Univerzitet u Beogradu
Poljoprivredni fakultet
Institut za poljoprivrednu tehniku
Naučni časopis
POLJOPRIVREDNA TEHNIKA
Godina XXXIX
Broj 3, 2014.
Strane: 85 – 97

University of Belgrade
Faculty of Agriculture
Institute of Agricultural Engineering
Scientific Journal
AGRICULTURAL ENGINEERING
Year XXXIX
No. 3, 2014.
pp: 85 – 97

UDK: 517.925

Originalni naučni rad
Original scientific paper

STOCHASTIC GENERATOR FOR A REAL-TIME WIND SIMULATION

Zuzana Palková*, Lukáš Tomášik

Slovak University of Agriculture in Nitra, Faculty of Engineering,
Department of Electrical Engineering, Automation and Informatics
Nitra, Slovakia

Abstract: In nature, wind is considered to be a stochastic attribute of weather, whose momentary speed and direction are easily measureable, but extremely hard or nearly impossible to predict. Wind simulation is used in many fields of computer processing and modeling. Our stochastic generator is primary designed for use in wind modeling in the simulations of irrigation processes. In real world there is an event, rather a movement that is generally considered to be unable to predict – a flight of a real object. In real conditions, every object thrown in any direction, other than straight down, moves along a ballistic curve. Our method is based on simulating the flight of high number of objects and collisions between them. Collisions are implemented for two main reasons – to move objects in a bounded finite space when collisions with borders are implemented, and to change the objects' directions at a time of a collision to make objects to stay in space.

Key words: *ball collision, stochastic generator, true-random generator, wind simulation*

INTRODUCTION

In nature, wind is considered to be a stochastic attribute of weather, whose momentary speed and direction are easily measureable, but extremely hard or nearly impossible to predict. With simplification, we can predict that values of the wind speed and direction will fall in a specific interval. Wind simulation is used in many fields of

* Corresponding author. E-mail: zuzana.palkova@uniag.sk

computer processing and modeling, including physical simulation, demos and of course, in some current games.

For simulating the stochastic attributes of wind in computer modeling, random numbers generators and generators based on deterministic chaos are used at most. Prediction of wind speed and direction, like most meteorological variables, is extremely difficult. Even with advanced technology, such as sophisticated numerical models and super computers, using climatological means is as accurate as predicting meteorological variables for a time period of more than a few days in advance (Tribbia, *et al*., 1987)

Our stochastic generator is primary designed for use in wind modeling in the simulations of irrigation processes. In these kinds of simulations, we need a real-time generator to simulate the actual wind speed and direction in a very small time delays – for example tens of seconds.

Sometimes knowing wind speed without concern for wind direction is sufficient and, thus, many of the wind studies do not consider a wind direction component (Skidmore, 1987). Generators based on statistic data of wind speed and direction per day or month are good for obtaining the maximum, minimum and average values of wind speed, as well as its direction, but are unsuitable for real-time simulation because of the absence of accurate data in the actual time step. These missing data cannot be calculated using interpolation or approximation because this would result in a quite flat real-time data.

General pseudo-random generators that are primary used in the fields of computer modeling, where sharp peeks in the output random data wave are allowed, or there is high need of speed. These pseudo-random generators are based, with small differences, on the following equation (Claus, *et al*., 1986):

$$z = (a * z + c) \bmod p \tag{1}$$

Where a, c and p are integer or float constants and z is called a "Seed". This is the initial pseudo-random value, usually calculated out of the machine system time from milliseconds or sometimes microseconds. When the new value Rand is generated, it replaces the one of the constant from the Equation 1. The data flow generated from this kind of generators can looks like Fig.1.

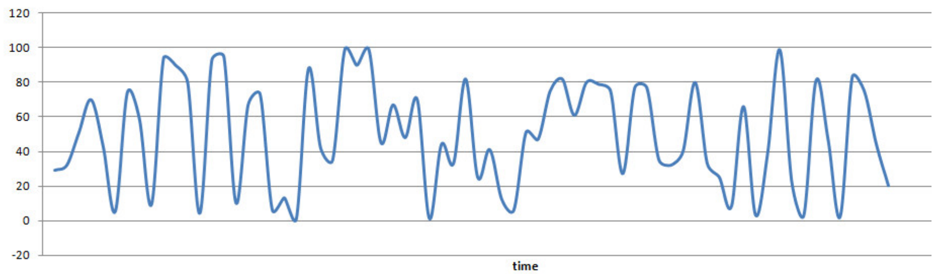

Figure 1. Data flow from a typical pseudo-random generator

As shown on Fig. 1, the output wave contains many sharp peaks. According to the principles of pseudo-random generators, the change of randomness of the data is very good, but the format of output wave is unsuitable for real-time wind modeling. This data will cause the wind model to very quickly increase and decrease its speed, or a direction, the event that is not natural.

