CaSTR 2009
Computer Algebra Systems in Teaching and Research

Computer Algebra Systems in Teaching and Research

Mathematical Physics and Modeling in Economics, Finance and Education

University of Podlasie
(Siedlce, Poland)
www.imif.ap.siedlce.pl

The College of Finance and Management
(Siedlce, Poland)
www.wsfiz.siedlce.pl

WYDAWNICTWO
WYŻSZEJ SZKOŁY FINANSÓW I ZARZĄDZANIA
SIEDLCE 2009

Chapter 1. Mathematical Modeling and Mathematical Physics

Technological Aspects of Creation and Maintenance in System Mathematica of the Knowledge Base of Forest Fire Models

Dmitry Barovik [1)] Valery Taranchuk [2)]

[1)] Belarussian State University
Minsk, Belarus
dimfpmi@tut.by

[2)] Belarussian State University
Minsk, Belarus
taranchuk@bsu.by

Abstract. *Technological and pedagogical aspects of integration of specialized and system software for computer modelling, structure and architecture of the software complex for forest fire modelling realized in* Mathematica, *the results of modelling knowledge base accumulation and geovisualization are discussed.*

1 Introduction

Mathematical modelling of processes and effects in different areas of natural sciences and economy gets more and more wide application. Increasing possibilities of computer hardware and information technologies, a demand of computer models in many different fields of activity brings a special importance to the process of education of experts in creation, adaptation, support of mathematical models. Designing, developing of computer models in a format of dialogue interactive systems demand knowledge in a very wide range of subjects, such as applied mathematics, programming, economy. Efforts of different experts which should be prepared are necessary. Reformation of a modern education system is required, in particular, in the direction of information and telecommunication technologies integration into educational process and into it's management. The special role is provided to essentially new methods of the educational material organization, creation of modern educational environments knowledge bases and the corresponding software providing effective tools for information receiving, self-training and self-testing.

One of the key components of computer models is software. Reasonable and effective combination, integration into single system of services and specialized modules, programs, packages is important. An effective program platform for computer models creation, preparation of interactive informational-pedagogical environments of new generation are systems of computer mathematics ([1]). The significant part of creation of interactive electronic educational resources is spend to the creation of computer graphics, necessary illustrations at different scenarios of work, and the problems of visualization of scientific data have absolutely independent role. Computer algebra system leaders are **Mathematica, Maple, MATLAB, MathCAD, Simulink,** etc. In these software complexes mathematical calculations, transformations and simplifications of expressions are carried out after inputting the equations, conditions and functions in the traditional mathematical notation. Practice of last years confirms efficiency of application of systems of intellectual calculations at the decision of problems of mathematical physics, mechanics of continuous environments, economy, etc.

In the work presented features of designing and realization of computer models of the forest fires, corresponding software services are described. Questions of use of functions of the kernel of system **Mathematica,** programming of specialized modules for knowledge base filling in and support are discussed; methodical features of the organization of the calculus experiments and processing of their results are stated.

2 Client-Server Architecture of a the Software Complex and its Realization by Functions of the Kernel of System Mathematica

Mathematical modelling of forest and peat fires is the major part of decision-making support systems and expert systems, used at the analysis of ecological conditions. Corresponding computer models are necessary for forecasting of development of extreme situations at concrete objects, large forests and different areas. Mathematical forest fire models developed nowadays (for example, [2, 3]) are very difficult, calculation of any variant requires much time even on enough powerful computers. Corresponding computer models can only be used by authors-developers and are not focused on inclusion in geographical information systems (GIS) and in systems of on-line information service.

We consider that modern IT level gives possibilities for designing and filling in knowledge bases of typical numerical solutions of problems of geoecology, for designing simplified engineering models and inclusion them in GIS, for creation of computer express models of the various natural phenomena. Computer models can effectively be created on a platform of different systems of intellectual calculations where they can be designed as interactive with possibilities of "connection" of estimations from the knowledge base. A possible principle of functioning of system is working with models at different levels. Applied mathematicians create and main-

tain computer models. Experts in some field of knowledge, registered in the server, work with models on one of the offered scenarios. At such level users work with system and computer models as "a black box", not distracting with "a mathematical stuffing", get results varying the input information and work in "easy to understand" environment. Users get output information in form of regulation documents, tables of data, maps and schemes. Administrators of system grant access for other users working with knowledge bases from client computers. At such level, calculations on mathematical models are not taken place; results are withdrawn from the knowledge base, processed and sent to users. For this purpose a wide enough set of specialized algorithms and of typical functions for data processing and visualization should be provided.

With such requirements to algorithms and software the separate role is taken away to software complex services designing and programming. Developers of such computer software face complicated requirements it's necessary to develop systems on program platforms where mathematical calculations are carried out by the system's kernel, the interface can be adjusted by the user, client-server architecture realization is possible.

One of the systems, satisfying the formulated requirements, is the system of computer algebra **Mathematica. In** it mathematical calculations, transformations and simplifications of expressions are carried out by the system itself after writing down equations, conditions and functions in the traditional mathematical notation, and a wide range of typical algorithms of preprocessing and visualization are realized by kernel functions ([4]). Practice of last years confirms efficiency of application of systems of intellectual calculations for solving problems of modelling, in particular, one of the forest fire models realized in **Mathematica** is described in [5].

Let's describe suggested regulations of functioning of the software complex for forest fires modelling, its architecture and the basic components. Fig. 1 shows the scheme of the software complex architecture.

