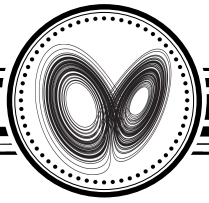
Bifurcation in Dynamical Systems. Deterministic Chaos. Quantum Chaos



Izvestiya Vysshikh Uchebnykh Zavedeniy. Applied Nonlinear Dynamics. 2023;31(1)

Article DOI: 10.18500/0869-6632-003023

Effective algorithms for solving functional equations with superposition on the example of the Feigenbaum equation

A. A. Polunovskii

A. A. Harkevich Institute of Information Transmission Problems of the RAS, Moscow, Russia
E-mail: apap2009@yandex.ru

Received 30.08.2022, accepted 9.11.2022, available online 19.01.2023, published 31.01.2023

Abstract. *Purpose.* New algorithms were consider for functional equations solving using the Feigenbaum equation as an example. This equation is of great interest in the theory of deterministic chaos and is a good illustrative example in the class of functional equations with superposition. *Methods.* The article proposes three new effective methods for solving functional equations — the method of successive approximations, the method of successive approximations using the fast Fourier transform and the numerical-analytical method using a small parameter. *Results.* Three new methods for solving functional equations were presented, considered on the example of the Feigenbaum equation. For each of them, the features of their application were investigated, as well as the complexity of the resulting algorithms was estimated. The methods previously used by researchers to solve functional equations are compared with those described in this article. In the description of the latter, the numerical-analytical method, several coefficients of expansions of the universal Feigenbaum constants were written out. *Conclusion.* The obtained algorithms, based on simple iteration methods, allow solving functional equations with superposition without the need to reverse the Jacobi matrix. This feature greatly simplifies the use of computer memory and gives a gain in the operating time of the algorithms in question, compared with previously used ones. Also, the latter, numerically-analytical method made it possible to obtain sequentially the coefficients of expansions of the universal Feigenbaum constants, which in fact can be an analytical representation of these constants.

Keywords: dynamic chaos, Feigenbaum equation, functional equations with superposition, power series.

Acknowledgements. This work was supported by the Russian Science Foundation under grant no. 22-11-00317, https://rscf.ru/project/22-11-00317/

For citation: Polunovskii AA. Effective algorithms for solving functional equations with superposition on the example of the Feigenbaum equation. Izvestiya VUZ. Applied Nonlinear Dynamics. 2023;31(1):8–19. DOI: 10.18500/0869-6632-003023

This is an open access article distributed under the terms of Creative Commons Attribution License (CC-BY 4.0).

© *Polunovskii A. A., 2023*

1. Known results in solving the Feigenbaum equation

1.1. Introduction. In the theory of deterministic chaos, the system of Feigenbaum's functional equations is of particular interest. This system has the following form [1–3]:

$$\begin{cases} g(x) = -\alpha \cdot g\left(g\left(-\dfrac{x}{\alpha}\right)\right), \\ \delta \cdot h(x) = \alpha \cdot g'\left(g\left(\dfrac{x}{\alpha}\right)\right) \cdot h\left(\dfrac{x}{\alpha}\right) + \alpha \cdot h\left(g\left(\dfrac{x}{\alpha}\right)\right), \end{cases} \tag{1}$$

with additional conditions

$$\begin{cases} g(0) = h(0) = 1, \\ g'(0) = h'(0) = 0, \end{cases} \tag{2}$$

where g and h are the desired functions, α and δ are the universal Feigenbaum constants, which are also unknown in this system of equations.

