

МЕТОДИ И АЛГОРИТМИ ЗА СИМУЛАЦИОННО МОДЕЛИРАНЕ НА ФРАКТАЛНИ ПРОЦЕСИ

METHODS AND ALGORITHMS FOR SIMULATION MODELLING OF FRACTAL PROCESSES

Evgeniya Gospodinova

Institute of Robotics

Bulgarian Academy of Sciences

jenigospodinova@abv.bg

Abstract

The report presents methods and algorithms for simulation modelling of fractal processes. Fractal processes based on fractal Brownian motion, fractal Gaussian noise and, fractal Gaussian noise- wavelet transformation are simulated. Based on the performed comparative analysis of the algorithms for simulation modelling of fractal processes with respect to the accuracy parameter, it follows that the algorithms based on the models of fractal Gaussian noise and fractal Gaussian noise- wavelet transformation have the smallest relative error with respect to the Hurst parameter. The value of the Hurst parameter is one of the most important characteristics determining the degree of self-similarity of fractal processes. The considered algorithms based on these two models can be applied for modelling of physiological data, including cardiological data, because they have fractal properties.

Keywords: fractal Gaussian noise, fractal Brownian motion, fractal Gaussian noise-wavelet transformation, simulation modelling, Hurst parameter, fractal process, cardiological data.

1. ВЪВЕДЕНИЕ

Симуляционното моделиране е програмно средство за изследване на сложни системи и процеси. Въпреки, че това моделиране не позволява постигане на абсолютна точност на резултатите от симулацията на изследваните процеси и системи, но на практика това не винаги е необходимо, достатъчно е да се получат резултати със зададена точност.

За да се избере подходящ модел за симулиране, трябва да се вземат предвид следните характеристики, които моделът да притежава:

- *Универсалност* – моделът да бъде подходящ за различни видове процеси и да позволява промяна на стойностите на дефинираните параметри. Ако моделът е достатъчно универсален, тогава аналитичните средства ще позволяват анализите да бъдат универсални;
- *Простота* – моделът да се описва с малък брой параметри и чрез тях да се прави интуитивна преценка на реални физични величини;
- *Лесно присвояване* – моделът да е подходящ за реален процес (система), чрез избор на подходящи стойности за дефинираните параметри, които се получават, чрез решаване на подходяща система от уравнения;

- *Поддаващ се на анализ* – почти винаги точният анализ е невъзможен, тъй като това изисква голям брой изчисления, но моделът трябва да осигурява с определена степен на точност приблизителен анализ;
- *Лесен за приложение* – моделът да бъде лесно приложим при експериментални изследвания, посредством компютърни симулации.

На базата на характеристиките, които трябва да притежава модела за симулиране на процеси или системи следва, че идеален модел няма. Следователно, за да се избере един модел като най-подходящ, трябва да се изследват няколко модела, които да бъдат сравнени по редица показатели и да се определят най-добрите между тях или най-добрия.

За симулационно моделиране на фрактални процеси могат да се използват следните модели: фрактално брауново движение (ФБД) [2, 6, 7], фрактален гаусов шум (ФГШ) [5, 8] и фрактален гаусов шум-уейвлет трансформация [1].

Моделирането и анализа на различни видове процеси и сигнали се явяват от важните и значими задачи, от решенията на които зависи прогреса в редица области на науката и техниката. Много често експерименталните данни представляват фрактални времеви редове, които значително се различават от конвенционалните редове по своите статистически свойства. Основните свойства на фракталните процеси са: самоподобие и фрактална размерност. Един процес е самоподобен, ако той може да се разложи на по-малки части, всеки от които е подобен на основната част. Степента на самоподобие е една от важните характеристики на фракталните процеси и тя може да се определи само от един параметър, известен като параметър на Хърст (H). Определянето на този параметър играе важна роля в изучаването на процеси и явления, имащи фрактални свойства [3, 4].

