

## DEVELOPMENT OF AN APPLICATION FOR THE THERMAL PROCESSING OF OIL SLIME IN THE INDUSTRIAL OIL AND GAS SECTOR

Gulnar Balakayeva<sup>1</sup>, Gaukhar Kalmenova<sup>1</sup>, Dauren Darkenbayev<sup>1</sup>, Christofer Phillips<sup>2</sup>

<sup>1</sup>Al-Farabi Kazakh National University, Almaty, Kazakhstan, <sup>2</sup>Newcastle University, Newcastle, United Kingdom

**Abstract.** The production activities of oil refineries and oil and gas-producing enterprises inevitably have an anthropogenic impact on the environment, so environmental issues and the rational use of natural resources are important. The most dangerous pollutants of all components in the natural environment are oil waste, and one of the most effective methods of processing is heat treatment. The task was set to neutralize oil waste by thermal processing of oil slime to an environmentally safe level. The studies are carried out by methods of mathematical and numerical simulation of thermal processing, the results of which describe changes in temperature and mass of the stream over time. Extensive calculations with varying current operating and other parameters allow us to optimize the flow of heat and mass transfer during the thermal processing of oil slime. Numerical modeling is implemented using the method of alternating directions in an implicit iterative scheme until a convergence condition is met. The purpose of this work is to create an application for solving research and practical problems of oil waste processing. The application used allows the solution of the problems of oil slime processing. With the help of color-animated illustrations and a graphical interface, it supports the visualization of the results obtained, and provides the possibility of interactive interaction with the user, while providing instant control of the results obtained for timely decision-making to prevent environmental pollution in the industrial oil and gas sector.

**Keywords:** oil slime, software, heat treatment, applied application, ADI, mathematical model

### OPRACOWANIE APLIKACJI DO TERMICZNEGO PRZETWARZANIA SZLAMÓW OLEJOWYCH W PRZEMYSŁOWYM SEKTORZE NAFTOWYM I GAZU

**Streszczenie.** Działalność produkcyjna rafinerii ropy naftowej oraz przedsiębiorstw wydobywających ropę i gaz nieuchronnie ma antropogeniczny wpływ na środowisko, dlatego tak ważne są kwestie środowiskowe i racjonalne wykorzystanie zasobów naturalnych. Najbardziej niebezpiecznymi zanieczyszczeniami wszystkich składników środowiska naturalnego są odpady olejowe, a jedną z najbardziej efektywnych metod ich przetwarzania jest obróbka cieplna. Zadanie polegało na neutralizacji odpadów olejowych poprzez termiczną obróbkę szlamu olejowego do poziomu bezpiecznego dla środowiska. Badania prowadzone są metodami matematycznej i numerycznej symulacji obróbki cieplnej, której wyniki opisują zmiany temperatury i masy strugi w czasie. Rozbudowane obliczenia przy zmiennym prądzie roboczym i innych parametrach pozwalają nam optymalizować przepływ ciepła i wymiany masy podczas termicznej obróbki szlamów olejowych. Modelowanie numeryczne realizowane jest metodą naprzemiennych kierunków w niewyjawnym schemacie iteracyjnym, aż do spełnienia warunku zbieżności. Celem pracy jest stworzenie aplikacji do rozwiązywania problemów badawczych i praktycznych przetwarzania odpadów olejowych. Zastosowana aplikacja pozwala na rozwiązanie problemów związanych z przetwarzaniem szlamów olejowych. Za pomocą kolorowych animowanych ilustracji i interfejsu graficznego wspiera wizualizację uzyskanych wyników oraz zapewnia możliwość interaktywnej interakcji z użytkownikiem, zapewniając jednocześnie natychmiastową kontrolę uzyskanych wyników w celu szybkiego podejmowania decyzji zapobiegających zanieczyszczeniu środowiska w sektorze gazu przemysłowego.

**Słowa kluczowe:** szlam olejowy, oprogramowanie, obróbka cieplna, zastosowana aplikacja, ADI, model matematyczny

### Introduction

The ecological state of our planet has been causing concern for a long time. The anthropogenic impact on the environment causes irreparable harm to it and is one of the serious sources of environmental pollution in the oil, and oil refining, industry. The modern world economy requires a colossal amount of energy resources, the main source of which is oil, and ecology is often relegated to the background. Modern volumes of produced hydrocarbons and the capacity of their processing enterprises bring environmental protection problems to the fore.

The oil industry occupies one of the first places among the leading industries in terms of the level of environmental impact due to the formation of a large amount of heterogeneous waste. One of these is oil slime – a colloidal system of high-molecular compounds of oil, mineral particles of various compositions, and reservoir water. This is the largest waste of the oil production and refining industries, characterized by the complexity of the chemical composition, and is in the process of constant transformation. Sludge is formed during the construction of oil and gas wells, during the field operation of oil refining fields, wastewater disposal, as well as during the cleaning of tanks and other equipment. Various types of oil slime are also common waste for the objects of the energy complex, transport, machine-building, chemical, and metallurgical enterprises. The nature of their formation in these industries is predominantly similar to the waste-forming processes in the oil industry. The most significant amounts of oil-containing waste are formed during the cleaning of tank farms of thermal power plants, airports, railway stations, and metallurgical plants. Steaming stations of railway tank cars and treatment facilities are also sources of large-capacity hydrocarbon-containing slurries of various phase and chemical compositions. Special attention should be paid to such a type

of oil-containing waste as smeared soils are formed during emergency oil spills. The main difference between them and oil slime is a lower concentration of hydrocarbons. The reduced viscosity of the spilled oil leads to the fact that it forms a monomolecular film layer on the surface of the relief. If its thickness does not exceed 10 mm, then the penetration of oxygen is delayed by about 5–10%, which does not significantly affect the vital activity of microorganisms. In the case when the ability of oxygen absorption by the oil layer is 80–90%, the process of photosynthesis is hindered, which leads to a decrease in the concentration of oxygen in the soil and contributes to the suppression of the vital activity of organisms until their ultimate death.

