UDC 004.942
DOI: 10.15587/1729-4061.2019.170824

EXAMINING A MATHEMATICAL APPARATUS OF Z-APPROXIMATION OF FUNCTIONS FOR THE CONSTRUCTION OF AN ADAPTIVE ALGORITHM

O. Kryazhych
PhD, Senior Researcher
Institute of Telecommunications and Global Information Space
Chokolovsky blvd., 13, Kyiv, Ukraine, 03186
Researcher
Institute of Technology and Business in České Budějovice
Okružní 517/10, 370 01 České Budějovice, Czech Republic
E-mail: economconsult@gmail.com

O. Kovalenko
PhD, Head of Laboratory
Laboratory of Physical and Technical Problems of Nuclear Radiation Sources
Institute for Nuclear Research of National Academy of Sciences of Ukraine
Nauky ave., 47, Kyiv, Ukraine, 03680
E-mail: akovalenko@kinr.kiev.ua

Проведеними дослідженнями запропоновано математичний апарат та методика побудови адаптивного алгоритму, основаного на Z-апроксимації функцій. Це необхідно для вдосконалення підходів до побудови алгоритмів, які змінюють свою поведінку в залежності від зміни вхідної інформації. Зазначене, у свою чергу, значно покращує результати виконання завдання, що реалізуються за допомогою такого алгоритму. Наприклад, рішення нелінійних задач, опис складних поверхонь, пошуку інформації.

Показано, що отримані на цьому рішення узгоджуються із застосуванням однакових алгоритмів для окремих груп функцій, які використовуються для апроксимації. Ці функції використовуються при побудові напряму для пошуку та дають можливість розробити модель погрішності Z-апроксимації з використанням початкових або заключних наближень.

Наведене визначення Z_m-апроксимації, як апроксимації з багатократним зменшенням інтервалу, що призводить до спрощення рекурентних формул, і є особливістю представленого підходу. Запропонована методика та базовий алгоритм дозволяють безпосередньо визначати ряд загальних та гіперболічних функцій з використанням Z_m-апроксимацій та паралельних обчислень. За підсумками досліджень представлений адаптивний алгоритм обчислення arctg x як функції, що є оберненою до tg x.

Представлене може бути використане при створенні адаптивного алгоритму пошуку в масивах неструктурованої та слабо систематизованої інформації. Подібний пошук застосовується для книг та підручників, які були викладені в мережу інтернет у форматах jpeg, pdf, або у вигляді фрагментів обох форматів. У цьому випадку на основі адаптивного алгоритму розробляється спеціальна модель, реалізація якої може бути виконана за декількома варіантами зі зміною напрямів руху

Ключові слова: пошуковий алгоритм, розподіл процесів, рекурентний запис, нев'язка, апроксимація

1. Introduction

Searching for information to solve a problem or to answer a question requires a reasonable selection of aggregate sources based on certain attributes. This follows from the common understanding of information as a scientific category [1] that defines information, knowledge, or data based on specific properties [2]. Using mathematical logic can infer such important properties of information as completeness and consistency, as well as define timeliness as a mathematical model of the event at a point in the time interval. A similar pattern is employed when solving a mathematical problem, particularly challenging when there are a series of limitations and there is a necessity to round data or apply any other transformation. That can lead to the occurrence and accumulation of error. In this case, the solution at a certain point within the interval over which the problem is considered shall hold if it meets the criterion of completeness and consistency.

The search for a solution or receiving a response to information request can be represented as the process of selecting an array of data in two parallel directions. For the first direction, the response is treated as a number and importance of the received information at the current time. The second direction makes it possible, by determining the measure of information entropy by Shannon [3], to obtain the answer of pragmatic value [1].

An interesting fact should be emphasized regarding the above. The number of results and the importance of information received from these sources over at a fixed time point can produce non-relevant information when outside the time period. While the pragmatic value makes it possible to build a logic chain about an object of query, including its development over time. This issue is particularly acute in

Received date: 08.05.2019 Accepted date: 16.06.2019 Published date: 29.06.2019

performing tasks and algorithms that are implemented in the interactive environment, especially in the Internet [4, 5]. Both when solving a mathematical problem and while investigating a certain issue or a task by searching for answers on the Internet, there are limitations [6] that would complicate the search. In this case, specialized algorithms are required for performing a search, which can change their performance during execution depending on the input data, that is, adaptive algorithms. All the above points to the relevance of research aimed at constructing adaptive algorithms.

2. Literature review and problem statement

Study [7] indicated that information can be converted by different systems that receive information from the world and process it for the purpose of revealing patterns and obtaining knowledge. A set of the search images of documents or records of facts (data) is an information array [8] of large volume [9]. The above can be considered the basis of information approach [10], because here the essence of transformation of information into data is revealed, with the emphasis that these are «Big data».

In this context, it should be noted that there was no any separate «problem» to process large volumes of information. That was considered to be a natural development [11] of information systems associated with transferring the technology of collection, transmission, processing, and storage of information into paperless form so that this information is provided for users' requests. Processing large volumes of information taking into consideration a constant increase in the rate of growth and capabilities for using specialized algorithms led to a simple solution. That is, one should distribute processes and separate the functions of information support into individual information flows. Next, to combine these flows based on specific attributes for individual data after processing and to arrange them thematically in databases and distributed automated data banks.