Целта на настоящата статия е да се симулират фрактални процеси на базата на фрактално брауново движение, фрактален гаусов шум и фрактален гаусов шум-уейвлет трансформация. Да се извърши сравнителен анализ на симулираните фрактални процеси по отношение на параметъра точност, чрез изчисляване на относителната грешка на определените стойности на хърст параметъра спрямо входните стойности на този параметър.

2. АЛГОРИТМИ ЗА СИМУЛАЦИОННО МОДЕЛИРАНЕ НА ФРАКТАЛНИ ПРОЦЕСИ

2.1. Алгоритми на базата на ФБД

В настоящата статия са разгледани следните два алгоритъма за симулиране на фрактални процеси на базата на ФБД: random midpoint displacement (RMD) и successive random addition (SRA).

2.1.1 Алгоритъм RMD [6]

Основният принцип на *RMD* алгоритъма е рекурсивно разделяне на зададен времеинтервал n пъти и генериране във всеки новообразуван интервал на нови точки, които са средни на крайните точки от предшестващия интервал. Правилото за генериране на средната точка $Z((a+b)/2)$, базирана на крайните две точки $Z(a)$ и $Z(b)$ е на принципа на ФБД. Отместването спрямо средната точка $Z_{RMD} = Z((a+b)/2) - [Z(a) + Z(b)]/2$ не зависи от $Z(a)$ и $Z(b)$ и е нормално разпределено със средна стойност нула, ако $Z(t)$ е ФБД.

Дисперсията на отместването на средната точка за k -тата итерация се определя със следното равенство:

$$\text{Var}[Z_{\text{RMD}}]=2^{-kH}(1-2^{2H-2})T^{2H} \quad (1)$$

Където:

- H е хърст параметъра;
- k е поредния номер на итерацията на процеса;
- T е времето.

Основните стъпки на алгоритъмът са следните:

Стъпка 1: Ако процесът Y се изчислява за времеинтервала $[0,1]$, то се правят следните установявания: $Y_0=0$; Y_1 е псевдо-случайно число с нормално вероятностно разпределение със средна стойност 0 и дисперсия $\text{Var}[Y_1]=\sigma_0^2$. Тогава $\text{Var}[Y_1-Y_0]=\sigma_0^2$.

Стъпка 2: Стойността $Y_{1/2}$ се определя като средна стойност на Y_0 и Y_1 :

$$Y_{1/2}=1/2(Y_0+Y_1)+d_1 \quad (2)$$

Където отместването d_1 е случайно число с нормално разпределение със средна стойност 0 и дисперсия S_1^2 . Отместването се умножава с мащабируем коефициент $1/2$. Визуализацията на тази и следващите стъпки е показано на фиг. 1. От дисперсията $\text{Var}[Y_{1/2}-Y_0]=\left(\frac{1}{2}\right)^{2H} \sigma_0^2$ при $0 \leq t_1 \leq t_2 \leq 1$, следва:

$$\begin{aligned} \text{Var}[Y_{1/2}-Y_0] &= \frac{1}{4} \text{Var}[Y_1-Y_0] + S_1^2 \\ \left(\frac{1}{2}\right)^{2H} \sigma_0^2 &= \frac{1}{4} \sigma_0^2 + S_1^2. \end{aligned} \quad (3)$$

Следователно:

