Testing Einstein's Formula on Brownian Motion Using Membrane Computing

Manuel A. Gálvez-Santisteban, Miguel A. Gutiérrez-Naranjo,
Daniel Ramírez-Martínez, Elena Rivero-Gil

Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla
Avda. Reina Mercedes s/n, 41012, Sevilla, Spain
E-mails: mangalsan@alum.us.es, magutier@us.es,
thebluebishop@gmail.com, elen.rg@gmail.com

Summary. Brownian motion refers to erratic movements of small particles of solid matter suspended in a fluid and it is the basis of the development of many fractals found in Nature. In this paper we use the Membrane Computing model of P systems with membrane creation and the software tool JPLANT [15] in order to check the Einstein's theory on the Mean Square Displacement of Brownian motion.

1 Introduction

In [5], a first study was presented by showing the relation between fractals and P systems. On the one hand, a *fractal* is a shape made of parts similar to the whole in some way. This self-similarity occurs over an infinite range of scales in pure mathematical structures but over a finite range in many natural objects such as clouds, coastlines or snowflakes. In many plants and also organs of animals, this has led to fractal branching structures. For example, in a tree the branching structure allows the capture of a maximum amount of sun light by the leaves; the blood vessel system in a lung is similarly branched so that the maximum amount of oxygen can be assimilated (see [11]).

On the other hand, as pointed out in [5], cell-like P systems have several properties which make them suitable for the study of fractals:

- P systems can be considered as a structure of nested processors placed in a tree-structure, i.e., we can consider computations on many scales.
- If we consider P systems where membranes can be dissolved, divided or created, we usually obtain a geometrical shape too irregular to be described in traditional geometrical language, both locally and globally.

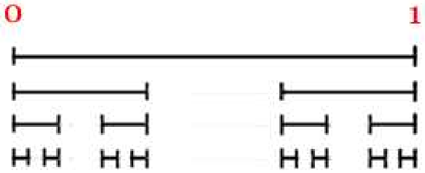


Fig. 1. First steps for the middle third Cantor set

- Computations in P systems are obtained by the application of a fixed (often only a few) set of rules. The application of these rules allows to obtain a configuration C_{n+1} from C_n.
- The computation of a P system is discrete, i.e., it is a process performed *step* by *step*.

In that paper a pair of examples were provided based on the cell-like model of P systems with membrane creation: the middle third Cantor set [2] (see Fig. 1) and the Koch curve [7, 8]. If we put together three Koch curves we have the fractal known as *Koch Snowflake* (see Fig. 2).

Self-similar fractals as Koch curve differs from natural fractals in one significant aspect. They are *exactly* self similar, and they cannot be considered as *realistic* models of natural fractals. In [5], *statistically self-similar* objects were also considered. The property that objects can look statistically similar while at the same time different in detail at different length scales, is the central feature of fractals in Nature. Randomizing a deterministic classical fractal is the first approach generating a realistic natural shape. Figure 3 shows a random Koch snowflake. Note that this fractal represents a *realistic* shape of a fractal from Nature.

In this paper we follow this research line and we present a study on the Brownian motion. Brownian motion refers to the erratic movements of small particles of solid matter suspended in a fluid and it is the basis of many random fractals found in Nature. The study of fractals and P systems needs, in the same way that other studies with P systems that involve a large number of configurations, the appropriate software in order to do the corresponding simulations. Our study considers a large amount of branches in the computational tree of a P system and for that we have used JPLANT, which is a software tool[1] that computes the first configurations of a computation and draws the corresponding graphical representation. This graphical representation provides the necessary information for carrying out our experiments.

The paper is organized as follows. First we recall the stochastic restricted P system model, its graphical representation and the software tool JPLANT used for its representation. In section 3, a brief introduction to the Brownian motion

[1] A detailed description of JPLANT can be found in [15].

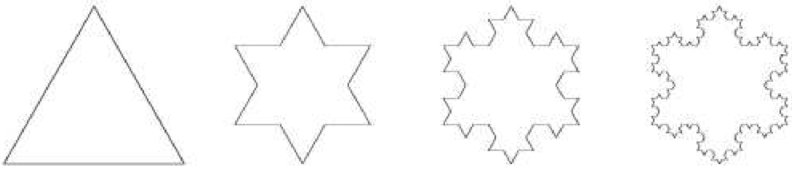


Fig. 2. First steps for the Koch Snowflake

together with our experiments are presented. Some conclusions and lines for future research are given in the last section.