The approach, applied in [12], is based on the criteria of completeness, consistency, and timeliness of the information obtained in terms of the value of information. Completeness of information can be represented as a measure of sufficiency of information for solving a particular problem or as a possibility to submit all meanings of truth from a set of logical operations using the formulae from the elements of this set [2]. Consistency can be defined as a property of the system from which one cannot derive the notion of inconsistency [13] when one set of parameters contradicts another one. In logical terms, it can be represented as a mistake [2] or ambiguity. Timeliness can denote receiving information at convenient or set time [12]. It can be represented, in line with approach from [14], as a mathematical dependence of information on the period of time over which this information is relevant. Then, given the above, the value of information can be represented via a category of relevance. However, the relevance should be considered in this case not as a match between the information and a request [15], but the possibility to provide complete and consistent information within a change in time.

This is exactly what is defined in [16] as the adaptation of any viable system to changes in the environment. However, in contrast to [7], whose authors indicate the need for a clear mathematical apparatus to automate the processing of information, paper [16] gives a schematic visualization, without a mathematical justification [17].

However, computing equipment perceives the process of handling information in a formalized manner. That is why the approach from [18] is widely used in the world, which possesses a proven mathematical apparatus for constructing algorithms, including those for search.

When searching for a solution or to find specific information on the Internet, a path to the answer may not always be described linearly. Often, answers are arranged along a certain curve described by a function or a series of functions. The purpose of the search is to find a specific set of variables with the transition between some points, sources of information, in line with a specific algorithm that finds information in the defined data structure and depends directly on the structure of data for which it is implemented [19]. Such a search can be implemented [20] by the method of possible directions by G. Zoutendijk [21], when one obtains a certain number of pole points which provide information in compliance with the predefined restrictions. However, there are problems when limits may be different depending on the conditions for their implementation. For example, the mathematical description of the convex or concave surface can be resolved using the G. Zoutendijk method. Or, when searching the Internet when the result is expected to be both structured and non-structured or poorly structured information, and the algorithm must adjust the search depending on the data type. In this case, an adaptive algorithm could be more universal by adding the new points-sources of information depending on the distribution of input data; given this, of interest is the use of Z-approximation, described, for example, in [22].

The method of Z-approximation of functions is originally based on adaptive algorithms capable of changing their functional features and providing variable computing accuracy at the same time. In addition, a special feature of Z-approximation is a possibility to change the structural patterns of the algorithm itself through the initial and final approximations and own parameters of the algorithm.

An analysis of results of the above studies allow us to assume that existing approaches to solving those problems that require handling the data of different structure are not enough to obtain a relevant response to the question posed. Recently, these issues have been mostly inherent on the Internet. This is due to the accumulation of large volumes of unstructured and poorly systematized information. And this leads to the limitation in the search [6] and complications in the formation of an array of responses to a query, to incompleteness and inconsistencies in such answers.

Specified part of the task could be solved by organizing the process of solving problems based on the construction of models to search for solutions based on adaptive algorithms.

3. The aim and objectives of the study

The aim of this work is to analyze the mathematical apparatus of Z-approximation of functions for constructing an adaptive algorithm, that is, the algorithm that changes its execution progress depending on the received input data or the data refined at the previous stages of implementation. Specifically, tasks on the implementation of adaptive algorithms are performed during search – finding a solution to functions, selecting the best variant (minimax problem), matching the data acquired to the predefined constraints when searching for a text, etc.

To accomplish the aim, the following tasks have been set:

– to analyze the requirements put forward to the construction of adaptive algorithms;

– to prove the efficiency of application of Z-approximation for the construction of adaptive algorithms based on the procedure proposed in the current work;

– to present the basic procedure for constructing an adaptive algorithm based on Z-approximation.

4. Requirements to adaptive algorithms

Suppose that there is a certain built model to search for a solution whose key values are at different points of the coordinate system. Between these points, the information sources, the desired solutions are likely. It is possible to represent a transition to searching for these solutions by small segments along a broken curve, which, resulting from smoothing, are described using the trigonometric and hyperbolic functions, exponent, logarithm. Then, given the above and the features of algorithms [23] that can change depending on input parameters, one can outline the basic requirements to constructing an adaptive algorithm in order to implement the described model of searching for a solution:

– the algorithm should provide a possibility for a recurrence record, that is, compute values based on the preceding terms in a sequence;

– the constants that are used in such algorithms should be represented either by a small number of digits or easily computed at arbitrary accuracy;

– a possibility to replace the initial or final approximations, that is the chosen adaptation mechanism for different types of query.

To fulfill the above, one can use the same algorithms for separate groups of functions that are employed. For example, one cannot use some of the methods that have less than $n/2$ constants at arbitrary digit size for the calculation of direct and inverse trigonometric and hyperbolic functions, exponent, and logarithm. Computing such constants at arbitrary accuracy requires a certain time and respective hardware, which no longer meets the requirement for work with different hardware. However, functions of the type y/x, $1/x$, $\sqrt{x}$, $\sqrt[3]{x}$ do not require calculation of such constants and they can be treated by a single separate calculation algorithm.

