NEURONAL NETWORKS IN BMS MODEL

JUAN CARLOS VÁSQUEZ B.

MASTER RESEARCH MEMORIES

Encadrant
BRUNO CESSAC
PhD et HDR

UNIVERSITE DE LA MEDITERRANEE
FACULTE DE SCIENCES DE LUMINY
CENTRE DE PHYSIQUE THEORIQUE
MARSEILLE
2007

CONTENTS

abstract

We have studied and simulated the dynamics of spiking neuronal networks on the specific framework of the model called BMS-model. We have choose particular types of the synpases and structure of network, i.e., the so called BMS -Laplacian NN on a $\mathbb{T}^2$ tori. We took special attention to understand the behavior of dynamical characteristics in presence of constant external input. We have found a great influence of stimuli in this type of model and network. Moreover we have characterised the creation of an more sensible region in the parameter space when an input is applied that we have called High- Input region. This region is richer in dynamical modes and the characteristic values of system may variate strongly there. We propose some specific theoretical results we have found and understand during the work and finally we comment some future lines of research in the subject.

CHAPTER 1

MODELING WITH NEURONAL NETWORKS

1.1 Introduction

Understanding the functioning of central nervous systems is a complex task that at the present time combines different scientific disciplines, from pure to applied sciences, most of times with different questions and motivations. However, standard lines of thought trying to resolve this problem can be elucidated ([1]), the basis consisting in:

- Finding out its general structure and its hierarchical organization until it most elementary component.

- Understanding what this units are made of and what they are made for, or more specifically how their presence or their absence manifests in the whole system. Usually this is done by analysing the effects and response of the system caused by defined stimulations.

Neuronal Networks (NN) models belong to a class of models where the units have the principal role and each unit is a network of more elemental components i.e. neurons.

Before going further we would like to note briefly the assumptions and biological details that define the domain of validity of this approach . First, although the biochemical process regulating the neurons are essential for the correct function of the system, we assume them not to be -in first approximation- responsables of the more complex neuronal process. Second, (following [1]), since the high number of neurons present in networks it is plausible that the genome do not controls so lower hierarchical levels like biochemical process in the neuron. Moreover, since so many neurons die at every moment, its individual role should be negligible. This leaded Cragg and Temperly to the idea that properties of the

next level, i.e., neuronal networks, comes from a collective behaviour [1]. However, biological evidence shows that a higher level, the role of this networks becomes weaker and genetics role increases. As a result of this arguments, taken as a smalls parts of larger and higher hierarchical systems conforming the central nervous system, the NN are a valid basic unit of analysis and also a compulsory description level since the formulation of a theory of larger systems requires at least at the beginning the study of its basic units isolated. Nevertheless, it remains yet finding out the nature of the turning point between the regime of simple networks and the organized higher structures whose conformation is controlled by the genome of the specific animal specie.Finally, we should remark that at this level a NN model based on interactions between neurones, is neglecting effects of secondary interactions between neurones (as graded potentials) and is faced with recent evidence that suggest other components as dendrites or glia cells to be directed involved in high level process.

Anyway, behind the comprehension of a large variety of the cerebral process like learning, memory, recognition or even more complex task combining different levels as vision rest a vast domain of applications, including neuronal machines called neurocomputers. Beyond this domain, one could expect that new methods and extensions of the existent ones for NN can be applied also to the study of other scientific problems in many fields of Knowledge which can be described as composed systems with non trivial dynamics as for example the Regulatory Genetic Networks and the Social/Economic Networks [2].

1.2 Brief Survey of Neurobiology

After recognition of the nervous networks as responsible of transport and process of the information in most forms of animal life, it was Galvani in 1791 who discovered the electric nature of nervous signals. But only until the early 20_{th} century the anatomist Ramon y Cajal showed that they were made of an assembly of well defined cells called neurons, which communicate via localised electrical or chemical process called synapses[3].See figure.

In the middle of the century, Hodgkin and Huxley explained the mechanism of creation and propagation of neuronal electric signals. The lipidic cell membrane of many types of cells contains voltage-gated ion channels which allow the cell to generate and propagate an electrical impulse called action potential, as a result of the manipulation of charge imbalances

[1] Usually composed system exhibits collective emergent behaviours not present when observing individual ones. More discussion about this point will be presented in third section about modeling and physics.

[2] In fact, that is the case and in both senses since theoretical analysis of other dynamical systems have been applied to the study of NN. However, is pertinent to note that usually the characteristics and assumptions of each domain and or model produce differences that make the extension to other domains or the applications of known results a non-trivial task.

[3] Given the diversity of functions performed by neurons naturally there exists a wide variety in shape, size and electrochemical properties in complex nervous systems as mammals

between the interior and the exterior environment, [4] but before a detailed explanation of it we describe an overall view of the neuronal transmission process.

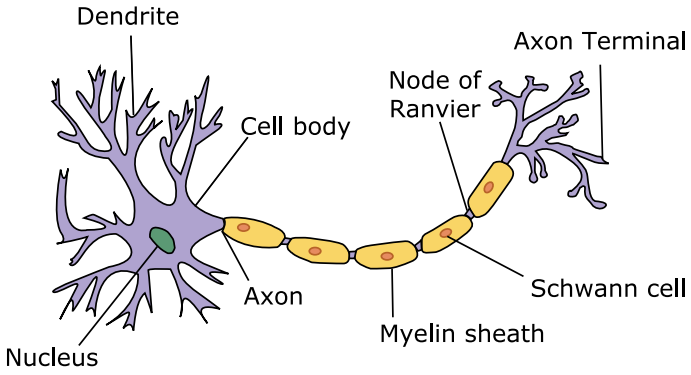


Figure 1.1: General Diagram of a Neuron

Once the action potential also known as spike is created and it is transported by the soma and the axon membrane until it reaches the axon terminal where it opens voltage-gated calcium channels, allowing calcium ions to enter the terminal. Calcium causes synaptic vesicles filled with neurotransmitter molecules to fuse with the membrane, releasing their contents into the synaptic cleft. The neurotransmitters diffuse across the synaptic cleft and activate receptors on the postsynaptic neuron, i.e., its dendrites.See figure. The synapses can be excitatory or inhibitory and will increase or decrease activity in the target neuron.This process is one way since axon do not posses chemoreceptor nor dendrites can secrete neurotransmitters. However, as there are neurones from 1 or 2 to over 1000 dendritic branches and similarly dendrites receiving thousands of synapses, all type of network is theoretically possible.

We describe in general terms the action potential process. Briefly, when an stimuli open some sodium-ion channels it causes a depolarization of the potential difference produced by the normal concentration differences of potasium an sodium ions inside and outside the cell. If this initial depolarization overpass a threshold value, more sodium channels get opened causing a bigger depolarization that reaches peak while the sodium channels get inactivated and potasium channels get opened to redistribute the ion flux repolarizating the cell and returning to the initial resting potential. After this process that takes about 100 ms, the sodium-channels rest inactivated and then less sensitive to stimuli for about 5ms producing what is known as the refractory period where the generation of another spike is very much difficult and then less probable. The next figure contains the schematic action potential and the graph of an actual recording of one.

[4]This action potential may be quite different between types of cells (see Wikipedia) so we will refer always to Neuronal Action Potentials

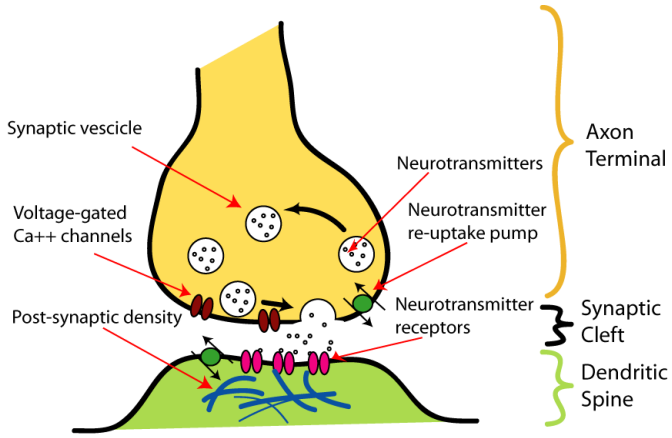


Figure 1.2: General Diagram of a synaptic junction

1.3 Neuronal Networks, Mathemathical Modeling and Physics

The most basic model of a NN based on neurons as logical gates, i.e. with only two possible internal states was devised by Pitts and McCulloch in 1943. A neuron has a few entries provided by the outputs of other neurons. These inputs are summed up and then determine the state of the neuron by comparison respect to a certain threshold value: If signal is larger than the threshold the neurone is active; otherwise it is inactive.After the Hodgkin-Huxley model (for details [2]), a more accurate description of each neuron was possible, specifically the action potential nature, including as most important improvements the explicit inclusion of time and explicit time evolving value of Voltage of membrane Potential and Inputs to the neuron.

If we imagine the neuron description level as an axe going from binary state to the Hodgkin-Huxley model, there is on the other side another axe we could call neuron population where the lowest level is a single isolated neuron, then a few neurones, then including weakly coupled populations and finally a big population. Moreover, considering differents models for the synapse, i.e. the types of interaction, one have a third axe. Obviously, for each ensemble of coordinates in the description axe coordinate system may exist languages and methods more appropriated than others going from dynamical systems and bifurcation theory to Ergodic theory, probability theory and statistical mechanics, including graph theory and others.We remark also that changing the neuron population may produce collective behaviour effects emerging from non-linearity of equations or simply by large number effects that hide some individual properties but enhance other ones.

As another important issue, if we take in account that biological systems are noisy it seems natural to include a new category of models including noise. However, it is the usual

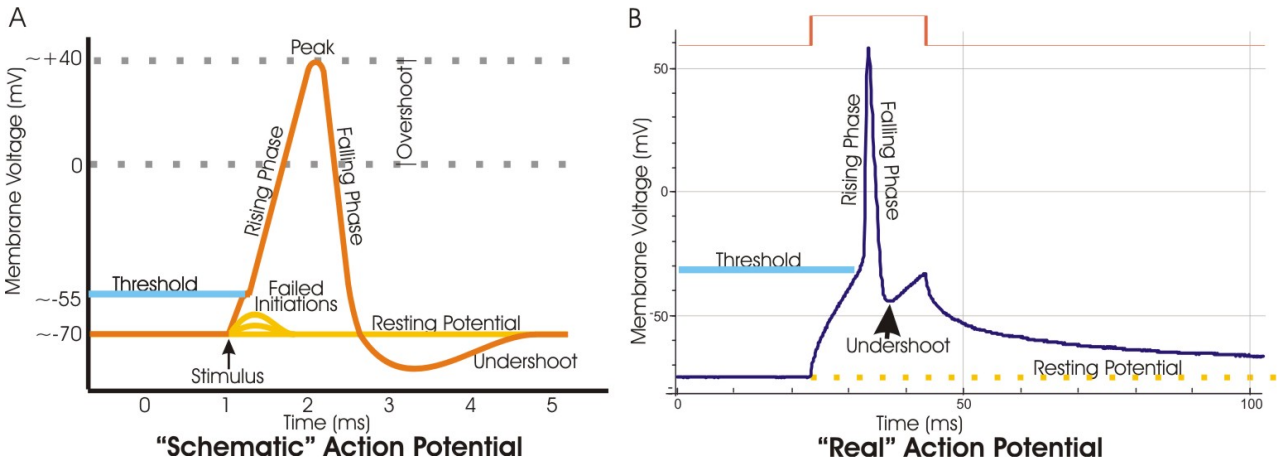


Figure 1.3: General Diagram of action potential

behaviour of noisy systems that average properties of the collective system are not affected if the noise intensity does not exceed certain level. By this way it is reasonable and justified at least in a approximated level the study of networks without including the noisy nature of its elements.

Finally, an obligated question in every modeling, more particularly in biological sciences is : which time model is more appropriated, a continuous one or a discrete one? And if we chose a discrete model, what is the correct characteristic time of the model. Obviously, the answer depends of the specific subject and its experimental evidence. But even restricting ourselves to the NN Modeling the question is very wide. We will comment in the next chapter some ideas about this issue for the specific case of the BMS-NN following [3].

1.4 Neuronal Models

1.4.1 Introduction

As we have described above, a neuron will usually fire an action potential or spike when its membrane potential reaches a certain threshold value. The mechanism well understood and modeled give raise to a set of differential equations, known as the Hodgkin-Huxley equations, which involve the different neuron parameters and dynamic variables. However, since it is a hard task, working with this equations must be justified by specific interest as response of parameter of neuron to a medication etc, and on the other hand, Neuronal Models not including explicitly biophysical mechanisms responsible of the spike event are more treatable and not far from the experimental evidence. Moreover, scientific NN community agrees on the fact that most part of information transmitted in a NN is contained in the spike

sequences[5].

A convenient an simple assembly of models producing spikes is are called Integrate-Fire models (IF), which define the spike event when the membrane potential reach the threshold and then the potential is reset to a value below the threshold potential. In that way, this models are left with the simpler task of modeling dynamics of membrane potential below the threshold which can be done at different complexity levels but anyway avoiding an exact description of the action potential

1.4.2 A brief introduction to Integrate-and-Fire Models

The basic leaky IF model -known also as passive IF- was proposed by Lapicque in 1907. In it, active membranes conductance are ignored, so the entire membrane conductance and synaptic inputs are modeled by leakage term which let us with [6] an equivalently electrical RC circuit equation with the resistor and the capacitor in parallel connection. That give us in the standard form:

$$\tau_m \frac{du}{dt} = E_L - u(t) + R_m I(t) \tag{1.1}$$

where E_L is the resting potential of the cell, taken usually null, $u(t)$ refers the membrane potential, R_m is the assumed total membrane resistance, $I(t)$ is the total current (i.e, I_R associated by the Ohm law to the resistance and potential value, and the I_C associated to the capacitor, so $I_C = C\frac{du}{dt}$, and finally $\tau_m \equiv R_m C$, with the obviously condition that when $u(t)$ reaches the threshold it resets to a certain value below threshold potential.

There are some remarkable characteristics. First, this model is exactly solvable. Second, after the time t_f of reaching the threshold, the membrane potential rest in the reset value while the spike is produced and the refractory period completed. Then, one could consider spikes as formal events characterized by the 'firing time' t_f [7] which indeed relate the model to an alternative approach to the study of neuronal dynamics based on the 'firing rates' and that is actually the natural way the experimental data comes out([5],[6]).

Moreover, it is easy to model a network of IF neurons. Consider the neuron i receives the spikes input coming from other neurons and the total current $I_i(t)$ is the sum of spikes coming from each neuron j connected to i. Modeling the synapses, as commented earlier, is not a trivial issue and may change properties and even the mathematical tools to be used. Nevertheless, as it seems a flexible way to create different types networks and the easiest one

[5] This change of description is quite far to be trivial,cause information could be encoded also by delay timing and other properties, but going further is out of the scope of this memories. For a detailed revision, cfr [3] and references therein.

[6] For details cfr. [5]

[7] this propriety relies on the fact that the dynamics is deterministic

also, a quite usual way is to use a J_{ij} weight, roughly modeling each synaptic connexion

$$I_i(t) = \sum_j J_{ij} \sum_{n(j)=1}^{N(j)} \alpha(t - t_n(j)); \tag{1.2}$$

where $t_n(j)$ is the n-th time of firing of neuron j, $\alpha(t)$ is a function modeling the spike[8] and the $\sum_n^N$ a small time integration.

Finally, there exists more general versions of the IF models [2] an others

$$\tau_m \frac{du}{dt} = F(u(t)) + G(u(t))I(t) \tag{1.3}$$

where F, G are non linear functions of u.

1.5 BMS NN Model: Context and Theoretical Results

The Beslon-Mazet-Soula(BMS) NN model, is a time discrete (Euler scheme) IF model, but in spite of the delicate questions concerning biological interpretations raised by the simplifications, especially time discretisation, it reveals some features (see theorems below) which may extend largely the scope of methods from dynamical systems theory, ergodic theory and statistical physics to look for answer important questions as:

- Which is the dynamics of a particular network and its relation with a transport-diffusion scheme?
- What is the relation between an stimulus (input) on the spiking sequences displayed by the NN?
- How to measure the information of a spiking sequence?
- What is the effect of stochastic perturbations (noise) in the BMS model? Is it similar to the effect of considering Brownian Noise in the continuous time IF NN models?

In this work we are concerned especially with the first two questions.
An Euler scheme formal time discretisation of (1.1) yields

$$V(t + dt) = V(t)\left(1 - \frac{dt}{\tau}\right) + \frac{I(t)}{C}dt \tag{1.4}$$

[8] see footnote 1.4.1

Then, setting $dt = 1$, which means choosing the time step, i.e, the model sampling time scale dt smaller than all characteristic times scales in the model[9] and defining $\gamma \equiv 1 - \frac{1}{\tau}$, we obtain

$$V(t+1) = \gamma V(t) + \frac{I(t)}{C} \tag{1.5}$$

This discretisation imposes to the the circuit characteristic time $\tau \equiv RC$ that $\tau \geq 1$ thus $\gamma \in [0, 1]$. Also we shall rescale units to make $C = 1$.

As in the continuous time IF model, a positive real number θ denotes the 'firing threshold value and this equation holds for $V(t) < \theta$. The firing of neuron is described now by

$$\begin{aligned} V(t^{(k)}) &\geq \theta \\ V(t^{(k)}+1) &= V_{reset} + I(t^{(k)}) \end{aligned} \tag{1.6}$$

where, from now, we shall consider the reset potential $V_{reset} = 0$. At this point we note that the firing is not instantaneous, since the membrane potential is over the threshold during the time interval $\left[t^{(k)}, t^{(k)}+1\right[$.We shall conveniently introduce the function $Z[x] = \chi(x \geq \theta)$ where χ is the indicatrix function, i.e, $Z[x] = 1$ whenever $x \geq \theta$ and $Z[x] = 0$ otherwise which allows to write the neuron evolution before and after the spike event (1.5,1.5) in a unique equation

$$V(t+1) = \gamma V(t)(1 - Z[V(t)]) + I(t) \tag{1.7}$$

1.5.1 Model Definition

The dimension of the NN is defined as the number N of neurons that compose it. The synaptic connexion from neuron j et i (i.e axon terminals of j et dendrites of i) is modeled by the synaptic weight W_{ij} so it is called 'excitatory' or 'inhibitory' whereas $W_{ij} > 0$ or $W_{ij} < 0$ respectively and it is nul if neuron j do not has action over neuron i. As consequence, the W_{ij} compose the named 'weight $(N \times N)$ matrix' W which defines an oriented and signed graph known as the 'NN associated to W' with vertex $i = 1 \ldots N$ representing the neurons an oriented signed edge $j \longrightarrow (\pm)i$ whenever W_{ij} is not nul. Then, is W who contents the Network structure information.

Now, each vertex (neuron) i is characterized by a real variable V_i called 'the membrane potential of i' which satisfies 1.7 for the same threshold value and evolve synchronously. The function $Z[V_i(t)]$ is called the 'firing state of neuron i at time t so when its value is 1 one say that neuron i 'fires' otherwise i is quiescent. Each 'total current' term $I_i(t)$ is composed by the 'external current[10] applied to the neuron i' denoted as I_i^{ext} and the quantity called 'Synaptic current'

$$I_i^s(\mathbf{V}(t)) = \sum_{j=1}^{N} W_{ij} Z[V_j(t)] \tag{1.8}$$

[9] However, this point requires a more extensive discussion , cfr [3].

[10] Rigorously, a potential since we divide it by a capacity C set equal to 1.

where by notation $\mathbf{V}(t) = \{V_i(t)\}_{i=1}^N$ is the vector of membrane potentials. Finally, putting all together, the synchronous dynamics of the BMS NN model is given by:

$$\mathbf{V}(t+1) = \mathbf{F}(\mathbf{V}(t)), \tag{1.9}$$

$$F_i(\mathbf{V}(t)) = \gamma V_i(t)(1 - Z[V_i(t)]) + \sum_{j=1}^{N} W_{ij} Z[V_j(t)] + I_i^{ext}(t); \quad i = 1 \dots N \tag{1.10}$$

We will restrict ourselves to the autonomous BMS model, i.e, with no explicit time dependency, thus, I_i^{ext} must be is constant, however the extension of some results to non-autonomous cases has been discussed in [3].

1.5.2 Preliminary Results (from Cessac [3])

Lemma 1. *Since $\gamma < 1$ on can restrict the configuration space to a compact set $\mathbb{M} = [V_{min}, V_{max}]^N$ such that $\mathbf{F}(\mathbb{M}) \subset \mathbb{M}$ where*

$$V_{min} = \min\left(0, \frac{1}{1-\gamma}\Big[\min_{i=1,\dots N} \sum_{j|W_{ij}<0} W_{ij} + I_i^{ext}\Big]\right) \tag{1.11}$$

$$V_{max} = \max\left(0, \frac{1}{1-\gamma}\Big[\max_{i=1,\dots N} \sum_{j|W_{ij}>0} W_{ij} + I_i^{ext}\Big]\right) \tag{1.12}$$

$$\tag{1.13}$$

Definition 1. Dividing the interval $[V_{min}, V_{max}]$ in $B_0 = [V_{min}, \theta)$ and $B_1 = [\theta, V_{max}]]$ creates the so called ' natural partition' of the configuration space defined as follows. Call $\Lambda = \{0,1\}^N$ and let $\eta = \eta_1 \dots \eta_N \in \Lambda$ that is a N-dimensional vector of binary components called the spiking state. Then $\mathbb{M} = \bigcup_{\eta\in\Lambda}$ where

$$\mathbb{M}\eta = \{\mathbf{V} \in \mathbb{M} | V_i \in B_{\eta_i}\} \tag{1.14}$$

We note that this partition has a simple Cartesian product structure which each domain is an hypercube with edges parallels to the coordinates directions. Nevertheless, we remark that this is not a Markov partition of the configuration space for the map $\mathbf{F}$.

Proposition 1. *Denote $\mathbf{F}\eta$ the restriction of the evolution map $\mathbf{F}$ to the domain $\mathbb{M}\eta$. Then whatever $\eta \in \Lambda$*

i $\mathbf{F}\eta$ *is affine and differentiable in the interior of its domain $\mathbb{M}\eta$.*

ii $\mathbf{F}\eta$ *is a contraction with coefficient $\gamma(1-\eta_i)$ in the direction i.*

iii Denote $\mathbf{F}\eta_i$ *the i-th component of* $\mathbf{F}\eta$*. then*

$$\mathbf{F}(\mathbb{M}\eta) = \prod_{i=1}^{N} \mathbf{F}\eta_i(B_{\eta_i}) \tag{1.15}$$

Definition 2. The singularity set for the map **F**is define as

$$S = \{\mathbf{V} \in \mathbb{M} | \exists i, V_i = \theta\} \tag{1.16}$$

This set is a finite union of $N-1$ hyperplanes corresponding to faces of the hypercubes. We say that S is small because in metric sense it has null Lebesgue measure and in topological sense it is non residual. Since **F** is discontinuous in S it has an important effect in the dynamics. Indeed, any taking an open ball set centered in a vector **V** in the configuration space such that the ball do not intersects S will be contracted every iteration and asymptotically will be indistinguishable of the trajectory of the vector **V**.On the other hand if it intersects the singularity for any two vectors arbitrarily close but with with different associated η (in one component), the distance between their iteration rest finite. This effect is known as weak initial condition sensitive effect because it only happens when the singularity set is crossed.

1.5.3 Asymptotic Dynamics

Definition 3. The attracting set *A*is the largest invariant set such that there exists an open set $A \subset U$ such that

$$A = \bigcap_{t=0^{\infty}} \mathbf{F}^t(U) \tag{1.17}$$

Indeed, A may be empty but for most cases we expect the asymptotic dynamics be contained there. As noted in [3], when A is not empty the ω-limit set, the set of accumulation points of $\mathbf{F}^t(\mathbb{M})$ is the closure of A.

Definition 4. The distance between the forward trajectory of a initial vector $\mathbf{V}(0)$ and the set S is defined as $d(\mathbf{V}^+, S) = \inf_{t\geq 0} \min_{i=1,\dots N} |V_i(t) - \theta|$

In the same way we can define the distance of the ω-limit set to the singularity taking $dAS = d(\omega(\mathbb{M}), S) = \inf_{\mathbf{V}\in\omega(\mathbb{M})} d(\mathbf{V}^+, S)$. The next results are what make us to simulate the last quantity defined.