2 P Systems with Membrane Creation

In this paper we will consider stochastic restricted P systems with membrane creation. This P system model has already been used for the study of graphics with P systems (see [16]). This model follows a research line in Membrane Computing that incorporates randomness into membrane systems (see [1, 10, 13] and the references therein). In this model, to pass from a configuration of the system to the next one we apply to every object present in the configuration a rule chosen at random, according to given probabilities, among all the rules whose left–hand side coincides with the object[2]. The second ingredient in this model is membrane creation, which was first introduced in [6, 9]. However, our needs are far simpler than the models found in the literature. In this *restricted* model we only consider object-evolution rules and creation rules.

The non-determinism is one of the main features of P systems and the possibility of reaching different configurations leads us to consider different graphical representations in the evolution of a P system.

A restricted P system with membrane creation is a tuple $\Pi = (O, \mu, w_1, \ldots, w_m, R)$ where:

1. O is the alphabet of *objects*. There exist two distinguished objects, F and W that always belong to the alphabet.
2. μ is the initial *membrane structure*, consisting of a hierarchical structure of m membranes (all of them with the same label; for the sake of simplicity we omit the label).
3. $w_1, \ldots, w_m$ are the multisets of objects initially placed in the m regions delimited by the membranes of μ.
4. R is a finite set of *evolution rules* associated with every membrane, which can be of the two following kinds:
   a) $a \xrightarrow{p} v$, where $a \in O$, v is a multiset over O, and $0 \leq p \leq 1$ is a real number representing the probability of the rule. This rule replaces an object a present in a membrane of μ by the multiset of objects v.

[2] This idea was also presented in [16].

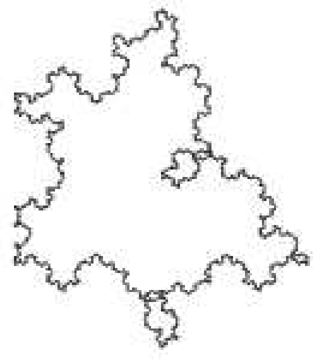


Fig. 3. Random Koch Snowflake

b) $a \xrightarrow{p} [v]$, where $a \in O$, v is a multiset over O, and $0 \leq p \leq 1$ is a real number representing the probability of the rule. This rule replaces an object a present in a membrane of μ by a new membrane with the same label and containing the multiset of objects v.

The addition of the probabilities of the rules with the same left-hand side must be one. If there is only one rule for a given left-hand side, then its probability must be one and, for the sake of simplicity, we omit it.

A membrane structure (extending the membrane structure μ) together with the objects contained in the regions defined by its membranes constitute a configuration of the system. A computation step is performed applying to a configuration the evolution rules of the system in a non-deterministic maximally parallel way.

A rule in a region is applied if and only if the object occurring in its left–hand side is available in that region; this object is then consumed and the objects indicated in the right–hand side of the rule are created inside the membrane. The rules are applied in all the membranes simultaneously, and all the objects in them that can trigger a rule must do it. When there are several possibilities to choose the evolution rules to apply, non-determinism takes place.

2.1 Graphical Representation

In this section we show how to use, through a suitable graphical representation, restricted P systems with membrane creation to model branching structures. The key point of the representation relies on the fact that a membrane structure is a *rooted tree of membranes*, whose root is the skin membrane and whose leaves are the elementary membranes. It seems therefore a perfect frame to encode the branching structure.

Let us suppose that the alphabet O of objects contains the objects F and W and let us fix the lengths l and w.

A simple model to graphically represent a membrane structure is to make a depth-first search of it, drawing, for each membrane containing the object F, a segment of length $m \times l$, where m is the multiplicity of F. If the number of copies of F in a membrane increases along the computation, the graphical interpretation is that the corresponding segment is lengthening. Analogously, the multiplicity of the symbol W specify the width of the segments to be drawn as follows: if the number of objects W present in a membrane is n, then the segment corresponding to this membrane must be drawn with width $n \times w$.

Each segment is drawn rotated with respect to the segment corresponding to its parent membrane. In order to determine the rotation angle we need to fix a third parameter δ. Such angle δ together with the length l and the width w will determine the picture of the P system.

In order to compute the rotation angle of a segment with respect to its parent membrane we consider two new objects that can appear in the alphabet: $+$ and $-$. The rotation angle will be $n \times \delta$, where n is the multiplicity of objects "+" minus the multiplicity of objects "−" in the membrane. That is, each object "+" means that the rotation angle is increased by δ whereas each object "−" means that it is decreased by δ.

