Computer tools in education, 2021
№ 2: 58–65
http://cte.eltech.ru
doi:10.32603/2071-2340-2021-2-58-65

INTERACTIVE PRESENTATION OF MATHEMATICAL OBJECTS

Pankov P. S.[1], Doctor of Physical-Mathematical Sciences, Professor, ✉ pps5050@mail.ru, orcid.org/0000-0002-0261-3113
Burova E. S.[2], Assistant Professor, burova_e@auca.kg

[1]National Academy of Sciences of Kyrgyzstan, Chui ave. 256-a, Bishkek, 720071, Kyrgyz Republic
[2]American University of Central Asia, 7/6 Aaly Tokombaev Street, Bishkek, 720060, Kyrgyz Republic

Abstract

Earlier, the first author introduced a definition of independent computer presentation of an object (the user can master the object without reference to similar objects). The paper presents examples of descriptions of interactive mathematical models: for notions in natural languages (without any other languages as media; verbs are presented as the user's actions); for mathematics itself (with minimal explanations) and for physics.

Keywords: *mathematics, tasks, computer presentation, independent presentation, interactive presentation.*

Citation: P. S. Pankov, E. S. Burova, "Interactive Presentation of Mathematical Objects," *Computer tools in education*, no. 2, pp. 58–65, 2021; doi: 10.32603/2071-2340-2021-2-58-65

Acknowledgements: *The authors are grateful to the Presidium of National Academy of Sciences of Kyrgyzstan for partial support of this investigation.*

1. INTRODUCTION

Earlier, we introduced [1] a definition of independent computer presentation of an object (without reference to similar objects). Particularly, it means that the user is able to master foundations of the subject by using corresponding software (with interactive actions with feedback) without any preliminary knowledge, regardless or with minimal use of their native language. Also, we introduced definitions of "almost-closed or affectable objects" (including both humans and computers), of "commands" and "language".

The aim of this paper is to recount some general statements and to propose examples of interactive mathematical models in various fields. We do not refer to existing software because we hope that the programmer would be able to implement mentioned and other models better. Moreover, the programmer may test sketches with their colleagues and students and receive proposals and notes immediately.

The second section contains statements on human-computer interaction generally.

The third section expounds a method to present natural languages independently by means of mathematical models of notions. Verbs are presented as the user's actions.

Remark. Existing vocabularies do not "define" notions. They explain words by other words which is a circular reasoning.

The fourth section proposes various mathematical notions as tasks to be solved by "Drag-and-Drop".

The fifth section contains three examples in physics.

Remark. Because of a huge amount of publications and existing software, we do not insist on originality of all proposals in the paper. For instance, similar methods are implemented in [2]. We hope that our wordings are more general.

Main features of this paper were presented at the online conference "Computer Assisted Mathematics" on July 29, 2021.

2. DEFINITIONS and HYPOTHESES

Definition 1. If low energetic outer influences can cause sufficiently various reactions and changing of the inner state of the object (by means of inner energy of the object or of outer energy entering into object besides of influences) at any time then such (permanently unstable) object is an *almost-closed* or *affectable* object (af-object below), and such outer influences are *commands*.

Definition 2. A system of commands such that any af-object can achieve desired efficiently various consequences from other one is a language.

These definitions unite humans and computers.

Remark. The human does not have a built-in timer, so a well-known definition of universal language (mentioned by Church's thesis) does not include commands of type "in ten minutes output the result". But such commands are necessary for a genuine universal language.

The second law of thermodynamics says that each action of an af-object increases entropy H, $\Delta H > 0$. R. Landauer [3] proved that while treating one bit of information $\Delta H \geq k_B log 2$ (k_B is the Boltzmann constant).

Let t_{pass} be the time of an af-object's passing from the first stationary state to the second one. R.Landauer noted that ΔH increases as t_{pass} decreases. We substituted the following

Hypothesis 1 [4]. $\Delta H \geq const \cdot t_{pass}^{-2}$ as t_{pass} decreases.

Definition 3. Let any object (real or virtual) be meant. If an algorithm acting at a computer: generates (randomly) a sufficiently large amount of instances covering all essential aspects of the object, gives a command related to the object in each situation, perceives the user's actions and performs their results clearly on a display, detects whether a result fits the command, then such algorithm is said to be a computer interactive presentation of the object.

To implement such presentation, the following construction is necessary for a programmer.

Definition 4. An interactive mathematical model consists of: description of media (objects presented as sets on display); permitted and prohibited relations between objects (overlapping, intersection, inclusion) and the user's actions (moving and transforming objects); a command with random generation of auxiliary objects or words; temporal sequence of relations between objects to be done by the user in order to guess and fulfill the command.

