СОФИЙСКИ УНИВЕРСИТЕТ
„СВ. КЛИМЕНТ ОХРИДСКИ“

Алгоритъм за права и обратна връзка между реакционни мрежи и системи ОДУ

Петър Чернев

Дипломна работа
Специалност Математическо моделиране в икономиката

Реакционни мрежи

- Произлизат от химията
- Имат приложение при моделиране на системи, къдеот взаимодействията са от трансформтивен характер

$$\mathrm{Na_2SO_4} \xrightarrow{\mathrm{H_2O}} \mathrm{Na^+} + \mathrm{SO_4^{2-}}$$

$$(2\mathrm{Na^+}, \mathrm{SO_4^{2-}}) + (\mathrm{Ba^{2+}}, 2\mathrm{Cl^-}) \longrightarrow \mathrm{BaSO_4} + 2\mathrm{NaCl}$$

Реакционни мрежи

- Ние разглеждаме абстрактни реакции между множество от елементи, който трансформират реагиращите елементи към други от множеството.

$$X \xrightarrow{\mu} Q$$

$$S + X \xrightarrow{k} 2X$$

$$S \xrightarrow{\nu} 2S$$

Динамика на реакционна мрежа

$$A + B \xrightarrow{k_1} C$$

$$D + C \xrightarrow{k_2} B$$

$$\dot{a} = -\underline{k_1 ab}$$

$$\dot{b} = -\underline{k_1 ab} + \underline{k_2 dc}$$

$$\dot{c} = \underline{k_1 ab} - \underline{k_2 dc}$$

$$\dot{d} = -\underline{k_2 dc}$$

Динамика на реакционна мрежа

$$E = \{A_i\}_{i=1}^{n}$$

множество от елементи

$$\rho : \sum_{i=1}^{n} l(A_i)A_i \xrightarrow{k} \sum_{i=1}^{n} r(A_i)A_i$$

обща дефиниция на реакция

$$\varphi(t) = k \prod_{j=1}^{n} a_j(t)^{l(A_j)}$$

функция на потока на реакцията (flow rate).
Зависи само то лявата страна на реакцията.

$$s_i = r(A_i) - l(A_i)$$

стоикиометрия (stoichiometry) на елемента
Показва ефекта от протичането на реакцията върху масата на елемента.

$$\dot{a}_i(t) = s_i \varphi(t)$$

динамиката на масата на елемента.

Динамика на реакционна мрежа

$$\{\rho_i = (l_i, r_i, k_i)\}_{i=1}^{p}$$

множество от реакции – реакционна мрежа

$$\varphi_j(t) = k_j \prod_{i=1}^{n} a_i(t)^{l_j(A_i)}$$

фунцкията на потока за j-тата реакция

$$s_{ij} := r_j(A_i) - l_j(A_i)$$

стоикиометрията на i-тия елемент в j-тата реакция

$$\dot{a}_i(t) = \sum_{j=1}^{p} s_{ij}\varphi_j(t)$$

стоикиометрията на i-тия елемент в j-тата реакция

Векторно представяне

$$\boldsymbol{A} = (A_1, \ldots, A_n)$$
$$\boldsymbol{\rho} = (\rho_1, \ldots, \rho_p)$$

подреждаме елементите и реакциите

$$\boldsymbol{\varphi}(t) = (\varphi_1(t), \ldots, \varphi_p(t))^T$$

вектор на потоците

$$\boldsymbol{S} = (s_{ij}) \in \mathbb{Z}^{(n \times p)}$$

матрица на стоикиометрията

$$\dot{\boldsymbol{a}}(t) = \boldsymbol{S} \cdot \boldsymbol{\varphi}(t)$$

Имплементация

```python
class GompertzBatemanReactionNetwork(ReactionNetwork):
    REACTIONS = """
        P + X -> 2X + P
        S -> P
        P -> Q
    """


network = GompertzBatemanReactionNetwork(rates=[0.9, 0.5, 1.1])
solution = network.solve_ivp(initial_masses=[1, 1, 1])
network.plot(solution)
```

Реакционна мрежа от система ОДУ

$$(S, P, X)$$
$$\dot{s}(t) = -k_1 s(t)$$
$$\dot{p}(t) = k_1 s(t) - k_2 p(t)$$
$$\dot{x}(t) = kp(t)x(t)$$

