IRSTI 28.17.15
UDC 004.94 **DOI 10.70239/arsu.2026.t83.n1.08**

NUMERICAL MODELING OF THE TWO-TEMPERATURE PLASMA EQUATION IN THE PYTHON PROGRAMMING ENVIRONMENT

NURULLAEV J.A.*, BAWETDINOV A.A.

***Nurullaev Jusipbay Abdiualiyevich** – PhD, Karakalpak state university named after Berdakh, Nukus, Uzbekistan
E-mail: njusipbay@mail.ru, https://orcid.org/0000-0003-0935-5388
Bawetdinov Asqar Axmet uli – Master's student, Karakalpak state university named after Berdakh, Nukus, Uzbekistan
E-mail: bawetdinovasqar2003@gmail.com, https://orcid.org/0009-0000-6841-9041

Abstract. This study presents numerical modeling and simulation of a two-temperature plasma system using the Python programming environment. This paper presents numerical modeling and simulation of a two-temperature plasma system using the Python programming environment. The model is based on a connected system of nonlinear equations of thermal conductivity describing the processes of energy exchange between electronic and ionic components. To obtain numerical solutions, an implicit finite-difference Crank–Nicholson scheme was implemented, providing second-order accuracy in time and space. The computational process was carried out using the NumPy, SciPy and SymPy libraries, and visualization of the results and user interaction were implemented using the Mathplotlib and Tkinter tools. The developed program allows you to interactively enter parameters, calculate temperature evolution and visualize the results in real time. Numerical experiments were carried out for the values of parameters corresponding to the natural and climatic conditions of the Republic of Karakalpakstan, where the density of the solar energy flux varies in the range of 1000-2800 W/m2. The results show the changes over time and space in electron and ion temperatures, confirming the Crank–Nicholson approach and the reliability of the developed software for modeling energy transfer in low-pressure plasma conditions.

Key words: two-temperature plasma, Crank–Nicolson scheme, numerical modeling, heat conduction equations, Python programming, electron–ion energy exchange, thermal diffusion, computational simulation.

Introduction

The study of plasma processes remains an important area of research in applied physics and computational mathematics. In many practical situations, especially under low pressure and disequilibrium conditions, the temperatures of electrons and ions are not equal. Such conditions lead to the development of two-temperature plasma models, in which each component has its own energy balance equation.

In contrast to the simplified single-temperature approaches, the two-temperature description allows for a more realistic representation of the mechanisms of energy exchange. The interaction between light electrons and relatively heavy ions significantly affects the transport properties, thermal diffusion, and general dynamics of the plasma. These effects are becoming particularly relevant in applications related to energy transfer, interaction with solar radiation and laboratory plasma systems.

Analytical consideration of related nonlinear equations of thermal conductivity is usually limited to highly simplified cases. When realistic parameters and boundary conditions are introduced, analytical solutions are usually unavailable. For this reason, numerical methods play a central role in the study of such systems. Among the various approaches to discretization, implicit finite-difference schemes are often preferred because of their stability properties when solving diffusion-type equations.

In this work, a coupled two-temperature plasma model is considered and implemented in the Python programming environment. The Crank–Nicolson scheme is applied to approximate the governing equations in both space and time. Particular attention is paid to the numerical stability of the method and to the selection of parameters that correspond to regional climatic conditions [1-11].

The main objective of the study is to construct a computational model that allows simulation of electron–ion energy exchange under physically meaningful assumptions. To achieve this objective, the mathematical formulation of the model is specified, the numerical algorithm is implemented, and a series

of computational experiments is performed to analyse temperature evolution [7].

Materials and methods of research

In this study, numerical modelling was carried out based on the following relatively simplified heat conduction equations that describe a two-temperature plasma system [6, 9].

