Adaptive algorithms for the numerical simulation of slow-fast oscillator networks

Marie Postel

LJLL, UPMC Univ Paris 06 et EPI MYCENAE

Frédérique Clément (EPI MYCENAE, Inria Paris-Rocquencourt), Alexandre Vidal (LaMME, Univ Evry et EPI MYCENAE)

Mathematical context

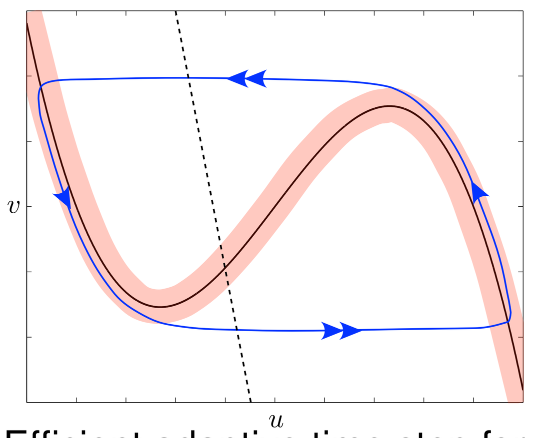
Slow-fast (SF) systems

$$\dot{u} = f(u, v, \varepsilon),\quad \dot{v} = \varepsilon g(u, v, \varepsilon),$$

- $0 < \varepsilon << 1$: time scale separation parameter (fixed small);
- $u \in \mathbb{R}^m$: fast variables; $\dot{u}$: fast dynamics;
- $v \in \mathbb{R}^n$: slow variables; $\dot{v}$: slow dynamics;
- f and g assumed to be smooth.

While (u, v) lies in a $O(\varepsilon)$ neighborhood of $\{f = 0\}$, both u and v evolve slowly ($O(\varepsilon)$ speed). Otherwise, u is fast ($O(1)$ speed).

Example: FitzHugh-Nagumo (FHN) $\quad \dot{x} = -y + x^3 - \lambda x,\quad \dot{y} = \varepsilon(a_0 x + a_1 y + a_2)$



- black: nullclines;
- blue: limit cycle;
- simple arrow: slow motion;
- double arrow: fast motion;
- pink: $O(\varepsilon)$ neighborhood containing the slow motions.

Efficient adaptive time step for numerical simulations:

$$\min\left(\Delta t_{\max}; \frac{\Delta t}{||f(u^k, v^k, \varepsilon)||}\right)$$

Network of N SF oscillators driven by global variables

$$\left.\begin{aligned} \dot{u_j} &= f_j(X, \sigma, \varepsilon), \\ \dot{v_j} &= \varepsilon g_j(X, \sigma, \varepsilon), \end{aligned}\right\} \quad j \in [\![1, N]\!] \qquad \dot{\sigma} = h(X, \sigma, \varepsilon). \tag{1}$$

where $u_j \in \mathbb{R}^m, v_j \in \mathbb{R}^n$, $\sigma \in \mathbb{R}^q$, $X = (u_j; v_j)_{j \in [\![1,N]\!]}$.

$$(f_j; g_j) : \mathbb{R}^{(m+n)N} \times \mathbb{R}^q \times [0, \varepsilon_0] \to \mathbb{R}^m \times \mathbb{R}^n$$
$$h : \mathbb{R}^{(m+n)N} \times \mathbb{R}^q \times [0, \varepsilon_0] \to \mathbb{R}^q.$$

Global variable σ can display slow and fast motions too.

Biological application

Intracellular calcium oscillations in GnRH neurons

Neuronal population located in the hypothalamus and responsible for the pulsatile secretion of GnRH (Gonadotropin Releasing Hormone) which drives the endocrine control of the reproductive function in mammals.

Pulsatile oscillations of intracellular calcium level [1] :

- Proper amplitude and pulse frequency in each cell resulting in asynchrony between cells;
- Recurrent synchronization episodes with regular frequency.

Mathematical model (see [2])

Population of N cells.

For each cell ($j \in [\![1, N]\!]$), model based on FHN:

- Fast variable x_j : electrical activity;
- Slow variable y_j : ionic activity;
- Slow variable Ca_j: intracellular calcium level
  - driven by a thresholded action of x_j,
  - acting as a feedback on the x_j dynamics.