The main subject of this article is to present a new method of stochastic generator for use in the computer wind simulation based on clear natural principles. In nature, many events occur, that are considered to be stochastic and it is hard to simulate them on computers. These events are appropriate for true-random generators, where the character of the data wave is truly unpredictable and nondeterministic. After simulating these events on a computer, we cannot fully preserve the true-random factor, but we can approximate this behavior as much as possible.

MATERIAL AND METHODS

Our intention was to design a stochastic generator based on a clear natural principles. Even the designed generator is executed on a computer, its output random values are independent on computer local system time, unlike other general random functions. Its output values are nearly the same as would be measured when simulating the generator's job in real world with real conditions. In the previous sentence, the word "nearly" was used – we omitted some attributes and events that normally occur in nature, but for our purpose they are unnecessary.

In real world there is an event, rather a movement that is generally considered to be unable to predict – a flight of a real object. In real conditions, every object thrown in any direction, other than straight down, moves along a ballistic curve. Generally, actual position of a flying object has to be simulated from its starting position to actual position at a specific time. The reason is that this movement cannot be directly calculated, because of quite large number or factors that influence the movement of an object, including drag force, gravity, and the Corioli's force. The drag force depends at most on object's speed – higher the object's speed, the higher the drag force and vice versa – this is probably the most important factor, why the flight along the ballistic curve is not predictable – the drag force slows down the object, simultaneously the drag force is lowered by slowing of an object.

Simulating the flight of one object was inadequate for our intention – there are no random values of a specific pre-defined interval that can be extracted out of the simulation, we would be able to extract only a position of an object at a time. Our method is based on simulating the flight of high number of objects and collisions between them. The collision is implemented for two main reasons – to move objects in a bounded finite space when collisions with borders are implemented, and to change the objects' directions at a time of a collision. As objects, we have chosen to work with balls as there is no need to consider objects' rotations.

Our method of stochastic generator has been divided into three basic steps – first step is to simulate totally elastic 2D collision of balls of finite mass including Newton's conservation of momentum. In the second step, the drag force of the fluid is included in the computation. The drag force causes the movement of the ball to be harder to predict, because this force slows down the ball in both directions and is directly proportional to the square of the object's speed. There are some other factors that influence the value of this force, like density of the fluid where the object moves and the shape of the object.

With possible application of gravity, the ball would move along the ballistic curve. There is no equation to directly compute the trajectory of ballistic curve, existing programs simulate the flight of object – for example a projectile – to get the hit point. This means that the spot, where the ball hits the ground is not able to be determined in one-step computation. In our method, we omitted the gravity, because it has a minor change in the stochastic factor of the whole generator. In the last third step, we calculate the area of the triangles drawn by every triplet of balls. The higher number of triplets, the smoother the generator.

Step 1: Elastic 2D collision

We considered two balls named b_1 and b_2 with finite masses m_1, m_2 and velocities v_1 and v_2. The Fig. 2 illustrates the typical collision of these two balls. The two balls collide, when the distance between their centers is equal to the sum of their radiuses. In computer modeling based on discrete time frames, we allow this distance to be less in just one frame, to just properly detect collision.

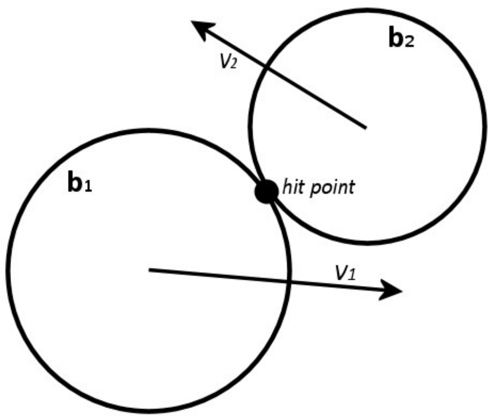

Figure 2. General situation of ball collision in 2D

According to the Fig. 2, we mark the spot, where two balls collide the "hit point". There are actually three phases of ball collision – the first phase takes place one frame before the collision, the second phase is the frame when the balls collide and the balls "exchange" their velocity vectors according to the conservation of momentum and conservation of kinetic energy. Finally, the last third phase is the frame where the balls moves in changed directions – the frame right after the collision. The situation illustrated on Fig. 2 is a combination of the first and the second phase.

