https://doi.org/10.15407/csc.2024.04.068
UDC 519.85

Yu.H. TARASICH, PhD (Eng.), junior research scientist,
V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine,
Glushkov ave., 40, Kyiv, Ukraine, 03187,
ORCID: https://orcid.org/0000-0002-6201-4569,
yutarasich@gmail.com

ALGEBRAIC MODELLING OF EXPERIMENTS ON THE EXAMPLE OF PROTON THERAPY

Despite the rapid development of the chemical industry and science, discoveries in the field of health care, the emergence of drugs and therapeutics based on nanotechnology and the development of radiation therapy technologies, the safety of biomedical applications of the latest products, and the search for new methods and approaches to the diagnosis and treatment of cancer are an open issue. One of the safest and fastest methods for researching the behaviour of new materials and tools and selecting the best candidates is the modelling of relevant processes, particularly computer molecular modelling based on mathematical models. However, despite a large number of available methods and modelling tools, for most of them, the successful application is possible only for a narrow range of tasks and experiments.

As one of the possible solutions to this problem, we propose a new approach to computer molecular modelling based on the synergy of the algebraic approach, namely, algebraic modelling and biological knowledge at different levels of abstraction, starting from quantum interactions to interactions of biological systems.

We see one of the directions of application of this approach in the possibilities of modelling the radiation therapy process — starting from modelling the accelerators' work and ending with modelling the interaction of the particles' beam with the matter at the level of quantum in-teractions. In particular, in the article, we consider the possibilities of forward (specific and symbolic) and backward (symbolic) algebraic modelling on the example of models of the higher level of abstraction, which allows us to visualize certain interactions and to build charts of dependencies for specific models, and to determine the presence of the desired scenarios (forward modelling) or a set of initial environment parameters (backward modelling) in symbolic form.

Keywords: *Molecular Modelling, Algebraic Modelling, Modelling of Biological Experiments, Proton Therapy Modelling, Theory of Agents and Environments Interaction, Symbolic Modelling.*

Concept of the Research

The purpose of our study is to conduct modelling of experiments aimed at studying physical, chemical or biological processes.

We consider experiment modelling related to the study of the properties of a certain process and the determination of its final results in terms of the experimental environment. We also consider backward modelling when the final results (properties of a physical entity, which is a substance or process) are specified and when finding the initial parameters and the corresponding actions leading to these properties is necessary. For example, it

Cite: Tarasich Yu.H. Algebraic Modelling of Experiments on the Example of Proton Therapy. *Control Systems and Computers*, 2024, 4, 68—84. https://doi.org/10.15407/csc.2024.04.068

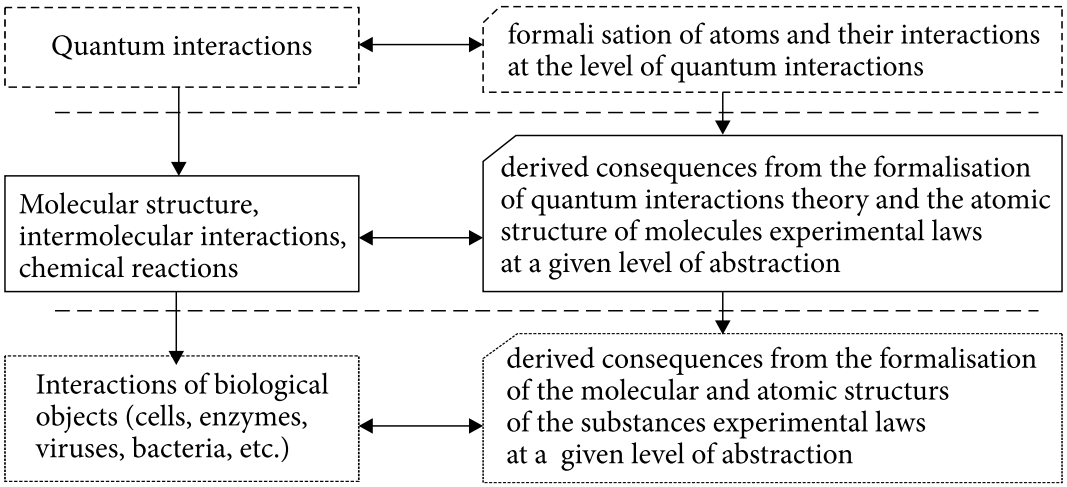


Fig. 1. Hierarchy of the researched processes and the relevant knowledge required for modelling

can be the task of modelling the radiation therapy process, when the energy of the proton beam, the length of its trajectory and the necessary characteristics of the irradiated substance are specified. In this case, modelling the processes of the interacting protons with matter, the depth at which the Bragg peak will be reached and other factors is possible. Backward modelling, in turn, will enable the identification of the required values of the beam energy, the angle and/or duration of irradiation to reach the Bragg peak at a given depth.

The implementation of this task requires formalisation or the formal presentation of the subject area knowledge within which the experiment is conducted. Knowledge formalisation can be carried out at any level of abstraction — at the level of the atomic structure of substances and quantum–mechanical interactions, at the level of the molecular structure of substances, taking into account their intermolecular interactions, at the level of interacting substances and at the level of biological objects. The appropriate level of abstraction should be chosen according to the purpose of the experiment and in consideration of the impact on the results from the lowest level of abstraction.

Consider the following hierarchy of researched processes and the relevant knowledge required for modelling (Fig. 1):

Knowledge of quantum interactions includes the basic properties and processes of the interactions of elementary particles, particularly the behaviour of electrons during various processes in the atomic environment (e.g. energy change and transitions between atomic orbitals) and the formation of interatomic bonds.

Quantum interactions are the basis of the mechanisms of occurrence and the properties of the force of interaction of the electromagnetic nature between molecules. This allows us to consider the formation and decomposition processes of molecules under different values of environmental parameters, such as temperature, pressure and presence of catalysts. The highest level of abstraction is the interactions of biological objects at the level of a subject area. For example, in biology, it can be the interaction between a virus and a cell or the interactions of enzymes within the cellular environment. In physics, it is the interactions of different bodies or substances.

The application of certain knowledge in modelling depends on the purpose of the modelling. For example, modelling the interactions in mechanics does not necessarily require knowledge of quantum theory. Accordingly, using laws that are not derived from theories of a lower level of abstraction but are established and confirmed experimen-

tally is possible. On the other hand, phenomena in biology and chemistry can be explained using the lowest level of knowledge, which involves such branches of science as quantum biology and quantum chemistry.

Taking into account the subject areas of the experiments and the tasks set before the modelling, we consider the need to use forward and backward algebraic modelling:

1) Study of Process Properties (Forward Modelling)

The input data are a set of so-called agents, which are experimental subjects, and their attributes. The agent can be both an elementary particle (e.g. electron, proton and photon) and a complex biological object. The values of the agents' attributes determine the initial states of all agents. Any quantitative characteristic can be defined as an attribute of an agent, e.g. the number of electrons in an orbital, particle charge, molar mass and substance amount. All agents interact in an environment. Accordingly, determining and setting the values of the attributes of the environment, such as temperature and pressure, are possible. The environment can also be an agent that interacts with similar ones in a higher-level environment. Attribute values can be specified not only by specific values but also by possible ranges of values or more complex formulas.

We have formalised knowledge at a given level of abstraction that determines agents' behaviours in the environment. Behaviour represents a tree of possible actions of an agent in the environment; i.e. it interprets all possible interactions. Thus, we have two components to define interactions. The first is an agent's atomic action that changes the environment consisting of all agent attributes. The second is a possible sequence of actions defined by a formal concept of behaviour.

In this case, the task of modelling is to determine the reachability of certain properties of the studied process presented in a formal form.

In contrast to simulation modelling and probabilistic methods, ***algebraic modelling*** has the feature of possibly considering multiple scenarios of system behaviour rather than one specific scenario only.