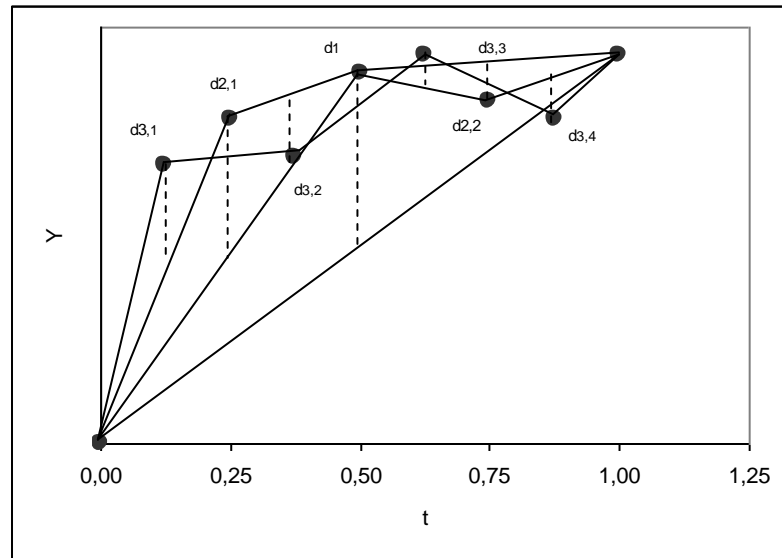
$$S_1^2 = \left(\frac{1}{2}\right)^{2H} (1-2^{2H-2}) \sigma_0^2. \quad (4)$$

Стъпка 3: Мащабируемият фактор се намалява с $\sqrt{2}$ и приема стойност $1/\sqrt{8}$. Двата интервала от 0 до $1/2$ и от $1/2$ до 1, получени от предишната стъпка се разделят наполовина. Изчислява се стойността на $Y_{1/4}$ като се намира средната стойност на първия интервал и се добавя отместването $d_{2,1}$, което е случайно число с нормално разпределение умножено с текущия мащабируем фактор $1/\sqrt{8}$. По аналогичен начин се изчислява стойността на $Y_{3/4}$. Дисперсията S_2^2 на $d_{2,*}$ се определя по следния начин:

$$\begin{aligned} \text{Var}[Y_{1/4}-Y_0] &= \frac{1}{4} \text{Var}[Y_{1/2}-Y_0] + S_2^2 \\ \left(\frac{1}{2^2}\right)^{2H} \sigma_0^2 &= \frac{1}{4} \left(\frac{1}{2}\right)^{2H} \sigma_0^2 + S_2^2 \end{aligned} \quad (5)$$

Следователно:

$$S_2^2 = \left(\frac{1}{2^2}\right)^{2H} (1 - 2^{2H-2}) \sigma_0^2. \quad (6)$$



Фигура 1. Първите три стъпки на алгоритъма RMD

Стъпка 4: Мащабируемият коефициент се намалява с $\sqrt{2}$ и приема стойността $1/\sqrt{16}$. Получените интервали в предишната стъпка се разделят отново наполовина всеки и се изчисляват новите точки:

$$\begin{aligned} Y_{1/8} &= \frac{1}{2}(Y_0 + Y_{1/4}) + d_{3,1} \\ Y_{3/8} &= \frac{1}{2}(Y_{1/4} + Y_{1/2}) + d_{3,2} \\ Y_{5/8} &= \frac{1}{2}(Y_{1/2} + Y_{3/4}) + d_{3,3} \end{aligned} \quad (7)$$

Във всяка формула $d_{3,*}$ се изчисляват като различни случайни числа с нормално разпределение, които се умножават с текущия мащабируем коефициент $1/\sqrt{16}$. На следващата стъпка се определят нови стойности за Y при $t = \frac{1}{16}, \frac{3}{16}, \dots, \frac{15}{16}$ използвайки мащабируем фактор отново намален с $\sqrt{2}$. Дисперсията S_3^2 на $d_{3,*}$ се определя по следния начин:

$$\begin{aligned} \text{Var}[Y_{1/8} - Y_0] &= \frac{1}{4} \text{Var}[Y_{1/4} - Y_0] + S_3^2 \\ \left(\frac{1}{2^3}\right)^{2H} \sigma_0^2 &= \frac{1}{4} \left(\frac{1}{2^2}\right)^{2H} \sigma_0^2 + S_3^2 \\ S_3^2 &= \left(\frac{1}{2^3}\right)^{2H} (1 - 2^{2H-2}) \sigma_0^2. \end{aligned} \quad (8)$$

