TEDA: A Computational Toolbox for Teaching Ensemble Based Data Assimilation

Elias D. Nino-Ruiz(✉) and Sebastian Racedo Valbuena

Applied Math and Computer Science Laboratory, Department of Computer Science, Universidad del Norte, Barranquilla 080001, Colombia
{enino,racedo}@uninorte.edu.co

Abstract. This paper presents an intuitive Python toolbox for Teaching Ensemble-based Data Assimilation (DA), TEDA. This toolbox responds to the necessity of having software for teaching and learning topics related to ensemble-based DA; this process can be critical to motivate undergraduate and graduate students towards scientific topics such as meteorological anomalies and climate change. Most DA toolboxes are related to operational software wherein the learning process of concepts and methods is not the main focus. TEDA facilitates the teaching and learning process of DA concepts via multiple plots of error statistics and by providing different perspectives of numerical results such as model errors, observational errors, error distributions, the time evolution of errors, ensemble-based DA, covariance matrix inflation, precision matrix estimation, and covariance localization methods, among others. By default, the toolbox is released with five well-known ensemble-based DA methods: the stochastic ensemble Kalman filter (EnKF), the dual EnKF formulation, the EnKF via Cholesky decomposition, the EnKF based on a modified Cholesky decomposition, and the EnKF based on B-localization. Besides, TEDA comes with three toy models: the Duffing equation (2 variables), the Lorenz 63 model (3 variables), and the Lorenz 96 model (40 variables), all of which exhibit chaotic behavior for some parameter configurations, which makes them attractive for testing DA methods. We develop the toolbox using the Object-Oriented Programming (OOP) paradigm, which makes incorporating new techniques and models into the toolbox easy. We can simulate several DA scenarios for different configurations of models and methods to better understand how ensemble-based DA methods work.

Keywords: Data Assimilation · Ensemble Kalman Filter · Education · Python

1 Introduction

Computational tools are widely applied in different contexts of science to develop curiosity and interest in undergraduate and graduate students towards some specific discipline. This seems to be an extraordinary strategy in which instructors

GitHub Package Repository https://github.com/enino84/TEDA.git.

D. Groen et al. (Eds.): ICCS 2022, LNCS 13353, pp. 732–745, 2022.
https://doi.org/10.1007/978-3-031-08760-8_60

and students feel comfortable boarding topics from different branches of science. Indeed, it has helped in the context of (applied) math (and in general STEM courses), wherein students offer some sort of resistance to learning [10]. Nowadays, Data Assimilation (DA) has become a relevant field of research, learning, and teaching owing to recent meteorological anomalies and climate change. In sequential DA, an imperfect numerical forecast $\mathbf{x}^b \in \mathbb{R}^{n\times 1}$ is adjusted according to a vector of real noisy observations $\mathbf{y} \in \mathbb{R}^{m\times 1}$, where $\mathbf{x}^b \in \mathbb{R}^{n\times 1}$ and $\mathbf{y} \in \mathbb{R}^{m\times 1}$ are the background state and the observation vector, respectively, n is the model size, and m denotes the number of observations. To estimate prior parameters, typically, ensembles of model realizations are used. Since each ensemble member has a high computational cost, ensemble sizes are commonly bounded by the hundreds while model resolutions are in the millions. This provides a set of difficulties that are of high interest to the data assimilation community, researchers in general, and instructors. For instance, if seen from a researcher's perspective, the impact of sampling errors is something of high relevance, especially since these methods can be employed, for instance, to predict hurricane initialization with a fairly high percentage of certainty [2]. Another application of DA that is highly active is the combination of Global Positioning System (GPS) and Meteorology (MET) with DA methods to improve weather predictions [12]. However, from an instructor or student perspective, who is trying to explain or learn these topics, then the availability of software, toolboxes, or other tools to aid in teaching is still very limited. There are several DA-oriented toolboxes but usually they are designed for investigative use, and most of them are not flexible when wanting to introduce new models or methods. In [9], authors implemented an open interface standard for a quickly implement of DA for numerical models. This implementation has a few well known methods and focuses on parameter estimation. In [1] authors present a tutorial in Python that shows the implementation of common sequential methods and the idea behind them. These implementations have good approaches to introduce researchers, professors or students to the area of DA, but have some minor inconveniences, for example, by the time this paper is written [9] is available in Java, C++, and FORTRAN which are well know programming languages but are not that simple to understand, also the toolbox needs to be downloaded. We propose a Python-based toolbox to facilitate the comprehension of DA topics that can be used offline or online in Google Colab. This toolbox follows the Object-Oriented Programming (OOP) paradigm, which brings flexibility; it counts with different ensemble-based methods, numerical models, and covariance and precision matrix estimators. Our implementation is education-oriented, and therefore, it is more detailed than other operational DA implementations.

The structure of this paper is as follows: Sect. 2 briefly discuss ensemble-based methods, well-known issues in the context of ensemble DA, and numerical toy models; the topics are restricted to all functionalities in our educational toolbox, in Sect. 3 presents the toolbox and how to employ it to simulate and test ensemble methods. Section 4 demonstrates some results of the use of the

toolbox with different methods and models. Conclusions and future directions of TEDA are stated in Sect. 5.