Processing and disposal of oil slime is an important environmental and economic task. One of the most effective methods of processing oil slime is heat treatment. Thermal methods of oil slime processing are based on the processes of thermal decomposition of petroleum products [1]. As a result of the complete thermal decomposition of oil products, the final degradation products – CO<sub>2</sub> and H<sub>2</sub>O – are formed. It also significantly reduces the volume of inorganic substances, such as metals and salts, and allows the effective elimination of waste.

### 1. Related works

In this section, we discuss related research papers that are most relevant to our topic. To discussion on an analysis of the scientific and technical literature on the thermal processing of oil slime we collected, analyzed, and coded author-assigned keywords of papers. In the works the method of solving the heat equation (ADI) is well considered. Considering the works, programming in the high-level language Python was performed for use in the thermal treatment of oil slime [17, 20, 24].

## 2. The aim and objectives of the study

The aim of the study is to develop a software package for the analysis and intensification of oil slime processing, starting from the study of the characteristics of oil slime and the physical and chemical foundations of the process and ending with the issuance of practical recommendations for the disposal of oil slime to minimize the anthropogenic impact on the environment. The task was set to neutralize oil waste by thermal processing of oil slime to an environmentally friendly level.

- A model has been developed for processing oil slime using convective heating, which makes it possible to evaluate the efficiency and uniformity of heating depending on the temperature and velocity of the flow that heats the oil waste.
- A software implementation of the algorithm for the numerical solution of a non-stationary, inhomogeneous heat conduction problem was implemented using the method of alternating directions according to an implicit iterative scheme until the convergence condition is met.

## 3. Mathematical and numerical modeling

Modeling is one of the main tools for conducting research in the modern world. It allows you to conduct theoretical experiments quickly and cost-effectively, most reliably displaying the objects under study.

The increase in the capabilities of computer technology opens up more and more prospects and opportunities for studying processes and phenomena.

A mathematical model and computer simulation of the heat conduction process and the implementation of the program code that graphically displays the results of the model, that is, the construction of a digital model of the process, for subsequent use as a module, is described here including processing of the graphical representation of data [7]. To achieve this goal, the following tasks are solved:

- Analysis of the subject area using various information sources of scientific content (books, articles, research papers, etc.).
- Construction of a mathematical model of heat propagation in two-dimensional space over time with variable coefficients.
- Development of a software product that simulates the process of temperature change over time and displays the results in graphical form.

When developing a mathematical model for the thermal processing of oil residues, we consider the heating of the product layer. The mathematical model of the process is characterized by heat and mass equations and forms systems of second-order nonlinear differential equations. Primary and boundary conditions include the Dirichlet and Neumann conditions. The solution is implemented using the method of alternating directions according to a non-obvious difference scheme until a convergence condition is met [4].

Numerical heat transfer is a broad term denoting the procedures for the solution, on a computer, of a set of algebraic equations that approximate the differential (and, occasionally, integral) equations describing conduction, convection, and/or radiation heat transfer. The usual objective in any heat transfer calculation is the determination of the rate of heat transfer to or from some surface or object. In conduction problems, this requires finding the temperature gradient in the material at its surface.

When solving a multidimensional problem, the amount of computation with the required accuracy is of great importance. The solution to the posed differential equation includes a combination of several methods and schemes, which, after transformations, allow us to arrive at a numerical solution to the problem.

Explicit and implicit schemes have the same order of precision. The number of actions at all nodes for an explicit

scheme on a new layer is proportional to the number  $(N-1)^2$  of grid nodes:

$$Q_{\text{explicit}} = O\left(\frac{1}{h^2}\right)$$

In the case of an implicit scheme, it is necessary to solve a system of  $(N-1)^2$  equations. This requires significantly more steps than with an explicit scheme:

$$Q_{\text{implicit}} = O\left(\frac{1}{h^4}\right)$$

In terms of stability, the implicit scheme is stable for any values of  $t$  and  $h$ , while the explicit scheme is only stable when  $\tau \leq \frac{1}{4}h^2$

Economical difference schemes, including the scheme alternating directions, combine the advantages of implicit and explicit schemes [10]. Economical schemes for several the most complex problems in physics and technology allow finding a solution to the numerical model [9]. The main idea of such methods is the sequential solution of one-dimensional problems along rows and along columns.

The longitudinal-transverse scheme (implicit method of alternating directions) expresses this idea most clearly.

To solve multidimensional problems that lead to partial differential equations, the method of alternating directions is most often used. This method is an iterative method for solving systems of linear or nonlinear equations in the approximate solution of boundary value problems for partial differential equations in difference or projection-difference methods.