Следователно дисперсията S_n^2 на $d_{n,*}$ е:

$$S_n^2 = \left(\frac{1}{2^n}\right)^{2H} (1 - 2^{2H-2}) \sigma_0^2. \quad (9)$$

Стъпка 5: По същият начин се изчисляват и останалите точки, докато се достигне желанния брой $n=2^{\text{брой_стъпки}}$. Формира се финалната последователност $X_0, X_1, X_2, \dots, X_n$, чрез полагагане $X_i \leftarrow Y_{i/2^{\text{брой_стъпки}}}$, $i=0, 1, 2, \dots$

2.1.2. Алгоритъм SRA [7]

Алгоритъмът SRA подобно на RMD алгоритъма използва средни точки, но за да се повиши стабилността на симулираната последователност, отместванията с подходяща дисперсия се добавят към всички точки. Целта на вмъкването на средните точки се състои в създаването на гаусови нараствания, които да са корелирани, а с добавянето на отместване към всички точки се цели постигане на нормално разпределение на симулирания фрактален процес.

Алгоритъмът се състои от следните стъпки:

Стъпка 1: Ако процесът Y се изчислява за времеинтервала $[0,1]$, то се правят следните установявания: $Y_0=0$, а Y_1 – псевдослучайно число с нормално разпределение със средна стойност 0 и дисперсия $Var[Y_1]=\sigma_0^2$. Тогава $Var[Y_1-Y_0]=\sigma_0^2$.

Стъпка 2: Стойността $Y_{1/2}$ се определя като средна стойност на Y_0 и Y_1 :

$$Y_{1/2} = \frac{1}{2}(Y_0 + Y_1). \quad (10)$$

Стъпка 3: Добавят се отмествания с подходяща дисперсия към всички точки:

$$\begin{aligned} Y_0 &= Y_0 + d_{1,1}, \\ Y_{1/2} &= Y_{1/2} + d_{1,2}, \\ Y_1 &= Y_1 + d_{1,3}. \end{aligned} \quad (11)$$

Отместванията $d_{1,*}$ се определят чрез случайни числа с нормално вероятностно разпределение. От дисперсията $Var[Y_{t_2} - Y_{t_1}] = |t_2 - t_1|^{2H} \sigma_0^2$ при $0 \leq t_1 \leq t_2 \leq 1$, следва:

$$\begin{aligned} Var[Y_{1/2} - Y_0] &= \frac{1}{4} Var[Y_1 - Y_0] + 2S_1^2 \\ \left(\frac{1}{2}\right)^{2H} (\sigma_0^2) &= \frac{1}{4} \sigma_0^2 + 2S_1^2. \end{aligned} \quad (12)$$

Следователно:

$$S_1^2 = \frac{1}{2} \left(\frac{1}{2^1}\right)^{2H} (1 - 2^{2H-2}) \sigma_0^2. \quad (13)$$

Стъпка 4: Стъпки 2 и 3 се повтарят. Следователно дисперсията S_n^2 на $d_{n,*}$ е:

$$S_n^2 = \frac{1}{2} \left(\frac{1}{2^n}\right)^{2H} (1 - 2^{2H-2}) \sigma_0^2. \quad (14)$$

Стъпка 5: По същият начин се изчисляват и останалите точки, докато се достигне желания брой $n=2^{\text{брой стъпки}}$. Формира се финалната последователност $X_0, X_1, X_2, \dots, X_n$, чрез полагање $X_i \leftarrow Y_{i/2^{\text{брой стъпки}}}$, $i=0, 1, 2, \dots$

2.2. Алгоритъм на базата на ФГШ

ФГШ е процес на нарастванията на ФБД $X=\{X_k:k=0,1,\dots\}$ и се представя последния начин:

$$X_k = B_H(k+1) - B_H(k) \quad (15)$$

Където X_k има нормално вероятностно разпределение за всяко k . Автокорелационната функция $r(\cdot)$ на ФГШ има следната форма:

$$r(k) = \frac{1}{2} \left[|k-1|^{2H} - 2|k|^{2H} + |k+1|^{2H} \right] \quad \text{за } k \in \mathbb{Z} \quad (16)$$

