CZU: 551.501.7:502.3:517.97:004 https://doi.org/10.59295/tehfizuvs25.15

APLICAREA METODEI 4DEnVar ÎN PREDICȚIA POLUĂRII AERULUI
APPLICATION OF THE 4DEnVar METHOD IN AIR POLLUTION PREDICTION

Adrian PALADI[1] email: adrian.paladi@usm.md

[1]Moldova State University

Rezumat. *Acest articol prezintă dezvoltarea și implementarea unui cadru computațional bazat pe Python, construit pe tehnica cvasi-dinamică de asimilare a datelor de tipul ansamblu-variațional 4D (4DEnVar), pentru monitorizarea poluării aerului în Chișinău în cadrul platformei eALERT. Modelul computațional propus poate integra, de asemenea, observații de mediu provenite din mai multe locații, utilizând un model simplificat de tipul advecție-difuzie, ale cărui parametri sunt estimați empiric pe baza variabilității temporale derivate din observații. Un ansamblu de stări inițiale perturbate este utilizat pentru a reprezenta incertitudinea dependentă de flux și pentru a propaga dinamica modelului la fiecare ciclu de asimilare. Prin această abordare hibridă ansamblu-variațională, sistemul generează analize ale concentrațiilor poluanților atmosferici și prognoze pentru o zi cu o acuratețe ridicată. Rezultatele validării indică reducerea erorii medii pătratice și un scor pozitiv al performanței pentru indicatorii monitorizați, confirmând o precizie ameliorată și o eficiență computațională susținută pentru aplicațiile de mediu.*

Cuvinte-cheie*:Aplicații de mediu, calculul variațional, asimilarea datelor 4DEnVar, scripting în Python.*

INTRODUCTION

Medium-range numerical environmental pollution prediction aims to forecast pollution levels over 1-10 days from the current state by solving the initial value problem of a set of partial differential equations (PDEs). Data assimilation (DA) is a key technique used to enhance forecast accuracy by optimally estimating the current state of the environment through the integration of observational data and model forecasts. In this study, a four-dimensional ensemble variational, that is, 4DEnVar, DA method was developed, using air pollution data from the eALERT monitoring platform [1, 2].

Compared to the standard four-dimensional variational, that is, 4DVar, approach, which is widely regarded as one of the most advanced DA techniques, the method employed in this research offers three notable advantages: dynamically estimates the background error covariance (BEC) throughout the assimilation cycle, unlike 4DVar, which typically uses a pre-estimated, static BEC; employs a fully anisotropic ensemble covariance, improving the representation of spatial error structures, and avoids the need for adjoint models, making it more suitable for handling nonlinear problems efficiently.

4DVar assimilation is a technique that uses a perfect forecast model to find the best-fitting model trajectory within a time window, while 4DEnVar is a hybrid method that combines the strengths of 4DVar and ensemble methods to overcome limitations. Variational calculus is the broader mathematical field that underpins both 4DVar and 4DEnVar, which are specific applications that minimize the objective function to fit observations with a model. In essence, 4DEnVar is a type of variational calculus application, but it adds an ensemble component to improve the traditional 4DVar by not relying as heavily on a model fitting or the use of adjoint computational models.

