The Rise and Fall of Ideas' Popularity (Supplementary material)

Piero Mazzarisi[s], Alessio Muscillo[m], Claudio Pacati[s], and Paolo Pin[s,b]

[s]Department of Economics and Statistics, Università di Siena, Italy
[m]Department of Economics, Statistics and Business, Universitas Mercatorum, Italy
[b]BIDSA, Università Bocconi, Milan, Italy

October 2024

1 Simulations of the model

Remembering the interpretation of α and δ as interest saturation and influencing enthusiasm, the inequalities from Proposition 1 in the paper tell us that the system fluctuates if and only if they are both high enough. Figure 1 shows an example decomposition of the parameter space into stable and unstable regions.

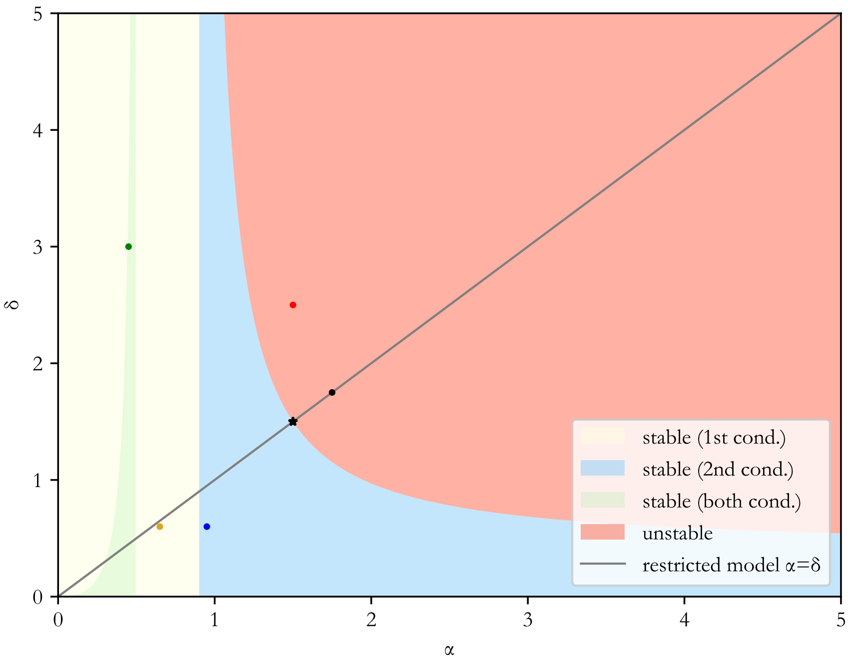


Figure 1: Layout in the (α, δ) parameter space, for $\beta = 0.5$ and $\xi = 0.4$, of the region of instability (red), and of stability due to the first condition (yellow), to the second (blue) or to both conditions (green). The five dots represent the parameters used to produce Figures 2 (yellow dot), 3 (blue dot), 4 (green dot), 5 (red dot), and 6 (black dot). The black star highlights the Hopf bifurcation point of the restricted model with $\alpha = \delta$, the effects of crossing which are shown in Figures 7 and 8.

Figures 2, 3, and 4 show examples of the model dynamics when the parameters do satisfy the first, or the second, or both conditions of stability, respectively. Figures 5 and 6 show examples of parameters satisfying the instability condition.

Figures 7 and 8 illustrate the evolution of the solution of the restricted model when crossing a Hopf bifurcation in the parameter space.