Both specific and symbolic algebraic modelling can be performed.

In the first case, the value of the agent's initial attributes is explicitly specified. Next, we analyse property reachability using specific values.

The following is an example. '*The reaction temperature is 10. This reaction is a dissociation reaction. The amount of the substance is 15 mol*'. This will be written as (T == *10* && *ReactionType* == *Dissociation* && *SubstanceAmount* == *15*).

We can build and analyse charts (e.g. a change in the concentration of a substance during a reaction depending on a change in temperature or a change in the energy of a proton beam in the substance). However, this experiment will take place within the framework of ***One*** scenario.

At each step of the simulation, we also receive the specific numerical values of the attributes according to the calculations.

Having built a symbolic algebraic model, we can assign arbitrary initial values to the agent's attributes, such as the following: '*The temperature of the first substance is 20–60. The temperature of the second substance is 30–70. The temperature of the second substance should be higher than that of the first*'.

This can be written as follows:

$$(T1 >= 20) \,\&\, (T1 <= 60)$$
$$\&\, (T2 >= 30) \,\&\, (T2 <= 70) \,\&\, (T2 > T1).$$

These initial formulas or algebraic constraints can be arbitrarily complex.

In this case, the initial formula of the initial state of the experiment determines a set of possible scenarios. Therefore, at each step of algebraic modelling, we will not receive specific numerical values of attributes but a formula covering a ***Set of Scenarios***. The final result will not be one scenario that achieves the desired property but all scenarios from the initial formula in which this property is reachable.

2) Derivation of an agent with Given Properties (Backward Modelling)

Algebraic modelling also makes it possible to model an experiment in which the initial data are the properties of the process or agent. The task is to determine the initial state in which the synthesis of a given agent or process is possible.

Backward modelling occurs as modelling from the given properties to a possible set of initial at-

tributes according to formal knowledge that defines all possible interactions at a given level of abstraction.

An example is the derivation of a substance with certain properties or the identification of the necessary initial parameters of proton therapy (e.g. beam energy and the angle and duration of irradiation).

Next, we consider the application of algebraic modelling to the modelling of radiation therapy processes. In our opinion, the study of heavy particle therapy (protons/alpha particles) — i.e. the study and modelling of the necessary parameters that will allow the localisation of the Bragg peak in the tumour and, accordingly, the identification of the parameters for which radiation therapy will have a minor effect on healthy cells — deserves special attention. For this, we have to find the initial parameters at which the property determining the degree of this damage is achievable. For example, it can be the maximum number of affected healthy cells and the localisation of the Bragg peak in the tumour.

State of the Art

Despite the rapid developments in science and the chemical industry, discoveries in the field of health care and the emergence of drugs and therapeutics based on nanotechnology and radiation therapy technologies, the safety of the biomedical applications of the latest products and the search for new methods and approaches for the diagnosis and treatment of cancer remain contentious issues. Radiation therapy is the primary or additional treatment method for 75%–85% of all cancer patients and is one of the most effective methods in inoperable cases.

Indeed, the study of the effectiveness of existing radiation methods and tools, which would ensure the delivery of the optimal radiation dose to the pathological focus with minimal damage to normal tissues, remains an open issue despite the achievements in the field of radiation therapy. The main tasks that require the involvement of additional research methods and tools are as follows:

- the possibility of modelling different scenarios of dose delivery at different organ positions;
- determining the accuracy of targeting and the optimal distribution of the intensity of the radiation beam for each irradiation zone;
- selection of the optimal fractionation scheme and determination of the optimal total duration of the therapy course for various tumours;
- modelling of the operation of particle accelerators and dose delivery systems.

We think that one of the safest and fastest methods for researching the behaviour of new materials and tools and selecting the best candidates is the modelling of relevant processes, particularly computer molecular modelling based on mathematical models.

Hybrid models and methods for systems biology and medicine (including working with formal hybrid specifications, such as temporal and hybrid automata) and the combination of models by integrating combinatorial and continuous constraints and using machine learning to design models and define their parameters [1–2] are essential steps in solving open problems in the field of modelling and in the research of relevant processes and systems. Mathematical models have been successfully used to study oxygen transport, tumour angiogenesis and various cancer treatment methods [3–5]. However, despite the increasing availability of existing protein and nucleic acid data, as well as modelling methods and tools [6–14], unfortunately, there are a number of disadvantages, such as low accuracy, limitations by borders of biological experiments, the need for a responsible selection of research methods and tools, and errors in the structures of the molecules with which the software works, that can be critical for conducting experiments. Accordingly, developing and using a wide range of combined methods and tools for modelling and computing large molecular systems remains an open issue.

As a possible solution, we propose using an algebraic approach and the corresponding formal methods, which have proven their effectiveness in many other areas. The main idea of our research is to apply the technology of algebraic modelling and quantum chemical apparatus for modelling and verifying organic chemistry problems, particularly the modelling and verification of different ap-

proaches to the question of studying the effects of radiation therapy on cancer treatment.

At this research stage, we have developed a methodology for the formalisation of complex organic and inorganic substances, as well as chemical processes and reactions, which is based on the formalisation of the interactions between atoms and molecules at the level of quantum interactions. Modelling the substances and their interactions at the level of their atomic structures provides a mechanistic understanding of their behaviours; the use of formal algebraic methods allows proving properties and finding relevant scenarios for the effective analysis of the behaviours of various objects in real time, considering not individual scenarios but sets of possible behaviours. In particular, this approach makes it possible to consider the entire process of radiation therapy, starting from the operation of the accelerator and ending with the study of the interaction of a beam of accelerated particles with matter, thus simulating and testing the feasibility of various scenarios of particle impact on tumours and healthy cells.

Theoretical Background

Agents and Environments. The proposed algebraic approach is implemented in the system of insertion modelling, which is based on the theory of agents and environments launched by Ukrainian academician O.A. Letichevsky and British scientist D. Gilbert [15].

The basic idea involves agent interactions in a certain environment. The environment may also be an agent that interacts with similar agents in a higher-level environment and so on. Each agent has its own type, which is determined by the attributes of the agent. Each attribute is typed and belongs to a certain theory in which predicates and operations are defined. Thus, we can define the arithmetic, symbolic, bit and byte attributes. Because of the presence of a large number of different types of attributes in different theories, defining the problem of formula executability in the chosen theory is important, i.e. solving the problem of finding attribute values in the formula with which the formula is true. This problem is solved using so-called solver systems and systems for the automatic proof of theorems, which are the bases of the algebraic approach in modelling.

Behaviour Algebra. The interactions of agents in the environment are determined by behaviour. Behaviour represents a tree of possible actions of the agent in the environment. Behavioural expression is determined with a formula built from actions and behaviours using behavioural algebra operations:

- The prefixing operation '.' determines that some action x precedes behaviour B. It is written as $x.B$.
- The non-deterministic choice '+' defines alternative behaviours. It is written as $A + B$.
- Algebra is also extended by the sequential 'A; B' and by the parallel '$A \parallel B$' composition of behaviours [15].

The semantics of the action of each agent is defined as a triple $B = <P, A, Q>$, where P is a pre-condition, presented in the form of a formula in a certain theory, Q is a post-condition, and A is a process that visualises the transition of the agent between states. In other words, if the pre-condition is executable, the agent's state will change according to the post-condition. The pre-condition is determined by the predicate, which is a Boolean expression over the formulas of the corresponding theories, such as equality or inequality in linear arithmetic. The post-condition changes the environment and also uses predicates, assignment operators and operations in the chosen theory.

A behavioural equation is an equality in which the name of the behaviour is on the right side, and the expression in the behaviour algebra over actions and other behaviours is on the left side. With a behavioural equation, we can describe some processes of agents' interactions in an environment. This can depict processes and phenomena at the level of physical or chemical terms. Finding a scenario in the form of a sequence of actions leading to the desired property is possible by solving the behavioural equation. The solution is derived with the help of algebraic modelling.