Като вход приемаме множеството от елементите и системата ОДУ. Определяме уникалните членове в десните страни. Те са достатъчни, за да определим броя на реакциите, техните коефициенти на скорост и левите им страни

$$\{k_1 s(t), k_2 p(t), kp(t)x(t)\}$$

$$S \xrightarrow{k_1}$$
$$P \xrightarrow{k_2}$$
$$P + X \xrightarrow{k}$$

Реакционна мрежа от система ОДУ

От коефициентите на уникалните членове в десните страни определяме матрицата на стоикиометрия. От нея и коефициентите в левите страни, определяме коеифициентите в десните страни.

$$\dot{p}(t) = k_1 s(t) - k_2 p(t)$$

$$s_{21} = 1, \quad s_{22} = -1, \quad s_{23} = 0$$

$$s_{ij} = r_j(A_i) - l_j(A_i)$$

$$S \xrightarrow{k_1} P$$

$$P \xrightarrow{k_2} Q$$

$$P + X \xrightarrow{k} P + 2X$$

Имплементация

```
system = ODESystem("s'=-ks"
                   "x'=np^2x"
                   "p'=ks-npx",
                   ['s', 'x', 'p'])
network = system.to_reaction_network(reaction_rates={'k': 1, 'm': 1, 'n': 1})
```

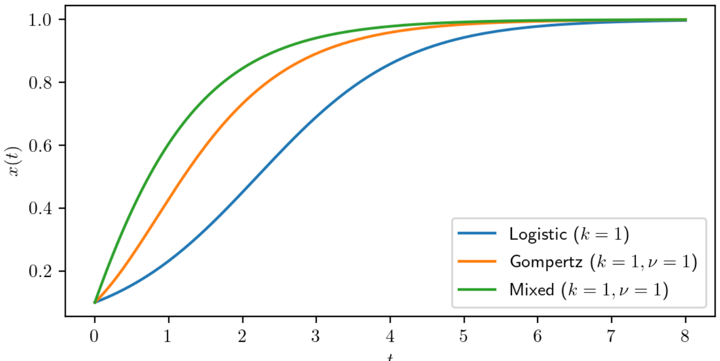
Конструирали сме ReactionNetwork обект,
който има всичката функционалност, както
ако е построен от система реацкии.

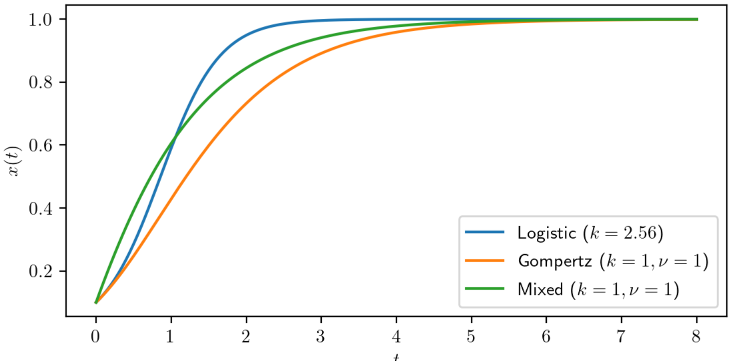
Логистичен модел, модел на Гомперц и смесен модел