There are two main ways to master the situation: to make experiments "by means of the user's arbitrary actions" (for mathematics) or to use life experience (for physics).

To treat such software the user should have the eye ability and the "intellectual eye" or "measuring imagery" ability (the term and the way to measure was proposed [1]) . As a real example, consider the activity of air controllers before involving computers. The human hears numbers from the pilots, constructs a dynamic process in mind promptly, transforms it into numbers (without any calculations) and communicates them to the pilots. All these numbers are saved and (almost) exact calculations can be made. The difference between this result and the air controller's one is an objective numerical measure of the ability.

Proposed presentations can be classified as avatar (A-presentation) and non-avatar (N-presentation). They can be explained by “a controlled drone”: the user either watches the drone (A-presentation) or watches images sent by the video camera of the drone (N-presentation).

Remark. There is duality in perception of N-presentations. If the user watches some changes on the screen then they can imagine either self-motion in the space or motion of space with objects in it. The first is used in computer games and the second is used in mathematical software “Mathematica”, “Matlab”, “MathCad”.

The following hypothesis is based on Socrates’ method (dialogue “Meno”).

Hypothesis 2 [5]. By using suitable tasks and questions, proposing to conduct experiments the teacher can drive a student (or, more effectively, a group of students) at independent discovery of many facts of elementary mathematics, of discrete mathematics, some facts of analysis and of physics.

Analogous proposes were made by many authors, for instance, [6].

Sometimes successful solving a task (finding a fact) means mastering a new (for the student) notion. In such case the software or the teacher may announce: (C!) “Congratulations! You have mastered the notion...”.

Remark. Such announce must be made *after* success but not *before* (as the theme of lesson).

If software is used as an examination then the following principles are necessary. *Generativity*: a complete task must not exist before the testing and must be generated randomly. *Uniqueness*: each examinee obtains a different version of task, of the same level of difficulty. *Concreteness*: the examinee’s respond may be: exact or approximate number; word; short phrase; action "Drag-and-Drop" by a computer mouse.

3. MATHEMATICAL MODELS AND INDEPENDENT PRESENTATION OF LANGUAGES

We proposed to develop such software [7]. Immediately, the user begins to think in the learned language, without translation in mind.

The following techniques are proposed for the user’s guessing:

G1) uniqueness of the action (or the sequence of actions) which subdues the command and the situation naturally;

G2) similarity (some objects have the same property and this property is mentioned twice);

G3) alternation (new notion and new word appear together as an alternative to preceding notions).

G4) relation (new notion or two notions appear with an obvious relation between the new one and preceding ones or between them).

G5) using a narrative phrase that is naturally connected with the situation.

G6) the new noun only has a learned attribute.

G7) “if this task (given by preceding notions) has a solution then the only action solve it”.

G8) consolidation of knowledge.

Some examples of interactive mathematical models. Main learned words are denoted by capital letters; auxiliary and random learned words are denoted with italic capital letters; auxiliary random words are denoted with italic letters.

Commands are given both in oral and written forms. Further, for G8) some commands may be given in oral form only.

Unless otherwise agreed, intersections of initial positions of objects are empty.

3.1. Verb PUT and some nouns. N-presentation.

Environment: *THING*(movable); *PLACE*(fixed).

Conditions: 1)$THING \subset PLACE$.

(G1) Command: PUT *NAME-OF-THING* into *NAME-OF-PLACE*.
(Further, "NAME-OF-" will be omitted).
Additional environment: *THING2*.
Conditions: 2)*THING2* ⊂ *PLACE*.
(G3) Command: PUT *THING2* into *PLACE*.
Additional environment: *PLACE2*.
Conditions: 3)*THING* ⊂ *PLACE2*.
(G3) Command: PUT *THING* into *PLACE2*.

3.2. Colors. N-presentation.
Environment: *C1T* = *COLOR1 thing*, *C2T* = *COLOR2 thing*; *C1P* = *COLOR1 place*.
Conditions: 1) *C1T* ⊂ *C1P*. 2) *C1T* ⊂ *C1P*.
(G2)-(G3) Command: Put *COLOR1 thing* into *place*. Put *COLOR2 thing* into *place*.

3.3. Verb TAKE. N-presentation.
Environment: *thing* ⊂ *place1*; *place2*.
Conditions: 1) *thing* ∩ *place1* = ∅; 2) *thing* ⊂ *place2*.
(G1)Command: TAKE *thing* from *place1* and put it into *place2*.