To find the solution, the technique of unfolding equations is applied, which is used in the technique of unfolding rewriting rules [16].

Data on substances (protein sequences, structures, interactions, functions and experimental

data) can be extracted from well-known worldwide databases (UniProt, GenBank, RefSeq (information on protein sequences), Protein Data Bank (protein structures), DIP, BioGRID (protein interactions), InterPro, Pfam and Gene Ontology (GO) (functional notations), The Human Protein Atlas and Genotype-Tissue Expression (GTEx) (protein expression levels in different tissues, organs and cell types), etc.), or received from biologists, chemists, medics. The received data are processed and rewrited in algebraic form using special translators or entered manually.

For modelling hybrid systems, the system of insertion modelling is expanded with the possibility of analytically solving differential equations, whose operators are the executing algebraic specifications.

Formalisation of Quantum and Molecular Interactions in Terms of Behaviour Algebra

We consider the model of the substance as a multi-agent environment in which agents interact at different levels of abstraction. We are conducting the formalisation of the substance model in the insertion modelling system to examine and analyse the specified properties using the algebraic modelling of this multi-level system.

Agents that interact in the substance as an environment at the highest level are characterised by a set of attributes that we define in the process of formalisation. Each agent performs a particular set of actions under certain conditions, which form its behaviour. We will compile behavioural equations for interacting agents based on arbitrary initial attribute values.

Thus, the structure of the substance is given by a set of agents of the MOLECULE type, which, in turn, contain a function representing the set of atoms that make up the given molecules. The properties of the atoms are determined by the structure of their nuclei and the number and organisation of orbital electrons. Accordingly, the main attributes that characterise substances are the structures of their molecular orbitals and the nuclear models of atoms that are part of the substance, which will allow us to consider all the processes of their interactions at the level of quantum interactions. In addition, the mass of the substance, the amount of the substance in moles and the concentration per unit volume, among other factors, are specified (these parameters can be set as the initial values of the experiment or be determined during modelling).

We define the ATOM agent type as the lowest-level agent.

The attributes of this type of agent will be represented by the numerical values of the following quantities: quantum numbers (principal quantum number (*principalQuantumNum*), the spin quantum number (*spinQuantumNum*)), electronegativity (*electronegativity*) and valence (*bondingAbility*), charge (*charge*), relative atomic mass (*mass*), electronic configuration of the atom (*orbital*) and so on.

Thus, we define the *elements:(int)->ATOM* function corresponding to 118 chemical elements, which will be used to set and store the attribute values of the agents of the created type following the information provided in the periodic table and the electronegativity tables of chemical elements. Accordingly, we obtain a database of elements that will be used to model higher-level agents (molecules and substances) and the interactions between them.

For example, let us consider a fragment of the formula for saving information about existing elements:

elements(1).name == H && elements(1).principalQuantumNum == 1 &&
elements(1).electronegativity == 2.2 &&
elements(1).orbital(1,0,1) == 1 &&
elements(1).spinQuantumNum (1,0,1,1) == 0.5 &&
elements(1).protonsNum == 1 && elements(1).massNumber == 2 &&
elements(1).neutronsNum == elements(1).massNumber - elements(1).protonsNum &&
elements(1).electronsNum == 1 &&
*elements(1).mass == elements(1).protonsNum * protonMass +*
*(elements(1).massNumber — elements(1).protonsNum) * neutronMass + elements(1).electronsNum*electronMass &&*
elements(1).charge == elements(1).protonsNum * protonCharge +
*elements(1).electronsNum * electronCharge &&...*
elements(3).name == Li && elements(3).principalQuantumNum == 2 &&

elements(3).electronegativity == 0,95 &&
elements(3).orbital(1,0,1) == 2 && elements(3).orbital (2,0,1) == 1 &&
elements(3).spinQuantumNum(1,0,1,1) == 0.5 && elements(3).spinQuantumNum(1,0,1,2) == 0.5 && elements(3).spinQuantumNum(2,0,1,1) == 0.5 &&
elements(3).protonsNum == 3 && elements(3).massNumber == 7 &&
elements(3).neutronsNum == elements(3).massNumber - elements(3).protonsNum &&
elements(3).electronsNum == 1 &&
*elements(3).mass == elements(3).protonsNum * protonMass +*
*(elements(3).massNumber — elements(3).protonsNum) * neutronMass + elements(3).electronsNum*electronMass &&*
*elements(3).charge == elements(3).protonsNum * protonCharge +*
*elements(3).electronsNum * electronCharge && …*

The attributes of the MOLECULE type of agent represent the numerical values of the following: the set of atoms that comprise it (*atomsID*), the electronic configuration of the molecule (*molOrbital*), bond length (*bondLength*), bond energy (*bondEnergy*), dipole moment (*dipoleMoment*), molar mass (*molarMass*), bond order (*bondMO* — by the method of molecular orbitals, *bondV* — by the method of valence bonds), bond type (*bondType*), to name a few, and so on:

MOLECULE:obj(molAtomsNum:(int), atoms Names:(int)->PERIODIC_ELEMENTS, atomsID: (int)->int, bondLength:(int, int)->real, bondEnergy: (int, int)->real, bondMO:(int, int)->real, bondV: (int, int)->real, bondType: (int, int)->BOND_TYPE, molOrbital:(int,int,int,int)->int, dipoleMoment: real, molarMass:real, relativeMolecularMass:real, radius:real, ….),

The electronic configuration of a molecule is determined by the functional attribute *molOrbital: (int, int, int, int) -> int*.

Accordingly, for each orbital, we will have the formula *molOrbital:(a1, a2, i, j) = k*,
where:

• *a1* and *a2* are the numbers of the first and second atoms, respectively,

• *i* is the type of orbital (bonding orbitals instead of σ, π, etc., denoted as −1, −2, −3 and so on. For anti-bonding orbitals, we define the notation 1, 2, 3),

• *j* is the number of the "cell" with electrons in the corresponding orbitals (equal to 1, 2, etc.),

• *k* is the number of electrons in the corresponding "cell".

The structure and characteristics of the molecules and substances are considered both from the perspectives of the method of valence bonds and the method of molecular orbitals as two complementary methods.

The agent of the SUBSTANCE type represents the substance:

SUBSTANCE:obj(moleculesID:(int)->int, moleculesNum:int, nu:real, concentration:real,
mass:real, volume:real, dencity:real,…)

Thus, we consider a substance that consists of *n* molecules (*moleculesID:(int)->int*). The initial values necessary for modelling, such as the formula/structure of the molecule/substance, can be set manually or obtained from the .mol file using a special parser.

For example, we can obtain the following representation of acetic acid:

elNum == 118 && molNum == 1 && subNum == 1 &&
substances(1).moleculesID == 1 &&
molecules(1).atomsNum == 8 &&
molecules(1).atomsNames(1) == C && molecules(1).atoms Names(2) == C &&
molecules(1).atomsNames(3) == O && molecules(1).atoms Names(4) == O &&
molecules(1).atomsNames(5) == H && molecules(1).atoms Names(6) == H &&
molecules(1).atomsNames(7) == H && molecules(1).atoms Names(8) == H &&
molecules(1).bondV(1,2) == 1 && molecules(1).bondV(1,5) == 1 &&
molecules(1).bondV(1,6) == 1 && molecules(1).bondV(1,7) == 1 &&
molecules(1).bondV(2,3) == 1 && molecules(1).bondV(2,4) == 2 &&
molecules(1).bondV(3,8) == 1 && …

In accordance with each atom, which is part of the molecule, we put the corresponding chemical element, thus assigning the attribute values of this element to the corresponding atom, which will be formalised using the following action:

setAtomsData *= (Forall (i:int) (1<=i<=elNum && molecules(molNum).atomsNames(atomsInMolNum) == elements(i).name)->*

"MOLECULE # M1: action 'setAtomsData';"
(atoms(molecules(molNum).atomsID(atomsInMolNum)).name = elements(i).name;
atoms(molecules(molNum).atomsID(atomsInMolNum)).principalQuantumNum = elements(i).principalQuantumNum;
atoms(molecules(molNum).atomsID(atomsInMolNum)).electronegativity = elements(i). electronegativity;…),
elNum – The number of chemical elements is an attribute of the environment.