Problem formulation:

$$\frac{\partial T_e}{\partial t} = D_e \frac{\partial^2 T_e}{\partial x^2} \alpha\,(T_e\ -\ T_i) + S_e(x,t) \quad (1)$$

$$\frac{\partial T_i}{\partial t} = D_i \frac{\partial^2 T_i}{\partial x^2} + \alpha(T_e - T_i) + S_i(x,t) \quad (2)$$

Here:

$T_e(x,t)$ – electron temperature

$T_i(x,t)$ – ion temperature

D_e, D_i – electron and ion thermal conductivity coefficients (W/(m·K))

α – energy exchange coefficient between electrons and ions

$S_e(x,t), S_i(x,t)$ – external energy sources (such as solar radiation, electric charge flow, and others)

The system simultaneously accounts for heat diffusion and the exchange processes between the temperatures. The model is discretized on a 1D spatial coordinate (e.g. $x \in [0, L]$) and and on the time interval $t \in [0, T]$. For the discretization, an implicit Crank–Nicolson numerical differential scheme was employed, implemented using the NumPy and SciPy libraries within the Python programming environment.

The Crank–Nicolson scheme provides a solution with second-order accuracy.

Steps for applying the Crank–Nicolson scheme to equations (1) and (2):

Discretization of the domain in time and space:

- Spatial grid nodes: $x_j = j\Delta x, j = 0,1, \dots, N$
- Temporal grid nodes: $t^n = n\Delta t, n = 0,1, \dots, M$
- $T_e^n[j]$ – the electron temperature at time t^n and position x_j

The Crank–Nicolson scheme (semi-implicit). The scheme is computed for each j based on the following formulas:

For electrons:

$$\frac{T_e^{n+1}[j] - T_e^n[j]}{\Delta t} = \\ = D_e \cdot \frac{1}{2}\left(\frac{T_в^{n+1}[j+1] - 2T_e^{n+1}[j] + T_e^{n+1}[j-1]}{(\Delta x)^2} + \frac{T_в^n[j+1] - 2T_в^n[j] + T_в^n[j-1]}{(\Delta x)^2}\right) - \\ -\alpha\frac{(T_e^{n+1}[j] + T_e^n[j] - T_i^{n+1}[j] - T_i^n[j])}{2} + \frac{S_в(x_j, t^n) + S_e(x_j, t^{n+1})}{2} \quad (3)$$

For ions:

$$\frac{T_i^{n+1}[j] - T_i^n[j]}{\Delta t} = \\ = D_e \cdot \frac{1}{2}\left(\frac{T_i^{n+1}[j+1] - 2T_i^{n+1}[j] + T_i^{n+1}[j-1]}{(\Delta x)^2} + \frac{T_i^n[j+1] - 2T_i^n[j] + T_i^n[j-1]}{(\Delta x)^2}\right) + \\ -\alpha\frac{(T_e^{n+1}[j] + T_e^n[j] - T_i^{n+1}[j] - T_i^n[j])}{2} + \frac{S_i(x_j, t^n) + S_i(x_j, t^{n+1})}{2} \quad (4)$$

Transforming the equation into matrix form at each time step, T_e^{n+1} and T_i^{n+1} are determined a tridiagonal system of linear equations is obtained:

$$A \cdot T_e^{n+1} = B \cdot T_e^n + f \tag{5}$$

$$A \cdot T_i^{n+1} = B \cdot T_i^n + g \tag{6}$$

Here, A and B – are tridiagonal matrices, while f and g – represent external source (forcing) functions (along with $S_в, S_i$, and α).

Boundary conditions:

- $T_e[0] = T_e[N] = 0$ (or another Dirichlet condition)
  $T_i[0] = T_i[N] = 0$

Software product and computing environment (Python). Modern computational tools and programming languages play a central role in the numerical investigation of two-temperature plasma models. Such models are typically expressed as complex systems of differential equations that are difficult or impossible to solve analytically and therefore require numerical methods. For this reason, the numerical modelling in this work employed the Python programming environment and contemporary computational approaches [7, 12]:

Python was selected as the computational platform for implementing the proposed numerical model. The choice was mainly motivated by the availability of stable numerical routines and convenient tools for data processing and visualisation.