The method is well described in the mathematical literature but attempts to implement it often turn out to be incorrect or inaccurate. Inaccuracy manifests itself when, considering the boundary conditions, their specification at intermediate time steps is neglected. This neglect can cause instabilities even if the scheme used itself is unconditionally stable in terms of spectral characteristics.

There are various methods for solving boundary problems for the thermal equation, including the variable direction method. In the scheme of the method of the variable direction, as shown in all Division methods, the number of independent spatial variables (two in the two-dimensional case) divides the time period  $\tau$ . In each particle-time layer, one of the spatial differential operators is approximated by an implicit (scalar pursuit in the direction of the corresponding coordinate is performed), and the others are implicit. In the subsequent fractional step, the next in turn of the differential operator is executed not explicitly, and the others are executed explicitly [25]. This scheme is often referred to as the Peaceman-Reachford scheme (the name of the authors who first proposed it).  $T(x, t)$  together with the main values of the lattice function, i.e. the value of  $T$  can be considered when  $t = t_{n+\frac{1}{2}} = t_n + \frac{\tau}{2}$ ,  $T = T^n$

and the intermediate value  $T = T^{n+\frac{1}{2}}$  is entered in the middle of  $T = T^{n+1}$ . Where the transition from layer  $n$  to layer  $n+1$  is carried out in two steps  $\tau/2$ . In the two-dimensional case, it can be shown that the scheme of the variable direction method is absolute constant [18].

To solve the heat equation with constant coefficients, an implicit difference scheme can be used, which is constructed by approximating the differential operator with a difference operator on the  $(n+1)$  time layer. The difference scheme approximates the equation with the order  $O(\Delta\tau + h^2)$ . We let  $\Delta\tau/h^2 = r$  and rewrite it in the form (1):

$$A_k T_{k+1} + B_k T_k + C_k T_{k-1} = F_k \quad (1)$$

To check the stability of the circuit according to the maximum principle, we have:

$$A_k = r > 0, C_k = r > 0, -B_k = 1 + 2r > A_k + C_k$$

The last inequality is always fulfilled. This means that the implicit scheme is absolutely stable.

The advantage of the variable direction method is high accuracy since the method has a second order of accuracy in time. The disadvantage can be attributed to the conditional stability when the number of spatial variables is more than two. In addition, the variable direction method is absolutely constant. The mathematical model obtained as a result of heat treatment of oil waste describes the change in temperature and mass of the flow over time and allows its control [16].

Due to the lack of experimental data for development, (as a temporary replacement) modeling of the heat conduction process is used. The heat equation is a second-order partial differential equation that describes the temperature distribution in a given region of space and its change in time.

In mathematical physics, many different problems of heat conduction can be found, differing in the conditions of the process, but to study the work with computer models, the equation of a two-dimensional heat conduction equation with variable coefficients has been chosen.

To solve this equation, the following input parameters will be used, which are required for calculations:

- $\tau$  is the time step;
- $h$  – grid step;
- $x_1, x_2, y_1, y_2$  – values that describe the limits of the region on lines along the OX and OY axes;
- $t$  is the estimated time;
- $N_x, N_y$  – the number of nodes in the grid.

Based on the input parameters, the temperature value is distributed on a coarse mesh uniformly with step  $h$  and bounded by the boundaries  $x_1, x_2, y_1,$  and  $y_2$  along both spatial axes. There are many reasons that affect the values of the thermal conductivity parameters, both internal and external [8]. External causes may include pressure, ambient temperature, magnetic field, etc. Internal causes include the nature of the material, its structure, moisture, density, porosity, and others.

When solving the equation, the following parameters should be obtained:

- $(x, y)$  – coordinates on the plane.
- $T(x, y, t)$  is the temperature value at point  $(x, y)$  at time  $t$ .

Based on the found parameter values, a graphic display can be produced for further analysis and use by the user.

The studies are carried out by methods of mathematical and numerical simulation of thermal processing, the results of which describe changes in temperature and mass of the stream over time. Extensive calculations with varying parameters allow the optimization of the flow of heat and mass transfer during the thermal processing of oil slime.

#### 4. Development of the mathematical part of the problem

When analyzing the subject area and researching solution methods, the following algorithm for the numerical solution of the equation was identified. Figure 1 describes the sequence of steps for calculating heat and mass transfer values in two-dimensional space.

- The values of the initial conditions are entered.
- Using a local one-dimensional scheme and formulas, calculations are performed using the sweep method and the variable direction method.
- The result of the calculations is the values  $T(x, y, t)$  and the corresponding coordinates  $(x, y)$  on a given plane at the calculated time  $t$ .
- The results of calculating the required function are displayed in tabular and graphical forms.