METHODOLOGY

The workflow for implementing the 4DEnVar algorithm for air pollution prediction begins with the preparation of an ensemble of this model possible states that represent uncertainty in the system, including variations in initial pollutant concentrations [3]. The ensemble mean and perturbations from that mean are then computed to characterize the background error statistics. Each ensemble member is subsequently propagated forward in time using the selected atmospheric chemistry and diffusion models across the defined assimilation window, with model outputs stored at the observation times and locations used for assimilation. At each observation time, ensemble mean and perturbations are recalculated, and model data are mapped to observation space to simulate the values that would be observed by each ensemble member. Observational data from sensors are compiled and assigned realistic error estimates that reflect both measurement uncertainty and representativeness error [4]. To suppress false long-range correlations, localization is applied to restrict the spatial influence of ensemble perturbations on observations, and hybrid blending with static climatological background-error covariances may be employed to improve stability in regions with scattered observational coverage. The analysis update is formulated in a reduced control space as a linear combination of ensemble perturbations at the initial time, and the corresponding objective function is minimized iteratively using efficient optimization algorithms such as the conjugate-gradient or limited-memory Broyden-Fletcher-Goldfarb-Shanno quasi-Newton update formula methods to determine the optimal weighting coefficients [3]. Limited memory here refers to an algorithm modification that stores only a small number of past updates, making the algorithm practical for large-scale problem. The objective function minimization balances the fit to observations with a penalty on deviations from the prior (background) state, and re-linearization may be applied if the observation operator exhibits nonlinear behavior. The optimal combination of ensemble perturbations is then applied to the ensemble mean to produce the analysis state at the initial time, with consistent transformation of ensemble perturbations if an analysis ensemble is required to represent updated uncertainty. This analysis state serves as the initial condition for generating forecasts, during which the model is integrated forward, and any necessary chemistry and diffusion adjustments are applied. Diagnostic evaluation and quality control follow, involving the comparison of observations and model equivalents before and after assimilation to compute residual (*observed value - predicted value*) statistics, assess the consistency between ensemble spread and forecast error, and evaluate rank histograms or normalized residuals, that is, it measures how much the members of an ensemble differ from each other, and therefore reflects the uncertainty or confidence of the forecast. Spatial maps of analysis increments, biases, and pollutant concentration changes are inspected to assess the physical plausibility and impact of the assimilation. Finally, the system parameters, including localization radii, inflation factors, hybrid blending weights, and observation error magnitudes, are adjusted iteratively based on diagnostic feedback and forecast performance. Computational implementation requires additional considerations such as parallelization of ensemble runs and evaluations for computational efficiency, data compression for memory management, and systematic validation through observing system simulation experiments or twin experiments prior to deployment in real-time applications.

RESULTS AND DISCUSSION

Fig. 1 illustrates the workflow of the 4DEnVar general algorithm as applied to air pollution modeling and forecast. Also, the schematic of the 4DEnVar system used in the study [5] clearly describes the integration of ensemble-based forecasts with variational data assimilation within the Global and Regional Assimilation and Prediction System - Global Forecast System (GRAPES-GFS) model, which is used for simulating and predicting atmospheric conditions on a global scale. It is widely used in studies on numerical weather prediction, data assimilation, and air pollution modeling. Ensemble members generated from perturbed initial conditions provide a probabilistic representation of the atmospheric state, while observations from satellites and ground stations are assimilated to correct model biases. The analysis increment, guided by an objective function, updates the model's initial conditions over a four-dimensional assimilation window, both temporal and spatial, enhancing the accuracy of air quality and weather predictions.