After getting the actual hit point, we find the axis of collision. The axis of collision is the axis that passes through the centers of the balls and the actual hit point. Then we break up the velocity vectors v_1, v_2 into their components v_{x1}, v_{y1}, v_{x2}, v_{y2} in two-dimensional space, as shown on the Fig. 3.

Finding the axis of collision is a very important operation, because all the change in velocities of both balls occurs only in component vectors v_{x1} and v_{x2} – the vectors parallel to the axis of collision. The component vectors v_{y1} and v_{y2} remain unchanged.

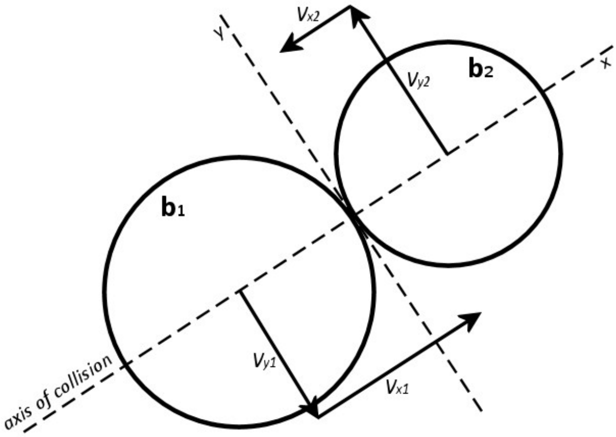

Figure 3. Breaking the velocity vectors into their components according to the axis of collision

When the balls in a collision are of the same mass, we can just exchange their velocity vectors v_{x1} and v_{x2} and make them opposite by multiplying them by a scalar-*1*. The situation after this operation is illustrated on the Fig. 4.

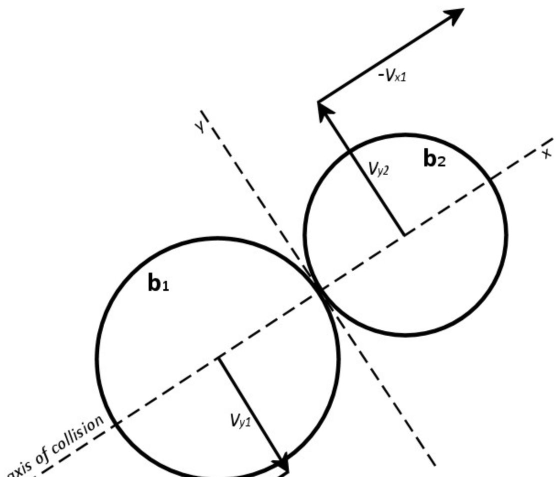

Figure 4. Situation after exchanging the velocity component vectors

The sum of the new component vectors gives the new velocity vectors v_{n1}, v_{n2}. After the collision, the balls move according to these vectors.

When the two balls are not of the same mass, the exchange of the velocity component vectors is not enough and their values must be calculated using following formulas:

$$v_{final1} = \frac{v_1(m_1 - m_2) + 2m_2v_2}{m_1 + m_2} \quad v_{final2} = \frac{v_2(m_2 - m_1) + 2m_1v_1}{m_1 + m_2} \tag{2}$$

Our random generator works with balls of different (random) masses, the calculation of velocities is made using the Eq. 2 with a little change. As we work in a 2D space, all vectors have two coordinates *X* and *Y*. Even, there is a vector of acceleration for each ball, the calculation is applied for velocity vectors only.

The procedure of finding the final velocities after the collision starts with finding the unit vector *u* from the two balls' coordinates.

$$u = [\mathrm{p}_{1x} - \mathrm{p}_{2x}\,;\ \mathrm{p}_{1y} - \mathrm{p}_{2y}] \tag{3}$$

In the above equation, p_{1x} and p_{1y} represent the position of first ball. Similarly, p_{2x} and p_{2y} represent the position of second ball. Both these positions are taken right at a time of a collision. It principally does not matter which ball is marked *1* or *2*.