Енергетичният спектър $f(\lambda, H)$ на ФГШ се определя със следното уравнение []:

$$f(\lambda, H) = A(\lambda, H) \left[|\lambda|^{-2H-1} + B(\lambda, H) \right] \quad \text{при } 0 < H < 1 \text{ и } -\pi \leq \lambda \leq \pi \quad (17)$$

Където:

$$A(\lambda, H) = 2 \sin(\pi H) \Gamma(2H+1) (1 - \cos \lambda) \quad (18)$$

$$B(\lambda, H) = \sum_{j=1}^{\infty} \left[(2\pi j + \lambda)^{-2H-1} + (2\pi j - \lambda)^{-2H-1} \right] \quad (19)$$

$\Gamma(\cdot)$ - Гама функция

Основната трудност при изчисляване на енергетичния спектър на ФГШ е свързан с изчисляване на безкрайната сума в израза $B(\lambda, H)$.

Основните стъпки на алгоритъма, предложен от Paxson [8] са:

Стъпка 1: Създава се последователност от стойности $\{f_1, f_2, \dots, f_{n/2}\}$, където $f_j = \hat{f}\left(\frac{2\pi j}{n}, H\right)$ съответства на енергетичния спектър на ФГШ за честоти от $2\pi/n$ до π .

Стъпка 2: Всяка стойност от последователността $\{f_i\}$ се умножава с експоненциално разпределена случайна променлива със средна стойност единица.

Стъпка 3: Създава се последователност от комплексни числа $\{z_1, z_2, \dots, z_{n/2}\}$ с модул $|z_i| = \sqrt{\hat{f}_i}$ и аргумент равномерно разпределен в интервала $(0, 2\pi)$. Използването на случаен аргумент прави генерираната последователност, нормално разпределена, което е изискване за ФГШ.

Стъпка 4: Създава се последователност $\{z_0^1, z_1^1, \dots, z_{n-1}^1\}$, която е разширена версия на $\{z_1, z_2, \dots, z_{n/2}\}$ при следните условия:

$$z_j^1 = \begin{cases} 0 & \text{ако } j = 0; \\ z_j & \text{ако } 0 < j \leq \frac{n}{2}; \\ \overline{z_{n-j}} & \text{ако } \frac{n}{2} < j < n. \end{cases} \quad (20)$$

Където: $\overline{z_{n-j}}$ е комплексно спрегнато на z_{n-j} .

Стъпка 5: Използва се обратната трансформация на Фурие за представяне на генерираната последователност от честотната област във времевата област.

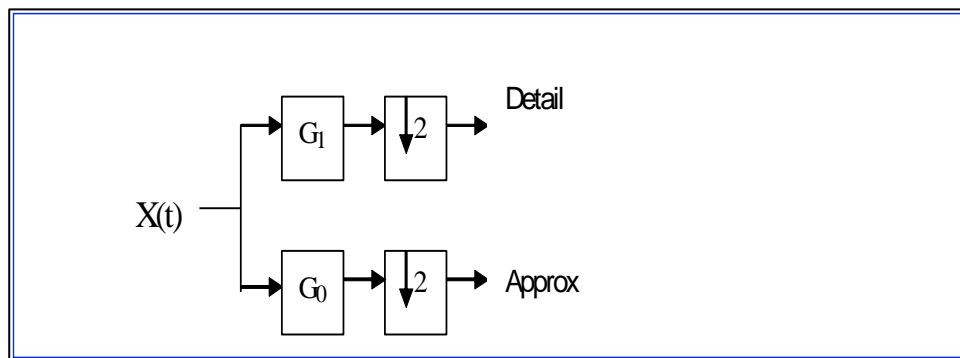
2.3. Алгоритъм на базата на ФГШ и уейвлет трансформация

2.3.1. Уейвлет трансформация

Уейвлет трансформацията е математически метод, който е подходящ за моделиране и анализ на фрактални процеси. Уейвлет трансформацията е по-добра от трансформацията на Фурие, поради следните причини:

- Фуриеровата трансформация се използва за трансформация на стационарните сигнали от честотната във времевата област, като сигналът се трансформира като сума от синусоиди с различни честоти. Тя не може да представи информацията във времевата област.
- Уейвлет трансформацията може да представи сигнала във времевата и честотната област едновременно. Тя може да трансформира както стационарни така и нестационарните сигнали без загуба на информация.