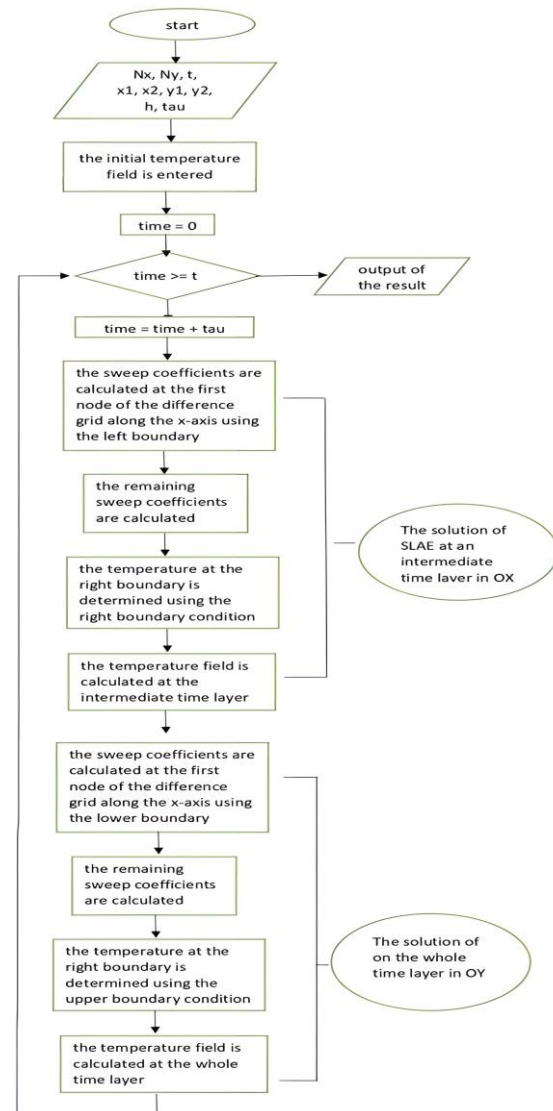


Fig. 1. The block diagram of the calculation algorithm

#### 5. Description of the application

The constant increase in the capabilities of computing opens up great prospects for the study of processes and phenomena of the surrounding world. Computer simulation is, in fact, the simulation implemented using computer technology.

The construction of a computer model is based on abstraction from the object under study. The more significant properties are identified and used in a computer model, the more capabilities a system using this model can have. Computer modeling consists of carrying out a series of computational experiments on a computer, the purpose of which is to analyze, interpret and compare the results with the real behavior of the object and then refine the model, if necessary.

Computer simulation provides a number of possibilities and advantages:

- Study of a wide range of objects, including phenomena past and future, objects that cannot be reproduced in real conditions, non-repeating phenomena, etc.;
- Visualization of abstract objects and objects of nature;
- Conducting multiple tests of the model;
- Conducting experiments without the risk of negative consequences for humans or the environment;
- Finding the optimal design of the object, without making test samples.

The developed software package for solving the problems of oil slime processing has the structure shown in figure 2.

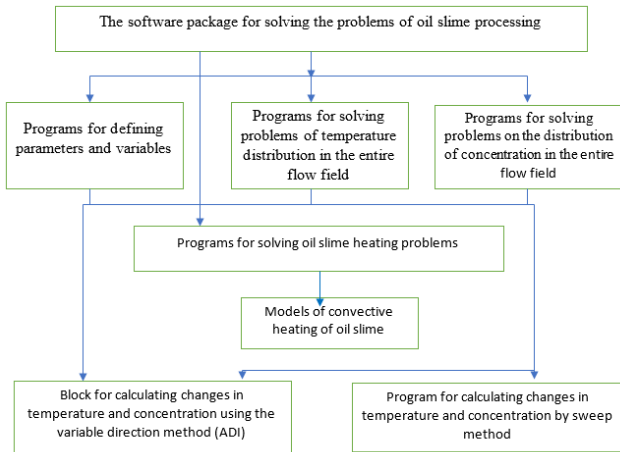


Fig. 2. The structure of the applied application

The main stages of computer modeling of the task are as follows [6]:

- 1) Statement of the problem and its analysis (the purpose of creation, identification of initial data);
- 2) Building an information model (determining the parameters and their relationship, describing the mathematical relationship between the parameters);
- 3) Development of a method and algorithm for the implementation of computer models (selection of a method for obtaining initial data, drawing up an effective algorithm, and checking the correctness of the algorithm);
- 4) Development of a computer model (selection of software tools, development);
- 5) Conducting an experiment (developing a research plan, experiment, and analysis of the results).

During the experiment, it may become clear that it is necessary to make changes, refinements, or improvements to the computer model. In this case, it is necessary to return to the stage, that in the first queue affects the identified problem or task.

The program consists of three main parts:

- databases on the composition of oil slime;
- software, including equations and algorithms for calculating flows and temperatures under stationary conditions and their dynamic changes under various influences;
- a user interface that allows the entry of conditions and seeing the results of calculations.

The following describes the functionality and behavior of the application.

The software is implemented in the Python object-oriented programming environment. When creating the program the following are used: heat transfer theory, numerical methods for solving ordinary differential equations, the method of variable directions, etc.

The web application architecture represents relationships and interactions between components such as user interfaces, transaction processing monitors, databases, and others (Fig. 4).

The developed application allows for computer modeling of heat and mass transfer in soils and grounds, the spread of pollutants in the environment in order to further minimize environmental risks.

We've tackled interacting with databases using Flask-SQLAlchemy with advanced Authentication using Flask-Login (Figure 5).

It used necessary libraries: sqlalchemy, flask-sqlalchemy, flask-login, werkzeug, and configparser.

The file app.py contains the code for the Dash application and will be composed of several different layouts to accommodate creating the user account, logging in, and displaying the graphed data.