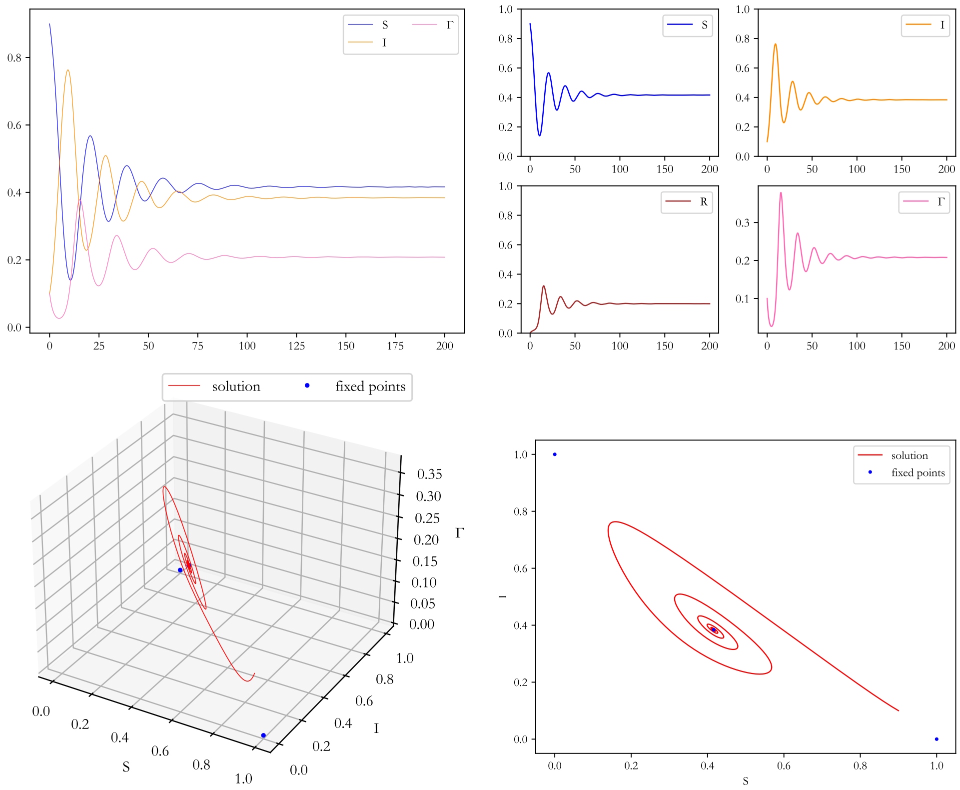


Figure 2: SIRS model with feedback, with fixed point satisfying the first stability condition: $\alpha = 0.65$, $\beta = 0.5$, $\delta = 0.6$, $\xi = 0.4$, $S(0) = 0.9$, $I(0) = \Gamma(0) = 0.1$.

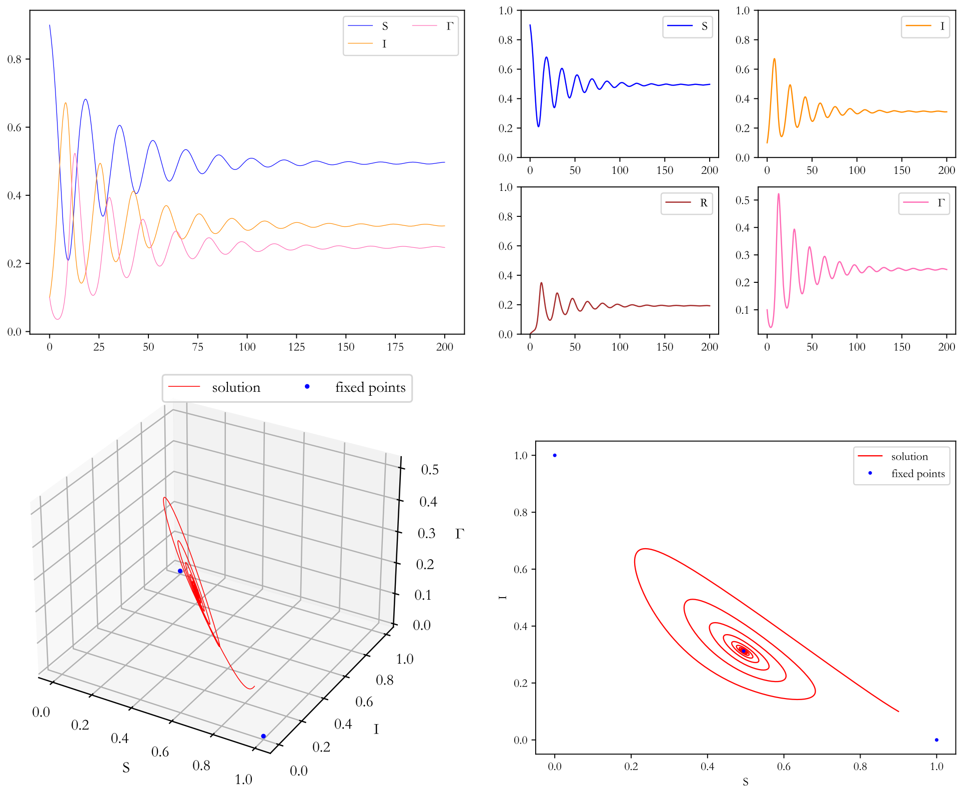


Figure 3: SIRS model with feedback, with fixed point satisfying the second stability condition: $\alpha = 0.95$, $\beta = 0.5$, $\delta = 0.6$, $\xi = 0.4$, $S(0) = 0.9$, $I(0) = \Gamma(0) = 0.1$.