One should also take into consideration that the requirement regarding the constants should be met both in the algorithm for the initial and final approximations. Based on this, the following methods are more suitable to perform the tasks set in the current study and to implement an adaptive algorithm:

– recurrence record of expansions into series. In this case, the initial approximation y_0 and respective x_0 should be represented as constants or expressions that are easy to calculate;

– recurrence record of expansions for polynomials. In some cases, it is rational, though not always easy when implementing the programming of such an algorithm;

– calculation of the continued fraction, including the expansion with residuals, which is rather convenient for representation in the form of a database table;

– an interesting method to implement the search is the computation of values based on the preceding terms in a sequence to calculate the infinite works;

– methods of Z-approximation, which are also interesting for the implementation of search when one uses a predefined large set of methods for initial and final approximations and there is a possibility to choose the ratio of complexity of the recurrent relationship and the original or final approximation;

– regular iterative formulae obtained in advance. Again, they are convenient as there is a large set of methods for obtaining iterative formulae and initial approximations that can be represented in a database table.

The specified methods can be implemented in algorithms with a sequentially-parallel architecture.

All the methods above have a significant advantage – a possibility to reduce the time of computing through the use of a variable computing accuracy. From this perspective, the most interesting are the methods of Z-approximation, under which each next iteration increases the accuracy of calculations.

5. Efficiency of the application of *Z*-approximation for constructing adaptive algorithms

To prove the effectiveness of the application of Z-approximation and adaptive algorithms on this basis, one can consider the model of error in Z-approximation of a function using the initial or final approximations. In general, such a model can be recorded in the form:

$$\Delta \approx C_n \left(x/N^m\right)^n \approx \left\lceil x/N^{mn+l} \right\rceil,$$

where C_n are the constants, which mostly depend on parameter n; N is a number that defines the magnitude of reduction in the interval; m is the number of iterations, performed based on a recurrent formula; n is the ordinal number of a term in the expression, which is deducted at when executing approximations $l=-\log_N |C_n|$.

Based on this model, one can determine the values for parameters m and n.

It should be noted that for many functions, such as e^x, $\sin x$, $\cos x$, and similar, the value C_n would provide for a significant contribution to the reduction in the method's errors. However, among some values for n and m, a greater impact will be exerted by magnitude N^{mn}. This is caused by the fact that, for example, in the case when the coefficient is proportional to magnitude $1/(n+m)!$, then, from certain values for n and m, function N^{mn} grows faster than $(n+m)!$. Thus, for $N=2$, $n=4$, $m=4$, one will obtain $(n+m)!=40{,}320$, $2^{mn}=65{,}536$, and at $m=10$, $n=4$, $(n+m)!=87{,}178{,}291{,}200$, $2^{mn}=1{,}099{,}511{,}627{,}776$.

In addition, it should be noted that the use of Z-approximation is equivalent to multiple reduction in the interval, proportional to the value for magnitude $1/N^m$. Increasing the magnitude N leads to complications of recurrent formulae based on Z-approximation, (hereinafter as Z_m). In addition, increasing N increases the possibility for the parallelization of recurrent relationships based on Z_m-approximation. To explore this, the original model of error can be represented in the form:

$$\Delta \approx C_n' \left(x/\left(N+\Delta N\right)^{m+\Delta m}\right)^{n+\Delta n},$$

where ΔN, Δm, Δn are the increments of respective parameters N, m, n.

Logarithm of this expression:

$$\ln\Delta \approx \ln C_n' + \left(n+\Delta n\right)\left(\ln x - \left(m+\Delta m\right)\ln\left(N+\Delta N\right)\right).$$

It follows from the latter that increasing the magnitudes of Δm and Δn affects a decrease in the error Δ. However, at consecutive and parallel calculations the algorithms for the choice of parameters will differ. Therefore, it should be noted that increasing the parameter m leads to the increase in the number of iterations based on the recurrent formula and to a larger error. Increasing the parameter n would increase the number of terms at the initial or final approximation, as well as decrease C'_n. Therefore, for each fixed N we must choose the ratio of m to n.

One must enter the concept of a base sequence for recurrence ratios. A base sequence of recurrence ratios refers to the sequence of recurrent formulae, whose p-term has a parameter n_p greater than n_{p-1} in expression $Z(nx_m)=f[Z(x_m)]$, where $x_m=x/n^m$.

Here are examples of such sequences.

For computing the function $\sin x$:

$$Z_{m-1}=\sum_{k=0}^{n}(-1)^k C_{2n+1}^{2k+1} Z_m^{2k+1}\left(1-Z_m\right)^{n-2k},$$

where

$$Z_m=\sin\left[x/(2n+1)^m\right],\ \ m=m_0,\ \ m_0-1,\ \ldots,0,$$

hence, $Z_0=\sin x$.

This can be represented in the simplest form in general:

$$Z_{m-1}=T_{2n+1}\left(Z_m\right),$$

where $T_{2n-1}(Z_m)$ is a Chebyshev polynomial.

One can use the Euler's method to compute $\operatorname{tg} x$ [24]:

$$Z_{m-1}=\frac{nZ_m}{1-}\,\frac{\left(n^2-1\right)Z_m^2}{3-}\,\frac{\left(n^2-4\right)Z_m^2}{5-\ldots}\,\frac{\left(n^2-p\right)Z_m^2}{-2p+1-\ldots},$$

where $Z_m=\operatorname{tg}x/n^m$, $Z_0=\operatorname{tg}x$.