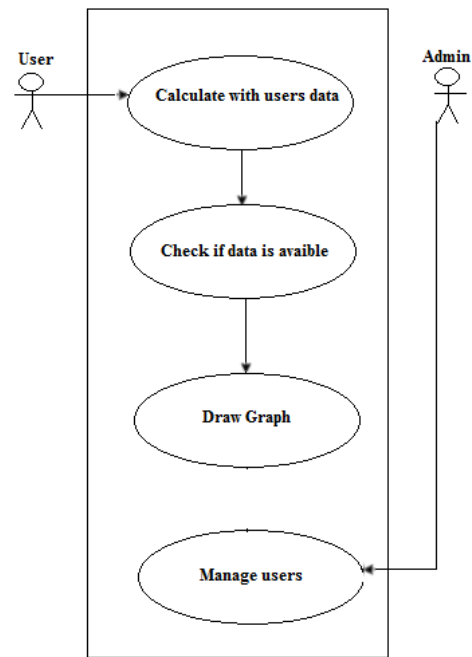


Fig. 3. The description of functionality and behavior of the application

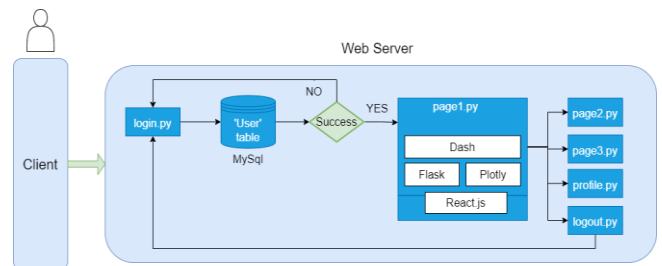


Fig. 4. The architecture of the web application

The users\_mgt.py file is the database and stores the username, password, and email address in the Users table. To protect user passwords, the password will be hashed using the werkzeug library. Werkzeug is an advanced web server gateway interface (WSGI) utilities library. Configuring for Flask-Login is kept in the config.py file.

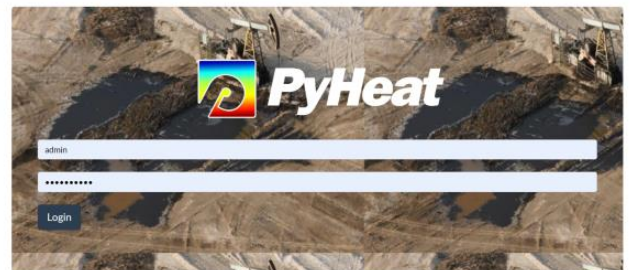


Fig. 5. Web application login

## 6. Results

Information visualization is the process of presenting abstract data in the form of images that can help in understanding the meaning of the data. Data visualization helps one to present large and complex datasets in a simple and visual way. At the end of the project, it is important to be able to report on its results so that even for non-professionals who do not have the technical knowledge, everything becomes clear and understandable [23].

Data visualization is an integral part of the application. It describes the data visualization framework in Python-Dash, which permits working not only with interactive graphs but also to display them on a website. Dash application code is declarative

and responsive, making it easy to create complex applications that contain many interactive elements [3]. The user interface library Dash will be useful for those who use Python for data analysis, exploration, visualization, modeling, and reporting [11].

Dash makes it much easier to create GUIs (Graphical User Interfaces) for data analysis.

There was no need to write any JavaScript or HTML code when a Dash application is running in a web browser. Dash provides a rich set of interactive web components. Dash provides a simple reactive decorator to bind data analysis code to the Dash UI. When an input element changes (for example, when an item is selected in a dropdown list or when a slider is moved), the Dash decorator provides Python code with a new input value.

Below is the code of the Python application using Dash:

```
import dash
import dash_bootstrap_components as dbc
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
import plotly.graph_objects as go
app = dash.Dash()
app.layout = html.Div([
```

The following are specified in the code:

The application is initialized via `dash.Dash`, where specified the app name.

The layout attribute defines what will be on the web page, and only the Div header will be displayed on the page. All HTML components are in the `dash_html_components` module and describe high-level components that are interactive and generated using JavaScript, HTML, and CSS via the library `React.js`.

The application is started via `run server`. `debug=True` was set to be able to debug the code. If the application is ready to run on the server, then, for security reasons, the flag should be set to `False`.

The application is started via the `run server`. `debug=True` was set to be able to debug the code. If the application is ready to run on the server, then, for security reasons, the flag should be set to `False`.

The imported libraries that are needed for the dash system to work, and the python library for creating administrative systems, are based on `React`. Dash applications running under `Flask` and the external Dash interface display components using `React.js`, which are Javascript user interface libraries. The style for the menu and content was defined as follows:

# the style arguments for the sidebar

```
SIDEBAR_STYLE = {
    "position": "fixed",
    "top": 0,
    "left": 0,
    "bottom": 0,
    "width": "16rem",
    "padding": "2rem 1rem",
    "background-color": "#f8f9fa",
}
```

# the styles for the main content position it to the right of the sidebar and

# add some padding.

```
CONTENT_STYLE = {
    "margin-left": "18rem",
    "margin-right": "2rem",
    "padding": "2rem 1rem",
}
```

To make the time-series graph interactive, it is necessary to create a callback function for the drop-down menu and the output space. However, the hyphen does not allow callbacks for components that are not present in the layout.

To make apps interactive so that things actually happen when the user clicks on them, it needs two callbacks, one to return the different pages of the app, and the other to update the graph.