Proposition 2. *If* $d(\mathbf{V}^+, S) > \epsilon > 0$ *then* $\mathbf{V}(0)$ *has a local stable manifold of diameter* ϵ*.*

Theorem 1. *If* $dAS = d(\omega(\mathbb{M}), S) > \epsilon > 0$ *then:*

i *There exists a finite Markov partition encoding symbolically the dynamics on* $\omega(\mathbb{M})$*. This partition is constructed with a refinement of the natural partition, where the elements have the form* $\omega(\mathbb{M}) \cap \bigcap_{t=0}^{T} \mathbf{F}^{-t}[\mathbb{M}\eta_t]$ *and have diameter* $\leq \epsilon$*.*

ii *The parameter T, in the previous item depends on* $d(\omega(\mathbb{M}), S)$ *and* $T \to \infty$ *when* $d(\omega(\mathbb{M}), S) \to \infty$*.*

iii $\omega(\mathbb{M}), S)$ *is a finite union of stable periodic orbits with finite period.*

With this Markov partition we are able to construct a transition graph, with a transition matrix associated, but we remark this is not the transition graph we construct in the application section since we are concerned first with the natural partition an how it may contains different type of orbits and then the simulation of dAS to find the regions where the complexity increases as $dAS \to 0$.

Finally we would like to note as is done in [3] that the quantity $T_d = 2^{N\frac{\log(dAS)}{\log(\gamma)}}$ is an upper bound of the nnumber of Markov partition elements and hence of the maximal period of orbits.

CHAPTER 2

APPLICATION: THE BMS LAPLACIAN NN MODEL IN A T^2 CHAIN

2.1 Context and Motivations

The BMS LAPLACIAN, is defined is an specific BMS model where each neuron has symmetric excitatory synapses for each of its neighbors while also it present an auto-inhibitory spiking activity twice strong i.e

$$W_{i\pm1,i} = \alpha, \qquad W_{ii} = -2\alpha, \qquad \text{and} W_{ij} = 0 \quad \text{other synapses} \quad \forall i = 1, \ldots, N$$

where we use the convention for neuron N that $N + 1 \equiv 1$ and for neuron 1 that $0 \equiv N$ to establish a closed chain configured as a $\mathbb{T}^2$ Tori. The selection of this type of model has some specific motivations we exposed below and its similarity with the discrete definition of a Laplacian operator on a 1-D space is not casual. Anyway, it is clear that having a 1-D type structure with just few connections an almost all of same type and intensity make it a suitable easier one to start an specific study of BMS models but capable of contain non trivial dynamics.

The first reason two choose this type of BMS model is in order to study (in future work) based in the work of Vieville and Kornprobst ([4])its relation with the variational approach to modeling cortical maps with feedbacks. Basically, a cortical map models the central system process as computations of 'maps' of quantitative values which according to generative approaches can be modeled as an optimization problem relating the input proposed with the output achieved by the system ([4]). Then, the cited authors have implemented this optimization problem using a discrete network which results in a system of coupled linearized differential equations. Finally they state that it appears deterministic spiking neuronal networks can be linked to the specifications proposed by this representation of the optimisation problem for cortical maps. Since the cortical map modeling finally relates input-output we

look for response about if an spiking NN model achieve similar results and what conditions relates the input and output response in such type of spiking models as the BMS.

Secondly, as the transport phenomena or diffusion in usual physical systems can be related with a variational principle, we would like to know also if a transport-type BMS model may hide too a variational principle controlling its general evolution. Moreover, a future work may include the study of statistical distribution of orbits and related issues concerning a variational principle formulation.

2.2 The N=3 Case

For the explicit analysis of this type of BMS-Model we shall bound optimally the configuration space as it simplifies considerably the task of identifying correctly all the possible different dynamical regimes. However, as it would appear, when an external input is present it reveals the interdependencies between the bounds (a type of broken symmetry effect) that might become hard to manipulate manually for general input configurations and also for simple input configurations but acting on NN (even smaller ones) with more than 2 active -i.e. non null- synaptic connexions per neuron. As a previous remark-result our way of proceeding is not really applicable in order to optimize a reliable simulation. The configuration space will be denoted generally as $\mathbb{M} = \bigtimes_{i=1}^{3}[V_{min,i}, V_{max,i}]$. We shall also write $Z_i \equiv Z[V_i]$, the 2^3 spiking states (vertex) labeled $j = 4Z_1 + 2Z_2 + Z_1$ and its corresponding sets in $\mathbb{M}$ denoted as η_j.

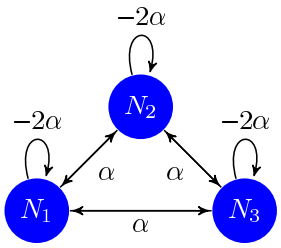


Figure 2.1: General Diagram of the N=3 BMS Laplacian Tori

2.2.1 Non External Input

Here we have for $i = 1, 2, 3$:

$$V_{min,i} = -2\alpha \qquad V_{max,i} = \gamma\theta + 2\alpha \text{ with } V_{max,i} \geq \theta \tag{2.1}$$

In this case (no input) V_{min}, V^{max} do not depends of index i so we may drop it. We define then

$$C_1 = \gamma V_{min} + \alpha \qquad C_2 = \gamma\theta + \alpha \tag{2.2}$$
$$C_3 = \gamma V_{min} + 2\alpha \qquad C_4 = \gamma\theta + 2\alpha \tag{2.3}$$

The case $V_{max} < \theta$ is irrelevant so we focus on $V_{max} \geq \theta$. Then, the mapping of $\mathbb{M}$ by one iteration of F is totally described as:

$$\eta_0 \equiv [V_{min}, \theta[\times [V_{min}, \theta[\times [V_{min}, \theta[\qquad F(\eta_0) \equiv [\gamma V_{min}, \gamma\theta[\times [\gamma V_{min}, \gamma\theta[\times [\gamma V_{min}, \gamma\theta[\subset \eta_0 \tag{2.4}$$
$$\eta_1 \equiv [V_{min}, \theta[\times [V_{min}, \theta[\times [\theta, V_{max}] \qquad F(\eta_1) \equiv [C_1, C_2[\times [C_1, C_2[\times \{-2\alpha\} \tag{2.5}$$
$$\eta_2 \equiv [V_{min}, \theta[\times [\theta, V_{max}] \times [V_{min}, \theta] \qquad F(\eta_2) \equiv [C_1, C_2[\times \{-2\alpha\} \times [C_1, C_2[\tag{2.6}$$
$$\eta_3 \equiv [V_{min}, \theta[\times [\theta, V_{max}] \times [\theta, V_{max}] \qquad F(\eta_3) \equiv [C_3, C_4[\times \{-\alpha\} \times \{-\alpha\} \tag{2.7}$$
$$\eta_4 \equiv [\theta, V_{max}] \times [V_{min}, \theta[\times [V_{min}, \theta[\qquad F(\eta_4) \equiv \{-2\alpha\} \times [C_1, C_2[\times [C_1, C_2[\tag{2.8}$$
$$\eta_5 \equiv [\theta, V_{max}] \times [V_{min}, \theta[\times [\theta, V_{max}] \qquad F(\eta_5) \equiv \{-\alpha\} \times [C_3, C_4[\times \{-\alpha\} \tag{2.9}$$
$$\eta_6 \equiv [\theta, V_{max}] \times [\theta, V_{max}] \times [V_{min}, \theta[\qquad F(\eta_6) \equiv \{-\alpha\} \times \{-\alpha\} \times [C_3, C_4[\tag{2.10}$$
$$\eta_7 \equiv [\theta, V_{max}] \times [\theta, V_{max}] \times [\theta, V_{max}] \qquad F(\eta_7) \equiv \{0\} \times \{0\} \times \{0\} \subset \eta_0 \tag{2.11}$$
$$\tag{2.12}$$

The position of the all constants C_j respect to the threshold value (here $\theta = 1$) corresponds to a partition of the parameter space as shown in figure 2.2. Then, on can calculate the transition graph of the system for each region as shown in figure 2.3. We should remember, that our partition $\{\eta_i\}$is not a Markov partition for the system so this graphs will not generate the code for applying the symbolic dynamics methods.

Since the dynamics is contracting for every region no intersecting the singularity set S ,we deduce from graphs of figure 2.3 that only initial states never passing through η_0 can survive the neuronal death in the case of non external input. Moreover, although it is not evident from the graphs, no orbits of period greater than 3 can be generated in this case. This result is establish below as a two lemmas and as we suspect, this is the mechanism who prohibits the existence of orbits of period $P > N$ for $\mathbb{T}^2$-Networks without external input[1].

Lemma 2. *Let be $N = 3$ the size of BMS-Laplacian NN on a $\mathbb{T}^2$ Tori. Let be $\eta_a \to \eta_b \to \eta_c$ a valid sequence of spiking states for the system such that*

i Any spiking state η_i of the sequence is the 'zero spikes' state (noted usually η_0).

ii Between the initial spiking state and the final one, all the neurons have fired only once while its neighbors where quiescent.

[1]However its application is not straightforward before knowing the admitted transitions graphs for an specific N value. This fact makes us to wish try in the future to rewrite it in a more general way, to explain directly that result.

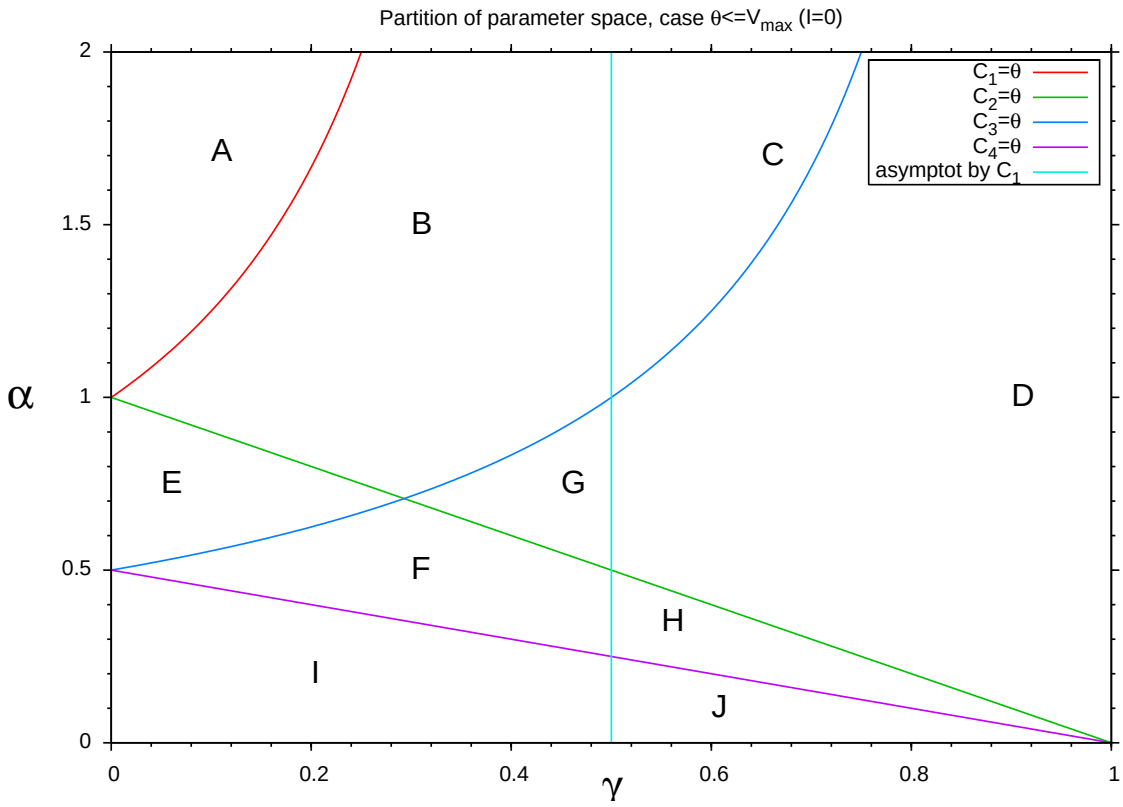


Figure 2.2: Dynamical different regions from equation.2.24 in the parameter space (γ, α) $(I = 0)$

Then, in any orbit having the spiking sequence, the transition backward at fourth iteration, i.e. $\eta_c \to \eta_b$ *is forbidden*

Proof. Let be $t = 0$ when the system is in state η_a,$\mathbf{V}(0) = (V_1(0), V_2(0), V_3(0),)$ the membrane potential vector, and i the neuron firing at that time. Then

$$V_i(1) = -2\alpha < \theta \qquad \text{while} \qquad V_{j,k}(1) = \gamma V_j(0) + \alpha, \quad j,k \neq i$$

Let be named k the index of the only neuron that fires at this iteration so

$$V_i(2) = -2\alpha\gamma + \alpha < \theta, \qquad V_k(2) = -2\alpha < \theta, \qquad V_j(2) = \gamma^2 V_j(0) + \gamma\alpha + \alpha \text{ with} j \neq i,k$$

Finally, at this iteration third neuron fires and we have

$$V_i(3) = -2\alpha\gamma^2 + \gamma\alpha + \alpha, \qquad V_k(3) = -2\alpha\gamma + \alpha, \qquad V_j(3) = -2\alpha$$

From the condition that k were the only neuron firing at second iteration we had $-2\alpha\gamma+\alpha < \theta$ so the system can not go backward to the spiking state η_b. □

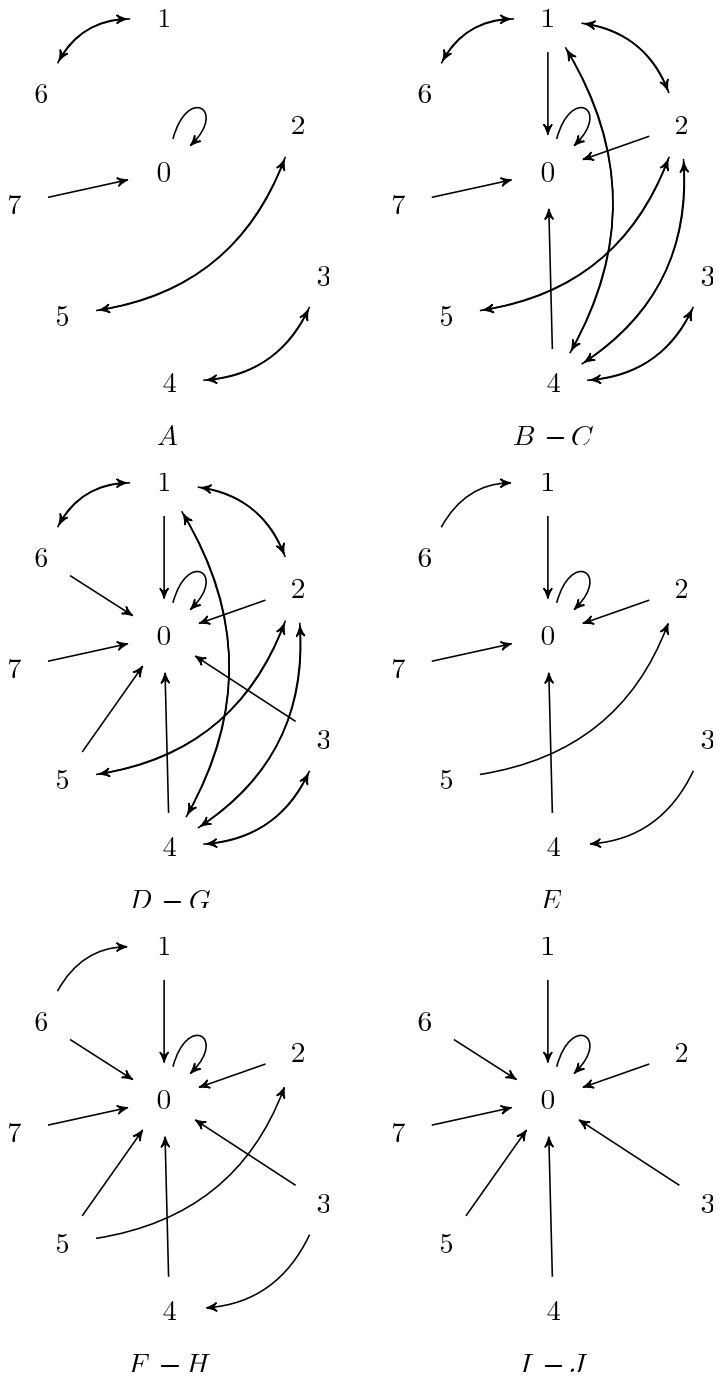


Figure 2.3: Transition Graphs N=3 BMS-Laplacian corresponding figure 2.2

Lemma 3. *Let be $N = 3$ the size of BMS-Laplacian NN on a $\mathbb{T}^2$ Tori. Let be $\eta_a \to \eta_b$ a valid transition of spiking states for the system such that the spiking state η_a has only one firing neuron and on the other state the other two neurons fires. Then, the next transition is $\eta_b \to \eta_0$ or $\eta_b \to \eta_a$ and if the latter the system is necessarily on this bi-periodic orbit for all the subsequent times.*

Proof. Let be $t = 0$ when the system is in state η_a,$\mathbf{V}(0) = (V_1(0), V_2(0), V_3(0),)$ the membrane potential vector, and i, j the neurons firing at that time, k the other one. Then

$$V_{i,j}(1) = -\alpha < \theta \qquad \text{while} \qquad V_k(1) = \gamma V_k(0) + 2\alpha \geq \theta$$

As K fires at this iteration, we have

$$V_{i,j}(2) = -\alpha\gamma + \alpha, \qquad V_k(2) = -2\alpha$$

The first case is $V_{i,j}(2) = -\alpha\gamma + \alpha < \theta$ arriving to the neuronal death. On the second one $V_{i,j}(2) = -\alpha\gamma + \alpha \geq \theta$ (which by the way means $\alpha(1-\gamma) \geq \theta$), and as $V_k(2) < \theta$ we are on the state η_b. Then, the next iteration will get us to

$$V_{i,j}(3) = -\alpha, \qquad V_k(3) = -2\alpha\gamma + 2\alpha$$

and since from the last iteration $\geq \alpha(1-\gamma) < 2\alpha(1-\gamma)$, the system is in the state η_a. □

Applying this two lemmas to create the periodic sequences following transition graphs on 2.3 we see that the maximal period is $P = 3$, e.g. one have that $\eta_1 \to \eta_2$ in graph2.3(G-D).It admits to go on η_5 or going to η_4. In the first case we arrive to the neuronal death or on a bi-periodic orbit. On the other hand, going to η_4 implies by 2 to go on a bi-periodic orbit with η_3 by 3or returning to η_1 so the 3-orbit is created. As a final important remark we note that a physical way to see this is thinking in the possible ways of periodic diffusion of a 'perturbation' (the spike) on an isolated $\mathbb{T}^2$ chain. On a 3-chain is quite transparent that a perturbation transported normally can take at most 3 steps to come back to its initial configuration. However on bigger chain, or only with our N-3 case lemmas, this fact becomes less evident.

2.2.2 Input case $\mathbf{I}^{ext} = (0, 0, I^{ext})$ with $I^{ext} < \theta$

The definitions of last section requires a modification. We will note $\underset{min,i}{V_{max,i}} \equiv \underset{min}{V_{max}}$ respectively for $i = 1, 2$ and $\underset{min,3}{V_{max,3}} \equiv \underset{min}{V^I_{max}}$. Then

$$V_{min} = -2\alpha; \qquad V_{max} = \begin{cases} \frac{\alpha}{1-\gamma} & \text{if } V_{max} < \theta \wedge V^I_{max} \geq \theta \\ \gamma\theta + \alpha & \text{if } V_{max} \geq \theta \wedge V^I_{max} \geq \theta \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

For $i = 3$ and restrained to the input intensity range, we have[2]

$$V^I_{min} = -2\alpha + I \tag{2.14}$$

$$V^I_{max} = \begin{cases} \frac{I}{1-\gamma} & \text{if } V_{max} < \theta \wedge V^I_{max} < \theta \\ \gamma\theta + I & \text{if } V_{max} < \theta \wedge V^I_{max} \geq \theta \\ \gamma\theta + 2\alpha + I & \text{if } V_{max} \geq \theta \wedge V^I_{max} \geq \theta \end{cases} \tag{2.15}$$

We note here, that the optimal bounds of configuration space for the neurons with input are related to those of neuron without input, since they depends of the dynamic of the network itself, i.e. for a single neuron if its neighbors can or not fire will change the limit values that its membrane potential can take but also if neuron can fire, it will modify the limit values of those of its neighbors. To resume, the input signal has created an asymmetry since the position of the bounds respect the threshold value is not anymore the same for all neurons.

The definitions of eq.2.2 remaining unchanged, we add now the following definitions

$$D_1 = \gamma V^I_{min} + I \qquad D_2 = \gamma\theta + I \tag{2.16}$$

$$D_3 = \gamma V^I_{min} + \alpha + I \qquad D_4 = \gamma\theta + \alpha + I \tag{2.17}$$

$$D_5 = \gamma V^I_{min} + 2\alpha + I \qquad D_6 = \gamma\theta + 2\alpha + I \tag{2.18}$$

Let's start the analysis with the case $V_{max} < \theta$. Here, neurons $i = 1, 2$ can not fire and the right inequality condition equivalent (see eq. 2.14) to $I > \theta(1 - \gamma)$ and when it is satisfied, the dynamics may admit the transition $\eta_0 \to \eta_1$. More explicitly the dynamical mapping is

$$\eta_0 \equiv [V_{min}, V_{max}] \times [V_{min}, V_{max}] \times [V^I_{min}, \theta[$$
$$F(\eta_0) \equiv [\gamma V_{min}, \gamma V_{max}] \times [\gamma V_{min}, \gamma V_{max}] \times [D_1, D_2[\tag{2.19}$$

$$\eta_1 \equiv [V_{min}, V_{max}] \times [V_{min}, V_{max}] \times [\theta V^I_{max}]$$
$$F(\eta_1) \equiv [\gamma V_{min}, \gamma V_{max}] \times [\gamma V_{min}, \gamma V_{max}] \times \{-2\alpha + I\} \subset \eta_0 \tag{2.20}$$

The parameter space as showed below is then divided in 3 regions depending of the relations

$$C \equiv D_2 < \theta \tag{2.21}$$

$$A \equiv D_2 \geq \theta \ \wedge \ D_1 < \theta \tag{2.22}$$

$$B \equiv D_2 \geq \theta \ \wedge \ D_1 \geq \theta \tag{2.23}$$

where all are considered for an specific value I. We must remark that in 2.22, as $D_1 < D_2$ then the relation $D_1 < \theta$ is implicitly contained while its other sense forbidden. The corresponding dynamical transition graphs are show in fig 2.5. However this interface evolves with the change of I value as shown in fig 2.6.

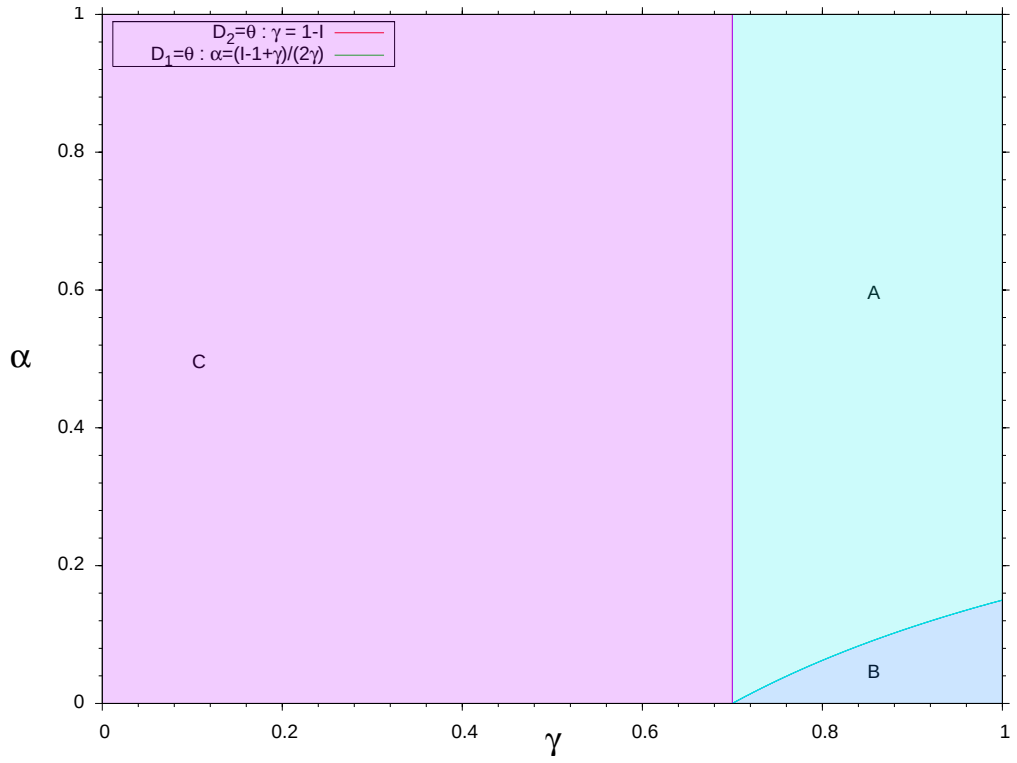


Figure 2.4: Different dynamical regions from eq.2.19-2.20 in the parameter space(γ, α) for the case $V_{max} < \theta \leq V^I_{max}$ with $I = 0.4$

As is easily inferred, the regions (A,B) in fig 2.4,2.6 do not exist if $I = 0$ since the transition $\eta_0 \rightarrow \eta_1$ is allowed only if $D_2 \geq \theta$ and that, only may happens at the right side of the straight $\gamma = 1 - I$. Thus initially the parameter region space for this case is controlled by a transition graph of type (C) where only the neuronal death is possible. We remark also that while region (B) is a biperiodic one, the region (A) is enough complex since it permits the neurons stay in η_0 a finite time before coming back to η_1 creating in that way orbits of all different periods adjusting the parameters γ, α. This statement is con sequence of a more general result about the firing time that will be formally exposed in next section.However, we would like to note that in the latter case, $V^I_{max} = D_2$, which is in fact the lowest value of V^I_{max} when still remaining at a higher value than threshold. Indeed, we have just put in evidence on a very specific case the fact that whenever $D_2 \geq \theta$ transitions from η_0 (i.e., the state where any neuron is firing) to another different spike state region are admitted. As consequence is this condition who 'measure' how 'strong' is actually the external current value. As consequence, we shall name 'high input region' and 'low input region' respectively the parameter regions defined by the condition $D_2 \geq \theta \Leftrightarrow I > \theta(1 - \gamma)$.