3.4. Verbs related to animated objects. (Animated objects are moving (stirring) slightly). N-presentation. Environment: *ANIMATED*; *EDIBLE*.
Conditions: 1) *EDIBLE* ∪ *ANIMATED* ≠ ∅.
(G1) Command: GIVE *EDIBLE* to *ANIMATED*.
Environment: *animated*; *thing*; a wall with a hole less than the ball between them; (*thing* ∩ wall ≠ ∅) is prohibited.
Condition: 1) *thing*, hole and *ANIMATED's* eyes are collinear.
(G1) Command: SHOW *thing* to *animated*.

3.5. Avatar verbs. A-presentation.
Verb GO OUT (EXIT). Environment: A-object ⊂ FENCE(fixed).
Conditions: 1) A-object ∩ *place1* = ∅.
(G1) Command: GO OUT of the FENCE.
Verb COME IN (ENTER). Environment: A-object ⊂ FENCE(fixed).
Conditions: 1) A-object ⊂ *FENCE*.
(G1) Command: COME IN the FENCE.

3.6. Tool verb RING and preposition WITH.
Environment: A BELL(fixed); a STICK(movable).
Conditions: 1) BELL ∩ STICK ≠ ∅ [ringing].
(G1) Command: RING the BELL WITH the STICK.

3.7. Verb COMPOSE.
Environment: CIRCLE-WITHOUT-PART(fixed); PART(movable); PLACE-FOR-PART(not visible).
Conditions: 1) PART ≈ PLACE-FOR-PART.
(G1) Command: COMPOSE the CIRCLE.
Example of complicated notion.

3.8. Verb FIND.
Environment: Some circles(movable) [with a letter U under each]; circle(movable) [with a letter T under it]; board(fixed). (Firstly letters are not visible).
Preliminary: put, shift.
Conditions: 1) letter T ⊂ board.
(G5)-(G7) Command: Shift circles, FIND letter T and put it on board.

4. MATHEMATICAL OBJECTS AND NOTIONS

S. M. Ulam [8] put the problem of evident presentations of mathematical objects and processes with feedback. For instance, "pull the four-dimensional solid through the two-dimensional surface". We implemented some such processes by means of the notion of kinematical space, for instance [9, 10]. These and other items are described below. Background is in the spectrum from white till black; sometimes chess color (light grey and dark grey) for 2D-spaces is used. Drag-and-Drop object, or Avatar object is green and is denoted as A-object below. Function, or result of A-object is red and is denoted as F-object below. Target for F-object is yellow and is denoted as T-object below. Approaching T-object is accompanied by music of "hot-cold" type too. Tracks of A-object (light green) and F-object (light red) can also stay while 2D-motion.

Examples.

4.1. Solving of the equation $F(x) = 0$. A-point can move along the abscissa axis only. T-object is the abscissa axis.

4.2. Searching for $minF(x)$. A-point can move along the abscissa axis only. T-object is gradient of yellow color down.

4.3. Solving of the system of equations $F(x, y) = u, G(x, y) = v$ (firstly, linear ones; the user discovers linearity). A-point is (x, y), F-point is (F, G), T-point is (u, v).

4.3a. Solving of the equation $\sqrt{z} = w \neq 0$ for complex numbers. The origin $z = 0$ repels A-point. The user discovers the following: to reach T-point going around the origin is necessary.

4.3b. The item 4.3 is interpreted as transformations of the plane. For example, F-point is the mirror reflection of A-point.

4.4. Searching for $minF(x, y)$. The value of F and T-object as gradient of yellow color down are on a separate part of a display.

Measures as invariants.

4.5. Length of a curve. A-object with F-object is a red curve with the leading green endpoint. While pulling its length preserves. T-objects are several curves of various lengths. The user is to detect the T-object with corresponding length and pull A-object on this T-object. (C!) "length".

4.6. Area of a figure. A-object with F-object is a red rounded figure with the green boundary. While pulling its area preserves. T-objects are several figures of various areas. The user is to detect the T-object with corresponding area and pull A-object on this T-object. (C!) "area".

Remark. Programming of preserving area while continuous transformations of a figure is an interesting task itself.

4.7. N-presentations of non-Euclidean spaces filled with T-objects and brown Obstacles. The user drives a green car, with additional possibilities to put marks etc. The screen is the windshield of the car. The task is to find and erase T-objects without breaking Obstacles.

4.7a. Moebius band. The user can verify that a right boot left on a street will be met as a left boot after passing half of the street.