Array operations and matrix manipulations required for the finite-difference implementation were performed using NumPy. Time integration procedures were carried out with functions provided in the SciPy library, including routines from the scipy.integrate module. In several stages of the work, symbolic transformations were verified with the help of SymPy in order to check intermediate analytical expressions.

Visualization of numerical results was organized through Matplotlib, which allowed comparison of temperature profiles under different parameter sets. The computational experiments were conducted in the Jupyter Notebook environment, providing flexibility during testing and adjustment of the algorithm.

To simplify interaction with the program, a basic graphical interface was implemented using Tkinter. This interface enables parameter input and execution of calculations without modifying the source code directly. The program structure was organized into separate functional components, which makes it possible to adjust physical parameters or extend the model with minimal changes to the overall implementation [12].

Parameter selection and adjustment. In the numerical analysis of two-temperature plasma models, the accuracy of the physical parameters incorporated into the model is of critical importance. These parameters determine the model's level of physical realism, the reliability of numerical computations, and the practical significance of the final results. In this study, the following key parameters were selected:

- Electron temperature $T_e[K]$
- Ion temperature $T_i[K]$
- Particle density $n[m^{-3}]$
- Thermal conductivity coefficient $K[W/(m \cdot K)]$
- Diffusion coefficient $D[m^2/s]$
- Incident external energy flux $Q[W/m^2]$

The above parameters were selected based on the natural, climatic, and physico-geographical

conditions of the Republic of Karakalpakstan. Specifically, the average solar energy flux varies within the range $Q \approx 800 \div 1000$ W/m^2 throughout the year [13]. These values are consistent with the climate data provided by NASA EarthData and the Ministry of Innovative Development of the Republic of Uzbekistan.

In addition, the electron and ion temperatures were defined within the following ranges:

- • Ion temperature: $T_i \approx 1000 \div 3000$ K
- • Electron temperature: $T_e \approx 10^4 \div 2 \cdot 10^4$ K

These values are characteristic of low-pressure plasma conditions, where the difference between them reflects the two-temperature nature of the plasma. Since electrons are lighter particles, they absorb more energy and move faster, resulting in $T_e > T_i$ [5].

The particle density nnn determines the degree of plasma ionization. In the model, values in the range of $n \approx 10^{17} \div 10^{20}$ m^{-3} were used, which correspond to typical laboratory plasma conditions [13].

The thermal conductivity coefficient K and the diffusion coefficient D determine the heat and particle transport properties of the plasma. They are calculated based on the following equations:

$$D = \frac{k_B \cdot T}{m \cdot \nu}$$

$$K_e = \frac{1.96\, n_e\, k_B\, \lambda_e\, T_e^{1/2}}{m_e^{1/2}} \quad (7)$$

Here, $k_в$is the Boltzmann constant, m is the particle mass, is the particle collision frequency, and $\lambda_в$ is the mean free path of electrons. These parameters are determined based on the kinetic theory of plasma and exhibit a strong dependence on temperature and density [5].

Table 1. Numerical values of physical parameters used in the plasma simulation

Parameter	Values	Explanation
D_e	$1.5 \cdot 10^{-2}$ m²/s	Electron thermal diffusion
D_i	$3.0 \cdot 10^{-3}$ m²/s	Ion thermal diffusion
α	0.02 s⁻¹	Electron–ion energy exchange
Q	900 W/m²	Solar energy influx
$T_e(0)$	12000 K	Initial electron temperature
$T_i(0)$	1000 K	Initial ion temperature

In the study, the temporal and spatial variations of these parameters were also taken into account, as the physical state of the plasma changes under external influences such as energy sources, convection, or radiation. Therefore, the following functional dependencies of the parameters are introduced:

$$T_в(x,t), \qquad T_i(x,t), \quad n(x,t), \quad Q(t)$$

Such an approach enhances the realism of plasma models and enables more reliable simulation of energy systems under the conditions of Karakalpakstan.