It is assumed that g and h are accounting analytical functions with a quadratic extremum at zero, defined on the real axis $\mathbb{R}$. It is required to find a pair of functions (g, h) and a pair of numbers (α, δ). Now there is only an approximate numerical solution of this system:

$$\begin{cases} g(x) = 1 - 1.52763... \cdot x^2 + 0.104815... \cdot x^4 + \\ \qquad + 0.0267057... \cdot x^6 - 0.0035274... \cdot x^8 + ..., \\ \alpha = 2.502907876.., \\ h(x) = 1 - 0.325651... \cdot x^2 - 0.50554... \cdot x^4 + \\ \qquad + 0.014560... \cdot x^6 - 0.000881... \cdot x^8 - ..., \\ \delta = 4.669201609... . \end{cases} \tag{3}$$

For the first time in 1979, M. Feigenbaum in his articles [3, 4] published the first 12 decimal places of the constant α and 13 decimal places of the constant δ. In 1991, Keith Briggs, using the same calculation method, obtained the values of these constants with an accuracy of 150 decimal places [5]. Later, in 1999, Simon Plouffe refined Briggs' results, and calculated 1018 decimal places of constants α and δ [6].

1.2. Discretization. The collocation method. The main method of discretization of the system (1) used in previous works is the collocation method (MK) [3, 5, 7]. This method is based on the representation of the desired functions and in the form of a sum of basic functions

$$\begin{cases} g(x) \approx g_N(x) = 1 + \sum\limits_{i=1}^{N} g_i^N \cdot \phi_i(x), \\ h(x) \approx h_N(x) = 1 + \sum\limits_{i=1}^{N} h_i^N \cdot \phi_i(x), \end{cases} \tag{4}$$

where $g_i, h_i \in \mathbb{R}$, $\{\phi_n\}_{n=0}^{\infty}$ is a set of basis functions, and N is a fixed integer. Basically, due to the analyticity of the functions g and h [2], this basis takes a sequence of degrees $\{x^n\}_{n=0}^{\infty}$ [3, 5, 7]. The functions g and h, in turn, were represented as power series

$$\begin{cases} g(x) \approx g_N(x) = 1 + \sum\limits_{i=1}^{N} g_i^N \cdot x^{2 \cdot i}, \\ h(x) \approx h_N(x) = 1 + \sum\limits_{i=1}^{N} h_i^N \cdot x^{2 \cdot i}. \end{cases} \tag{5}$$

Other bases were also used for calculations, for example in the article [8] the decomposition was carried out by Chebyshev polynomials.

Having fixed some basis of functions, we now choose a set of N points $\{x_j\}_{j=1}^N$, evenly distributed on the half-interval $(0, 1]$. Substituting into the system (1) the representation of the functions g and h in the form (5) and considering the obtained equalities at the points $\{x_j\}_{j=1}^N$, we get a system of $2N$ nonlinear equations

$$\begin{cases} g_N(x_j) = -\alpha_N \cdot g_N\left(g_N\left(-\dfrac{x_j}{\alpha_N}\right)\right), \\ \delta_N \cdot h_N(x_j) = \alpha_N \cdot g'_N\left(g_N\left(\dfrac{x_j}{\alpha_N}\right)\right) \cdot h_N\left(\dfrac{x_j}{\alpha_N}\right) + \alpha_N \cdot h_N\left(g_N\left(\dfrac{x_j}{\alpha_N}\right)\right), \\ j = 1, .., N, \end{cases} \tag{6}$$

with respect to the coefficients of g_i, h_i decompositions (5). Adding to the system (6) the following relations

$$\begin{cases} \alpha = -\dfrac{1}{g(1)}, \\ \delta = \alpha \cdot (g'(1) + h(1)), \end{cases} \tag{7}$$

obtained by considering the equations (1) for $x = 0$, we obtain a closed system of equations with respect to the coefficients $h_1, ..., h_N, g_1, ..., g_N$ и α_N, δ_N.

Numerically solving the system (6) by Newton's method, we obtain approximate values of α_N, δ_N constants α and δ. From computational practice, it is obtained that in the case of choosing decompositions (5), we observe the convergence of this method [3, 5, 7].

$$\begin{cases} \lim\limits_{N\to\infty} \alpha_N = \alpha, \\ \lim\limits_{N\to\infty} \delta_N = \delta, \\ \lim\limits_{N\to\infty} g_N(x) = g(x), \\ \lim\limits_{N\to\infty} h_N(x) = h(x), \end{cases} \tag{8}$$

where $x \in [-1, 1]$.