2 Preliminaries

Learning or teaching STEM (Science, Technology, Engineering and Mathematics) topics can be challenged; instructors often apply motivational strategies to keep students focused on their course content [5]. DA is not the exception: this subject combines information from different branches of science: Statistics, Applied Math, and Computer Science. DA can be seen as a challenging field to study, do research, and teach at a first look. However, we believe that by using a proper toolbox wherein numerical experiments are run, and analyses are provided; students can quickly assimilate concepts from different fields of science. We consider DA as a field of extreme importance given current meteorological and climate conditions. Climate change is a real issue that all people must be aware of. It impacts different branches of our lives, starting with the planet we live on. Waterfloods, forest fires, and tsunamis are just consequences of everything we have done to our world so far. Despite how we got to this point, the main question remains: what can we do to mitigate the impact of climate change in our lives, economies, societies, and planet? We can answer this question in many manners, one of them is by providing scientists, instructors, and students with a context such as DA.

Our toolbox provides filter implementations, numerical models, and multiple plots to enrich learning and teaching. We release our learning toolbox with ensemble-based filter implementations, numerical models, and DA scenarios to efficiently perform the teaching process; we briefly discuss all of them in this Section.

2.1 Ensemble-Based Data Assimilation

The Ensemble Kalman Filter (EnKF) is a sequential ensemble-based method which employs an ensemble of model realizations to estimate moments of prior error distributions:

$$\mathbf{X}^b = \left[\mathbf{x}^{b[1]}, \mathbf{x}^{b[2]}, \ldots, \mathbf{x}^{b[N]}\right] \in \mathbb{R}^{n \times N}, \tag{1a}$$

where $\mathbf{x}^{b[e]} \in \mathbb{R}^{n \times 1}$ denotes the e-th ensemble member, for $1 \leq e \leq N$, at time k, for $0 \leq k \leq M$. The empirical moments of (1a) are employed to estimate the forecast state $\mathbf{x}^b$:

$$b = \frac{1}{N} \sum_{e=1}^{N} \mathbf{x}^{b[e]} \in \mathbb{R}^{n \times 1},$$

and the background error covariance matrix $\mathbf{B}$:

$$\mathbf{P}^b = \frac{1}{N-1} \boldsymbol{\Delta}\mathbf{X}^b \left[\boldsymbol{\Delta}\mathbf{X}^b\right]^T \in \mathbb{R}^{n \times n},$$

where the matrix of member deviations reads:

$$\boldsymbol{\Delta}\mathbf{X}^b = \mathbf{X}^b - b\mathbf{1}^T \in \mathbb{R}^{n\times N}\,. \tag{1b}$$

The posterior ensemble can then be built by using synthetic observations:

$$\mathbf{X}^a = \mathbf{X}^b + \boldsymbol{\Delta}\mathbf{X}^a\,,$$

where the analysis updates can be obtained by solving the following linear system in the stochastic EnKF [4]:

$$\left[\mathbf{H}\mathbf{P}^b\mathbf{H}^T + \mathbf{R}\right]\boldsymbol{\Delta}\mathbf{X}^a = \mathbf{D}^s \in \mathbb{R}^{m\times N}\,,$$

or in the dual EnKF formulation [8]:

$$\left[\left[\mathbf{P}^b\right]^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right]\boldsymbol{\Delta}\mathbf{X}^a = \mathbf{H}^T\mathbf{R}^{-1}\mathbf{D}^s \in \mathbb{R}^{n\times N}\,,$$

the innovation matrix reads $\mathbf{D}^s \in \mathbb{R}^{m\times N}$ whose e-th column $\mathbf{y} - \mathbf{H}\mathbf{x}^{b[e]} + \boldsymbol{\varepsilon}^{[e]} \in \mathbb{R}^{m\times 1}$ is a synthetic observation with $\boldsymbol{\varepsilon}^{[e]} \sim \mathcal{N}\left(\mathbf{0}_m, \mathbf{R}\right)$. Note that, the synthetic observations are sampled about the actual ones $\mathbf{y}$. Yet another possible implementation is via the space spanned by the ensemble of anomalies (1b), this is

$$\mathbf{X}^a = \mathbf{X}^b + \underbrace{\boldsymbol{\Delta}\mathbf{X}^b\mathbf{W}^*}_{\boldsymbol{\Delta}\mathbf{X}^a} \tag{2a}$$

where the analysis innovations $\boldsymbol{\Delta}\mathbf{X}^a$ can be computed by solving the following linear system onto the ensemble space:

$$[(N-1)\mathbf{I} + \mathbf{Q}^T\mathbf{R}^{-1}\mathbf{Q}]\mathbf{W}^* = \mathbf{Q}^T\mathbf{R}^{-1}\left[\mathbf{Y}^S - \mathbf{H}\mathbf{X}^b\right]$$

where $\mathbf{Q} = \sqrt[-1]{N-1}\mathbf{H}\boldsymbol{\Delta}\mathbf{X}^b \in \mathbb{R}^{m\times N}$. The implementation (2a) is well-known as the EnKF based on Cholesky decomposition.

