http://doi.org/10.35784/iapgos.7641

*received: 25.04.2025 | revised: 21.11.2025 | accepted: 02.12.2025 | available online: 30.06.2025*

# DEVELOPMENT OF AN ALGORITHM FOR CALCULATING ION EXCHANGE PROCESSES USING THE PYTHON ECOSYSTEM

**Iryna Chub[1], Oleksii Proskurnia[2], Kateryna Demchenko[1], Oleksandr Miroshnyk[3], Taras Shchur[4], Serhii Halko[5]**

[1]State Biotechnological University, Department of Automation and Computer Integrated Technologies, Kharkiv, Ukraine, [2]H.S. Skovoroda Kharkiv National Pedagogical University, Department of Mathematics, Kharkiv, Ukraine, [3]State Biotechnological University, Department of Electricity Supply and Energy Management, Kharkiv, Ukraine, [4]Department of Agricultural Engineering, State Biotechnological University, Kharkiv, Ukraine, [5]Dmytro Motornyi Tavria State Agrotechnological University, Department of Electrical Engineering and Electromechanics Named after Prof. V.V. Ovharov, Zaporizhia, Ukraine

***Abstract.*** *Ion exchange plays a key role not only in eliminating water hardness and reducing the concentration of unwanted ions, but also in protecting the environment. Due to its versatility, it is an indispensable tool in modern environmental technologies. Calculation methods are an important component of the implementation of ion exchange technologies, as they play a crucial role in their effective operation. There are calculation methods used in the design of ion exchange systems, but similar methods have not yet been developed to determine operating parameters during operation. This complicates the optimization of the process in real conditions and requires further research and development. The calculation method should be easy to use, but at the same time provide high accuracy of results, taking into account such important factors as water composition, filtration rate, temperature and type of ion exchange resins. It is especially important that the method be developed based on open Python libraries to ensure accessibility and versatility. The differential mathematical approach using Python provides significantly higher accuracy, efficiency and flexibility in calculations. Therefore, the development of an algorithm for solving systems of nonlinear equations by Newton's method will allow achieving maximum accuracy of results. The article develops an algorithm for solving a nonlinear system of equations by Newton's method, which allows determining the distribution of ions in the working zone of the filter in both liquid and solid phases. The implementation of the developed algorithm is carried out using the Python ecosystem. Based on the calculation results, a graph of the output curve is constructed and a complex software module is developed that allows controlling the most important operating parameters of ion-exchange filters. Also, the use of a complex software module will contribute to reducing the operating costs of ion-exchange filters and increasing the efficiency of processes.*

**Keywords**: Python, Newton's method, ion exchange, ion concentration, Ion exchange technology

# OPRACOWANIE ALGORYTMU OBLICZANIA PROCESÓW WYMIANY JONOWEJ Z UŻYCIEM EKOSYSTEMU PYTHON

***Streszczenie.*** *Wymiana jonowa odgrywa kluczową rolę nie tylko w usuwaniu twardości wody i redukowaniu stężenia niepożądanych jonów, ale także w ochronie środowiska. Ze względu na swoją uniwersalność jest niezastąpionym narzędziem w nowoczesnych technologiach ochrony środowiska. Metody obliczeniowe stanowią istotny element wdrażania technologii wymiany jonowej, gdyż odgrywają kluczową rolę w ich efektywnym działaniu. Istnieją metody obliczeniowe wykorzystywane przy projektowaniu układów wymiany jonowej, lecz podobne metody służące określaniu parametrów roboczych w trakcie pracy nie zostały jeszcze opracowane. Utrudnia to optymalizację procesu w warunkach rzeczywistych i wymaga dalszych prac badawczo-rozwojowych. Metoda obliczeniowa powinna być prosta w użyciu, a jednocześnie zapewniać wysoką dokładność wyników i uwzględniać takie istotne czynniki, jak skład wody, szybkość filtracji, temperatura i rodzaj żywic jonowymiennych. Szczególnie ważne jest, aby metodę tę opracowano w oparciu o otwarte biblioteki Pythona, co zapewni dostępność i wszechstronność. Podejście matematyczne oparte na różniczce z wykorzystaniem języka Python zapewnia znacznie większą dokładność, wydajność i elastyczność obliczeń. Zatem opracowanie algorytmu rozwiązywania układów równań nieliniowych metodą Newtona pozwoli na osiągnięcie maksymalnej dokładności wyników. W artykule opracowano algorytm rozwiązywania układu równań nieliniowych metodą Newtona, który pozwala na określenie rozkładu jonów w strefie roboczej filtra zarówno w fazie ciekłej, jak i stałej. Implementację opracowanego algorytmu przeprowadzono z wykorzystaniem ekosystemu Python. Na podstawie wyników obliczeń skonstruowano wykres krzywej wyjściowej oraz opracowano kompleksowy moduł programowy umożliwiający sterowanie najważniejszymi parametrami pracy filtrów jonowymiennych. Ponadto zastosowanie kompleksowego modułu programowego pozwoli na obniżenie kosztów eksploatacji filtrów jonowymiennych i zwiększenie efektywności procesów.*

**Słowa kluczowe**: Python, Metoda Newtona, wymiana jonowa, stężenie jonów, technologia wymiany jonowej

## Introduction

The development of methods for calculating ion exchange filters is extremely relevant, as it ensures an increase in the efficiency of the above processes [11, 28]. Accurate calculation of filter parameters allows optimizing the use of resources, reducing operating costs, and reducing the environmental impact [2, 29].