The solution of the system (6) can be simplified by taking advantage of an important property of the second equation from the system (1) — the number δ is an eigenvalue of the operator

$$[Lf](x) = -\alpha \cdot f\left(g\left(\frac{x}{\alpha}\right)\right) - \alpha \cdot g'\left(g\left(\frac{x}{\alpha}\right)\right) \cdot f\left(\frac{x}{\alpha}\right), \tag{9}$$

and the function h is an eigenvector. Feigenbaum showed [3] that the constant δ is the largest modulo eigenvalue of this operator (9) relative to the function g obtained from the first equation of the system (1). Based on this, by calculating the coefficients $g_1, ..., g_N$ and α_N from the corresponding equations of the system (6), we can find the number δ_N and the vector $\mathbf{h}^N = (h_1, ..., h_N)^T$ as the maximum modulo eigenvalue, and the corresponding eigenvector of the operator (9), calculating this operator at points $\{x_j\}_{j=1}^N$. Keith Briggs used a power-law method to solve this problem [5].

1.3. Discretization. The method of uncertain coefficients. Another approach to the discretization of the system (1) is the method of indefinite coefficients (OLS) [3, 9]. This method is also based on the representation of the solution in the form of power series (5). However, unlike the collocation method, we do not calculate the values of functions at specific

points, but consider a system of equations obtained after substituting the series (5) into the system (1), and equating the coefficients obtained at the corresponding degrees in the left and right parts of the equalities. After all the necessary transformations, we obtain the following system of equations for the coefficients of the decomposition of the function g:

$$\begin{cases} g_1^N = G_1\left(g_1^N, ..., g_N^N, \alpha_N\right), \\ ...., \\ g_N^N = G_N\left(g_1^N, ..., g_N^N, \alpha_N\right) \end{cases} \tag{10}$$

and, accordingly, a system of equations for the coefficients of the decomposition of the function h:

$$\begin{cases} \delta_N \cdot h_1^N = H_1\left(h_1^N, ..., h_N^N, \alpha_N, g_1^N, ..., g_N^N\right), \\ ...., \\ \delta_N \cdot h_N^N = H_N\left(h_1^N, ..., h_N^N, \alpha_N, g_1^N, ..., g_N^N\right). \end{cases} \tag{11}$$

Adding to the systems (10) and (11) the relations (7), we get a closed system with coefficients $h_1, ..., h_N, g_1, ..., g_N$ and α_N, δ_N. Solving this system by Newton's method, we, as in the case of the collocation method, get some approximation of α_N, δ_N to the constants α, δ. Similarly, finding the number δ and the function h can be simplified by considering the system (11) with calculated $g_1, ..., g_N$ and α_N as a task to find the maximum modulo eigenvalue and the corresponding eigenvector.

The main difficulty of this discretization approach is to calculate the functions G_i and H_i, where G_i are polynomials of $g_1, ..., g_N$ and rational functions of α_N, and H_i are polynomials of $g_1, ..., g_N$, rational functions of α_N, and linear in $h_1, ..., h_N$. In the case of a large number of N, these functions become cumbersome, and require special approaches to their calculation. However, the systems (10) and (11), in comparison with the system (6), have closer properties to the original system (1). We can say that the systems (10) and (11) are a projection onto a finite-dimensional space of the system (1), where the functions and are replaced by finite-dimensional vectors $\mathbf{g}^N = (g_1, ..., g_N)^T$ and $\mathbf{h}^N = (h_1, ..., h_N)^T$.

Due to the fact that the calculation of the second equation of the system (1) with a known solution of the first one in one or another discretization, is already known by effective methods of eigenvalue problems, we will consider algorithms on the example of solving the first equation of the original system.

The discretization method considered in this section can be used for comparison with the collocation method and their mutual verification.

2. Description of the proposed new calculation methods

All previously known calculations of the first equation from the system (1) were somehow reduced to the multidimensional Newton method, requiring *a complex computational procedure — inversion of the Jacobi matrix*. Let's now consider the methods proposed in this article to avoid this expensive operation.