[2]We remark that the structure of the system manifest in the equations the impossibility of the case $V^I_{max} < \theta \leq V_{max}$, but more general couplings may permit this case.

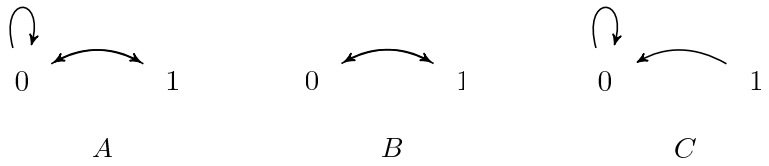


Figure 2.5: Transition Graphs from eq 2.19,for regions A,B,C corresponding figure2.4

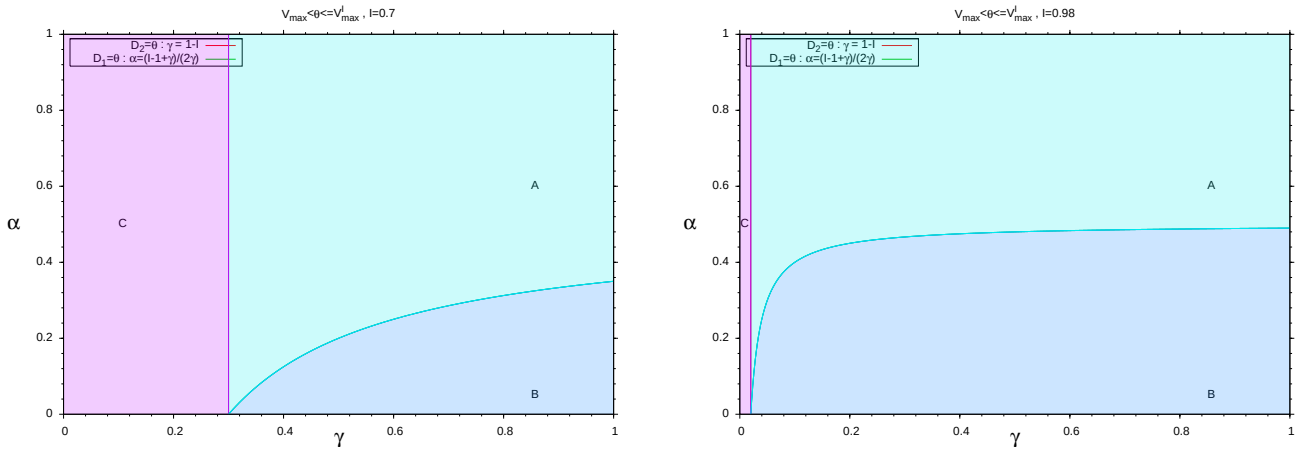


Figure 2.6: Parameter space different dynamical regions for case $V_{max} < \theta \leq V^I_{max}$ with $I = 0.7$ et 0.99

In next cases we shall try to label this 2 main regions by coloring it as follows

Now we shall examine the more complex case $\theta \leq V_{max} < V^I_{max}$. The mapping is described by the set of regions defined as follows:

$$\eta_0 \equiv [V_{min}, \theta[\times [V_{min}, \theta[\times [V^I_{min}, \theta[\qquad F(\eta_0) \equiv [\gamma V_{min}, \gamma\theta[\times [\gamma V_{min}, \gamma\theta[\times [D_1, D_2[\quad (2.24)$$

$$\eta_1 \equiv [V_{min}, \theta[\times [V_{min}, \theta[\times [\theta, V^I_{max}] \qquad F(\eta_1) \equiv [C_1, C_2[\times [C_1, C_2[\times \{-2\alpha + I\} \quad (2.25)$$

$$\eta_2 \equiv [V_{min}, \theta[\times [\theta, V_{max}] \times [V^I_{min}, \theta] \qquad F(\eta_2) \equiv [C_1, C_2[\times \{-2\alpha\} \times [D_3, D_4[\quad (2.26)$$

$$\eta_3 \equiv [V_{min}, \theta[\times [\theta, V_{max}] \times [\theta, V^I_{max}] \qquad F(\eta_3) \equiv [C_3, C_4[\times \{-\alpha\} \times \{-\alpha + I\} \quad (2.27)$$

$$\eta_4 \equiv [\theta, V_{max}] \times [V_{min}, \theta[\times [V^I_{min}, \theta[\qquad F(\eta_4) \equiv \{-2\alpha\} \times [C_1, C_2[\times [D_3, D_4[\quad (2.28)$$

$$\eta_5 \equiv [\theta, V_{max}] \times [V_{min}, \theta[\times [\theta, V^I_{max}] \qquad F(\eta_5) \equiv \{-\alpha\} \times [C_3, C_4[\times \{-\alpha + I\} \quad (2.29)$$

$$\eta_6 \equiv [\theta, V_{max}] \times [\theta, V_{max}] \times [V^I_{min}, \theta[\qquad F(\eta_6) \equiv \{-\alpha\} \times \{-\alpha\} \times [D_5, D_6[\quad (2.30)$$

$$\eta_7 \equiv [\theta, V_{max}] \times [\theta, V_{max}] \times [\theta, V^I_{max}] \qquad F(\eta_7) \equiv \{0\} \times \{0\} \times \{I\} \subset \eta_0 \quad (2.31)$$

The change of the mapping as function of the parameter γ, α, I can be seen looking at the behavior of the constants $\{C_j, D_k\}$ respect the threshold. The set of fig.2.30-3.2(at the end of document) try to resume it.

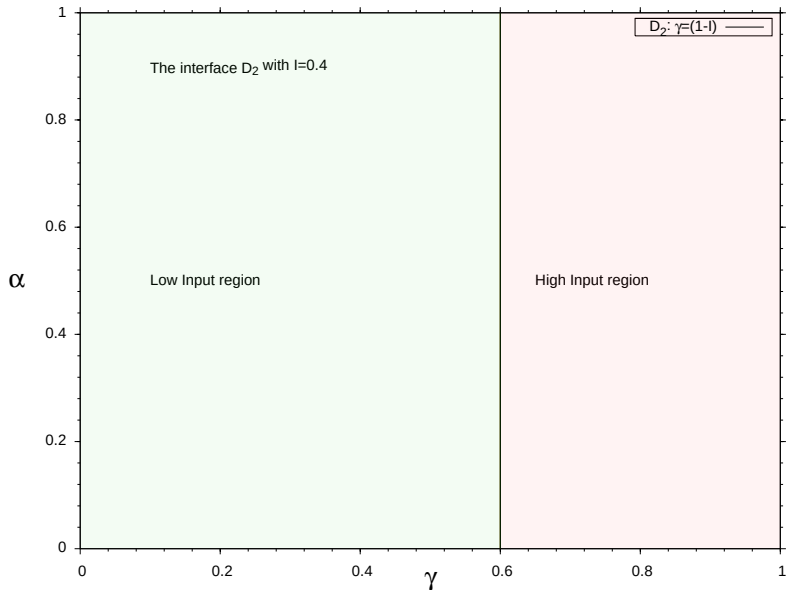


Figure 2.7: Parameter space characterized as Low and High Input Regions for I=0.4

Two main differences with the Fig2.3 must be remarked. First, the apparition of new configuration for higher values of $I = 0.5$ since the sign of some equations is inversed, specifically those of D_3, D_5. Second, in the case without input the asymptote of C_1 in $\gamma = 0.5$ had no effect on the system (i.e., each pair of regions $B - C$, $F - G$, $I - J$ composed indeed a single region with only one corresponding transition graph) it was marked anyway because in the input case, they are not trivially the same parameter region. Indeed after certain I value where for D_3 there are present 2 disconnected curves at every side of the asymptote, the inequality relations $D_3 < \theta$ and $D_3 > \theta$ have its positions (below or above) respect the curve $D_3 = \theta$ inversed at each side of the asymptote.

From the point of view of the transition graphs associated to each region in figures 2.30-3.2 respect those of figure 2.3 associated of figure 2.2, we can establish its differences in a general framework. The transition graphs in the case of input will present in general the same structure than those without input but transition graphs for regions in the high input region will present the edges of the state η_0 as one of the cases in figure 2.5, i.e., with an allowed transition from η_0 to the state η_1. The type of difference is that in the non input case, when the transition $\eta_2 \to \eta_1$ was allowed, its inverse transition and the transitions $\eta_2 \to \eta_4$, $\eta_1 \to \eta_4$ with their respective inverse, and $\eta_1 \to \eta_6$, $\eta_2 \to \eta_5$, $\eta_4 \to \eta_3$ were also allowed. Moreover, the transition from this states to the neuronal death $\eta_{4,2,1} \to \eta_0$ used to disappear together. In the input case, when transition $\eta_2 \to \eta_1$ is allowed the transition $\eta_4 \to \eta_1$ will be necessarily allowed too, while the others transitions mentioned may remain forbidden or be allowed as a complete set (i.e., all or any). Additionally the drop of the transitions $\eta_{4,2,1} \to \eta_0$ is not

neither simultaneous. The same manner, transitions $\eta_6 \to \eta_1$, $\eta_3 \to \eta_4$ and $\eta_5 \to \eta_2$ used to appear simultaneously in the graphs of figure 2.3, but now the first in appear is $\eta_6 \to \eta_1$ and the other two eventually in another region will be added to the graph at same time. Inversely, the first in disappear is $\eta_6 \to \eta_0$ and then $\eta_{5,3} \to \eta_0$ disappear of the graph. To resume, what is important here is the apparition of transitions from η_0 and the no apparition of new loops apart of the possible one on η_0, i.e., $\eta_0 \to \eta_0$.

As an example we have constructed the transition graph for the region Y, noted in figure2.30($I = 0.8$). From the dynamical point of view this is the region the most complex cause as it has the maximum of allowed transition between states and every state can to η_0 and then go out to begin another different route. As we showed, entering to η_0 with the loop active and also the ability of going out to η_1 may create orbits of all periods adjusting the parameters α, γ of the network.

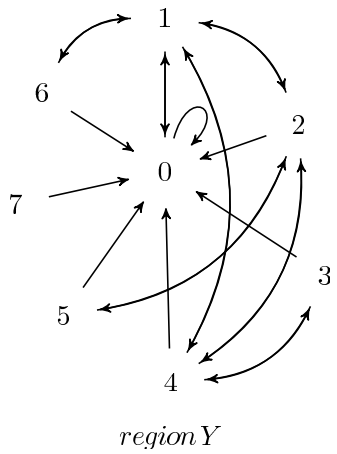


Figure 2.8: Transition Graphs from eq **??**,for regions Y corresponding figure2.30

2.3 Some other Theoretical Results

Definition 5. We will note int the following, the total current to neuron i at time t the quantity

$$I_i^t(\mathbf{Z}(t)) = \sum_{j=1}^{N} W_{ij}\, Z[V_j(t)] + I_i = -2\alpha Z[V_i(t)] + \alpha\big(Z[V_{i+1}(t)] + Z[V_{i-1}(t)]\big) + I_i$$
$$\equiv -2\alpha Z_i(t) + \alpha\big(Z_{i+1}(t) + Z_{i-1}(t)\big) + I_i \quad (2.32)$$

2.3.1 Transition Graphs and Firing Time

Lemma 4. *Let be N the size of a $\mathbb{T}^2$-BMS-Laplacian NN. Suppose there is a set of neurons(not necessarily consecutive) indexed by its position number i which is exposed to a constant external input signals I_i. Then, if any of the input signal I_k is such that $I_k > \theta(1-\gamma)$, there will exist at least one allowed transition 'zero spike' state noted η_0 and the a spiking state with its spike component k non null.*

Proof. Since in state η_0 no one fires, the interval $[V_{min,k}, \theta]$ is mapped on $[\gamma V_{min,k} + I_k, \gamma\theta + I_k]$, and from the hypothesis it follows that $\gamma\theta + I_k \geq \theta$. □

For identical consecutive inputs signals values satisfying the condition of the lemma, between neurons $i, i+k$. Moreover supposed that $\gamma V_{min,j} + I < \theta$ for all neurons in receiving the input. Then, will have a transition graph as in figure 2.9

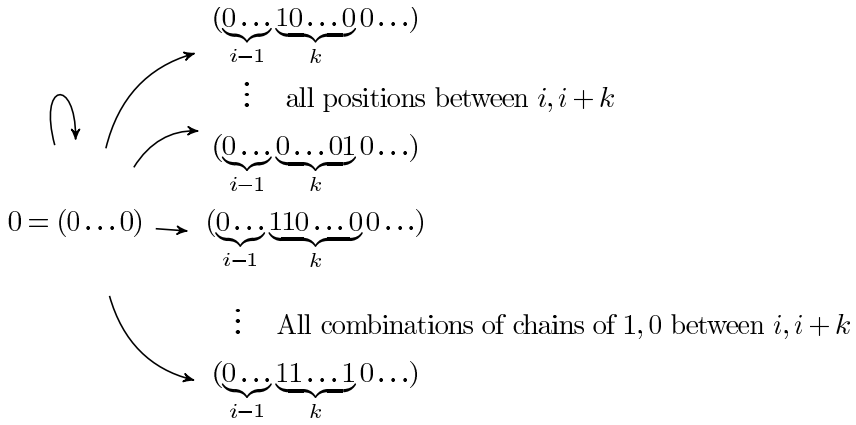


Figure 2.9: Transition Graphs exemplifying the application of Lemma 4 in a particular input case

Definition 6. The neuronal death $\mathbf{V}_D$is the defined as the stationary state with no spiking activity so it can be formally expressed by

$$\mathbf{F}_D^t(\mathbf{V}_D) = \mathbf{V}_D \quad \forall t \geq 1 \tag{2.33}$$

$$F_D(V_i) = \gamma V_i + I_i^{ext} \tag{2.34}$$

Where the second equation describes explicitly that no spike is produced during the evolution.

Lemma 5. *Let be N the size of a $\mathbb{T}^2$-BMS NN. Suppose there is a set of neurons(not necessarily consecutive) indexed by its position number i which is exposed to a constant external input signals I_i. As consequence of lemma4, if any of the input signals I_k is such that $I_k > \theta(1-\gamma)$*

the system can not get to the neuronal death. Indeed, the firing time of the the neuron k, noted τ^k has an upper bound given by

$$\tau^k = \frac{\theta(1-\gamma) - I_k}{(V_{min}(1-\gamma) - I_k)\log(\gamma)} \tag{2.35}$$

Proof. The longest time to fire is when the neuron do not receive any excitatory spike i.e, $I_k^s(t) = 0 \quad \forall t > 0$. Thus its membrane potential after a time t is done by $V_k(t) = \gamma^t V_k(0) + I\frac{1-\gamma^t}{1-\gamma}$. Taking the care of correct sign operations, then resolving for t with the condition $V_k(t) = \theta$ and maximizing it inside the configuration space, we get to the result. □

We note that the last lemma, is related with some results of Cessac, where the author showed the existence of ghost orbits for the condition $I = \theta(1-\gamma)$, i.e., the orbits taking infinite time to fire(**??**).

Lemma 6. *Let be N the size of a $\mathbb{T}^2$-BMS-**Laplacian NN**. Suppose there is a set of neurons(not necessarily consecutive) indexed by its position number i which is exposed to a constant external input signals I_i. If all of the input signals I_k are such that $I_k < \theta(1-\gamma)$ the system is able to get to the neuronal death. Moreover the membrane potential values of neurons for this state are*

$$V_{D,i} = \begin{cases} 0 & \text{if } I_i^{ext} = 0 \\ \frac{I_i^{ext}}{1-\gamma} & \text{otherwise} \end{cases} \tag{2.36}$$

Proof. For no spiking activity we have the membrane potential after a time t is done by $V_k(t) = \gamma^t V_k(0) + I\frac{1-\gamma^t}{1-\gamma}$. As the neuronal death is the limit of this evolution, all the configuration space will be contracted to the point 0 if $I = 0$ or to $\frac{I_i^{ext}}{1-\gamma}$ otherwise. □

Definition 7. We will note in the following, the accumulated synaptic input current to neuron i from time 0 to time t the quantity

$$J_i(t) = \alpha \sum_{n=1}^{t} \gamma^{t-n}(Z_{i+1}(k) + Z_{i-1}(k)) \tag{2.37}$$

Thus the membrane potential at time t of a neuron i that has produced it last fire at time $t = 0$ is done by

$$V_i(t) \;=\; \sum_{n=1}^{t} \gamma^{t-n} \prod_{k=n}^{t-1} (1 - Z_i(k)) I_i^s(\mathbf{Z}(n-1)) \;=\; \frac{I_i}{1-\gamma} - \gamma^{t-1}\Big(\frac{I_i\gamma}{1-\gamma} + 2\alpha\Big) + J_i(t) \tag{2.38}$$

Proposition 3. *Let be $\tau_{i+1}^m, \tau_{i-1}^n$ the notation for the m-th and n-th firing times of neurons $i+1, i-1$ respectively. We have*

$$J_i(t) = \alpha \sum_{\{0 \le \tau_{i+1}^m \le t-1\}} + \gamma^{\tau_{i+1}^m} + \alpha \sum_{\{0 \le \tau_{i-1}^n \le t-1\}} \gamma^{\tau_{i-1}^n} \tag{2.39}$$

then, it may be written as

$$J_i(t) = \alpha \sum_{\substack{\{m\in\{0,\dots,t-1\}:\\ Z_{i+1}(m)=1\}}} \gamma^m + \alpha + \sum_{\substack{\{m\in\{0,\dots,t-1\}:\\ Z_{i+1}(m)=1\}}} \gamma^n \tag{2.40}$$

and now we can see that

$$0 \leq J_i(t) \leq \frac{2\alpha(1-\gamma^t)}{1-\gamma} < \frac{2\alpha}{1-\gamma}, \qquad t < \infty \tag{2.41}$$

Proposition 4. *The function defined as*

$$\begin{aligned}\tau_i = \inf_{t>0}\left\{\frac{I_i}{1-\gamma} - \gamma^{t-1}\Big(\frac{I_i\gamma}{1-\gamma} + 2\alpha\Big) + J_i(t) \geq \theta\right\} \\ = \inf_{t>0}\left\{t \geq 1 + \frac{1}{\log(\gamma)}\log\left(\frac{I_i + (1-\gamma)(J_i(t)-\theta)}{I_i\gamma + 2\alpha(1-\gamma)}\right)\right\}\end{aligned} \tag{2.42}$$

is a function of raster plot, i.e., it maps the firing times of the system

From last proposition on can construct the raster plots in a direct way computing all possibilities. Anyway, it shows us that the dynamic is periodic at the asymptotic limit except if $\tau_i = \infty$ that means it is a ghost orbit. Moreover, the period distribution is controlled by the Input I_i.

2.3.2 The minimal distance attractor-threshold: *dAS*

The first case we consider is the neuronal death where naturally, the minimal distance to threshold is $\min(1, 1 - \frac{I}{1-\gamma})$, where the second option comes from the case where input is present but its intensity is such that $I < \theta(1-\gamma)$ so not sufficient to take out the system from the η_0 state when it falls there. As the periodic orbit cases are more elaborated, for clarity we should discuss the input and non input cases separately beginning with the non input case. Suppose that for a P-periodic orbit , all neurones have fired at least once[3]. Suppose then, that the neuron k is the one in the orbit who gets closest to the threshold and name τ the time elapsed since its last firing time and the instant it arrives to the closest position. Thus, by hypothesis $1 \leq \tau \leq P$. If $\tau = 1$ that means that $V_k(\tau) = J_k(\mathbf{Z}(\tau - \mathbf{1})) = 0, -\alpha \text{or} -2\alpha$ and the minimum distance value may be $dAS = 1, 1+\alpha, 1+2\alpha$ respectively. More generally, $V_k(\tau) = \prod_{k=1}^{\tau}\gamma^{\tau-k}J_k(Z_k(\tau-k))$. so the minimal distance is $dAS = |\theta - V_k\tau|$. As we do not know the spiking sequences of all periodic orbits generated before simulate the dynamics we can not predict the form of the dAS dependences. However, we can use this construction to

[3]This assumption is almost an unproved lemma since except in presence of a very particular input signal, we find difficult to imagine any type regular diffusion of a perturbation in a chain that do not pass never through some sites of the chain.

explain any behavior founded. Moreover, we can can anticipate that for a fixed γ value dAS is a linear-piecewise continuous function of the α parameter. However, as simulation shows, the minimal size of a linear interval may be quite small.

For an specific case, lets assume that the system admits bi-periodic orbits in certain $[\alpha_{min}, \alpha_{max}]$ region. Then, the dynamics of neuron k is necessarily one fire time(by hypothesis of non null activity) and one quiescent time. At the quiescent time as showed above, $dAS \geq \theta$ so it cannot produce a null value. For the firing time, there are nine different possible configurations, that depends of the specific spiking pattern of each possible bi-periodic orbit. Two of them can not produce the values lower than $dAS = 1$. The straightforward computation of the relevant ones yields

$$dAS = \begin{cases} \theta - 2\alpha(1-\gamma) \\ \theta - \alpha(1-2\gamma) \\ \theta - \alpha(2-\gamma) \\ \theta - \alpha(1-\gamma) \\ \theta + 2\alpha\gamma \\ \theta - \alpha\gamma \\ \theta - 2\alpha \\ \theta - \alpha \end{cases}$$

We plot in figure **??** the conditions annulling dAS at some region of the parameter space γ, α. Some of this curves will correspond for more larger regions to the dAS zero values than the others as the creation of the bi-periodic orbits associated to the others may be not allowed everywhere. Also, this curves will not correspond to dAS zeros in regions where exist orbits of other periods that may have closer values to threshold than those of bi-periodic orbits. As an example, we found in the $N = 5$ simulation dAS zero values that are at same α values predicted by this equations, but when we look dAS as a function of α for a fixed γ value the slope do not corresponds to the predicted for any of this equations, so we realize that that zero curve coincides in its α solution values to those equation $dAS = 0$ but its slope is different as it correspond to an orbit of greater period with another $V_k(\tau)$ form. Now, for the case of input, this method remains applicable but, now we will have to different type of neurons and so its $J_i(t)$ will be different, letting doubling the number of terms appearing in the set of possible dAS values. 2.3.2 for one only value of external input, or multiplying it by the number of different input intensities. The, as seen from simulations, the input regime present a more complex structure with much more '$dAS = 0$ curves in the parameter space. Anyway, for a fixed γ value the character piecewise-linear is conserved .