Inside the membranes other objects can appear that do not have geometrical interpretation. They are related to the development of the graph in time.

For example, let us consider Π_2 the following restricted P system with membrane creation:

- The alphabet of objects is $O = \{F, W, B_l, B_s, B_r, L, L_1, E\}$.
- The initial membrane structure together with the initial multiset of objects is $[F^2\, W\, B_l\, B_s\, L_1\, E]$.
- The rules are:

$$\begin{array}{ll}
B_l \xrightarrow{1/2} [+\, F\, W\, B_l\, B_s\, L\, E] & L \ \rightarrow L\, F \\
B_l \xrightarrow{1/2} [-\, F\, W\, B_l\, B_s\, L\, E] & L_1 \rightarrow L_1\, F^2 \\
B_r \xrightarrow{1/2} [+\, F\, W\, B_l\, B_s\, L\, E] & E \ \rightarrow E\, W \\
B_r \xrightarrow{1/2} [-\, F\, W\, B_l\, B_s\, L\, E] & B_s \rightarrow [F\, W\, B_l\, B_r\, L_1\, E]
\end{array}$$

There exist two rules for the evolution of the object B_l and two possibilities for the evolution of the object B_r. The probability for each choice is $1/2$. Notice that we do not make explicit the probability of the rule when this is one.

Figure 4 shows four different configurations after the second step of this P system with the angle $\delta = 15$.

2.2 Software

As usual, the hand-made simulation of the evolution of a P system is a heavy task. In this paper we use a new software tool called JPLANT[3]. It computes

[3] A detailed description with examples can be found in [15].

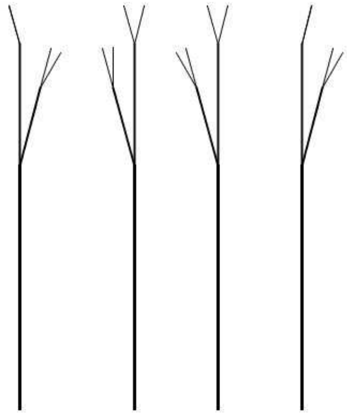


Fig. 4. Four configurations after the second step

the first configurations of a computation of a stochastic restricted P system with membrane creation and draws the corresponding graphical representation of the configurations of such computation.

The program has been written in Java and it has a nice intuitive user-friendly graphical interface. The output is a picture with a set of connected segments drawn according to the rules described in Section 2. For each new configuration, a new picture is drawn, so the output of this tool is a sequence of pictures which can be saved in several computer graphic formats.

The graphical representation of one configuration is not unique. It depends on the parameters l, w and δ which determine the length and width of the segments as well as the rotation angle with respect to the segment corresponding to the parent membrane. These parameters must be also provided by the user and with the initial configuration and the rules, they are the input of the tool.

3 Brownian motion

Brownian motion refers to erratic movements of small particles of solid matter suspended in a liquid. These movements can only be seen under microscope. After the discovery of such movement of pollen it was believed that the cause of the motion was biological in nature. However, about 1828, the botanist Robert Brown realized that a physical explanation, rather than the biological one, was correct. The effect is due to the influence of very light collisions with the surrounding molecules. The standard theory of Brownian motion due to Einstein, Smoluchowski, Langevin, Fokker and Planck is based on the model where a particle moves in a dense medium which generates friction and random collisions.

In 1905 Einstein published a mathematical study of this motion, which eventually led to Perrin's Nobel prize-winning calculation of Avogadro's number. A rigorous probabilistic model of Brownian motion was proposed by Wiener in [18]. He constructed a process which exhibits random behavior very similar to that of Brownian motion. The theoretical problems connected with Brownian motion have many interesting applications in different fields, such as in the theory of sound [14], in physical chemistry [4] and biophysics [17].

In this paper we will consider the special case in which the particle moves a constant distance in each time unit (constant speed) and after each time unit the particle randomly chooses a new direction. The question is to know if we can make any prediction about the total displacement after n steps.

Instead of asking for the total expected displacement, i.e., the displacement of a particle averaged over many samples, the specialized literature focuses on the average of the square of the displacements, the *mean square displacement*. In 1905, Einstein showed that the mean square displacement is *proportional* to time[4]. The factor of proportionality depends on the speed, the step length and the dimension of the space. This is the fundamental property of Brownian motion, verified experimentally in 1908 by the French physicist Jean Perrin (see [12]).