Similarly, we can write recurrence ratios for the generated by fractional-rational approximations.

To compute $\operatorname{ctg} x$:

$$Z_m=\frac{\sum_{k=0}^{[n/2]}(-1)^k C_n^{2k} Z_{m-1}^{n-2k}}{\sum_{k=0}^{(n-1)/2}(-1)^k C_n^{2k+1} Z_{m-1}^{n-2k-1}},$$

where $Z_m=\operatorname{ctg}x/n^m$, $Z_0=\operatorname{ctg}x$.

Using [25], one can accept the assumption on a rounding error: when the recurrent relation is used for function $y=e^x$:

$$Z_m=2Z_{m-1},\ \ m=\overline{n,1},$$

then the absolute error that is caused by a rounding error will equal:

$$\Delta_n=2^n\Delta_0\left(e^{x/2^n}+\overline{\Delta_0}\right)^{2n-1},$$

where $\left|\overline{\Delta_0}\right|\le\Delta_0$, and the magnitude x is determined from equation $e^x=2^p l^x$.

That is, the error may exceed the magnitude 2^{n-t-1}, where t is the digit size of numbers. If we consider, instead of function $y=e^x$, the function $v=e^x-1$, then the magnitude for an error will not exceed the magnitude $O(n2^{-t})$.

The same applies to functions $\cos x$, $\ln x$, $\operatorname{arctg} x$. Other trigonometric functions do not need such transforms that follows, in particular, from [26].

To construct adaptive algorithms using Z_m-approximation, one should introduce a mathematical representation of Z_m-function. A Z_m-function denotes a direct or inverse recurrence ratio in the following form:

$$Z_{m+1}=f\left(Z_m\right) \qquad (1)$$

or

$$Z_{m-1}=f\left(Z_m\right). \qquad (2)$$

(1) assigns some initial approximation Z_0 while function Z_m acts as a desired function. In case (2), some initial approximation Z_{m0}, is assigned, while function Z_0 acts as a desired function.

Formula (2) can be obtained from expression:

$$Z\left(x_m/n\right)=f\left[Z\left(x_m\right)\right] \qquad (3)$$

by substituting:

$$x_m=x/n^m, \qquad (4)$$

when Z_m is determined:

$$Z_m=Z\left(x_m\right), \qquad (5)$$

where n can be represented as the basis of the calculation system.

Formula (5) can be obtained from expression:

$$Z\left(nx_m\right)=f\left[Z\left(x_m\right)\right] \qquad (6)$$

by replacing (4) at the adopted determination (5).

It is possible to take, as the initial approximation for (2), several terms from the expansion of function Z_m into a Taylor series.

To assess the absolute error of the method, formulae (1) or (2) must be substituted with expression:

$$Z_m=\overline{Z_m}+\overline{\Delta_m}, \qquad (7)$$

where $\overline{Z_m}$ is the exact value for Z_m; $\overline{\Delta_m}$ is the absolute error of Z_m.

Ultimately, we obtained an estimate:

$$\overline{\Delta}_{m+1}\le k_1\overline{\Delta}_m \qquad (8)$$

or

$$\overline{\Delta}_{m-1}\le k_2\overline{\Delta}_m, \qquad (9)$$

where k_1, k_2 are constants.

It follows from expression (8) that the assessment of absolute error of the method for the case of a direct recurrence sequence takes the form:

$$\overline{\Delta_m}\le k_1^m\overline{\Delta}_0. \qquad (10)$$

We can conclude from expression (9) that the assessment of absolute error of the method for the case of an inverse recurrence sequence takes the form:

$$\overline{\Delta_0} \le k_2^m \overline{\Delta}_m. \qquad (11)$$

And considering that there is a certain error in the initial data and some assessment of this error, which can be represented as:

$$\overline{\Delta}_0 \le p,$$

where p is the constant, we obtain the following from (10):

$$\overline{\Delta_m} \le k_1^m p, \qquad (12)$$

and for the inverse recurrence sequence from (11):

$$\overline{\Delta}_0 \le k_2^m p.$$

In a general case, Z_m-approximation can be represented in the form:

$$Z(\varphi(x_m)) = f[Z(x_m)],$$

where $x_m = \varphi^{-1}(x)$, $\varphi^{-1}(x)$ is the function inverse to function $\varphi(x)$.

For example, for function $y=\ln x$, by performing m times the operation of extraction of root with power n, we obtain $\lim\limits_{m\to\infty} x_m \to 1$, that is, $\ln x_m = 0$ at $m \to \infty$.

In this case, $\ln x = n^m \ln x_m$.

Therefore, one can record:

$$\ln x_{m+1} = n \ln \sqrt[n]{x_m},$$

that is, $Z_{m+1}=nZ_m$.

However, in the latter case, there is a possibility to greatly accumulate the error. But for the case $n=2$ one can use formula:

$$\ln(1+x) = 2\ln(0+x/1+\sqrt{1+x}).$$

Based on the above, we can deduce a series of methods for obtaining recurrence formulae to calculate the series of functions using Z_m-approximation, for example, for the most widely used $\cos x$, $\sin x$, a^x and certain hyperbolic functions.

