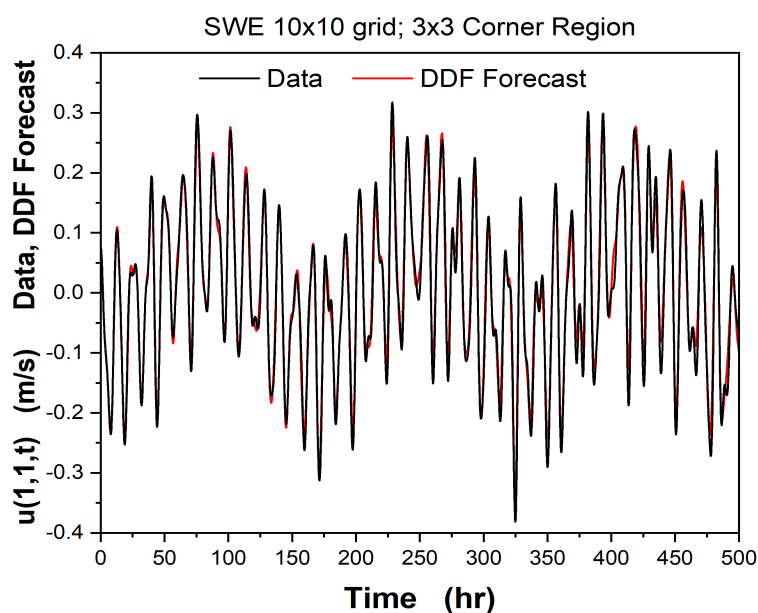


Figure 3: SWE on a 10×10 grid. The sensor region is comprised of 9 locations, blue dots in Figure (2), in a 3x3 corner location. The sensor region is comprised of 9 locations, blue dots in Figure (8). We display the data and the DDF forecast for the x-velocity $u(1,1,t)$. The time delay parameters here are $D_E = 20$, $\tau = 20\Delta t$ with $\Delta t = h = 0.1hr$.

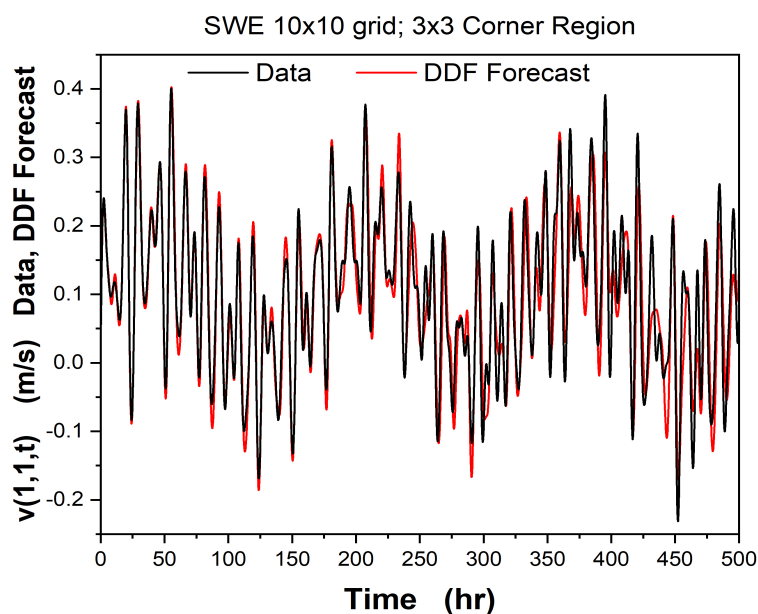


Figure 4: SWE on a 10×10 grid. The sensor region is comprised of 9 locations, blue dots in Figure (2), in a 3×3 corner location. We display the data and the DDF forecast for the y-velocity $v(1,1,t)$. The time delay parameters here are $D_E = 20, \tau = 20\Delta t$ with $\Delta t = h = 0.1hr$.

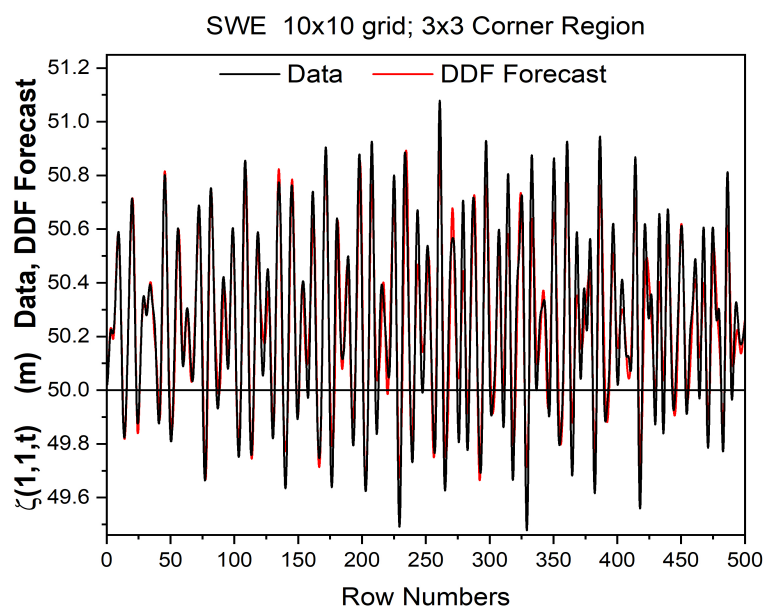


Figure 5: SWE on a 10×10 grid. The sensor region is comprised of 9 locations, blue dots in Figure (2), in a 3x3 corner location. We display the data and the DDF forecast for the fluid height $\zeta(1, 1, t)$. The fluid rest height is $H_0 = 50\text{m}$. The time delay parameters here are $D_E = 20, \tau = 20\Delta t$ with $\Delta t = 0.1\text{hr}$.

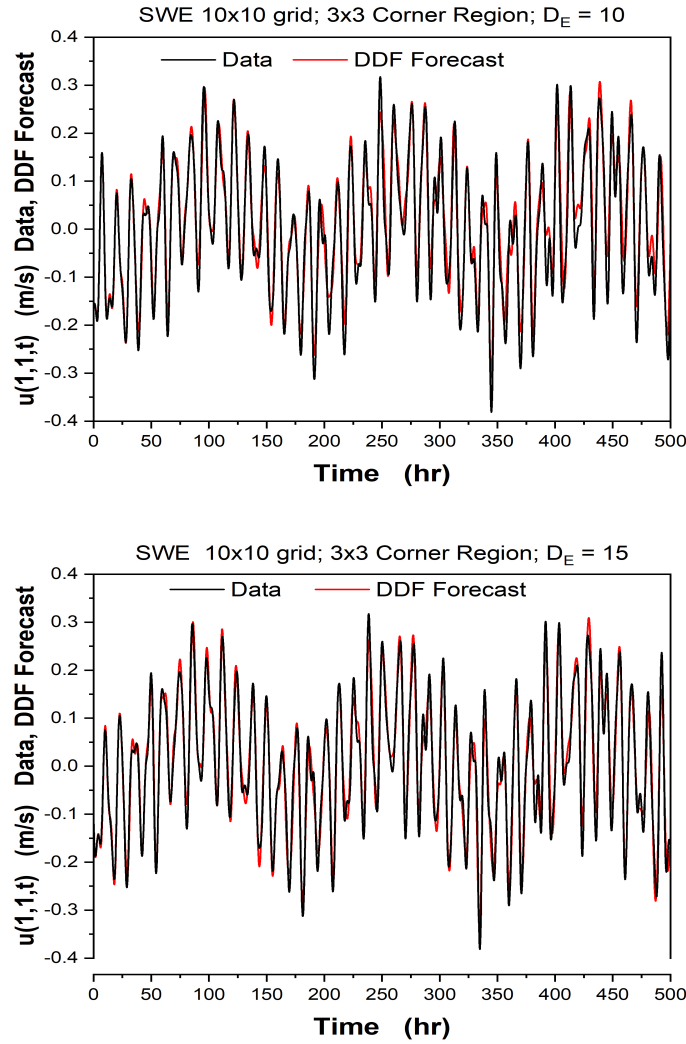


Figure 6: SWE on a 10×10 grid. The sensor region is comprised of 9 locations, blue dots in Figure (2), in a 3x3 corner location. We display the data and the DDF forecast for the x-velocity $u(1,1,t)$. $\tau = 20\Delta t$ with $\Delta t = h = 0.1hr$. **Top Panel** $D_E = 10$ **Bottom Panel** $D_E = 15$. The results with $D_E = 20$ are in Figure (3).

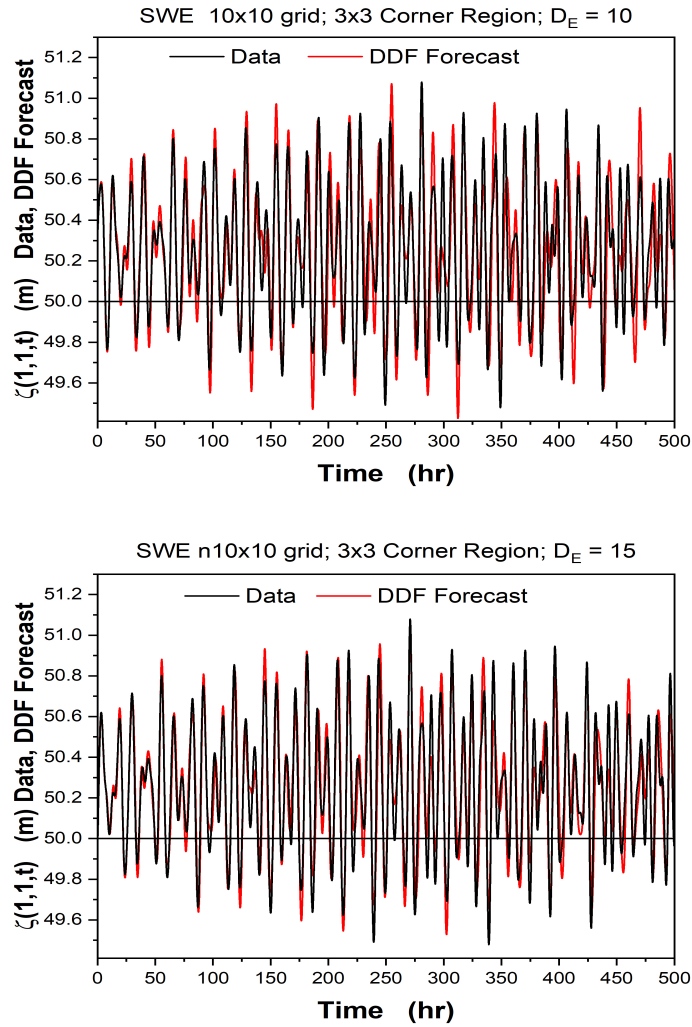


Figure 7: SWE on a 10×10 grid. The sensor region is comprised of 9 locations, blue dots in Figure (2), in a 3x3 corner location. We display the data and the DDF forecast for the fluid height $\zeta(1, 1, t)$. The fluid rest height is $H_0 = 50\text{m}$. $\tau = 20\Delta t$ with $\Delta t = h = 0.1\text{hr}$. **Top Panel** $D_E = 10$ **Bottom Panel** $D_E = 15$. The results with $D_E = 20$ are in Figure (5).