A significant number of engineering methods have been developed and implemented to calculate the technological performance of ion exchange filters [7, 13, 26]. Some of them require experiments, while others are less accessible to users due to the closed nature of the developers' databases [8, 19, 22]. The use of calculations based on mathematical models also has certain difficulties [4, 14, 22]. In this regard, there is a need to develop a method that would be accessible to every user and ensure high accuracy of the results [6, 10, 15].

Such a method should be easy to use while providing accurate calculations that take into account important factors such as water composition, filtration rate, temperature, and type of ion exchange resin [1, 17, 23]. It is especially important that the method be developed based on open source Python libraries [9]. A key element is to create a software with networking capabilities based on these libraries to quickly and efficiently calculate the filter parameters. This will provide convenience and accessibility for specialists of different levels, as well as demonstrate the extensive capabilities of Python in data processing and process modeling [1, 24].

Publications have presented an algorithm for calculating a cationic filter taking into account the distribution of cations in the working area [3, 27]. For this purpose, a graph-analytical calculation method was used [3], which has certain difficulties in use. Also, this method requires considerable time and effort to build graphs and perform calculations. Modern computational methods in Python allow automating the calculation process, which reduces the time required for calculations and provides higher performance when solving complex problems.

Therefore, the development of an algorithm for solving a system of nonlinear equations by the Newtonian method using Python libraries will allow achieving high accuracy of the results. The article propose a new algorithm for calculation of ion exchange using Python. The presented approach is very interesting and important for industrial application to speed up processing or develop efficient monitoring systems.

## 1. Materials of research

In this study, we developed an algorithm for solving a system of nonlinear equations using Newton's method in Python. The use of Python libraries for numerical methods (in particular, NumPy, SciPy) makes it easy to scale calculations to large data sets and perform calculations for multidimensional systems [18, 25]. The differential mathematical approach using Python provides significantly higher accuracy, efficiency, and flexibility in calculations compared to the graph-analytical method [20, 25]. Based on the developed algorithm and using powerful Python tools for working with data and their visualization, the necessary graphs of the distribution of cations between phases are created. In addition, a graph of the output filtration curve is plotted, which clearly demonstrates the dynamics of the ion exchange process and its efficiency. Python provides a high level of flexibility and accuracy in visualizing complex technological processes, facilitating deeper analysis and optimization of filter performance parameters.

## 2. Summary of the main material

For the purpose of analysis, let's consider the softening process in an apparatus with a fixed layer of cationite. It proceeds under non-equilibrium conditions [3, 5, 28]. According to the theory of dynamics, mass transfer occurs and is completed in the working zone, which gradually moves from top to bottom [2, 21, 29]. The mass transfer zone reflects the kinetics of the process in the sorbent, is related to the conditions of mass transfer, and depends on the speed, solution concentration, and characteristics of the ionite (grain diameter). The process of cationite depletion occurs in layers [3, 17]. As a result, a working zone is formed, which has a certain length and speed at which it moves down the layer. A certain concentration field, called the concentration front, is formed in the working part of the apparatus [2, 3, 17]. The working area, gradually moving down the filter, reaches its lower section. At this point, the cations that were retained by the filter begin to break through into the filtrate. If all changes in the concentration in the initial solution are recorded, the shape of the output curve repeats the shape of the concentration front, Fig. 1. When cations break through into the filtrate, the ion exchange filter is turned off for regeneration. At this point in the working area, the cationite is not completely depleted and has a certain average value.

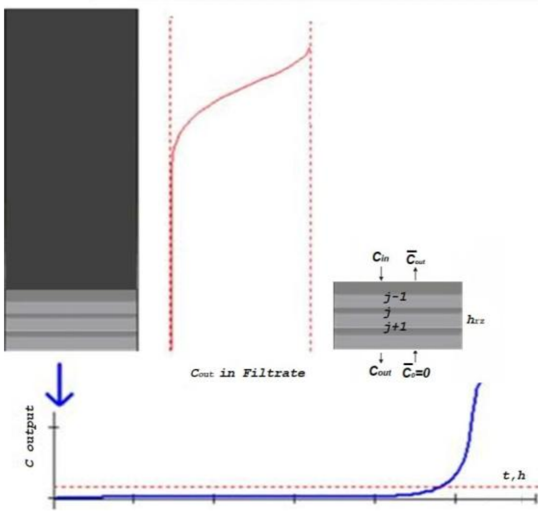


*Fig. 1. Movement of the formed sorption front through the ionite layer and the output curve*

Here, in Fig. 1, the dark color indicates the saturated or used part of the cationite, and the light color indicates the unused ionite that does not interact with water. The reason for this is the presence of a working zone in the filter, in which a concentration front is formed and moves through the filter in a parallel transfer mode.

To take into account the non-equilibrium conditions that arise in an ion exchange filter, namely the formation of a concentration front, an algorithm for calculating the concentrations in the filter working zone *h(rz)* was developed.

Using equations (1)-(2), determine the cation concentrations in water and cationite of each layer.

To solve the system of nonlinear equations (1), (2), which is a mathematical model of the distribution of cations in the filter working zone and takes into account non-equilibrium conditions, the Newton method was proposed in [7].

$$\bar{c}_{jdyn} = a \cdot f(c_j), j = 1, 2, \ldots, m \tag{1}$$

$$u(\bar{c}_{jdyn} - \bar{c}_{j+1dyn}) - v(c_{j-1} - c_j), j = 1, 2, \ldots, m \tag{2}$$

where *u* and *v* are the velocities of the sorption front and water m/s; $\bar{c}_{jdyn}$, $c_j$ – cation concentrations in cationite and water on layer *j* under dynamic conditions, mg-eq/dm$^3$, mg-eq/dm$^3$; $a \leq 1$ is a coefficient that takes into account non-equilibrium conditions; *m* the number of layers.