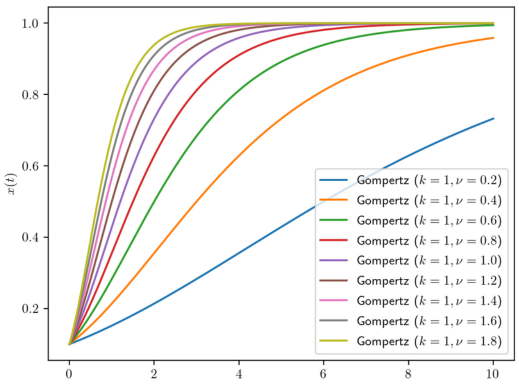
$$S + X \xrightarrow{k} 2X$$

$$S + X \xrightarrow{k} 2X + S$$
$$S \xrightarrow{\nu} Q$$

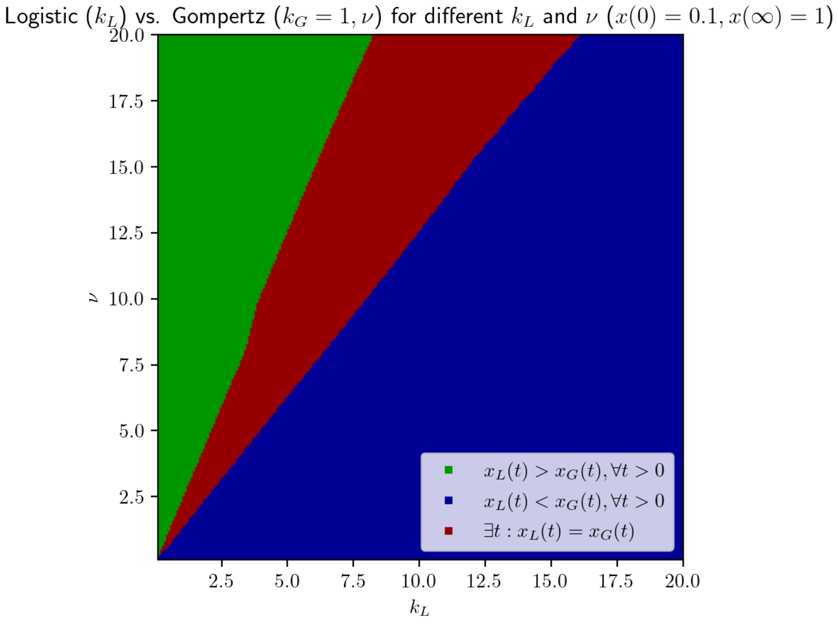
$$S + X \xrightarrow{k} 2X + S$$
$$S \xrightarrow{\nu} X$$