364 generated $N_O = 15,000$ time steps of size 0.1 hr. $N_c = 1000$ centers were se-
365 lected from these data. These were used in a Polynomial plus Gaussian RBF,
366 Equation (8). 1000 hr of these data were used to train the RBF representa-
367 tion of the discrete time flow vector field, the 500 hr of forecasts were made
368 of the 9 regional state variables.

369 In the estimation of the parameters we found $B = 1.0 \times 10^{-9}$, $R = 1.0 \times$
370 10^{-6} , so $\sigma \approx 707$, $D_E = 20$ and $\tau = 20h = 2hr$ gave the best forecasts. See
371 Figure (3), Figure (4), and Figure (5) for the data and the DDF Forecast for
372 $u(1, 1, t)$, $v(1, 1, t)$, and $\zeta(1, 1, t)$ with $D_E = 20$.

373 We note that values of the embedding dimension $10 \leq D_E \leq 20$ gave
374 forecasts of more or less equal quality. In Figure (6) and then Figure (7) we
375 display the x-velocity $u(1, 1, t)$ for $D_E = 10$ and $D_E = 15$ respectively, and
376 then $\zeta(1, 1, t)$ for $D_E = 10$ and $D_E = 15$ respectively.



State Variable	D_E	RMS Error
$u(1,1,t)$	10	0.034 m/s
$u(1,1,t)$	15	0.028 m/s
$u(1,1,t)$	20	0.02 m/s
$\zeta(1,1,t)$	10	0.37 m
$\zeta(1,1,t)$	15	0.37 m
$\zeta(1,1,t)$	20	0.32 m

Table 2: 3x3 Corner example, Figure (2). RMS error in DDF forecasting of the state variable $u(1,1,t)$ and $\zeta(1,1,t)$ as we vary D_E . These are evaluated in the prediction region; $N = 5000$ h = 500 hr.

377 The RMS error is evaluated as

$$\begin{aligned}
 RMS(u) &= \sqrt{\frac{1}{N} \sum_{n=1}^N \left[u_{Data}(1,1,n) - u_{DDF}(1,1,n) \right]^2} \text{ m/s}, \\
 RMS(\zeta) &= \sqrt{\frac{1}{N} \sum_{n=1}^N \left[\zeta_{Data}(1,1,n) - \zeta_{DDF}(1,1,n) \right]^2} \text{ m}, \quad (18)
 \end{aligned}$$

378 and the results are in Table 2.

379 In the underlying mathematical theorem associated with time delay em-
 380 bedding Takens (1981); Eckmann and Ruelle (1985); Sauer et al. (1991);
 381 Abarbanel (1996); Kantz and Schreiber (2004) there is no restriction on the
 382 time delay τ . From a geometrical result one knows that if the dimension
 383 of $\mathbf{TD}(n)$, namely D_E is large enough, then the unprojection we wish to
 384 achieve is accomplished. The results of Sauer et al. (1991) indicate that if
 385 the dimension is larger than $2D_A + 1$, with D_A the information dimension of
 386 the strange attractor, the unprojection will work. We do not know the D_A
 387 of the global SWE model, but if it is near 300 or so, then $D_E \approx 10 - 20$ is a
 388 consistent with this.

389 In the next Figures, we compare the DDF regional forecasts with the
 390 ‘data’ comprised of the solutions to the 300 SWE ODEs at the regional grid
 391 points. In this first set of results, we choose the sensor region to be comprised
 392 of small clusters of sites.

393 In Figure (3) we display the x-velocity $u(1,1,t)$ at the regional grid loca-
 394 tion (1,1) as forecast by the DDF dynamics trained as indicated. In Figure



395 (4) we make the same comparison for the y-velocity $v(1, 1, t)$ at the (1,1)
 396 regional grid location. In Figure (5) the same comparison is made for the
 397 height of the fluid $\zeta(1, 1, t)$. The height of the fluid when at rest is $H_0 = 50m$.

398 4.2 Region With Sparse, Distributed Sensors

399 In the next set of outcomes for DDF regional forecasting, we now select our
 400 sensors to be dispersed over the ‘global’ region. Figure (8) shows ten sensors
 401 dispersed among the 100 grid locations of the global dynamical regime. The
 402 sensor locations were selected at random among the 100 possible sites available
 403 on the ‘global’ grid. The main purpose of this example is to demonstrate that
 404 the sensor sites where $\{v_1(\mathbf{R}, t), v_2(\mathbf{R}, t), \zeta(\mathbf{R}, t)\}$ are observed, in the SWE
 405 example, need not all be contiguous.

406 The success of DDF forecasting using sensors in broadly dispersed sensor
 407 region \mathbf{R} suggests one could use the strategy to forecast in a quite broad
 408 geographical sub-region of a global dynamical system.

409 4.3 Off Center; 2x2 Region

410 Our final example is a 2x2 subregion randomly selected to be off center in
 411 the global 10x10 grid. This is shown in Figure (12).

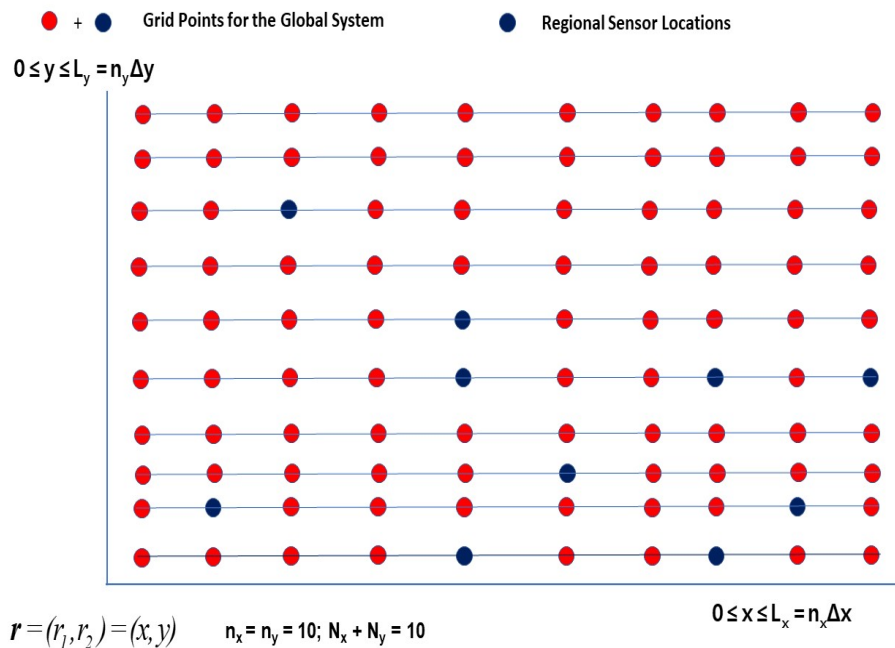


Figure 8: Sparse Regional Sensor Locations. SWE on a 10×10 grid. The sensor region, indicated by blue dots, is comprised of 10 locations selected at random among the 100 global grid points.

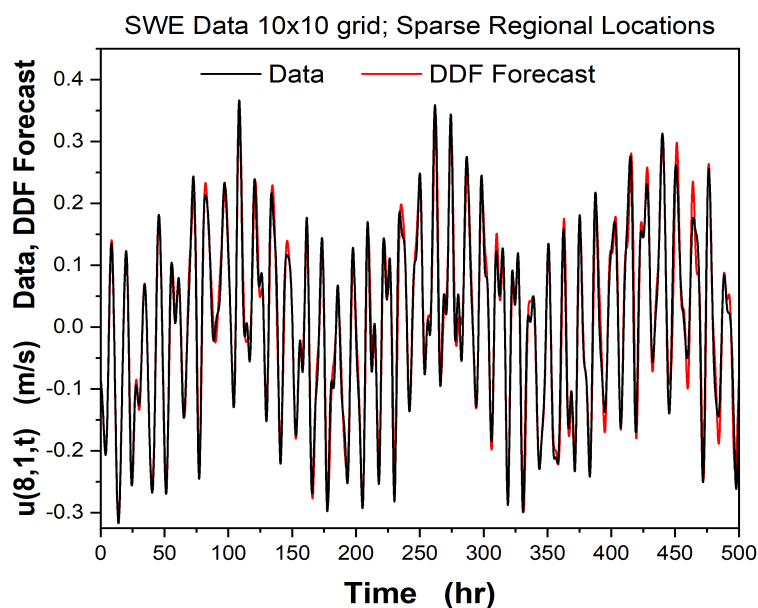


Figure 9: SWE on a 10×10 grid. The sensor region is comprised of 10 locations, blue dots in Figure (8), selected at random among the 100 global grid points. We display the data and the DDF forecast for the x-velocity $u(8, 1, t)$. The time delay parameters here are $D_E = 20$, $\tau = 17h$; $h = 0.1hr$.