6. Basic procedure for constructing an adaptive algorithm based on Z_m-approximation

An algorithm will be implemented correctly in the case when for any input data belonging to the assigned region, the solution is unique and stable. Formally, this can be represented as follows.

Assume that in order to solve a problem there is some algorithm A that ensures obtaining a value y, which matches the input data x. That is:

$$y = A(x).$$

For the case when the input data were assigned at a certain error Δx, the value for $x+\Delta x$ is matched with value:

$$y + \Delta y = A(x + \Delta x)$$

or

$$\Delta y = A(x + \Delta x) - A(x).$$

The algorithm will be defined as robust in terms of input data if $\|\Delta x\| \to 0$ will always provide for $\|\Delta y\| \to 0$. Otherwise, the algorithm will be unstable in terms of input data.

If the algorithm is unstable, even minor errors Δx would lead to significant errors Δy, which would ultimately lead to a distorted result. In this case, one can define the global and local robustness. For the case of global robustness, the process would always be stable regardless of errors in individual operations in the course of the algorithm execution. The local robustness is observed in the case when there is an error that exceeds some critical value. During algorithmizing, especially in the processes of information search, weakly stable tasks emerge. This corresponds to the case $\|\Delta y\| \le M\|\Delta x\|$, when the constant M is large enough. Although the algorithm is formally robust, error could reach an unacceptably large value.

Our analysis of errors in Z_m-approximations of functions has shown that in the case of application of recurrence formulae of the type $Z_{m-1}=f(x,Z_m)$, on observes robustness in terms of initial data. The error of the method considering initial data is within $|\xi_0| \le M^m L < C$, where C is a constant, M is determined based on $|\Delta_{m-1}| \le M|\Delta_m|$, and L – from the error of initial approximation $|\xi_{m_0}| \le L$.

Based on the obtained estimates, one can choose for each specific case the parameters m and n to obtain the required accuracy in intermediate calculations and to weaken the influence of rounding errors. This is a theoretical proof of using the methods of Z_m-approximations for constructing adaptive algorithms.

The adaptive Z_m-algorithm implies a certain general algorithm that makes it possible to produce two or more separate algorithms. Moreover, each of these algorithms is quite effective for solving specific tasks based on the computation of Z_m-function.

The adaptive Z_m-algorithm is built by adding/removing parameters for certain values, and by using several kinds of approximations of the desired functions.

An example of the construction of such an algorithm is the following algorithm for computing $\operatorname{tg} x$ and $\operatorname{arctg} x$ at arbitrary bit size. Use the algorithm to compute $\operatorname{arctg} x$ based on recurrence formula:

$$Z_{m+1} = Z_m \Big/ \left(1+\sqrt{1+Z_m^2}\right),$$

where $Z_1 = x$, $\operatorname{arctg} x \approx 2^m Z_{m+1}$, $Z_m = \operatorname{tg}(x/2^m)$ is impractical due to calculating the square root m times.

Therefore, the algorithm for computing $\operatorname{arctg} x$ should be constructed as a function inverse to $\operatorname{tg} x$ applying an iterative formula:

$$y_{i+1} = y_i - \sum_{k=0}^{p} (-1)^k Z_i^{2k+1} \big/ (2k+1), \quad Z_i = \frac{\operatorname{tg} y_i - x}{1 + x \operatorname{tg} y_i}.$$

The computing Z_m-algorithm can be represented as follows:

1. Compute y_0 – the value for $\operatorname{arctg} x$, computed at fixed bit size.

2. $p=0$, $p=1$, $p=2$, $p=3$.

If $p=0$, we obtain an iterative formula of third order, that is, the accuracy of the calculation improves. For case $p=3$, we obtain an iterative formula of fifth order and the order could further grow. However, in this case, to compute Z_i, it is necessary to compute the inverse function tg y_i of an arbitrary order.

3. Inverse function tgy_i – in a general case, it is calculated using recurrence ratio:

$$Z_{m-1} = T_{2n+1}(Z_m)/T_{2n}(Z_m),$$

where $T_n(Z)$ is the Chebyshev polynomial;

$$Z_m = \mathrm{tg}\left(y_i/N^m\right), \quad N = 2n+1, \quad n = 0,1,2..., \quad Z_0 = \mathrm{tg}\, x.$$

4. Initial approximation. It can be accomplished based on three separate algorithms.

4. 1. As an initial approximation Z_{m_0} we use an appropriate number of terms sin y_m and cos y_m, that is:

$$\mathrm{tg}\, Z_m = \frac{\sin y_m}{\cos y_m} = \frac{\sum\limits_{k=0}^{\infty} (-1)^k y_m^{2k+1}/(2k+1)!}{\sum\limits_{k=0}^{\infty} (-1)^k y_m^{2k}/(2k)!},$$

where $y_m = y_i/N^m$.