Основният алгоритъм на уейвлет трансформацията е показан на фиг. 2, който се състои от 2 вълнови филтъра: G_0 → нискочестотен вълнов филтър, G_1 → високочестотен вълнов филтър. След нискочестотния филтър се получава апроксимираща (*Approx*) информация, а след високочестотния филтър подробна (*Detail*) информация. Този алгоритъм използва рекурсивен, пирамидален алгоритъм при който *Approx* информацията, получена на изхода на нискочестотния филтър става входна информация за следващото ниво на декомпозиция [1].



Фигура 2 Основен алгоритъм на уейвлет трансформация

Уейвлет трансформацията се базира на Multi-Resolution анализ, който се състои от сума от подпространства $\{V_i, i \in Z\}$, за които са в сила следните свойства:

1. $\cap V_i = \{0\}, \cup V_i$
2. $V_i \subset V_{i-1}$;
3. $x(t) \in V_i \leftrightarrow x(2^i t) \in V_0$;
4. функцията $\phi_0(t)$ за вектор V_0 се нарича мащабируема функция, а сумата $\{\phi_0(t - j), j \in Z\}$ е ортонормален базис за V_0 .

Уейвлет алгоритмите се състоят от 2 компонента:

- Уейвлет функция ($\psi_{j,k}$), отнасяща се към високочестотния филтър – изпълнява се два или повече пъти и изчислява уейвлет коефициенти;
- Машабируема функция ($\phi_{j,k}$), отнасяща се към нискочестотния филтър – създава по-изгладена версия на оригиналните данни. Получените данни стават входни за следващата стъпка на алгоритъма.

Математическите описания на горните функции са:

$$\phi_{i,j}(t) = 2^{-i/2} \phi_0(2^{-i}t - j), j \in Z \quad (21)$$

$$\psi_{i,j}(t) = 2^{-i/2} \psi_0(2^{-i}t - j), j \in Z, \quad (22)$$

където: ϕ_0 - машабируема функция,

ψ_0 - mother вълна и произхожда от ϕ_0 .

Прилагането на Multi-Resolution анализа за входните данни x , означава представянето на x във всяко от апроксимиращите подпространства V_i :

$$approx_i(t) = \sum_j a_x(i, j) \phi_{i,j}(t). \quad (23)$$

Където, коефициентът $a_x(i, j)$ се изчислява чрез умножение на входните данни x с $\phi_{i,j}$:

$$a_x(i, j) = \langle x, \phi_{i,j} \rangle. \quad (24)$$

Апроксимиращата информация $approx_i$ е груба апроксимация на входните данни x спрямо $approx_{i-1}$ и се губи информация $detail_i(t) = approx_{i-1}(t) - approx_i(t)$. Информацията $detail$ на входните данни x се определя с формула:

$$\det ail_i(t) = \sum_j d_x(i, j) \psi_{i,j}(t), \quad (25)$$

където коефициентът $d_x(i, j)$ се изчислява чрез умножение на входните данни x с $\psi_{i,j}$:

$$d_x(i, j) = \langle x, \psi_{i,j} \rangle. \quad (26)$$

Multi-resolution анализът представя входната информация x като сума от апроксимираща и подробна информация:

$$\begin{aligned} x(t) &= approx_N(t) + \sum_{i=1}^N \det ail_i(t) \\ &= \sum_j a_x(N, j) \phi_{N,j}(t) + \sum_{i=1}^N \sum_j d_x(i, j) \psi_{i,j}(t). \end{aligned} \quad (27)$$

Уравнение (27) се нарича уейвлет декомпозиция на сигнала x , а коефициентите $a_x(j, k)$ и $d_x(j, k)$ се определят чрез умножаване на входните данни с функцията $\phi_{j,k}$ и респективно с функцията $\psi_{j,k}$.