2.3.3 Te minimal distance external input-discharge probability: dIP

The minimization of the distance to input is at this moment a question that requires more reflection as it is a harder question without any previous specific theoretical result. Indeed it seems at first look, that no answer can be done in a very general framework. However we

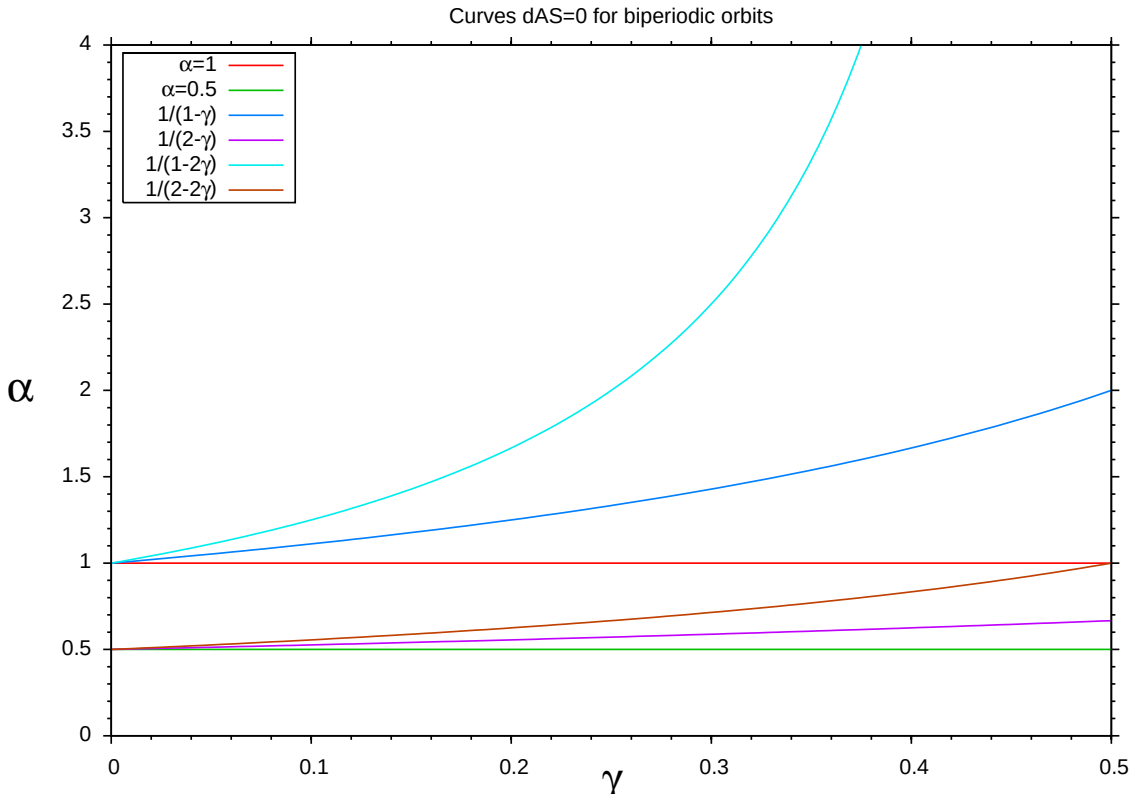


Figure 2.10: Curves for the condition $dAS = 0$ assuming only all bi-periodic orbits allowed

would like to set here some ideas that have been explored and the knowledge they have left us.

Definition 8. The vector **PD** is the set of discharge probability values of the network, defined as:

$$PD_i = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} Z_i \qquad \forall i = 1, \ldots N \tag{2.43}$$

and in a P-periodic orbit, one can drop the limit while replacing T by the value P. Thus, every PD_i is equal to the fraction m_i/P where m_i is the number of times the neuron i has fired during the whole orbit.

Definition 9. The distance between an Input $\mathbf{I^{ext}}$ and the vector **PD** which is calculated at the system with the same input is noted as dIP and the defined

$$\sum_{i=1}^{N} (I_i - PD_i)^2 \tag{2.44}$$

From the local perspective, not knowing the attractor set of the system before simulating it, we may ask for the best and the worst case of activity induced given an specific intensity of the step-shaped constant external input with the restriction $I < \theta$. The best case is that of neurons which receive input firing at least every two iterations. Since for any such neuron we have after firing $J_k(t)$ is equal to one o the values $-2\alpha + I, -\alpha + I, I$, we take the lowest one and suppose neuron do not receive any spike at next iteration letting the input to guaranteer by itself the spiking every two times. This yields to the condition $I \geq \frac{\theta+2\alpha\gamma}{1+\gamma}$. This is obviously an overestimation since dynamics involves neighbors neurones receiving also the input so therefore contributing with their spikes to maintain the high activity. This is why this condition actually implies $PD_i \geq 1/2$. Subsequently, for the worst case we take the case where the neuron has received all the possible spikes and the input signal but they still are insufficient to guaranteer the spike every two times. This yields us to the condition $I < \frac{\theta-2\alpha}{1+\gamma}$ which will imply necessarily $PD_i < 1/2$. This bounds will be important at the time of look for regions in the parameter space α, γ higher an lower activity for a fixed Input intensity value.

From a global point of view, we think about doing an statistic analysis of the activity of th NN. In this idea, for an step shaped-input we would wish be able to treat the neurons as two different groups: those which have external input an those which do not. Unfortunately, we can not do that since neurons in the borders of the input-and no input region (In a $\mathbb{T}^2$ chain, they are 4, two with their own stimuli and one neighbor stimulated while the other not, and the other two without any own stimuli and one neighbor stimulated while the other not [4]). Now we have two ways of avoiding this frontier problem;

- First one, supposed the system large enough to neglect this border effects, i.e., $N \to$ larger such that if k is the index of the middle frontier $\sum_{i=2, i\neq k, k+1}^{N-1}(PD_i - I_i)^2 \gg 2PD_N + 2PD_1$ since neurons $1, N$ define always the second frontier between input and no input region. Then we can neglect the different terms

- Or suppose the coupling constant, here, the α value who quantifies the spike force, is such that $\alpha \ll \theta$. Then the influence of neighbors everywhere is weak an we can assume $PD_N \approx PD_{N-1}$ and $PD_1 \approx PD_2$. In this way, we are not neglecting but correctly including the terms in the created groups.

As a remark, before continuing we can say that the second method is robust while the first one not, in the sense of network structure, cause if it changes i.e., we change the weight matrix, adding some other non null components, the number of frontier neurons may be rapidly increased invalidating our approximation of neglecting the frontier terms.

[4]Here, someone could say they are all different, since this difference will logically propagates to all neurons in the chain making them all differents a cause of their neighbor. It is true, but here we are concerned first with the first order effects, so we consider every neuron inside each of the two regions at exception of the border one as having identical neighbors

After assuming one of the two alternatives as hypothesis we can write

$$dIP = \begin{cases} \frac{N}{2}\left(\frac{\bar{n}}{P}\right)^2 + \frac{N}{2}\left(I - \frac{\bar{m}}{P}\right)^2 & \text{if } N \text{ is even} \\ \frac{N-1}{2}\left(\frac{\bar{n}}{P}\right)^2 + \frac{N+1}{2}\left(I - \frac{\bar{m}}{P}\right)^2 & \text{if } N \text{ is odd} \end{cases} \tag{2.45}$$

where $\bar{n}, \bar{m}$ are the average number of spikes during a whole orbit of neurons without stimuli and with stimuli respectively.

How to proceed now without knowing explicitly the spike sequence of orbits? We begin with the heuristic result that as larger the Network [5], larger the set of available periods. We understand this, actually as the number of different 'spike trains' or perturbations patterns that may travel periodically in the chain, like the increasing of the number of available vibrating modes on a excited string as the length of the string is increased. On the other hand we know from the proposition of firing time that when the system is allowed simultaneously to stay and to go out of the 'zero spike' state it can take every possible period orbit value, but more over, from the lemma of transition graph and the transition graph example in figure 2.9, we might accept that almost every average activity values of spiking neurons can be obtained by going out from η_0 and coming back to it from very specific spiking states. We could, expect even more, saying that all averages firing activities of neurones without input can be obtained with any period value. For this we should admit that a period cycle were almost completely conformed but at the end something failed and it fall to the neuronal death to going out sometime after. Although, this analysis is nor rigorous neither formally stated, we believe it may justify that we treat $\bar{n}, \bar{m}$ as free variables that may be adjusted to find the minimal orbit. Since this proceeding depends of allowing any P-periodic orbit, were are in the high input regime.

The figures 2.11-2.12 show in a density plot, the dIP value calculated with the precedent reasoning. With remark the displacement of the lowest value region (in black) with the change of input value. As a final idea, we propose the use of the limit conditions of the local point of view to improve this approximation since one should try to minimize the function using Lagrange multipliers for taking account of the limit values of the PD_i and a more reliable characterization of the lowest value region. However this step is only useful when α is little since the region of parameter space I, γ between the conditions $I \geq \frac{\theta+2\alpha\gamma}{1+\gamma}$ and $I < \frac{\theta-2\alpha}{1+\gamma}$ where no bound can be a priori imposed on the PD_i values, covers quickly the parameter space $[0,1]_I \times [0,1]_\gamma$ and when $\alpha = 0.5$ this region is already completely inside the non-specific bound region.

[5] Indeed we are on a large one in order to be able to do statistics over it

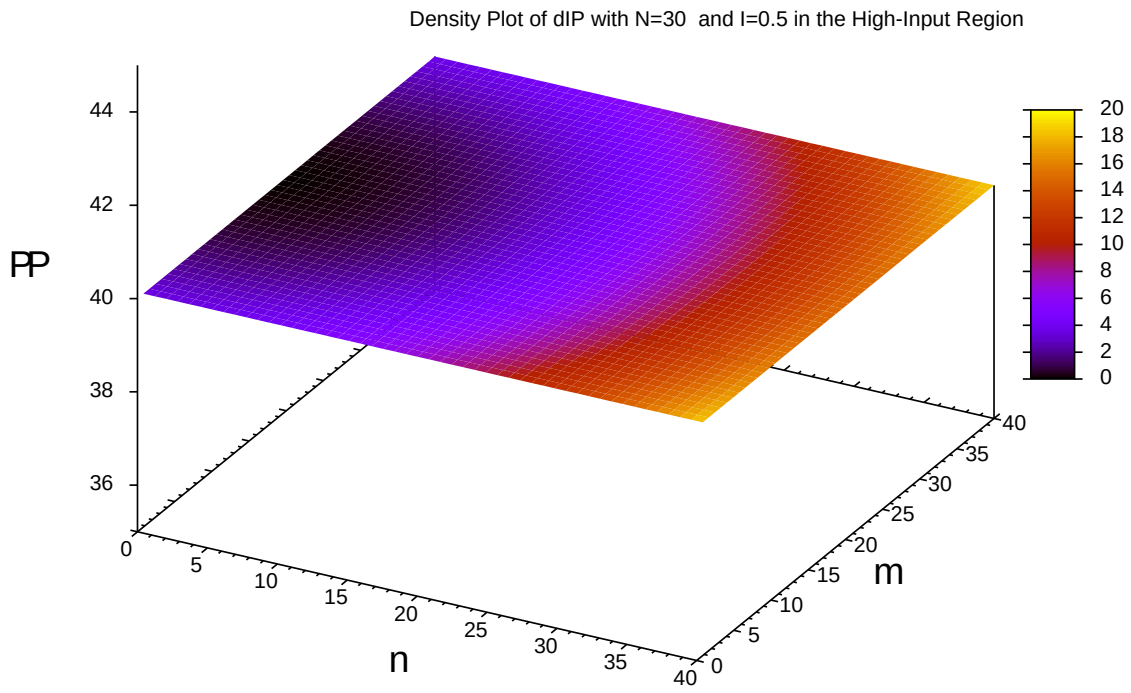


Figure 2.11: Density plot for dIP value with section 1.3.3 assumptions, I=0.5

2.4 Simulation and Numerical Results

2.4.1 The Simulation Problems

When dealing with NN, where the number of neurons is expected to be large, in order to have an reliable but also optimal simulation, it is important to have a good control of the parameters concerning the quality of the data given by the simulation, and therefore its dependences of the other Network parameters as size, synaptic weight etc. In BMS model these issues are crucial.

Beginning with a fixed size network, the simulation, in its simplest scheme is intended to run for different values (most as possible) of parameters α, γ and for each combination of them, to run over different values (restricted in the sub-threshold region) of the intensity for a step-shaped, constant in time, external current $\mathbf{I}^{ext}$, as it was considered in the last section. The simulation confidence, is controlled by the following 2 quantities:

i The 'number of Initial conditions' (NBCI), to be take randomly inside the N-dimensional

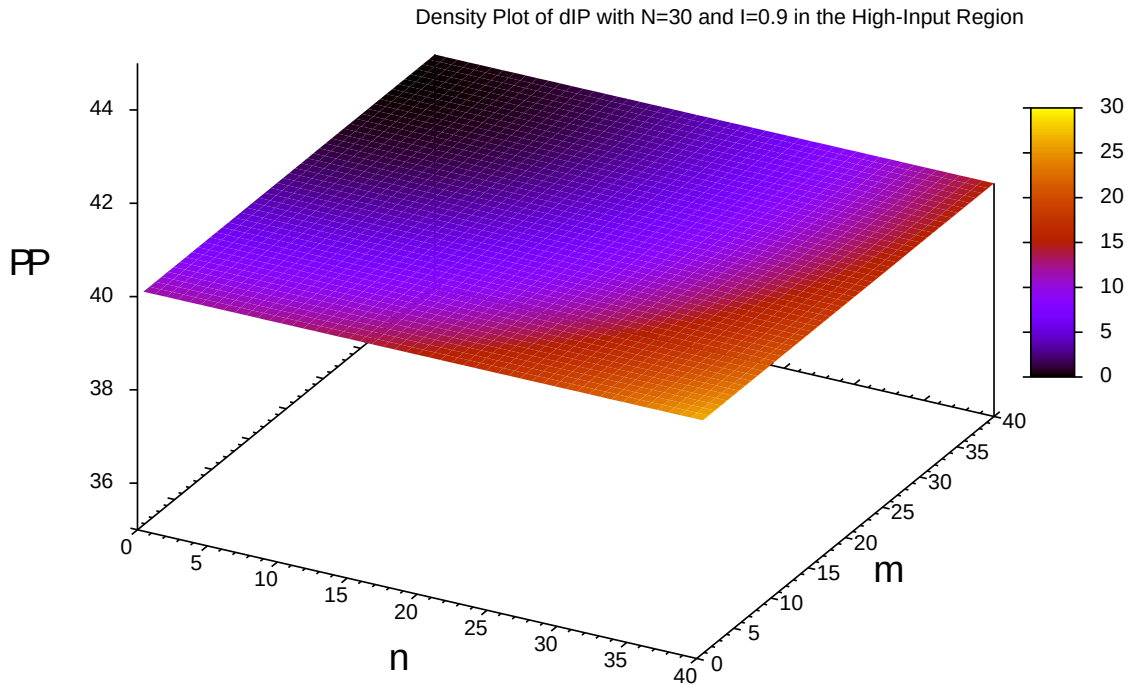


Figure 2.12: Density plot for dIP value with section 1.3.3 assumptions, I=0.9

space of membrane potential values, they are supposed to exhaust all the different attractors of the system.

ii The 'transient time' (T), who makes evolve the system in order to eliminate non-steady behaviors before the analysis of the forward orbit.

Is it possible to determinate *ab-initium* if NBCI and T are fixed values or if there exist dependencies of the other parameters $(N, \alpha, \gamma, \mathbf{I}^{ext})$? Obviously, the answer will depends of what we are looking for, i.e in our case, to characterize the NN and its response to the input signal, by computing the minimum distance of the attractor set to threshold and the distance between the external Input signal and the response (output) signal. However, as we shall discuss an analytical answer may not exist or rest only as an estimation.

The first difficult is dealing with the neuronal death state and the non steady-spike sequence supposed to be complete between the lapse we defined as 'transient time'.We know that the system may goes to neuronal death or periodic activity. but almost always passing through a sequence of non-steady spiking sequence that depends on the initial condition.

However, the stationary state of periodic orbit will not be found never before all neurons have fired at least once and using results of the last section we know that the time all neurons take to discharge at least once may be arbitrary large and indeed grows exponentially with γ.

Therefore, the election of a transient time instead of being meaningful becomes delicate since one can not be sure of the value at which we began to look for the period of the orbit[6]. The natural option to assume 'ad-hoc' period = 1 for this cases will let us with a not optimal code to differentiate neuronal death of larger period orbits in large size systems, since simulations shows BMS Laplacian NN has natural orbits of order of its sizes. Moreover, taking that solution will leave unrecognizable regions with orbits of periods of order 2^{NT_D} ([3]) [7].

To resume the problem, we can not know *a priori*, how long the non-steady spiking sequence, which depends of the IC can take to dissapear and get to neuronal death or onto a periodic orbit, cause the raster plot function associated will contain dependencies of the initial condition itself and anyway it contains the term J that has inside actually the information of the dynamics of all the network. The final consequence of this it that chossing any transient value will create inevitably a dependence of the $NBCI$ value to be taken. We shall try to show below how it happens.

Suppose now that when simulating, we take a transient time and NBCI fixed values. Then, we will take randomly many ICs and hopefully all of them go onto an steady periodic orbit or inside our 'numerical neuronal death' i.e.- a N-dimensional ball centered in the neuronal death vector as defined in2.33 and of radii ϵ 10^{-8}- in time inferior than our transient time. Then the $vectV$ taken for characterize the forward orbit will give us a reliable values. But, taking much more ICs we can overpass our chance and fall on a weird initial condition $\mathbf{V}(0)$ which at a the after the transient time is not yet a point inside the possibles steady states of the system nor a vector ϵ close to the neuronal death state. Wherever this vector is going, the simplest and fastest one algorithm to find the period will fail (and without a careful code it will go onto an infinite loop)and moreover the simulation may give wrong values in the other quantities computed as they mostly will include values of the non-steady behavior lapse.

The only way we find to treated this problem of the transient time at simulation is manually fixing the transient time and $NBCI$ values in a way the transient time be high enough to handle the number of random NBCI proposed and do that for each region of the space or parameters. As the we believe, this method can be accomplished successfully in the case of BMS Laplacian because the α dependence is very weak a cause of the 'always excita-

[6]Clearly, taking a 'wrong' membrane potential vector $\mathbf{V}$ as reference for looking its period, will gives us no period if it is actually going to neuronal death.

[7]This last argument may be more relevant in other types of BMS models since as we know the BMS Laplacian at this level of external input restriction does not present this higher order periodic orbits type.

tory character' of synapses since they can make fire easily the neighbors in most part of the range α considered thus conducing very quickly to the stable periodic orbits. In that way, we look for a a transient time only in function of the parameter γ. Nevertheless note that this is not the best situation we can hope for a simulation since we remember the first firing time depends of γ in an exponential way so the transient time too. As consequence it would be difficult to simulate larger networks for the complete range of the γ parameter. We include the curve of the suitable transient times and NBCIs values found for a reliable simulation or $N = 5$-network.
However, a clear answer about why a manually fixing of this values may be acceptably done

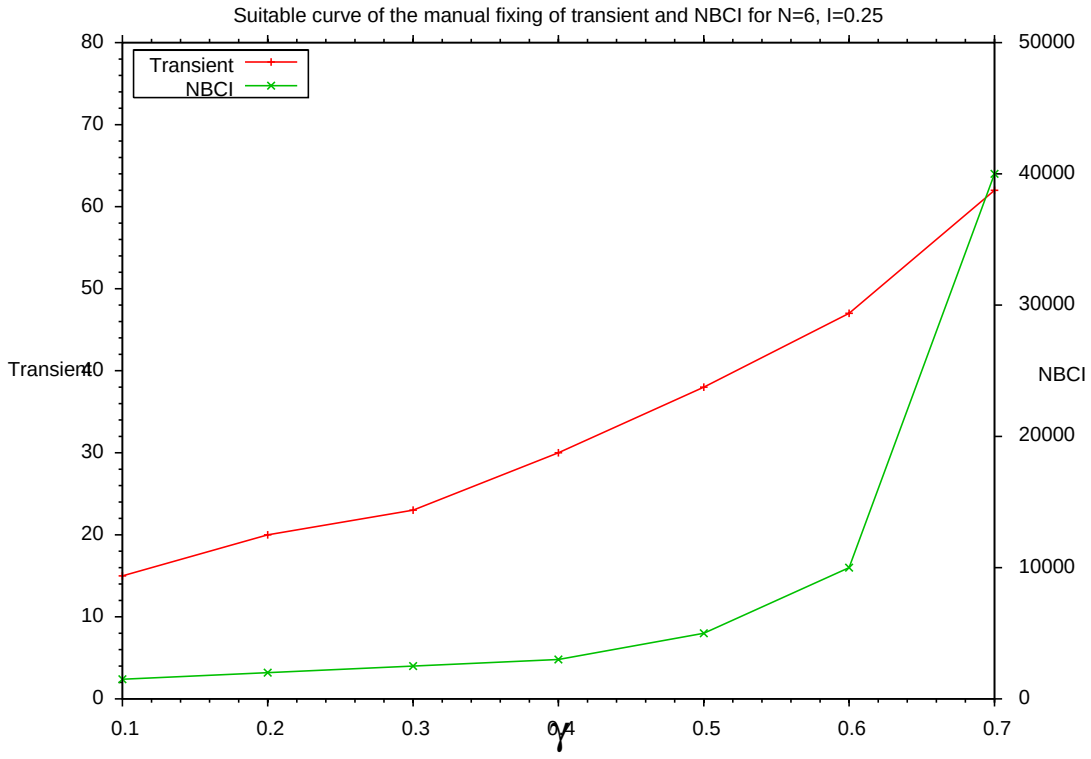


Figure 2.13: Curve of NBCI and respective transient time found for a suitable simulation of a N=6 BMS-Laplacian Network

is mostly related with the second major problem of simulating every (in general) BMS model. This problem comes out from the fact we do not know the order of the Markov partition of the membrane potential N-dimensional space before computing the $d\mathbb{A}S$. Therefore, certain parameter regions may need higher $NBCI$ in order to exhaust completely the attractor set and finally compute its lower $d\mathbb{A}S$ and the orbit responsible of it. Then, for this regions we would be obligated to increase the transient time in order to eliminate non-steady behavior results as discussed above.

To determine if we have choose a good NBCI to exhaust the attractor set, we simulate one γ, α point of fixed size network with a very high NBCI value. Then we take a more acceptable in a numerical sense, NBCI value for simulate more the surrounding regions expecting obtain similar characteristic values. If the results shows large differences between the characteristic system values obtained and those produced with the last simulation, we increase the $NBCI$ until the results are similar and this is robust in $NBCI$.Then we look for a lower transient value capable of contain all non-steady behaviors produced with that NBCI and after find it we are ready to simulate all the region optimally, i.e, taking many closer points in the parameter space.

2.4.2 Simulation Results

The following results corresponds to a simulations run for $N = 5$ and some explicitly noted for N=8. When an input is present its form is done by $\mathbf{I} = (0, 0, I, I, I)$, i.e., the neurons $1, 2$ do not have input while the neurons $3, 4, 5$ has the same intensity input value. Most of results use the density color map(DCM) where a 3-D data set is plotted as mapped on 2-D region using a color scale to represent the third coordinate. This results very useful since to study a very irregular 2-D surface is from far more difficult. Nevertheless this technique is not directly clear since the number of colors it used to map is normally 256, so that might hide information were the third coordinate takes a lot of different but closer values. As we shall see, a map only for that region is the solution to visualize particular structures [8]. We note in figure 2.14 that the distribution of curves $dAS = 0$ is very similar a those plotted theoretically in section 1.3.2. We see in the figure 2.15 the period distribution in the parameter space. We note in figure 2.16 the strong influence of stimuli in the the distribution of curves $dAS = 0$. We note also the more complex regime after $\gamma = 0.6$ as predicted theoretically in section 1.3.2.In next figure taking a view on a fixed γ value we show that the surface is not regular, and is more complex than we initially believe and even more if we take an input slightly higher. We take a general zoom of the high input region, and look at the minimal dAS and the period distribution of the corresponding orbit. But we can retire the zoom and look for the period of the corresponding orbit in all the simulated region in figure 2.20. We realize that even, while the distance to the attractor stay stable in value, the period of the orbit is now aleatory distributed between orbits of period 8 and 4.

Figures 2.21,2.22 show that the high input region ($I > \theta(1 - \gamma)$ presents more complex behavior. We may take a zoom for the high input region to see its complexity in an explicit way(figures 2.23,2.24.

In figure2.25 we show all the discharge probabilities for a fixed Input and γ values. First we confirm that the activity of neurons depends of the type of neighbors they have

[8]Even if the number of colors used to map had not limits, the eye resolution and the number of pixels of the display screen are limiting factors to see an arbitrary set of 3-D data.