By having appropriate initial values and formalising the rules and laws of quantum mechanics, chemistry, physics and biochemistry, it is possible to determine the algebraic representation of the electronic configuration of a substance/molecule, identify the properties of compounds and model the possibility of specific reactions.

Algebraic Modelling

Setting specific initial values for the agents and environment attributes, we can consider a specific scenario of the given property reachability (forward-specific algebraic modelling). This modelling method allows us to check the correctness of the knowledge formalisation, obtain graphs of changes in the values of the agents' attributes depending on the specific parameters of the environment and find the specific values of the attributes of the agents or the environment at the end of the experiment.

For example, we need to find the products of the NaOH dissociation reaction and determine the value of the reactant's concentration at time t1. We also know the values of the reaction rate coefficient, the volume of the solution and the mass of the substance. The formula for the initial state of the environment for a specific model will take the following form:

reactions(1).reagentsNum == 1 && reactions(1).reagentsID(1) == 1 &&
reactions(1).solutionVolume(1) == 10 && reactions (1).rateCoeff == 1 &&
reactions(1).reactTime == 10 &&
substances(1).moleculesID == 1 && substances(1). mass == 35 &&
substances(1).nu == 0 && substances(1).concentration == 0 &&
molecules(1).atomsNum == 3 && molecules(1).formula == undefined &&
molecules(1).atomsNames(1) == H && molecules(1). atomsNames(2) == O &&
molecules(1).atomsNames(3) == Na &&
molecules(1).bondV(1,2) == 1 && molecules(1).bond V(2,3) == 1 && …

The result of the modelling will be one specific scenario (Fig. 2, *a*).

The advantage of symbolic algebraic modelling is the ability to assign a set of possible values to agent attributes. For the problem described above, the formula for the initial state of the environment can take the following form:

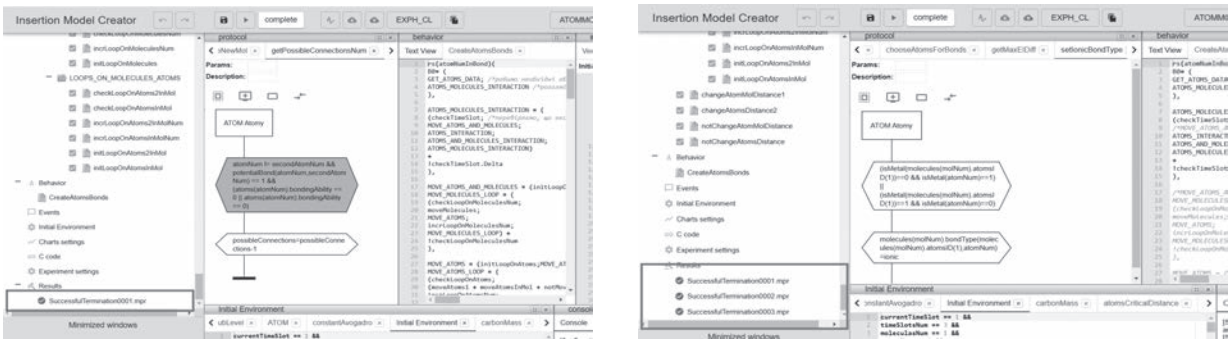
reactions(1).reagentsNum == 1 && reactions(1).reagentsID(1) == 1 &&
1<=reactions(1).rateCoeff && reactions(1).rateCoeff <= 2 &&
reactions(1).reagents(1).concentration == 0 && reactions(1).solutionVolume == 10 &&
***5<=reactions(1).reactTime && reactions(1).reactTime <= 10** &&*
substances(1).moleculesID == 1 && substances(1). mass == 35 &&
substances(1).nu == 0 && substances(1).concentration == 0 &&
molecules(1).atomsNum == 3 && molecules(1).formula == undefined &&
molecules(1).atomsNames(1) == H && molecules(1). atomsNames(2) == O &&
molecules(1).atomsNames(3) == Na &&
molecules(1).bondV(1,2) == 1 && molecules(1).bond V(2,3) == 1 && …

After conducting forward symbolic algebraic modelling, we will obtain a set of possible scenarios. These, for example, will allow us to determine the reach of the end of the reaction process for given sets of values of the rate constant and reaction time, or to obtain possible sets of values of the reaction products' amounts received in a given time interval and so on (Fig. 2,*b*).

The behavioural equation that describes the occurrence of the dissociation reaction of a substance with the formula of EOH will have the following form:

REACTION_EOH_DISSOCIATION = (
(setAtomsData); (getAmountMolM); (getAmount Substance);
(getReagentConcentration);
(CONCENTRATION_TO_TIME); (EOH_DISSOCIATION)),

where setAtomsData, getAmountMolM, getAmountSubstance and getReagentConcentration are the actions that describe the chemical formulas for



a *b*

Fig 2. Results of forward algebraic modelling: *a* — specific algebraic modelling; *b* — symbolic algebraic modelling

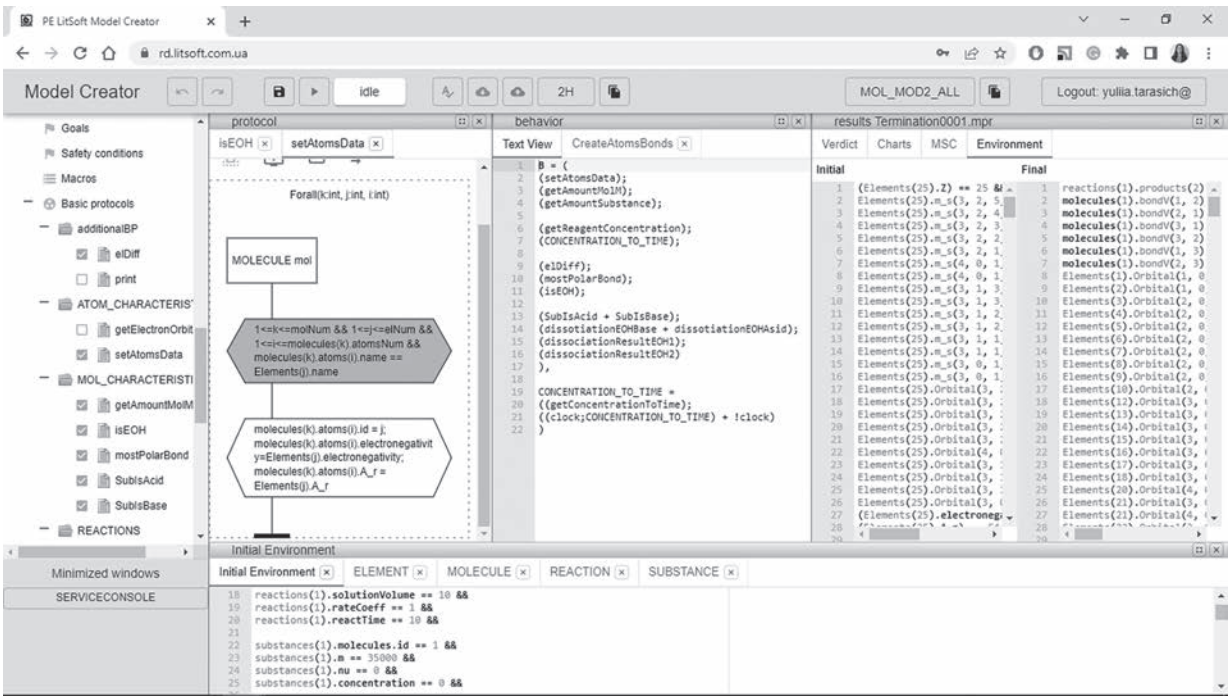


Fig 3. Model Creator Tool

finding the structure of a molecule, molar mass, amount of a substance and its concentration in a solution; CONCENTRATION_TO_TIME and EOH_DISSOCIATION represent behaviours that describe the process of changing the concentration of a substance during a reaction and determining reaction products. The corresponding behavioural equations are as follows:

EOH_DISSOCIATION = ((isEOH); (SubIsAcid + SubIs Base);

(dissotiationEOHBase + dissotiationEOHAsid); (dissociationResultEOH1); (dissociationResultEOH2)),

CONCENTRATION_TO_TIME = ((getConcentration ToTime);

((clock;CONCENTRATION_TO_TIME) + !clock)))

Accordingly, it is easy to calculate the necessary values for modelling reactions, such as molecular mass, concentration and density, by knowing the structure of the substance (setAtomsData action). For example, the calculation of the molecular mass of a molecule is formalised by the following action:

GetAmountMolM *= (Forall (i:int) (1<=i<=molecules (molNum).atomsNum)->*
" MOLECULE # M1: action 'Get_Molecule_Mass';"
(molecules(molNum).mass =
molecules(molNum).mass + atoms(molecules(molNum).atomsID(i)).mass))

That is, by looking through the set of MOLECULE-type agents in the pre-condition and, accordingly, the subset of each molecule's atoms, we find its molecular mass as the sum of the corresponding atomic masses.

We use the Model Creator tool for modelling and model verification (Fig. 3). It uses symbolic modelling techniques, including algebraic and deductive formal methods, to solve complex problems.

The Model Creator includes a number of systems and libraries for implementing algebraic formal methods and integrating with other software systems.

The key features of the platform are the testing technology, model-based development, support for the development process of a critical system or quality of the service system, verification and validation, and cybersecurity [17].

Model Creator works based on the insertion modelling system IMS, developed on the basis of the algebraic programming system APS [18, 19]. Insertion modelling focuses on building models and studying the interactions of agents and environments in complex multi-agent systems [15].

As one research direction is the algebraic modelling of radiation therapy processes to study their effects on cancer treatment, we add the additional attributes (radius (r) to determine the beam passage area, density (ro), mass stopping power (S) and the length of segment (l), which the protons must pass) to the agent of SUBSTANCE type to determine the value of the physically absorbed dose.

In more detail, examples of the first steps in the formalisation of some radiation therapy processes and the operation of particle accelerators, as well as the possibilities and advantages of using algebraic modelling, particularly backward algebraic modelling, are discussed in the next section of this article.

Algebraic Modelling of Proton Therapy Processes

The modelling of radiation therapy processes includes the modelling of the operation of particle accelerators and the modelling of the irradiation process. Accordingly, we divide this experiment into two stages.

For the first stage, a modelling of the operation of the synchrotron is chosen. We select the synchrotron because it has the smallest loss of protons, in contrast to the cyclotron and the synchrocyclotron. Therefore, it has a much smaller effect on increasing the radiation background.

A proton synchrotron is a cyclic resonant proton accelerator with a constant orbital radius and a magnetic field that increases with time according to the decreasing frequency of the accelerating electric field (R = const, $B(t)\uparrow$, $\omega(t)\uparrow$). At the same time, the values of the magnetic field and the frequency of the accelerating electric field change in strict accordance with each other while ensuring the constancy of the radius. In modern accelerators with a large radius, the beam itself affects the change in frequency of the accelerating electric field (with the help of special signal electrodes — beam position sensors).

The task in this stage then is to model the maintenance of the growth of particle energy and the constancy of the orbit radius through the corresponding growth of the magnetic field and the frequency of the accelerating fields, as well as the achievement by the particles of the energy index necessary for irradiation.

The study and formalisation of the physics of interaction of a proton beam with a substance comprise the second stage of research/modelling.

In our opinion, research on heavy particle therapy (protons/alpha particles), namely, research and modelling of the necessary parameters that will allow the localisation of the Bragg peak in the tumour, deserves special attention. Thus, for this experiment, we focused on modelling the irradiation of matter with a proton beam.

In this stage, we start working with agents of the SUBSTANCE type. In this case, the agent of SUBSTANCE type is an environment for the agent of PARTICLE type. The main task in this stage is to determine the characteristics of the proton beam and the systems for delivering the dose to the patient. We consider the need for both forward and backward symbolic algebraic modelling as comp-

lementaring methods that will allow us to formalise certain properties of the researched process and determine their reachability for a different set of scenarios. They will also help us determine the necessary sets of initial attribute values that will allow us to reach a given property (e.g. localisation of the proton beam in the tumour).

Algebraic Modelling of Synchrotron Operation

The synchrotron was defined as the initial environment into which the PARTICLE agents (protons or ions) are immersed. A new type, *particle_type {ion, proton}*, was created to define the particle type.

The following attributes are defined for the PARTICLE agent type: particle type (*particlesType*), mass (*initialMass* and *mass*), charge (*charge*), momentum (*momentum*), velocity (*velocity*), radius of motion (*radiusOfMovement*), energy (*initialEnergy* and *energy*), particle frequency (*frequency*). Accordingly, the PARTICLE agent type will be formalised as follows:

PARTICLE:obj(particlesType:particle_type, initial Mass:real, mass:real, charge:real, momentum:real, velocity:real, initialEnergy:real, energy:real, radiusOfMovement:real, frequency:real, particleID:(int) ->int)

The type *particlesType={ion, proton}* was created to define the type of particle and enable further modelling, analysis and comparison of different kinds of radiation therapies.

The main attributes of the environment are the induction of the magnetic field (*magneticFieldInduction*), the frequency of the accelerating field (*acceleratingFieldFrequency*), the operating time of the accelerator (*time*), the radius (*radius*) and the indicator of the energy that the particle must acquire before exiting the synchrotron (*energyNeeded*). The induction of the magnetic field, the frequency of the accelerating field and time can be specified not by specific values but by possible intervals of values.

To simulate the operation of the accelerator at a higher level, we have the following behavioural equation:

SYNCHROTRON_WORK *= ((addToSynchrotron + !addToSynchrotron) ; inSynchrotron ; ACCELERATION);*

ACCELERATION *= (startAcceleration; (isRneeded + notIsRneeded) ; (notEneeded.ACCELERATION + !not Eneeded.Delta))*

The specified behaviour consists of five agent actions (e.g. *addToSynchrotron*, *inSynchrotron*, *startAcceleration*, *isRneeded* and *notEneeded*) and the corresponding actions marked with a negation; i.e. the pre-condition is not fulfilled.