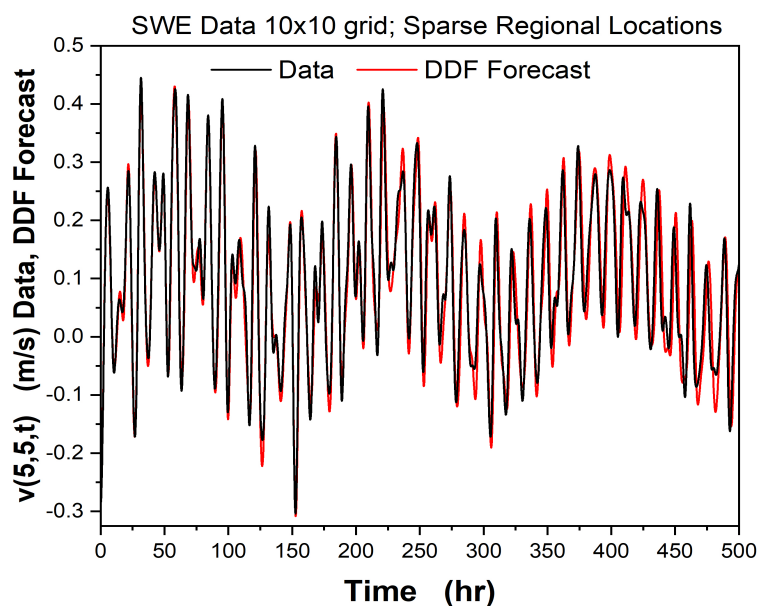


Figure 10: SWE on a 10×10 grid. The sensor region is comprised of 10 locations, blue dots in Figure (8), selected at random among the 100 global grid points. We display the data and the DDF forecast for the y-velocity $v(5,5,t)$. The time delay parameters here are $D_E = 20, \tau = 17\Delta t$ with $\Delta t = 0.1hr$.

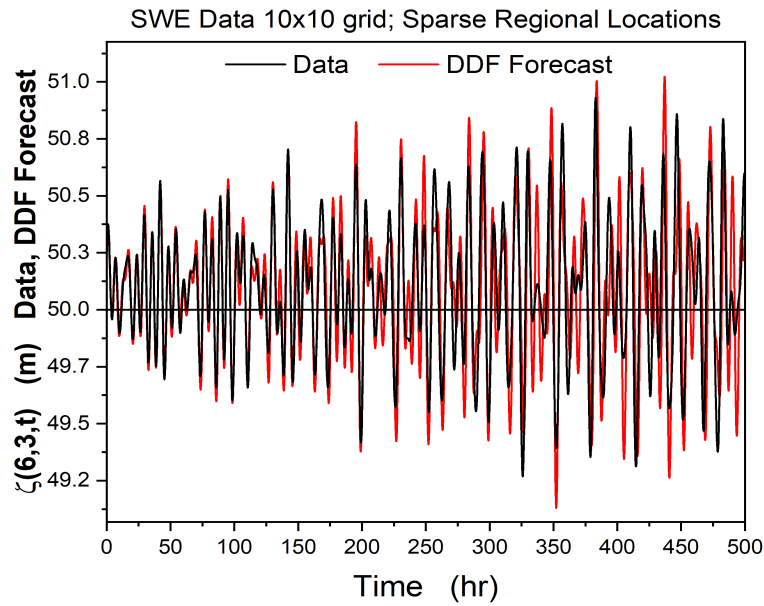


Figure 11: SWE on a 10×10 grid. The sensor region is comprised of 10 locations, blue dots in Figure (8), selected at random among the 100 global grid points. We display the data and the DDF forecast for the fluid height $\zeta(6, 3, t)$. The fluid rest height is $H_0 = 50\text{m}$. The time delay parameters here are $D_E = 20, \tau = 17\Delta t$ with $\Delta t = 0.1\text{hr}$.

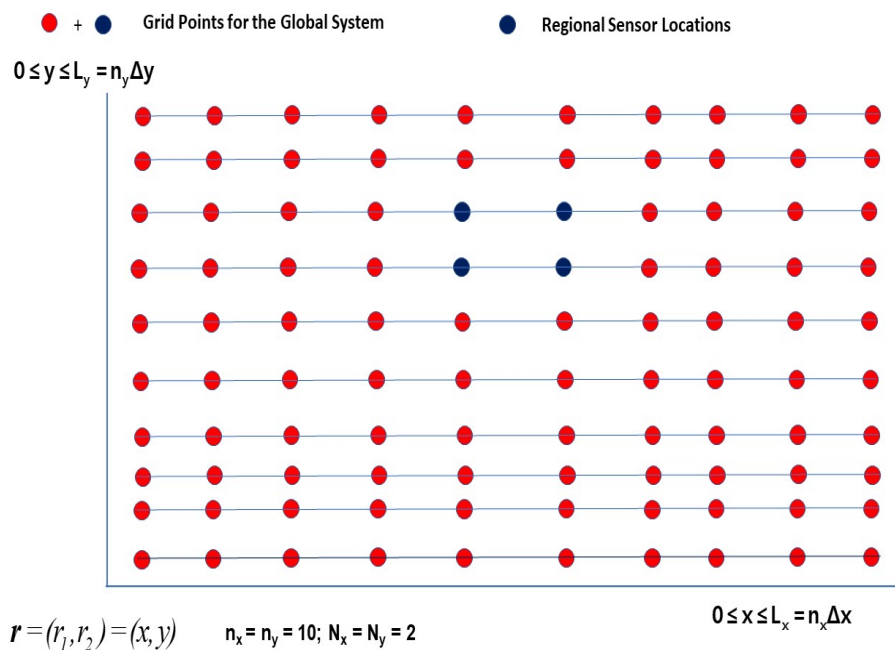


Figure 12: SWE global 10x10 grid. Regional grid is 2x2 and is located off center within the global grid.

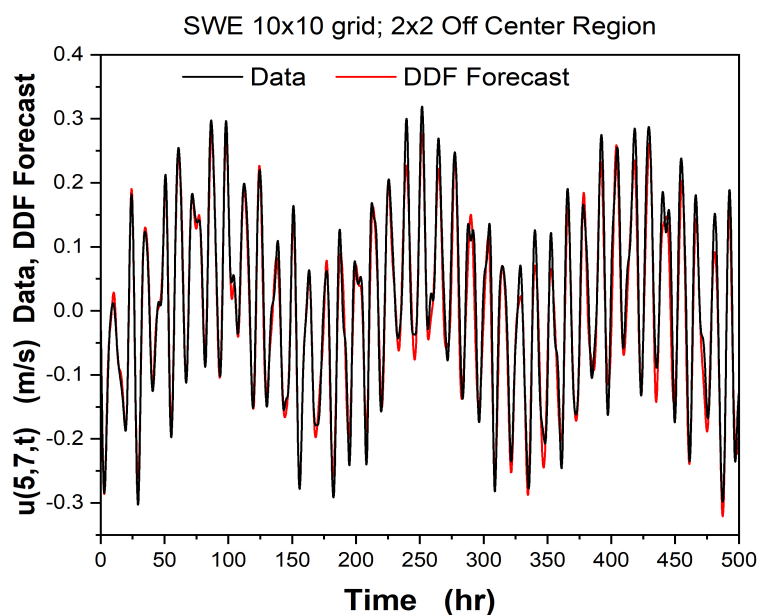


Figure 13: SWE on a 10×10 grid. The sensor region is comprised of 4 locations, blue dots in Figure (8), in a 2×2 off center location. The sensor region is comprised of 4 locations, blue dots in Figure (8), in a 2×2 off center location. We display the data and the DDF forecast for the x-velocity $u(5, 7, t)$. The time delay parameters here are $D_E = 20$, $\tau = 17\Delta t$ with $\Delta t = 0.1hr$.

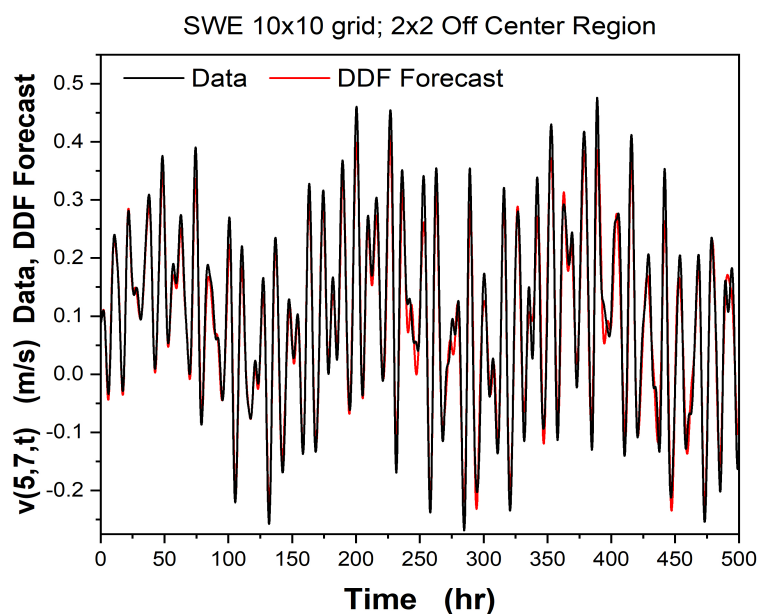


Figure 14: SWE on a 10×10 grid. The sensor region is comprised of 4 locations, blue dots in Figure (8), in a 2x2 off center location. We display the data and the DDF forecast for the y-velocity $v(5, 7, t)$. The time delay parameters here are $D_E = 20, \tau = 17\Delta t$ with $\Delta t = 0.1hr$.