The boundary conditions for equations (1) and (2) are those for the parallel transfer stage only:

$$z \to \infty, C \to 0, \bar{C} \to 0$$

$$z \to -\infty, C \to C_0, \bar{C} \to f(C_0)$$

After transforming the system of equations (1) and (2), the final formula was obtained:

$$\frac{u}{v}[f(c_j) - f(c_{j+1})] - c_{j-1} + c_j = 0, j = 1,2, \ldots, m \tag{3}$$

It should be noted $\bar{c}_{jdyn}$ that was substantiated in [6, 7].

$$\bar{c}_{jdyn} = \Psi(c_j)$$

$$\bar{c}_{dyn} = a\frac{k\bar{c}_{\infty}c_j}{1+kc_j} \tag{4}$$

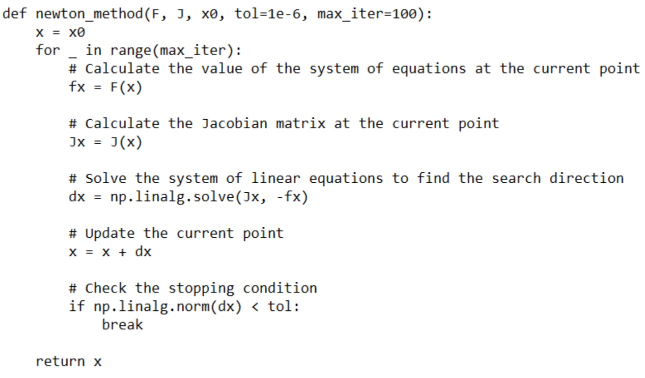
where *k* is the equilibrium constant; $\bar{c}_{\infty}$ is the maximum capacity of the ionite, mg-eq/g; $\bar{c}_{dyn}$ is the concentration of the absorbed substance in the ionite under pseudo-equilibrium conditions, mg-eq/g; $c_j$ is the concentration of a substance in the liquid phase, mg-eq/dm$^3$; *a* is a coefficient that takes into account non-equilibrium conditions [3].

Find the solution to the system of nonlinear equations for each layer (1) and (2).

For the first time, the Newton method–based algorithm for solving a system of nonlinear equations has been adapted to the specific operating conditions of ion-exchange filters, taking into account their physicochemical regularities, mass-transfer kinetics, and structural characteristics of the technological process. The developed algorithm is tailored for modeling the sodium-cation exchange process and provides enhanced accuracy and numerical stability compared to universal implementations. The structure of the algorithm consists of the following elements (steps). First, it is necessary to define a function that performs the calculation by Newton's method and specify the following parameters:

- *F* (function): a system of nonlinear equations;
- *J* (function): the Jacobi matrix of the system of equations;
- *x*0 (numpy.array): the initial estimate of the solution;
- *tol* (float, optional): the allowable error. Default values: 1e-6;
- *max_iter* (*int, optional*): the maximum number of iterations. Default value: 100.

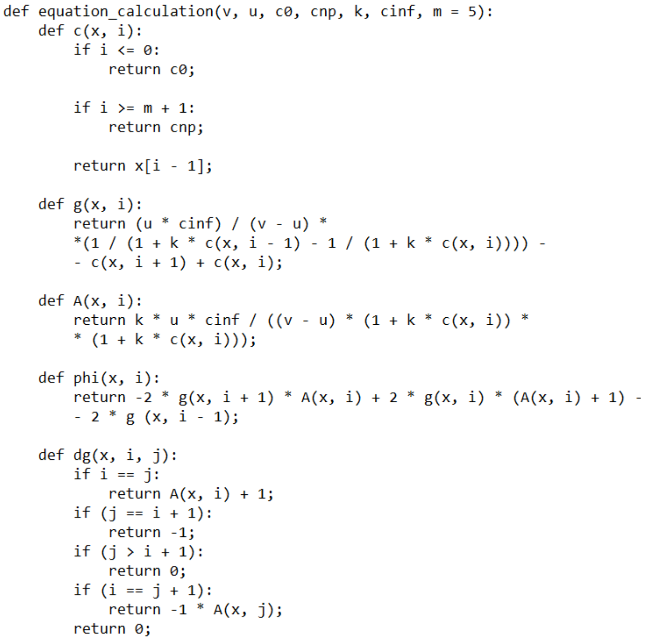
An example of a program snippet with the newton_method function is shown in Fig. 2.



```
def newton_method(F, J, x0, tol=1e-6, max_iter=100):
    x = x0
    for _ in range(max_iter):
        # Calculate the value of the system of equations at the current point
        fx = F(x)

        # Calculate the Jacobian matrix at the current point
        Jx = J(x)

        # Solve the system of linear equations to find the search direction
        dx = np.linalg.solve(Jx, -fx)

        # Update the current point
        x = x + dx

        # Check the stopping condition
        if np.linalg.norm(dx) < tol:
            break

    return x
```

