

macroframe-forecast: Smooth and Constraint-Consistent Forecasting

Sakai Ando¹, Shuvam Das¹, and Sultan Orazbayev¹

¹ International Monetary Fund

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: [↗](#)

Submitted: 03 September 2025

Published: unpublished

License

Authors of papers retain copyright
and release the work under a
Creative Commons Attribution 4.0
International License ([CC BY 4.0](#)).

Summary

The Python package macroframe-forecast generates forecasts that are both temporally smooth and consistent with user-specified constraints. The framework consists of model estimation and reconciliation: first, unconstrained forecasts are produced using user-specified statistical or machine learning models; second, these forecasts are adjusted to satisfy linear equality and inequality constraints while minimizing volatility over time in the forecast horizon. The package offers a user-friendly interface for specifying constraints as strings and supports advanced customization of weights and smoothing parameters. An example - forecasting U.S. GDP with a terminal GDP growth rate target - illustrates the usage.

Statement of need

In forecasting economic time series, statistical models often need to be supplemented with procedures that impose constraints while preserving smoothness over time. For example, GDP forecasts generated using models such as autoregressions or decision trees may not align with the long-term growth rates anticipated by forecasters. In such cases, forecasters aim to adjust the time series so that it converges smoothly to the desired long-term growth path. However, ad hoc constraint imposition, such as manually altering only the terminal value in a long time series, can introduce undesirable discontinuities between the penultimate and terminal values. Other such examples in economic forecasting can be found in Ando (2024) and Ando & Kim (2023).

Systematically imposing constraints while retaining smoothness is important but challenging. Constraints often stem from accounting identities and expert judgment, making their incorporation essential for internal consistency. Smoothness is equally critical, as optimal forecasts typically exhibit less volatility than historical data, as in random walk. Achieving both objectives manually is resource-intensive, especially when dealing with numerous variables and constraints, raising the question of how to systematically impose constraints and smoothness.

Existing packages in R and Python assist forecast reconciliation and smoothing separately but not jointly. For instance, the hts (Hyndman et al., 2021) and FoReco (Girolimetto & Di Fonzo, 2023) packages in R support reconciliation, but the reconciled forecast may not be smooth over time. This is also the case for hierarchicalforecast (Olivares et al., 2024) package in Python. On the other hand, packages, such as smooth (Svetunkov, 2024) and forecast (Hyndman et al., 2024) for R and statsmodels (Seabold & Perktold, 2010) for Python, provide methods to generate smooth forecasts but do not have the functionality to impose constraints.

To our knowledge, no package supports the simultaneous application of both reconciliation and smoothing, and this is the gap that macroframe-forecast attempts to fill. A more detailed explanation of macroframe-forecast can be found in Ando et al. (2025).

Method

The framework consists of two steps, where the first step provides users with a flexible choice of forecasting models, and the second step allows users to adjust the first-step forecasts so that the forecasts are smooth over time and satisfy various constraints, such as accounting identities and pre-specified targets. For the theoretical properties, see Ando (2024).

Suppose there are M time series to forecast, and if we stack the forecast horizon of all the time series, the size is N . The first-step forecast $\bar{y} \in \mathbb{R}^N$ can be generated using any model in the `sktime` (Löning et al., 2019) python package. The second-step forecast \tilde{y} is generated by solving the following quadratic programming problem

$$\tilde{y} = \arg \min_{y \in \mathbb{R}^N} (y - \bar{y})^\top W^{-1} (y - \bar{y}) + y^\top \Phi y \quad \text{s.t.} \quad C_{\text{eq}} y - d_{\text{eq}} = 0, \quad C_{\text{ineq}} y - d_{\text{ineq}} \leq 0.$$

The first term penalizes deviations from the first-step forecast \bar{y} , weighted by W^{-1} , where W is an estimator of the first step forecast error covariance matrix. It ensures that accurate forecasts are adjusted less than less accurate ones. The second term enforces smoothness via Φ , a block-diagonal matrix built from a degenerate penta-diagonal matrix used in the calculation of the Hodrick-Prescott filter (Hodrick & Prescott, 1997). $C_{\text{eq}} y - d_{\text{eq}} = 0$ refers to equality constraints, and $C_{\text{ineq}} y - d_{\text{ineq}} \leq 0$ refers to inequality constraints.

Weight matrix selection

By default, W is estimated using time series cross validation and Oracle Approximating Shrinkage (OAS) (Chen et al., 2010), which shrinks the sample covariance toward a scalar. An alternative is OAS with diagonal target (Ando & Xiao, 2023), which is robust when variables differ in scale. Users can also specify identity weighting or custom matrices.

Smoothness parameters

The smoothness matrix Φ is defined as

$$\Phi = \begin{bmatrix} \lambda_1 F_1 & & & \\ & \ddots & & \\ & & \lambda_M F_M & \end{bmatrix}, \quad F_i = \begin{bmatrix} 1 & -2 & 1 & 0 & & & \\ -2 & 5 & -4 & 1 & & & \\ 1 & -4 & 6 & -4 & & & \\ 0 & 1 & -4 & 6 & & & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & 6 & -4 & 1 & 0 \\ & & & -4 & 6 & -4 & 1 \\ & & & 1 & -4 & 5 & -2 \\ & & & 0 & 1 & -2 & 1 \end{bmatrix}, \quad \lambda_i = \frac{\lambda_i^*}{\sigma_i^2},$$

where $i = 1, \dots, M$. Default parameter values λ_i^* follow HP filter conventions (e.g., 1600 for quarterly data). It is then scaled by the first step forecast error variance σ_i^2 , obtained from the diagonal elements of the weight matrix W for each time series i , to ensure unit invariance. Setting $\lambda_i^* = 0$ disables smoothing for time series i , reducing the problem to pure reconciliation.