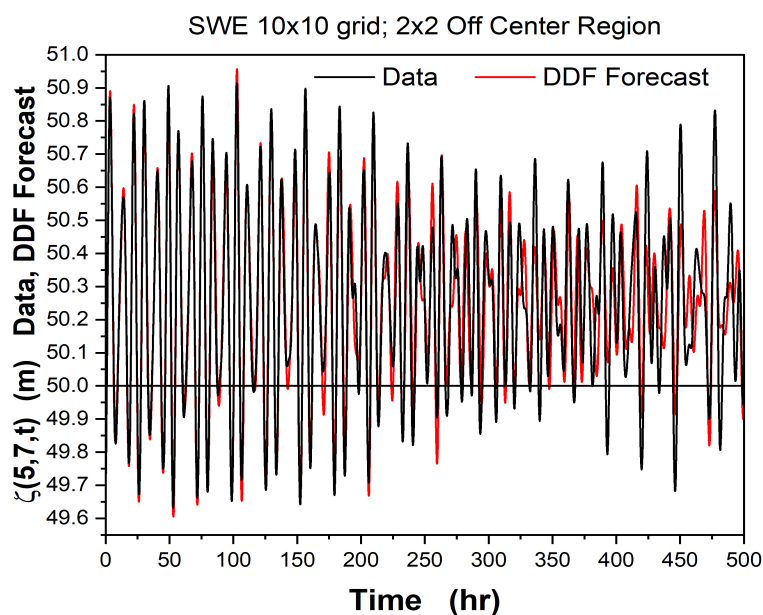


Figure 15: SWE on a 10×10 grid. The sensor region is comprised of 4 locations, blue dots in Figure (8), in a 2x2 off center location. We display the data and the DDF forecast for the fluid height $\zeta(5,7,t)$. The fluid rest height is $H_0 = 50\text{m}$. The time delay parameters here are $D_E = 20$, $\tau = 17\Delta t$ with $\Delta t = 0.1\text{hr}$.

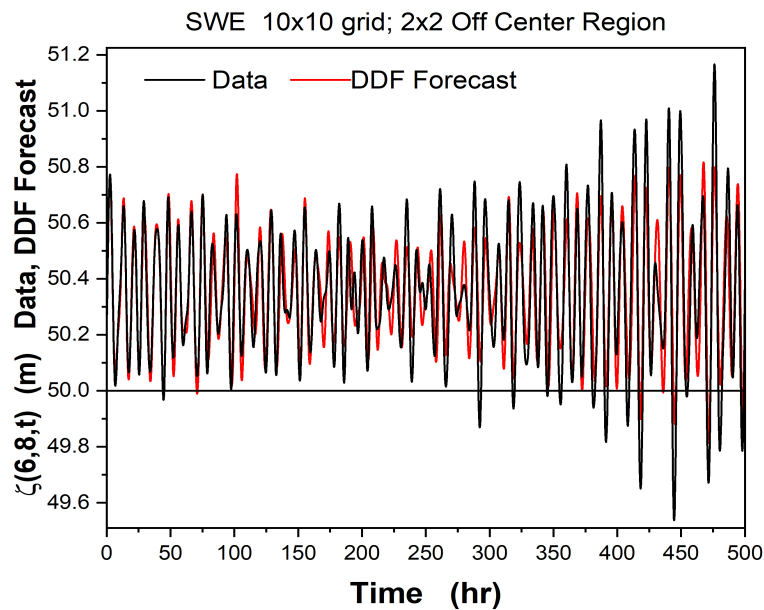


Figure 16: SWE on a 10×10 grid. The sensor region is comprised of 4 locations, blue dots in Figure (8), in a 2x2 off center location. We display the data and the DDF forecast for the fluid height $\zeta(6, 8, t)$. The fluid rest height is $H_0 = 50\text{m}$. The time delay parameters here are $D_E = 20$, $\tau = 17\Delta t$ with $\Delta t = 0.1\text{hr}$.



412 5 Addressing Noisy Data

413 Our twin experiment examples using ‘global data’ from a shallow water flow
 414 have heretofore implicitly assumed we had data with a very high signal to
 415 noise ratio. In this section we take these very clean data and add Gaussian
 416 noise before constructing a DDF model of the regional dynamics.

417 We used the 10x10 SWE data and focused on the 3x3 corner region. For
 418 each of the 27 time series in this region we added Gaussian noise with zero
 419 mean and variance $S\sigma_{\mathbf{O}}^2$. $\sigma_{\mathbf{O}}^2$ is the variance of the signal $\mathbf{O}(\mathbf{R}, t)$ in the
 420 sensor region. In this configuration the signal to noise ratio is $S/N = 1/S$ or
 421 in dB $S/N = 10 \log_{10}[1/S]$. For small S the data is essentially noise free, as
 422 S approaches and exceeds unity, the noise level slowly overcomes the signal.

423 In Figure (17) we show the y-velocity $v(1,1,t)$, data and DDF Forecast
 424 for $S = 0.001$. In Figure (18) the fluid height $\zeta(1,1,t)$ for the same noise
 425 level is shown as the data and the DDF Forecast.

426 In Figure (19) we display the the y-velocity $v(1,1,t)$ data and DDF
 427 Forecast when $S = 0.01$, and in Figure (20), the data and DDF Forecast for
 428 the fluid height $\zeta(1,3,t)$ are shown when $S = 0.1$.

429 As S is increased beyond this, we see significant degradation in the DDF
 430 Forecast relative to the data. In Figure (21) S is now unity, and the DDF
 431 Forecast, as can be seen has become much less accurate. This noise level
 432 corresponds to a signal to noise ratio of 0 dB.

433 We summarize the robustness to added noise in the data in Figure (22)
 434 where the RMS error in the x-velocity is shown for the range of noise level S
 435 we considered.

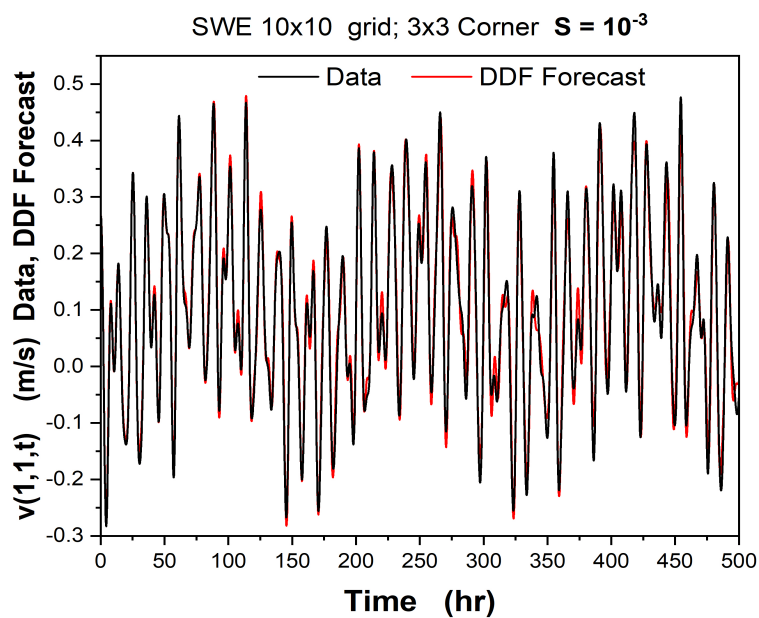


Figure 17: SWE on a 10×10 grid. The sensor region is comprised of 9 locations, blue dots in Figure (2), in a 3x3 off corner location. We display the y-velocity $u(1,1,t)$ data and DDF forecast. $S = 0.001$; $S/N = 30$ dB.

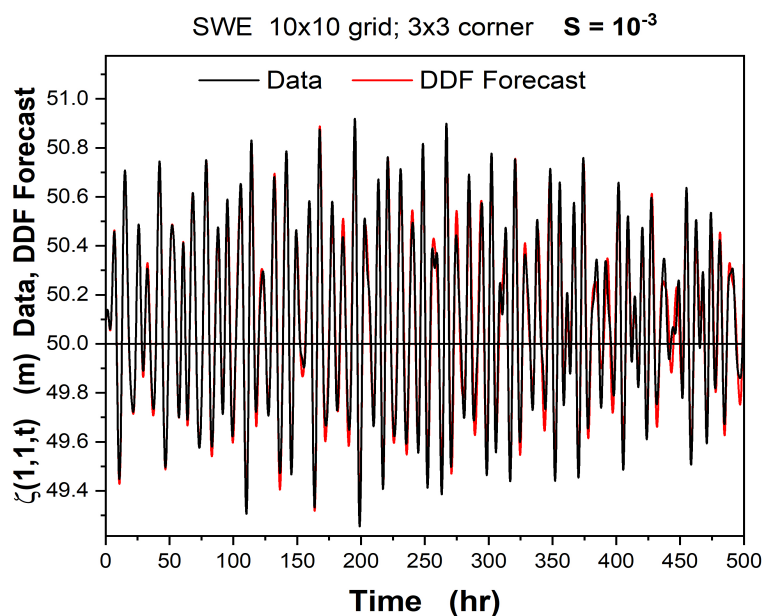


Figure 18: SWE on a 10×10 grid. The sensor region is comprised of 9 locations, blue dots in Figure (2), in a 3x3 off corner location. We display the fluid height $\zeta(1, 1, t)$ data and DDF forecast. $S = 0.001$; $S/N = 30$ dB.