Pre-accelerated in the injector (auxiliary accelerator) to a certain energy, the particles enter through the inflector channel into a vacuum toroidal chamber located in a ring-shaped magnet covering the entire orbit. We can determine the mass and charge of the particles if we work with ions (action *addToSynchrotron*):

addToSynchrotron *= Forall(i:int)(*
(1<=i<=particlesNum && particles(i).particlesType==ion) ->
(particles(i).mass = atoms(particles(i).partcleID).mass;
particles(i).charge = atoms(particles(i).partcleID).charge))

The injection occurs at a small value of the magnetic field. The particles start rotating in the synchrotron chamber along an orbit of constant radius (action *inSynchrotron*).

inSynchrotron *= Forall(i:int) (*
(1<=i<=particlesNum) ->
(particles(i).radiusOfMovement = radius;
particles(i).velocity = particles(i).charge magnetic FieldInduction *particles(i).mass))*

In one or more places of the ring, there are accelerating spaces, where particles passing through are accelerated in an alternating electric field. As the energy of the particles increases, their speed and frequency of rotation increase (action *startAcceleration*), so the constancy of the radius of the orbit is maintained by the corresponding increase in the magnetic field and the frequency of the accelerating fields (actions *isRneeded* and *notIsRneeded*).

startAcceleration *= (Forall(i:int) (*
(1<=i<=particlesNum) ->
(particles(i).energy = POW((POW(particles(i).mass,2) POW(lightSpeed,4) + POW((particles(i).charge * radius * magneticFieldInduction),2) * POW(lightSpeed,2)),0.5));*
*particles(i).radiusOfMovement = (particles(i).mass * particles(i).velocity) / (particles(i).charge*magneticFieldInduction))),*

isRneeded = *(Forall(i:int)*
((1<=i<=particlesNum && particles(i).radiusOfMovement == radius) ->
(1))),

notIsRneeded = *(Forall(i:int)*
((1<=i<=particlesNum && particles(i).radiusOfMovement != radius)) ->
*(magneticFieldInduction = (particles(i).mass * particles(i).velocity) / (particles(i).charge* radius))))*

If a proton beam suddenly flies faster than the required speed, then it flies into the acceleration gap at a negative value of the voltage and slows down. If the speed of movement is lower, the effect will be the opposite — the particle is accelerated and catches up with the main flow of protons. As a result, a dense and compact beam of particles moves at the same speed. Reaching the required energy value can then be controlled by changing the magnetic field induction value (action Not_E_Needed):

notEneeded = *(Forall(i:int)*
((1<=i<=particlesNum && particles(i).energy < energyNeeded) ->
(magneticFieldInduction = magneticFieldInduction + increasingInductionCoef)))

Let us consider the proton as an example of an accelerated particle. The proton is an elementary particle without internal degrees of freedom, with energy of rest mc^2 = 938,27 MeV and with charge qp = +1.602·10–19 C. To calculate ionisation losses and proton scatter on atoms and nuclei, we need to know proton velocity $\vec{v}$ or its impulse $\vec{p}$, which can be calculated by determining kinetic energy E.

Let us describe the process of calculating these attributes in the form of a behavioural equation:

PROTONS_DATA = ((getParticlesVelocity); (getParticlesMomentum)),

where *getParticlesVelocity* and *getParticlesMomentum* are the actions of the agent of the PARTICLE type. We find proton velocity v — action *getParticlesVelocity* and then impulse p — action *getParticlesMomentum*.

getParticlesVelocity = *(Forall (i:int)*
((1<=i<=particlesNum)->
*(particles(i).velocity = particles(i).charge * magneticFieldInduction * particles(i).mass)),*

In this case, the velocity of the particle at the time of the departure of the particles from the accelerator is determined by the formula $v = qBm$.

The proton impulse is determined by the ratio

$$\frac{v}{c} = \frac{pc}{E + mc^2}.$$

getParticlesMomentum = *(Forall (i:int)*
(1<=i<=particlesNum)->
*(particles(i).momentum = (particles(i).velocity * (particles(i).energy + particles(i).mass * POW(lightSpeed,2))) / POW(lightSpeed,2))),*

Thus, we obtain a formal model of the particle accelerator, which allows us to model the use of synchrotrons to accelerate protons, electrons and ions.

Forward Algebraic Modelling of the Processes of Interaction of a Proton Beam with Matter

Sequences of the possible proton interactions with the substance (inelastic interactions with the electrons of atoms, such as inhibition and braking ability, elastic interactions with atomic nuclei (scattering) and nuclear reactions) are considered in the form of behavioural equations.

For each interaction, we determine the transition using a hybrid scheme, such as the addition of a differential equation that identifies the change in the number of particles carried through a unit surface, depending on the time; the change in the energy of the radiation absorbed by the substance per unit mass; and the differential angular distribution of particles after passing a layer of matter with a certain thickness.

For example, the linear brake capacity of a particle at depth x is determined by the formula $S = \left(-\frac{dE}{dx}\right)$. The action of calculating the linear brake capacity of the particle will then be formalised as follows:

GetParticleS = *(Forall (i:int)*
(1<=i<= partNum && substance(subNum).currentIrradiationSegmentLength <= substance(subNum).irradiationSegmentLength) ->
(DIFF(1))),

where DIFF(1) is the differential equation presented in the following form:

*1: -d(particles(i).energy) = substance(subNum).BrakingCapacity * d(substance(subNum).currentIrradiationSegmentLength)*

Accordingly, we will obtain the ***ProtonMaterInteraction.act*** file (contains a discrete component) with the formalisation of all agents' actions; the ***ProtonMaterInteraction.diff*** file with the formalisation of the differential equations used in the model; the ***ProtonMaterInteraction.behp*** file, which describes behaviour; and the ***ProtonMaterInteraction.env_descript*** file with the formalisation of agent types, agent attributes, initial values of the attributes, and the environment.

As water is considered an excellent tissue substitute for modelling and conducting experiments in proton therapy because of its similar density and other properties, this substance is chosen for the first experiments.

For water, the empirical ratio between the energy of a particle and its distance in the environment R is written as $R = \alpha E^p$, $\alpha = 1.9 \times 10^{-3}$, for a proton and 1.73×10^{-3} for α-particle, $p = 1.8$ for a proton and 1.5 for α-particle.

For the case in which the energy of the remaining beam should be enough to pass the path ($R - x$), we get $R - x = \alpha E(x)^p$. It follows from this expression that the expression for energy will have the following form: $E(x) = \left(\frac{R-x}{\alpha}\right)^{1/p}$. By differentiating this equation along the coordinate, we will determine the linear stopping power of the particle at depth x, and we can determine the formula for calculating the radiation dose distribution.

The dose distribution ($D(x)$) formula during the passage of a monochromatic proton beam in water will have the following form:

$$D(x) = \left(-\frac{1}{\rho}\frac{dE}{dx}\right) = \frac{1}{\rho p \alpha^{1/p}(R-x)^{1-1/p}},$$

where ρ is the density of the substance and will be formalised as follows:

getCurrentDose *= (((1) ->*
*(matter(matterNum).currentDose = 1/(matter(matterNum).substanceDensity * koefP * POW(koefAlpha, (1/koefP))* POW((matter(matterNum).IrradiationSegmentLength — matter(matterNum).currentIrradiationSegmentLength),(1–1/koefP))))),*

Accordingly, the complete behaviour equation for this experiment will be written in the following form:

B = ((GET_SUB_DATA) ; (SYNCHROTRONWORK); (PROTONS_BEAM_DATA);
(PROTON_SUBSTANCE_INTERACTION)),
GET_SUB_DATA *= ((setAtomsData); (getAmountMolM);*
(GetSubstanceMass);
(GetSubstanceDensity)),
SYNCHROTRONWORK *= (*
(addToSynchrotron); (inSynchrotron); (ACCELERATION)),
ACCELERATION *= (*
(startAcceleration); (isRneeded + notIsRneeded);
(noEneeded.ACCELERATION + !noEneeded)),
PROTONS_BEAM_DATA = ((getParticlesVelocity); (getParticlesMomentum)),
PROTON_SUBSTANCE_INTERACTION *=*
((getCurrentDose); ((ChangeX;PROTON_SUBSTANCE_INTERACTION) + !ChangeX))

Using algebraic methods, we can determine an example of a scenario of the desired property reachability (e.g. the necessary characteristics of a proton beam) with the specific attributes of the environment. An example of a fragment of the formula of the environment's initial state for specific algebraic modelling is as follows:

*…&& energyNeeded == 25*POW(10,6) && magneticFieldInduction == 1.3 &&*
radius == 28 && Forall (i:int) ((1<= i <= particlesNum) &&
particles(i).particlesType == proton && particles(i).initialMass == 1.6726/POW(10,27) &&
*particles(i).initialEnergy == 1.4*POW(10,9) && particles(i).velocity == 0 &&*
particles(i).radiusOfMovement == 0 && particles(i).charge == 1.6022/POW(10,19)) &&
substances(1).moleculesID == 1 && substances(1).irradiationRadius == 0.05 &&
substances(1).irradiationSegmentLength == 0.25 && substances(1).absorbedDose == 1.2 && substances(1).molecNum == 100 && molecules(1).atomsNum == 3 &&
molecules(1).atomsNames(1) == H && molecules(1).atomsNames(2) == H &&
molecules(1).atomsNames(3) == O && molecules(1).bondV(1,3) == 1 &&
molecules(1).bondV(2,3) == 1 && …

Thus, as a result of this task modelling, we can receive charts of the change in the energy of the proton beam in the substance, the distribution of the radiation dose and so on.