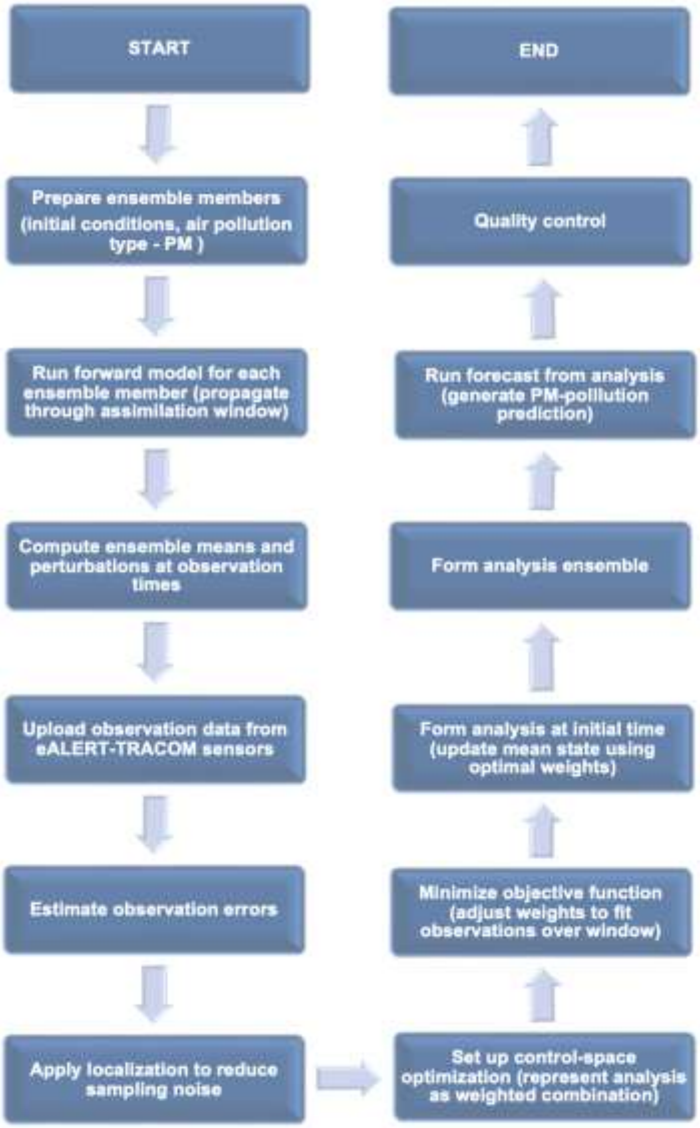
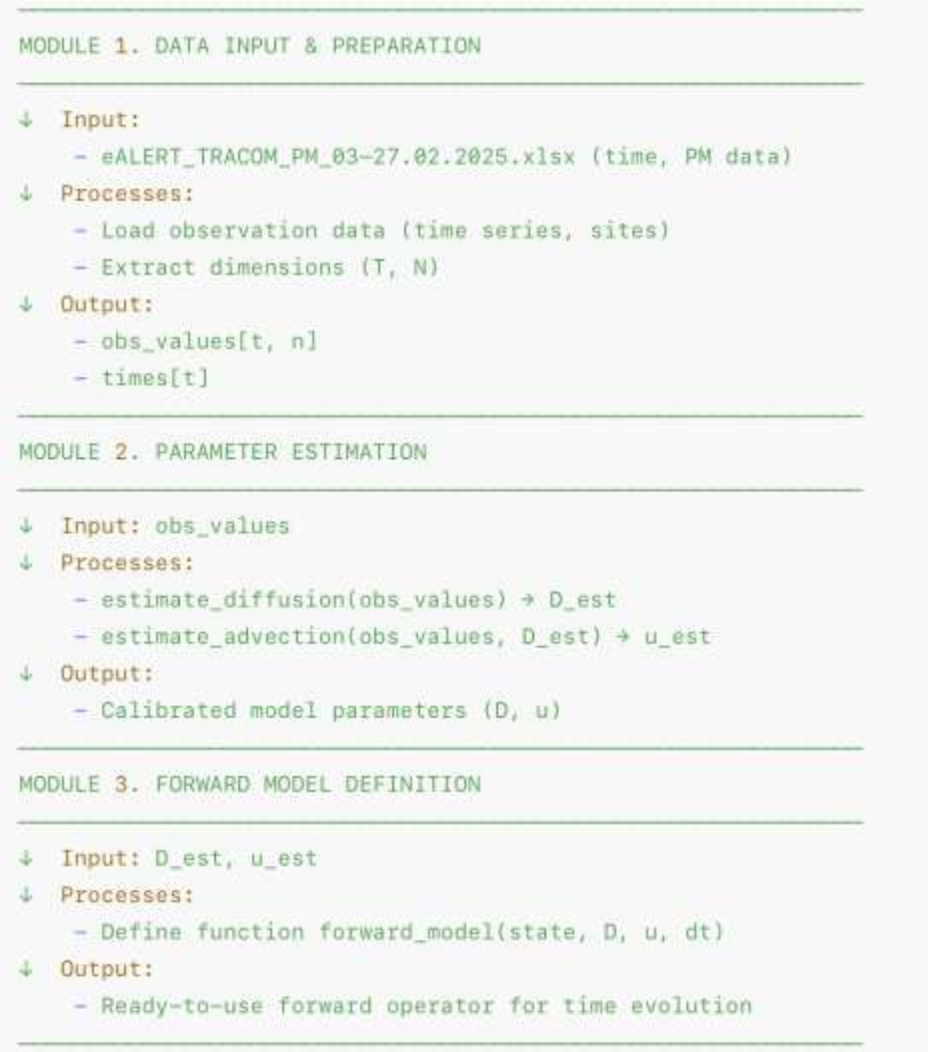


Fig. 1. Workflow of the 4DEnVar algorithm for air pollution prediction.