In operational DA scenarios, the ensemble size can be lesser than model dimensions by order of magnitudes, and as a direct consequence, sampling errors impact the quality of analysis increments. To counteract the effects of sampling noise, localization methods are commonly employed. In the context of covariance tapering, the use of the *spatial-predecessors* concept can be employed to obtain sparse estimators of precision matrices [6]. The predecessors of model component i, from now on $P(i,\, r)$, for $1 \leq i \leq n$ and a radius of influence $r \in \mathbb{Z}^+$, are given by the set of components whose labels are lesser than that of the i-th one.

In the EnKF based on a modified Cholesky decomposition (EnKF-MC) [8] the following estimator is employed to approximate the precision covariance matrix of the background error distribution:

$$\widehat{\mathbf{B}}^{-1} = \widehat{\mathbf{L}}^T\widehat{\mathbf{D}}^{-1}\widehat{\mathbf{L}} \in \mathbb{R}^{n\times n}\,,$$

where the Cholesky factor $\mathbf{L} \in \mathbb{R}^{n \times n}$ is a lower triangular matrix,

$$\left\{\widehat{\mathbf{L}}\right\}_{i,v} = \begin{cases} -\beta_{i,v,k} & , v \in P(i,r) \\ 1 & , i = v \\ 0 & , otherwise \end{cases},$$

whose non-zero sub-diagonal elements $\beta_{i,v,k}$ are obtained by fitting models of the form,

$$\mathbf{x}_{[i]}^T = \sum_{v \in P(i,r)} \beta_{i,v,k} \mathbf{x}_{[v]}^T + \boldsymbol{\gamma}_i \in \mathbb{R}^{N \times 1}, 1 \leq i \leq n,$$

where $\mathbf{x}_{[i]}^T \in \mathbb{R}^{N \times 1}$ denotes the i-th row (model component) of the ensemble (1a), components of vector $\boldsymbol{\gamma}_i \in \mathbb{R}^{N \times 1}$ are samples from a zero-mean Normal distribution with unknown variance σ^2, and $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose diagonal elements read,

$$\begin{aligned} \{\mathbf{D}\}_{i,i} &= \widehat{\mathbf{var}}\left(\mathbf{x}_{[i]}^T - \sum_{v \in P(i,r)} \beta_{i,v,k} \mathbf{x}_{[j]}^T\right)^{-1} \\ &\approx \mathbf{var}\left(\boldsymbol{\gamma}_i\right)^{-1} = \frac{1}{\sigma^2} > 0, \text{ with } \{\mathbf{D}\}_{1,1} = \widehat{\mathbf{var}}\left(\mathbf{x}_{[1]}^T\right)^{-1}, \end{aligned}$$

where $\mathbf{var}(\bullet)$ and $\widehat{\mathbf{var}}(\bullet)$ denote the actual and the empirical variances, respectively. The analysis equations can then be written as discussed in [8].

2.2 Numerical Models

The Lorenz 96 model is a dynamical system formulated by Edward Lorenz in [7]; it mimics the behaviour of particles fluctuacting in the atmosphere. The dynamics of particles $x_i \in \mathbb{R}$, for $1 \leq i \leq n$, are described by the following set of Ordinaty Differential Equations:

$$\frac{dx_i}{dt} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F,$$

wherein periodical boundary conditions are assumed $x_{-1} = x_{N-1}, x_0 = x_N, x_{N+1} = x_1$. This is one of the most widely employed toy models in the context of DA given its chaotic nature when the external force F equals 8.

The Lorenz 63 model is a dynamic system of three Ordinary Differential Equations introduced by Edward Lorenz, again. The equations are described as:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x), \frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y, \frac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z,$$

where $\sigma, \rho,$ and β are parameters corresponding to the Prandtl number (usually 10), Rayleigh number , and some physical dimension of the layer itself (usually 8/3), respectively. The system exhibits a chaotic behavior when $\rho = 28$. This

model can be seen as a simplified mathematical model for atmospheric convection.

The Duffing Equation is a non-linear second order differential equation presented by Georg Duffing in [3]. This equation is used to model damped and driven oscillators. This initial value problem reads:

$$\ddot{x} + \delta\dot{x} + \alpha x + \beta x^3 = \gamma\cos(\omega t), \text{ with } x(0) = 0, \text{ and } \dot{x}(0) = 1,$$

where δ controls the amount of damping, α denotes the linear stiffness, β stands for the amount of non-linearity in the restoring force, and γ is the amplitude of the periodic driving force.