Next, we shows the result of our experiments in order to check the fundamental property of the Brownian motion. We use the probabilistic P system

`Initial configuration:` $[F\,W\,H]$

`Rules:`
$$\begin{aligned}
H &\xrightarrow{1/24} [-\,F\,W\,H]\\
H &\xrightarrow{1/24} [-^2\,F\,W\,H]\\
H &\xrightarrow{1/24} [-^3\,F\,W\,H]\\
&\qquad\cdots\\
H &\xrightarrow{1/24} [-^{23}\,F\,W\,H]\\
H &\xrightarrow{1/24} [F\,W\,H]
\end{aligned}$$

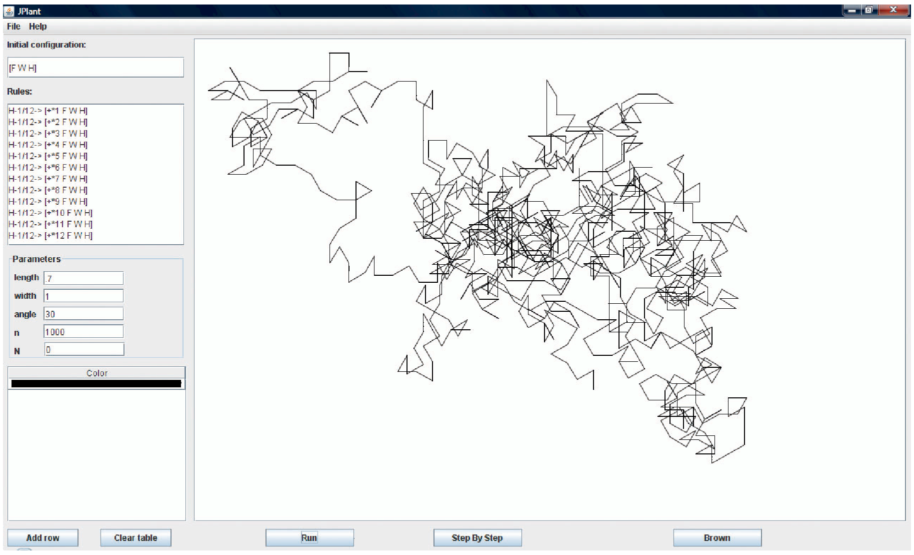
where all the segments have the same length and each new step can a deviation of $n \times 15$ degrees, where n is non-deterministically chosen in $\{0, 1, 2, \ldots, 23\}$.

Since JPLANT is able to simulate discrete Brownian trajectories, we can try to link classical applications of such movements with this new modeling software.

3.1 The experiment

A particle dropped into a fluid describes a Brownian trajectory. Because of the impacts along its path with other particles, several different routes can be traced for this single traveler. For each impact, the trajectory of this moving particle changes its direction. If we assume that no energy is lost due to the impacts,

[4] See [3] for details.



Initial configuration: [F W H]

Rules:

```
H-1/12-> [+*1 F W H]
H-1/12-> [+*2 F W H]
H-1/12-> [+*3 F W H]
H-1/12-> [+*4 F W H]
H-1/12-> [+*5 F W H]
H-1/12-> [+*6 F W H]
H-1/12-> [+*7 F W H]
H-1/12-> [+*8 F W H]
H-1/12-> [+*9 F W H]
H-1/12-> [+*10 F W H]
H-1/12-> [+*11 F W H]
H-1/12-> [+*12 F W H]
```

Parameters

length .7
width 1
angle 30
n 1000
N 0

Fig. 5. Snapshot from JPLANT showing a Brownian trajectory.

the set of possible trajectories can be modeled by JPLANT, just by giving some interpretation to the parameters.

This model of trajectory is quite common in physical and biological systems; from solid state electronics to cell membrane dynamics, even in stochastic signal processes there are many examples of Brownian motions describing several noisy behaviors.

When not only one particle but many of them (a gas cloud, for example) are moving into a fluid, this is called a diffusion process. Depending on the conditions and materials, these processes have different behaviors. The same happens when a black ink drop falls into a glass of liquid; depending on the densities, the shapes of the molecules, the viscosities, the temperature, etc., the black cloud will spread faster or slower. In this way, a diffusion process is the result of an overlapping of many Brownian motions evolving in a parallel way.