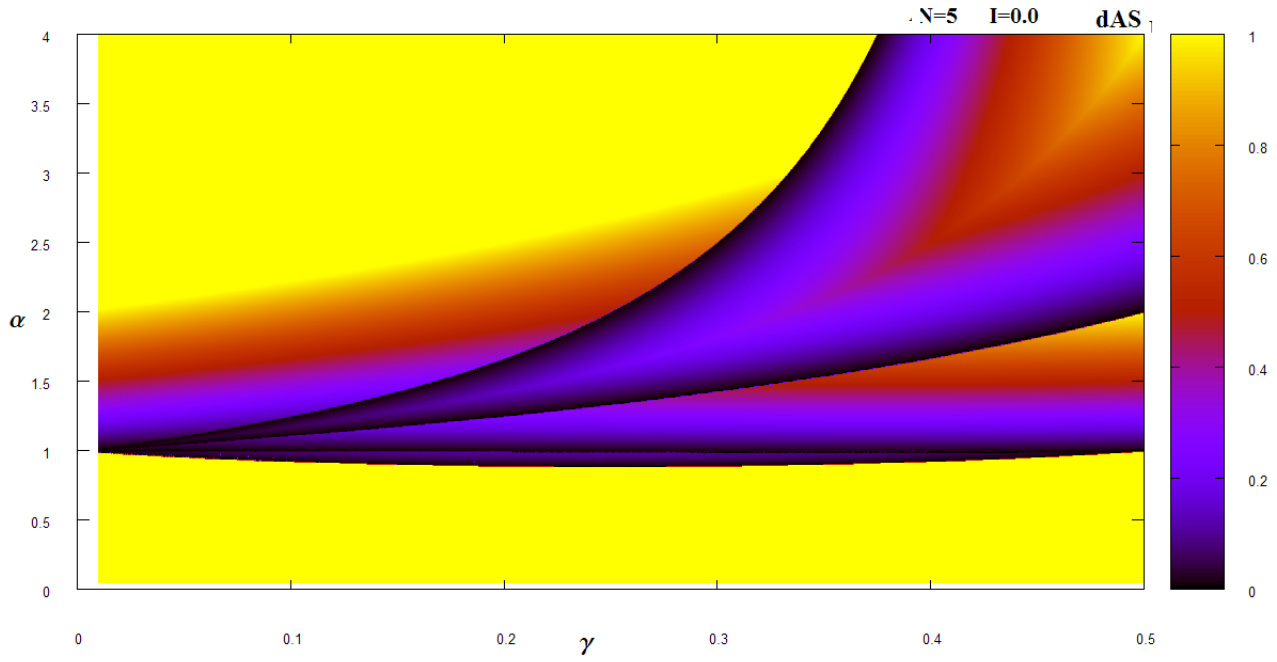


Figure 2.14: Density Color Map (DCM) of the minimal distance between atracttor -threshold for a 5-BMS-Laplacian NN with non external input.

as mentioned in section 1.3.3. That let us with only three different discharge probabilities.Nevertheless in most part of the parameter space region, the minimal dIP is achieved by an orbit with the same activity for all the stimulated neurons. Moreover, we note that for weaker synapses the activity of the neurons without stimuli can remain null but for stronger synapses this is not possible. Finally, a surprising result is that in the α region where the minimizing orbit present null activity in the neurons without stimuli, the activity in the neurons with stimuli can became lower and still minimising the dIP value. This effect is the competition between parameters that allow or not certain orbits to exist. We prefer postpone the DCM of the three different Discharge Probabilities to the end, and continue here with another important result.
Now we look at a following set of graphics for a N=8 Network. We show the minimal distance dIP for to fixed $\alpha values$ in all the range of the parameter γ but for different inputs in each graph. We have already shown in a the 5-Network that in the high input region the complexity in the parameter space is higher and here is verified too but what we want to show is that for certain Input values the complexity region presents a similar behavior in the values distribution but for a more higher input value the distribution of values of the dIP change radically its form. Another fact we would like to note is that at each specific input value, in a fixed γ value the period for the lower α parameter is inferior or equal to the corresponding to the higher α value. This is shown in figure 2.29 only for the input value $I = 0.8$. What

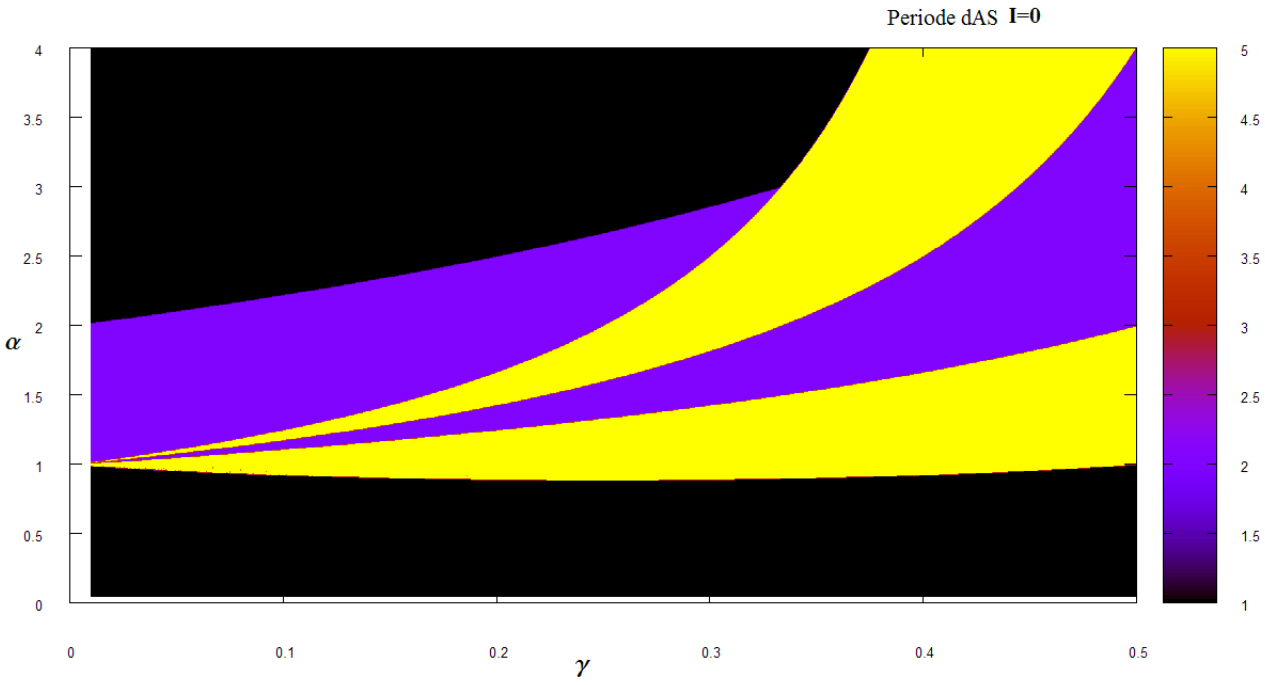


Figure 2.15: DCM of the Period of the orbit producing the minimal dAS in figure 2.14.

this shows is that for weaker synapses the complexity of the neuronal activities produced are lower those of the strong ones.

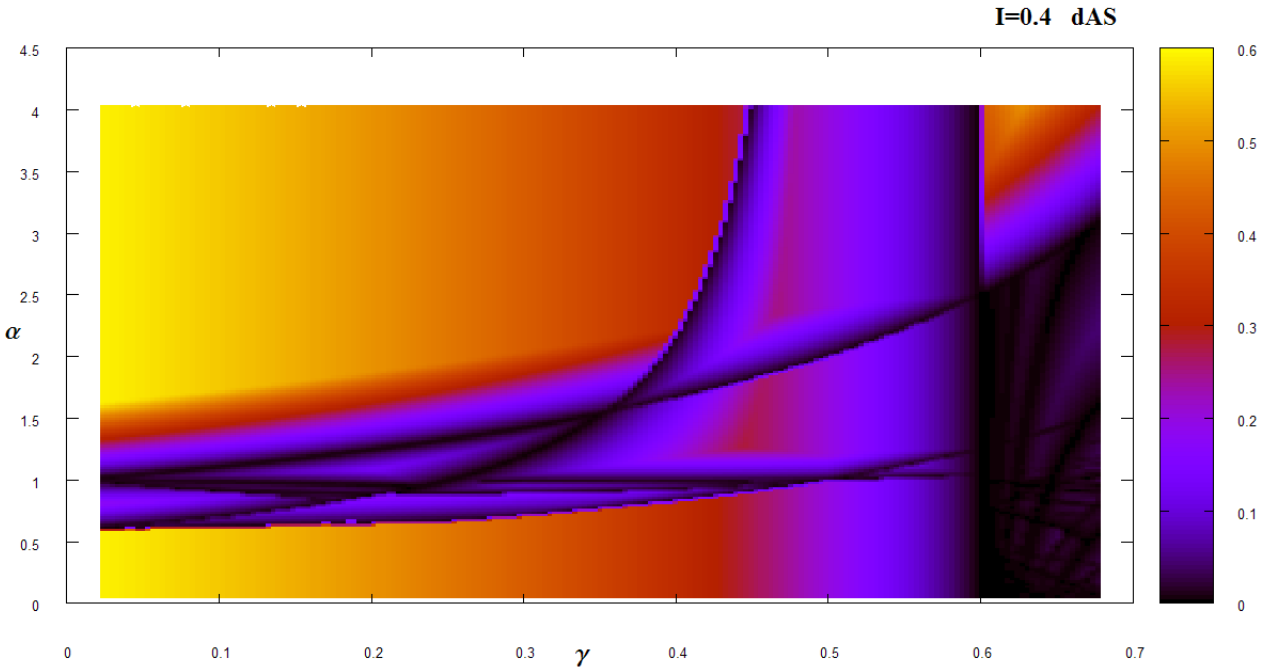


Figure 2.16: Density Color Map (DCM) of the minimal distance between attractor -threshold for a 5-BMS-Laplacian NN with Input I=0.4

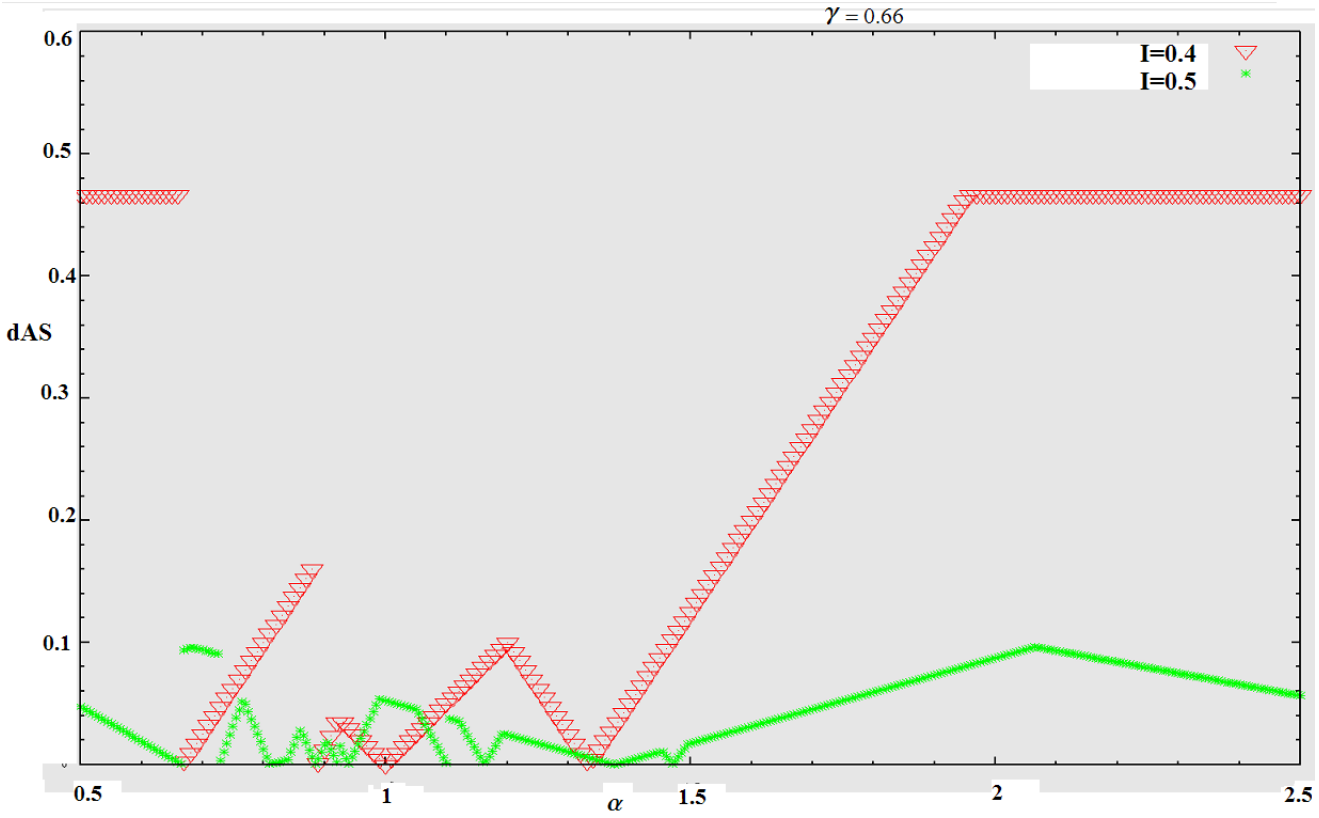


Figure 2.17: minimal distance between attractor -threshold for a 5-BMS-Laplacian NN with Input I=0.4,0.5 and $\gamma = 0.66$.

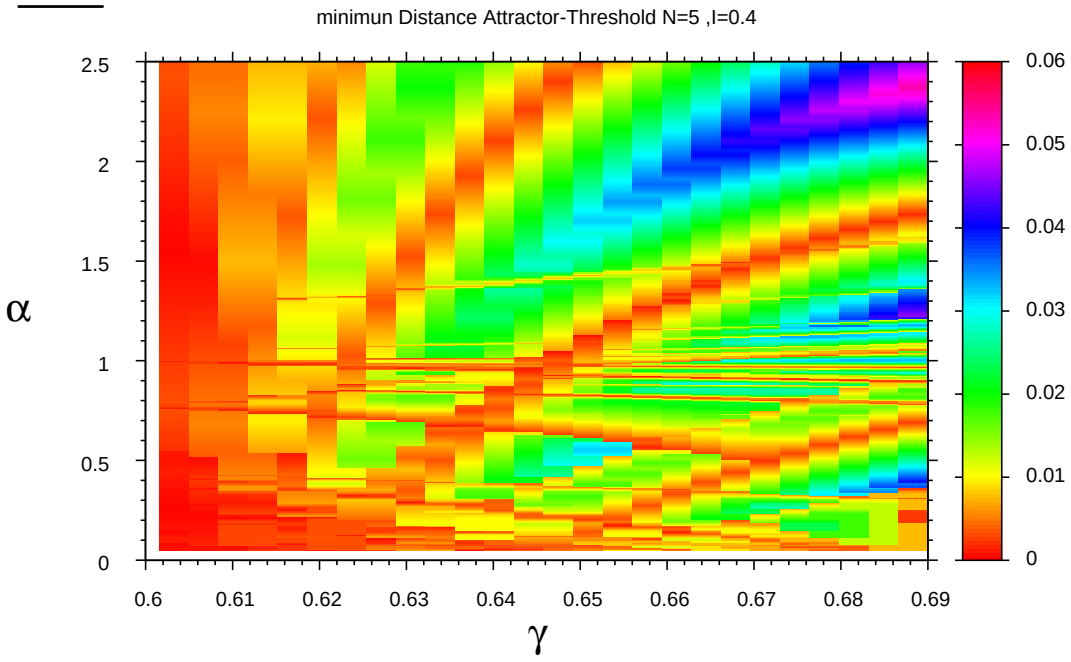


Figure 2.18: DCM of the minimal distance between atracttor -threshold for a 5-BMS-Laplacian NN with Input I=0.4

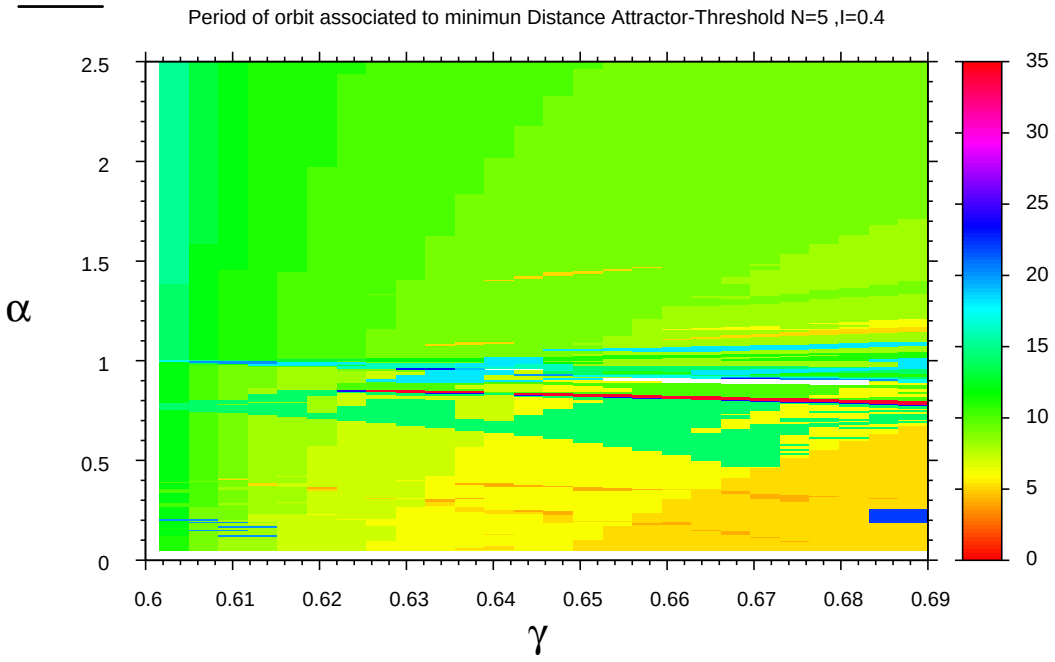


Figure 2.19: DCM of the period of the corresponding orbit to the minimal distance between attractor -threshold for a 5-BMS-Laplacian NN with Input I=0.4

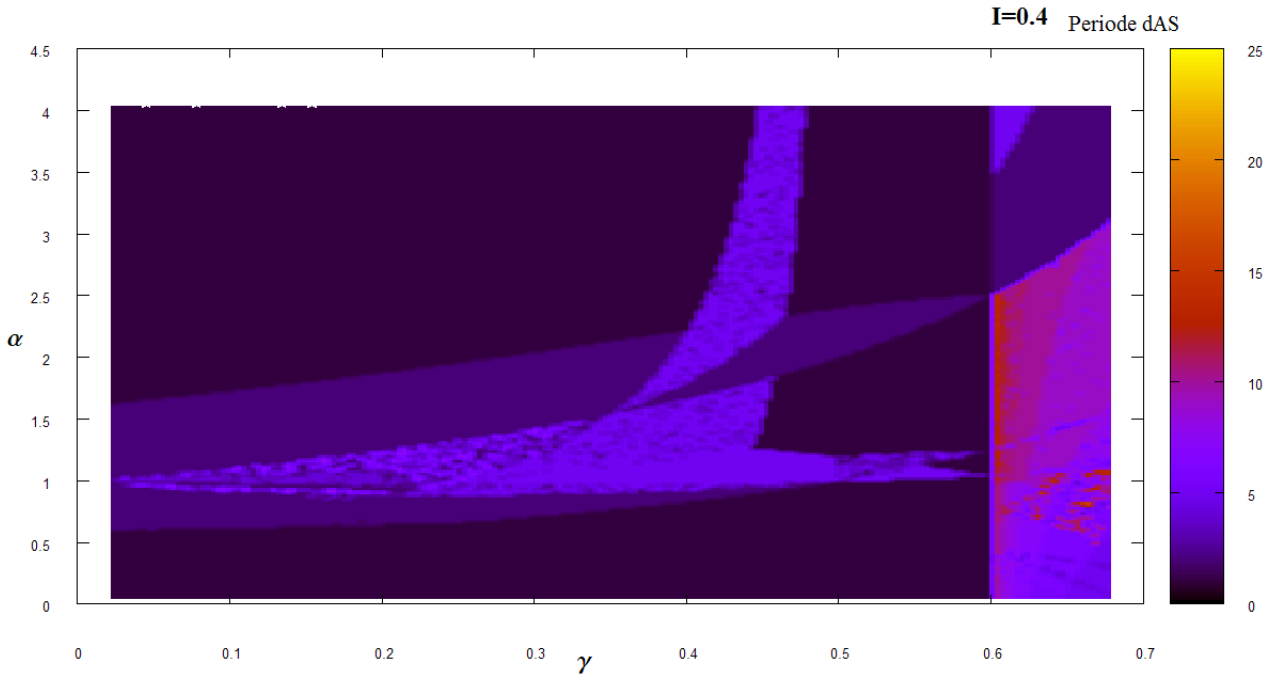


Figure 2.20: DCM of the period of the orbit corresponding to minimal distance between attractor -threshold for a 5-BMS-Laplacian NN with Input I=0.4

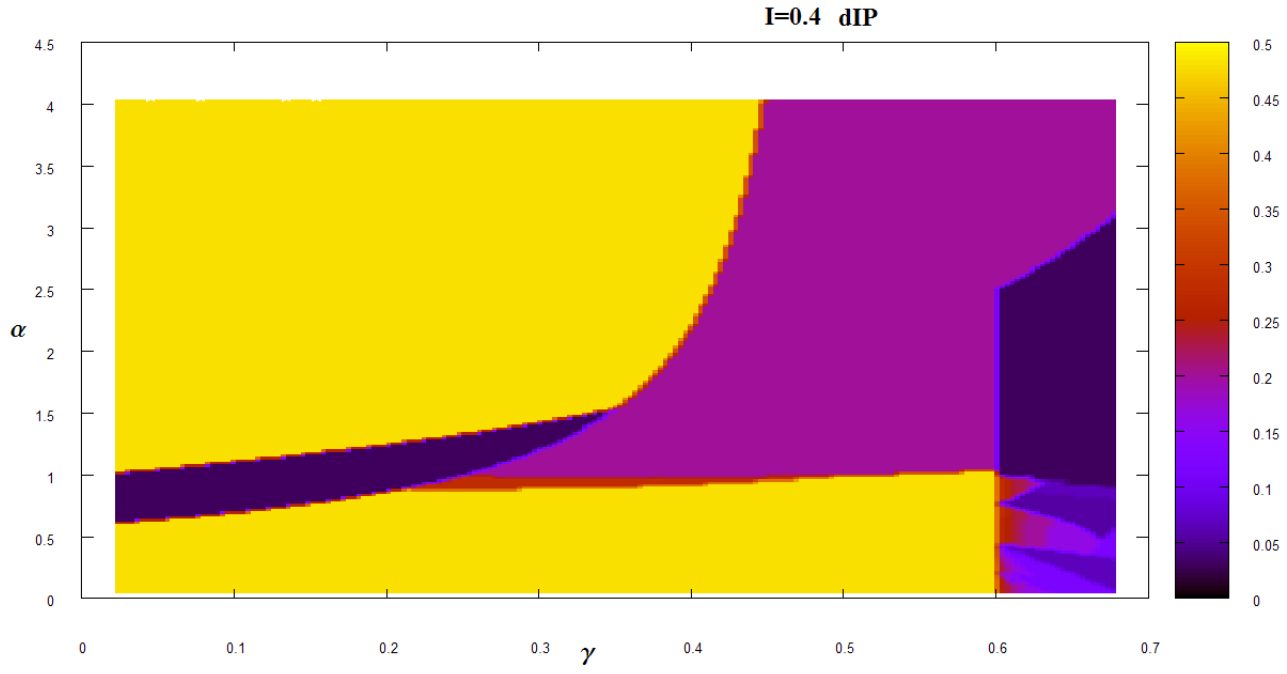


Figure 2.21: DCM of the minimal distance Input-Discharge Probability for a 5-BMS-Laplacian NN with Input I=0.4