4.7b. Topological torus (a square with opposite sides glued). This space used to be discovered by many programmers independently. Motion in arbitrary direction will led to the initial position someday.

4.7c. Riemann surface of the function $\sqrt{z}$, with the third coordinate up. (See 4.3a above).

4.7d. Riemann surface of the function $\sqrt{(z^2 - a^2)}$, with the third coordinate up. Passing between two unbreakable pillars only leads to another part of the space.

4.7e. Motion with creating the Riemann surface of the function $H(z, w) = 0, H$ is a given polynomial. This is the only way to investigate its branching points and general structure.

These spaces allow multiple users who can see and meet others naturally.

4.7f. Projective plane with the third coordinate up. While motion along the street trees on this side move to us as usually but trees on the opposite side move from us. This space does not permit multiple users.
4.8. N-presentations for 4D-space filled with 4D-solids.
4.8a. The 3D-coordinates are presented as usually, the fourth coordinate (call it "deep") is denoted with continuous darkening of the environment. We look at the space through 3D-slit and can "deep" and "undeep". The task is to detect 4D-solids. For instance, the 4D- "deepical" cone is seen as the sequence "little ball" — "enlarging ball" — "none" while motion "deep".
4.8b. Denote coordinates as X, Y, Z, W. The user can choose each of 2D-subspaces XY, XZ, XW, YZ, YW, ZW, see projections of 4D-solids and rotate them around the chosen plane.

The task: to extract a right boot from 3D-space, make some rotations and return it to same 3D-space as a left boot.
4.9. Symmetry.
4.9a. The task. Fill in:

$$WTFkTWTFkTWTFkTWTF\_TTWTFkT$$

(C!) "translational symmetry".
4.9b. The task. Fill in:

$$WTH**TWT**H\_W$$

(C!) "reflection (mirror, axial) symmetry".
4.9c. The task. Fill in:

$$:IHHX\,|\,\#\#IN\_N$$

$$IOI$$

$$NONI\#\#\,|\,XHHI:$$

(C!) "central symmetry".
4.10. Initial value problem for ordinary differential equations.

There are two vertical lines (the right line is yellow), little arrows (right, right up, right down) in the strip between lines and red A-point at the left line. Task: "moving along arrows reach the yellow line".

If the trajectory drawn by the user is close to the solution of the initial value problem then (C!) "field of directions" else "Try again".

Advanced version: The user inputs a polynomial $f(x, y)$ and a number y_0. The software calculates values $f_k = f(x_k, y_k)$ for any rectangular grid and draws little arrows $(x_k, y_k) - (x_k + h, y_k + hf_k)$ between lines $x = a$ and $x = b$, puts the A-point at (a, y_0).

Task: "solve the initial value problem $y(a) = y_0$ for the differential equation $y'(x) = f(x, y'(x))$ approximately".

5. MATHEMATICAL MODELS OF PHYSICAL PROCESSES

Tasks on approximate forecasting of physical processes.
5.1. Trajectory of thrown ball.

The red ball (A-object) flies off the gun about 60° to a horizontal yellow line (T-object) with light green track and stops. If the user does nothing then the red ball flies off again. The user is to drag the red ball and continue the trajectory until the yellow line.

If the trajectory drawn by the user is an approximate parabola then (C!) "parabola" else "Try again".

5.2. Central of gravitation of a triangle.

Given a triangle with horizontal base.

"Where must a support be put for equilibrium?"

5.3. Radioactive decay of atoms.

Given a coordinate system with two marks: numbers of atoms of radioactive isotope (millions) at $time = 0$ and at $time = 1 hour$. The second number is less than the first one. A-point is at the first mark.

"Draw a graph of changing numbers of atoms of radioactive isotope from $time = 0$ until $time = 10 hours$."

If the graph drawn by the user is an approximate damped exponential curve then (C!) "damped exponential curve" else "Try again".

6. CONCLUSION

We hope that successful implementation of proposed and other such presentations of mathematical objects would distinguish new essential features of various mathematical objects and be interesting both for programmers and for users regardless their relation to mathematics.