3 Teaching Data Assimilation (TEDA) Python Toolbox

We provide a simple manner to learn and teach DA-related concepts: a computational toolbox named TEDA (Teaching Ensemble-based Data Assimilation). Our toolbox is written using Python, and we employ the Object-Oriented-Programming (OOP) paradigm to make easier the merging of our toolbox into any other educational or research program. The github repository of our package reads https://github.com/enino84/TEDA.git. We allow the users to analyze results from different perspectives. For instance, our visualizations allow users to understand error correlations, the structure of correlations, model trajectories, ensemble uncertainties, the effects of assimilation on model trajectories and correlations, the results of applying covariance tapering on precision matrices or localization methods on covariance ones, the error evolution of small perturbations, prior and posterior estimation of errors, among others. We employ some useful metrics to provide a wide spectrum of the forecast and the assimilation processes through specialized metrics, for instance, the ℓ_2-norm of errors or the Root-Mean-Square-Errors, both of them well-known in the DA community. We release TEDA with five well-known sequential data assimilation methods: the EnKF, the stochastic EnKF, the EnKF via Cholesky, the EnKF via a modified Cholesky decomposition, and the EnKF via B-Localization. Besides, three numerical models are available for testing and creating DA benchmarks: Lorenz 96, Lorenz 63, and Duffing's Equation. We briefly discuss all methods and models in our toolbox in Sect. 2. We release TEDA in two different manners: as a Python package and a Jupyter notebook.

As a Python package, TEDA consists of five major classes: the *Model* and the *Analysis* ones, both of them abstract classes; these ones define the methods that must be implemented for the definition of new models and ensemble methods. The *Background* class has the methods to create the initial ensemble, the initial forecast, and other required methods for simulations. The *Observation* class defines methods to generate synthetic observations jointly their error distributions (Gaussian by default). The *Simulation* class triggers simulations in DA scenarios given objects of the previous four classes; from here, we can run simulations by calling the "run" method, which will generate outputs such as error plots and statistics. We can modify all parameters for classes and methods

in TEDA as needed. We document classes and methods by using Docstring; a convention described within PEP 257 [11]. The purpose of these is to provide to final users brief overviews of objects, attributes, and methods. We show the class diagram of TEDA in Fig. 1. A representation of how folders are arranged can be seen in Fig. 2.

The TEDA notebook is an implementation of our toolbox on Jupyter notebooks. This allows students and instructors to simulate, test, and study sequential data assimilation methods easily. This notebook can be deployed in any web-based interactive development environment for notebooks, code, and data. For instance, we can run on-line the TEDA notebook by using Google Colab, a well-known free cloud service to host and execute Jupyter notebooks. This aspect is relevant since computer power is not needed from students to run our toolbox (which can be seen as a social inclusion initiative); a conventional laptop or low-memory computer with a web browser is sufficient for TEDA. This also can support breaking the social gap regarding students and computational resources; this is a well-known issue in developing countries. Since TEDA is based in OOP, instructors and students can develop their own methods and integrate them into our toolbox as required.

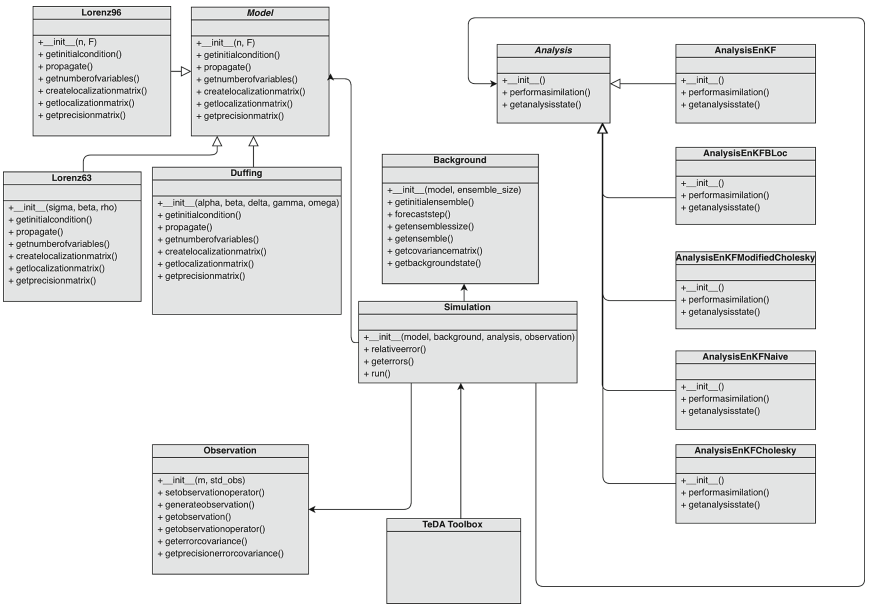


Fig. 1. Class diagram of the TEDA learning toolbox.