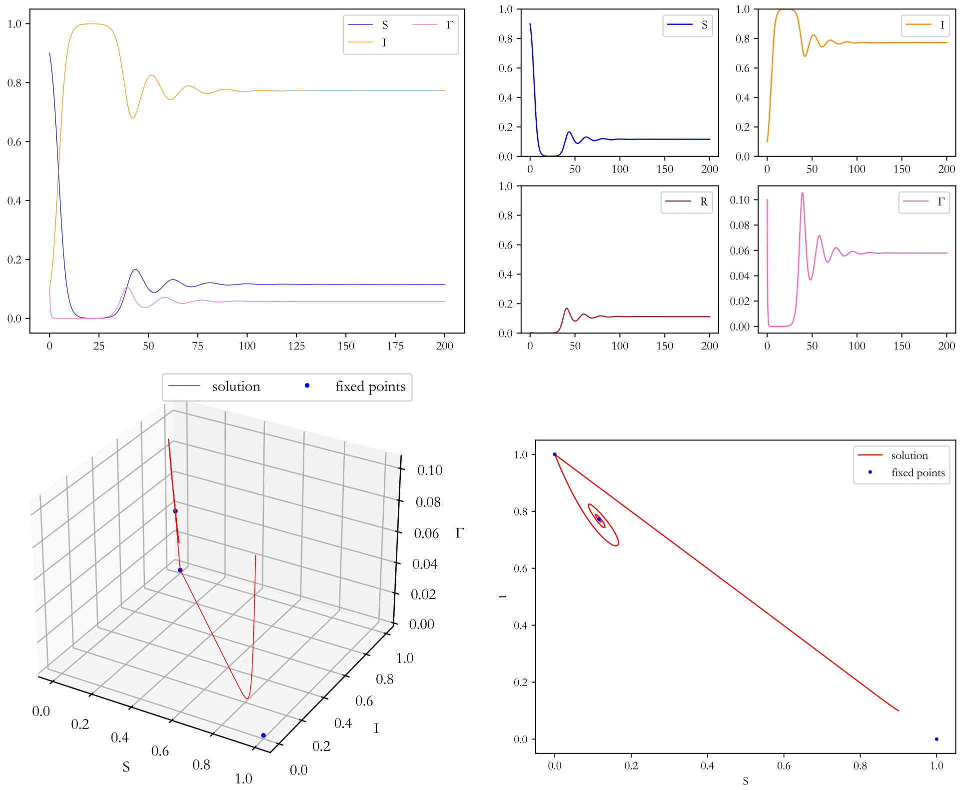


Figure 4: SIRS model with feedback, with fixed point satisfying both stability conditions: $\alpha = 0.45$, $\beta = 0.5$, $\delta = 3$, $\xi = 0.4$, $S(0) = 0.9$, $I(0) = \Gamma(0) = 0.1$.