Before breaking the ball's velocity vector into component vectors, we calculated the unit vector *n*, marked as *un*:

$$un = \left[\frac{n.X}{\sqrt{n.X^2 + n.Y^2}}\ ;\ \frac{n.Y}{\sqrt{n.X^2 + n.Y^2}}\right] \tag{4}$$

To break velocity vector into its component vectors we calculated the dot product of ball's velocity vector and the unit vector *un*. The equation we used looks like as follows:

$$\left|v_{x1(2)}\right| = v_{1(2)}.X \cdot un.X + v_{1(2)}.Y \cdot un.Y \tag{5}$$

Components vectors v_{x1} and v_{x2} are parallel to the axis of collision, but to calculate the final velocity vectors of each ball out of this component vectors, we must preserve the component vectors that are preserve to the axis of collision – vectors v_{y1} and v_{y2}. The equations similar to the previous one, but instead of vector *un*, we used the normal vector of *un*.

$$\left|v_{y1(2)}\right| = v_{1(2)}.X \cdot un.Y + v_{1(2)}.Y \cdot (-un).X \tag{6}$$

The next step is to calculate new size of velocity components vectors v_{x1} and v_{x2} using the general Eq. 2. Vectors v_{y1} and v_{y2} remain unchanged.

To calculate the final velocities right after the collision is simple - the sum of the components vectors gets us the final vector of velocity.

Step 2: The drag force

Drag force is a force that counteracts the movement of any real object in real fluids. In the non-inertia systems that we are simulating, it results in a negative acceleration against the movement of the ball.

The drag force translated into 2D space is calculated using the following equation:

$$F_d = \left[\frac{1}{2}\,\varrho\,C_d\,v.X^2\,A\,;\,\frac{1}{2}\,\varrho\,C_d\,v.Y^2\,A\right] \tag{7}$$

Where:

F_d density of the fluid,

C_d drag coefficient, for a sphere the value is *0.47,*

A area of an object perpendicular to the direction of movement, in our case calculated as πr^2.

According to the Newton's second law of movement, the negative acceleration to the ball's movement is calculated this way:

$$a = \left[-\frac{F_d.X}{m}\,;-\frac{F_d.Y}{m}\right] \tag{8}$$

Where:

m weight (mass) of the ball.

Step 3: Creating triangles out of ball triplets

With every ball triplet, our generator creates a triangle, which vertexes are equal to the balls' centers. The perimeters and areas of these triangles are very stochastic and almost impossible to predict. The generator could use only calculation of perimeters, but we decided to calculate the area of triangles to generate more peaks in a random data wave. After the area of the triangle is known, we compare it to the maximum possible area of triangle – when using square space, like in our case, the maximum possible area of triangle is the half of the area of space where the balls move.

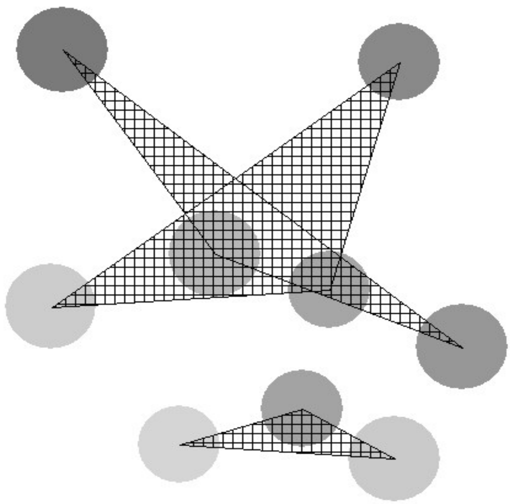

Figure 5. Creating triangles out of ball triplets

On Fig. 5, a screenshot of our generator is shown. The total size of cross-filled area is compared to the maximum possible area of triangle multiplied by the number of triplets. On the picture above, there are three ball triplets. The higher number of triplets, the smoother the generator is.

RESULTS AND DISCUSSION

Using our generator, we simulated the flight and collisions of balls in three liquids, which vary in density:

- air,
- water,
- mercury.