2.3.2. Алгоритъм FGN-DW[1]

Този алгоритъм за симулиране на фрактален процес се базира на ФГШ и уейвлет трансформация. Основната стратегия на този алгоритъм е подобна на алгоритъма на Raxson, в който обратната трансформация на Фурие е заменена с обратна уейвлет трансформация. За прилагането на уейвлет трансформация е използван алгоритъма на Добеши, защото с него се постига по-голяма точност при симулиране на фрактален процес, отколкото при използването, например на уейвлет алгоритъма на Хаар. Причината за това е, че уейвлет алгоритъма на Добеши създава по-изгладени уейвлет коефициенти в сравнение с уейвлет алгоритъма на Хаар.

За изчисляване енергетичния спектър на ФГШ се използва следната формула:

$$f(\lambda, H) = c_f |\lambda|^{1-2H} + O(|\lambda|^{\min(3-2H, H)}), \quad (28)$$

където:

- $c_f = \sigma^2 (2\pi)^{-1} \sin(\pi H) \Gamma(2H + 1)$,
- $\sigma^2 = \text{Var}[X_k]$,
- $\Gamma(\cdot)$ – Гама функция,
- $O(\cdot)$ – остатъчна грешка.

Алгоритъмът се състои от следните стъпки:

Стъпка 1: Използвайки уравнение (28) се изчислява последователност от стойности $\{f_1, f_2, \dots, f_{n/2}\}$, където $f_j = \hat{f}(\frac{2\pi j}{n}, H)$ съответства на енергетичния спектър на ФГШ за честоти от π/n до π .

Стъпка 2: Всяка стойност от последователността $\{f_i\}$ се умножава с експоненциално разпределена случайна променлива със средна стойност единица.

Стъпка 3: Създава се последователност от комплексни числа $\{z_1, z_2, \dots, z_{n/2}\}$ с модул $|z_i| = \sqrt{\hat{f}_i}$ и аргумент равномерно разпределен в интервала $(0, 2\pi)$. Използването на случаен аргумент прави генерираната последователност нормално разпределена, което е изискване за ФГШ.

Стъпка 4: Използва се обратната вълнова трансформация за представяне на генерираната последователност от честотната област във времевата област. За обратната вълнова трансформация е използван уейвлет алгоритъма на Добеши с 4 уейвлет коефициента.

3. СРАВНИТЕЛЕН АНАЛИЗ НА АЛГОРИТМИТЕ ЗА СИМУЛАЦИОННО МОДЕЛИРАНЕ НА ФРАКТАЛНИ ПРОЦЕСИ

За симулиране на фрактални процеси на базата на представените четири алгоритъма е разработена програма на езика C++. Програмата се състои от следните модули:

- Модул за симулиране на фрактални процеси със следните входни параметри: параметър на Хърст със стойности: $H=0.6, 0.7, 0.8$ и 0.9 при дължина на процеса $N=32768$ (2^{15}) точки;
- Модул за определяне относителната грешка на хърст параметъра на симулираните процеси чрез прилагане на уейвлет-базиран статистически метод.

Сравнителният анализ на представените в статията алгоритми за симулационно моделиране на фрактални процеси е направен на базата на параметъра точност. Точността е определена чрез изчисляване на относителната грешка на определените стойности на хърст параметъра спрямо входните стойности на този параметър. В Таблица 1 са показани получените резултати.