*Fig. 2. A program snippet with the newton_method function*

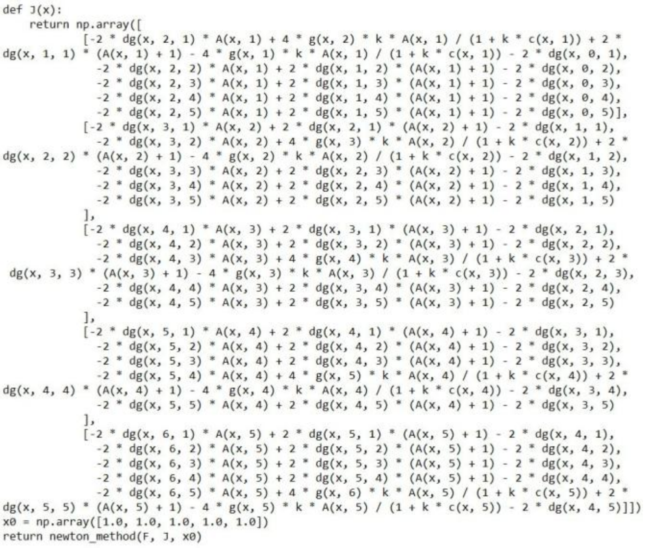
The next step is to create an equation_calculation function that directly performs the calculation based on Newton's method [12, 16], a program snippet with the created function is shown in Fig. 3.



```
def equation_calculation(v, u, c0, cnp, k, cinf, m = 5):
    def c(x, i):
        if i <= 0:
            return c0;

        if i >= m + 1:
            return cnp;

        return x[i - 1];

    def g(x, i):
        return (u * cinf) / (v - u) *
        *(1 / (1 + k * c(x, i - 1) - 1 / (1 + k * c(x, i)))) -
        - c(x, i + 1) + c(x, i);

    def A(x, i):
        return k * u * cinf / ((v - u) * (1 + k * c(x, i)) *
        * (1 + k * c(x, i)));

    def phi(x, i):
        return -2 * g(x, i + 1) * A(x, i) + 2 * g(x, i) * (A(x, i) + 1) -
        - 2 * g (x, i - 1);

    def dg(x, i, j):
        if i == j:
            return A(x, i) + 1;
        if (j == i + 1):
            return -1;
        if (j > i + 1):
            return 0;
        if (i == j + 1):
            return -1 * A(x, j);
        return 0;
```

*Fig. 3. A fragment of the program with the created function equation_calculation that directly performs the calculation based on 's method*

The algorithm also needs to create a Jacobi matrix to solve the system of equations. An example of a program snippet with an algorithm for solving a matrix is shown in Fig. 4.



```
def J(x):
    return np.array([
            [-2 * dg(x, 2, 1) * A(x, 1) + 4 * g(x, 2) * k * A(x, 1) / (1 + k * c(x, 1)) + 2 *
dg(x, 1, 1) * (A(x, 1) + 1) - 4 * g(x, 1) * k * A(x, 1) / (1 + k * c(x, 1)) - 2 * dg(x, 0, 1),
              -2 * dg(x, 2, 2) * A(x, 1) + 2 * dg(x, 1, 2) * (A(x, 1) + 1) - 2 * dg(x, 0, 2),
              -2 * dg(x, 2, 3) * A(x, 1) + 2 * dg(x, 1, 3) * (A(x, 1) + 1) - 2 * dg(x, 0, 3),
              -2 * dg(x, 2, 4) * A(x, 1) + 2 * dg(x, 1, 4) * (A(x, 1) + 1) - 2 * dg(x, 0, 4),
              -2 * dg(x, 2, 5) * A(x, 1) + 2 * dg(x, 1, 5) * (A(x, 1) + 1) - 2 * dg(x, 0, 5)],
            [-2 * dg(x, 3, 1) * A(x, 2) + 2 * dg(x, 2, 1) * (A(x, 2) + 1) - 2 * dg(x, 1, 1),
              -2 * dg(x, 3, 2) * A(x, 2) + 4 * g(x, 3) * k * A(x, 2) / (1 + k * c(x, 2)) + 2 *
dg(x, 2, 2) * (A(x, 2) + 1) - 4 * g(x, 2) * k * A(x, 2) / (1 + k * c(x, 2)) - 2 * dg(x, 1, 2),
              -2 * dg(x, 3, 3) * A(x, 2) + 2 * dg(x, 2, 3) * (A(x, 2) + 1) - 2 * dg(x, 1, 3),
              -2 * dg(x, 3, 4) * A(x, 2) + 2 * dg(x, 2, 4) * (A(x, 2) + 1) - 2 * dg(x, 1, 4),
              -2 * dg(x, 3, 5) * A(x, 2) + 2 * dg(x, 2, 5) * (A(x, 2) + 1) - 2 * dg(x, 1, 5)
            ],
            [-2 * dg(x, 4, 1) * A(x, 3) + 2 * dg(x, 3, 1) * (A(x, 3) + 1) - 2 * dg(x, 2, 1),
              -2 * dg(x, 4, 2) * A(x, 3) + 2 * dg(x, 3, 2) * (A(x, 3) + 1) - 2 * dg(x, 2, 2),
              -2 * dg(x, 4, 3) * A(x, 3) + 4 * g(x, 4) * k * A(x, 3) / (1 + k * c(x, 3)) + 2 *
 dg(x, 3, 3) * (A(x, 3) + 1) - 4 * g(x, 3) * k * A(x, 3) / (1 + k * c(x, 3)) - 2 * dg(x, 2, 3),
              -2 * dg(x, 4, 4) * A(x, 3) + 2 * dg(x, 3, 4) * (A(x, 3) + 1) - 2 * dg(x, 2, 4),
              -2 * dg(x, 4, 5) * A(x, 3) + 2 * dg(x, 3, 5) * (A(x, 3) + 1) - 2 * dg(x, 2, 5)
            ],
            [-2 * dg(x, 5, 1) * A(x, 4) + 2 * dg(x, 4, 1) * (A(x, 4) + 1) - 2 * dg(x, 3, 1),
              -2 * dg(x, 5, 2) * A(x, 4) + 2 * dg(x, 4, 2) * (A(x, 4) + 1) - 2 * dg(x, 3, 2),
              -2 * dg(x, 5, 3) * A(x, 4) + 2 * dg(x, 4, 3) * (A(x, 4) + 1) - 2 * dg(x, 3, 3),
              -2 * dg(x, 5, 4) * A(x, 4) + 4 * g(x, 5) * k * A(x, 4) / (1 + k * c(x, 4)) + 2 *
dg(x, 4, 4) * (A(x, 4) + 1) - 4 * g(x, 4) * k * A(x, 4) / (1 + k * c(x, 4)) - 2 * dg(x, 3, 4),
              -2 * dg(x, 5, 5) * A(x, 4) + 2 * dg(x, 4, 5) * (A(x, 4) + 1) - 2 * dg(x, 3, 5)
            ],
            [-2 * dg(x, 6, 1) * A(x, 5) + 2 * dg(x, 5, 1) * (A(x, 5) + 1) - 2 * dg(x, 4, 1),
              -2 * dg(x, 6, 2) * A(x, 5) + 2 * dg(x, 5, 2) * (A(x, 5) + 1) - 2 * dg(x, 4, 2),
              -2 * dg(x, 6, 3) * A(x, 5) + 2 * dg(x, 5, 3) * (A(x, 5) + 1) - 2 * dg(x, 4, 3),
              -2 * dg(x, 6, 4) * A(x, 5) + 2 * dg(x, 5, 4) * (A(x, 5) + 1) - 2 * dg(x, 4, 4),
              -2 * dg(x, 6, 5) * A(x, 5) + 4 * g(x, 6) * k * A(x, 5) / (1 + k * c(x, 5)) + 2 *
dg(x, 5, 5) * (A(x, 5) + 1) - 4 * g(x, 5) * k * A(x, 5) / (1 + k * c(x, 5)) - 2 * dg(x, 4, 5)]])
x0 = np.array([1.0, 1.0, 1.0, 1.0, 1.0])
return newton_method(F, J, x0)
```