References

1. P. S. Pankov, "Independent learning for Open society," in *Collection of papers as results of seminars conducted within the frames of the program "High Education Support"*, Foundation "Soros-Kyrgyzstan", Bishkek, 1996, no. 3, pp. 27–38.
2. V. N. Dubrovskii, "Visualization of Functional Dependences in Dynamic Geometry Systems," *Computer tools in education*, no. 4, pp. 93–112, 2020 (in Russian); doi: 10.32603/2071-2340-2020-4-93-112
3. R. Landauer, "Irreversibility and heat generation in the computing process," *IBM Journal of Research and Development*, vol. 5, no. 3, pp. 183–191, 1961; doi: 10.1147/rd.53.0183
4. P. S. Pankov, "Adiabaticheskie pokazateli zamknutykh sistem" [Adiabatic exponents of closed systems], *Vestnik KNU im. Zh. Balasagyna. Seriya 3. Estestvenno-tekhnicheskie nauki*, pp. 146–147, 2003 (in Russian).
5. P. Pankov, Zh. Dzhanalieva, and A. Naimanova, *Induktivnoe i eksperimental'noe izuchenie matematicheskikh distsiplin (Matematicheskie fakty i ponyatiya, kotorye mogut byt' obnaruzheny samostoyatel'no)* [Inductive and experimental study of mathematical disciplines (Mathematical facts and concepts that can be discovered on one's own)], Saarbrücken, Deutschland: Lap Lambert Academic Publishing, 2015 (in Russian).
6. 1C Company, V. N. Dubrovskii, V. A. Bulychev et al., "How to make a mathematical discovery," in *obr.1c.ru.* [Online] (in Russian). Available: https://obr.1c.ru/mathkit/lessons5.html
7. P. S. Pankov, J. Sh. Aidaraliyeva, and V. S. Lopatkin, "Active English on computer," in *Proc. of Conf. Improving Content and Approach in the Teaching of English Language in the Context of Educational Reform*, Bishkek, 1996, pp. 25–27.
8. S. M. Ulam, "Chapter VIII. Computing machines," in *A Collection of Mathematical Problems*, New York: Interscience Publ., 1960.
9. P. S. Pankov, B. J. Bayachorova, and A. V. Terehin, "Computer Presentation of Four-Dimensional Spaces," *in Proc. of The 8-th Int. Conf. on Computer Graphics and Visualization "GraphiCon-98"*, Moscow, 1998, p. 204.
10. A. A. Borubaev, P. S Pankov, and A. A. Chekeev, *Spaces Uniformed by Coverings*, Budapest: Hung.-Kyrgyz friendship soc, 2003 (in Russian).

Received 05.07.2021, the final version — 30.07.2021.

Pavel Pankov, Doctor of Physical-Mathematical Sciences, Professor, Corresponding Member of National Academy of Sciences of Kyrgyzstan, Head of Laboratory of Computational Mathematics of Institute of Mathematics, ✉ pps5050@mail.ru

Elena Burova, Assistant Professor of Applied Mathematics and Informatics Department of American University of Central Asia, burova_e@auca.kg

Компьютерные инструменты в образовании, 2021
№ 2: 58–65
УДК: 004.9:510
http://cte.eltech.ru
doi:10.32603/2071-2340-2021-2-58-65

Интерактивное представление математических объектов

Панков П. С.[1], доктор физико-математических наук, профессор, ✉ pps5050@mail.ru, orcid.org/0000-0002-0261-3113
Бурова Е. С.[2], старший преподаватель, ✉ burova_e@auca.kg

[1]Национальная академия наук Кыргызстана, 256-а, просп. Чуй, 720071, Бишкек, Кыргызская Республика
[2]Американский университет в Центральной Азии, 7/6, ул. Аалы Токомбаева, 720060, Бишкек, Кыргызская Республика

Аннотация

Ранее первый автор ввел определение независимого компьютерного представления объекта (пользователь может изучить объект без знаний об аналогичных объектах). В статье представлены примеры описаний интерактивных математических моделей: для понятий на естественных языках (без каких-либо других языков в качестве носителей; глаголы представлены как действия пользователя); для самой математики (с минимальными пояснениями) и для физики.

Ключевые слова: *математика, задачи, компьютерная презентация, независимая презентация, интерактивная презентация.*

Цитирование: Панков П. С., Бурова Е. С. Интерактивное представление математических объектов // Компьютерные инструменты в образовании. 2021. № 2. С. 58–65. doi: 10.32603/2071-2340-2021-2-58-65

Поступила в редакцию 05.07.2021, окончательный вариант — 30.07.2021.

Панков Павел Сергеевич, доктор физико-математических наук, профессор, член-корреспондент Национальной академии наук Кыргызстана, заведующий лабораторией вычислительной математики Института математики, ✉ pps5050@mail.ru

Бурова Елена Сергеевна, старший преподаватель кафедры прикладной математики и информатики Американского университета в Центральной Азии, burova_e@auca.kg