Results and its discussion

Figure 1 presents the Python program based on the two-temperature plasma equation. The program performs the following tasks:

- Accepts parameter inputs from the user via Tkinter.
- Computes a numerical (approximate) solution of the two-temperature plasma equation.
- Produces results that closely approximate the analytical solution under ideal conditions.
- Saves the computed solutions to a file.
- Plotting the graphs of $T_в(t)$, $T_i(t)$ and their differences in graphical form.

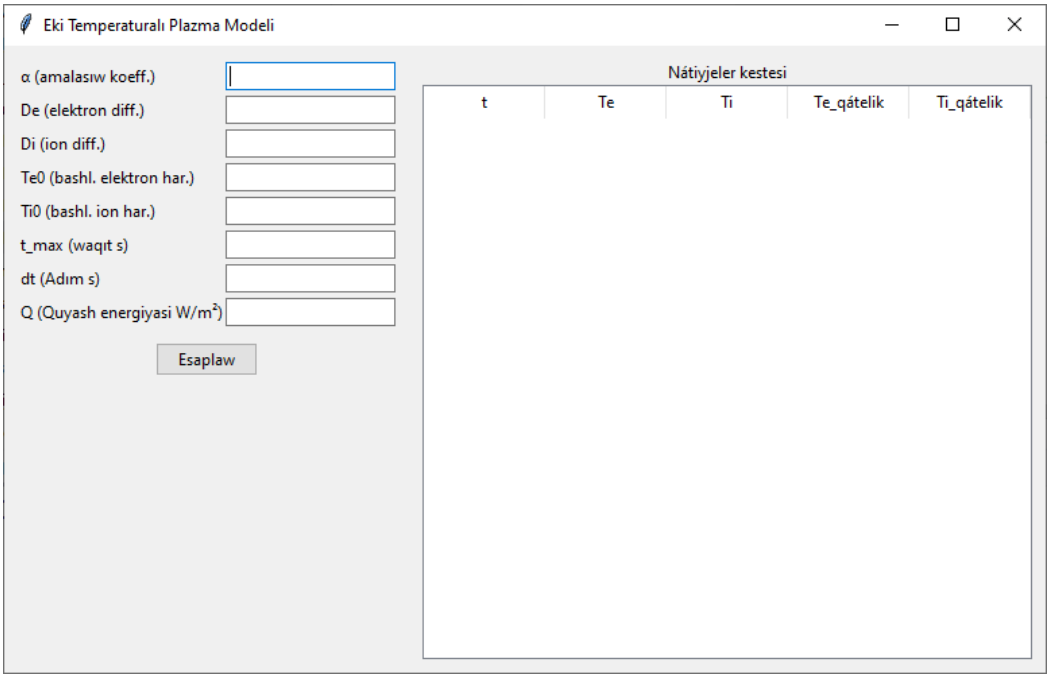

Figure 1. Two-temperature plasma model

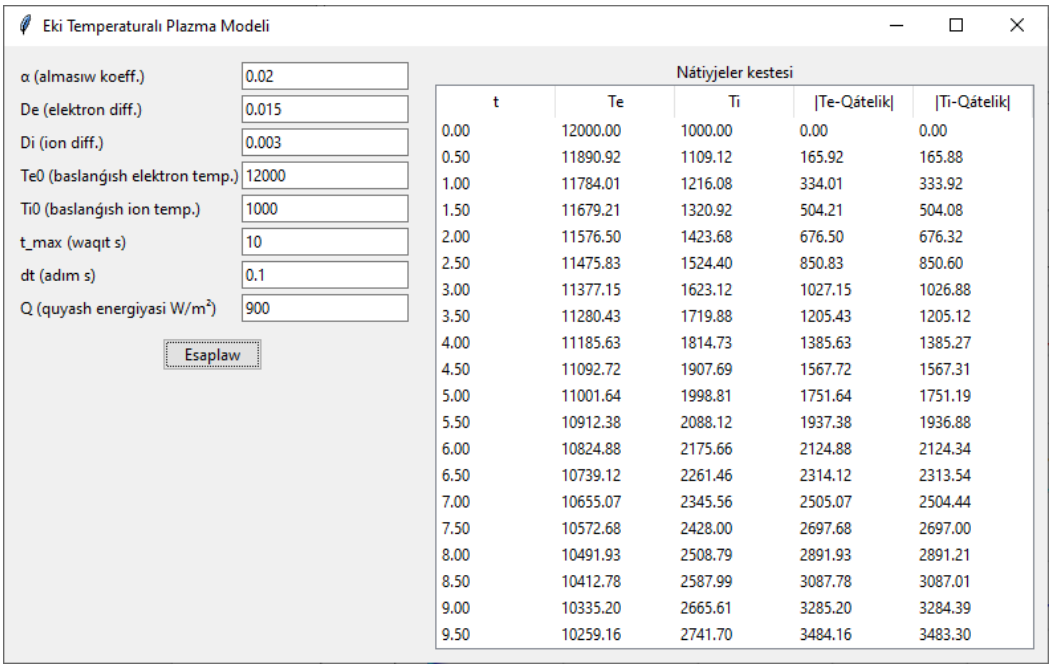

Figure 2. Parameter input and results

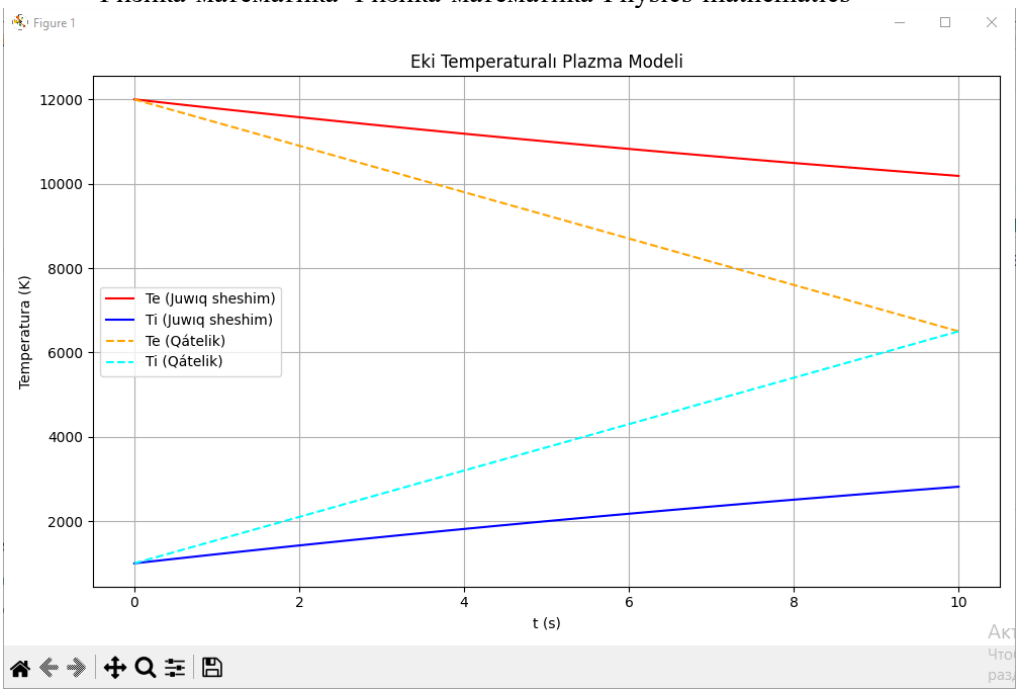

Figure 3. Graphical results

The software also provides the capability to save the obtained numerical results to a file. Once the Calculate button is pressed, the results are automatically written to the file.