2.1. The method of successive approximations.

2.1.1. ***Derivation of a system of recurrent equations.*** Let's return to the system of equations (10), which is a discretization of the first equation from the system (1) by the method of indefinite coefficients. Note that the system (10) is an equation of a fixed point relative to the vector $\mathbf{g}^N = (g_1, ..., g_N)^T$. Let's use this by first obtaining the following relations from the system (1)

$$\begin{cases} g(1) = -\dfrac{1}{\alpha}, \\ g'(1) = -\alpha. \end{cases} \tag{12}$$

Based on the chosen power decomposition (5), the conditions (12) can be represented in the final form

$$\begin{cases} 1 + \sum\limits_{i=1}^{N} g_i^N = -\dfrac{1}{\alpha}, \\ \sum\limits_{i=1}^{N} (2i) \cdot g_i^N = -\alpha. \end{cases} \tag{13}$$

Now, based on the system (10) and the relations (13), we write the following system of recurrent equations:

$$\begin{cases} g_1^{N,(n+1)} = -1 - \dfrac{1}{\alpha_N^{(n)}} - \sum\limits_{i=2}^{N} g_i^{N,(n)}, \\ g_k^{N,(n+1)} = G_k^N \left(g_1^{N,(n)}, ..., g_N^{N,(n)}, \alpha_N^{(n)} \right), \\ k = 2, .., N, \\ \alpha_N^{(n+1)} = -\sum\limits_{i=1}^{N} (2i) \cdot g_i^{N,(n+1)}, \end{cases} \tag{14}$$

$n \in \mathbb{N}$, where

$$k! \cdot G_k^N = \frac{d^k}{dx^k} \left(-\alpha_N \cdot g_N \left(g_N \left(-\frac{x}{\alpha_N} \right) \right) \right) \Big|_{x=0}. \tag{15}$$

Fixing some initial approximation $\mathbf{g}_0^N = (g_1^{(0)}, ..., g_N^{(0)})^T$ and $\alpha_N^{(0)}$, we can iterate the system long enough (14) to calculate the approximation $\mathbf{g}^N = (g_1, ..., g_N)^T$ and α_N for the given N.

2.1.2. ***Qualitative considerations on the convergence of this method.*** The equations in the system (14), where the functions G_k^N are present, represent the projection of the first equation from the system (1) onto a finite-dimensional space of dimension N with the basis $\{x^n\}_{n=0}^N$. However, such a projection has a feature of the original functional equation — hyperbolic divergence. When directly integrating only equations with functions G_k^N from the system (14), the iterative process will diverge. This divergence is due to the presence of a one-dimensional dimension in the functional space of unimodal functions, in the direction of which the first equation from the system (1) — the equation for a fixed point g — is not a compressive, but a stretching mapping [1,2,9].

To stabilize the convergence of iterations, additional equalities are introduced (12) derived from the functional equation itself (1). With the transformation of these relations (12) into the form given in the system (14), the iterative process (14) becomes convergent to the desired unknowns.

It is known from computational practice that if you gradually increase the number of N and select each time the initial value obtained from the previous calculation, then a convergent process is obtained for $n \to \infty$ and $N \to \infty$ to the desired g and α. Numerical calculations show that the iterative process converges linearly.

2.1.3. ***Features of function calculations*** G_k^N**.** Formulas in the system (13), where the functions G_k^N participate, do not depend on each other and can be calculated in parallel, however, the calculation of the functions themselves is of some complexity and requires additional methods. Using symbolic calculations, it is possible to precompute derivatives in functions G_k^N,

obtaining algebraic expressions that can already be used in iterations (14). Another approach for calculating functions G_k^N is to use the Faa-di-Bruno [10] formula expressed in terms of Bell polynomials $B_{n,k}$, while calculating the polynomials themselves by recurrent formulas [11].

2.2. The fast Fourier transform method applied to the method of successive approximations.

2.2.1. ***Modification of the Sequential Approximation Method (MPP) using the fast Fourier transform (Fast Fourier Transformation — FFT).*** Calculating the functions G_k^N from the system (13) is a more general problem — calculating the kth derivative of the composition of functions. Let's consider the possibility of using a fast Fourier transform to solve this problem. Let 's introduce the notation

$$F[a_m] := FFT[a_m], \tag{16}$$

where $FFT[a_m]$ — fast Fourier transform of the sequence a_m.