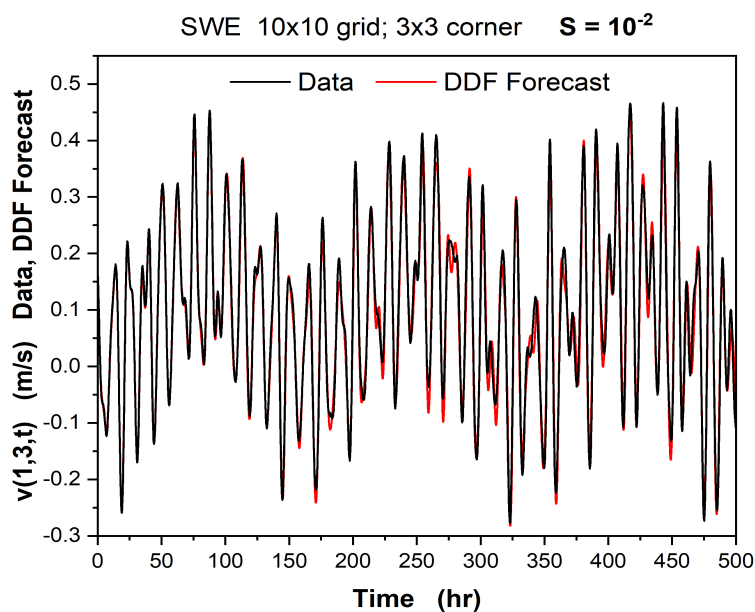


Figure 19: SWE on a 10×10 grid. The sensor region is comprised of 9 locations, blue dots in Figure (2), in a 3x3 off corner location. We display the y-velocity $v(1,3,t)$ data and DDF forecast. $S = 0.01$; $S/N = 20$ dB.

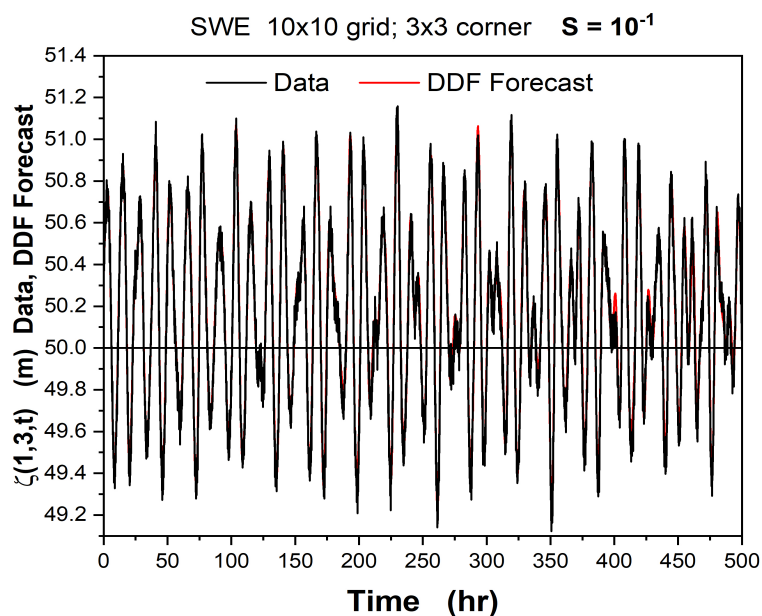


Figure 20: SWE on a 10×10 grid. The sensor region is comprised of 9 locations, blue dots in Figure (2), in a 3x3 off corner location. We display the xfluid height $\zeta(1,3,t)$ data and DDF forecast. $S = 0.1$; $S/N = 10$ dB.

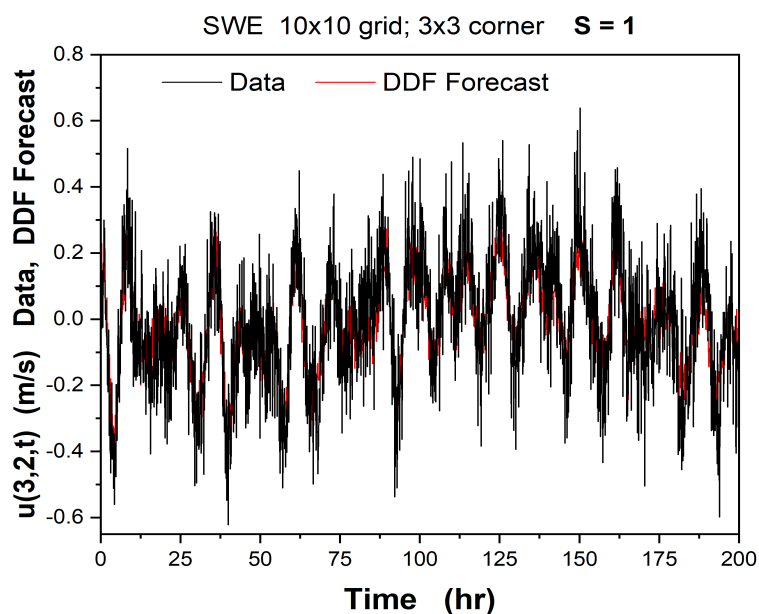


Figure 21: SWE on a 10×10 grid. The sensor region is comprised of 9 locations, blue dots in Figure (2), in a 3x3 off corner location. We display the x-velocity $u(3,2,t)$ data and DDF forecast. $S = 1$; $S/N = 0$ dB. At $S = 1$ the DDF Forecast has degraded substantially.

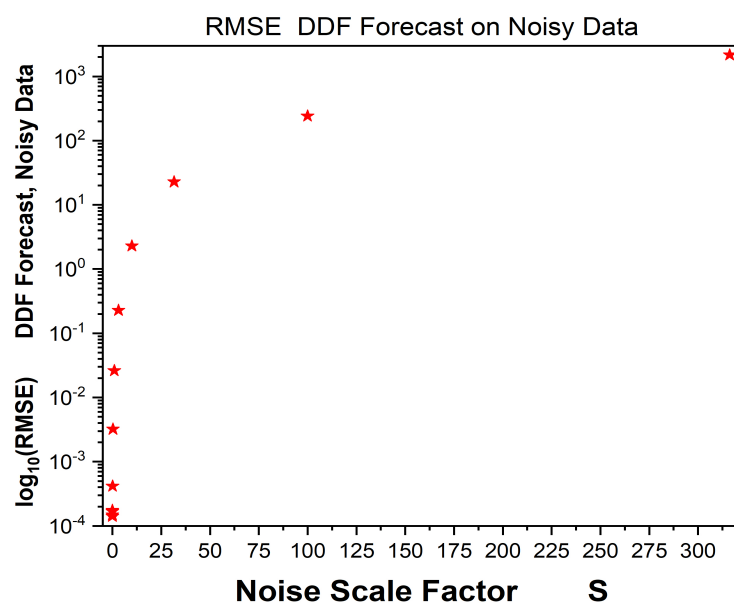


Figure 22: SWE on a 10×10 grid. The sensor region is comprised of 9 locations, blue dots in Figure (2), in a 3x3 off corner location. We display the RMS error in the x-velocity as a function of S over the range we considered.



436 6 Summary and Discussion

437 6.1 General Remarks

438 We have introduced a method for building a nonlinear discrete time fore-
 439 casting system for observed state variables in geophysical dynamical settings
 440 where the underlying model of the dynamics is not known. The method re-
 441 lies on observed data to train model parameters in a representation of the
 442 unknown dynamical rules. Time series of the data are considered as samples
 443 of the distribution in state space visited by the trajectories of the selected
 444 physical variables, and the representation of the vector field of the dynamical
 445 flow uses well tested interpolating basis functions to give information among
 446 the observed samples.

447 The method for representing the unknown dynamics of the physical flow
 448 adopted in our work uses weighted linear combinations of radial basis func-
 449 tions (RBFs) Hardy (1971); Micchelli (1986); Broomhead and Lowe (1988);
 450 Schaback (1995); M. D. J. (2002); Buhmann (2009), and the estimation of the
 451 weights is a linear algebra problem. This linear algebra operations require
 452 a Tikhonov regularization, also known as ridge regression Gruber (1998);
 453 Press et al. (2007); van Wieringen (2021). These methods and algorithms
 454 implementing them are well tested and widely discussed in the literature.

455 Selection of observations in a region means that the full state space of the
 456 dynamics is not sampled. This entails a projection from the full state space
 457 to the subspace of selected observables. Construction of an equivalent state
 458 space to the unknown physical state space is, in effect, an **unprojection** from
 459 the observed subspace. This is accomplished using time delay embedding
 460 widely analyzed in the nonlinear dynamics literature Takens (1981); Eckmann
 461 and Ruelle (1985); Sauer et al. (1991); Abarbanel (1996); Kantz and Schreiber
 462 (2004).

463 The Physics of using time delays comes from the fact that the temporal
 464 evolution of the observed state variables depends on the full set of (unknown)
 465 state variables; see Equation (3) and Equation (4). Over a time delay the
 466 full set of state variables is in operation, and the information about the
 467 unobserved state variables is conveyed to the observed variables through this
 468 feature of the dynamics.

469 The use of time delay embedding introduces at least two additional quan-
 470 tities to be estimated by data: the value of the time delay $\tau = hT_h$ and the
 471 number of required time delays D_E . If there are many time scales in the



472 physical system of interest, many time delays maybe useful along with differ-
 473 ent numbers of delayed observables D_E Judd and Mees (1995); Hirata et al.
 474 (2006).

475 These considerations led us to formulate the DDF task for forecasting
 476 the future of D_R dimensional observed state variables $\mathbf{O}(\mathbf{R}, t)$ located in a
 477 spatial region \mathbf{R} as

$$\mathbf{O}(\mathbf{R}, n+1) = \mathbf{O}(\mathbf{R}, n) + \mathbf{f}_{\mathbf{R}}(\mathbf{TD}(n), \boldsymbol{\chi}) + \frac{h}{2} \left[[\mathcal{F}(\mathbf{R}, n), 0] + [\mathcal{F}(\mathbf{R}, n+1), 0] \right]. \quad (19)$$

478 The $\mathbf{O}(\mathbf{R}, t)$ are D_R dimensional $\mathbf{O}(\mathbf{R}, t) = \{O_\alpha(t)\}; \alpha = 1, 2, \dots, D_R$.
 479 The dynamical map for the regional observables, written in components, is
 480 seen to be

$$O_\alpha(n+1) = O_\alpha(n) + f_\alpha(\mathbf{TD}(n), \boldsymbol{\chi}) + \frac{h}{2} \left[[\mathcal{F}(\mathbf{R}, n), 0] + [\mathcal{F}(\mathbf{R}, n+1), 0] \right]_\alpha. \quad (20)$$

481 in which $\mathbf{TD}(t_n) = \mathbf{TD}(n)$ are time delay vectors of dimension $D_E D_R$

$$\mathbf{TD}(t) = [\mathbf{O}(\mathbf{R}, t), \mathbf{O}(\mathbf{R}, t - \tau), \mathbf{O}(\mathbf{R}, t - 2\tau), \dots, \mathbf{O}(\mathbf{R}, t - (D_E - 1)\tau)], \quad (21)$$

482 and $\boldsymbol{\chi} = \{w_{\alpha q}, c_{\alpha j}, \sigma, B, D_E, T_h\}; \tau = hT_h$.