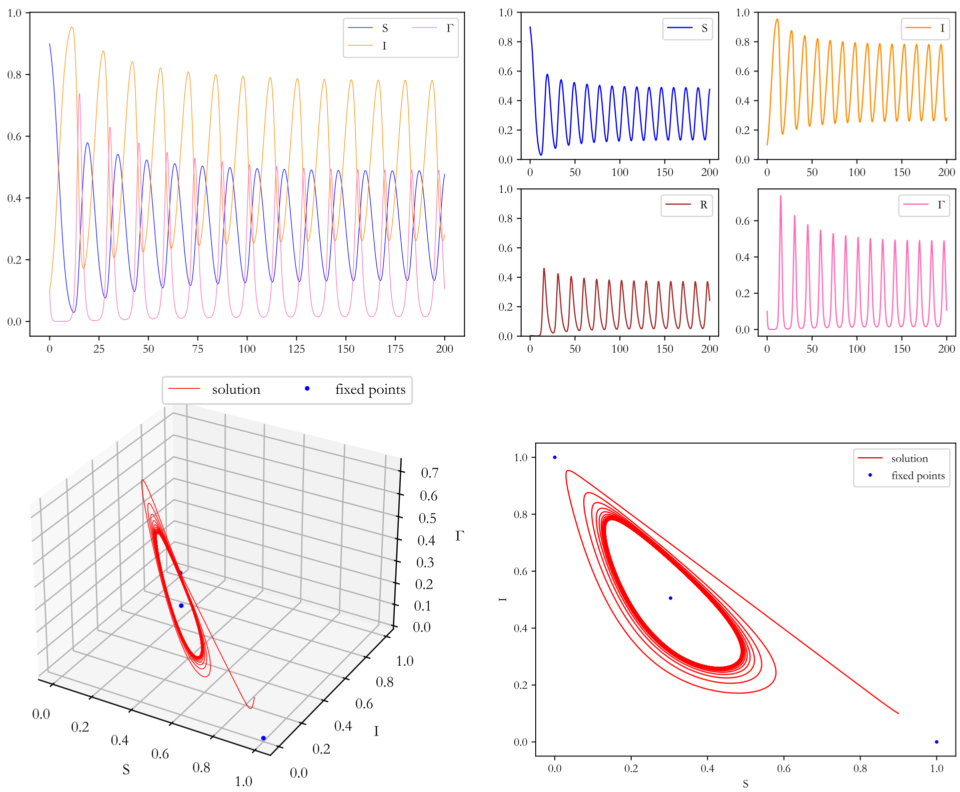


Figure 5: SIRS model with feedback, with fixed point not satisfying the stability conditions: $\alpha = 1.5$, $\beta = 0.5$, $\delta = 2.5$, $\xi = 0.4$, $S(0) = 0.9$, $I(0) = \Gamma(0) = 0.1$.