Таблица 1: Сравнителен анализ на алгоритмите за симулационно моделиране на фрактални процеси

Метод	Алгоритъм	Относителна грешка на хърст параметъра $\Delta H(\%)$			
		H=0.6	H=0.7	H=0.8	H=0.9
ФБД	RMD	-5.3%	-4.6%	-4.1%	-3.5%
	SRA	-4.9%	-4.4%	-4.0%	-3.4%
ФГШ	Raxson	-1.9%	-1.6%	-1.5%	-1.3%
ФГШ и уейвлет трансформации	FGN-DW	-0.9%	-0.7%	-0.5%	0.1%

На базата на получените резултати от сравнителния анализ на алгоритмите за симулиране на фрактални процеси могат да се направят следните изводи:

- Относителната грешка на хърст параметъра, определена чрез уейвлет-базиран статистически метод е с отрицателни стойности. С увеличаване входната стойност на хърст параметъра, относителната грешка на този параметър намалява и при четирите изследвани алгоритъма;
- Най-малка е относителната грешка на хърст параметъра при алгоритъма FGN-DW. Причината за това е използването на уейвлет трансформация за представянето на симулираната последователност от честотната във времевата област.
- Алгоритъмът на Raxson за симулиране на фрактални процеси на базата на ФГШ също имат малки относителни грешки (по-малки от 2%) за всички изследвани стойности на хърст параметъра.

ЗАКЛЮЧЕНИЕ

Представените алгоритми на базата на ФГШ и ФГШ-уейвлет трансформация, като най-добри алгоритми по отношение на параметъра точност на симулираните фрактални процеси могат да се приложат за моделиране на кардиологични данни, защото е доказано, че те притежават фрактални свойства. Кардиологичните данни, отнасящи се до интервалите между сърдечните удари се характеризират с по-ниски стойности на хърст експонентата при болни субекти в сравнение със здравите контроли. Следователно с представените алгоритми на базата на ФГШ и ФГШ-уейвлет трансформация могат да се моделират кардиологични данни както за здрави така и за болни субекти.

БЛАГОДАРНОСТИ

Научното изследване е проведено като част от проекта „Изследване на приложението на нови математически методи за анализ на кардиологични данни“ № КП-06-Н22/5 от 07.12.2018 г., финансиран от Фонд „Научни Изследвания“.

ЛИТЕРАТУРА

- [1] Jeong, H.-D., McNickle, D., Pawlikowski, K. 1999. Fast Self-Similar Teletraffic Generation Based on FGN and Wavelets. *In Proceeding of the IEEE International Conference on Networks, ICON'99*, Australia.
- [2] Flandrin, P. 1992. Wavelet Analysis and Synthesis of Fractional Brownian Motion. *IEEE Transactions on Information Theory*, 38, 2, 910-917.
- [3] Ernst, G. 2014. *Heart Rate Variability*. London Springer-Verlag, <https://doi.org/10.1007/978-1-4471-4309-3>
- [4] Turcott, R.G., Teich, M.C. 1996. Fractal character of the electrocardiogram: Distinguishing heart-failure and normal patients. *Ann Biomed Eng* 24, 269–293. <https://doi.org/10.1007/BF02667355>
- [5] Delignières, D. 2015. Correlation Properties of (Discrete) Fractional Gaussian Noise and Fractional Brownian Motion. *Mathematical Problems in Engineering*, Volume 2015, Article ID 485623, 7 pages. <https://doi.org/10.1155/2015/485623>
- [6] Lau, W-C., Erramilli, A., Wang, J., Willinger, W. 1995. Self-Similar Traffic Generation: The Random Midpoint Displacement Algorithm and Its Properties. *IEEE International Conference on Communications*, Vol.1, pp= 466-472.
- [7] Jeong, H.-D., McNickle, D., Pawlikowski, K. 1998. Generator of Pseudo-random Self-Similar Sequences Based on SRA. *Technical Report TR-COSC 03/98*.
- [8] Paxson, V. 1997. Fast, approximate synthesis of fractional Gaussian noise for generating self-similar network traffic. *Computer Communication Review*, 27.