Table 1. Stochastic data flow in air– ϱ=1.220 kg.m^3, 3 triplets

Time step	Data	Time step	Data	Time step	Data	Time step	Data	Time step	Data
1	12,58	26	8,62	51	39,48	76	37,39	101	12,77
2	22,56	27	14,31	52	41,79	77	40,97	102	20,82
3	35,23	28	23,56	53	37,74	78	47,17	103	23,66
4	40,32	29	32,19	54	39,53	79	43,48	104	34,27
5	38,68	30	25,51	55	38,19	80	28,27	105	42,58
6	25,82	31	16,75	56	37,41	81	21,91	106	31,44
7	26,07	32	23,81	57	49,00	82	40,78	107	19,25
8	17,34	33	22,92	58	47,88	83	53,91	108	11,26
9	6,83	34	14,15	59	34,41	84	38,15	109	9,08
10	8,70	35	8,83	60	27,16	85	29,86	110	10,72
11	13,13	36	13,84	61	32,27	86	27,61	111	10,53
12	10,59	37	15,19	62	50,12	87	22,48	112	19,30
13	16,12	38	20,16	63	44,49	88	19,11	113	21,97
14	14,62	39	29,63	64	31,30	89	26,56	114	14,70
15	15,52	40	25,80	65	34,44	90	20,50	115	12,84
16	26,98	41	29,83	66	29,97	91	15,42	116	27,95
17	31,65	42	22,28	67	28,84	92	18,28	117	39,10
18	36,15	43	16,11	68	34,05	93	19,11	118	39,43
19	29,33	44	24,11	69	33,46	94	25,91	119	39,16
20	29,37	45	24,21	70	34,93	95	31,12	120	37,11
21	33,63	46	25,89	71	35,09	96	34,96	121	37,07
22	32,57	47	29,15	72	36,44	97	29,91	122	33,64
23	21,26	48	30,64	73	43,47	98	26,77	123	30,05
24	10,57	49	23,14	74	53,03	99	19,02	124	26,03
25	6,18	50	21,74	75	51,53	100	11,94	125	31,20

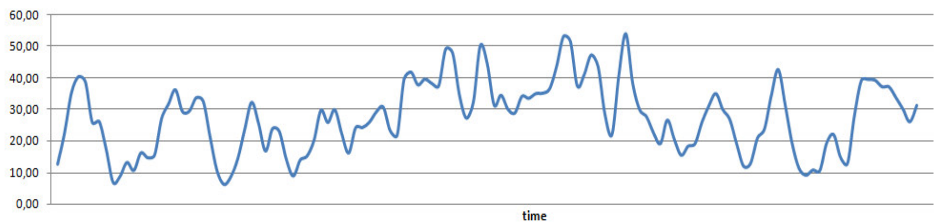

Figure 6. Stochastic data flow in air

Table 2. Stochastic data flow in water–ϱ=1000 kg.m^3, 3 triplets, 2 cycles

Time step	Data	Time step	Data	Time step	Data	Time step	Data	Time step	Data
1	16,49	26	14,36	51	25,84	76	41,32	101	13,72
2	21,00	27	20,51	52	26,14	77	40,01	102	12,40
3	24,33	28	15,76	53	33,09	78	37,96	103	13,90
4	25,58	29	23,01	54	32,59	79	32,58	104	19,06
5	38,02	30	33,08	55	31,63	80	38,56	105	22,34
6	48,82	31	39,37	56	26,15	81	45,58	106	23,02
7	42,25	32	38,91	57	23,74	82	37,92	107	35,92
8	27,43	33	42,45	58	26,32	83	25,93	108	41,68
9	26,41	34	48,34	59	31,87	84	23,85	109	38,62
10	33,03	35	41,74	60	32,96	85	30,40	110	33,72
11	39,21	36	28,92	61	25,59	86	38,05	111	26,92
12	41,78	37	28,95	62	33,36	87	35,40	112	21,78
13	42,44	38	32,19	63	41,13	88	32,34	113	20,10
14	46,45	39	25,57	64	33,10	89	31,78	114	22,95
15	34,18	40	24,32	65	27,39	90	45,76	115	19,92
16	20,66	41	32,91	66	19,56	91	54,70	116	16,70
17	10,96	42	32,26	67	20,88	92	63,38	117	20,54
18	19,56	43	21,51	68	26,99	93	47,42	118	20,65
19	28,63	44	14,48	69	24,90	94	45,12	119	19,35
20	27,08	45	7,01	70	25,95	95	55,69	120	18,02
21	26,39	46	11,17	71	31,29	96	54,99	121	20,45
22	25,16	47	15,91	72	33,40	97	48,60	122	21,90
23	20,72	48	13,45	73	36,43	98	31,64	123	20,93
24	8,64	49	16,46	74	39,91	99	21,33	124	24,63
25	13,66	50	20,85	75	40,96	100	19,64	125	20,14