An example: single-variable GDP forecast

Using U.S. nominal GDP data (1950-2024), we forecast the values for 2025-2030 under the constraint that 2030 growth rate equals 4 percent. When the column name of the variable to

be forecasted is GDP, the constraint is expressed as $\text{GDP}_{2030} - 1.04 * \text{GDP}_{2029} = 0$. The MFF class generates the first-step forecast via the default pipeline and reconciles them to satisfy the constraint while smoothing the trajectory. Assuming that the user has read the GDP data as a pandas dataframe named `df0` with index year and GDP, the forecasts (both first and second-step) can be generated as follows:

```
from macroframe_forecast import MFF
from macroframe_forecast.examples import generate_example_GDP_df

df0 = generate_example_GDP_df()
m = MFF(df0, equality_constraints=["GDP_2030 - 1.04 * GDP_2029"])
m.fit()
```

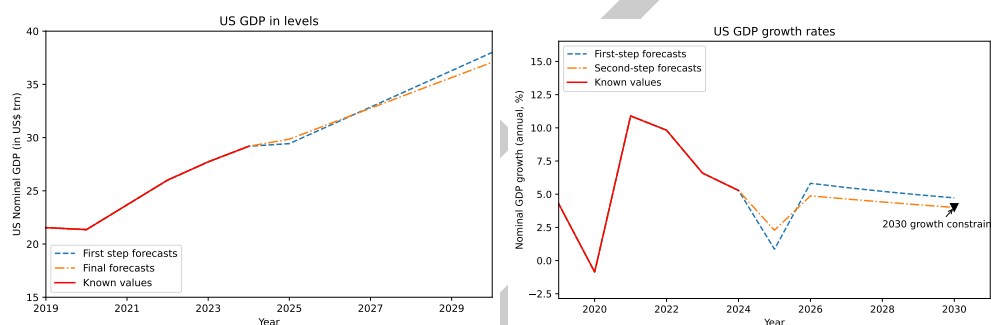


Figure 1: US GDP forecasts: (a) levels and (b) growth.

Figure 1: Forecasts of U.S. nominal GDP, 2025–030, under the 4 percent growth constraint in 2030.

Source: IMF April 2025 World Economic Outlook database and authors' calculations.

Note: Panel (a) shows the forecast of annual US GDP in trillion USD. Panel (b) shows the growth rates computed from the level forecasts.

The second-step forecast can also be customized by providing exogenously defined first-step forecast, weight matrix, or the smoothness parameters. These can be achieved by using the function Reconciliation available in the package.

Acknowledgements

We thank Mingmei Xiao, Yuki Sato, and Doga Bilgin for their earlier contributions.

References

- Ando, S. (2024). Smooth Forecast Reconciliation. *IMF Working Papers*, 2024(066), 1. <https://doi.org/10.5089/9798400268922.001>
- Ando, S., Das, S., & Orazbayev, S. (2025). A Python Package to Assist Macroframework Forecasting. *IMF Working Papers*, 2025(172), 1. <https://doi.org/10.5089/9798229023535.001>
- Ando, S., & Kim, T. (2023). Systematizing macroframework forecasting: High-dimensional conditional forecasting with accounting identities. *IMF Economic Review*, 72(4), 1386–1410. <https://doi.org/10.1057/s41308-023-00225-8>
- Ando, S., & Xiao, M. (2023). High-Dimensional Covariance Matrix Estimation: Shrinkage Toward a Diagonal Target. *IMF Working Papers*, 2023(257), 1. <https://doi.org/10.5089/9798400260780.001>

- 98 Chen, Y., Wiesel, A., Eldar, Y. C., & Hero, A. O. (2010). Shrinkage algorithms for MMSE
99 covariance estimation. *IEEE Transactions on Signal Processing*, 58(10), 5016–5029.
100 <https://doi.org/10.1109/TSP.2010.2053029>
- 101 Girolimetto, D., & Di Fonzo, T. (2023). *FoReco: Point forecast reconciliation*. <https://cran.r-project.org/package=FoReco>
- 102
- 103 Hodrick, R. J., & Prescott, E. C. (1997). Postwar u.s. Business cycles: An empirical
104 investigation. *Journal of Money, Credit and Banking*, 29(1), 1–16. <http://www.jstor.org/stable/2953682>
- 105
- 106 Hyndman, R., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M.,
107 Petropoulos, F., Razbash, S., Wang, E., & Yasmeen, F. (2024). *forecast: Forecasting*
108 *functions for time series and linear models*. <https://pkg.robjhyndman.com/forecast/>
- 109 Hyndman, R., Lee, A., Wang, E., & Wickramasuriya, S. (2021). *Hts: Hierarchical and grouped*
110 *time series*. <https://CRAN.R-project.org/package=hts>
- 111 L ning, M., Bagnall, A., Ganesh, S., Kazakov, V., Lines, J., & Kir ly, F. J. (2019). *Sktime: A*
112 *unified interface for machine learning with time series*. <https://arxiv.org/abs/1909.07872>
- 113 Olivares, K. G., Garza, A., Luo, D., Chall , C., Mergenthaler, M., Taieb, S. B., Wickramasuriya,
114 S. L., & Dubrawski, A. (2024). *HierarchicalForecast: A reference framework for hierarchical*
115 *forecasting in python*. <https://arxiv.org/abs/2207.03517>
- 116 Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with
117 python. *9th Python in Science Conference*.
- 118 Svetunkov, I. (2024). *Smooth: Forecasting using state space models*. <https://CRAN.R-project.org/package=smooth>
- 119