4. 2. When performing in this part of the algorithm the sequential calculation of sin y_m and cos y_m, the computation is based on recurrence formulae:

$$u_{k+1} = -u_k y_m^2/2k(2k+1), \quad t_{k+1} = t_k + u_{k+1}, \quad u_1 = t_1 = y_m,$$

$$v_{k+1} = -v_k x_m^2/(2k-1)2k, \quad Z_{k+1} = Z_k + v_{k+1},$$

$$Z_1 = v_1 = 1, \quad k = 1,2,...$$

4. 3. It is possible to directly calculate the initial approximation Z_{m0} also in the form of as a continued fraction:

$$Z_{m_0} = \mathrm{tg}\, y_m = \frac{y_m}{1-}\,\frac{y_m^2}{3-}\,\frac{y_m^2}{5-}\cdots\frac{y_m^2}{-2n+1}.$$

5. Another separate algorithm for computing arctg x according to the following formula:

$$\mathrm{arctg}\, x = y_0 - \mathrm{arctg}\, Z_0 = y_0 - \left(\frac{Z_0}{1+}\,\frac{Z_0^2}{3+}\,\frac{4Z_0^2}{1+}\cdots\frac{n^2 Z_0^2}{+2n+1}\right),$$

where $Z_0 = (\mathrm{tg}\, y_0 - x)/(1 + x\ \mathrm{tg}\, y_0)$.

5. 1. The given continued fraction can be computed based on recurrence ratios:

$$R_{n-1} = (A_n - B_{n-1})Z_0^2/\left((B_n - 2) + R_n\right), \quad n = k, \quad k-1,..,\ 2;$$

$$\mathrm{arc\,tg}\, x = y_0 - Z_0/(1 + R_1),$$

where $A_n = n^2$, $B_n = 2n+1$.

This algorithm is complete.

The reported algorithm makes it possible to represent the formalization of a search problem in the array of unstructured information. The problem can be stated in the terms of fuzzy logic and solved at steps, the transition between which in the coordinate system can be described by elementary and/or trigonometric functions:

1) A starting point for the search is the title of a publication, for example, monograph «ABC».

2) Choose the terms of linguistic variables that correspond to subset A.

3) For each term, choose the value that best characterizes it.

4) Proceed to subset B and choose the identical terms there, strictly following the commutativity rule on the terms from sets A and B.

5) Search and sort out appropriate values from the characteristics chosen at stage 4 in subset B assigning «1» or «0» to each obtained value.

6) Upon obtaining extreme values, choose the intermediate values, the steps to which may occur not in a straight line by describes by various functions.

7) Assign «1» or «0» to each obtained intermediate value.

8) All values are assigned with the relevant functions of standard membership.

9) Form the information array to search.

10) Define production search rules sorting out the values similar to step 5.

When working with Google Scholar, production search rules can be specified as follows:

– If monograph «ABC»=ISBN 978-966-0000-00-0;
– If monograph «ABC»=BBK (number);
– If monograph «ABC»=UDC (number);
– If author Z=Scientific institution Q;
– If scientific institution Q=Decision of the Academic Council (number, date) and others, which are performed sequentially, including the substitution of the left and right sides, and sorting out individual data (in particular, parts of numbers and names).

A search result is predetermined by its matching at least a single production rule or a correspondence between parts of production rules and search.

In fact, when solving this problem, we complement the fuzzy subset A in the set X with a subset $\neg B$ with a membership function:

$$\mu_{\neg B}(x) = 1 - \mu_{\neg A}(x), \quad \forall x \in X.$$

Function $\mu_{(A:B)} : X \to [0;n]$ is the membership function of subset A (B) to the base set X (Google).

The proposed approach makes it possible to find the fragments, appropriate in terms of structure, from an array of unstructured information, where a single record contains, at the same time, the information of different structures, which significantly increases the number of variants for obtaining an accurate answer.

The approach reported was implemented when searching for references to monograph [27] at Google Scholar. The book was available at one of the web sites on the Internet in a .pdf format as a single file, which contained the non-structured information in the form of a picture with fragments of text fields with text symbols without semantic content. The book was referenced several times that were not displayed in the profile of scientists at Google Scholar.

From the entire array of unstructured information, we have chosen the second page for the implementation of the algorithm (Fig. 1), which includes the codes of UDC, BBK, ISBN that can clearly identify the publication as authentic, as well as the authors' surnames and the title of the scientific institution, which makes it possible to expand a query to search and additionally confirm compliance with query results if the listed codes are presented only as an element of graphic information.