Global variable σ representing the state of the network:

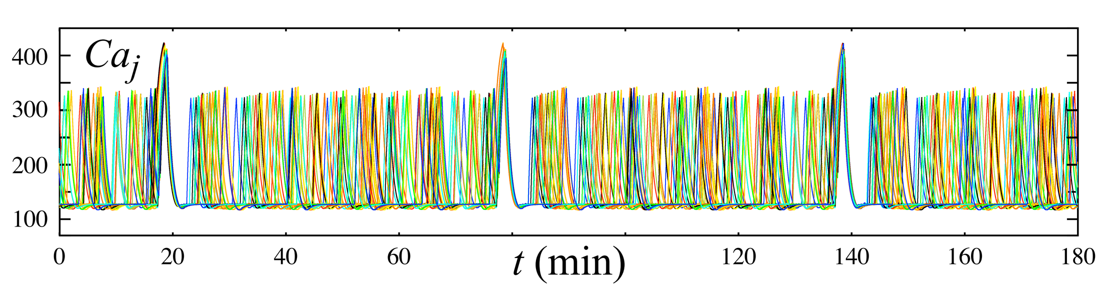
- increases very slowly during the asynchronous phase,
- $\sigma > \sigma_{\text{on}} \to$ change in the x_j dynamics : synchronization.
- quickly decreases when the mean calcium level $< Ca >$ among cells is above a threshold.

$$\left.\begin{aligned} \dot{x_j} &= -y_j + 4x_j - x_j^3 - \phi_{\text{fall}}(Ca_j), \\ \dot{y_j} &= \varepsilon k_j \left(x_j + a_1 y_j + a_2 - \phi_{\text{syn}}(\sigma)\right), \\ \dot{Ca_j} &= \varepsilon\left(\phi_{\text{rise}}(x_j) - v_{Ca}(Ca_j - Ca_{\text{bas}})\right), \end{aligned}\right\} \quad j \in [\![1, N]\!] \qquad \dot{\sigma} = \varepsilon\delta\sigma - \gamma(\sigma - \sigma_0)\phi_\sigma\left(< Ca > - Ca_{\text{desyn}}\right). \tag{2}$$

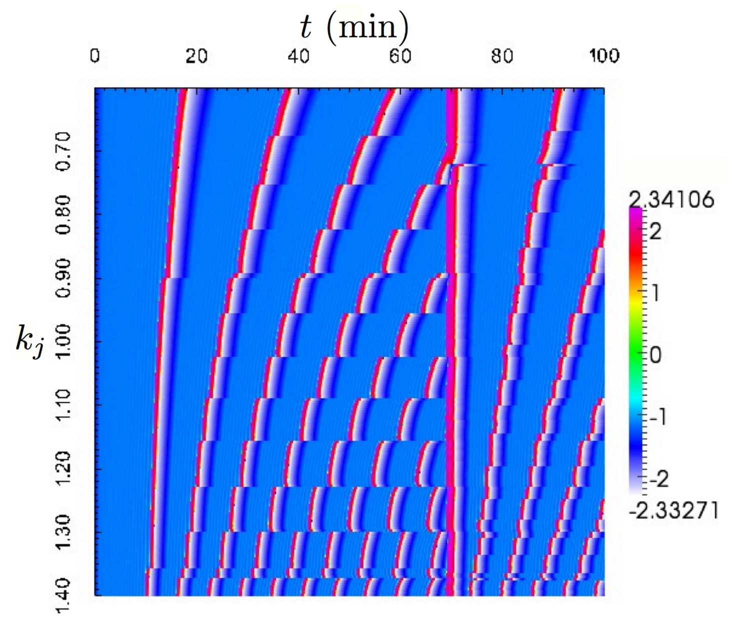
$$\phi_{\text{fall}}(Ca) = \frac{\mu Ca}{Ca + Ca_0}, \quad \phi_{\text{syn}}(\sigma) = \frac{\eta}{1 + \exp(-\rho_{\text{syn}}(\sigma - \sigma_{\text{on}}))},$$
$$\phi_{\text{rise}}(x) = \frac{\lambda}{1 + \exp(-\rho_{Ca}(x - x_{\text{on}}))}, \quad \phi_\sigma(u) = \frac{1}{1 + \exp(-\rho_\sigma u)}.$$

Cell-dependent k_j values:

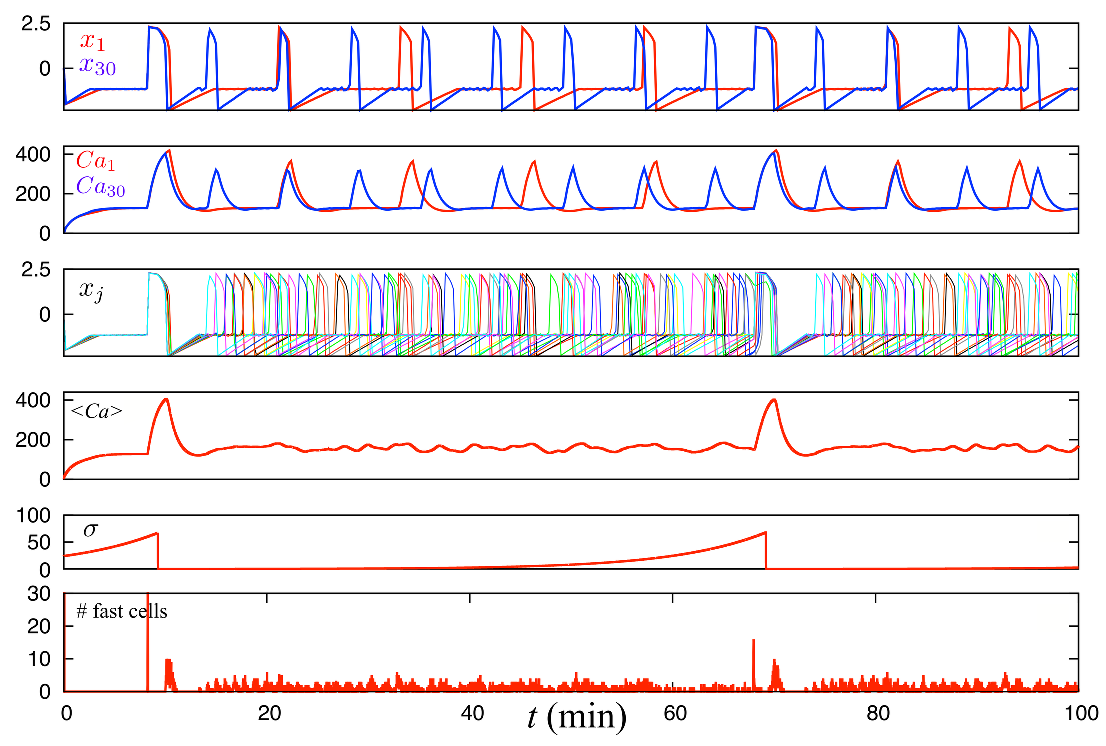
- Variability in calcium oscillations among cells ;
- Asynchrony between synchronization episodes.



Individual calcium time series generated by model (2) with 50 cells.



Solutions x_j of (2) with $N = 200$ and k_j between 0.6 and 1.4.



Solutions of (2) with $N = 30$ and number of cells in fast motion over time.

Goal Simulation and analysis of complex behaviors produced by networks of type (1). For the example model (2):

- Impact of the parameter value distribution;
- Partial recruitment of neurons for synchronization episodes;
- Doublet of synchronization episodes (see figure below).

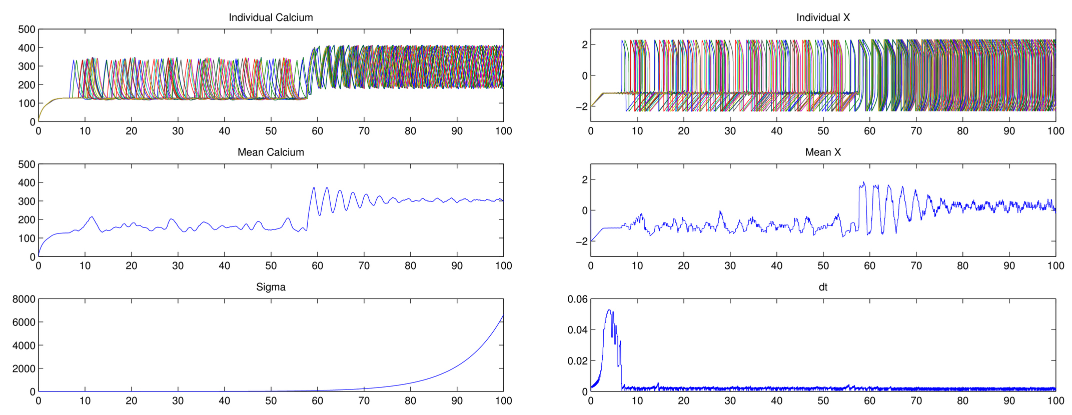
Need for a **fast** and **accurate** simulation tool.