483 As noted this implies that a representation of the vector field for the
 484 flow of each observed quantity is required. We recognize that depending
 485 on the number of state variables one chooses to observe and the number of
 486 geographical locations at which one observes them, the computational burden
 487 required for DDF forecasting may increase substantially. It is unlikely to
 488 match the requirements for a full scale regional or global model as presently
 489 implemented.

490 The regional forcing of the system is given as $\mathcal{F}(\mathbf{R}, t_n) = \mathcal{F}(\mathbf{R}, n)$. The
 491 external forcing must be given in a DDF formulation just as it is required
 492 when we present a detailed physical model of the dynamics Roeckner et al.
 493 (2003); Staff (2021).

494 It is important that the forcing driving the global dynamics enters in an
 495 additive fashion in many physical systems, fluid dynamics among them, so
 496 the intrinsic properties of the global or regional system is separated from the



497 forces that drive the system into motion. Once the intrinsic aspects of the
 498 dynamics is encoded in the representation of the required flow vector fields,
 499 it should respond to any presentation of forcing as does a detailed model of
 500 the dynamical system.

501 6.2 The Specific Results in this Paper

502 In the example of DDF regional forecasting we investigated in this paper we
 503 took as our global dynamics the PDEs of shallow water flow realized on a
 504 rectangular mid-latitude plane. Clearly this is a useful but major reduction
 505 in complexity compared to the collection and set of data from real field
 506 observations. We have found it instructive to begin in this manner, and the
 507 results illustrate the issues to be encountered in a practical application of the
 508 formulation.

509 In this framework we demonstrated in a number of scenarios that ob-
 510 servations of a subset of ‘global’ shallow water flows can be use to build a
 511 discrete time flow $\mathbf{O}(\mathbf{R}, t) \rightarrow \mathbf{O}(\mathbf{R}, t + h)$ allowing for accurate forecasting
 512 beyond a temporal domain where data has been previously collected.

513 We also noted that in the choice of time delay embedding parameters
 514 $\{D_E, \tau = T_h h\}$ there is a range of D_E over which excellent forecasting can
 515 be achieved.

516 By adding Gaussian noise to the SWE data, we found a rather robust
 517 ability to make accurate DDF Forecasts in a region of the ‘global’ model
 518 until about $S = 1$, or 0 dB signal to noise.

519 6.3 Looking Forward

520 In investigating the examples we presented we were required to generate our
 521 own data, performing what is often called a *twin experiment* Abarbanel (2022)
 522 at a choice of grid points used to approximate solutions to the SWE PDFs.
 523 As one proceeds to using our results to guide DDF weather forecasting, at no
 524 stage do we approximately solve some physical PDEs on a grid of some spatial
 525 resolution, we have no restrictions on where the regional measurements may
 526 be performed.

527 Equally important to note is that the information about the actual dy-
 528 namics of the overall system of interest are not encoded into physical prop-
 529 erties of a model such as effective viscosities at a selected spatial resolution
 530 or parametrizations of sub-grid scale dynamics, etc.



531 The observed information is now located in the estimated parameters χ of
532 the vector field representations and in the time delay reconstruction of a space
533 equivalent to the full state space of the system. That is, the information in
534 the data remains as it was, but it is redistributed to a different representation
535 of the global dynamics than is encountered in the physical equations of the
536 global problem.

537 In working with observed data, there are no grid points. These are only
538 introduced to aid in the solution of the PDEs of a physical model. The
539 locations where one sets out sensors are totally up to the users. The discrete
540 time predictive models of observed quantities summarized in Equation (15)
541 are built with out concern for the spatial resolution of grid points in a physical
542 model, without specifying the dynamics of subgrid scale quantities, such as
543 cloud dynamics, and setting aside similar concerns in the formulation of
544 detailed models of the atmosphere and ocean.

545 As we pointed out earlier, the gains in using DDF models, must be rec-
546 ognized as having been achieved by relinquishing many of the details about
547 earth systems processes that are captured by detailed physically based PDEs
548 for earth systems processes. In DDF only the observed state variables can
549 be forecast. When that is what actually wishes to know, it provides another
550 way to sample data on observed quantities and forecast them.

551 Acknowledgements

552 We acknowledge support the Office of Naval Research, Grants N00014-20-1-
553 2580 and N00014-19-1-2522. Discussions with Erik Bollt, Daniel Gauthier,
554 and Steve Penny were central to pursuing this work.



555 7 Appendix

556 7.1 Performing DDF on Observed Data

557 The purpose of this section is to provide the reader with more detail on how
 558 the implementation of DDF works in practice and to identify what steps
 559 would need to be taken by the reader in order to implement DDF to fit their
 560 own data, observed or numerically generated for a twin experiment. Clearly
 561 the first step is to collect the data, and we assume that is where we start.

562 In order to implement DDF some choices must be made, the first choice
 563 will be in selecting a function representation of the flow vector field. We have
 564 tested Taylor Series Representations, RBF's, and RBF's plus polynomial.
 565 While the best choice will be dependent upon the system being studied,
 566 we've found general success with various representations.

567 In the coming sections we'll discuss different options and offer some guid-
 568 ance on choosing a representation for a data set. After choosing a function
 569 representation, we must choose a way to select centers for the RBF's, choose
 570 an appropriate time delay embedding dimension and time delay, and finally
 571 perform a hyper parameter search.

572 7.2 Designing a Representation of the Discrete Time 573 Flow Vector Field

574 To start every problem in DDF we must first choose a function representa-
 575 tion, $\mathbf{f}(\mathbf{S}(\mathbf{r}, t_n), \chi)$, for the data set we are studying. The way we choose to
 576 represent vector fields for a discrete time map will have the largest impact on
 577 the predictive power of DDF, we've found that this choice carries far more
 578 weight than a good choice of hyper-parameters or an apt selection of centers.

579 This paper has only focused on the use of RBF's, specifically the Gaussian
 580 RBF plus the first order polynomial of each observed dimension. For the
 581 shallow water flow data we found that using using polynomial terms in our
 582 $\mathbf{f}(\mathbf{S}(\mathbf{r}, t_n), \chi)$ representation to match the polynomial terms in the actual
 583 SWEs and letting the Gaussian RBF's capture the rest of the behavior of
 584 the SWEs works quite well.

585 We will go on to discuss more details of RBF's, we want to note that
 586 other choices exist; these choices include but are not limited to Multilayer
 587 Perceptrons, Hermite Interpolation, or any other numerical tool for interpo-
 588 lation. One item to keep in mind is that the RBF expansion is linear in the



589 RBF weights, allowing us to use linear algebra to estimate them.

590 7.2.1 Choosing a Radial Basis Function

591 The Radial Basis Function (RBF) was invented by Hardy (1971) for
 592 interpolation among observed samples of a function of multivariate variables.
 593 There is now an extensive literature on RBF's and their effectiveness as
 594 interpolation functions. In this literature, many different choices for the
 595 form of the RBF have been found to work Hardy (1971); Micchelli (1986);
 596 Broomhead and Lowe (1988); Schaback (1995); M. D. J. (2002); Buhmann
 597 (2009).

598 The parameters in our choice of RBF $\mathbf{f}(\mathbf{TD}(r, n), \boldsymbol{\chi})$, appearing in Equa-
 599 tion (14), are estimated as follows

600 The $\mathbf{O}(\mathbf{R}, t)$ are D_R dimensional $\mathbf{O}(\mathbf{R}, t) = \{O_\alpha(t)\}; \alpha = 1, 2, \dots, D_R$.
 601 The dynamical map for the regional observables is

$$\begin{aligned} O_\alpha(n+1) &= O_\alpha(n) + \\ f_\alpha(\mathbf{TD}(n), \boldsymbol{\chi}) &+ \frac{h}{2} \left[[\mathcal{F}(\mathbf{R}, n), 0] + [\mathcal{F}(\mathbf{R}, n+1), 0] \right]_\alpha \\ \mathbf{TD}(t) &= [\mathbf{O}(\mathbf{R}, t), \mathbf{O}(\mathbf{R}, t-\tau), \mathbf{O}(\mathbf{R}, t-2\tau), \dots, \mathbf{O}(\mathbf{R}, t-(D_E-1)\tau)]. \end{aligned} \quad (22)$$

602 To estimate the parameters $\boldsymbol{\chi}$ we minimize the objective function with
 603 respect to $\boldsymbol{\chi}$.

$$\begin{aligned} C(\boldsymbol{\chi}) &= \sum_{N_c+1}^{N_T} \left\{ [\mathbf{O}(n+1) - \mathbf{O}(n) - \mathbf{f}(\mathbf{TD}(n), \boldsymbol{\chi}) \right. \\ &\quad \left. - \frac{h}{2} \left[[\mathcal{F}(\mathbf{R}, n), 0] + [\mathcal{F}(\mathbf{R}, n+1), 0] \right] \right\}^2. \end{aligned} \quad (23)$$

604 In Equation (23)

$$f_\alpha(\mathbf{TD}(n), \boldsymbol{\chi}) = \sum_{q=1}^{N_c} w_{\alpha q} \psi((\mathbf{TD}(n) - \mathbf{TD}^c(q))^2, \sigma) + \sum_{j=1}^{D_R} c_{\alpha j} \mathbf{O}_j(n) \quad (24)$$

605 Even though we can choose any RBF for ψ , the $C(\boldsymbol{\chi})$ is always linear in
 606 the weights $w_{\alpha q}, c_{\alpha j}$, enabling the use of the linear algebra of Ridge Regression
 607 or Tikhonov regularization in estimating them.