Having defined the state of the agents and the environment as symbolic, we will determine the

presence of the scenario, provided that it exists in a symbolic form. For example, it is possible to determine the scenarios of reaching the Bragg peak at a given depth for a given range of initial proton beam energy values and the degree of irradiation of healthy cells for different types of particles. In this case, as an example, we can set the possible energy ranges of the proton beam as follows:

…&& (15 POW(10,6) <= energyNeeded) &&(energyNeeded < 25* POW(10,6)) &&…*

Thus, we can consider the model behaviour for different initial conditions (e.g. irradiation time, angle of the particles' introduction, the substance to be irradiated and the length of the beam path in the substance), which will allow us to study and analyse the necessary parameters for improving the characteristics of the proton beam and system dose delivery to the patient.

Solving differential equations in the insertion modelling system is considered part of the software complex of the algebraic modelling system (APS) in Aplan language. A special converter and solver were developed for formal models that contained differential equations. Therefore, the ProtonMaterInteration.ACT and ProtonMaterInteration.diff files will be transferred to the converter of differential equations, which will convey the prepared equations to the solver by completing the necessary transformations. The resulting equations will be substituted in the ProtonMaterInteration. ACT file instead of DIFF (<Number>). Other mathematics systems, such as Mapple, can also be used to solve the equations.

Solving or using an approximation of the solutions of these equations will determine the step of algebraic modelling.

Backward Algebraic Modelling of the Interaction Processes of a Beam of Protons with Matter

As described above, by having a process model and specifying the properties that need to be reached, one can perform backward modelling according to the formal knowledge that defines all possible agent interactions at a given level of abstraction. In this case, the model simulation results will be a set of initial attributes of the agents and the environment.

Determining the necessary initial values of the energy of the proton beam is possible by knowing the characteristics of the irradiated substance and the path length that the proton beam must pass before stopping in the tumour (reaching the Bragg peak).

In this case, we will essentially use the same algebraic model as that for forward modelling (i.e. all formal domain knowledge that defines all possible interactions at a given level of abstraction must be considered). The difference is in the writing of the formula for the environment's initial values — we should not specify the initial values of the attributes that we have at the beginning of the experiment but the properties or desired values of the attributes of the agents that should be obtained at the end of the process. For example, we can specify the value of the maximum absorbed dose and the length of the path that the protons must take before localisation in the tumour:

…&& energyNeeded == 0 && Forall (i:int) ((1<= i <= particlesNum) &&
particles(i).particlesType == proton && particles(i). initialMass == 1.6726/POW(10,27) &&
*particles(i).initialEnergy == 1.4*POW(10,9) && particles(i).velocity == 0 &&*
particles(i).radiusOfMovement == 0 && particles(i). charge == 1.6022/POW(10,19)) &&
substances(1).moleculesID == 1 && substances(1).irradiationRadius == 0.05 &&
substances(1).irradiationSegmentLength == 0.25 && substances(1).absorbedDose == 1.2 && …

Accordingly, determining the values/ranges of values of the maximum possible radiation doses is feasible by formalising the mechanisms and regularities of irradiation processes and by considering the indicators of the formation of tumours' individual radiation sensitivity levels. The application of backward algebraic modelling will allow us to analyse the necessary indicators for improving the characteristics of the proton beam and systems for delivering doses to patients.

Considering all the processes at the level of interactions of individual substance atoms, particularly at the level of quantum interaction, we can check the reachability of the minimum effect of a

beam of charged particles on healthy cells and the level of tumour irradiation, completely modelling the real process of passing the beam through biological matters (e.g. maximally considering the change in density of the environment and ionisation losses).

Conclusion

The implementation of innovative technologies and research tools in cancer treatment, particularly radiation therapy, has brought it to a qualitatively new level in recent years. However, despite significant successes, this issue requires active experimental research, proof of safety and successful application results. In addition, imaging technology, treatment planning and delivery need further improvements and research. One of the effective, safe and cost-effective methods that allow solving this problem is computer molecular modelling.

Understanding the multi-scale nature of cancer and the processing of many intracellular and extracellular factors acting on different temporal and spatial scales becomes possible as a result of modelling these processes as complex hybrid systems.

Multi-level modelling based on a simulation or probabilistic approach is a daunting task, as it requires the processing of a huge amount of information. Using an algebraic approach allows us to abstract from information of varying levels of complexity and to solve complex problems by deducing relevant knowledge using formal methods.

The main advantage of using algebraic modelling is that it provides an opportunity to derive consequences from laws and, therefore, can offer new facts and theories that will enable us to solve complex problems. In other words, using an algebraic approach allows for the formal proof of certain properties of objects (in this case, charged particles, atoms, organic and inorganic substances, and so on) or processes or the search for objects or the values of their parameters that correspond to given properties. Using an algebraic approach allows the modelling of reversible reactions and biochemical processes, taking into account the structure and dimensions of cell compartments. Moreover, we do not focus on the laboratory-defined rules of chemical reactions but model their course based on the laws of quantum physics, quantum chemistry, and quantum mechanics.

On the other hand, the complexity of algebraic models can lead to the impossibility of solving some problems using SMT (Satisfiability Modulo Theories) solvers or to exponential data explosion, which becomes possible because of the need to model complex hybrid systems. In this case, a possible way to address the situation is using approximation and abstraction methods, or slices or heuristics, and applying the technology of combined use of algebraic modelling and AI methods. In particular, we create a neural system that analyzes the current state of the environment and determines the most effective action that will reach to the desired property. We train a neural network on a sequence of actions that reach a certain property. The neural network will determine an action as effective if it leads to the desired property, which will significantly narrow the search.

In the first research stage, the work of particle accelerators (cyclotron and synchrotron), the structure and possible interactions of particles, atoms, organic and inorganic substances, and the physics of the interaction of a proton beam with matter were formalised at the upper level.

The next stage involved modelling the interactions of the beams of the accelerated particles (protons and ions) with amino acids/proteins/cells and, in particular, modelling the possible interactions of protons with matter at the level of quantum interactions.

Although the research is not yet finished, and we are currently working on expanding the base of formalized knowledge, the first obtained results of modelling interatomic and intermolecular interactions, intracellular processes (apoptosis) indicate that the algebraic modelling method is an effective and promising approach to conducting research in oncology, in particular in research methods and tools of radiation therapy, chemotherapy, researching the development and application of nanoparticles, etc. In radiation therapy, this is the modelling and detection of the properties of systems and radiation beams aimed at causing the most minor damage to healthy cells (with maximum conside-

ration of changes in the density of the environment, ionization losses, radiosensitivity, and so on.). In pharmacology — the search for effective pharmacological (biologically active) agents.