*Fig. 4. A program fragment with the algorithm for solving the Jacobi matrix*

The function $J(x)$ builds a Jacobi matrix taking into account the nonlinear influence of concentrations and interlayer interactions.

The structure of the matrix is five-point (almost tridiagonal), which is typical for problems involving the approximation of derivatives in space. Sorption effects are taken into account through additional fractional terms. The algorithm is unique and complex, as it includes both differentiation and physico-chemical models of the process. This implementation guarantees the accuracy, stability, and convergence of the numerical solution.

Then, based on the obtained concentration values in each layer, a concentration curve is constructed – the relationship between the concentrations in water and sorbent.

To do this, the concentration in water and, accordingly, in the ion exchange resin on each layer mi was determined according to the above algorithm. Fig. 5 shows the calculated values of the concentrations in water and ionite of each layer, i.e., gives an understanding of the ionite depletion in the working area.

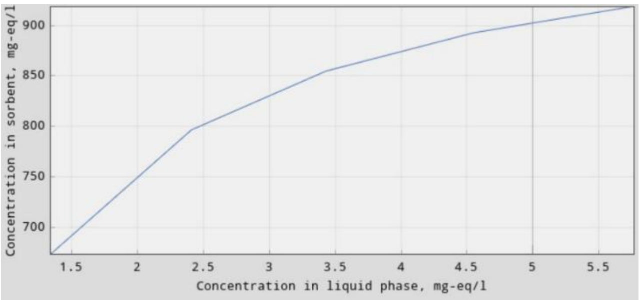


*Fig. 5. Estimated concentrations in water and ionite in the filter working area*

The development of the above algorithm made it possible to create a comprehensive calculation of the operating parameters of the ion exchange filter and improve the previously developed method [3] for determining rational operating parameters. In particular, the need to determine the unused (tail) capacity in the working area was eliminated ($V_{uz}$ unused zone). Instead, it is necessary to determine the total amount of cations in g-eq that participated in the exchange, taking into account the established distribution of cations in the filter working area, and calculate the dynamic exchange capacity. It is worth noting that the previous algorithm was based on a graph-analytical method of determining the distribution of ions in the working area and depended on the construction of graphs. The solution of systems of nonlinear equations by the Newton method using Python libraries allowed us to achieve high accuracy of the results.

Consider the following developed program module for calculating an ion exchange filter. In it, according to the developed algorithm above, the dynamic exchange capacity of the entire ionite loaded into the filter is determined, taking into account saturated and not fully saturated layers, as shown in Fig. 1. The intermediate calculations to be made are as follows:

- determines the height of the working zone *h(rz)* for a given filtration rate and initial concentration of the treated water [17];
- determines the speed of movement of the working area *u*, m/h and the time of mass transfer [3];
- $W_d$ is determined by the dynamic exchange capacity (the number of gram equivalents involved in the exchange), according to the equation (5):

$$W_d = \left(V_f - V_{rz}\right)\cdot \overline{C}_0 + \sum_{j=1}^{m} \overline{C}_{jdyn}\cdot \Delta V_{jrz} \tag{5}$$

$V_f$, $V_{rz}$ is the volume of the filter and the volume of the working zone; $\Delta V_{jrz}$ is the volume of the working zone of layer *j*; $\overline{C}_0$ is the cationite capacity taking into account the regeneration conditions, g-eq/m$^3$;

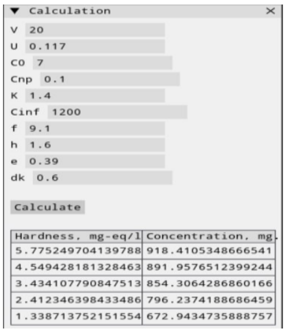
- determine the operating time of the filter, is determined by the filter operation time until it is taken out of operation for regeneration, according to the relation (6):

$$\tau_{brk(\text{im})} = \frac{W_{din}}{(C_{en}-C_{ex})\cdot f\cdot v} \tag{6}$$

- based on the data obtained, a graph of the dependence of the concentration of a substance in the leachate on the time of appearance of the concentration in the leachate, i.e. the output curve, is drawn.