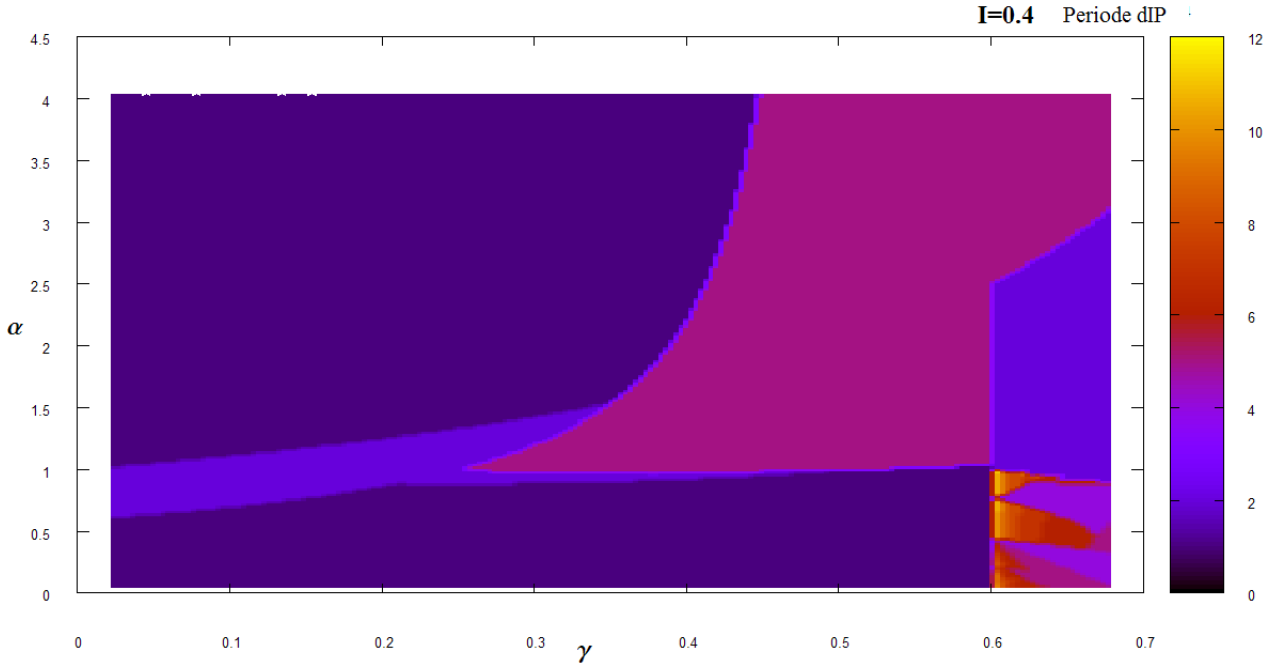


Figure 2.22: DCM of the period of the orbit corresponding to the minimal dIP in figure 2.21

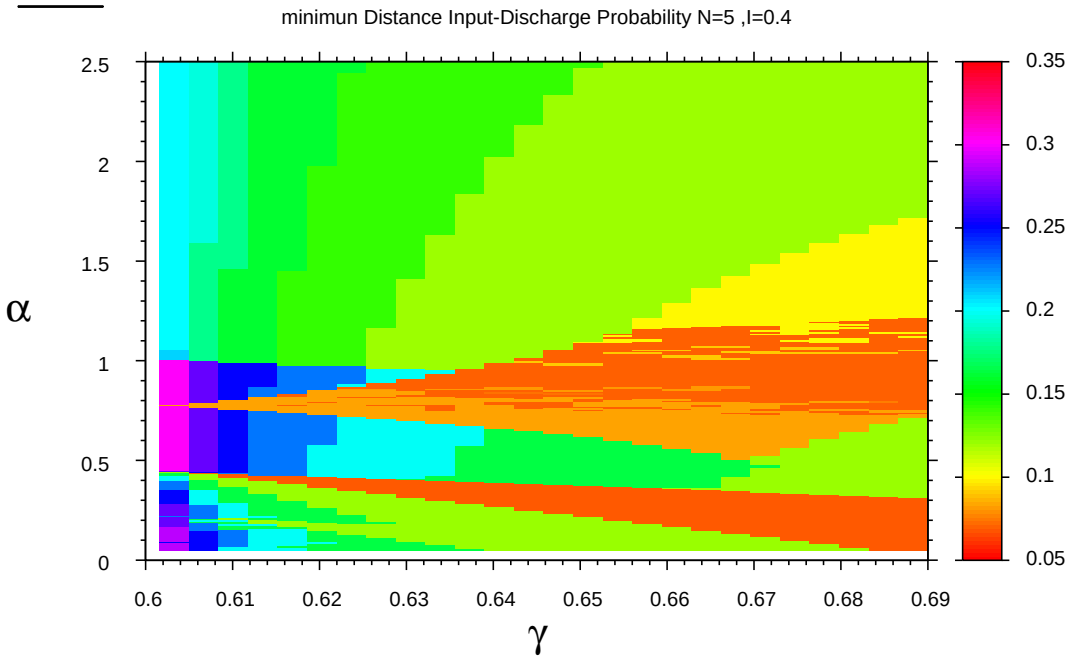


Figure 2.23: DCM of the minimal distance between Input and discharge Probability for a 5-BMS-Laplacian NN with Input $I = 0.4$

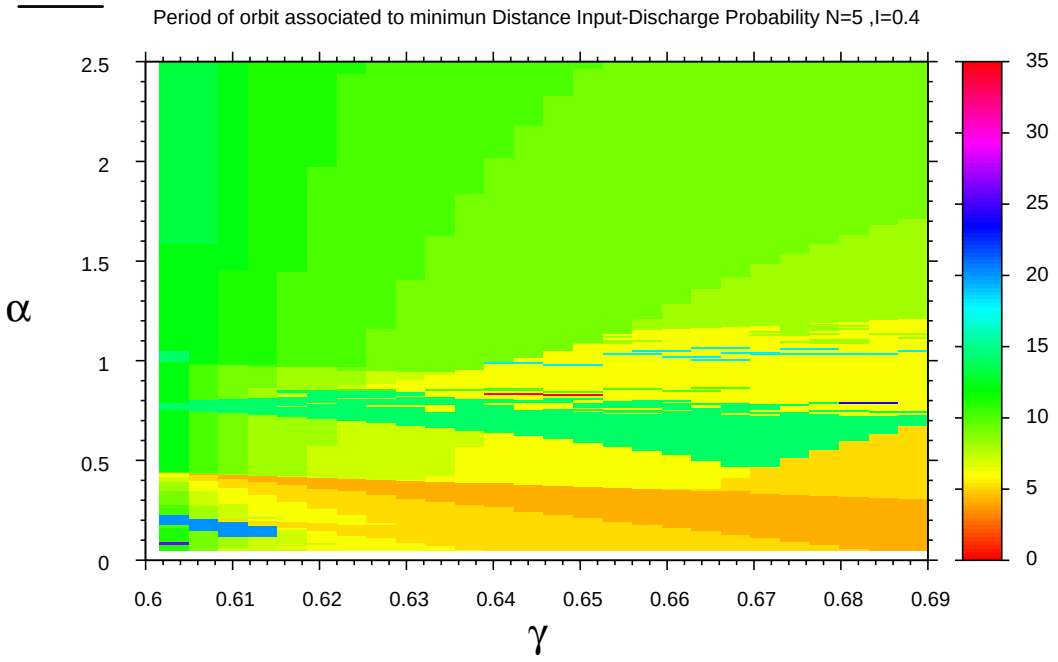


Figure 2.24: DCM of the period of the corresponding orbit to the minimal distance between Input-Discharge Probability for a 5-BMS-Laplacian NN with Input I=0.4

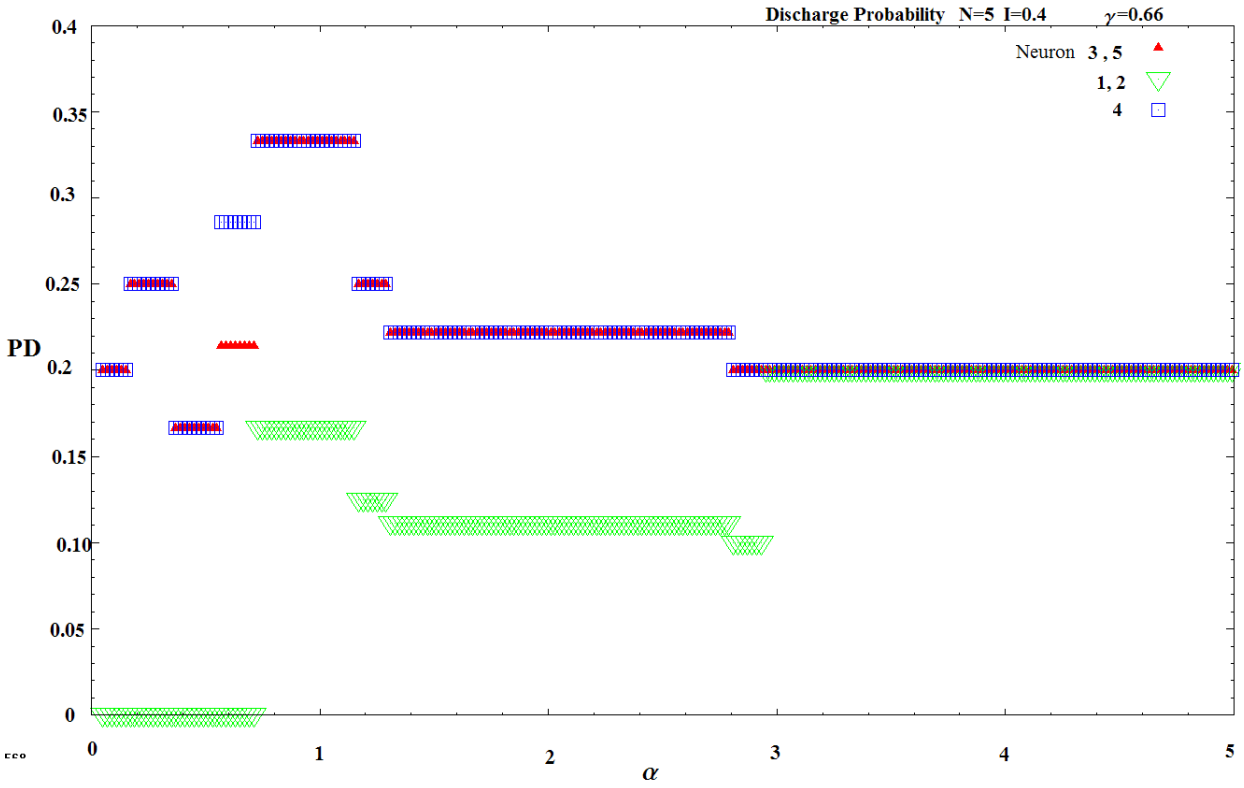


Figure 2.25: Discharge Probability of the 5 neurons of a BMS-Laplacian NN with Input I=0.4 from the orbits that minimize the distance with the Input vector

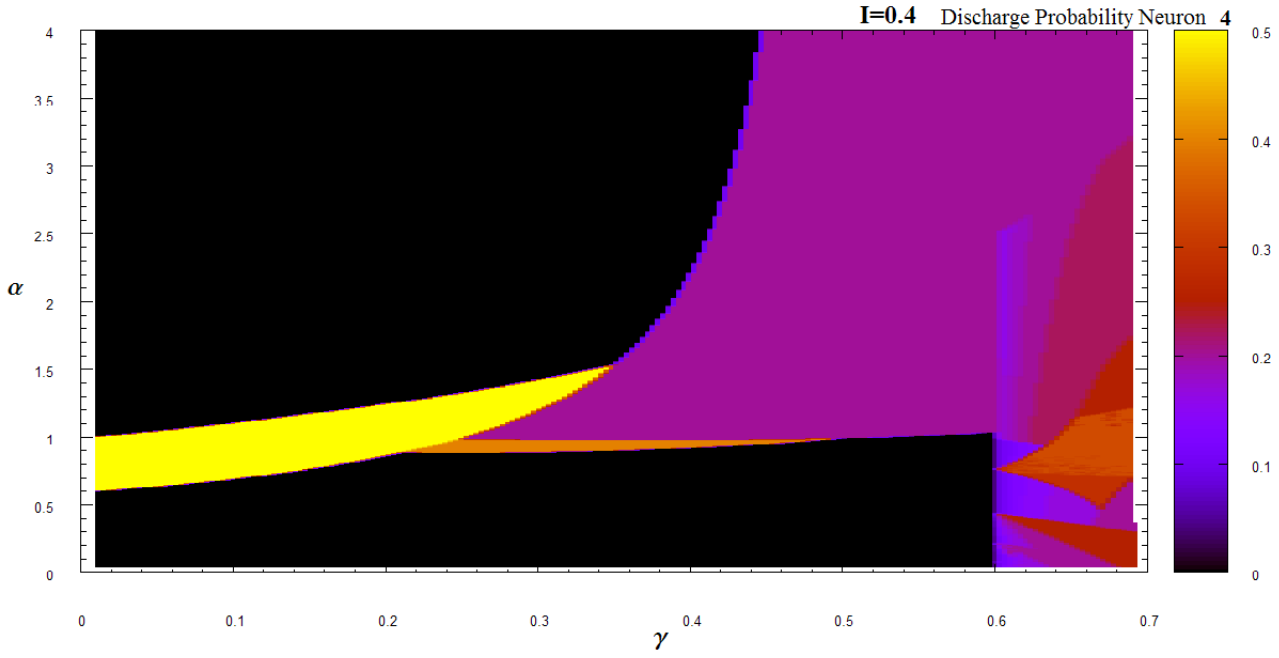
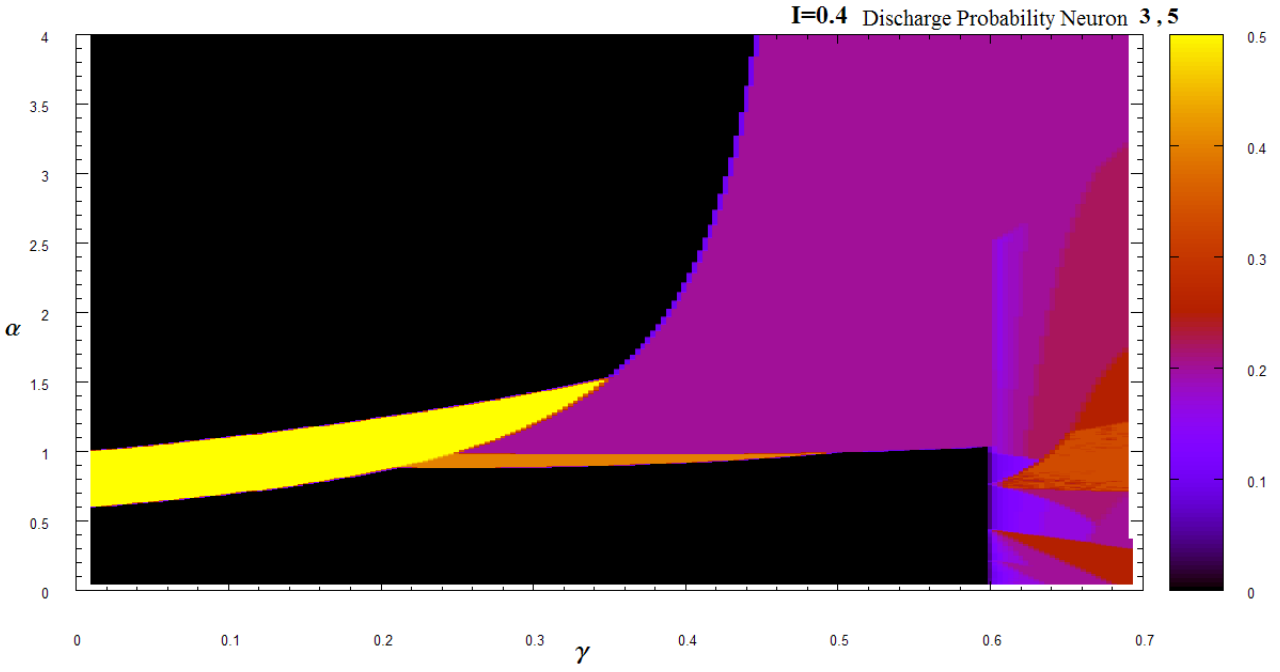
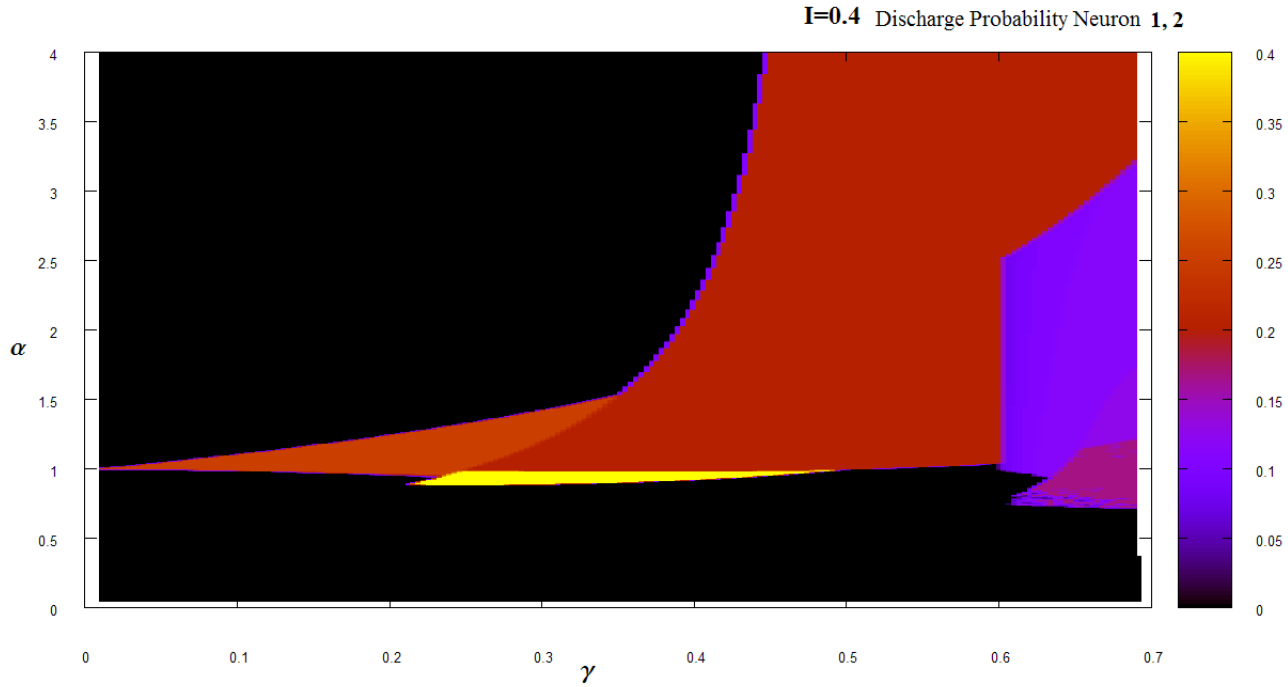


Figure 2.26: DCM for the 3 different discharge probabilities that minimize the distance to the output in $N = 5$ BMS Laplacian NN with $I = 0.4$


I=0.4 Discharge Probability Neuron 3 , 5
α
γ
0
0.5
1
1.5
2
2.5
3
3.5
4
0.1
0.2
0.3
0.4
0.5
0.6
0.7


I=0.4 Discharge Probability Neuron 1, 2
α
γ
0
0.5
1
1.5
2
2.5
3
3.5
4
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.05
0.15
0.25
0.35

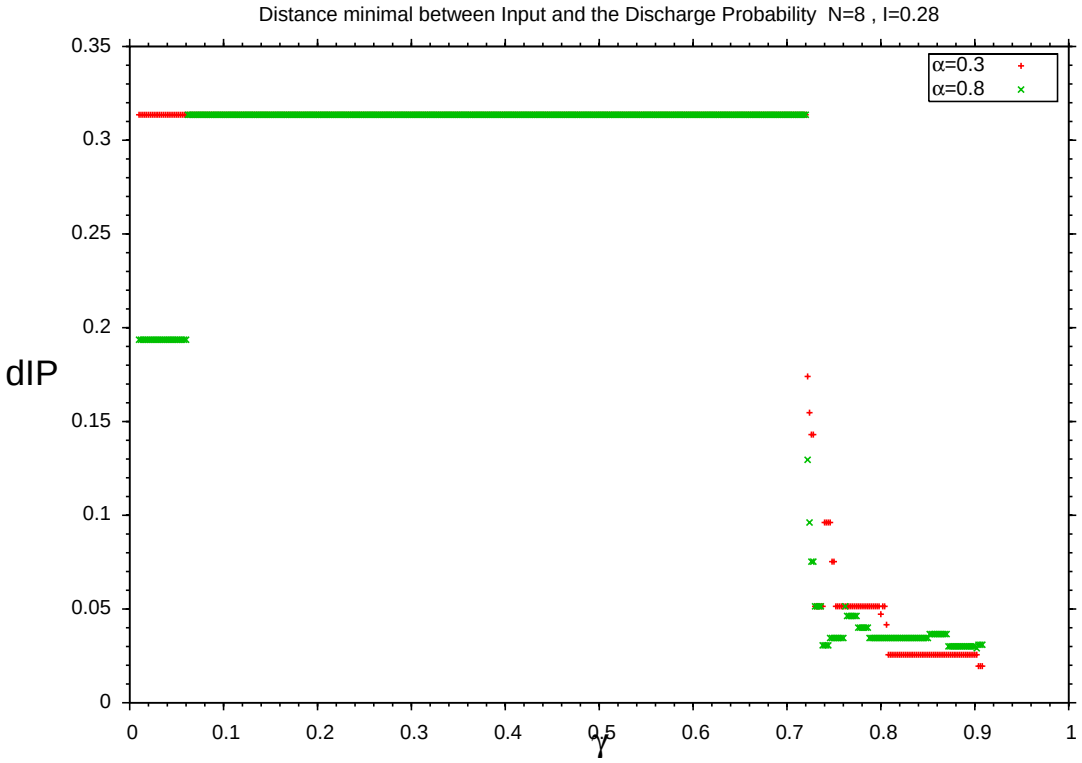


Figure 2.26: Minimal distance between Input and discharge Probability for a 8-BMS-Laplacian NN with Input $I = 0.28$

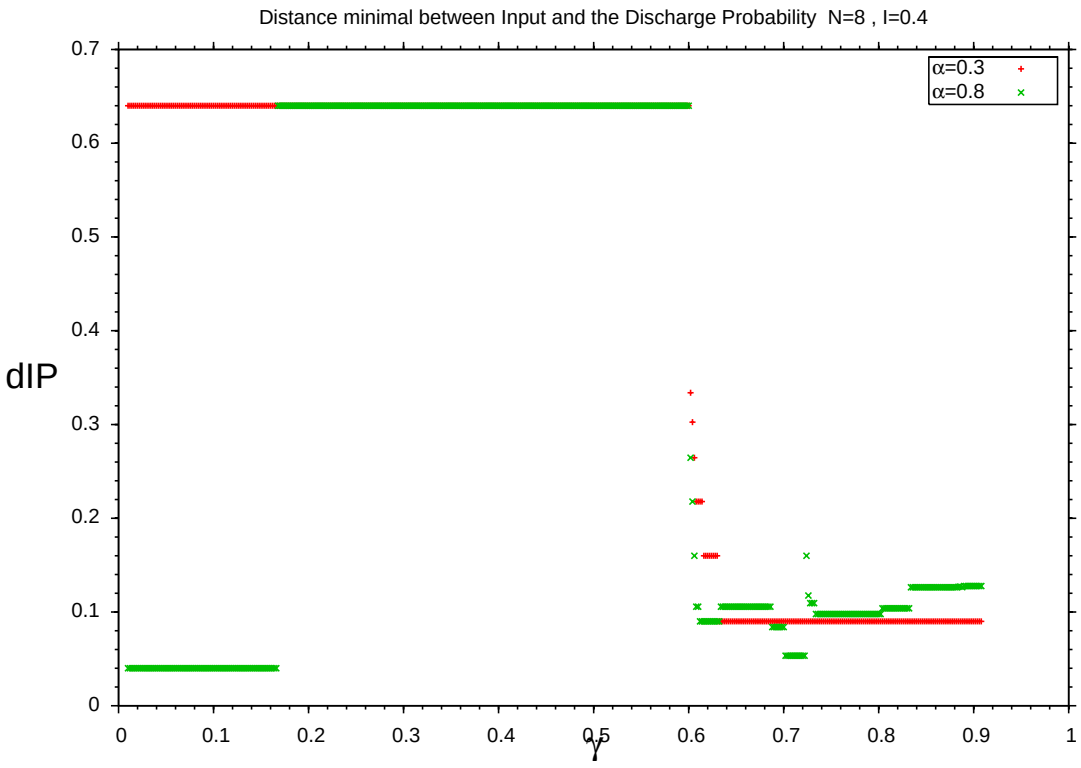


Figure 2.27: Minimal distance between Input and discharge Probability for a 8-BMS-Laplacian NN with Input $I = 0.4$