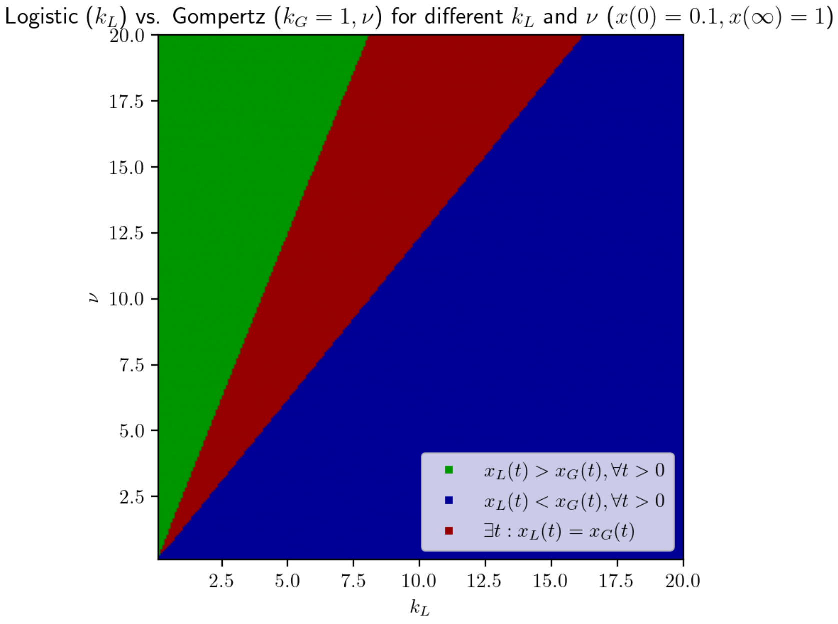




Относително поведение на логистичния модел и модела на Гомпертц



симулирано

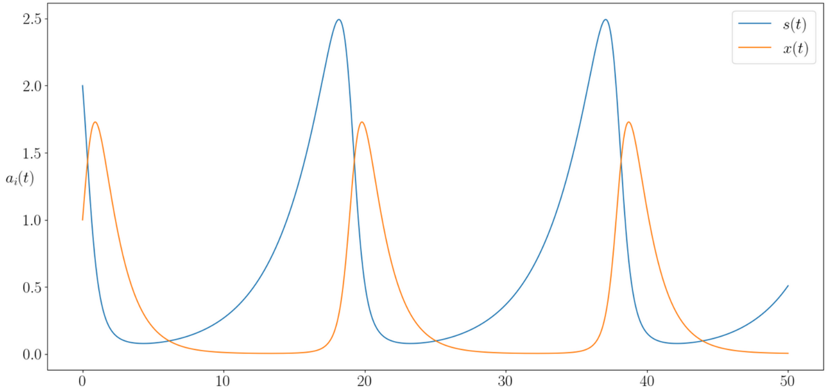


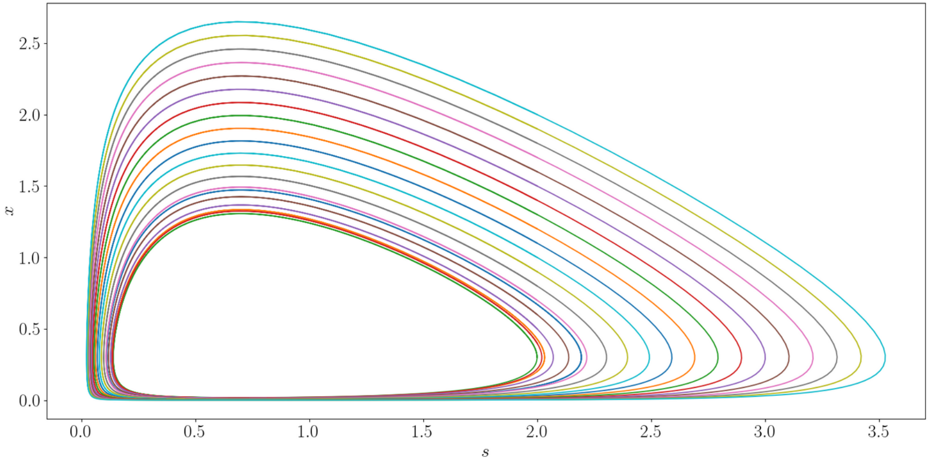
теоретично

Модел на Лотка-Велтера

$$\dot{s}(t) = -ks(t)x(t) + \nu s(t)$$
$$\dot{x}(t) = ks(t)x(t) - \mu x(t)$$

$$X \xrightarrow{\mu} Q$$
$$S + X \xrightarrow{k} 2X$$
$$S \xrightarrow{\nu} 2S$$





Икономически модел базиран на реакционна мрежа

C – фирмен капитал
H – капитал на домакинствата
R – агрегат на суровини и инфраструктура, налични на фирмите
Y – агрегат на стоки и услуги, налични за закупиване

$$\alpha R + \beta C \xrightarrow{k_1} Y + H$$

$$(1-\beta) C \xrightarrow{k_2} R$$

$$Y + \delta H \xrightarrow{k_3} C$$

$$Y \xrightarrow{k_4} C$$

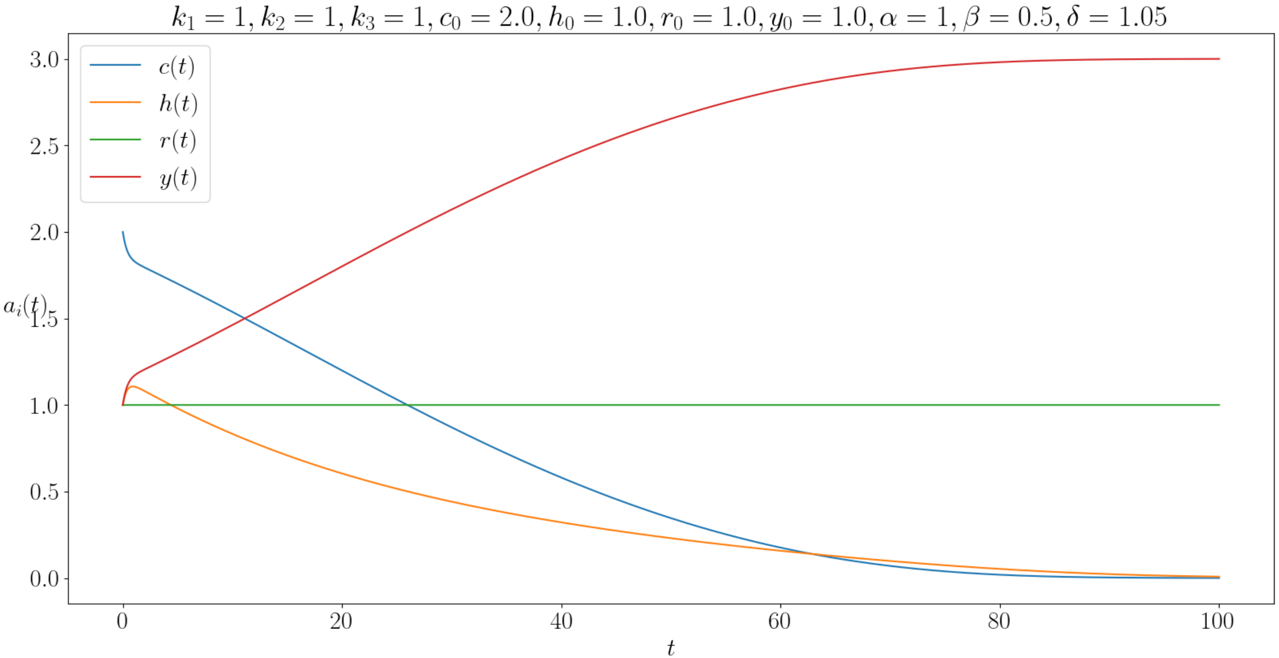
$$\dot{c}(t) = -\beta k_1 c^{\beta} r^{\alpha} - (1-\beta) k_2 c^{(1-\beta)} + k_3 h^{\delta} y + k_4 y$$