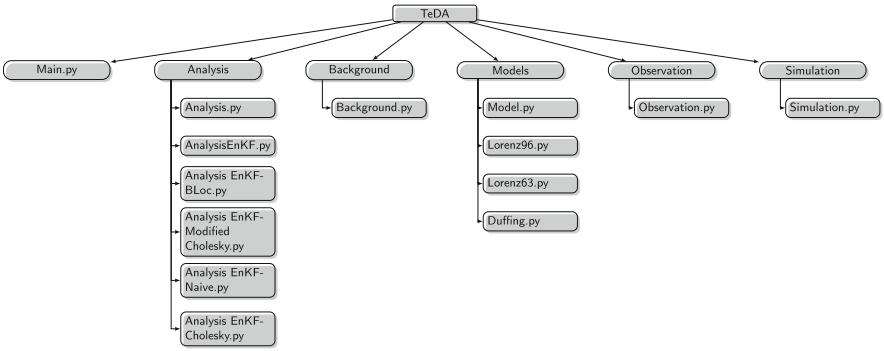


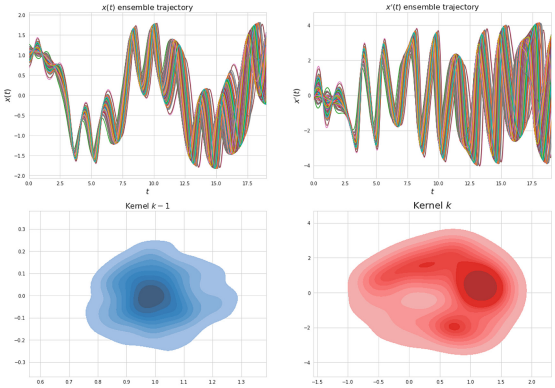
Fig. 2. Tree of the TEDA folder. The necessary files to run (or use) the assimilation toolbox.

4 Demonstration of the TEDA Toolbox

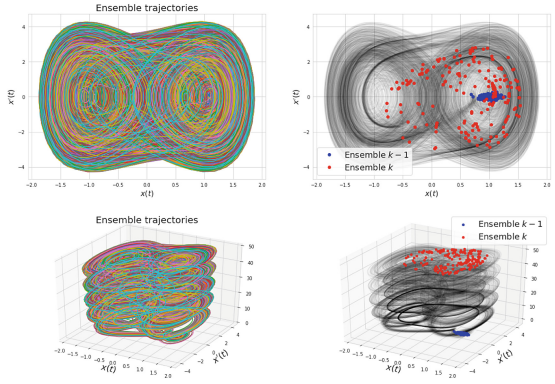
In this Section, we briefly show the capabilities of our package and all potential analyses that students and instructors can conduct via TEDA. Our toolbox attacks four important points during learning ensemble DA: initial ensemble generation, assimilation step, effects of sampling errors, and covariance matrix estimation. Our toolbox provides visual reports of all previous concepts jointly quantitative (statistical) analyses. All plots in this Section can be generated from the TEDA visualization toolkit.

In the context of initial ensemble generation, TEDA allows instructors and students to set up initial perturbation errors to create initial ensembles of model realizations. In Fig. 3a and 3b, we show a visual report of generating an initial ensemble for the Duffing equation via TEDA. In the first row, we show how the initial random perturbations are adjusted according to Duffing dynamics. For this model, by adding Normal white noise to the initial condition ($x(0) = 0$ and $x'(0) = 1$), we can see how the small initial perturbations are amplified by the non-linear dynamics encapsulated in the numerical model. Besides, the chaotic behavior of this model is clear; small perturbations ($t = 0$) tend to completely different conditions in time ($t = 18$). This can help students to understand how non-linear dynamics can drive forecasts to completely different states in future steps. The second row shows that initial Gaussian perturbations (in blue) are mapped to non-Gaussian kernels after numerical model integration (in red). For instance, the ensemble distribution is non-Gaussian (and even more, it is multi-modal). This makes clear the fact that non-Gaussian distributions are frequently found in real-life contexts. This is a well-known issue in the DA community; despite non-Gaussian errors being present in background distributions, scientists rely on Gaussian assumption on prior errors mainly to computational resources: for Gaussian assumption on prior and observational errors, closed-form expression can be obtained for computing analysis members. In the third row, the first column shows the different trajectories for all initial perturbations. Trajectories are identified by colors; again, it is evident the chaotic behavior of

the Duffing model; the second column allows us to identify zones from which initial perturbations start (blue dots) and where they end (red dots) after model integration. Despite the fact that initial solutions are close (in space), we can see how the non-linear nature of the Duffing equation makes them follow different paths. In the last row, the first column denotes a three-dimensional plot wherein we can see how ensemble members take different trajectories as time evolves. The Duffing equation is time-dependent, and therefore, we can see the multi-modal spiral behavior for each solution. Besides, the second column shows where the initial conditions start and where they end after the model propagation. The last two rows can be employed to show how non-linear dynamics affect model trajectories even for small synthetic perturbations. As can be seen, many explanations can be obtained by just analyzing plots or their combination. This provides a wide vision of how spatial paths relate to small perturbations in initial conditions and how Normal initial perturbations are non-linearly mapped to non-Gaussian shapes after model propagation.



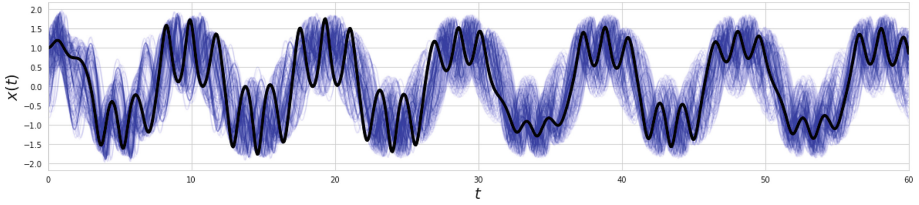
(a) Initial ensemble representation and statistics.