All the program code is available on the developer's own GitHUB [3].

All the calculation results and the output curve are displayed on the screen in the form of a graph showing the result of the developed program (Fig. 6, 7).



V 20
U 0.117
C0 7
Cnp 0.1
K 1.4
Cinf 1200
f 9.1
h 1.6
e 0.39
dk 0.6

| Hardness, mg-eq/l | Concentration, mg |
|---|---|
| 5.775249704139788 | 918.4105348666541 |
| 4.549428181328463 | 891.9576512399244 |
| 3.434107790847513 | 854.3064286860166 |
| 2.412346398433486 | 796.2374188686459 |
| 1.338713752151554 | 672.9434735888757 |

*Fig. 6. Calculated values of the ion exchange filter parameters obtained on the basis of the created program*

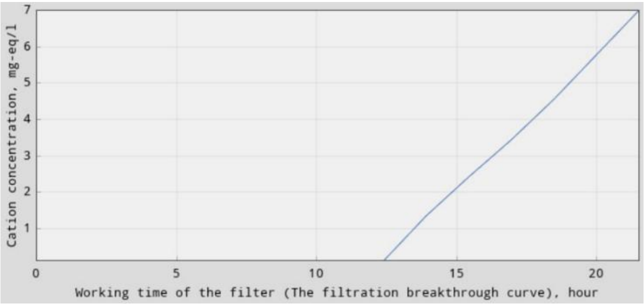


*Fig. 7. Breakthrough time and operational parameters of the ion exchange filter*

To confirm the adequacy of the developed method of determining the operating parameters of the cationic filter, a comparison with the previously proposed method of calculation, which is based on graphoanalytical constructions and estimation of the unused capacity of the sorbent loaded in the filter [3].

Calculations were carried out for the filter with given characteristics: diameter 3.4 m, height of cationite KU-2-8 loading – 1.6 m, cationite grain diameter – 0.8 mm. The filter capacity was 182 m³/h at filtration rate of 15 m/h. Initial concentration of hardness cations in water ($C_{en}$) was 7 g-eq/m³, concentration in filtrate ($C_{ex}$) was 0.1 g-eq/m³. Exchange capacity of cationite during operation, calculated by known total exchange capacity (1500 g-eq/m³), was 1048 g-eq/m³.

Calculations were performed for different filtration rates and the results are presented in Table 1. In Table 1, the first column contains the calculated data τbrk1 obtained using the previously developed algorithm [17]:

$$\tau_{brk(pr)} = \frac{E_f}{(C_{en}-C_{ex})\cdot v\cdot f}\left(V_f - V_{uz}\right) \qquad (7)$$

The second column contains the calculated data $\tau_{brk(im)}$ obtained using the improved algorithm according to formula (6). The table also shows the experimental values of the time of cation breakthrough into the filtrate $\tau_{exp}$. Data analysis shows satisfactory coincidence of calculated and experimental values. The average deviation of the calculated values of the filter operation time ($\tau$) from the experimental values does not exceed 5%.

As can be seen from Table 1, the discrepancies between the calculated values obtained by the two methods are insignificant. This confirms the possibility of using the developed method for determining the operating parameters of Na-cationic filters in order to reduce operating costs and improve the efficiency of their operation.

The developed method of determining the operating parameters of cationic filter showed a high degree of consistency with the experimental data. Its application allows to optimize the operation of filters, reducing operating costs and increasing the efficiency of water treatment process.

The algorithm for calculation of ion concentration inside the working zone of the apparatus has been developed and realized. On the basis of this algorithm the method of calculation of rational working parameters of cationic filter is improved, allowing to determine the working capacity and time of effective operation of the filter when changing the filtration rate. Regulation of technological parameters in the process of operation provides saving of water and reagents for auxiliary needs of water treatment on average by 12–17%.

A software package for automated execution of all necessary calculations to determine the optimal operating parameters of cation-exchange units has been developed. It is proposed to integrate all implemented algorithms into an automated operator-technologist workstation. The implementation of this automated workstation will enable a reduction in operating costs, including up to 17% savings in water and reagent consumption, as well as an increase in the utilization efficiency of the cation-exchange resin within the filter.

*Table 1. Comparison of ion breakthrough time in filtrate at different filtration rates*

| Filtration rates, m/h | $\tau_{brk(pr)}$, h | $\tau_{brk(im)}$, h | $\tau_{exp}$, h |
|---|---|---|---|
| 20 | 11.23 | 12.3 | 12.7 |
| 15 | 15.74 | 17.1 | 16.8 |
| 10 | 25.17 | 26.8 | 27.0 |

## 3. Conclusions

In the course of the study, an algorithm for the numerical solution of a nonlinear system of equations was developed and implemented to model the distribution of ion concentrations in the working zone of an ion-exchange filter. The algorithm is based on a modified Newton's method with the construction of a Jacobian matrix whose elements take into account the specific features of mass transfer processes and nonlinear dependencies between ion concentrations and the sorption characteristics of the medium. The conducted analysis confirmed the existence of a unique solution within the operating range of parameters, ensuring the stability and convergence of the computations.

The proposed approach accurately accounts for the complex interaction of phase flows and the nonlinear behavior of concentrations in the system, enabling effective modelling and control of purification and separation processes.

Furthermore, the algorithm for the technological calculation of the ion-exchange filter was improved, allowing for the determination of optimal operational parameters during real-time operation. The implementation of the algorithm within the Python ecosystem ensures high flexibility, scalability, and integration with other engineering analysis tools.