To do this, it was declared `@app.callback`. Inside `@app.callback`, we pass the Output function that is imported at the top of the file. Output parameters:

1. ID of the component that is being updated.
2. The parameter of the component that is being updated.

Then it was passed the Input function to `app.callback`.

```
@app.callback(
    [Output(f"page-i-link", "active") for i in range(1, 4)],
    [Input("url", "pathname")],
)
def toggle_active_links(pathname):
    if pathname == "/":
        # Treat page 1 as the homepage / index
        return True, False, False
    return [pathname == f"/page-i"] for i in range(1, 4)]
```

Implemented a Dash callback to display graphs that describe changes in temperature and mass flow over time. There is also a slider on the page where you can choose from the composition of the oil slime. This is the essence of reactive programming via callbacks – when the input changes, so do the output.

To update the graph according to the dropdown selection, it is necessary to establish a link between the input data and the output (graph). This will be achieved by adding a callback function.

The following is a callback for a time-series graph.

```
@app.callback([ Output('graph', 'figure'),
                Output('graph1', 'figure')],[Input('input1', 'value'),Input('input2', 'value'),Input('input3', 'value'),Input('input4', 'value'),Input('input5', 'value'),Input('input6', 'value'),Input('input7', 'value')])
```

A Dash application can also have multiple Outputs, i.e. display multiple updatable components on a web page. It is just necessary to put a list of Outputs in the callback decorator, which will be updated depending on the Inputs. The function to be decorated must return as many Output values as specified in the decorator.

```
[Output(f"page-i-link", "active") for i in range(1, 4)],
[Input("url", "pathname")],
)
def toggle_active_links(pathname):
    if pathname == "/":
        # Treat page 1 as the homepage / index
        return True, False, False
    return [pathname == f"/page-i"] for i in range(1, 4)]
```

The Dash code itself is declarative and reactive, so creating complex applications that include many interactive elements is easier. In addition, Dash provides a simple reactive decorator that allows the binding of data analysis code to the Dash user interface. If the input element changes, the Dash decorator provides the Python code with a new input value. So a Python function can perform different actions with the new input value and it can see the results in figure 6.

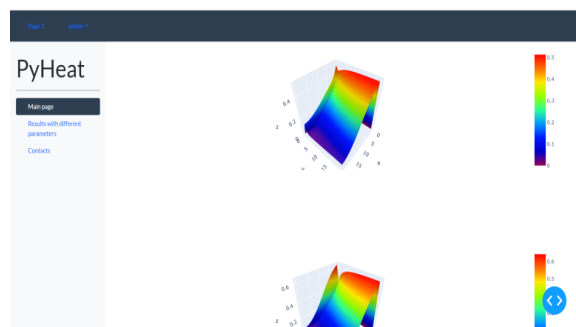


Fig. 6. View of the interactive web application

Dash application properties are stored in the interface (in the browser). This allows the use of applications written using Dash in multi-user mode: several independent sessions can be opened, in which the actions of one user will not affect the data

of other users. Dash application code is functional: it can read values from Python global properties, but cannot modify them. This functional approach can be easily justified and tested – it is just inputs and outputs without any side effects or properties.

The main part of the developed code is based on the above flowchart and uses the methods mentioned above. Solution of the heat and mass distribution equation is performed via the python Def Calculate function.

```
@ def Calculate(l,h,n,m,ti,u0i,vi,T0i)
....
....
fig =
go.Figure(data=[go.Surface(z=T,colorscale='Rainbow')])
return fig
```

The function def Calculate (l, n, ti, u0i, vi, T0i, C0i) is used to process data, it receives a value as input, processes it, and returns the result to the program.

The function variables are:

T – temperature;

t – time;

u – the projection of the velocity vector on the Ox axis;

v – the projection of the velocity vector on the Oy axis;

l – length of oil slime;

h – width of oil slime ;

N – number of points by coordinates.

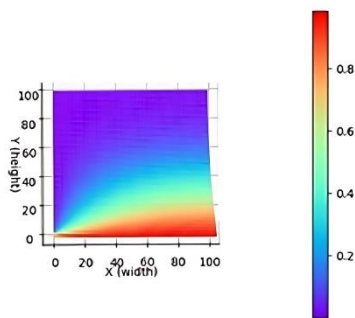


Fig. 7. The graph of oil slime temperature change at temperatures of 250°C at a velocity of 0.83 m/s

Using the statement *return* from the function, it returns the value fig. In the course of the program, calculations are made for the problem of thermal processing of oil slime under consideration. The results of the program are displayed in graphical (Fig. 7–9) forms. In the work of the program used  $\tau = 0.1$ ,  $h = l = 0.5$  m,  $t = 30$ ,  $N_x * N_y = 100 * 100$ .

The main way to present the results is to graph and color the entire area depending on the value of the calculation results.

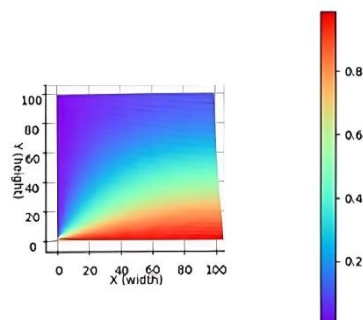


Fig. 8. The graph of oil slime temperature change at temperatures of 250°C at a velocity of 1.6 m/s

To construct graphic elements, the Dash image generation library was used. Also, the written program has the ability to save the construction results in the .png raster image format, download plot as a png, zoom, pan, orbital rotation, turntable rotation, reset the camera to default, reset the camera to last save, toggle show

the closest data on hover. These ways of presenting the results help to evaluate better the course of the modeled process: its speed and area of propagation under given conditions.