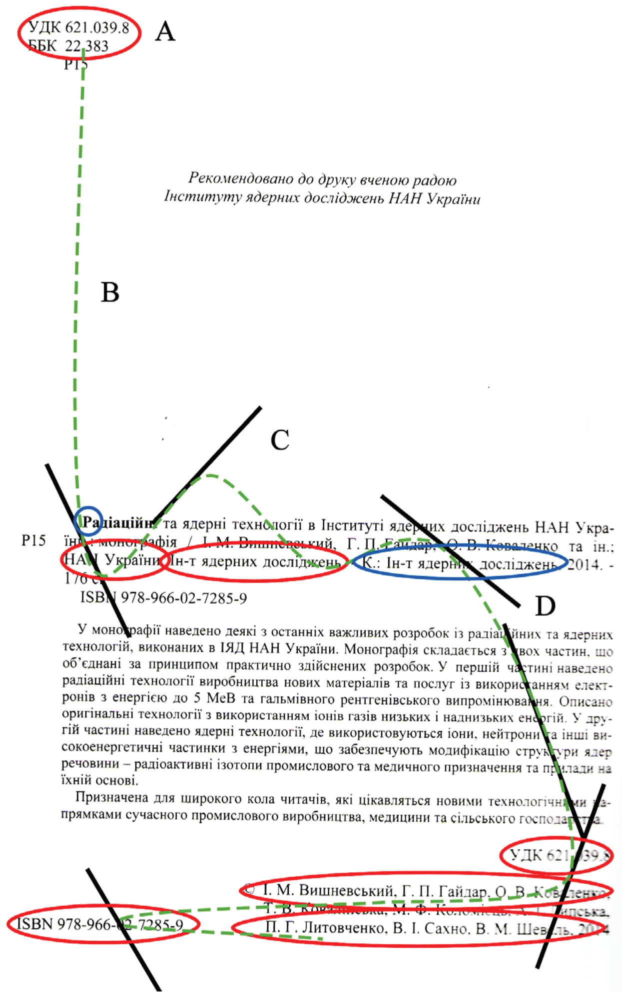

Fig. 1. Search algorithm for an array of unstructured information: A – starting point; B – built function; C – vector lines; D – contours of results

The above is taken as the base points (highlighted in red) from the top edge of the page, with starting point A. The page is conditionally divided into rectangular cells. The smaller the side of the cell, the more accurate is the search based on steps to get the pole points, which in this study are represented by a word or a word fragment. The dotted line B represents the built function along the directions given by vectors C. These directions are used to calculate, at the pole points with selected words to search, the values for a transition that implies the repeated steps of the algorithm. In a given example, the «negative value» is the absence of a word or a word fragment in the cell. In this case, the choice of a word for forming the query along the direction stops.

Next, we generated a series of queries using the rules for constructing adaptive algorithms, which consisted of a word or a code at each base point with words, word, or word fragment at pole points belonging to the region of this base point. These queries were used to construct a sampling of the best result, which contained the words or codes that made it possible to identify the compliance of reference to the monograph.

The result, based on which we found the references and identifed the monographs, is shown by contours D. This algorithm found six previously missing links.

7. Discussion of results of studying the construction of an adaptive algorithm

Everybody knows that functions can be computed using known expansions of trigonometric functions, as stated in [25, 26]. However, at algorithmizing and machine computation of complex calculations involving the construction of graphs, there is a task on dividing curves into segments [21], on smoothing and finding the points [20]. In such cases, one must comply with certain accuracy. And the proposed algorithm makes it possible to apply different approaches to computing while maintaining maximum accuracy.

Based on the reported algorithm, one can directly determine $\sin x$, $\cos x$, $\mathrm{tg}\, x$ by determining $y_m=x$, or $\mathrm{tg}\, x$ using Z_m-approximations. And it is possible to calculate $\mathrm{arctg}\, x$ by applying this algorithm when using an iterative formula for computing inverse functions.

This general algorithm can be extended with parallel computation of $\mathrm{tg}\, x_m$ as the ratio of two sums, as well as by the parallelized computing of iterative formula $\mathrm{arctg}\, x$.

Similarly, other functions can be represented, for instance, hyperbolic.

The reported mathematical apparatus greatly facilitates the implementation of the method in the form of the algorithm, in contrast to [26]. This is explained by the fact that a series of functions that are used in the calculations may be assigned in a tabular form and represented in a database. Then, when executing an adaptive algorithm, part of the solutions related to the calculation direction, if they meet the initial conditions, are taken from the table.

In the future, a similar method could be implemented for the search algorithm. The most interesting task for the implementation of such an algorithm is to search for unstructured and poorly organized information. For example, when digitizing books one can obtain documents in the formats jpeg, pdf, or as fragments of both formats. That considerably complicates the subsequent search [6]. This leads to that many interesting sources of information are lost in the large arrays of information on the Internet. An adaptive algorithm to search for such documents can be implemented through the development of a special model whose realization implies several variants. Specifically, based on the method of possible directions by G. Zoutendijk [21]. However, when using this method, there are the non-linear constraints that can be circumvented, time and again, by using approximations. That is, by the implementation of distributed adaptive algorithm for solving individual search tasks. So, in general, the search model would represent an adaptive algorithm that could build a path between the keywords specified for search. However, the comparison would be based on characters or lexemes found at the points of solution to the function that describes the search path.

8. Conclusions

1. In the proposed approach, we defined and proved the requirements to adaptive algorithms that could be practically used in applications and Web developments. Noteworthy

is the capability to use the same algorithms for individual groups of functions used for approximation in the construction of a search direction.

2. We have proven the efficiency of application of Z-approximation and adaptive algorithms on this basis to solve the task on adaptive search by constructing the model of error in Z-approximation of function using the initial or final approximations. We give the definition of Z_m-approximation as the approximation with a multiple frequent reduction of interval proportional to the value of magnitude $1/N^m$, which is a special feature of the proposed approach.

3. The basic procedure for constructing an adaptive algorithm based on Z_m-approximation has been proposed; the general Z_m-algorithm is provided using an example of computing tg$x$ and arctg$x$ at arbitrary bit size. Based on the presented algorithm, we have shown a possibility to direct determine a series of general and hyperbolic functions using Z_m-approximations and parallel computing.