Numerical simulation

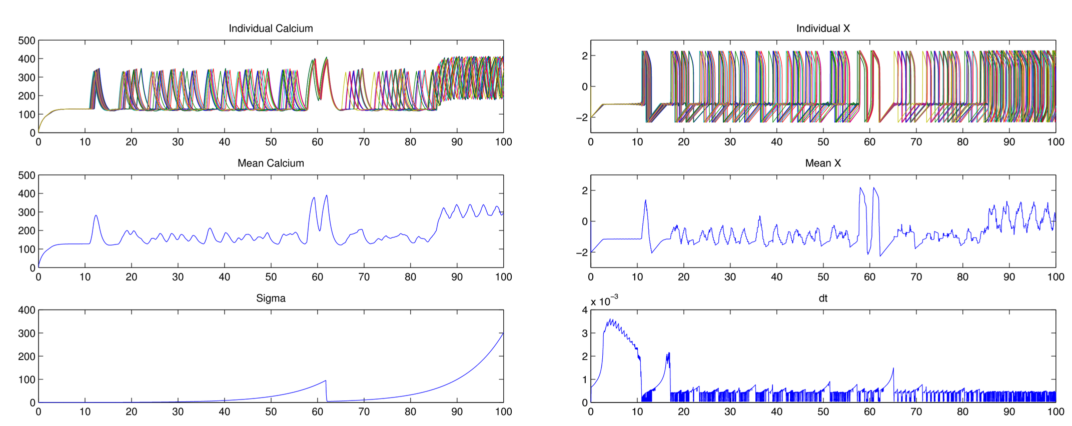
Challenges for simulation

- Large scale problem: $(m + n)N + q$ coupled equations;
- Non-linear dynamics: complex local behaviors;
- Time-step selection: error vs. cpu time.

Classical choices for the adaptive time step based on the global vector field are inefficient **since cells are not synchronized**:



Matlab `ode45` solution with default tolerances (Rel. 10^{-3}, Abs. 10^{-6})



Matlab `ode45` solution with more stringent tolerances (Rel. 10^{-9}, Abs. 10^{-12})

Strategy: taking into account the time scale separation in each cell dynamics.

- Use chosen (elementary) schemes on two time grids:
  - the coarse time grid associated with slow motions,
  - the fine time grid associated with fast motions;
- Choose the refinement factor according to the time scale separation for optimizing the adaptive scheme order.
- At each integration time step, discriminate the cells in slow and fast motions from the fast dynamics evaluation.

Splitting

Preservation of the elementary scheme order

From [3], given a vector field $V = A + \varepsilon B$ and the associated differential operator $L_V = L_A + L_{\varepsilon B}$.

Order 1 integrators: $\quad \mathcal{AB} = e^{\delta L_A} e^{\delta L_{\varepsilon B}}, \quad \mathcal{BA} = e^{\delta L_{\varepsilon B}} e^{\delta L_A}.$

Order 2 integrators (splitting):

$$\mathcal{SABA} = e^{\frac{\delta}{2} L_A} e^{\delta L_{\varepsilon B}} e^{\frac{\delta}{2} L_A}, \quad \mathcal{SBAB} = e^{\frac{\delta}{2} L_{\varepsilon B}} e^{\delta L_A} e^{\frac{\delta}{2} L_{\varepsilon B}}$$

Higher orders obtained with more alternations of L_A and $L_{\varepsilon B}$.

Slow-fast splitting At a given point $(\bar{X}, \bar{\sigma})$ of system (1) phase space, using the evaluations of $f_j(\bar{X}, \bar{\sigma}, \varepsilon)$, we distinguish the N_f cells in slow motion and the N_s cells in fast motions:

- $X_f : (m + n)N_f$ variables (u_j, v_j) of cells in fast motion,
- $X_s : (m + n)N_s$ variables (u_j, v_j) of cells in slow motion,

With these new variables, system (1) writes:

$$\begin{aligned} \dot{X_f} &= F_f(X, \sigma), && \text{with } F_f(\bar{X}, \bar{\sigma}) = O(1) \\ \dot{X_s} &= F_s(X, \sigma), && \text{with } F_s(\bar{X}, \bar{\sigma}) = O(\varepsilon) \\ \dot{\sigma} &= H(X, \sigma). \end{aligned}$$