Conclusion

In this article we have formulated the numerical model of a two-temperature plasma system and implemented it in the Python programming environment. The numerical analysis used in this work gives an insight into the possibility of simulating two temperature plasma using a fully implicit finite-difference method. The Crank–Nicolson scheme provides stable time integration and adequate numerical accuracy for the class of diffusion-type equations under consideration.

Literally basic physics is proven with computational experiments. Specifically, we highlight that the electron temperature develops quickly upon the input of external energy while the ion component evolves slowly due to their relatively large mass. The interaction term only allows for gradual energy redistribution between the two subsystems, so it promotes thermal equilibration at long timescales.

The developed Python implementation proved convenient for testing different parameter configurations and observing their influence on the solution. The modular structure of the code allows further extensions of the model without substantial modifications to the core algorithm.

The presented approach may serve as a basis for more advanced investigations, including multidimensional extensions, incorporation of additional physical effects, or refinement of boundary conditions. Further research may also focus on improving computational efficiency and adapting the model for practical plasma-based energy applications.

References

1. Samarskii A.A. The Theory of Difference Schemes. New York: Marcel Dekker, 2001. 786 p.

2. Miroshnichenko A.P. Fizika plazmy. Moskva: Nauka, 2004. 368 s.

3. Aripov M.M., Utebaev D., Nurullaev J.A. Convergence of high-precision finite element method schemes for the two-temperature plasma equation // AIP Conference Proceedings, 2021, Vol. 2325, 020059. DOI: 10.1063/5.0040505

4. Crank J., Nicolson P. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type // Advances in Computational Mathematics, 1996, Vol.

6, No. 1, Pp. 207-226. DOI: 10.1007/BF02127704 (Reprinted from Proceedings of the Cambridge Philosophical Society, 1947, Vol. 43, No. 1, Pp. 50-67)

5. Chen F.F. Introduction to Plasma Physics and Controlled Fusion. 3rd ed. Cham: Springer International Publishing, 2016. 490 p. DOI: 10.1007/978-3-319-22309-4

6. Utebaev D., Atadjanov Kh.L., Nurullaev J.A. Finite element method schemes of higher accuracy for solving non-stationary fourth-order equations // Bulletin of KazNU. Mathematics, Mechanics and Computer Science, 2023, Vol. 118, No. 2, Pp. 42-56.

7. Steinkamp V. Python for Engineering and Scientific Computing. Bonn: Rheinwerk Publishing, 2024. 511 p.

8. Moskalkov M.N., Utebaev D. Comparison of Some Methods for Solving the Internal Wave Propagation Problem in a Weakly Stratified Fluid // Mathematical Models and Computer Simulations, 2011, Vol. 3, No. 2, Pp. 264-271. DOI: 10.1134/S207004821102010X

9. Gabov S.A. Novye zadachi matematicheskoy teorii voln. M: Nauka, 1998. 448 s.

10. Wu P., Xie Y., Jiang H., Niu L., Natsuki T. Computationally efficient complex envelope approximate Crank–Nicolson scheme and its open region problem for anisotropic gyrotropic plasma // Physics of Plasmas, 2020, Vol. 27, No. 10, 102109. DOI: 10.1063/5.0020205

11. Wu P., Xie Y., Jiang H., HanYi D., Natsuki T. Bandpass approximate Crank-Nicolson implementation for anisotropic gyrotropic plasma open region simulation // Optik, 2021, Vol. 242, 166941. DOI: 10.1016/j.ijleo.2021.166941

12. Python Software Foundation. Python Documentation. https://docs.python.org

13. NASA Surface Meteorology and Solar Energy Data. https://power.larc.nasa.gov/

Әдебиеттер тізімі

1. Samarskii A. A. The Theory of Difference Schemes. – New York: Marcel Dekker, 2001. – 786 p.

2. Мирошниченко А. П. Физика плазмы. – Москва: Наука, 2004. – 368 с.