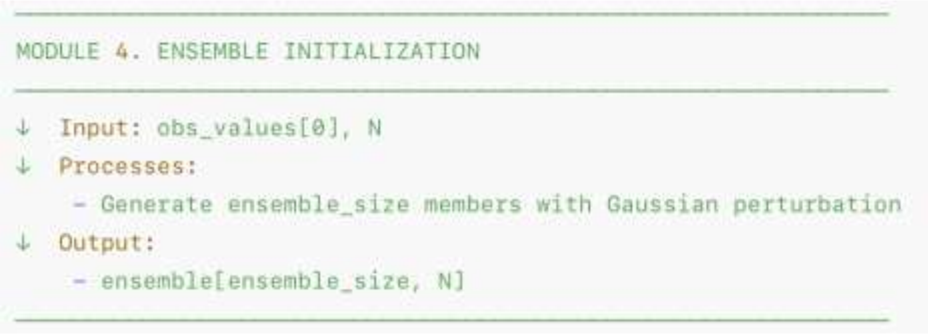
The main components of Python simulation script for the generic 4DEnVar algorithm applied to the eALERT air quality monitoring platform can be grouped by their functional purpose and numbered as 15 modules according to the script's structure as follows:

I. Data and model initialization: 1. Load observation data (reads time series data, extracts time steps and observation matrix, defines number of time steps (T) and number of monitoring stations (N)); 2. Estimate forward model parameters (empirical estimation of diffusion (*D*) and advection (*u*) coefficients from observed air pollution variability); 3. Define calibrated forward model, as follows:



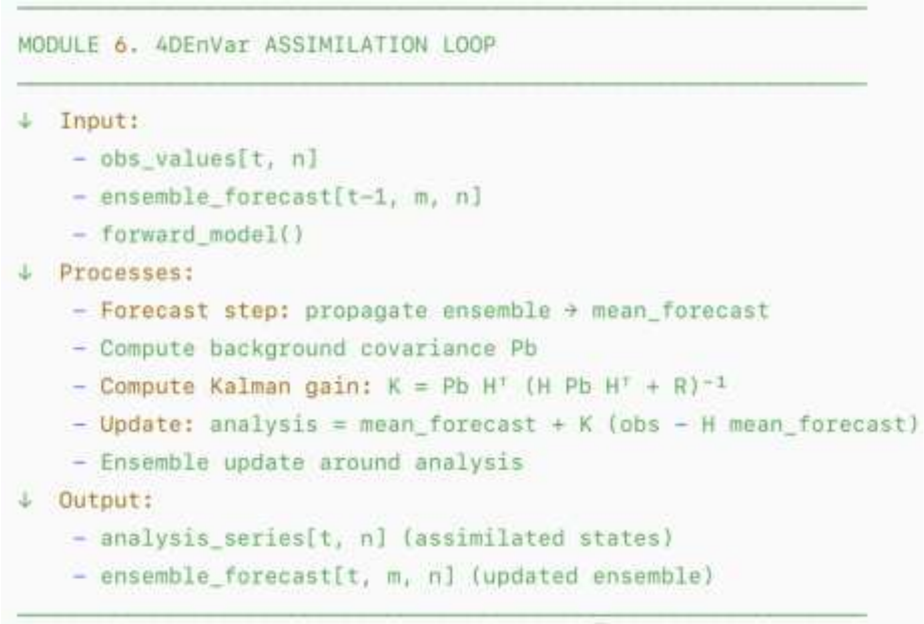
```
────────────────────────────────────────────────────────────
MODULE 1. DATA INPUT & PREPARATION
────────────────────────────────────────────────────────────
↓  Input:
    - eALERT_TRACOM_PM_03-27.02.2025.xlsx (time, PM data)
↓  Processes:
    - Load observation data (time series, sites)
    - Extract dimensions (T, N)
↓  Output:
    - obs_values[t, n]
    - times[t]
────────────────────────────────────────────────────────────
MODULE 2. PARAMETER ESTIMATION
────────────────────────────────────────────────────────────
↓  Input: obs_values
↓  Processes:
    - estimate_diffusion(obs_values) → D_est
    - estimate_advection(obs_values, D_est) → u_est
↓  Output:
    - Calibrated model parameters (D, u)
────────────────────────────────────────────────────────────
MODULE 3. FORWARD MODEL DEFINITION
────────────────────────────────────────────────────────────
↓  Input: D_est, u_est
↓  Processes:
    - Define function forward_model(state, D, u, dt)
↓  Output:
    - Ready-to-use forward operator for time evolution
────────────────────────────────────────────────────────────
```