Associated differential operator: $L = F_f \frac{\partial}{\partial X_f} + F_s \frac{\partial}{\partial X_s} + H \frac{\partial}{\partial \sigma}$

Decomposition of the vector field at $(\bar{X}, \bar{\sigma})$ and splitting:

- if σ is slow, i.e. $H(\bar{X}, \bar{\sigma}) = O(\varepsilon)$:

$$\mathcal{SBAB} \quad \text{with} \quad L_A = F_f \frac{\partial}{\partial X_f}, \quad L_{\varepsilon B} = F_s \frac{\partial}{\partial X_s} + H \frac{\partial}{\partial \sigma}.$$

- if σ is fast, i.e. $H(\bar{X}, \bar{\sigma}) = O(1)$:

$$\mathcal{SABA} \quad \text{with} \quad L_A = F_f \frac{\partial}{\partial X_f} + H \frac{\partial}{\partial \sigma}, \quad L_{\varepsilon B} = F_s \frac{\partial}{\partial X_s}.$$

Note: the splitting depends on which cells are in slow and fast motions at a given point.

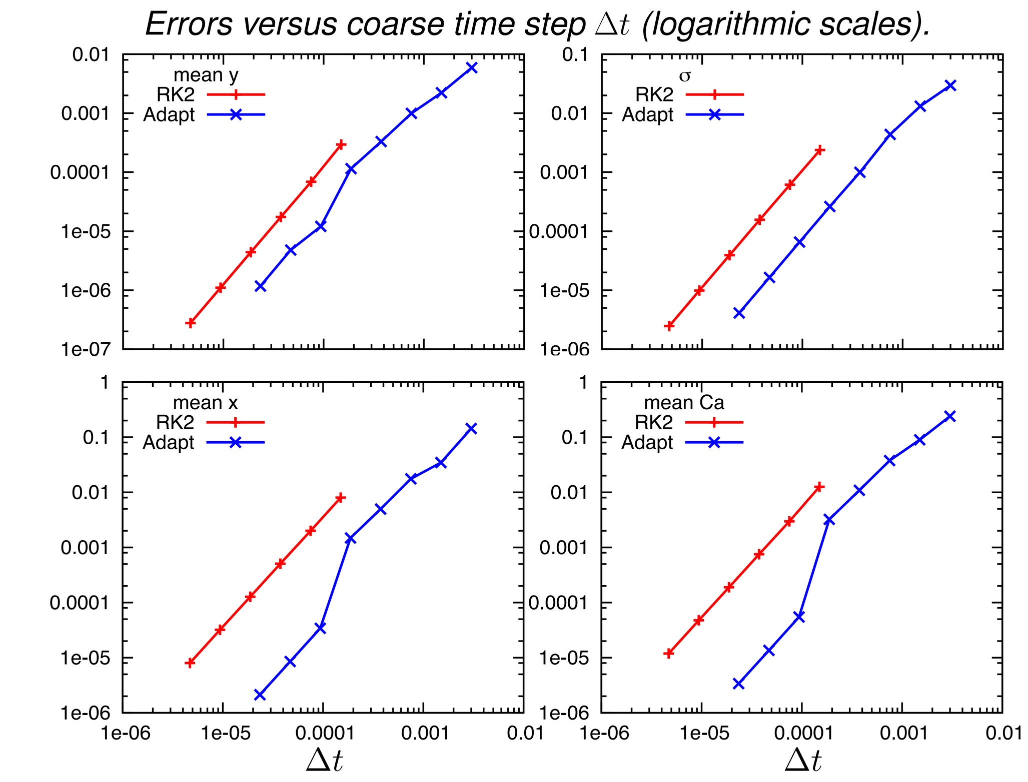
Adaptive scheme

One step algorithm To optimize the performances of the slow-fast splitting algorithm applied to model (2), we use RK4 on the coarse grid and RK2 on the fine grid as elementary scheme.

