The CFSpy Python package is introduced to perform reachability analysis of nonlinear system via Chen–Fliess Series.

CFSpy: a Python Library for the Computation of Chen–Fliess Series.

Background: Chen–Fliess series provide an input–output representation of nonlinear control–affine systems.

Goal: provide a Python–exclusive package that computes Chen–Fliess series and uses SciPy Optimize to perform reachability analysis.

Definition (Chen-Fliess Series)

Given the free monoid $(X^*, \cdot, \emptyset)$, and the formal power series $c \in \mathbb{R}^{\ell}\langle\langle X \rangle\rangle$, the *Chen-Fliess series* associated to c is the functional, $F_c[\cdot](t) : L^p[0,T] \to \mathbb{R}^{\ell}$, described by

$$F_c[u](t) = \sum_{\eta \in X^*} (c, \eta) E_\eta[u](t) \tag{11}$$

Main Results

1. Look at the iterated integral backwards
2. Algorithms for the broadcasting computation of the iterated integrals and the Lie derivatives

Definition (Iterated Integral)

Consider the word $\eta = x_{i_1} \cdots x_{i_r} \in X^*$, the *backward* iterative integral of $u \in L^p[0,T]$ associated to η is the operator $H_\eta(\cdot)$ described recursively by

$$H_{x_{i_1}}(H_{x_{i_2}}(\cdots H_{x_{i_r}})) \tag{23}$$

where

$$H_{x_{i_j}}(\cdot) = \begin{cases} \int u_{i_j} & \text{if } j = r, \\ \int (\cdot) & \text{if } j = 0, \\ u_{i_j} \int (\cdot) & \text{otherwise.} \end{cases} \tag{24}$$

Algorithms: broadcasting computation of the backward iterative integral and the Lie derivative.

Algorithm (Block of iterated Integrals)

Inputs Given the truncation length N, the inputs u of the system in (12)

Output The matrix $\mathcal{U}$ of the stacking of iterative integrals $E_\eta[u](t)$ for $|\eta| \leq N$

$U_0 \leftarrow 1 \oplus_v u$

$U_1 \leftarrow [\mathbf{0} \mid S(U_0)_{:,\hat{T}} \Delta]$

$\mathcal{U} \leftarrow U_1$

For k in $\{1, \cdots, N-1\}$ do:

1. $V \leftarrow \mathbf{1}_m \otimes U_k$
2. $v \leftarrow [I_m \otimes \mathbf{1}_{N_{U_k}}]u$
3. $M \leftarrow v \odot V$
4. $U_{k+1} \leftarrow [\mathbf{0} \mid S(M)_{:,-\hat{T}} \Delta]$
5. $\mathcal{U} \leftarrow U_k \oplus_v \mathcal{U}$

Algorithm (Block of Lie derivatives)

Inputs Given the truncation length N, the inputs u of the system in (12)

Output The matrix $\mathcal{G}$ of the stacking of Lie derivatives $L_\eta h(x)$ for $|\eta| \leq N$

$G_0 \leftarrow \mathbf{1}_m \otimes h$

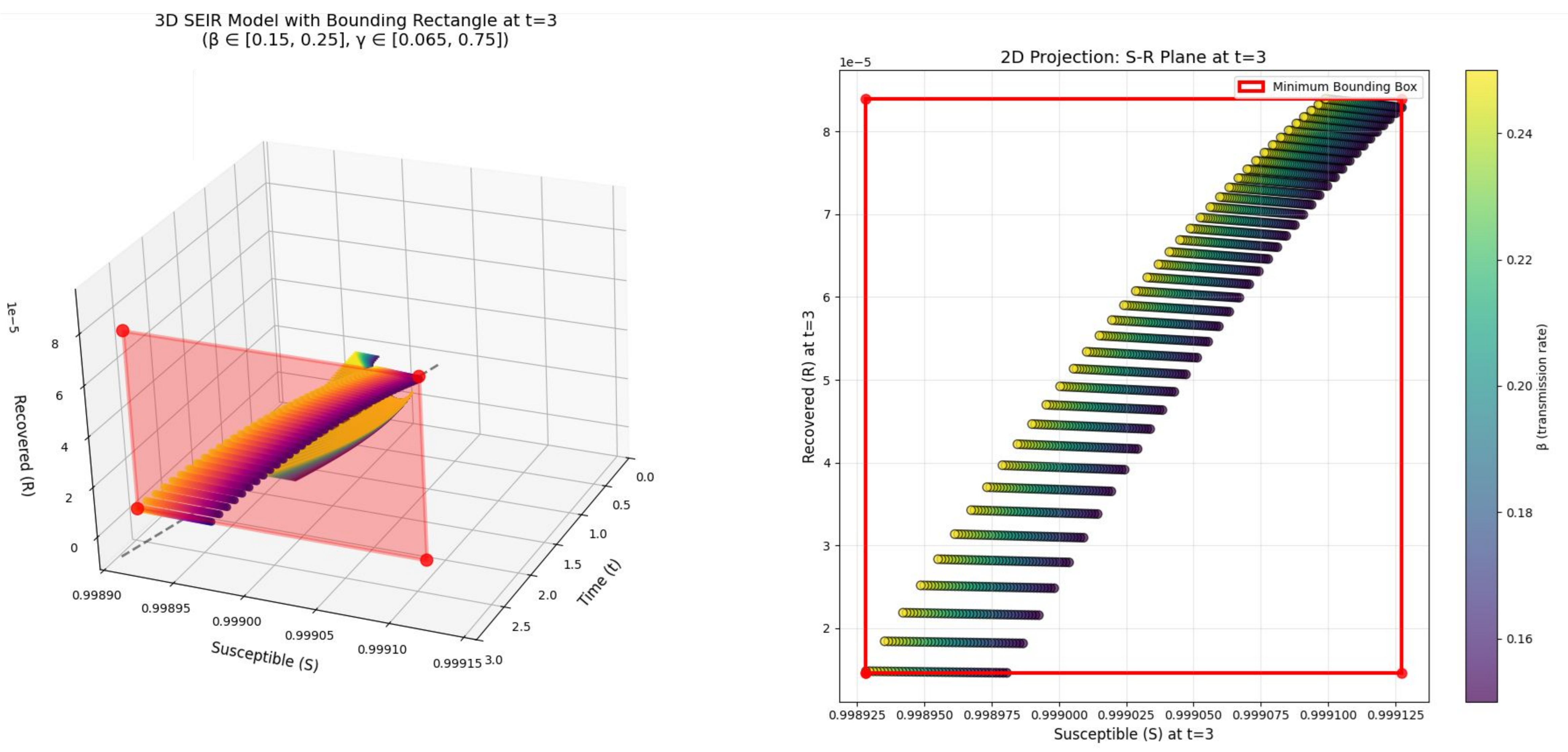
$G_1 \leftarrow \frac{\partial}{\partial x} G_0 \cdot g$

$\mathcal{G} \leftarrow G_1$

For k in $\{1, \cdots, N-1\}$ do:

1. $V \leftarrow \mathbf{1}_m \otimes \frac{\partial}{\partial x} G_k$
2. $v \leftarrow [I_m \otimes \mathbf{1}_{N_{G_k}}]g$
3. $M \leftarrow V \cdot v$
4. $G_{k+1} \leftarrow M$
5. $\mathcal{G} \leftarrow G_{k+1} \oplus_v \mathcal{G}$

Simulation: a SEIR epidemiologic model is used to obtain the minimum bounding box at t=3s.



Limitation: the algorithms need to be optimized to make the computation faster.

Ivan Perez Avellaneda, Ph.D.