Note that if $f \sim \sum_{n=0}^{N} f_n \cdot x^n$, the Fourier transform of the coefficients f_n can be represented as

$$F[f_n] := \left\{ f\left(\exp\left(-\frac{2\pi n}{N} i \right) \right),\ n = 0, ..., N-1 \right\}. \tag{17}$$

If g_n — coefficients of the representation of the function g in the form of a power series, then the first equation from the system (1), considering it at the points $\left\{ x_n = \exp\left(-\frac{2\pi i}{N} n\right),\ n = 0, ..., N-1 \right\}$ and given the decompositions (5), it will be possible to rewrite as follows:

$$F[g_n^N] = -\alpha_N \cdot g_N\left(F\left[\frac{g_n^N}{\alpha_N} \right] \right), \tag{18}$$

getting from here

$$g_n^N = -\alpha_N \cdot F^{-1}\left[g_N\left(F\left[\frac{g_n^N}{\alpha_N} \right] \right) \right], \tag{19}$$

which, in fact, is a shorter system entry (10). Based on this, we will rewrite the recurrence relations (14) in the following form:

$$\begin{cases} g_1^{N,(n+1)} = -1 - \dfrac{1}{\alpha_N^{(n)}} - \sum\limits_{i=2}^{N} g_i^{N,(n)}, \\ g_k^{N,(n+1)} = -\alpha_N^{(n)} \cdot F^{-1}\left[g_N^{(n)}\left(F\left[\dfrac{g_k^{N,(n)}}{\alpha_N^{(n)}} \right] \right) \right], \\ \alpha_N^{(n+1)} = -\sum\limits_{i=1}^{N} (2i) \cdot g_i^{N,(n+1)}. \end{cases} \tag{20}$$

The equations (20) differ from the system (14) only in the way the functions G_n^k are calculated; this does not affect the convergence of the iterative process itself. The convergence rate of this process is similar to the convergence rate of the system (14).

2.2.2. ***Evaluation of the complexity of the proposed algorithm and comparison with the method of successive approximations and the Newton method.*** Based on the fact that the complexity of the fast Fourier transform algorithm is $O((N) \cdot \log(N))$ [12], the main difficulty in this system (20) will consist in calculating the polynomial $g_N N$-th degree at N points. However, due to the fact that the calculation of the polynomial at different points can

be carried out in parallel, the complexity of the algorithm represented by the relations (20) can be estimated as $O(N) \cdot N/m$ for each iteration, where m — the number of parallel processes. Collectively, the complexity of the algorithm will be $O(N) \cdot N^2/m$, taking into account the linear convergence of the method of successive approximations.

In comparison with other methods of calculating functions G_n^k, the use of the fast Fourier transform is optimal.

In the case of symbolic computation of G_n^k, it is necessary to decompose the composition of N-th degree polynomials at each iteration step, which leads to $O(N!)$ difficulty. This complexity is obviously much higher than $O(N^2)$.

In the case of using the Faa-di-Bruno formula [10], expressed in terms of Bell polynomials $B_{n,k}$, calculation by recurrent formulas [11], the complexity of calculating the functions G_n^k becomes $O(N^3)$. This complexity is also noticeably higher compared to using the fast Fourier transform.

In comparison with Newton's method, the method of successive approximations using FFT has two advantages.

Firstly, in the case of parallel calculation, this method can be linearly scaled, its computational complexity per processor is $O(N^2)/m$, where m is the number of processors in the system. Parallel scaling in the application of Newton's method is much more limited by the need to reverse the Jacobi matrix.

Secondly, the method of successive approximations using FFT requires only the desired vector of coefficients g_k^N to be stored in memory, and the vector of its Fourier transform $F[g_k^N]$, which requires $O(N)$ memory loading computer. In Newton's method, it is necessary to keep the entire Jacobi matrix in memory, which leads to a memory requirement$O(N^2)$.