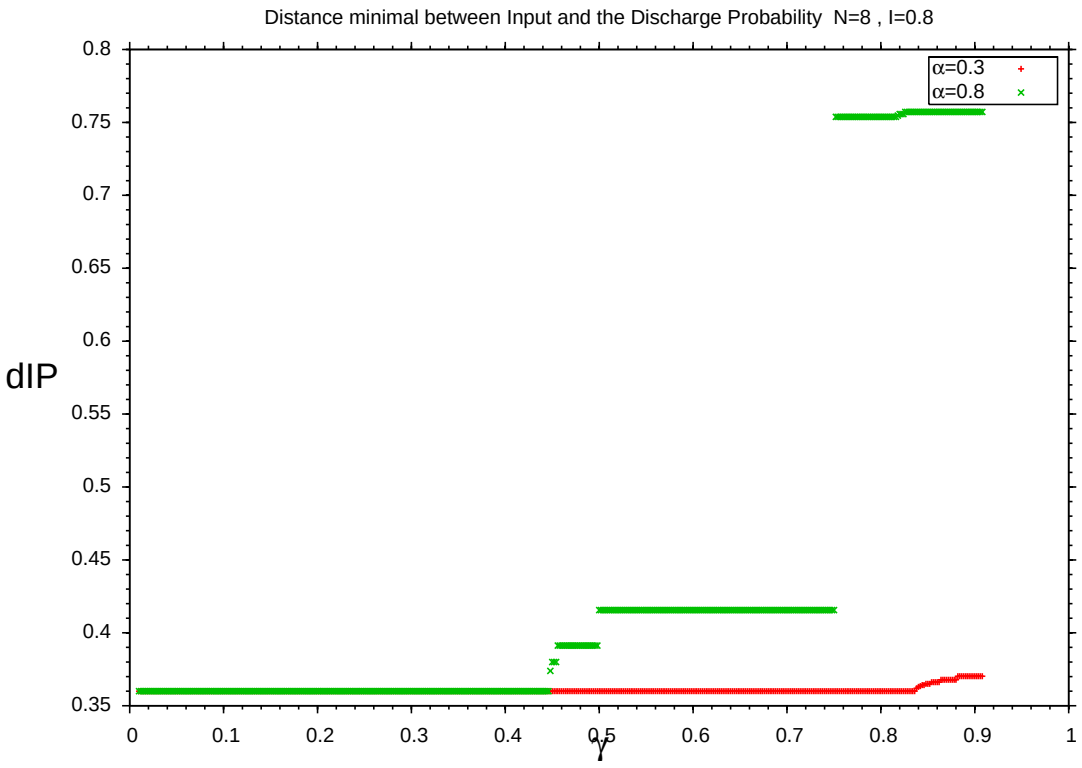


Figure 2.28: Minimal distance between Input and discharge Probability for a 8-BMS-Laplacian NN with Input $I = 0.28$

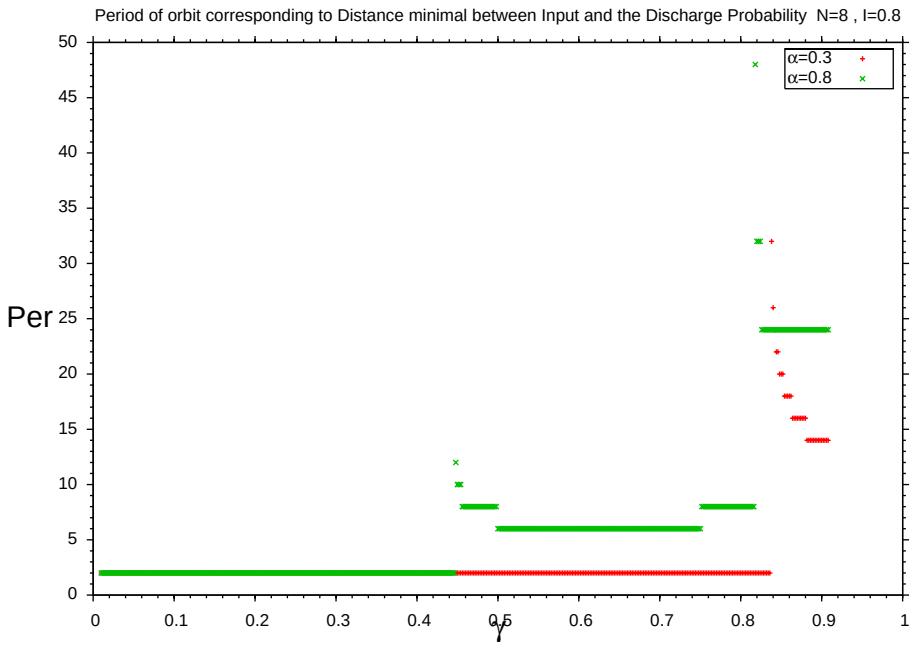


Figure 2.29: DCM of the period of the orbit corresponding to minimal distance between atracttor -threshold for a 5-BMS-Laplacian N with Input I=0.8

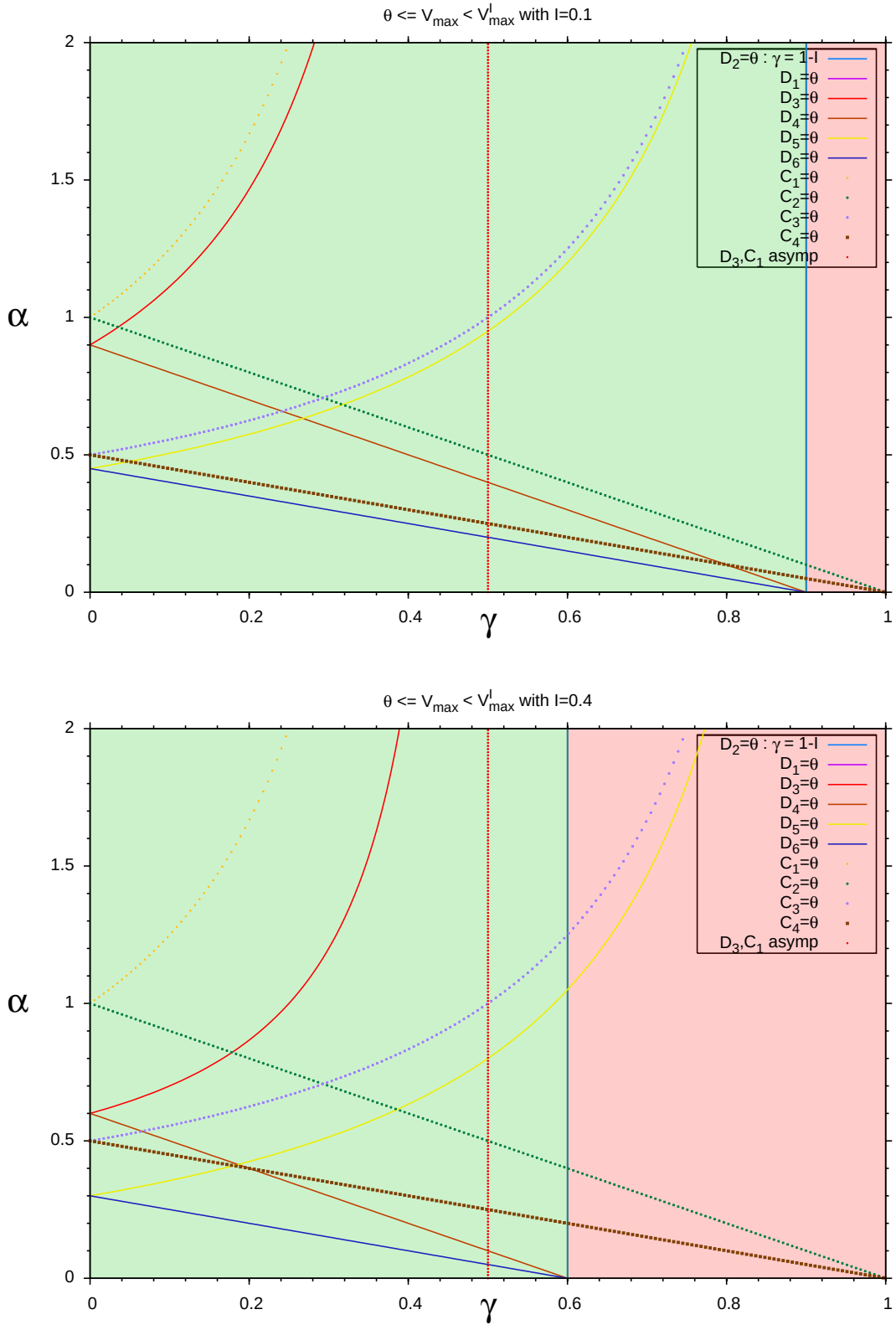


Figure 2.30: Diagrams of dynamical regions from eq 2.24 in the parameter space (γ, α) for different I values. (continuing next page)

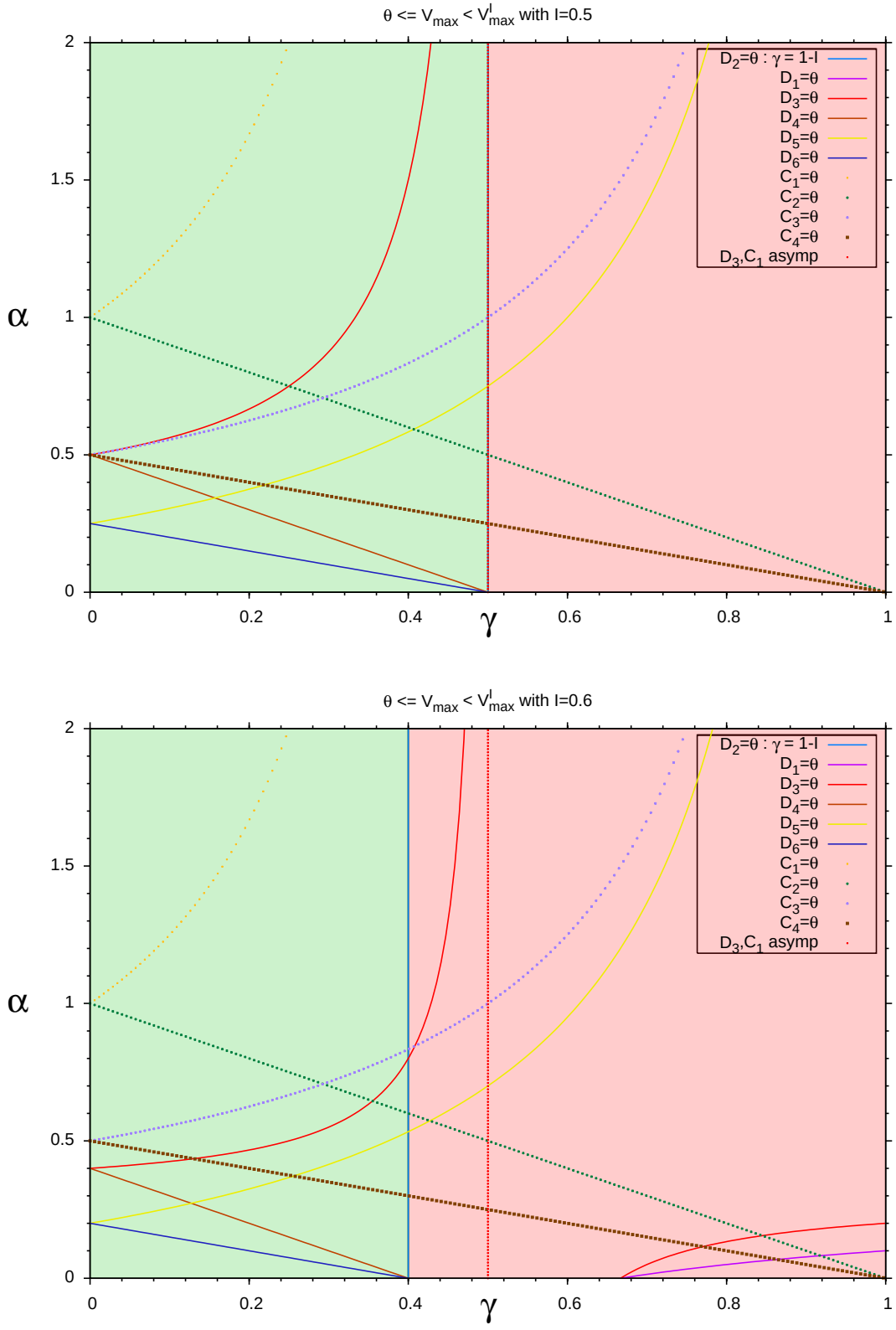


Figure 2.31: continuation(see figure 2.30 caption)

CHAPTER 3

CONCLUSIONS AND FUTURE RESEARCH

3.1 Conclusions

- We have verified at least numerically that in the BMS-Laplacian Model without applied external current, the minimal dAS and dIP are produced by orbits of period equal or lower than the Network size. As we have demonstrated that no orbits of period greater than N can not exits for $N = 3$-networks, we are induce to believe that this result is general for arbitrary size networks.

- We have demonstrated that in the BMS-Laplacian Model, the NN is very sensible to stimuli and it can cause many effects as:

  i Make the distribution of the orbits that produce the minimal dAS randomly in certain regions of parameter space in even sized Networks.

  ii If $I > \theta(1 - \gamma)$ they exist periodic orbits of period as large as one may want adjusting the parameter values.

- We conclude then that the BMS-Laplacian model has periodic dynamic except for a generic set of the configuration space, in all the parameter space simulated region. Nevertheless, the structure of the ω-limit set is complex and depends in a strong way of the parameter space and the input intensity value.

- After looking to numerical results and previous theoretical analysis, we do not see in clear way to related the minimization of dIP and dAS. But here we can state two numerical facts:

  i Anyway, it is true that satisfying the condition $I > \theta(1-\gamma)$ means in both cases to open the door to more complex parameter space distribution characteristics since it is the condition allowing the creation of more complex orbits.

  ii The weaker the spikes , i.e., lower α value, more complexity is present in the dAS value, i.e., more irregular an variable the distribution of values in the parameter space, while the dIP value on the other hand becomes simpler (in the same sense); and vice versa. This fact can be viewed as follows: if Spikes action is strong the membrane potential values will be mostly far from threshold except after spikes, and any way they will present a smaller set of possible membrane potential values since the firing activity is high and we mean frequent so the evolution of the membrane potential value has really few possibilities before being reset have. On the other hand a higher spike activity reflects on a higher number of orbits and periods as seen from the period distribution in figure 2.29 and that led the system produce more different values of the distance to the input signal.

- Simulations with higher Input values, shows the existence of another regime of behavior in the parameter space. This regime has not been yet well characterized nor understood.

3.2 Future work on the BMS-Models

- We would like to extend the results and prove formally some results for an arbitrary size Network with more even active synapses meaning more richer structures.
- We are highly interested also in the synaptic plasticity effects, that is, when the network can make evolve its synapses. So the model parameter α will be function of time and we can look how is its evolution if the network who is summited to an external constant input try to minimize the dIP that is, produce and output as similar as possible to the input signal.
- A more profound numerical an theoretical work focus on the results of Vieville and Kornprobst, to well understand the output or response signal given an specific input on an spiking Neuronal Network.
- Develop of some ideas concerning the spiking NN as a transport phenomena with maybe some specific type of distribution associated to its final states.

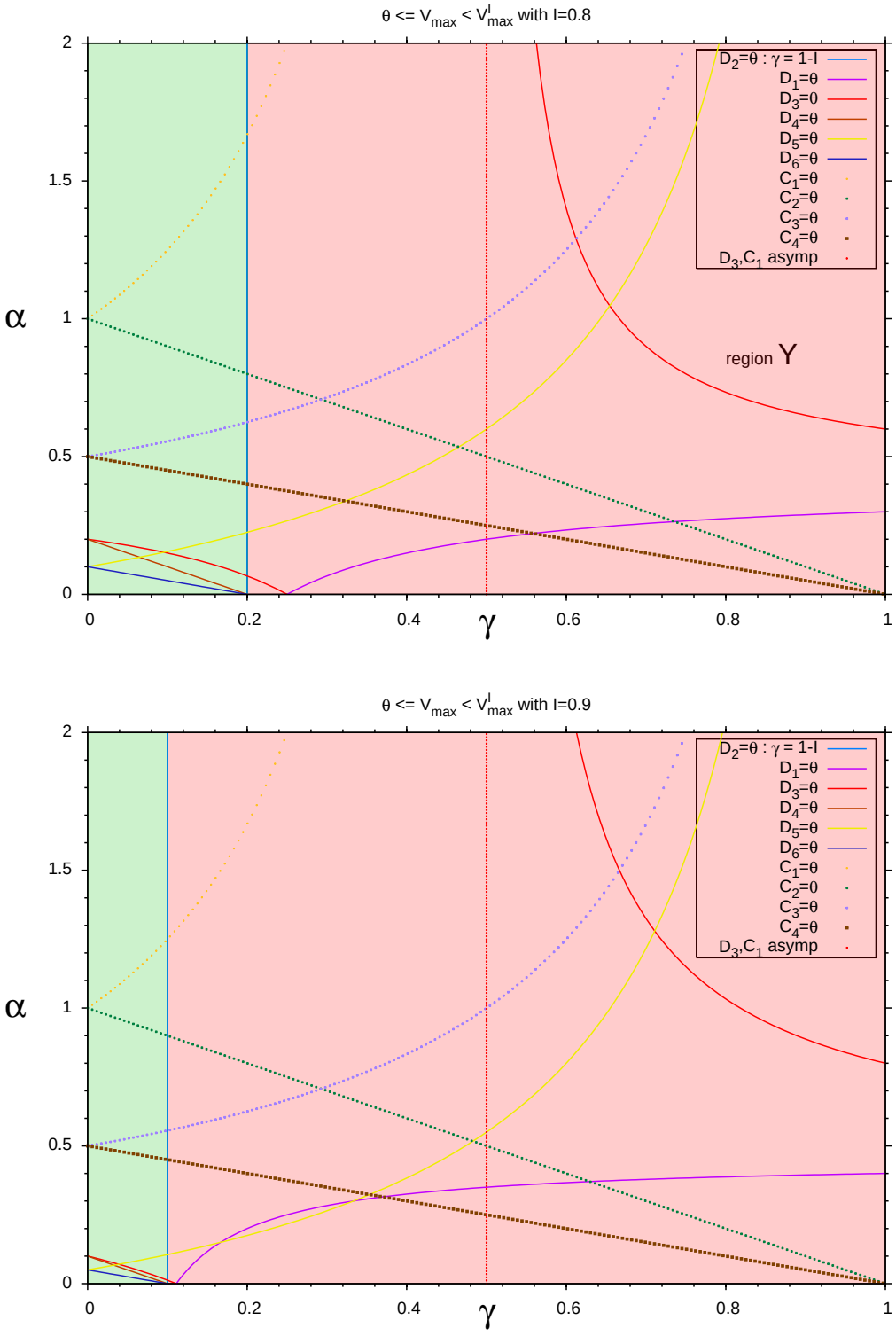


Figure 3.1: continuation

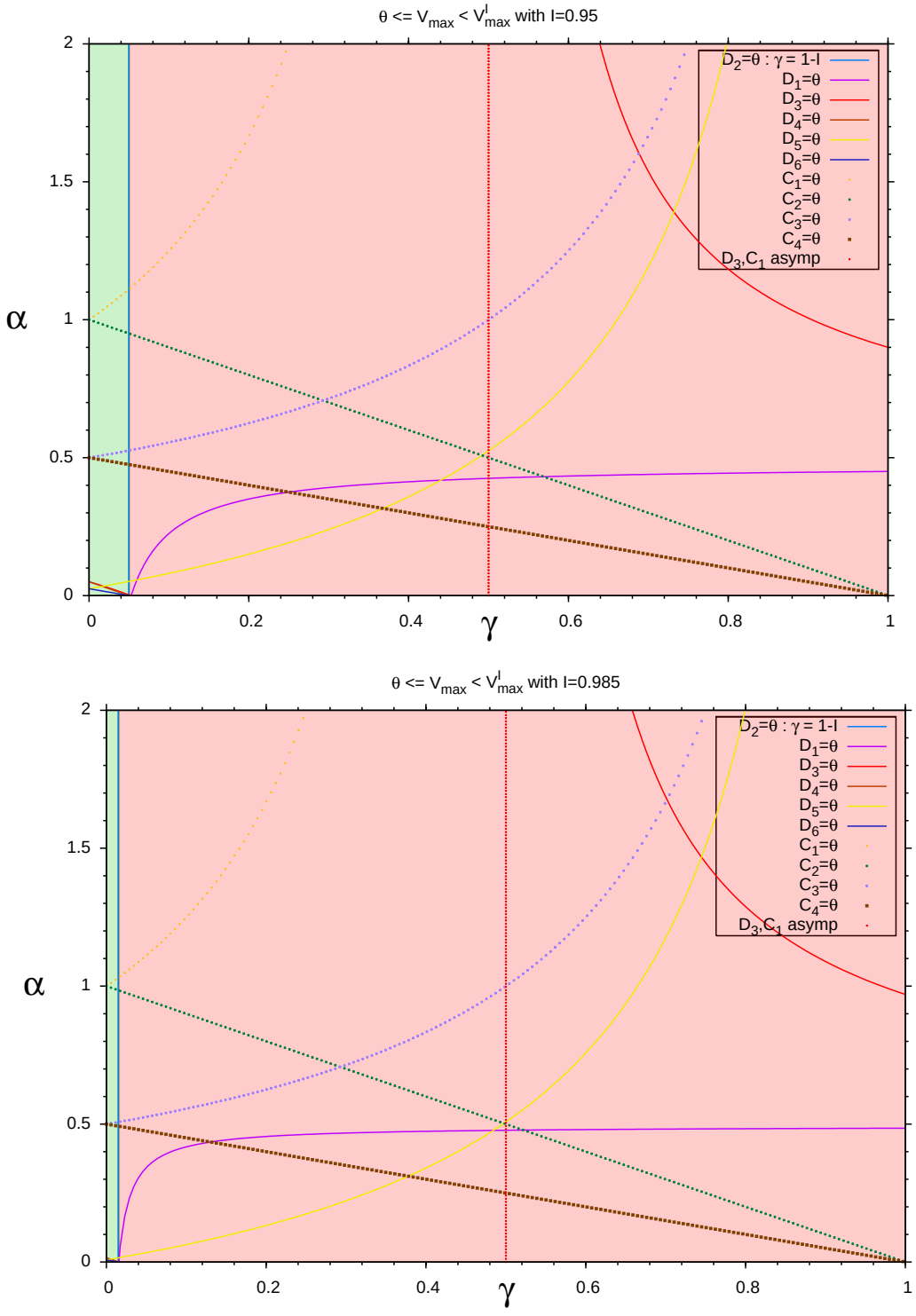


Figure 3.2: End of figure set 2.30.

Acknowledgments

I would like to thanks specially to my tutors Bruno Cessac and Thierry Vieville for its ideas,advises and guidance but also for make me feel active part of their research work. Additionally, I thanks to much to all the ODYSSEE Projet research team at the INRIA for hosting me in this great experience of doing scientific research. They have taught me a lot from its academic formation and research work but also from themselves. I really hope have the opportunity in the future of work with all people mentioned here again. Also I thanks the EGIDE program of the French Government that has provide me the financial support during the research period.

BIBLIOGRAPHY

[1] PEROTTA, R. P. *AN INTRODUCTION TO NEURONAL NETWORKS MODELLING* Saclay. p 300.

[2] CESSAC B. et SAMUELIDES M. *From Neuron to Neuronal Dynamics* To appear in EPJ special Topics.

[3] CESSAC B. *A discrete time Neuronal network with spiking neurons. Rigorous results on the spontaneous dynamics*. 2006 To appear in J. Math Biol.

[4] VIEVILLE T. et KORNPROBST P. *MOdeling Cortical Maps with Feedbacks.*

[5] DAYAN P. et ABBOTT L.F*Theorethical Neuroscience:Mathemathical Computational and Mathematical Modeling of Neuronal Systems*

[6] GERSTNER Wulfram and KISTLER Werner.*Spiking Neuronal Models.* Cambridge, 2002. 480p.

[7] ASH, Robert B. *Basic Probability Theory.* New York : John Wiley & Sons, 1970. 337p.