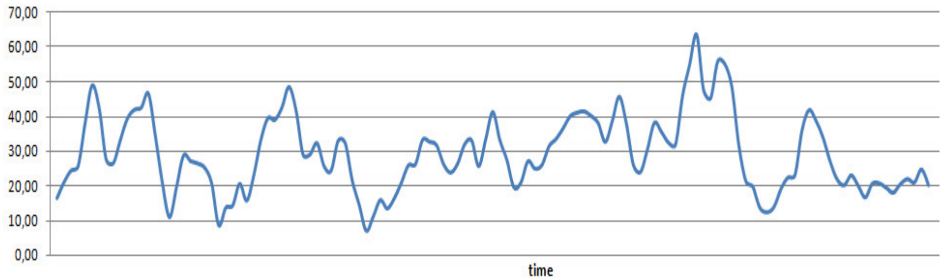

Figure 7. Stochastic data flow in water

Table 3. Stochastic data flow in mercury–ϱ=13534 kg.m^3, 3 triplets, 2 cycles

Time step	Data	Time step	Data	Time step	Data	Time step	Data	Time step	Data
1	5,95	26	36,53	51	38,38	76	22,57	101	46,41
2	7,09	27	28,47	52	33,95	77	29,09	102	55,27
3	7,27	28	29,22	53	30,25	78	28,72	103	49,37
4	14,23	29	31,54	54	34,65	79	23,40	104	45,18
5	20,52	30	38,45	55	34,83	80	13,98	105	48,59
6	19,36	31	37,57	56	32,66	81	11,78	106	52,89

Time step	Data	Time step	Data	Time step	Data	Time step	Data	Time step	Data
7	12,69	32	36,73	57	36,26	82	9,55	107	54,67
8	10,25	33	34,64	58	33,92	83	13,52	108	52,58
9	24,07	34	34,05	59	31,28	84	18,64	109	48,35
10	22,91	35	29,23	60	32,08	85	20,91	110	45,69
11	24,63	36	36,69	61	32,36	86	21,70	111	46,95
12	25,46	37	37,61	62	26,39	87	25,81	112	44,26
13	34,53	38	32,22	63	24,49	88	22,51	113	39,34
14	42,52	39	30,77	64	22,37	89	15,86	114	38,04
15	40,71	40	33,34	65	19,28	90	9,08	115	40,36
16	39,34	41	30,37	66	18,87	91	14,85	116	42,43
17	38,48	42	26,26	67	20,34	92	23,09	117	44,06
18	31,24	43	23,06	68	25,13	93	25,39	118	44,36
19	19,70	44	28,67	69	26,82	94	24,01	119	44,91
20	14,90	45	34,88	70	24,49	95	30,87	120	45,35
21	12,79	46	36,24	71	22,06	96	36,83	121	44,12
22	19,69	47	46,06	72	19,68	97	33,73	122	43,35
23	29,60	48	57,17	73	12,00	98	31,30	123	50,47
24	38,24	49	51,65	74	11,82	99	33,12	124	49,68
25	40,45	50	45,82	75	20,02	100	39,62	125	47,79

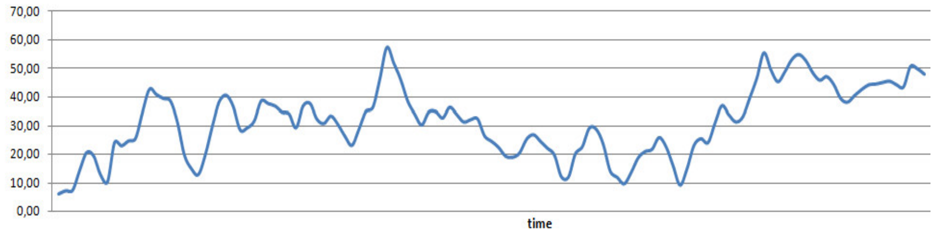

Figure 8. Stochastic data flow in mercury

As diagrams above show, with increasing the density of the fluid, the generator produces smoother waves. In all cases, the waves have very stochastic, non-periodic behavior.

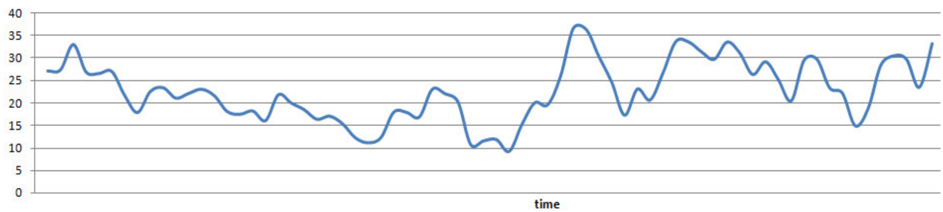

Figure 9. Stochastic data flow in water – 6 triplets

The waves are just an array of float numbers representing the current random value in a particular time. With these values we can simulate both speed and direction of wind, even with different parameters.