2.2.3. ***Application of the method of successive approximations with FFT together with the Newton method.*** It is also worth noting that the method of successive approximations with FFT can effectively work in tandem with the Newton method.

According to Newton's method, one additional iteration is necessary to refine the resulting significant digits. Due to the fact that, on average, at each iteration of the Newton method, the number of significant signs doubles, at the stage of the additional iteration, a large number of calculated significant signs are lost. The method of successive approximations with FFT would reduce the loss of a significant number of significant signs, if it is used at the stage of checking the calculated value by the Newton method.

2.3. Numerical-analytical method for calculating functional equations.

2.3.1. ***Description of the algorithm.*** Let us now consider a numerical-analytical algorithm that allows us to obtain solutions of functional equations when the coefficients of the decompositions of the desired solutions themselves are represented as series with respect to a small parameter. Let's introduce the free parameter β into the system of equations (1) by first entering the following expansions

$$\begin{cases} g_\beta(x,\beta) = 1 + g_1(\beta) \cdot x^2 + \sum\limits_{n=2}^{\infty} g_n(\beta) \cdot \beta^{n-1} \cdot x^{2n}, \\ h_\beta(x,\beta) = 1 + h_1(\beta) \cdot x^2 + \sum\limits_{n=2}^{\infty} h_n(\beta) \cdot \beta^{n-1} \cdot x^{2n}, \end{cases} \tag{21}$$

and

$$\begin{cases} g(x,\beta) = 1 + g_1(\beta)\cdot x^2 + \sum\limits_{n=2}^{\infty} g_n(\beta)\cdot x^{2n}, \\ h(x,\beta) = 1 + h_1(\beta)\cdot x^2 + \sum\limits_{n=2}^{\infty} h_n(\beta)\cdot x^{2n}. \end{cases} \tag{22}$$

Unknown coefficients will then also be presented in the form of series

$$\begin{cases} g_n(\beta) = \sum\limits_{k=0}^{\infty} g_{n,k}\cdot\beta^k, \\ \alpha(\beta) = \sum\limits_{k=0}^{\infty} \alpha_k\cdot\beta^k, \\ h_n(\beta) = \sum\limits_{k=0}^{\infty} h_{n,k}\cdot\beta^k, \\ \delta(\beta) = \sum\limits_{k=0}^{\infty} \delta_k\cdot\beta^k. \end{cases} \tag{23}$$

Then, after replacing $x \to x\cdot\alpha(\beta)$, you can rewrite the system of equations (1) as follows:

$$\begin{cases} g(\alpha(\beta)\cdot x,\beta) = -\alpha(\beta)\cdot g_\beta\big(g(x,\beta),\beta\big), \\ \delta(\beta)\cdot h(\alpha(\beta)\cdot x,\beta) = \alpha(\beta)\cdot g'_\beta\big(g(x,\beta),\beta\big)\cdot h\big(x,\beta\big) + \alpha(\beta)\cdot h_\beta\big(g(x,\beta),\beta\big). \end{cases} \tag{24}$$

At $\beta = 1$, due to the analyticity of the functions g and h, the expansions (21, 22), presented as a power series, become solutions of the system (1).

Substituting decompositions (21)–(23) into the system (24) and equating the coefficients obtained, we obtain a chain of equations from which unknown coefficients in decompositions can be calculated sequentially (23). In particular, for $x = 0$, $\beta = 0$ and for the degree of $x^2\cdot\beta^0$ from the first equation of the system (24) we obtain the system

$$\begin{cases} 1 + \alpha_0\cdot(1 + g_{1,0}) = 0, \\ 2\,g_{1,0} + \alpha_0 = 0, \end{cases} \tag{25}$$

the solution of which is

$$\begin{cases} \alpha_0 = 1 + \sqrt{3}, \\ g_{1,0} = -\dfrac{1}{2} - \dfrac{1}{2}\sqrt{3}. \end{cases} \tag{26}$$