3. Aripov M. M., Utebaev D., Nurullaev J. A. Convergence of high-precision finite element method schemes for the two-temperature plasma equation // AIP Conference Proceedings. – 2021. – Vol. 2325. – Art. 020059. – DOI: 10.1063/5.0040505.

4. Crank J., Nicolson P. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type // Advances in Computational Mathematics. – 1996. – Vol. 6, No. 1. – P. 207–226. – DOI: 10.1007/BF02127704. (Reprinted from Proceedings of the Cambridge Philosophical Society, 1947, Vol. 43, No. 1, Pp. 50-67)

5. Chen F. F. Introduction to Plasma Physics and Controlled Fusion. – 3rd ed. – Cham: Springer International Publishing, 2016. – 490 p. – DOI: 10.1007/978-3-319-22309-4.

6. Utebaev D., Atadjanov Kh. L., Nurullaev J. A. Finite element method schemes of higher accuracy for solving non-stationary fourth-order equations // Bulletin of KazNU. Mathematics, Mechanics and Computer Science. – 2023. – Vol. 118, No. 2. – P. 42–56.

7. Steinkamp V. Python for Engineering and Scientific Computing. – Bonn: Rheinwerk Publishing, 2024. – 511 p.

8. Moskalkov M. N., Utebaev D. Comparison of some methods for solving the internal wave propagation problem in a weakly stratified fluid // Mathematical Models and Computer Simulations. – 2011. – Vol. 3, No. 2. – P. 264–271. – DOI: 10.1134/S207004821102010X.

9. Габов С. А. Новые задачи математической теории волн. – М.: Наука, 1998. – 448 с.

10. Wu P., Xie Y., Jiang H., Niu L., Natsuki T. Computationally efficient complex envelope approximate Crank–Nicolson scheme and its open region problem for anisotropic gyrotropic plasma // Physics of Plasmas. – 2020. – Vol. 27, No. 10. – Art. 102109. – DOI: 10.1063/5.0020205.

11. Wu P., Xie Y., Jiang H., HanYi D., Natsuki T. Bandpass approximate Crank–Nicolson

implementation for anisotropic gyrotropic plasma open region simulation // Optik. – 2021. – Vol. 242. – Art. 166941. – DOI: 10.1016/j.ijleo.2021.166941.
12. Python Software Foundation. Python Documentation. URL: https://docs.python.org
13. NASA. Surface Meteorology and Solar Energy Data. URL: https://power.larc.nasa.gov/

PYTHON БАҒДАРЛАМАЛАУ ОРТАСЫНДА ЕКІТЕМПЕРАТУРАЛЫ ПЛАЗМА ТЕҢДЕУІН САНДЫҚ МОДЕЛЬДЕУ

НУРУЛЛАЕВ Ж.А.*, БАВЕТДИНОВ А.А.

***Нуруллаев Жүсіпбай Абдиуалиевич** – PhD, Бердақ атындағы Қарақалпақ мемлекеттік университеті, Нукус қ., Өзбекстан
E-mail: njusipbay@mail.ru, https://orcid.org/0000-0003-0935-5388
Баветдинов Асқар Ахметұлы – Магистрант, Бердақ атындағы Қарақалпақ мемлекеттік университеті, Нукус қ., Өзбекстан
E-mail: bawetdinovasqar2003@gmail.com, https://orcid.org/0009-0000-6841-9041

