

MP_02 :

Bond Pricing Model

$$\text{Bond Price} = \left(\sum_{i=1}^n c_i \cdot e^{-r(t_i) \cdot t_i} \right) + P \cdot e^{-r(T) \cdot T} - \text{Accrued Interest}$$

- c_i represents the coupon payment at time t_i
- P is the principal payment at maturity T
- $r(t_i)$ is the continuously compounded zero-coupon rate at time t_i
- $e^{-r(t_i) \cdot t_i}$ is the discount factor

Rate Function Specification

$$r(t_i) = a + b \cdot t_i + c \cdot t_i^2 + d \cdot e^{f t_i} + g \cdot \ln(t_i - h)$$

The parameter h was defined with bounds to satisfy $h < t_{\min}$ to ensure the logarithmic function remains well-defined for all observed maturities.

Methodology

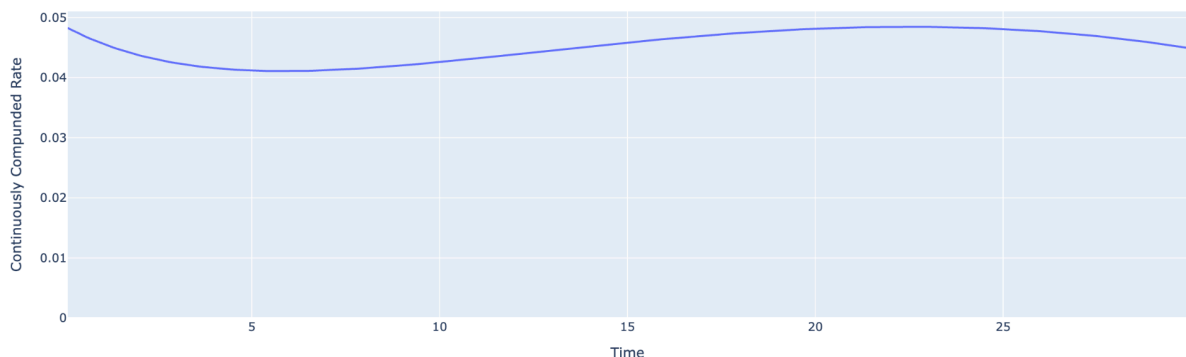
The model parameters were estimated using numerical optimization to minimize the sum of squared errors between predicted and observed clean prices. We tried 2 approaches, employing the L-BFGS-B method (Python) and GRG-Non Linear Solver (Excel). The objective function minimized:

$$SSE = \sum_{j=1}^N \left(\text{Observed Price}_j - \text{Predicted Price}_j \right)^2$$

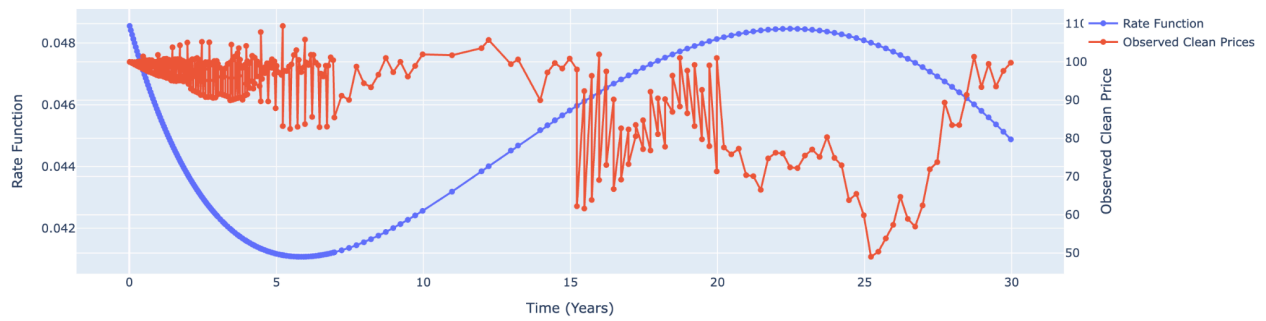
Where N represents the total number of bonds in the dataset (347).

Results and Analysis

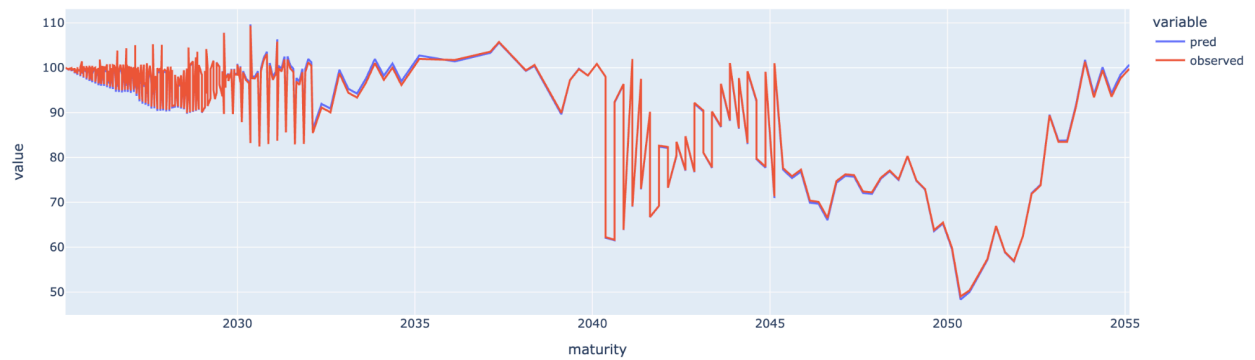
- **R² Score:** 0.998 (In Python), 0.999 (In Excel), **SSE:** 55.935 (In Python), 9.817 (In Excel)
- **Correlation Analysis:** The correlation coefficient between the rate function and observed clean prices is -0.408, confirming the expected inverse relationship between interest rates and bond prices



Rate Function vs Observed Clean Prices [Correlation -0.4080293348156128]



Actual vs Predicted Values [R^2 0.9985829974762582]



Python	a	b	c	d	f	g	h
Value	0.229697	0.0075171	-0.000102	-0.000677	0.04292	-0.0868	-7.9760
Initial_V	0.2720	0.0086	-0.0001	0.0375	0.0150	-0.1225	-7.9728

Excel	a	b	c	d	f	g	h
Value	0.10191	0.003018093	-0.0000608	0.038793	0.018854	-0.04339	-9.6344
Initial_V	0.2720	0.0086	-0.0001	0.0375	0.0150	-0.1225	-7.9728

Note : The SSE can be reduced further, it all depends on what initial values and model you are using. We have executed the model in Excel and Python.