II. Ensemble initialization and assimilation setup: 4. Initialize ensemble (creates ensemble of perturbed initial states to represent model uncertainty), and 5. Assimilation arrays (pre-allocates arrays for forecasts and analyses):



```
────────────────────────────────────────────────────────────
MODULE 4. ENSEMBLE INITIALIZATION
────────────────────────────────────────────────────────────
↓  Input: obs_values[0], N
↓  Processes:
    - Generate ensemble_size members with Gaussian perturbation
↓  Output:
    - ensemble[ensemble_size, N]
────────────────────────────────────────────────────────────
```

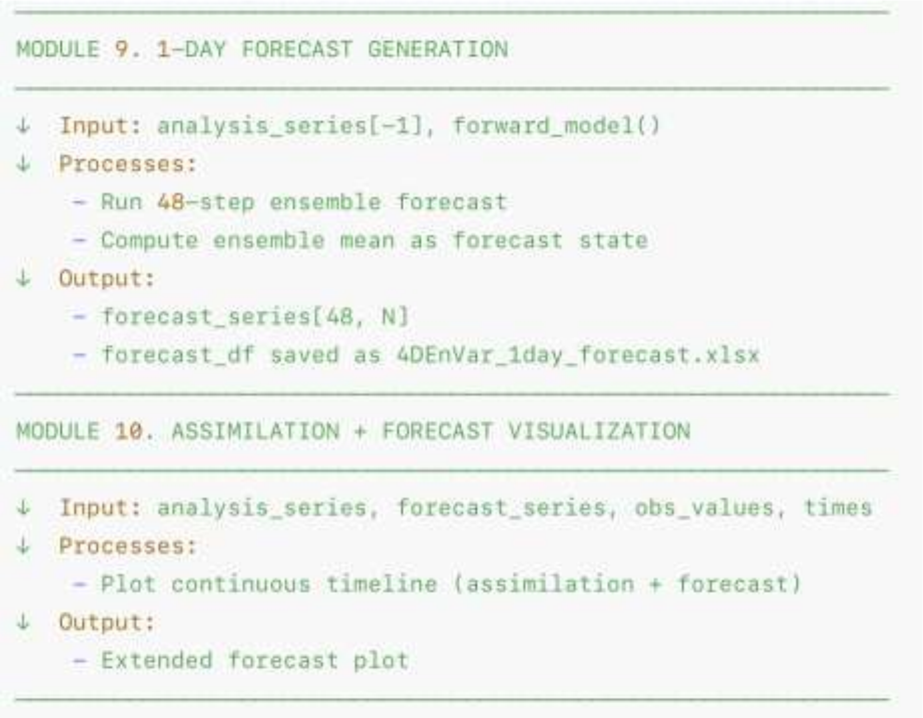
III. Core 4DEnVar data assimilation: 6. 4DEnVar assimilation loop (propagates ensemble through time using the forward model, computes ensemble means and covariance, applies the Kalman gain, updates analysis states with observations, perturbs ensemble members for next cycle), that is:



```
──────────────────────────────────────────────────────────
MODULE 6. 4DEnVar ASSIMILATION LOOP
──────────────────────────────────────────────────────────
↓  Input:
    - obs_values[t, n]
    - ensemble_forecast[t-1, m, n]
    - forward_model()
↓  Processes:
    - Forecast step: propagate ensemble → mean_forecast
    - Compute background covariance Pb
    - Compute Kalman gain: K = Pb Hᵀ (H Pb Hᵀ + R)⁻¹
    - Update: analysis = mean_forecast + K (obs - H mean_forecast)
    - Ensemble update around analysis
↓  Output:
    - analysis_series[t, n] (assimilated states)
    - ensemble_forecast[t, m, n] (updated ensemble)
──────────────────────────────────────────────────────────
```