The system of our stochastic generator is very open to an ability of influencing its behavior by changing a high number of attributes like balls' initial velocities or accelerations, masses, radiuses, fluid density and even the generator uses only balls, by changing the C_d *(Eq. 7)* value, we can simulate aerodynamic behavior of any other shapes. There is also a very high ability to change the smoothness and variability of the output data by changing the number of ball triplets. For comparison, we generated two data flows – one with 6 triplets and second with 12 triplets, both in water.

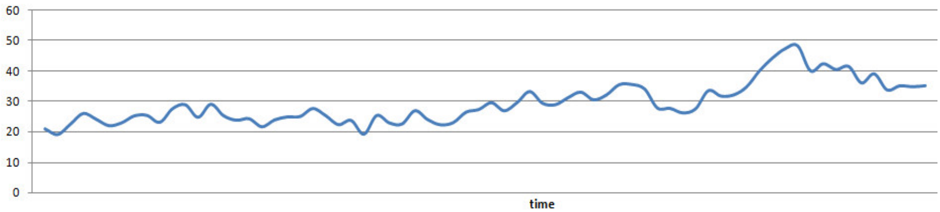

Figure 10. Stochastic data flow in water – 12 triplets

As Figs. 9 and 10 present, by increasing the number of ball triplets, we can simply increase the smoothness of the output data. The data wave becomes flatter.

CONCLUSIONS

Designed stochastic generator simulates true-random events in nature – flight of objects in a specific fluid where the drag force slows down any object, in other words simulates the unpredictable flight along a ballistic curve. To make the generator run for almost infinite time, we applied collisions of objects and omit gravity. The generator produces output random data flow that can be used in any application as its input data.

The character of data flow is non-periodic and nondeterministic. It does not contain sharp peaks and is smooth as the computer wind model has to be.

Main advantages of our stochastic generator are listed:

- based on a clear natural principle, which is considered to be stochastic or true-random,
- very high ability to produce stochastic and non-periodic waves,
- ability to easily set the smoothness of the generator – by increasing the density of fluid or by increasing the number of balls triplets,
- can run for a practically infinite time,
- easily to be implement in all fields of computer simulations of stochastic features.

To present the generator's primary goal – to simulate effects of wind in the irrigation process, we have created a time-based irrigation model that simulates the irrigation with a standard pivot sprinkler. With optimal weather conditions (no wind), the irrigated area is of an exactly circular. After adding a response to our generator, we change the trajectory of water drops according to the blowing wind. The direction of wind was set to NW and its velocity was generated during a period of time.

The results are shown on the Fig. 11.