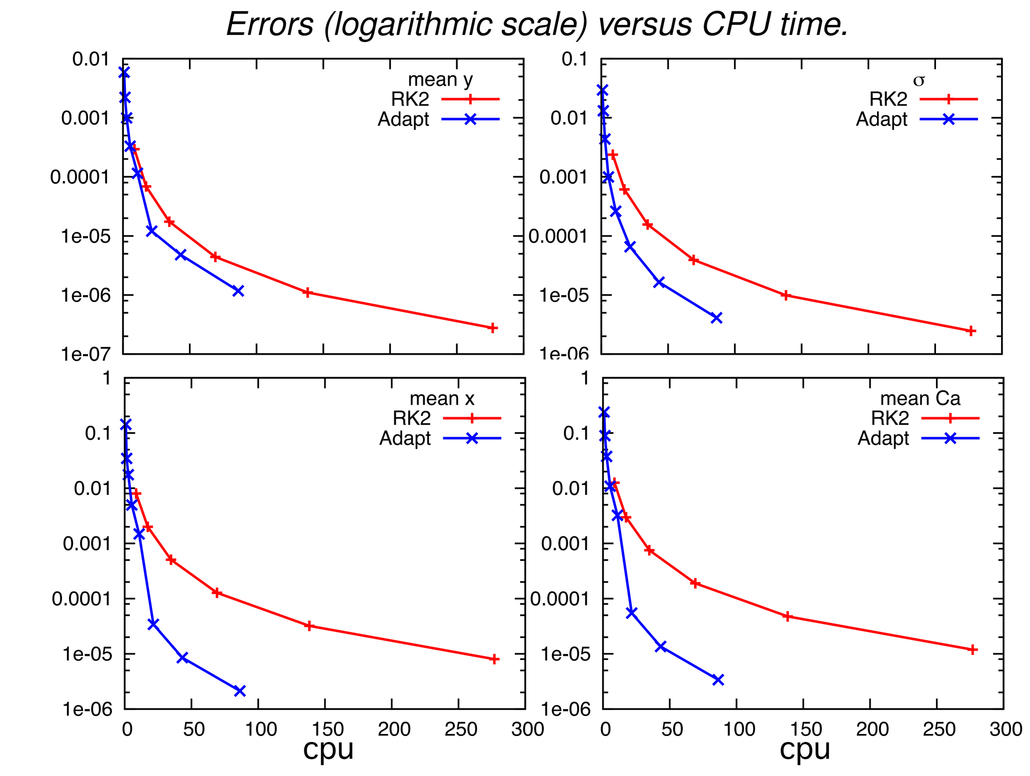
> **Data**: $\Delta t, \quad p = \lceil \frac{1}{\sqrt{\varepsilon}} \rceil, \quad X^n, \quad \sigma^n$
>
> Evaluation of $f_j(X^n, \sigma^n, \varepsilon)$ and $h(X^n, \sigma^n, \varepsilon)$
>
> Identify if σ is in slow or fast motion
>
> Distinguish cells in slow and fast motion (X_s^n and X_f^n)
>
> **if** *σ is slow* **then**
>
> Update σ and slow cells: $1 \times$ RK4 with $\frac{\Delta t}{2}$
>
> Update fast cells: $p \times$ RK2 with $\frac{\Delta t}{p}$ (slow cells fixed)
>
> Update σ and slow cells: $1 \times$ RK4 with $\frac{\Delta t}{2}$
>
> **else**
>
> Update σ and fast cells: $p \times$ RK2 with $\frac{\Delta t}{2p}$ (slow cells fixed)
>
> Update slow cells: $1 \times$ RK4 with Δt
>
> Update σ and fast cells: $p \times$ RK2 with $\frac{\Delta t}{2p}$ (slow cells fixed)
>
> **end**
>
> **Result**: $X^{n+1} = (X_f^{n+1}, X_s^{n+1}), \quad \sigma^{n+1}$

Consistency and performance Model (2), $N = 20$, $\varepsilon = 0.01$.

Errors versus coarse time step Δt (logarithmic scales).



Errors (logarithmic scale) versus CPU time.



References

[1] E.I. Terasawa *et al.*. Intracellular Ca2+ oscillations in luteinizing hormone-releasing hormone neurons derived from the embryonic olfactory placode of the rhesus monkey. *J Neurosci*, 19:5898–5909, 1999.

[2] M. Krupa, A. Vidal and F. Clément. A network model of the periodic synchronization process in the dynamics of calcium concentration in GnRH neurons. *J Math Neurosci*, 3:4, 2013.

[3] J. Laskar and P. Robutel. High order symplectic integrators for pertubed hamiltonian systems. *Celest Mech Dyn Astr*, 80:39–62, 2001.