608 In working with our example of the SWE, we found that the Gaussian
 609 RBF $\psi_G((\mathbf{TD}(n) - \mathbf{TD}^c(q))^2$ plus (1st order) polynomials in $\mathbf{TD}(n)$ to be
 610 more effective than $\psi_G((\mathbf{TD}(n) - \mathbf{TD}^c(q))^2$ alone.

611 To have a simplified notation for solving the regularized linear algebra
 612 problem, let us call

$$\mathbf{Y}(n) = \mathbf{O}(n+1) - \mathbf{O}(n) - \frac{h}{2} \left[[\mathcal{F}(\mathbf{R}, n), 0] + [\mathcal{F}(\mathbf{R}, n+1), 0] \right], \quad (25)$$

613 the vector of RBF coefficients,

$$\mathbf{P} = \{w_{\alpha q}, c_{\alpha j}\}, \quad (26)$$

614 and the RBF + Polynomial terms are the matrix \mathbf{M} .

615 Then the regularized cost function is given by Press et al. (2007)

$$C_B(\boldsymbol{\chi}) = \sum_n \left[\mathbf{Y}(n) - \mathbf{P}\mathbf{M}(n) \right]^2 + B\mathbf{P}^T\mathbf{P}. \quad (27)$$

616 T indicates the transpose.

617 The minimization of $C_B(\boldsymbol{\chi})$ with respect to \mathbf{P} , gives us the regularized
 618 solution

$$\mathbf{P} = \mathbf{Y} \cdot \mathbf{M}^T \frac{1}{(\mathbf{M}\mathbf{M}^T + B\mathbf{I})}. \quad (28)$$

619 After we introduce time delay embedding, the DDF parameters are $\boldsymbol{\chi} =$
 620 $\{w_{\alpha q}, c_{\alpha j}, \sigma, B, T_h, D_E\}$.

621 7.2.2 Including Polynomial Terms in Eq. (24)

622 In this paper we have found the most effective representation for SWE to
 623 be RBF representations with the inclusion of a polynomial term; the poly-
 624 nomial terms are in the $\mathbf{TD}(n)$. We retain only first order terms in this
 625 paper Schaback (1995); M. D. J. (2002).

626 With a general set of observations where we might have no particular in-
 627 sight into the dynamics, one should add the polynomials present in a general
 628 formulations of the problem Schaback (1995); M. D. J. (2002).



629 7.2.3 How to choose Centers

630 The choice of centers, $\mathbf{TD}^c(q)$ vector in the RBF, will be dictated by the
 631 data points in the training set.

632 We had initial success taking every n^{th} data point in the training window
 633 to be a center, this method is quick, but it may poorly saturate regions of
 634 low density and is prone to overlapping centers and to leaving empty space.

635 We recommend taking the approach of trying to saturate all important
 636 areas of space densely with centers to provide DDF with the most accurate
 637 update rule in those regions. With this in mind, we have opted to use K-
 638 means clustering to choose our centers for us. While there may be more
 639 advanced strategies, we have found K-means to work well enough, and we
 640 used it in finding all of the results in the present paper Du and Swamy (2006).

641 Another aspect to consider is the number of centers. The general rule
 642 of thumb is that more centers is better, but this comes at the cost of both
 643 memory and computational time. For the results in this paper, typically
 644 around 1000 centers were used.

645 One strategy could be to do preliminary testing with fewer centers and
 646 increase the number of centers. The fewer centers one can get by with, the
 647 better, as they will dramatically improve the computational time using the
 648 DDF map over the use of excessive centers.

649 7.2.4 Time Delay Embedding (TDE)

650 When studying physical systems, it is often very difficult or completely im-
 651 possible to measure all observable quantities of the system. There is in-
 652 formation that is lost in these unobserved quantities that must be recap-
 653 tured through the use of time delay embedding. When studying the sub-
 654 regions of the shallow water flows as global data or some region of the at-
 655 mosphere/oceans, we take full advantage of TDE in DDF to recapture the
 656 dynamical information of the system.

657 For the purpose of this paper, we took subregions of a grid generated by
 658 the SWE, and we required a choice of a time delay and dimension size for
 659 TDE.

660 To find an effective number of time delay dimensions, D_E , Taken's Theo-
 661 rem guarantees that the attractor can be reconstructed in as little as $2D_A + 1$
 662 where D_A is the box counting fractal dimension dimension of the original
 663 state space. However, by calculating the number of false nearest neigh-



bors Abarbanel (1996) at each D_E value, we can get an estimate of the ideal number of time delay dimensions. As this reconstruction proceeds, we eventually reach a point where the number of false nearest neighbors drops to a negligibly low number. This number of time delay dimensions serves as our estimate for D_E .

The time delay τ is conveniently taken to be a multiple of the observed temporal step h in the data collection. $\tau = T_h h$. Also the embedding dimension D_E for $\mathbf{TD}(t)$ is always an integer. So in minimizing the cost function $C(\chi)$ we must search a grid of integers $\{T_h, D_E\}$ along with a search in continuous variables $\{w_{\alpha q}, B, \sigma^2 = \frac{1}{2R}\}$.

We do not need to separately forecast all components of $\mathbf{TD}(t)$. If we forecast only $\mathbf{O}(t)$, the remaining components are forecast as well. We do require the full time delay vectors in the arguments of the RBF.

7.2.5 Finding the hyper-parameters R and B in χ

When performing DDF we are faced with the choice of picking a good value for hyper-parameters that affect our training and forecasting. For the case of the Gaussian RBF we have to pick an R value for the coefficient in the exponent of the RBF and a B value for the regularization term in Ridge Regression. To find a good value of B and R , it is typically easiest to perform the brute force method of grid searching across orders of magnitude of R and B ; once a good result is found, further sweeps can be conducted with greater precision if greater accuracy is desired. Alternatively, one method the reader could use would be to use a genetic algorithm to find a precise result after performing a wide grid sweep (this is discussed in the Future Improvements to DDF section under Differential Evolution).

It is the experience of the authors that if a large grid sweep across many orders of magnitude of R and B fail to provide any good results that the next best course of action would be to take another look at the function representation and try something new. We typically start with a basic order of magnitude sweep for both B and R starting both values at $1e-8$ and going up to $1e+1$ (the best value for B and R are very data set dependent, so initial large sweeps are always necessary). If the function representation is not working well, or is a bad fit for the data, then our experience has taught us that it is a fruitless endeavour to keep grinding away at testing more and more hyper-parameter values.



699 7.3 Programming DDF

700 We did all of our testing and programming in Python. Here we will offer
701 a few, hopefully, helpful suggestions for the reader in performing their own
702 DDF calculations as well as provide sample code.

703 7.3.1 PreBuilt DDF Code in Python

704 Here is a link to the code we used (written and tested in Python) that
705 have been published in GitHub. It includes the python scripts and Jupyter
706 Notebooks used to get the results in this paper, it also includes a few examples
707 as well as some information as to how we solve the SWE to generate the data
708 we use Clark

709 7.3.2 Memory Management

710 An important note to consider is the size of the matrices involved in training,
711 for they can grow to the size of gigabytes. For example, the largest matrix
712 involved in training is the \mathbf{M} matrix consisting of all values of the RBF's at
713 all times, an N by N_c , can have a length of up to 25,000 data points with as
714 many as 5,000 centers. Typically we use float64 for all our values resulting
715 this matrix using $25,000 \times 5,000 \times 8 / 1e9 = 1$ Gigabyte, then we'll need another
716 gigabyte for its transpose. This could be a limiting factor if one is running
717 multiple tests in parallel on a CPU or on a cluster with limited storage.

718 7.3.3 Parallelizing DDF

719 This section serves less as a guide and more as a suggestion and an obvious
720 disclaimer to take advantage of the ability to run DDF code in parallel. The
721 grid searching method discussed in the "Finding Hyper Parameters" section
722 of the appendix lends itself very readily to parallel programming. Even the
723 Differential Evolution method that is discussed in the Future Improvements
724 to DDF section can have it's trials ran in parallel. Note that a single trial of
725 DDF doesn't have much potential for parallel operations as the training is
726 just matrix multiplication and the forecasting is a step by step process that
727 relies of the result of the previous step's calculation.



7.4 Future Improvements to DDF

DDF isn't perfect and there is still much left to be explored with this method. There are still many RBF's that we never tested that could potentially work better than the Gaussian we've had success with. There are also smarter ways of choosing hyper-parameters other than brute force grid searching, such as using genetic algorithms like Differential Evolution.

7.4.1 Centers and RBF's

We took for granted that the obvious choices for DDF worked out as well as they did, choosing Gaussian RBF's and K-means clustering to put together our DDF experiments were ideas commonly used in the RBF literature Wu et al. (2012); however, they are far from the only good ideas there and others could possibly work better.

The centers could also be chosen by emphasizing places where the first or second derivative are greatest; this idea could work well because it could saturate regions where not many data points exist. Sánchez A (1995)

7.4.2 Differential Evolution

Differential Evolution is a genetic algorithm that could be implemented into DDF to improve our ability to create DDF models of observed systems by more precisely picking out sets of hyper parameters. It works by initiating a parent set of hyper-parameters from a user defined uniform distribution. From there, new "children" sets of hyper parameters are made from combining the parents in a algorithmic way and comparing the new forecast that is made to the old one. A user defined cost function is used to compare the the children to their parents, as generations go by, parents will gradually be replaced by children until all the parents converge on a minimum in the cost function. If the user is able to define a clever cost function, they could get a result that hope to surpass those from simple grid searching. The full method is described in Storn and Price (1997).