The graphs are built in the `plotly.graph_objects` module where there are special objects called graph objects. These objects can also be used to represent shapes, and they have some advantages over dictionaries:

Graphical objects provide accurate data validation. If an invalid property name or value is specified, an exception will be thrown with a clear description of the problem;

Using the statement *return* from the function, it returns the value fig. Graphical objects have Python documentation with complete API reference. Therefore, it is easy to find out about the available properties right in the development environment.

The properties of graphical objects can be accessed either by searching by a key in the dictionary (fig) or by using a point call (fig.layout);

Graphics objects support higher-level helper functions for updating already drawn shapes (`.update_layout()`, `.add_trace()`, etc.);

The mass loss histories depicted in Figures 7–9 were plotted for conditions where a common air stream temperature and slab size were kept constant in each. Thus, the graphs show the stream velocity effect on the process of mass transfer. In the course of convective heating, the faster-moving fluid improves, obviously, the heat transfer characteristics in the system, which consequently, enhances the vaporization and diffusion drives within the heated oil slimes.

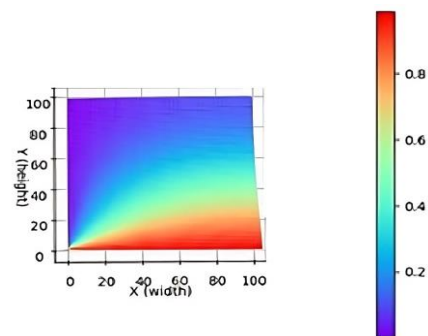


Fig. 9. The graph of oil slime temperature change at temperatures of 250°C at a velocity of 2.7 m/s

Also, the higher velocity fluids at the heated slab, the surface entrains more vapors with their higher momentum. This led to the relatively higher rates of volatilization obtained with higher stream velocities. The ultimate limits of mass loss are expected to be unaffected by the stream velocity, since these limits are controlled solely by the maximum temperatures ultimately achieved within the system. However, the slight effect noticeable in the three graphs could be, at least in part, attributed to the improved heat transfer conditions associated with the higher stream velocities, which might have resulted in slightly higher temperatures within the heated slabs due to the expected thinner boundary layers developed on the slab surface.

Covered in the present study to check this matter further, oil slimes were exposed to air streams at temperatures of 150, 250, 350, and 450°C and stream velocities 0.83, 1.6, and 2.7 m/s.

## 7. Conclusion

The study presented a detailed analysis of the methods and technologies for processing oil slime, based on which a thermal method was chosen for processing oil slime in order to minimize the impact of waste on the environment. A mathematical model has been built that describes the processes of heat and mass transfer during the evaporation of harmful substances from the oil slime reservoir. Mathematical calculations of the given differential equation were implemented with its reduction to a numerical solution in the form of a system of equations using solution methods.

The developed computer model includes not only the calculation of heat propagation in two-dimensional space over time, but also the construction of a graphical representation of the results obtained in the form of tables and graphs. This allows the user to look at the resulting model in several ways and makes it possible to analyze the results obtained using different approaches.

The developed application implements models and algorithms for management decisions in refining processes for the oil and gas industry and provides a network interface that can be used in the future in the industrial oil and gas sector.

This proposal for the processing of oil slime supports the application of effective technologies to solve environmental problems.

## 8. Acknowledgments

This research was funded by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan (Grant #AP09259208)