Albert Einstein studied Brownian motions and extracted some essential mathematical properties from them. For example, he showed that the mean square of the distance traveled by a particle is proportional to the elapsed time. Using the square of the distances instead of the distances themselves is a key point. The mean of the distances does not give any information because of the uniform distribution of the possible orientations for each step in the path. If we consider a one-dimensional Brownian motion, the expected value for the position is the origin (50% for positive step and 50% for negative step). Nonetheless we have a positive magnitude for each step with squared distances, which admits some parameter definitions. Einstein proposed the following expected value for the Mean Square displacement (MSD, for short):

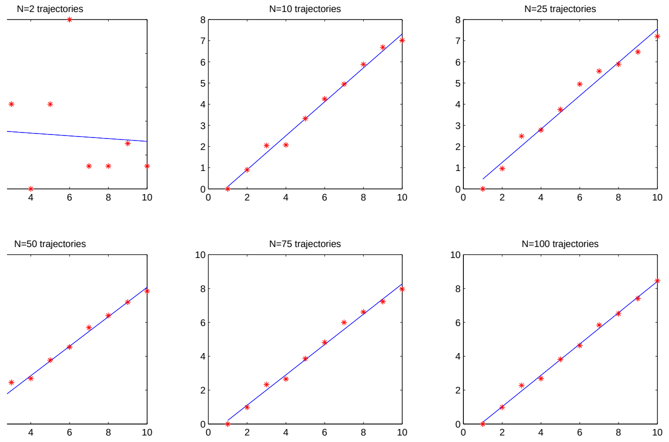


Fig. 6. When N increases, the fitting gets better.

$$MSD =< s^2 >= kDt + C \tag{1}$$

This expression can be calculated from a set of Brownian particles which are diffusing into a fluid. When the MSD data fits into a linear function, the slope is a measurement of how fast the particles are expanding into the fluid. The parameter k is a constant which depends on how many dimensions are considered (for three-dimensional diffusions $k = 6$, for two-dimensional diffusions $k = 4$). The parameter C is just an offset adjustment needed for the regression and D is the diffusion rate.

After a statistical analysis of the data provided by JPLANT, it is possible to model problems involving such diffusion processes. We propose two tasks to be accomplished by Brownian data from JPLANT:

- Check the Einstein's formula and its convergence. Einstein claimed that if the number of experiments grows, the linearity gets stronger, so the absolute error between raw data and the linear regression must tend to zero.
- If the probability of collision increases, that means that for a fixed time of diffusion, the ability to diffuse must decrease. As far as the diffusion rate measures this, the coefficient should get lower as probability increases.

Einstein's relation for the Mean Square Displacement claims that it is directly proportional to the elapsed time. Thus, if the number of observed trajectories under same conditions increases, the linear relation gets stronger, so the relation (1) fits into the data in a better way.

In order to check Einstein's claim we designed the following experiment:

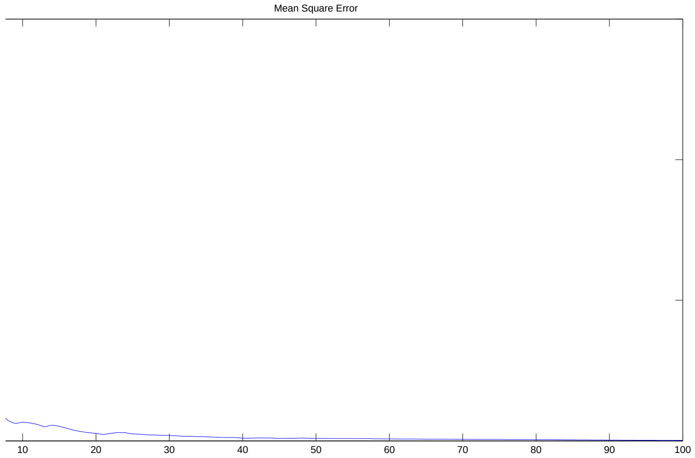


Fig. 7. MSE vs. number of trajectories

- We use JPLANT in order to get the data corresponding to one hundred bidimensional trajectories of Brownian particles.
- For each N with $1 \leq N \leq 100$, we calculate the MSD of the corresponding trajectories and its linear regression.
- In order to consider how the MSD fits to the regression line we calculate the Mean Squared Error.
- The experiments show that when the number of trajectories N considered increases, then the Mean Squared Error tends to zero as Einstein predicted.

Figure 6 illustrates the results of the experiment. In the first frame, only two trajectories are considered. It is easy to check that the points do not fit into the regression line. In the second frame, 10 trajectories are considered and the MSD fits better into the regression line. The following four frames shows the adjustment to the regression line for 25, 50, 75 and 100 trajectories so the data cloud gets tighter to the line as N gets higher.