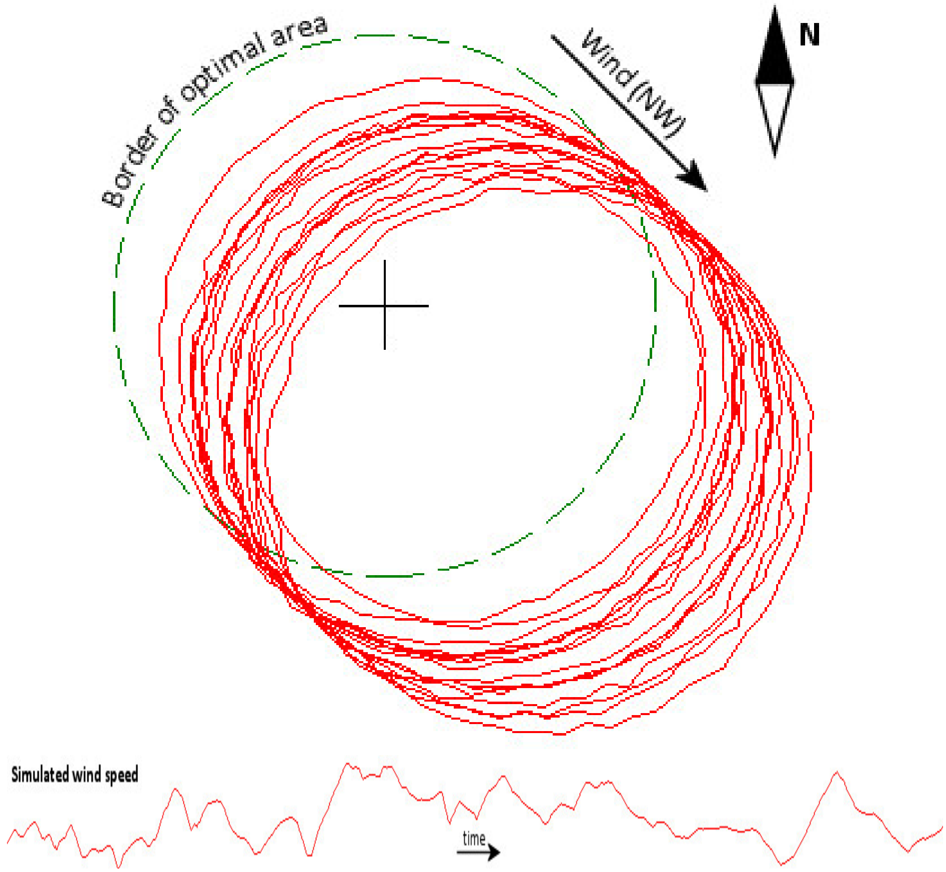

Figure 11. Results of the wind modeling using a pivot sprinkler

On Fig. 11, the sprinkler was placed at the cross mark, right in the middle of the circular irrigated area. According to the effect of wind, the shape of this area is changed as the trajectories of water drops are influenced by the wind speed and its direction.
Using of our designed generator helps to improve the agricultural production, as we can simulate the efficiency of irrigation during almost any weather conditions.

BIBLIOGRAPHY

[1] Claus, V., Schwill, A., 1986. *Schülerduden – Die Informatik*. Mannheim: Bibliographisches Institut 1986, ISBN 3-411-02213-2

[2] Capinski M., Zastawniak T. J., 2001. *Probability Through Problems*. Springer 2001, ISBN 978-0-387-95063-1

[3] Newton, R. G., 2002. *Thinking About Physics*. 2002. Princeton University Press 2002. ISBN 9780691095530

[4] Allen, M.P., Tildesley, D.J., 2002. *Computer Simulation of Liquids*. 1989. Clarendon Press 2002. ISBN 9-7801-9855645-9

[5] Stirzaker, D., 2005. *Stochastic Processes and Models*. 2005. Oxford University Press. ISBN 0-198-56814-2

[6] Gardiner, C.W., 2011. *Stochastic Methods.* 2011. Springer-Verlag Berlin and Heidelberg GmbH & Co. K. 2011. ISBN 3-540-70712-3

[7] Abarbanel, H.D.I., 1996. *Analysis of Observed Chaotic Data*. Springer-Verlag 1996.

[8] Tribbia, J.J., Anthes, R.A. 1987. Scientific basis of modern weather prediction. 1987. *Science* 237: 493-499

[9] Skidmore, E.L., 1987. Wind-erosion direction factors as influenced by field shape and wind preponderance. 1987. *Soil Sci. Soc. Am. J.* 51: 198-202

STOHASTIČKI GENERATOR ZA SIMULACIJU VETRA U REALNOM VREMENU

Zuzana Palková, Lukáš Tomášik

Slovački poljoprivredni univerzitet u Nitri, Tehnički fakultet, Institut za elektrotehniku, automatiku i informatiku, Nitra, Slovačka

Sažetak: U prirodi se vetar smatra stohastičkim atributom vremena, čiji se trenutni brzina i smer mogu lako izmeriti, ali ih je veoma teško ili gotovo nemoguće predvideti. Simulacija vetra se koristi u mnogim oblastima računarske obrade i modeliranja. Naš stohastički generator je primarno konstruisan za upotrebu u modeliranju vetra u simulacijama postupaka navodnjavanja. U stvarnom svetu postoji jedan događaj, ili bolje pokret koji smatra da je nemoguće predviđanje leta realnog objekta. U stvarnim uslovima, svaki predmet bačen u bilo kom smeru, izuzev pravo dole, kreće se po balističkoj krivoj. Naš metod se zasniva na simulaciji leta velikog broja objekata i njihovih međusobnih sudara. Sudari su uvršteni iz dva glavna razloga – da pomeraju objekte u ograničenom konačnom prostoru kada su uvršteni sudari sa granicama, i da menjaju smerove objekata u vreme sudara da bi objekti ostali u prostoru.

Ključne reči*: sudar lopte, stohastički generator, stvarni-nasumični generator, simulacija vetra*

Prijavljen: *Submitted*: 07.02.2014
Ispravljen: *Revised*:
Prihvaćen: *Accepted*: 07.09.2014