2.3.2. ***Derivation of the first coefficients of the desired expansions.*** Calculating the coefficients along the chain, solving a sequence of linear equations, for $\beta = 1$, we obtain the following expansions of constants α and δ:

$$\begin{aligned} \alpha &= 1 + \sqrt{3} + \left(-\frac{1}{12} - \frac{1}{12}\sqrt{3}\right) + \left(-\frac{37}{936} + \frac{53}{1872}\sqrt{3}\right) + \left(\frac{21967}{438048} - \frac{3053}{97344}\sqrt{3}\right) + \ldots, \\ \delta &= -4 - \sqrt{3} + \left(\frac{1}{6} + \frac{5}{9}\sqrt{3}\right) + \left(-\frac{346709}{623376} + \frac{79357}{311688}\sqrt{3}\right) + \\ &\quad + \left(-\frac{224225865065}{126833951088} + \frac{786425631715}{761003706528}\sqrt{3}\right) + \ldots. \end{aligned} \tag{27}$$

We also give decompositions of the first coefficients of the function g

$$
\begin{aligned}
g_1 &= -\frac{1}{2} - \frac{1}{2}\sqrt{3} + \left(\frac{1}{24} - \frac{1}{8}\sqrt{3}\right) + \left(-\frac{265}{1872} + \frac{101}{1248}\sqrt{3}\right) + \left(\frac{113621}{876096} - \frac{124663}{1752192}\sqrt{3}\right) + \ldots, \\
g_2 &= \frac{1}{4}\cdot\frac{3\sqrt{3}+5}{5\sqrt{3}+9} + \frac{1}{72}\cdot\frac{71\sqrt{3}+123}{265\sqrt{3}+459} + \left(-\frac{113}{5184}\cdot\frac{9973081\sqrt{3}+17273883}{13058763\sqrt{3}+22618441}\right) + \ldots, \\
g_3 &= -\frac{1}{12}\cdot\frac{7+4\sqrt{3}}{93\sqrt{3}+161} + \frac{1}{432}\cdot\frac{457975465\sqrt{3}+793236774}{110530697\sqrt{3}+191444783} + \ldots, \\
g_4 &= \frac{1}{3744}\cdot\frac{12327+7117\sqrt{3}}{51409\sqrt{3}+89043} + \ldots,
\end{aligned}
\tag{28}
$$

and functions h

$$
\begin{aligned}
h_1 &= \frac{1}{2} - \frac{1}{2}\sqrt{3} + \left(\frac{13}{24} - \frac{23}{72}\sqrt{3}\right) + \left(\frac{29947}{623376} - \frac{763}{46176}\sqrt{3}\right) + \ldots, \\
h_2 &= \frac{1}{12}\cdot\frac{45+26\sqrt{3}}{97\sqrt{3}+168} + \left(-\frac{1}{648}\cdot\frac{806125404+465416719\sqrt{3}}{50843527+29354524\sqrt{3}}\right) + \ldots, \\
h_3 &= -\frac{1}{12}\cdot\frac{424267\sqrt{3}+734852}{79976509\sqrt{3}+138523377} + \ldots.
\end{aligned}
\tag{29}
$$

2.3.3. ***Features of this method.*** The main difficulty in this method is to calculate the coefficients for the terms $x^j \cdot \beta^i$ after substituting the decompositions (21)–(23) into the system (24). However, due to the possibilities of symbolic calculations, these series were calculated with more than 15 terms, the analytical form of which quickly becomes very cumbersome when the order increases. Numerical calculations have shown the convergence of series (27)–(29) to the known calculated values (3). Calculations also showed that the terms of the series (27)–(29), which are coefficients in expansions (23), at $\beta = 1$ asymptotically decrease as a geometric progression with a denominator $q \approx 2.2$.

It is worth noting that these decompositions (27)–(29) are not unique, and depend on the choice of setting the parameter β in the equations (1). However, the simplest way to set the β parameter, which requires the least amount of calculations, was chosen here. Setting the parameter β in decomposition (21) at higher degrees would lead both to a more complex initial algebraic equation similar to (25) and to more complex calculations for finding higher-order coefficients. At the same time, the convergence of series with the β assignment at higher degrees in the decomposition (21) would lead to faster convergence of the resulting decompositions.