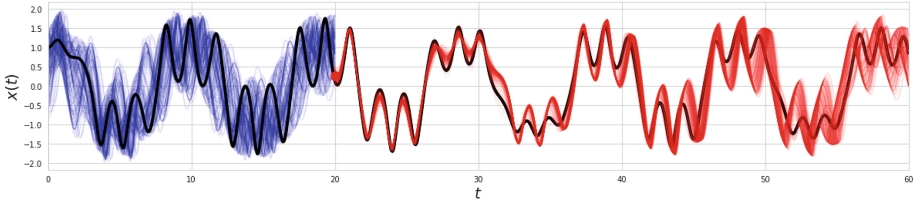
(b) Initial model representation in 2D and 3D.

Fig. 3. Some TEDA plots for data assimilation using the duffing equation

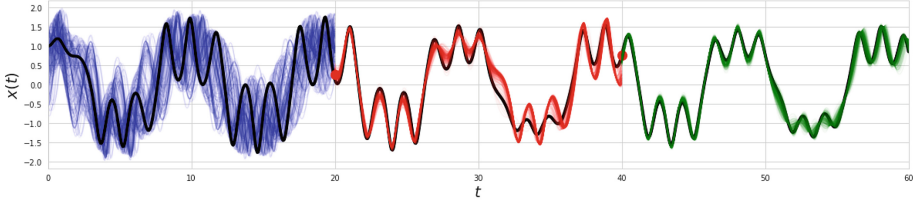
In Fig. 4, we show the effects of ensemble-based DA for different cases: No Data Assimilation (NODA), a single assimilation step, and two assimilation steps. Students will get to this point after creating an initial ensemble, as we did before, and even more, by providing the error distribution of observations jointly their time frequencies. In Fig. 4a, the solid black line denotes the actual state of the system while blue lines represent ensemble member trajectories. Note that uncertainty increases as no DA is performed, and ensemble trajectories are distant from actual states. However, as can be seen in Fig. 4b, a single assimilation step can reduce uncertainty in ensemble trajectories, and even more, it can route ensemble trajectories to actual system states. Uncertainties in blue forecasts are more significant than those in red ones. Of course, uncertainty increases as time evolves (i.e., ensemble trajectories in red). Figure 4c shows that we can reduce uncertainties, in time, by frequently injecting observations into an imperfect numerical forecast. This plot can support the understanding of uncertainties in errors as a function of time. Students can realize that as no actual information is injected into the numerical forecast, model errors will make forecasts diverge from real-life scenarios.



(a) No Data Assimilation (NODA).



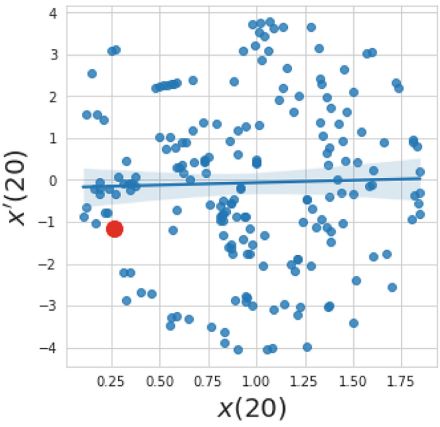
(b) A single assimilation step.



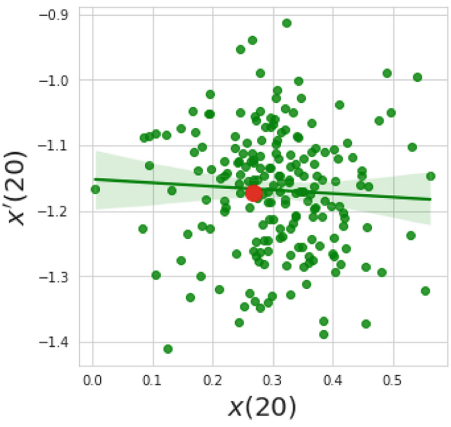
(c) Two assimilation steps.

Fig. 4. The effects of DA in the Duffing equation.

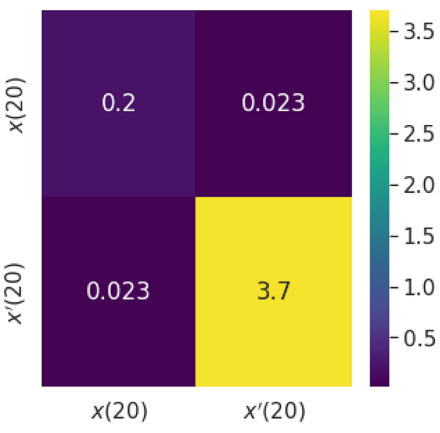
In Fig. 5, we can see how error correlations are developed in time (forecast) and how these are updated after DA (analysis). We show in blue and green how observations (red dots) drive the states of the numerical forecast. Besides, we can see how correlation matrices evolve after assimilation steps (i.e., from positive to negative correlations); we want observed components from the model space to get closer to observations. Similarly, Fig. 6 shows plots for the x_2 variable of the Lorenz-96 model. We can see how uncertainties increase and even more, how correlations are developed in time for model variables in Figs. 6c and 6d. These scenarios can show students how error correlations are developed in model components after forecast steps and analysis ones.