$$\dot{h}(t) = k_1 c^{\beta} r^{\alpha} - \delta k_3 h^{\delta} y$$

$$\dot{r}(t) = -\alpha k_1 c^{\beta} r^{\alpha} + k_2 c^{(1-\beta)}$$

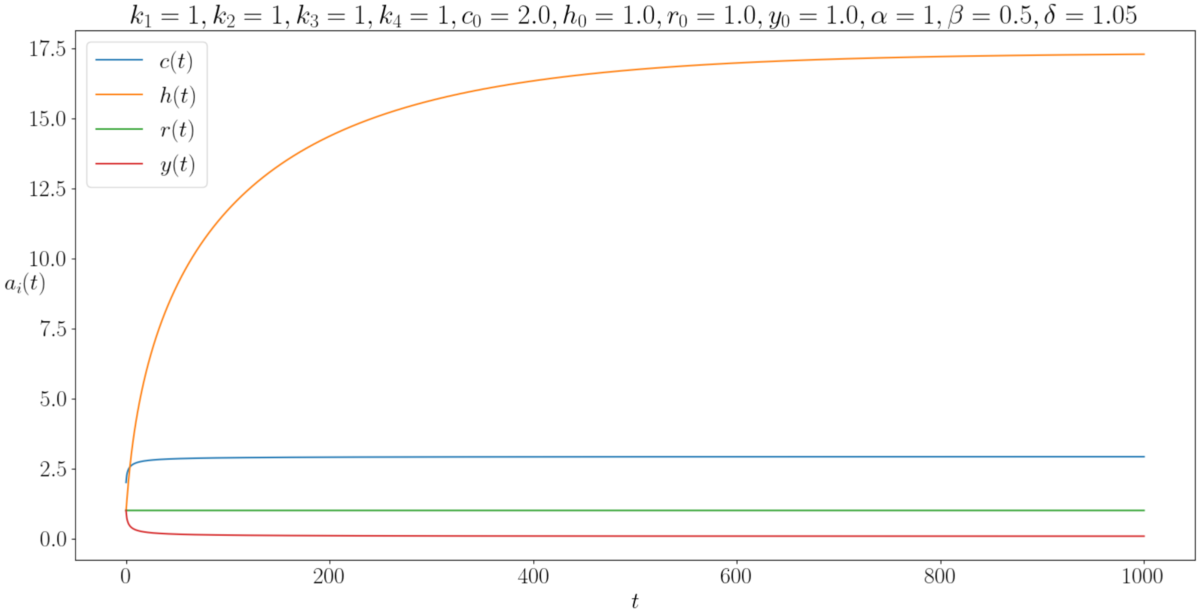
$$\dot{y}(t) = k_1 c^{\beta} r^{\alpha} - k_3 h^{\delta} y - k_4 y$$

Икономически модел без износ



Динамика на модела без износ на стока. Виждаме, че целият капитал в системата е “изтекъл” и се е трансформирал в стоки и услуги, поради закупуването на суровини и инфраструктура от фирмите от източник извън системата.

Икономически модел с износ



Динамика на модела с износ. В равновесие фирмите и домакинствата имат определен капитал, а наличните стоики и услуги са 0. Това означава, че производството и продажбте са в баланс.

Разширение на модела с контрол

Добавена е функционалност за упражняване на контрол над системата, като се дефинира функция на управление

$$\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{a})$$

зависеща от масите на елементите и дефниране на коефициентите на скоростите на реакциите и параметрите на системата като функции на нея:

$$\boldsymbol{k} = (k_1, \ldots, k_p)^T = \boldsymbol{k}(\boldsymbol{u}),$$
$$\alpha = \alpha(\boldsymbol{u}), \beta = \beta(\boldsymbol{u}), \delta = \delta(\boldsymbol{u})$$

На всяка стъпка от интегрирането, обновяваме тези параметри на базата на стойността на функцията на контрол в този момент.