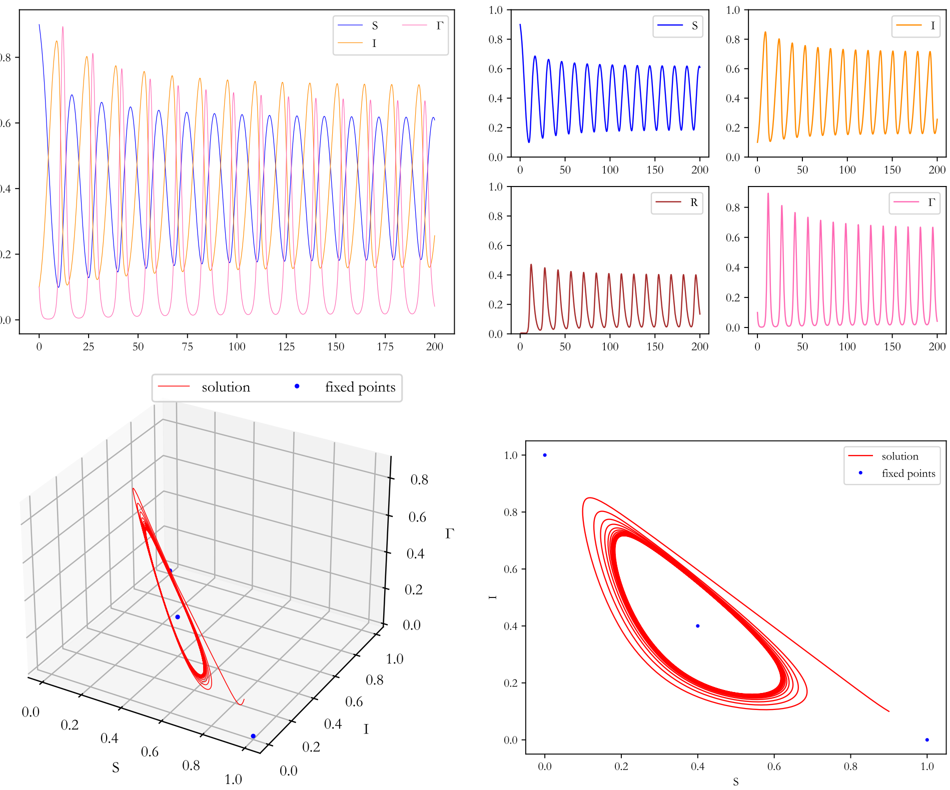
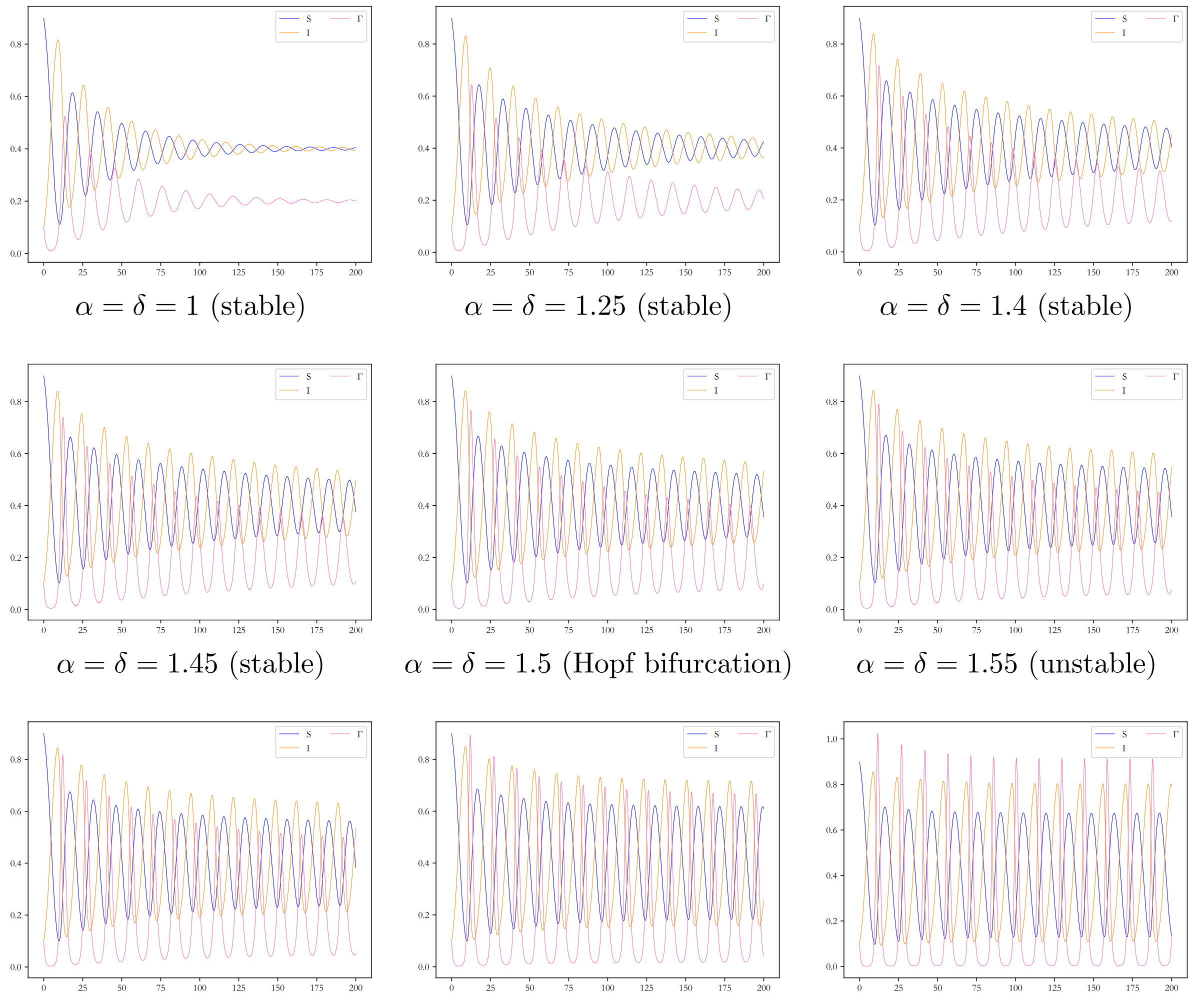


Figure 6: SIRS model with feedback, with fixed point not satisfying the stability conditions: $\alpha = 1.75$, $\beta = 0.5$, $\delta = 1.75$, $\xi = 0.4$, $S(0) = 0.9$, $I(0) = \Gamma(0) = 0.1$.