The principal program

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <float.h>
#include <time.h>
#include "fnjc-4.c"
int main(void)
{
  int M[4],NCI[3][10],transit[3][10];
  double In[4];
  int i,k,qmax,t,tau,input,ASper,IPper,l,per,i0;
  double  *V,*Vp,*Vin,Vmin,Vmax,*V0as,*V0ip,*Vtrans,*VevolAS,*VevolIP; /
  double *PD,*PD0;
  FILE *fdp, *fhp;
  char chaineatt[100],chainetraj[100], comportement[100],chaineactmoy[10
  char chaineatt2[100],chainetraj2[100],chaineactmoy2[100],graphcoding2[
  double et,sum0,distanceip,dist,dist1;
  double distanceas,distA,distA1,Imax;

  M[0]=5;  /*M[1]=6;*/ M[1]=8; M[2]=10;
  In[0]=0.01; In[1]=0.28;  In[2]=0.4;  In[3]=0.8;

  NCI[0][0]=150; NCI[0][1]=200; NCI[0][2]=350; NCI[0][3]=500;
NCI[0][4]=1500; NCI[0][5]=5000;
  NCI[0][6]=5000; NCI[0][7]=5000;
```

```
transit[0][0]=15;  transit[0][1]=20;transit[0][2]=25;transit[0][3]=30;t
transit[0][6]=15;  transit[0][7]=2000;
NCI[1][0]=1000;  NCI[1][1]=1000;  NCI[1][2]=1000;    NCI[1][3]=2000;  NCI[
NCI[1][6]=10000;   NCI[1][7]=10000;NCI[1][8]=10000;   NCI[1][9]=10000;
transit[1][0]=20+10;transit[1][1]=15+7+15;transit[1][2]=20+7+20;transit
transit[1][5]=44+7+50;transit[1][6]=63+7+60;  transit[1][7]=97+7+70;tra
NCI[2][0]=10000;  NCI[2][1]=50000;  NCI[2][2]=20000;    NCI[2][3]=25000;
NCI[2][6]=35000;   NCI[2][7]=100000;
transit[2][0]=19;transit[2][1]=20;transit[2][2]=30;transit[2][3]=40;tra
transit[2][6]=85;transit[2][7]=100;
theta=1.0;   epsilon=0.5;

for(input=1;input<2;input++){
  for(l=1;l<2;l++){
    N=M[l];
    srand48(time(NULL));
    T=20*N;//Duree de la plage temporelle ou est calculee la distance
    I=VALLOC(N,double);
    V=VALLOC(N,double);
    Vp=VALLOC(N,double);
    Vin=VALLOC(N,double);
    VevolAS=VALLOC(N,double);
    VevolIP=VALLOC(N,double);
    PD0=VALLOC(N,double);
    V0ip=VALLOC(N,double);
    V0as=VALLOC(N,double);


    if(input){
      for(i0=1;i0<4;i0++){Imax=In[i0];

        for(k=0;k<N;k++){
          if(k<(int)floor(((double)N/2.0))){I[k]=0.0;}
          else{I[k]=Imax;}
          printf("%lg \t",I[k]);
        }
        sprintf(comportement,"data-28/DP-Input%lg_N%d",I[N-1],N);
        fdp=fopen(comportement,"w");

        sprintf(decharge,"data-28/ProbDec_Input%lg_N%d",I[N-1],N);
        fhp=fopen(decharge,"w");
```

```
            for(r=0;r<450;r++){
            gamm=0.01+(0.002)*r;
               if(gamm<=0.1){NBCI=NCI[l][0]; transitoire=transit[l][0];}
               else if(gamm>0.1 && gamm<=0.2){NBCI=NCI[l][1]; transitoire=t
        else if(gamm>0.2 && gamm<=0.3){NBCI=NCI[l][2]; transitoire=transit
               else if(gamm>0.3 && gamm<=0.4){ NBCI=NCI[l][3]; transitoire=
               else if(gamm>0.4 && gamm<=0.5){NBCI=NCI[l][4]; transitoire=
        else if(gamm>0.5 && gamm<=0.6){ NBCI=NCI[l][5]; transitoire=trans
               else if(gamm>0.6 && gamm<=0.7){ NBCI=NCI[l][6];
transitoire=transit[l][6];}
        else if(gamm>0.7 && gamm<=0.8){ NBCI=NCI[l][7]; transitoire=tran
        else if(gamm>0.8 && gamm<0.9){ NBCI=NCI[l][8]; transitoire=trans
        else if(gamm>=0.9 && gamm<0.93){ NBCI=NCI[l][8]; transitoire=tra
        else{NBCI=NCI[l][9]; transitoire=transit[l][9];}

              qmax=NBCI;
              for(s=0;s<2;s++){
                 alpha=0.3+0.5*s;
                 sprintf(chainetraj,"data-28/1eTrajectoire_Input%lg_alpha%lg
                 sprintf(chaineatt,"data-28/1eTrajMoy_Input%lg_alpha%lg_gamm
                 sprintf(chaineactmoy,"data-28/1eactmoy_Input%lg_alpha%lg_ga
                 sprintf(graphcoding,"data-28/1ecode_Input%lg__alpha%lg_gamm

                 sprintf(chainetraj2,"data-28/2eTrajectoire_Input%lg_alpha%
                 sprintf(chaineatt2,"data-28/2eTrajMoy_Input%lg_alpha%lg_gam
                 sprintf(chaineactmoy2,"data-28/2eactmoy_Input%lg_alpha%lg_g
                 sprintf(graphcoding2,"data-28/2ecode_Input%lg__alpha%lg_gam

                W= creer_reseau_PPV1D(N,alpha);
                Vmax=gamm*theta+2*alpha+Imax;
                Vmin=-2.0*alpha;

                et=0.0;
                sum0=0.0;
                dist=20;
                distA=20;
                tau=0;

                do{ //Cond Init
                        for (i=0;i<N;i++)Vin[i]=Vmin+(Vmax-Vmin)*drand48(
```

```
                    Vtrans=Trans(Vin);
                                per=periode(Vtrans);
                    distanceas=distanceAS(Vtrans,per);
                                distA1=distA;
                                distA=min(distA,distanceas);

                                if(distA1!=distA)for(k=0;k<N;k++

                                PD=ProbDec(Vtrans,per);
                                distanceip=distanceIPD(PD,I);
                                sum0=distanceip + sum0;
                                et=distanceip*distanceip + et;

            dist1=dist;
            dist=min(dist,distanceip);

            if(dist1!=dist)for(k=0;k<N;k++){
              V0ip[k]=Vtrans[k];
              VevolIP[k]=Vin[k];
              PD0[k]=PD[k];
            }
            tau++;

            free(PD);
free(Vtrans);

        }while(tau<qmax);

        sum0=sum0/((double) qmax);
        et=sqrt(fabs(et/((double) qmax)-sum0*sum0));

        ASper=periode(V0as);
        IPper=periode(V0ip);

          if(ASper>N){
                attracteur(chaineatt,chainetraj,graphcoding,Vevol
                attracteur(chaineatt2,chainetraj2,graphcoding2,V
                // ActMoy(chaineactmoy,Vevol,transitoire+ASper+1
```

```
                        //attracteur(chaineatt,chainetraj,graphcoding,V0
                        //ActMoy(chaineactmoy,V0as,ASper);
                    }

                    if(IPper>N){
                        attracteur(chaineatt2,chainetraj2,graphcoding2,V
                        //ActMoy(chaineactmoy,Vevol,IPper+transitoire);
                        //attracteur(chaineatt,chainetraj,graphcoding,V0
                        //ActMoy(chaineactmoy,V0ip,IPper);
                    }

                  fprintf(fdp,"%lg \t %lg \t %lg \t %d \t %lg \t
%lg \t %lg \t %d \t %lg \n", gamm,alpha,distA,ASper,dist,sum0,et,IPper,PI
                  // printf("%lg \t %lg \t %lg \t %d \t %lg \t %lg \t %lg \
%d \t %lg \n", gamm,alpha,distA,ASper,dist,sum0,et,IPper,PD0[0]);

                  fprintf(fhp,"%lg\t%lg\t%lg\t",I[N-1],gamm,alpha);
                  for(k=0;k<N;k++){fprintf(fhp,"%lg \t",PD0[k]);}
                  fprintf(fhp,"\n");

                  for (i=0;i<N;i++) free(W[i]);free(W);
                }//FIN BOUCLE ALPHA

                fprintf(fdp,"\n"); //ISOLIGNES
                printf("N %d,input %d  gamma_ %lg (%d) ciclo finalizado \n",N
                fprintf(fhp,"\n");//ISOLIGNES pour PD0 file
              }//boucles r-gamma
              fclose(fhp);

            }//boucles Inputs
          }
          else{
            sprintf(comportement,"data-28/DP-BMSLap_N%d",N);
            fdp=fopen(comportement,"w");

            for(k=0;k<N;k++)I[k]=0.0;
    for(r=0;r<450;r++){
              gamm=0.01+ 0.002*r;

              if(gamm<=0.1){NBCI=NCI[1][0];transitoire=transit[1][0];}
              else if(gamm>0.1 && gamm<=0.2){NBCI=NCI[1][1]; transitoire=tra
```

```
            else if(gamm>0.2 && gamm<=0.3){NBCI=NCI[1][2]; transitoire=tra
            else if(gamm>0.3 && gamm<=0.4){ NBCI=NCI[1][3]; transitoire=tra
            else if(gamm>0.4 && gamm<=0.5){NBCI=NCI[1][4];  transitoire=tra
            else if(gamm>0.5 && gamm<=0.6){ NBCI=NCI[1][5];  transitoire=t
            else if(gamm>0.6 && gamm<=0.7){ NBCI=NCI[1][6];   transitoire=
            else if(gamm>0.7 && gamm<=0.8){ NBCI=NCI[1][7];   transitoire=
            else if(gamm>0.8 && gamm<=0.9){ NBCI=NCI[1][8];   transitoire=
            else if(gamm>=0.9 && gamm<=0.93){ NBCI=NCI[1][8];
transitoire=transit[1][8]+120;}
            else if(gamm>0.93 && gamm<=0.96){ NBCI=NCI[1][9];
transitoire=transit[1][9]+200;}
            else{NBCI=NCI[1][9]; transitoire=transit[1][9]+1000;}

          qmax=NBCI;

            for(s=0;s<2;s++){
                    alpha=0.3+0.5*s;
              sprintf(chainetraj,"data-24/eTrajectoire_BMSLaplacien_alpha%l
              sprintf(chaineatt,"data-24/eTrajMoy_BMSLaplacien_alpha%lg_gam
              sprintf(chaineactmoy,"data-24/eactmoy_BMSLaplacien_alpha%lg_g
              sprintf(graphcoding,"data-24/ecode_BMSLaplacien_alpha%lg_gamm

             W= creer_reseau_PPV1D(N,alpha);
              Vmax=min(gamm*theta+2*alpha,(2*alpha)/(1.0-gamm));
              Vmin=-2.0*alpha;

              et=0.0;
              sum0=0.0;

              dist=20;
              distA=20;
              tau=0;

              do{ //Cond Init
                for (i=0;i<N;i++)Vin[i]=Vmin+(Vmax-Vmin)*drand48();

                Vtrans=Trans(Vin);
                per=periode(Vtrans);

                distanceas=distanceAS(Vtrans,per);
                distA1=distA;
```

```
            distA=min(distA,distanceas);

            if(distA1!=distA)for(k=0;k<N;k++){
              V0as[k]=Vtrans[k];
              VevolAS[k]=Vin[k];
              //  PD0[k]=PD[k];
            }



          tau++;

          free(Vtrans);

        }while(tau<qmax);


        //  for(k=0;k<N;k++)  printf("%lg\t",V0as[k]);
        ASper=periode(V0as);
        //printf("\n%d\n",ASper);




        //attracteur(chaineatt,chainetraj,graphcoding,VevolAS,transito
        fprintf(fdp,"%lg \t %lg \t %lg \t %d  \n", gamm,alpha,distA,AS
        //printf("%lg \t %lg \t  %lg \t %d \n", gamm,alpha,distA,ASper




        for (i=0;i<N;i++) free(W[i]);free(W);
      }//FIN BOUCLE ALPHA

      fprintf(fdp,"\n"); //ISOLIGNES

      //  fprintf(fhp,"\n");//ISOLIGNES pour PD0 file
```

```
        printf("N %d,input %d  gamma_ %lg (%d) ciclo finalizado \n",N,inp


      }//boucles r-gamma


    }


    free(V);
    free(Vp);
    free(I);
    free(Vin);
    free(VevolAS);
    free(VevolIP);

    free(V0as);
    free(V0ip);

    free(PD0);

    fclose(fdp);
  }//boucles N
}//input
return EXIT_SUCCESS;
}
```

The Subroutines program

```
#include "Param.h"
#include "Fonctions.h"

/********** Macros ***************/

#define signe(r) (((r) < 0.0) ? -1.0 : 1.0)
#define max(a,b) (((a)>(b)) ? (a) : (b))
#define min(a,b) (((a)<=(b)) ? (a) : (b))

#define VALLOC(n,type) ((type*) mvalloc((n),sizeof(type)))
```

```
void *mvalloc(size_t n,size_t taille)

{
    void *p;

    p=calloc(n,taille);

    if (p == NULL)
      {
        printf("Allocation impossible (%d octets) \n", n * taille);
        exit(1);
      }
    else
      {
        //printf("Adresse pointeur alloue\t %X \n",p);
        return p;
      }

}
/** creation d'un reseau Laplacien 1D

nombre de sites N
dimension de la matrice des poids N^2
Intensite alpha
****/


double** creer_reseau_PPV1D(int N,double alpha)

{
  double **J;
  int k;

  J=VALLOC(N,double*);

  for (k=0;k<N;k++)
    {
      J[k]=VALLOC(N,double);
      J[k][(N+k-1)%N]=alpha;//Gauche
      J[k][(k+1)%N]=alpha;//Droite
      J[k][k]=-2*alpha;
```

```
    }

  //   ecrimat(J,N,N);
  return(J);
}



/****** Un pas de temps de la dynamique ********/

void F(double *Vv,double *Vvp)
{
  register int i,j;
  double sum;

  for (i=0;i<N;i++)
    {
      Vvp[i]=I[i];
      sum=0.0;
      for (j=0;j<N;j++) if (Vv[j]>=theta) sum+=W[i][j];
      if (Vv[i]>=theta) Vvp[i]+=sum;
      else Vvp[i]+=gamm*Vv[i]+sum;
    }


}


/*************************************************/
/***************TRANSITOIRE************/

double* Trans(double* V)
{
  register int k,j;
  int t,test,i;
  double *Vv,*Vvp,*temp;


  Vv=VALLOC(N,double);
  Vvp=VALLOC(N,double);
```

```
  for (i=0;i<N;i++)Vv[i]=V[i];
  t=0;test=0;
  //   for (i=0;i<N;i++)printf("%lg\t",Vv[i]);
  // printf("\n trans ha leido este vin \n");

  do{
        F(Vv,Vvp);



    temp=Vvp;
    Vvp=Vv;
    Vv=temp;



   t++;
  }while(t<transitoire);

  //for (i=0;i<N;i++)printf("%lg\t",Vv[i]);
  // printf("\n trans transmite este Vtrans \n");
  free(Vvp);
 return Vv;

 //free(Vv);;

}
/***************prob Decharge************/

double* ProbDec(double* Vtrans, int tau)
{
  register int k,j;
  int t,test,i;
  double *summ,*Vv,*Vvp,*temp;

  summ=VALLOC(N,double);
  Vv=VALLOC(N,double);
  Vvp=VALLOC(N,double);

  for(j=0;j<N;j++){
    Vv[j]=Vtrans[j];
```

```
      summ[j]=0.0;
                        }
   t=0;

   do{
          F(Vv,Vvp);


          temp=Vvp;
          Vvp=Vv;
          Vv=temp;
          for (k=0;k<N;k++){
                              if(Vv[k]>=theta){summ[k]=1.0/((double) tau)+sum

                              }




    t++;
   }while(t<tau);



   free(Vv);free(Vvp);

  return summ ;

  //free(summ);
}

/************distanceIPD*****************/
double distanceIPD(double *PD,double *I)

{ register int k;
   double sum1;
   sum1=0.0;

   for(k=0;k<N;k++){sum1=sum1+(PD[k]-I[k])*(PD[k]-I[k]);
   }
```

```
  return sum1;

}



/**** Representation attracteur ***/

void attracteur(char chainemoy[],char chainetraj[],char graphcoding[],do
{
  double m,mp,mmoy,mmax;
  int t=0;
  register int i,k;
  double *Vv,*Vvp,*temp;
  int *code1;
  // double *xmoy;
  FILE *fpmoy,*fptraj,*fpcode;




  printf("Representation de l'attracteur\n");

  // xmoy=VALLOC(N,double);
  Vv=VALLOC(N,double);Vvp=VALLOC(N,double);
  code1=VALLOC(N,int);

  for (i=0;i<N;i++) Vv[i]=V[i];

  mmoy=0.0; mmax=0.0;

  //srand48(time(NULL));
//   for (k=0;k<N;k++)  ((drand48() > 0.5)  ? (V[k]=theta*(1+drand48())):
//   for (k=0;k<N;k++) V[k]=2.0*drand48();


  /******* Trajectoire ****/

   m=0;
   for (k=0;k<N;k++) m+=Vvp[k];
   m=m/(double)N;
```

```
t=0;

 fpmoy=fopen(chainemoy,"w");
 fptraj=fopen(chainetraj,"w");
 fpcode=fopen(graphcoding,"w");
  do
     {

       mp=0;
       F(Vv,Vvp);

            fprintf(fptraj,"%d\t",t);

            for (k=0;k<N;k++){
                              if(Vv[k]>=theta){code1[k]=1;}
                              else{code1[k]=0;}
                              fprintf(fpcode,"%d \t %d \t %d \n",t,k,code1
                              fprintf(fptraj,"%lg\t",Vv[k]);
                             mp+=Vv[k];


            }

        fprintf(fptraj,"\n");
        fprintf(fpcode,"\n");
       mp=mp/(double)N;



      fprintf(fpmoy,"%d \t %lg \t %lg \n",t,m,mp);
     m=mp;

     // for (k=0;k<N;k++) xmoy[k]+=(double)(Vv[k]>=theta);

     temp=Vv;
     Vv=Vvp;
     Vvp=temp;


     t++;
```

```
    }    while (t<tau);

  // fprintf(fpmoy,"\n \n # ACTIVITE MOYEN \n #");

      // for (k=0;k<N;k++) fprintf(fpmoy," %lg \t",xmoy[k]/(double) T);
 fprintf(fpmoy," \n ");
 fprintf(fpcode,"\n");
 fprintf(fptraj,"\n");
  fclose(fpmoy);
  fclose(fptraj);
  fclose(fpcode);
  free(Vv); free(Vvp); free(code1);

  //free(xmoy);
}


/**** Activite moyenne ***/

void ActMoy(char chaine[],double* V,int tau)
{
  FILE *fp;
  double *xmoy,*temp,*Vv,*Vvp;
  register int i,k,t;

  printf("Activit  moyenne\n");
  xmoy=VALLOC(N,double);
  Vv=VALLOC(N,double); Vvp=VALLOC(N,double);
  for (i=0;i<N;i++) Vv[i]=V[i];


  /******* Trajectoire ****/


  t=1;


  do
    {
```

```
        F(Vv,Vvp);

        for (k=0;k<N;k++)
          xmoy[k]+=(double)(Vv[k]>=theta);


        temp=Vvp;
        Vvp=Vv;
        Vv=temp;


        t++;

      }
   while (t<tau);

   fp=fopen(chaine,"a");
   for (k=0;k<N;k++) fprintf(fp,"%lg\n",xmoy[k]/(double)tau);
  fprintf(fp,"\n");
  fclose(fp);

   free(xmoy);
   free(Vv); free(Vvp);

}




double distanceAS(double *Vtrans,int time)
{
   int t,i,test;
   double *Vv,*Vvp,*temp;
   register int k;
   double distan,rest,d;
```

```
        Vv=VALLOC(N, double ); Vvp=VALLOC(N, double );

        for ( i =0; i <N; i ++)Vv[ i ]= Vtrans [ i ];

        distan =200.0;

        t =0;

        // for(k=0;k<N;k++) printf("%lg \t", Vtrans [k ]);

             do{

               F(Vv, Vvp );

               temp=Vvp;
               Vvp=Vv;
               Vv=temp;


               for (k=0;k<N;k++){
                  d=distan;
                  rest=fabs(theta -Vv[k]);
                  distan=min(d, rest );
                 }
               // printf("------%lg \t %d ", distan , t );


               t++;
             }
            while (t<time +1);
            // printf("\n");
            free(Vvp); free(Vv);

  return distan;
}


/**** Calcul de Vmax ***/

double VM(double **Jij)
{
```

```
  double Vm=0.0,sum;
  int i,j;

  for (i=0;i<N;i++)
    {
      sum=0.0;
      for (j=0;j<N;j++)
        if (Jij[i][j]>=0.0) sum+=Jij[i][j];
      Vm=max(sum,Vm);
    }

  if(I[N-1]>1E-6)Vm=Vm+I[N-1];
  return Vm/(1.0-gamm);
}


/**** Calcul de Vsup pour lq trajectoire X***/

double VS(double *X)
{
  double Vs=0.0;
  double *Vv=VALLOC(N,double),*Vvp=VALLOC(N,double),*temp;
  int i,t;

  for (i=0;i<N;i++) Vv[i]=X[i];

  t=0;

  //Tracage(chainet,"transitoires --->");

  /*** Calcul des transitoires **/

  do
    {

      F(Vv,Vvp);

      temp=Vvp;
      Vvp=Vv;
      Vv=temp;
```

```
        t++;

      }
   while (t<transitoire);

   //Tracage(chainet,"Fait\n");

   /******* Trajectoire ****/


   t=1;


   do
      {

        for (i=0;i<N;i++) Vs=max(Vs,Vv[i]);

        F(Vv,Vvp);

        temp=Vv;
        Vv=Vvp;
        Vvp=temp;


        t++;

      }
   while (t<T);


   free(Vv); free(Vvp);

   return Vs;
}



/****** Periode d'une trajectoire *****/
```

```
int periode(double* V)
{
  int i,t,j,test,test2,testa,testb;
  double *Vv,*Vvp,*temp,x;

  Vv=VALLOC(N,double);
  Vvp=VALLOC(N,double);

  /*** Calcul periode ***/

  for (i=0;i<N;i++){ Vv[i]=V[i];
                     }
/*
**********test mort neuronel************
  test2=1;
  i=0;
  F(Vv,Vvp);
      temp=Vvp;
      Vvp=Vv;
      Vv=temp;

      while(i<N){
        test2=(fabs(Vv[i]-Vvp[i])<1E-5);
                  i++;
                  }
      if(test2){t=2;
                test=1;
               }
      else{t=1;
           test=0;
      }
*/
  t=1;
  do
    {

      F(Vv,Vvp);
      temp=Vvp;
      Vvp=Vv;
      Vv=temp;
```

```
        i=0;test=1;

        while ((i<N)&&(test))

         {
                x=fabs(Vv[i]-V[i]);

               //testa=(1E-16<x);
               testb=(x<=1E-8);
           test=testb;
           //printf("%lg %d %d \t", x,test,testb);
           i++;
         }

       t++;

    }while ((t<T)&&(!test));

  free(Vv);free(Vvp);

  return t-1;
}


/****** Periode d'une trajectoire *****/

int periodeb(double* V)
{
  int i,t,j,test,test2,testa,testb;
  double *Vv,*Vvp,*temp,x;

  Vv=VALLOC(N,double);
  Vvp=VALLOC(N,double);

  /*** Calcul periode ***/

  for (i=0;i<N;i++){ Vv[i]=V[i];
                     //printf("%20.10f \t",Vv[i]);

                    }
```

```
                        //printf("\n");
/*
**********test mort neuronel************
   test2=1;
   i=0;
   F(Vv,Vvp);
        temp=Vvp;
        Vvp=Vv;
        Vv=temp;

        while(i<N){
           test2=(fabs(Vv[i]-Vvp[i])<1E-5);
                      i++;
                      }
        if(test2){t=2;
                    test=1;
                  }
        else{t=1;
              test=0;
        }
*/
   t=1;

   do
     {

        F(Vv,Vvp);
        temp=Vvp;
        Vvp=Vv;
        Vv=temp;


         i=0;test=1;

         while  ((i<N)&&(test))

          {
                  x=fabs(Vv[i]-V[i]);

                 //testa=(1E-16<x);
                 testb=(x<=1E-8);
```

```
            test=testb;
                if(!testb){ printf("%20.10f\t",Vv[i]);
            printf("%d %lg %d %d \t", i, x,test,testb);}
            i++;
          }

        t++;

      }while ((t<T)&&(!test));

   free(Vv);free(Vvp);

   return t-1;
}
```

The Funtions header

```
typedef struct MAILLON
{
//   double *nom;
   double nom;
   int count;
   struct MAILLON *prec;
   struct MAILLON *suiv;
} maillon;
/*** Fonctions ***/
double** creer_reseau_fc(int,double, double);
double** creer_reseau_PPV1D(int,double);
void F(double *V,double *Vp);
void attracteur(char chainemoy[],char chainetraj[],char graphcoding[],do
void ActMoy(char chaine[],double* V,int per);
double distanceAS(double *V,int tau);
double VM(double **Jij);
double* ProbDec(double *V,int tau);
double distanceIPD(double *V,double *I);
```