The improved method for determining rational operating parameters allows for a more accurate determination of the operating parameters of the cationization process, taking into account the concentration-dynamic conditions. It makes it possible to calculate the parameters of an ion exchange filter.

The method can be recommended for predicting and analyzing breakthrough curves, estimating the dynamic capacity of the loaded sorbent, and optimizing water and reagent consumption. The application of the integrated software module will contribute to increased efficiency in the operation of ion-exchange filters, simplification of technology deployment across various industries, and reduction of operating costs in water treatment systems.

## References

[1] Abeywickrama J., Hoth N.: Optimization of an Ion Exchange Process for the Selective Recovery of Cobalt and Copper from Heap Leaching Waste Materials. Insights in Mining Science & Technology 3(1), 555605, 2022 [https://doi.org/10.19080/IMST.2022.03.555605].

[2] Abramova A. O.: Development of a software package for designing ion exchange systems for wastewater treatment in MATLAB. Information processing systems 34(73), 2023 [https://doi.org/10.32782/2663-5941/2023.3.1/09]

[3] Ajjrapetjan T. S., Chub I.: Water treatment at heat and power enterprises using Na-cationic filters. Monograph Kharkiv National University of Municipal Economy named after A. Beketov, 2024.

[4] Al_Issa H. A. et al.: Assessment of the Effect of Corona Discharge on Synchronous Generator Self-Excitation. Energies 15(6), 2022, 2024 [https://doi.org/10.3390/en15062024].

[5] Bazaluk O. et al.: Energy Saving in Electromechanical Grain Cleaning Systems. Applied Sciences 12(3), 2022, 1418 [https://doi.org/10.3390/app12031418].

[6] Ben Moshe S. et al.: Geophysically based analysis of breakthrough curves and ion exchange processes in soil. Hydrology and Earth System Sciences 25(6), 2021, 3041–3052 [https://doi.org/10.5194/hess-25-3041-2021].

[7] Cechinel M. A. P. et al.: Application of ecofriendly cation exchangers (Gracilaria caudata and Gracilaria cervicornis) for metal ions separation and recovery from a synthetic petrochemical wastewater: Batch and fixed bed studies. Journal of Cleaner Production 172, 2018, 1928–1945 [https://doi.org/10.1016/j.jclepro.2017.11.235].

[8] Chen S. et al.: Continuous silicic acid removal in a fixed-bed column using a modified resin: Experiment investigation and artificial neural network modeling. Journal of Water Process Engineering 49, 2022, 102937 [https://doi.org/10.1016/j.jwpe.2022.102937].

[9] Dogaru I., Dogaru R.: Using Python and Julia for Efficient Implementation of Natural Computing and Complexity Related Algorithms. 20th International Conference on Control Systems and Computer Science, Bucharest, Romania, 2015, 599–604 [https://doi.org/10.1109/CSCS.2015.37].

[10] Du Z. L., Zheng T., Wang P.: Experimental and modelling studies on fixed bed adsorption for Cu(II) removal from aqueous solution by carboxyl modified jute fiber. Powder Technology 338, 2018, 952–959 [https://doi.org/10.1016/j.powtec.2018.06.015].

[11] Gomelya M. et al.: Research into ion exchange softening of highly mineralized waters. Eastern-European Journal of Enterprise Technologies 4(10(82)), 2016, 4–9 [https://doi.org/10.15587/1729-4061.2016.75338].

[12] Hart L. L., Kamantsev A. S.: Asymptotic analysis of the complexity of some function approximation algorithms. Questions of Applied Mathematics and Mathematical Modelling: a collection of scientific papers 20, 2020, 31–44 [https://doi.org/10.15421/322003].

[13] Igberase E., Osifo P., Ofomaja A.: Mathematical modelling of $Pb^{2+}$, $Cu^{2+}$, $Ni^{2+}$, $Zn^{2+}$, $Cr^{6+}$ and $Cd^{2+}$ ions adsorption from a synthetic acid mine drainage onto chitosan derivative in a packed bed column. Environmental Technology 38(24), 2017, 3203–3220 [https://doi.org/10.1080/09593330.2017.1375027].

[14] Karaiev O. et al.: Mathematical modelling of the fruit-stone culture seeds calibration process using flat sieves. Acta Technologica Agriculturae 24(3), 2021, 119–123 [https://doi.org/10.2478/ata-2021-0020].

[15] Khasawneh A. et al.: Optimal Determination Method of the Transposition Steps of An Extra-High Voltage Power Transmission Line. Energies 14, 2021, 6791 [https://doi.org/10.3390/en14206791].

[16] Kiseleva E. M., Hart L. L., Dovgay P. A.: On a problem of numerical simulating the derivative of discrete time series with approximate values. Journal of Automation and Information Sciences 47(12), 2015, 1–17 [https://doi.org/10.1615/JautomatInfScien.v47.i12.10].

[17] Korak J. A., Huggins R., Arias-Paic M.: Regeneration of pilot-scale ion exchange columns for hexavalent chromium removal. Water Research 118, 2017, 141–151 [https://doi.org/10.1016/j.watres.2017.03.018].

[18] Lezhenkin O. M. et al.: Investigation of the separation of combed heap of winter wheat. Journal of Physics: Conference Series 1781, 2020, 12016 [https://doi.org/10.1088/1742-6596/1781/1/012016].

[19] Ma A. et al.: Ion exchange homogeneous surface diffusion modelling by binary site resin for the removal of nickel ions from wastewater in fixed beds. Chemical Engineering Journal 358, 2019, 1–10 [https://doi.org/10.1016/j.cej.2018.09.135].