The use of numerical-analytical algorithms was also used earlier by the author for partial differential equations [13].

2.3.4. ***Comparison with other methods.*** A distinctive feature of this numerical-analytical method, in comparison with numerical algorithms, is the possibility of obtaining the exact coefficients of the desired expansions. This feature makes it possible to use various methods to accelerate the convergence of power series that require high accuracy of the values of the expansion coefficients, such as the Pade approximation or the Shanks transform [14, 15]. Convergence acceleration methods allow, based on a small number of calculated decomposition coefficients, to obtain a much larger number of correct signs than those obtained by direct summation of the series.

Also, these decompositions can be useful for the theoretical study of numbers α and δ.

Conclusion

The article proposes three new approaches to solving nonlinear functional equations by the example of solving the Feigenbaum equation.

Two of them are related to the application of the numerical method of successive approximations to a discretized system (10). In the first method, it was possible to write out an iterative scheme for calculating the desired coefficients of decompositions of unknown functions. The main problem in this approach was the calculation of higher-order derivatives of the composition of functions represented by polynomials. In this regard, in the second method, the possibility was proposed to reduce the calculation of higher-order derivatives to a fast Fourier transform.

The third method was a step towards obtaining an analytical representation of unknown functions and constants of a nonlinear functional equation. To do this, an additional parameter β was introduced into the equation in such a way that the coefficients of the desired expansions could be found by sequentially solving linear equations obtained from the power decomposition of this functional equation.

In conclusion, we note that all the methods proposed in this paper have been demonstrated by the example of the Feigenbaum functional equation, but they can also be applied to other nonlinear functional equations with similar properties.

References

1. Schuster HG. Deterministic Chaos: An Introduction. Weinheim: Physik-Verlag; 1984. 220 p.
2. Feigenbaum MJ. Universal behavior in nonlinear systems. Los Alamos Science. 1980;1(1):4–27.
3. Feigcnbaum MJ. The universal metric properties of nonlinear transformations. Journal of Statistical Physics. 1979;21(6):669–706. DOI: 10.1007/BF01107909.
4. Feigenbaum MJ. Quantitative universality for a class of nonlinear transformations. Journal of Statistical Physics. 1978;19(1):25–52. DOI: 10.1007/BF01020332.
5. Briggs K. How to calculate the Feigenbaum constants on your PC. Australian Mathematical Society Gazette. 1989;16:89–92.
6. Broadhurst D. Feigenbaum constants to 1018 decimal places [Electronic resource]. 22 March 1999. Available from: http://www.plouffe.fr/simon/constants/feigenbaum.txt.
7. Briggs K. A precise calculation of the Feigenbaum constants. Mathematics of Computation. 1991;57(195):435–439. DOI: 10.2307/2938684.
8. Molteni A. An efficient method for the computation of the Feigenbaum constants to high precision [Electronic resource]. arXiv:1602.02357. arXiv Preprint; 2016. Available from: https://arxiv.org/abs/1602.02357.
9. Kuznetsov S. Dynamical Chaos. 2nd ed. Moscow: Fizmatlit; 2006. 356 p. (in Russian).
10. Faà di Bruno F. Sullo sviluppo delle funzioni. Annali di Scienze Matematiche e Fisiche. 1855;6:479–480 (in Italian).
11. Bell ET. Partition polynomials. Annals of Mathematics. 1927;29(1–4):38–46. DOI: 10.2307/1967979.
12. Heideman MT, Johnson D, Burrus C. Gauss and the history of the fast Fourier transform. IEEE ASSP Magazine. 1984;1(4):14–21. DOI: 10.1109/MASSP.1984.1162257.
13. Polunovskii AA. Time expansions of equations of mathematical physics. Differential Equations. 2020;56(3):381–391. DOI: 10.1134/S0012266120030106.
14. Van Dyke M. Perturbation Methods in Fluid Mechanics. New York, London: Academic Press; 1964. 229 p.
15. Baker GA, Graves-Morris P. Padé Approximants. 2nd ed. New York: Cambridge University Press; 1996. 746 p. DOI: 10.1017/CBO9780511530074.