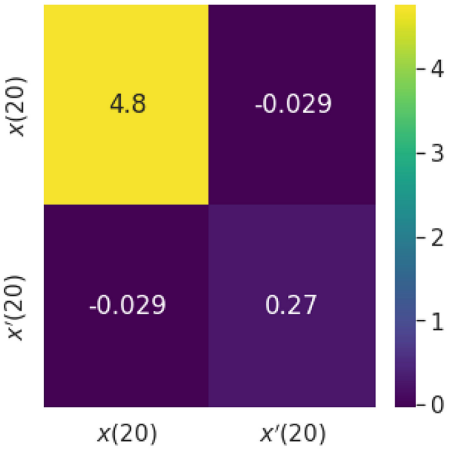
(a) Correlations before DA.



(b) Correlations after DA.

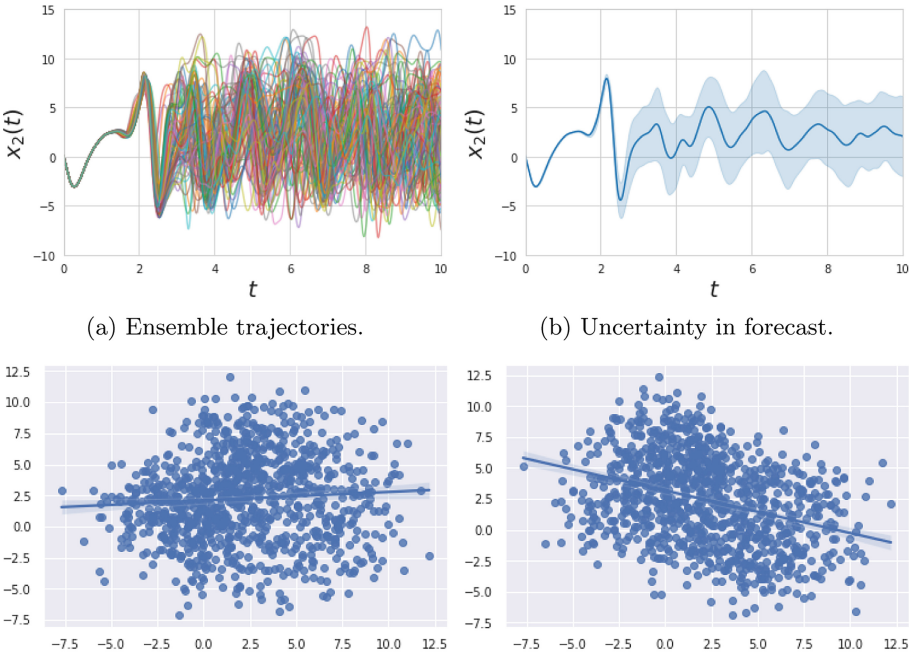


(c) Correlation matrix before DA.



(d) Correlation matrix after DA.

Fig. 5. Initial pool of background members for the experiments. Two dimensional projections based on the two leading directions are shown.



(a) Ensemble trajectories.

(b) Uncertainty in forecast.

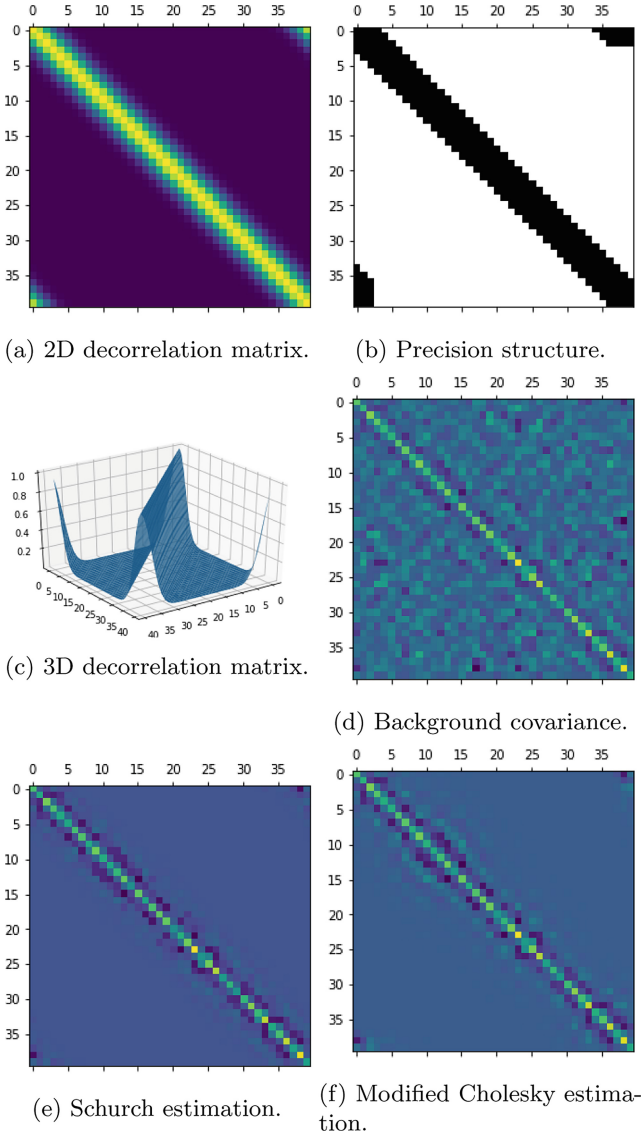
(c) Correlation x_20 and x_22.