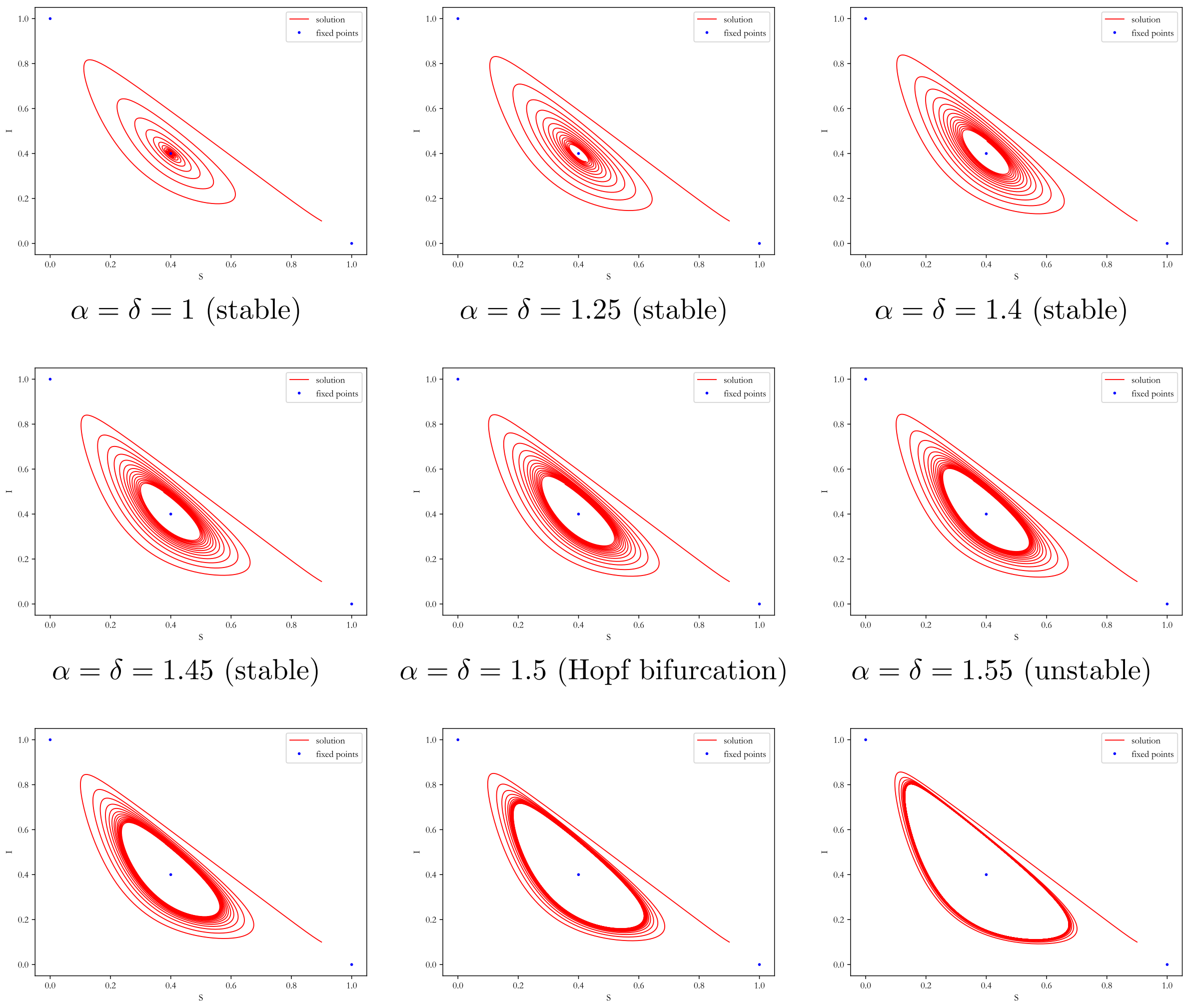


$\alpha = \delta = 1$ (stable) $\alpha = \delta = 1.25$ (stable) $\alpha = \delta = 1.4$ (stable)

$\alpha = \delta = 1.45$ (stable) $\alpha = \delta = 1.5$ (Hopf bifurcation) $\alpha = \delta = 1.55$ (unstable)

$\alpha = \delta = 1.6$ (unstable) $\alpha = \delta = 1.75$ (unstable) $\alpha = \delta = 2$ (unstable)

Figure 7: Evolution of the solution of the restricted version of the SIRS model with feedback when crossing a Hopf bifurcation point in the parameter space; $\beta = 0.5$, $\xi = 0.4$, $S(0) = 0.9$, $I(0) = \Gamma(0) = 0.1$, $\alpha = \delta$ from 1 (upper left – stable), to 1.5 (center – Hopf bifurcation), to 2 (lower right – unstable)



$\alpha = \delta = 1$ (stable) $\quad$ $\alpha = \delta = 1.25$ (stable) $\quad$ $\alpha = \delta = 1.4$ (stable)