## References

- [1] Abdrabboh M. A.: Studies in Heat and Mass Transfer in Oil Sand Beds. Ph.D. thesis. University of Calgary, Calgary, Alberta, Canada, 1983.
- [2] Abimbola A., Bright S.: Alternating-Direction Implicit Finite-Difference Method for Transient 2D Heat Transfer in a Metal Bar using Finite Difference Method. *International Journal of Scientific & Engineering Research* 6(6), 2015.
- [3] Altybay A. et al.: Software Application for the Investigation of the Wave Propagation in 1d and 2d Wave Equations with Singular Coefficients. *Telematique* 21(1), 2022, 7468–7474.
- [4] Anderson D. A. et al.: *Computational Fluid Mechanics and Heat Transfer*. 2nd ed., Taylor & Francis, 1977.
- [5] Badrul I.: Petroleum sludge, its treatment and disposal: a review. *Int. J. Chem. Sci.* 13(4), 2015, 1584–1602.
- [6] Balakayeva G. et al.: Digitalization of enterprise with ensuring stability and reliability. *Informatics, Control, Measurement in Economy and Environmental Protection* 13(1), 2023, 54–57 [http://doi.org/10.35784/iapgos.3295].
- [7] Balakayeva G., Darkenbayev D.: The solution to the problem of processing big data using the example of assessing the solvency of borrowers. *Journal of Theoretical and Applied Information Technology* 98(13), 2020, 2659–2670.
- [8] Balakayeva G. T. et al.: Numerical modeling of heat and mass transfer in a reactor for continuous movement of oxidation of oil slime. *Chemical Bulletin of Kazakh National University* 3, 2000, 47–55.
- [9] Balakayeva G.T. et al.: Using NoSQL for processing unstructured BigData. *News of the Republic of Kazakhstan series of Geology and Technical sciences* 6(438), 2019, 12–21.
- [10] Chang M. J. et al.: Improved alternating-direction implicit method for solving transient three-dimensional heat diffusion problems. *Numerical Heat Transfer, Part B: Fundamentals*, 1991, 69–84.
- [11] Dash User Guide, 2022 [https://dash.plotly.com].
- [12] Dong X. et al.: Treatment of layered oily sludge by a thermal steam process. 13th International Conference on Waste Management and Technology (ICWMT 13), Beijing, China, 2018.
- [13] Egazaryants S. V. et al.: Oil slime Treatment Processes. *Chemistry and Technology of Fuels and Oils* 5(51), 2015, 506–515.
- [14] Hanafi A. S.: Experimental and Analytical Studies of Volatilization and Burning of Oil Sand Particles. Ph.D. thesis, The University of Calgary, Calgary, Alberta, Canada, 1979.
- [15] Hu G. et al.: Recent development in the treatment of oil slime from petroleum industry. *Journal of Hazardous Materials* 2(61), 2014, 470–490.
- [16] Jadidi N. et al.: The Most Recent Researches in Oily Sludge Remediation Process. *Am. Journal of Oil and Chemical Technologies* 2(10), 2019, 340–348.
- [17] Johnson O. A., Affam A. C.: Petroleum sludge treatment and disposal: A review. *Environmental Engineering Research* 24(2), 2019, 191–201 [http://doi.org/10.4491/eer.2018.134].
- [18] Kalmenova G. B., Balakayeva G. T.: Developing a model of oil slime processing. *Bulletin of KazNTU. Series Physics and mathematics* 3, 2019, 552–555.
- [19] Kunlong H. et al.: Status and prospect of oil recovery from oily sludge: A review. *Arabian Journal of Chemistry* 3(8), 2020, 6523–6543.
- [20] Lubanovic B.: *Introducing Python: Modern Computing in Simple Packages* 2nd Edition. O'Reilly Media, 2019.
- [21] Ma Z. et al.: Modeling and Simulation of Oil slime Pyrolysis in a Rotary Kiln with a Solid Heat Carrier: Considering the Particle Motion and Reaction Kinetics. *Energy & Fuels* 28(9), 2014, 6029–6037 [http://doi:10.1021/ef501263m].
- [22] Norzilah A. H., Nursalasawati R.: Alternating Direction Implicit (ADI) Method for Solving Two Dimensional (2-D) Transient Heat Equation. *ASM Sci. J., Special for SKSM* 26(6), 2019, 28–33.
- [23] Python interface to Tcl/Tk, Python Documentation, 2019 [https://docs.python.org/3/library/tkinter.html].
- [24] Romano F. et al.: *Learn Web Development with Python*/Packt Publishing Ltd, 2018.
- [25] Shan Z.: A Matched, Alternating Direction Implicit (ADI) Method for Solving the Heat Equation with Interfaces. *Journal of Scientific Computing* 63(1), 2015, 118–137.
- [26] Yao W.: Research and application of oil slime resource utilization technology in the oil field. *IOP Conference Series Earth and Environmental Science* 170(3), 2018 [http://doi:10.1088/1755-1315/170/3/032026].
- [27] Zhang X. et al.: Thermal Desorption Process Simulation and Effect Prediction of Oil-Based Cuttings. *ACS Omega* 7(25), 2022, 21675–21683 [http://doi:10.1021/acsomega.2c01597].
- [28] Ziyi W. et al.: Application and development of pyrolysis technology in petroleum oily sludge treatment. *Environmental Engineering Research* 26(1), 2021 [http://doi.org/10.4491/eer.2019.460].

**Prof. Gulnar Balakayeva**  
e-mail: gulnardsa@gmail.com

Doctor of Physical and Mathematical Sciences, professor of the Department of Computer Science, Faculty of Information Technologies, Al-Farabi Kazakh National University, Almaty, Kazakhstan. Research interests: development of big data processing systems, modeling of physical and chemical processes.



<http://orcid.org/0000-0001-9440-2171>

**M.Sc. Gaukhar Kalmenova**  
e-mail: kalmenova.g.b@gmail.com

Ph.D. student of the Department of Computer Science, Faculty of Information Technologies, Al-Farabi Kazakh National University, Almaty, Kazakhstan. Research interests: computer technologies, software engineering, mathematical and computer modeling, numerical computing methods.



<http://orcid.org/0000-0002-4124-2423>

**Ph.D. Dauren Darkenbayev**  
e-mail: dauren.kadyrovich@gmail.com

Ph.D., associate professor of the Department of Computer Science, Faculty of Information Technologies, Al-Farabi Kazakh National University, Almaty, Kazakhstan. Research interests: big data processing, mathematical and computer modeling, development of computer systems for the educational process.



<http://orcid.org/0000-0002-6491-8043>

**Prof. Christofer Phillips**  
e-mail: chris.phillips@newcastle.ac.uk

Ph.D., professor. University of Newcastle upon Tyne, Newcastle, United Kingdom. Research interests: Big Data processing, mathematical and computer modeling, numerical computing methods.



<http://orcid.org/0000-0002-2470-1659>