IV. Output and visualization: 7. Save analysis results, and 8. Visualization of data assimilation.

V. Forecasting and validation: 9. 1-day forecast generation (extends assimilation results 1 day forward (48 steps), uses ensemble mean to predict pollutant concentrations); 10. Plot assimilation and forecast (combines observed, assimilated, and forecasted series for visualization); 11. Forecast skill decay (compares forecast to validation observations, computes root mean square error (RMSE) evolution to evaluate forecast degradation), for example:



```
──────────────────────────────────────────────────────────────
MODULE 9. 1-DAY FORECAST GENERATION
──────────────────────────────────────────────────────────────
↓  Input: analysis_series[-1], forward_model()
↓  Processes:
    - Run 48-step ensemble forecast
    - Compute ensemble mean as forecast state
↓  Output:
    - forecast_series[48, N]
    - forecast_df saved as 4DEnVar_1day_forecast.xlsx
──────────────────────────────────────────────────────────────
MODULE 10. ASSIMILATION + FORECAST VISUALIZATION
──────────────────────────────────────────────────────────────
↓  Input: analysis_series, forecast_series, obs_values, times
↓  Processes:
    - Plot continuous timeline (assimilation + forecast)
↓  Output:
    - Extended forecast plot
──────────────────────────────────────────────────────────────
```

VI. Statistical and performance analysis: 12. Performance metrics computation (calculates per-pollutant RMSE, bias, and correlation for both forecast and analysis, computes skill score, saves results); 13. Skill score visualization (bar plot of RMSE improvement (%) per pollutant indicator); 14. Forecast heatmap analysis (1-day pollutant-wise RMSE and bias heatmaps over forecast lead time); 15. Forecast degradation summary (computes overall forecast deterioration and improvement ratios, produces a compact final figure layout summarizing all data assimilation and forecast diagnostics).

CONCLUSIONS

4DEnVar-based computer simulations are characterized as quasi-dynamic because these simulations are based on or built using hybrid ensemble-variational method as their core data assimilation approach: the 4DEnVar method is not fully dynamic like 4DVar but rather a hybrid, quasi-dynamic approach. It represents the time evolution of forecast errors through an ensemble, providing a dynamic-like treatment of covariances without requiring the adjoint model. Thus, the 4DEnVar data assimilation technique uses ensemble-based flow-dependent covariances that vary in time like a dynamic system, but it does not integrate the full model adjoint during minimization. Instead, it uses ensemble members at multiple time slots to represent the time evolution of errors. Finally, it captures time-dependent relationships (temporal correlations) between model states within the assimilation window, but in a computationally more efficient, static variational framework.

REFERENCES:

1. *eALERT monitoring platform*. eALERT website [citat 07.11.2025]. Disponibil: https://ealert.md
2. SPRINCEAN, V., PALADI, F., LEU, A., JALENCU, M., SAVVA, M., CIOBU, V. *Ghid de utilizare a platformei eALERT*. Ghid științifico-metodic elaborat în baza proiectului de inovare și transfer tehnologic „Crearea platformei eALERT pentru monitorizarea mediului în regim de timp real și avertizarea instantanee a populației din Chișinău în cazul hazardurilor naturale și antropogene

periculoase (2022-2023)”, cu cifrul 22.80015.7007.262T. Chișinău: Editura USM, 2023. 96 p. ISBN 978-9-97562-662-0. Disponibil online: https://ephysimlab.usm.md/wp-content/uploads/Ghid_eAL-ERT_ versiunea_21_12_2023.pdf

3. BANNISTER, R.N. A review of operational methods of variational and ensemble-variational data assimilation. In: *Quarterly Journal of the Royal Meteorological Society*. 2017, vol. 143, no. 703, pp. 607-633. ISSN 0035-9009.
4. KALNAY, E. *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press, 2002, 368 p. ISBN 978-0-52179-629-3.
5. ZHU, S., WANG, B., ZHANG, L. et al. A four-dimensional ensemble-variational (4DEnVar) data assimilation system based on GRAPES-GFS: System description and primary tests. In: *Journal of Advances in Modeling Earth Systems*. 2022, vol. 14, no. 7, article e2021MS002737, 25 p. ISSN 1942-2466.