$\alpha = \delta = 1.45$ (stable) $\quad$ $\alpha = \delta = 1.5$ (Hopf bifurcation) $\quad$ $\alpha = \delta = 1.55$ (unstable)

$\alpha = \delta = 1.6$ (unstable) $\quad$ $\alpha = \delta = 1.75$ (unstable) $\quad$ $\alpha = \delta = 2$ (unstable)

Figure 8: Evolution of the solution (projection on the (S, I) plane) of the restricted version of the SIRS model with feedback when crossing a Hopf bifurcation point in the parameter space; $\beta = 0.5$, $\xi = 0.4$, $S(0) = 0.9$, $I(0) = \Gamma(0) = 0.1$, $\alpha = \delta$ from 1 (upper left – stable), to 1.5 (center – Hopf bifurcation), to 2 (lower right – unstable)

2 Google Trends analysis

The code used for the analysis is available on GitHub at this link.

2.1 Seasonal decomposition

The seasonal decomposition process uses the Python package `seasonal_decompose` and separates the time series additively into three components:

$$y_t = T_t + S_t + R_t,$$

where

- y_t is the observed time series;
- T_t is the trend, that is, the underlying, slow-moving progression of the series over time;
- S_t is the seasonality, that is, the repeating short-term cycle or pattern within the data (e.g., annual or monthly seasonality);
- R_t is the residual, that is, the random noise or variation left over after removing the trend and seasonality.

The trend is computed as a local average over a rolling time window of length 52 weeks (52 weeks for annual seasonality). The seasonal component is computed by averaging the detrended series for each week of every year. For example, since we are analyzing weekly data with the assumption of a 52-week seasonality, the average seasonal effect is computed by grouping the same week of every year (e.g., the 4th week of every year) and averaging these de-trended data. The seasonal component is thus the pattern that remains after removing the trend. The residual (or irregular) component is what is left after removing the trend and seasonality from the original time series, that is: $R_t = y_t - T_t - S_t$. It represents random fluctuations, outliers, or noise that cannot be explained by the trend or seasonality. Lastly, we normalize the residuals in the range $[-1, 1]$.

2.2 Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) is an algorithm used to measure the similarity between two time series, even if they are of different lengths or have temporal distortions (i.e., variations in speed or alignment). Unlike simple distance metrics such as Euclidean distance, which compares corresponding points in the series directly, DTW finds the optimal alignment between the two series by allowing for non-linear stretching or shrinking of the time axis. This flexibility makes DTW particularly useful for comparing time series that have similar overall patterns but are not perfectly synchronized. In our code, we used the `dtaidistance` Python package which provides efficient implementations of DTW.

The DTW algorithm computes the distance by constructing a cost matrix, where each entry represents the cumulative distance between points from the two series. The optimal alignment path is then identified by minimizing the total accumulated distance along this matrix. The result is a DTW distance, where a smaller distance indicates higher similarity between the two series, and a larger distance suggests greater dissimilarity.

2.3 Distance Computation for Residuals

For each word in our dataset, we computed the Dynamic Time Warping (DTW) distance between the residuals of the word and two reference processes: our endogenous oscillation model and a set of random walks.

- **Distance to our Model.** To compare the residuals to our model, we first generated a family of models where the parameter β, which controls the characteristics of the oscillation, was varied over a specified range $[0.01, 0.3]$. The other parameters of the model are set as: $\xi = 0.1$ and $\alpha = \beta + \xi + \sqrt{\xi(\beta + \xi)} + 0.01$. For each word, we computed the DTW distance between its residuals and the model corresponding to each β. We then selected the model with the β that minimized this distance, ensuring the closest match between the residuals and our model of endogenous oscillations. This process allowed us to quantify how well the model captures the behavior of each word's residuals after removing the trend and seasonality components.

- **Distance to Random Walks.** For comparison, we simulated 20 random walks (based on normal distributions with mean equal to 0 and variance equal to 1), each of the same length as the word's residuals. For each random walk, we computed the DTW distance to the residuals and then averaged these distances to obtain a single value representing the similarity of the word's residuals to a typical random walk. This comparison provides a baseline for determining whether the residuals follow a structured, oscillatory behavior or are better explained by random fluctuations.

By comparing the DTW distances to both the model and the random walks, we can assess whether the residuals exhibit patterns that are more consistent with endogenous oscillations than with random fluctuations.