[20] Markowska K. et al.: Analysis and improvement of power quality in the onboard electrical power systems within a self-propelled floating crane. International Journal of Electrical Power & Energy Systems 161, 2024, 110179 [https://doi.org/10.1016/j.ijepes.2024.110179].

[21] Miroshnyk O. et al.: Investigation of Smart Grid Operation Modes with Electrical Energy Storage System. Energies 16, 2023, 2638 [https://doi.org/10.3390/en16062638].

[22] Mudhoo A., Pittman Jr C. U.: Adsorption data modeling and analysis under scrutiny: A clarion call to redress recently found troubling flaws. Chemical Engineering Research and Design 192, 2023, 371–388 [https://doi.org/10.1016/j.cherd.2023.02.033].

[23] Pedregal Montes A. I. et al.: Modeling of Ion Exchange Processes to Optimize Metal Removal from Complex Mine Water Matrices. Water 13(21), 2021, 3109 [https://doi.org/10.3390/w13213109].

[24] Qawaqzeh M. Z. et al.: Research of Emergency Modes of Wind Power Plants Using Computer Simulation. Energies 14, 2021, 4780 [https://doi.org/10.3390/en14164780].

[25] Sahgal A.: Implementation of Numerical Methods for Solving Differential Equations using Python. International Journal for Research Publication and Seminar 14(4), 2023, 133–140 [https://doi.org/10.36676/jrps.2023-v14i4-019].

[26] Tarpeh W. A. et al.: Effects of operating and design parameters on ion exchange columns for nutrient recovery from urine. Environmental Science: Water Research & Technology 4(6), 2018, 828–838 [https://doi.org/10.1039/C7EW00478H].

[27] Trunova I. et al.: The Substantiation of Reconstruction of Power Distribution Networks Objects. IEEE 2nd KhPI Week on Advanced Technology (KhPIWeek), Kharkiv, Ukraine, 2021, 126–131 [https://doi.org/10.1109/KhPIWeek53812.2021.9570107].

[28] Wang Y. et al.: Adsorption equilibrium, kinetics, and dynamic separation of $Ca^{2+}$ and $Mg^{2+}$ ions from phosphoric acid–nitric acid aqueous solution by strong acid cation resin. Chinese Journal of Chemical Engineering 27(4), 2019, 1003–1010 [https://doi.org/10.1016/j.cjche.2018.12.025].

[29] Yu Z. et al.: Application of mathematical models for ion-exchange removal of calcium ions from potassium chromate solutions by Amberlite IRC 748 resin in a continuous fixed bed column. Hydrometallurgy 158, 2015, 165–171 [https://doi.org/10.1016/j.hydromet.2015.10.015].

**Ph.D. Eng. Iryna Chub**
e-mail: chub.irina.nik@gmail.com

Graduated from Kharkiv National University of Urban Economy, Kharkiv, Ukraine, in 1999, and was qualified as an Engineer Technologist. She received her Ph.D. in water supply and sewage from the National University of Water Management and Nature Resources, Rivne, Ukraine in 2010.

Her research interests include: Automation of water treatment processes and software development for water treatment and wastewater purification systems.

https://orcid.org/0000-0002-5546-3286

**Ph.D. Oleksii Proskurnia**
e-mail: o.i.proskurnia@gmail.com

Graduated from V. N. Karazin Kharkiv State University in 1998 and received the qualification of mathematician, teacher of mathematics and computer science. He received his Ph.D. in General Pedagogy and History of Pedagogy from the H. S. Skovoroda Kharkiv National Pedagogical University. Since 2016, associate professor of the Department of Mathematics.

His research interests include: teaching mathematics, mathematical statistics

https://orcid.org/0000-0002-3111-3417

**Ph.D. Eng. Kateryna Demchenko**
e-mail: yayaska31@gmail.com

Graduated from Kharkiv State Technical University of Agriculture, Kharkiv, Ukraine, in 2007, and was qualified as an automation and computer-integrated technologies Engineer. Her received his Ph.D. in computer systems and components from the National Technical University "Kharkiv Politechnic Institute", Ukraine, in 2014.

Her research interests include: fault tolerance, reliability and speed of computer systems.

https://orcid.org/0000-0002-3168-5351

**D.Sc. Eng. Oleksandr Miroshnyk**
e-mail: omiroshnyk@btu.kharkiv.ua

Graduated from Kharkiv State Technical University of Agriculture, Kharkiv, Ukraine, in 2004, and was qualified as an Electrical Engineer. He received his Ph.D. in electric engineering (power stations, systems and networks) from the National Technical University "Kharkiv Politechnic Institute", Ukraine, in 2009 and Doctor of Technical Sciences degree in 2016.

His research interests include: Smart Grid, storage system and renewable energy.

https://orcid.org/0000-0002-6144-7573

**Ph.D. Eng. Taras Shchur**
e-mail: shchurtg@gmail.com

Graduated from Kharkiv State Technical University of Agriculture in 2004, and was qualified as mechanical engineer. He received his Ph.D. in mechanical engineering (machines and means of mechanization of agricultural production) from the Kharkiv State Technical University of Agriculture, in 2009. His research interests include: Smart Grid and renewable energy.

https://orcid.org/0000-0003-0205-032X

**Ph.D. Eng. Serhii Halko**
e-mail: galkosv@gmail.com

Graduated from the Faculty of Electrification of Agriculture of the Tavria State Agrotechnical Academy in 1994 with a degree in "Electrification and Automation of Agriculture", was awarded the qualification "Electrical Engineer". In 2002, he defended his thesis for the degree of Candidate of Technical Sciences at the Specialized Academic Council of Donetsk National Technical University.

His research interests include: Smart Grid, modeling of power grid modes, and renewable energy.

https://orcid.org/0000-0001-7991-0311