References

1. Entsiklopediya kibernetiki / V. M. Glushkov et. al. (Eds.). Kyiv: Glavnaya redaktsiya ukrainskoy sovetskoy entsiklopedii, 1974. 1228 p.
2. Filosofskyi entsyklopedychnyi slovnyk. Kyiv: Abrys, 2002. 742 p.
3. Shennon K. Raboty po teorii informatsii i kibernetike. Moscow: Inostrannaya literatura, 1963. 832 p.
4. Searching the web: The public and their queries / Spink A., Wolfram D., Jansen M. B. J., Saracevic T. // Journal of the American Society for Information Science and Technology. 2001. Vol. 52, Issue 3. P. 226–234. doi: https://doi.org/10.1002/1097-4571(2000)9999:9999<::aid-asi1591>3.0.co;2-r
5. Figueroa A. Exploring effective features for recognizing the user intent behind web queries // Computers in Industry. 2015. Vol. 68. P. 162–169. doi: https://doi.org/10.1016/j.compind.2015.01.005
6. Ashmanov I., Ivanov A. Optimizatsiya i prodvizhenie saytov v poiskovyh sistemah. Sankt-Peterburg: Piter, 2011. 464 p.
7. Glushkov V. M. Kibernetika. Vychislitel'naya tekhnika. Informatika. Izbrannye trudy. Vol. 3. Kibernetika i ee primenenie v narodnom hozyaystve / V. S. Mihalevich et. al. (Eds.). Kyiv: Naukova dumka, 1990. 224 p.
8. Glushkov V. M. Kibernetika. Voprosy teorii i praktiki. Moscow: Nauka, 1986. 488 p.
9. Glushkov V. M., Stogniy A. A., Afanas'ev V. N. Avtomatizirovannye informatsionnye sistemy. Moscow: Znanie, 1973. 64 p.
10. Akademik V. M. Glushkov – pioner kibernetiki. Kyiv: Izdatel'stvo Yunior, 2003. 384 p.
11. Glushkov V. M. Osnovy bezbumazhnoy informatiki. izd. 2-e, ispr. Moscow: Nauka, Gl. Red. fiz.-mat. lit., 1987. 552 p.
12. Pleskach V. L, Zatonatska T. H. Informatsiyni systemy i tekhnolohiyi na pidpryiemstvakh: pidruchnyk. Kyiv: Znannia, 2011. 718 p.
13. Encyclopaedia of Mathematics (set) / M. Hazewinkel (Ed.). Springer, 1994.
14. Kovalenko O. V. Kontseptualni osnovy stvorennia bazy danykh naukovoho eksperymentu ta sposterezhennia // Matematychni mashyny i systemy. 2016. Issue 2. P. 91–101.
15. Diaz F. Autocorrelation and Regularization of Query-Based Retrieval Scores. Chap. 3: PhD thesis. Amherst, 2008.
16. Beer S. Brain of the Firm, Allen Lane. London: Herder and Herder, 1972. 416 p.
17. Beer S. Designing Freedom. House of Anansi Press, 1993. 110 p.
18. Algorithms and Data Structures. URL: https://people.inf.ethz.ch/wirth/AD.pdf
19. Knuth D. E. The Art of Computer Programming. Vo. 1. Fundamental Algorithms. 3-rd ed. Reading, Massachusetts: Addison-Wesley, 1997. 664 p.
20. Kriazhych O. O., Kovalenko O. V., Ivanchenko V. V. Sposib opysu zabrudnenoi terytoriyi: prohramna realizatsiya // Matematychne modeliuvannia v ekonomitsi. 2016. Issue 2. P. 22–35.
21. Zoutendijk G. Methods of feasible directions. A study in linear and non-linear programming. Elsevier Pub. Co., 1960. 178 p.
22. A Resolver-to-Digital Conversion Method Based on Third-Order Rational Fraction Polynomial Approximation for PMSM Control / Wang S., Kang J., Degano M., Buticchi G. // IEEE Transactions on Industrial Electronics. 2019. Vol. 66, Issue 8. P. 6383–6392. doi: https://doi.org/10.1109/tie.2018.2884209
23. Knuth D. E. The Art of Computer Programming. Vol. 3: Sorting and Searching. 2-nd ed. Reading, Massachusetts: Addison-Wesley, 1998. 780 p.
24. Eyler L. Integral'noe ischislenie. Vol. 1. Moscow: GITTL, 1956. 416 p.
25. Bahvalov N. S., Zhidkov N. P., Kobel'kov G. M. Chislennye metody. 6-e izd. Moscow: BINOM. Laboratoriya znaniy, 2008. 636 p.
26. Popov B. A., Tesler G. S. Priblizhenie funktsiy dlya tekhnicheskih prilozheniy. Kyiv: Naukova dumka, 1980. 352 p.
27. Radiatsiyni ta yaderni tekhnolohiyi v Instytuti yadernykh doslidzhen NAN Ukrainy: monohrafiya / Vyshnevskyi I. M., Haidar H. P., Kovalenko O. V. et. al. Kyiv: In-t yadernykh doslidzhen, 2014. 176 p.