Аңдатпа. Бұл жұмыста Python бағдарламалау ортасын қолдана отырып, екітемпературалы плазма жүйесін сандық модельдеу және симуляциялауы қарастырылады. Модель электрондар мен иондар арасындағы энергия алмасуын сипаттайтын бейсызық жылуөткізгіштік теңдеулерінің байланысқан жүйесіне негізделген. Сандық шешімдерді алу үшін уақыт пен кеңістік бойынша екінші ретті дәлдікті қамтамасыз ететін Кранк–Николсонның айқын емес ақырлы-айырымдық схемасы қолданылды. Есептеу процесі NumPy, SciPy және SymPy кітапханалары арқылы жүзеге асырылды, ал нәтижелерді визуализациялау мен пайдаланушымен өзара әрекеттесу Mathplotlib және Tkinter құралдарының көмегімен орындалды. Әзірленген бағдарлама параметрлерді интерактивті түрде енгізуге, температуралардың эволюциясын есептеуге және нәтижелерді нақты уақыт режимінде бейнелеуге мүмкіндік береді. Сандық эксперименттер Қарақалпақстан Республикасының табиғи және климаттық жағдайларына сәйкес келетін параметрлер үшін жүргізілді, мұнда күн энергиясының ағыны 1000–2800 Вт/м² аралығында өзгереді. Алынған нәтижелер электрондар мен иондар температураларының уақыттық және кеңістіктік динамикасын көрсетіп, ұсынылған Кранк–Николсон әдісінің тиімділігін және төмен қысымды плазма орталарындағы энергия алмасу процестерін модельдеуде әзірленген бағдарламалық құралдың сенімділігін дәлелдейді.
Түйін сөздер: екітемпературалы плазма, Кранк–Николсон схемасы, сандық модельдеу, жылуөткізгіштік теңдеулері, Python бағдарламалау, электрон–ион энергия алмасуы, жылулық диффузия, есептеуіш имитация.

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ УРАВНЕНИЯ ДВУХТЕМПЕРАТУРНОЙ ПЛАЗМЫ В СРЕДЕ ПРОГРАММИРОВАНИЯ PYTHON

НУРУЛЛАЕВ Ж.А.*, БАВЕТДИНОВ А.А.

***Нуруллаев Жусипбай Абдиуалиевич** – PhD, Каракалпакский государственный университет имени Бердаха, г. Нукус, Узбекистан
E-mail: njusipbay@mail.ru, https://orcid.org/0000-0003-0935-5388
Баветдинов Аскар Ахметулы – Магистрант, Каракалпакский государственный университет имени Бердаха, г. Нукус, Узбекистан
E-mail: bawetdinovasqar2003@gmail.com, https://orcid.org/0009-0000-6841-9041

Аннотация. В данном исследовании представлено численное моделирование и моделирование системы двухтемпературной плазмы с использованием среды программирования Python. В данной работе представлено численное моделирование и имитация двухтемпературной плазменной системы с использованием среды программирования Python. Модель основана на связанной системе нелинейных уравнений теплопроводности, описывающих процессы энергообмена между электронными и ионными компонентами. Для получения численных решений была реализована неявная конечно-разностная схема Кранка–Николсона, обеспечивающая второй порядок точности по времени и пространству. Вычислительный процесс осуществлялся с применением библиотек NumPy, SciPy и SymPy, а визуализация результатов и взаимодействие с пользователем реализованы с использованием инструментов Mathplotlib и Tkinter. Разработанная программа позволяет интерактивно вводить параметры,

вычислять эволюцию температур и визуализировать результаты в реальном времени. Численные эксперименты были проведены для значений параметров, соответствующих природно-климатическим условиям Республики Каракалпакстан, где плотность солнечного энергетического потока изменяется в диапазоне 1000–2800 Вт/м². Полученные результаты демонстрируют временную и пространственную динамику температур электронов и ионов, подтверждая эффективность предложенного подхода Кранка–Николсона и надёжность разработанного программного обеспечения для моделирования процессов переноса энергии в условиях низкодавленной плазмы.

Ключевые слова: двухтемпературная плазма, схема Кранка–Николсона, численное моделирование, уравнения теплопроводности, программирование на Python, энергообмен электрон–ион, тепловая диффузия, вычислительная имитация.