Figure 7 shows the relation between the number of experiments and the Mean Squared Error. It is clear that when the number of trajectories increases, the MSE tends to zero.

4 Conclusions and Future Work.

In this paper we have used JPLANT as a Brownian simulation tool, testing some Einstein's results and generating new possible paths of study, starting from the idea of a partial recreation of a real experiment. The same methodology, extended appropriately, could be applied to other biological processes or electronic models.

This way, we provide a new application for membrane computing, being useful to model and, maybe, extend classical ways of simulation for such problems.

As a goal for future simulations, we propose the modeling of real experiments maybe by extending the P system model with new types of rules that capture the dynamics of the real experiments: division, cooperation, dissolution, ... In the same way, a deeper study of the use of probabilities in Membrane Computing can be useful in order to model experiments from the real world. Thermic noise or biological membranes are good candidates to be simulated in the immediate future because of the extensive bibliography and the practical use of these concepts.

Acknowledgement

The authors acknowledge the support of the project TIN2006-13425 of the Ministerio de Educación y Ciencia of Spain, cofinanced by FEDER funds, and the support of the project of excellence TIC-581 of the Junta de Andalucía.

References

1. I.I. Ardelean, M. Cavaliere: Modelling Biological Processes by Using a Probabilistic P System Software. Natural Computing, 2(2), (2003), 173–197.
2. G. Cantor: *Über unendliche, lineare Punktmannigfaltigkeiten V*. Mathematische Annalen, 21, (1883), 545–591.
3. W. Ebeling: Non Linear Brownian Motion – Mean Square Displacement. Condensed Matter Physics, 7, 539, (2004), 1–12.
4. P.-G. de Gennes: Wetting: Statics and Dynamics. Reviews of Modern Physics 57, (1985), 827–863.
5. M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez: Fractals and P Systems. In Graciani–Díaz, C., Păun, Gh., Romero–Jiménez, A., Sancho–Caparrini, F. (eds.): Proceedings of the Fourth Brainstorming Week on Membrane Computing, Vol. II. Fénix Editora, Sevilla, (2006), 65–86.
6. M. Ito, C. Martín–Vide, Gh. Păun: A Characterization of Parikh Sets of ET0L Languages in Terms of P Systems. In Ito, M., Păun, Gh., Yu, S. (eds.): Words, Semigroups, and Transducers. World Scientific, (2001), 239–254.
7. H. von Koch. Sur one courbe continue sans tangente, obtenue par une construction géometrique élémentaire. Arkiv för Matematik 1, (1904), 681–704.
8. H. von Koch. Une méthode géometrique élémentaire pour l'étude de certaines questions de la théorie des courbes planes. Acta Mathematica 30, (1906), 145–174.
9. M. Madhu, K. Krithivasan: P Systems with Membrane Creation: Universality and Efficiency. In Margenstern, M., Rogozhin, Y. (eds.): Proceedings of the Third International Conference on Universal Machines and Computations. Lecture Notes in Computer Science, 2055, (2001), 276–287.
10. A. Obtulowicz, Gh. Păun: (In Search of) Probabilistic P Systems. Biosystems 70(2), (2003), 107–121.
11. H.-O. Peitgen, H. Jürgens, D. Saupe: Fractals for the Classroom. Springer-Verlag (1992).

12. H.-O. Peitgen, H. Jürgens, D. Saupe: Chaos and Fractals. New Frontiers of Scisnce, 2nd Ed. Springer, (2004).
13. D. Pescini, D. Besozzi, G. Mauri, C. Zandron: Dynamical Probabilistic P Systems. International Journal of Foundations of Computer Science 17(1), (2006), 183–204.
14. J.W. Rayleigh: The Theory of Sound, vol I. 2nd edition. New York, Dover, (1945).
15. E. Rivero-Gil, M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez: Graphics and P Systems: Experiments with JPLANT. In these Proceedings.
16. A. Romero-Jiménez, M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, M.J.: Graphical Modelling of Higher Plants Using P Systems. In H.J. Hoogeboom, Gh. Paun, G. Rozenberg, A. Salomaa (eds.) Membrane Computing, Seventh International Workshop, Lecture Notes in Computer Science, 4361, (2006), 496–506.
17. M. Schienbein and H. Gruler: Langevin Equation, Fokker-Planck Equation and Cell Migration. Bulletin of Mathematical Biology, 55, (1993), 585-608.
18. N. Wiener. Differential Space. Journal of Mathematical Physics, 2, (1923), 131–174.