One of the directions of our further work is the application of algebraic modelling methods to the modelling and research of processes and methods of regulation of intercellular and intracellular cascades of signal transmissions, which take part in the process of cell apoptosis, taking into account the influence of radiation therapy. In particular, it is envisaged to model the processes that cover:

- Study of the effect of radiation on cell DNA damage and increase in ROS (reactive oxygen species) as prerequisites for activating the cell's internal apoptosis pathway.
- Study the particle acceleration processes and passage of a beam of charged particles in matter. In particular, modelling of proton and heavy ion therapy to determine the necessary values of beam energy, angle and duration of irradiation to reaSch the Bragg peak at a given depth for various environmental parameters, and so on.
- Study of nanoparticles' influence on programmed cell death processes in combination with radiation therapy (oxidative stress induction and selective death of tumour cells; protection of healthy cells from exposure to radiation).

REFERENCES

1. Dobbe, R., Tomlin, C. J., 2015. "Hybrid Systems Modeling for (Cancer) Systems Biology". bioRxiv.
2. Lincoln, P., Tiwari, A., 2004. "Symbolic systems biology: Hybrid modeling and analysis of biological networks". *Hybrid Systems: Computation and Control: 7th International Workshop*, HSCC 2004, Philadelphia, PA, USA, March 25–27. Proceedings 7 (Springer Berlin Heidelberg, 2004), pp. 660–672.
3. Abate, A., Hillen, R. C., Aljoscha, S., 2012. "Wahl Piecewise affine approximations of fluxes and enzyme kinetics from in vivo 13C labeling experiments". *International Journal of Robust and Nonlinear Control*, 22(10), pp. 1120–1139.
4. Jones, B., 2017. "Clinical radiobiology of proton therapy: modeling of RBE". *Acta Oncologica*, 56(11), pp.1374–1378.
5. Chen, Y., Ahmad, S., 2011. "Empirical model estimation of relative biological effectiveness for proton beam therapy". *Radiation Protection Dosimetry*, 149(2), pp.116–123.
6. Dahle, T. J., Rykkelid, A. M., Stokkevåg, C. H., Mairani, A., Görgen, A., Edin, N. J., Rørvik, E., Fjæra, L. F., Malinen, E., Ytre-Hauge, K. S., 2017. "Monte Carlo simulations of a low energy proton beamline for radiobiological experiments". *Acta oncologica* (Stockholm, Sweden), 56(6), pp.779–786.
7. Bitencourt-Ferreira, G., Pintro, V., de Azevedo, W., 2019. "Docking with AutoDock4". *Methods in Molecular Biology*, pp.125–148.
8. Hughes-Oliver, J.M., Brooks, A.D., Welch, W.J., Khaledi, M.G., Hawkins, D., Young, S.S., Patil, K, Howell, G.W., Ng, R.T., Chu, M.T., 2012. "ChemModLab: a web-based cheminformatics modeling laboratory". *In Silico Biol*, 11(1-2), pp. 61–81.
9. Morency, L., Gaudreault, F. and Najmanovich, R., 2018. "Applications of the NRGsuite and the Molecular Docking Software FlexAID in Computational Drug Discovery and Design". *Methods in Molecular Biology*, pp. 367–388.
10. Pirhadi, S., Sunseri, J., Koes, D., 2016. "Open source molecular modeling". *Journal of Molecular Graphics and Modelling*, 69, pp. 127–143.
11. Liu, K., Sun, X., Jia, L., Ma, J., Xing, H., Wu, J., Gao, H., Sun, Y., Boulnois, F., Fan, J, 2019. "Chemi-Net: A Molecular Graph Convolutional Network for Accurate Drug Property Prediction". *Int. J. Mol. Sci*., 20, 3389.
12. Mathpal, D., Masand, M., Thomas, A., Ahmad, I., Saeed, M., Zaman, G.S., Kamal, M., Jawaid, T., Sharma, P.K., Gupta, M.M., Kumar, S., Srivastava, S.P., Balaramnavar, V.M., 2021. "Pharmacophore modeling, docking and the integrated use of a ligand- and structure-based virtual screening approach for novel DNA gyrase inhibitors: synthetic and biological evaluation studies". *RSC Advances*. Vol. 11(55), pp. 34462–34478.
13. Lin, X., Li, X., Lin, X., 2020. "A Review on Applications of Computational Methods in Drug Screening and Design". *Molecules*, vol. 25(6):1375.
14. Beentjes, C.H.L., Baker, R.E., 2019. "Quasi-Monte Carlo Methods Applied to Tau-Leaping in Stochastic Biological Systems". *Bull Math Biol*, vol. 81, pp. 2931–2959.

15. Letichevsky, A., Gilbert, D., 1999. "A Model for Interaction of Agents and Environments". In: Bert D., Choppy C., Mosses P.D. (eds) Recent Trends in Algebraic Development Techniques., WADT 1999, *LNCS,* vol. 1827, pp. 311–328.
16. Letichevsky, A.A., Godlevsky, A.B., Letychevsky, A.A., Potiyenko, S.V., Peschanenko, V.S., 2010. "Properties of a predicate transformer of the VRS system". *Cybernetics and Systems Analysis*, 46, pp. 521–532.
17. Model Creator Tool [online]. Available at: <https://rd.litsoft.com.ua/> [Accessed 06 Oct. 2024].
18. Letichevsky, A., Letychevskyi, O., Peschanenko, V., 2016. "Insertion modeling and its applications". *Computer Science Journal of Moldova,* 72(3), pp. 357–370.
19. Apsystem.org.ua. 2022. APS & IMS [online]. Available at: <http://apsystem.org.ua/> [Accessed 06 Oct. 2024].

Received 30.10.2024

Ю.Г. Тарасіч, доктор філософії, молодший науковий співробітник,
Інститут кібернетики імені В. М. Глушкова НАН України,
просп. Академіка Глушкова, 40, м. Київ, Україна, 03187,
ORCID: https://orcid.org/0000-0002-6201-4569,
yutarasich@gmail.com

АЛГЕБРАЇЧНЕ МОДЕЛЮВАННЯ ЕКСПЕРИМЕНТІВ НА ПРИКЛАДІ ПРОТОННОЇ ТЕРАПІЇ

Вступ. Незважаючи на стрімкий розвиток хімічної промисловості та науки, відкриття у галузі охорони здоров'я, появу ліків і терапевтичних засобів на основі нанотехнологій та розвиток технологій променевої терапії, безпека біомедичних застосувань новітніх продуктів та пошук нових методів і підходів до діагностики та лікування раку є відкритим питанням.

Методи. Одним із найбезпечніших і найшвидших методів дослідження поведінки нових матеріалів та інструментів є моделювання відповідних процесів, зокрема комп'ютерне молекулярне моделювання на основі математичних моделей. Однак, незважаючи на велику кількість доступних методів і засобів моделювання, для більшості з них успішне застосування можливе лише для вузького кола завдань і експериментів.

Мета статті. Як один із можливих шляхів розв'язання цієї проблеми ми пропонуємо новий підхід до комп'ютерного молекулярного моделювання, заснований на синергії алгебраїчного підходу та біологічних знань на різних рівнях абстракції, починаючи від квантових взаємодій до взаємодій біологічних систем.

Методи. Один із напрямів застосування запропонованого підходу ми бачимо в можливості моделювання процесу променевої терапії — починаючи від моделювання роботи прискорювачів і закінчуючи моделюванням взаємодії пучка частинок з речовиною на рівні квантових взаємодій.

Результати. Зокрема, у статті розглядаються можливості прямого (конкретного та символьного) та оберненого (символьного) алгебраїчного моделювання на прикладі моделей вищого рівня абстракції, що дозволяє візуалізувати певні взаємодії та будувати діаграми залежностей для конкретних моделей, а також визначати наявність бажаних сценаріїв (пряме моделювання) або набору початкових параметрів середовища (обернене моделювання) у символьній формі.

Ключові слова*: *молекулярне моделювання, алгебраїчне моделювання, моделювання біологічних експериментів, моделювання протонної терапії, теорія взаємодії агентів і середовищ, символьне моделювання.