References

- Abarbanel, H. D. I. (1996). *The Analysis of Observed Chaotic Data*. Springer-Verlag, New York.
- Abarbanel, H. D. I. (2022). *The Statistical Physics of Data Assimilation and Machine Learning*. Cambridge University Press.
- Brajard, J., Carrassi, A., Bocquet, M., and Bertino, L. (2021). Combining data assimilation and machine learning to infer unresolved scale parametrization. *Phil. Trans. R. Soc. A*, 379:20200086.
- Broomhead, D. S. and Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355.
- Buhmann, M. D. (2009). *Radial Basis Functions (Theory and Implementations)*. Cambridge University Press.
- Chantry, M., Christensen, H., Dueben, P., and Palmer, T. (2021). Machine learning for weather and climate modelling. *Philosophical Transactions of the Royal Society A*, 379.
- Clark, R. https://github.com/RandarserousRex/Driven_SWE_DDF.
- Comrie, A. C. (1997). Comparing neural networks and regression models for ozone forecasting. *Journal of the Air & Waste Management Association*, 47(6):653–663.
- Daw, A., Karpadne, A., Watkins, W. D., Read, J. S., and Kumar, V. (2017). Physics-guided neural networks (pgnn): An application in lake temperature modeling. In *Knowledge-Guided Machine Learning*, pages 353–372. Chapman and Hall/CRC.
- Du, K.-L. and Swamy, M. N. S. (2006). *Neural networks in a softcomputing framework*. Springer Science & Business Media.
- Dueben, P. D. and Bauer, P. (2018). Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, 11(10):3999–4009.
- Eckmann, J.-P. and Ruelle, D. (1985). Ergodic theory of chaos and strange attractors. *Reviews of Modern Physics*, 57:617–656.



- 787 Geer, A. J. (2021). Learning earth system models from observations: machine
 788 learning or data assimilation? *Phil. Trans. R. Soc. A*, 379:20200089.
- 789 Grover, A., Kapoor, A., and Horvitz, E. (2015). A deep hybrid model for
 790 weather forecasting. In *Proceedings of the 21th ACM SIGKDD interna-*
 791 *tional conference on knowledge discovery and data mining*, pages 379–386.
- 792 Gruber, M. (1998). *Improving Efficiency by Shrinkage: The James–Stein*
 793 *and Ridge Regression Estimators*. Boca Raton: CRC Press.
- 794 Hardy, R. L. (1971). Multiquadric equations of topography and other irreg-
 795 ular surfaces. *Journal of Geophysical Research*, 76:1905–1915.
- 796 Hirata, Y., Suzuki, H., and Aihara, K. (2006). Reconstructing state spaces
 797 from multivariate data using variable delays. *Physical Review E*, 74:026202.
- 798 Huang, X., Kroening, D., Ruan, W., Sharp, J., Sun, Y., Thamo, E., Wu,
 799 M., and Yi, X. (2020). A survey of safety and trustworthiness of deep
 800 neural networks: Verification, testing, adversarial attack and defence, and
 801 interpretability. *Computer Science Review*, 37:100270.
- 802 Jiang, S., Jin, F.-F., and Ghil, M. (1994). Multiple equilibria, periodic, and
 803 aperiodic solutions in a wind-driven, double-gyre, shallow-water model.
 804 *Journal of Physical Oceanography*, 25:764–786.
- 805 Judd, K. and Mees, A. (1995). On selecting models for nonlinear time series.
 806 *Physica D*, 82:426–444.
- 807 Kantz, H. and Schreiber, T. (2004). *Nonlinear Time Series Analysis, 2nd*
 808 *ed.* Cambridge University Press, Cambridge, UK.
- 809 Kashinath, K., Mustafa, M., Albert, A., Wu, J., Jiang, C., Esmaeilzadeh, S.,
 810 Azizzadenesheli, K., Wang, R., Chattopadhyay, A., Singh, A., Manepalli,
 811 A., Chirila, D., Yu, R., Walters, R., White, B., Xiao, H., Tchelepi,
 812 H. A., Marcus, P., Anandkumar, A., Hssanzadeh, P., and Prabhat
 813 (2021). Physics-informed machine learning: case studies for weather and
 814 climate modelling. *Philosophical Transactions of the Royal Society A*,
 815 379(2194):20200093.
- 816 Krasnopolsky, V. M., Chalikov, D. V., and Tolman, H. L. (2002). A neu-
 817 ral network technique to improve computational efficiency of numerical
 818 oceanic models. *Ocean Modelling*, 4(3-4):363–383.



- 819 M. D. J., P. (2002). Radial basis function methods for interpolation to func-
 820 tions of many variables. *Fifth Hellenic-European Conference on Computer*
 821 *Mathematics and its Applications; Athens, September 2001*.
- 822 Mackowiak, R., Ardizzone, L., Kothe, U., and Rother, C. (2021). Generative
 823 classifiers as a basis for trustworthy image classification. In *Proceedings of*
 824 *the IEEE/CVF Conference on Computer Vision and Pattern Recognition*,
 825 pages 2971–2981.
- 826 Micchelli, C. A. (1986). Interpolation of scattered data: Distance matrices
 827 and conditionally positive definite functions. *Constructive Approximation*,
 828 2:11–22.
- 829 Olver, P. J. (2017). Nonlinear ordinary differential equations. Technical re-
 830 port, University of Minnesota. <http://www-users.math.umn.edu/~olver/>.
- 831 Olver, P. J. (2020). *Introduction to Partial Differential Equations*. Springer
 832 Science+Business Media. DOI 10.1007/978-3-319-02099-0.
- 833 Pedlosky, J. (1986). *Geophysical Fluid Dynamics, 2nd Edition*. Springer
 834 Verlag. ISBN 0-387-96387-1.
- 835 Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007).
 836 *Numerical recipes: the art of scientific computing, 3rd Edition*. Cambridge
 837 University Press.
- 838 Roeckner, E., Bauml, G., Bonaventura, L., Brokopf, R., Esch, M., Giorgetta,
 839 M., Hagemann, S., Kirchner, I., Komblueh, L., Manzini, E., Rhodin, A.,
 840 Schlese, U., Schulzweida, U., and Tompkins, A. (2003). The atmospheric
 841 general circulation model echam5, part1, model description. Technical
 842 Report 349, The Max Planck Institute for Meteorology, Hamburg.
- 843 Sadourny, R. (1975). The dynamics of finite-difference models of the shallow
 844 water equations. *Journal of the Atmospheric Sciences*, 32:680–689.
- 845 Sánchez A, V. D. (1995). Second derivative dependent placement of rbf
 846 centers. *Neurocomputing*, 7(3):311–317.
- 847 Sauer, T. J., Yorke, J. A., and Casdagli, M. (1991). Embedology. *Journal of*
 848 *Statistical Physics*, 65:579–616.



- 849 Schaback, R. (1995). Error estimates and condition numbers for radial basis
 850 function interpolation. *Advances in Computational Mathematics*, 3:251–
 851 264.
- 852 Schultz, M. G., Betancourt, C., Gong, B., Kleinert, F., Langguth, M., Leufen,
 853 L. H., Mozaffari, A., and Stadtler, S. (2021). Can deep learning beat
 854 numerical weather prediction? *Philosophical Transactions of the Royal*
 855 *Society A*, 379(2194):20200097.
- 856 Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c.
 857 (2015). Convolutional lstm network: A machine learning approach for pre-
 858 cipitation nowcasting. *Advances in neural information processing systems*,
 859 28.
- 860 Staff, E. (2021). Ifs documentation-cy47r3; part iii: Dynamics and nu-
 861 merical procedures. Technical report, European Center for Medium
 862 Range Weather Forecasting. [https://www.ecmwf.int/en/publications/ifs-](https://www.ecmwf.int/en/publications/ifs-documentation)
 863 [documentation](https://www.ecmwf.int/en/publications/ifs-documentation).
- 864 Stevens, B., Satoh, M., Auger, L., Biercamp, J., Bretherton, C. S., Chen,
 865 X., Duben, P., Judt, F., Khairoutdinov, M., Klocke, D., Kodama, C.,
 866 Kornblueh, L., Lin, S.-J., Philipp Neumann, P., Putman, W. M., Rober,
 867 N., Shibuya, R., Vanniere, B., Vidale, P. L., Wedi, N., and Zhou, L. (2019).
 868 Dyamond: the dynamics of the atmospheric general circulation modeled
 869 on non-hydrostatic domains. *Progress in Earth and Planetary Science*,
 870 6:61–78.
- 871 Storn, R. and Price, K. (1997). Differential evolution—a simple and efficient
 872 heuristic for global optimization over continuous spaces. *Journal of global*
 873 *optimization*, 11(4):341–359.
- 874 Takens, F. (1981). Detecting strange attractors in turbulence. *Lecture Notes*
 875 *in Math.*, 898:366–381.
- 876 Vallis, G. K. (2017). *Atmosphere and Ocean Fluid Dynamics: Funda-*
 877 *mentals and Large-Scale Circulation*. Cambridge University Press. doi:
 878 10.1017/9781107588417.
- 879 van Wieringen, W. N. (2021). Lecture notes on ridge regression,; version
 880 0.40. <https://arxiv.org/1509.09169>.



- 881 Wandel, N., Weinmann, M., and Klein, R. (2020). Learning incompressible
882 fluid dynamics from scratch—towards fast, differentiable fluid models that
883 generalize. *arXiv preprint arXiv:2006.08762*.
- 884 Watson-Parris, D. (2021). Machine learning for weather and climate are
885 worlds apart. *Phil. Trans. R. Soc. A*, 379:20200098.
- 886 Wedi, N. P., Polichtchouk, I., Dueben, P., Anantharaj, V. G., Bauer, P.,
887 Boussetta, S., Browne, P., Deconinck, W., Gaudin, W., Hadade, I., Hat-
888 field, S., Iffrig, O., Lopez, P., Maciel, P., Mueller, A., Saarinen, S., Sandu,
889 I., Tiago Quintino, T., and Vitart, F. (2020). A baseline for global weather
890 and climate simulations at 1 km resolution. *Journal of Advances in Mod-
891 eling Earth Systems*, 12:e2020MS002192.
- 892 Wu, Y., Wang, H., Zhang, B., and Du, K.-L. (2012). Using radial basis func-
893 tion networks for function approximation and classification. *International
894 Scholarly Research Notices*, 2012.