(d) Correlation x_20 and x_21.

Fig. 6. Some visualizations of TEDA for the Lorenz-96 model.

Figure 7 shows examples of some TEDA plots about localization aspects for the Lorenz-96 model. In Figs. 7a and 7b, we show prior structures (pre-defined student parameters) of background errors for covariance matrix localization via Schur product and precision matrix estimation, respectively. TEDA also enriches analyses by providing additional plots. For instance, a 3d representation of the decorrelation matrix can be seen in Fig. 7c. This figure clearly shows how we expect error correlations to decrease in space. This can provide a visual explanation of error decay regarding a given radius length. Figure 7d shows the ensemble covariance before localization is applied. As can be seen, sampling errors develop correlation in spatially distant model components. For instance, there is no clear structure of error relationships regarding model dynamics. We can then consider applying any pre-defined structures on the samples to mitigate the impact of sampling errors. Results can be seen in Figs. 7e and 7f for the Schur product and modified Cholesky estimators, respectively.

After all experiment are run, TEDA generates a static HTML file where step by step plots are shown, explanations are given, and comparisons are made in selected DA methods for experiments. Images are stored in sub-folders, these can be utilized as needed.



(a) 2D decorrelation matrix.

(b) Precision structure.

(c) 3D decorrelation matrix.

(d) Background covariance.

(e) Schurch estimation.

(f) Modified Cholesky estimation.

Fig. 7. Some TEDA plots for covariance matrix and precision matrix estimation.

5 Conclusions and Future Research

The TEDA toolbox is an educational software to support the learning and teaching process of DA aspects. TEDA is released with five ensemble-based methods, three numerical models, and a powerful visualization toolbox to facilitate analyses. Since it is OOP (and written in Python following the standard PEP 257), we can easily add new methods and models. Our toolbox is released jointly with

a Jupyter notebook to show a practical case of TEDA and, even more, to exploit cloud resources (i.e., Google Colab by just uploading the Jupyter notebook). TEDA can create multiple DA scenarios with different parameters, for instance, various observational networks, ensemble sizes, inflation factors, among many others. This flexibility makes TEDA exceptional to compare results from multiple methods and simulations. This can be exploited in academic contexts to support the learning process. Our toolbox generates a static HTML file for each simulation wherein information about ensemble generation, analysis steps, and other relevant aspects are discussed. We expect to increase the number of models and methods in future releases of TEDA.

References

1. Ahmed, S.E., Pawar, S., San, O.: PYDA: a hands-on introduction to dynamical data assimilation with python. Fluids **5**(4) (2020). https://doi.org/10.3390/fluids5040225
2. Asch, M., Bocquet, M., Nodet, M.: Data assimilation: methods, algorithms, and applications. SIAM (2016)
3. Duffing, G.: Erzwungene Schwingungen bei veränderlicher Eigenfrequenz und ihre technische Bedeutung. No. 41–42, Vieweg (1918)
4. Evensen, G.: The ensemble kalman filter: theoretical formulation and practical implementation. Ocean Dyn. **53**, 343–367 (2003). https://doi.org/10.1007/s10236-003-0036-9
5. Freeman, K.E., Alston, S.T., Winborne, D.G.: Do learning communities enhance the quality of students' learning and motivation in stem? J. Negro Educ. 227–240 (2008)
6. Levina, E., Rothman, A., Zhu, J., et al.: Sparse estimation of large covariance matrices via a nested lasso penalty. Ann. Appl. Statist. **2**(1), 245–263 (2008)
7. Lorenz, E.N.: Predictability: a problem partly solved. In: Proceedings of the Seminar on Predictability, vol. 1 (1996)
8. Nino-Ruiz, E.D., Sandu, A., Deng, X.: An ensemble Kalman filter implementation based on modified Cholesky decomposition for inverse covariance matrix estimation. SIAM J. Sci. Comput. **40**(2), A867–A886 (2018)
9. OpenDA-Association. Openda (2021). https://github.com/OpenDA-Association/OpenDA
10. Tharayil, S., et al.: Strategies to mitigate student resistance to active learning. Int. J STEM Educ. **5**(1), 1–16 (2018). https://doi.org/10.1186/s40594-018-0102-y
11. Van Rossum, G., Warsaw, B., Coghlan, N.: Pep 8-style guide for python code. Python. org 1565 (2001)
12. Wang, B., Zou, X., Zhu, J.: Data assimilation and its applications. Proc. Natl. Acad. Sci. **97**(21), 11143–11144 (2000)