On a client computer requests to the server are prepared and executed (of course if a user has sufficient privileges). Request consists of three groups of parameters: forest fuel material characteristics for a concrete forest area, geometry of ignition source, parameters of climatic conditions. At the server result, satisfying to criteria of request, is taken from the knowledge base; if the suitable result is not found then it is calculated according to the computer model ([5]). The information is sent to the User in the unified format. The service installed on a client computer, allows to process information from the knowledge base in the several forms including tabular data, schemes, graphics, density and isolines plots, maps of vector fields and animation ([6]).

The architecture of the software complex consists of the three main parts:

- the calculation module in **Mathematica** kernel of distribution of forest fires on the simplified mathematical models;
- toolkit for filling in and working with a relational database realized in **Mathematica** 6;
- a subsystem of visualization of the results of the calculations, realized on the basis of the specialized module connected to the **Mathematica** system.

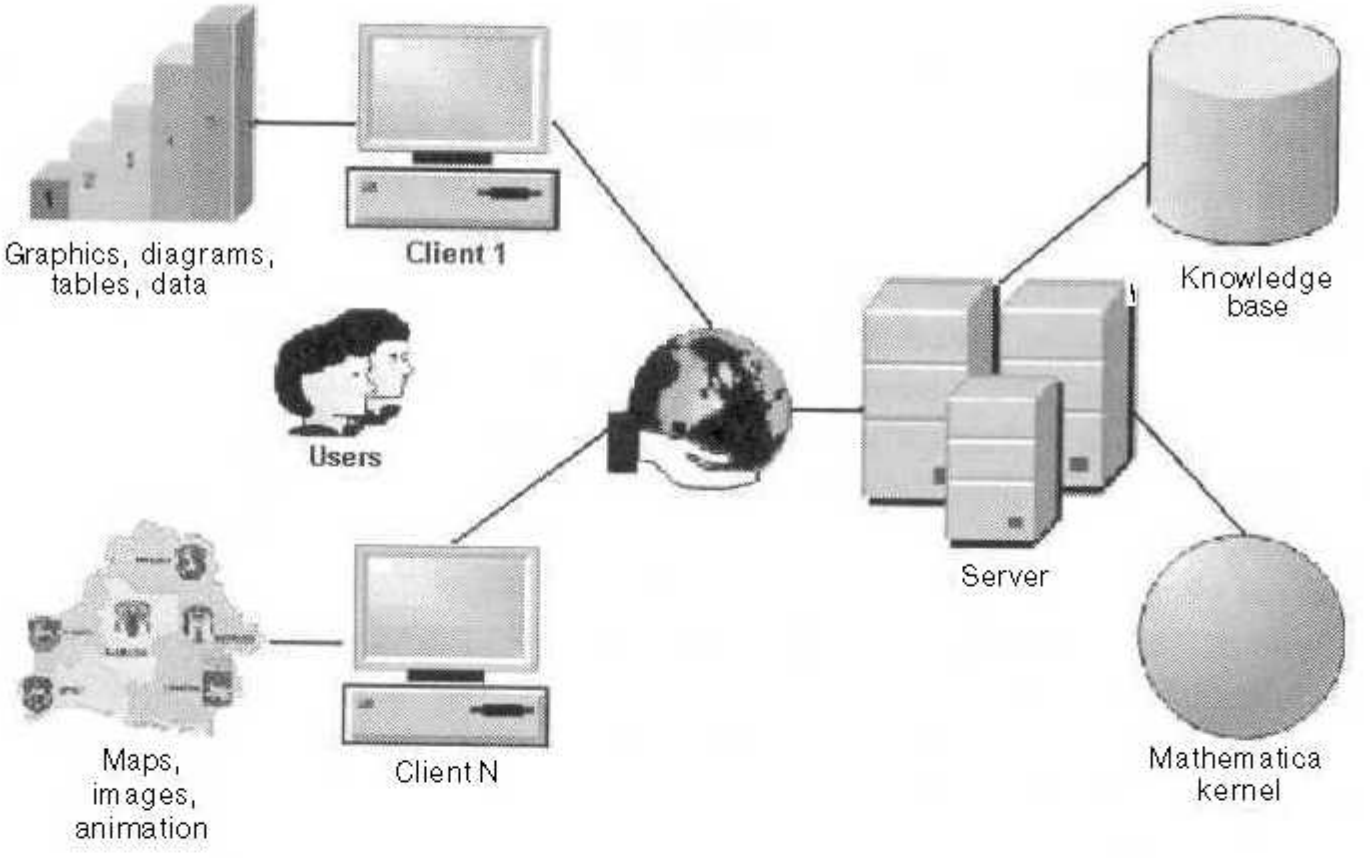

Figure 1: The general scheme of the software complex.

3 The Description of the Module for Filling in and Working with Relational Databases

For storing the results of mathematical modelling of forest fire distribution in different moments of time the specialized organization of a relational database including three tables *(Modelling, Param* and *Data)* is offered as it is shown in Fig. 2.

Table *Modelling* stores the unique identifier *(id)* and the verbal *description* of the concrete calculated result of modelling. Table *Param* stores in a database a list of controllable parameters of modelling, which can be both constants and variables changing in time, can be as a single number so as a vector or even a two or three dimensional table. According to the architecture of system, table *Param* is logically connected with table *Modelling* with cardinality one to many (and this connection is supervised by DBMS), since in the different calculated results of modelling different parameters can be controlled (for example, at large-scale forest fires some microprocesses can not be controlled because of their insignificant influence on a fire as a whole). At last, table *Data* stores concrete numerical values of parameters *(id param* from table *Param)* in different moments of *time.*

For work with a database satisfying the scheme described above a special Mathematica module is developed, containing the following functions:

connect fdatabaseNameJ establishes a connection to a database *databaseName;*

resetModelling[connection, modelling Code, modellingDescription] creates or establishes connection to the calculated result of modelling with a verbal code *modellingCode;*

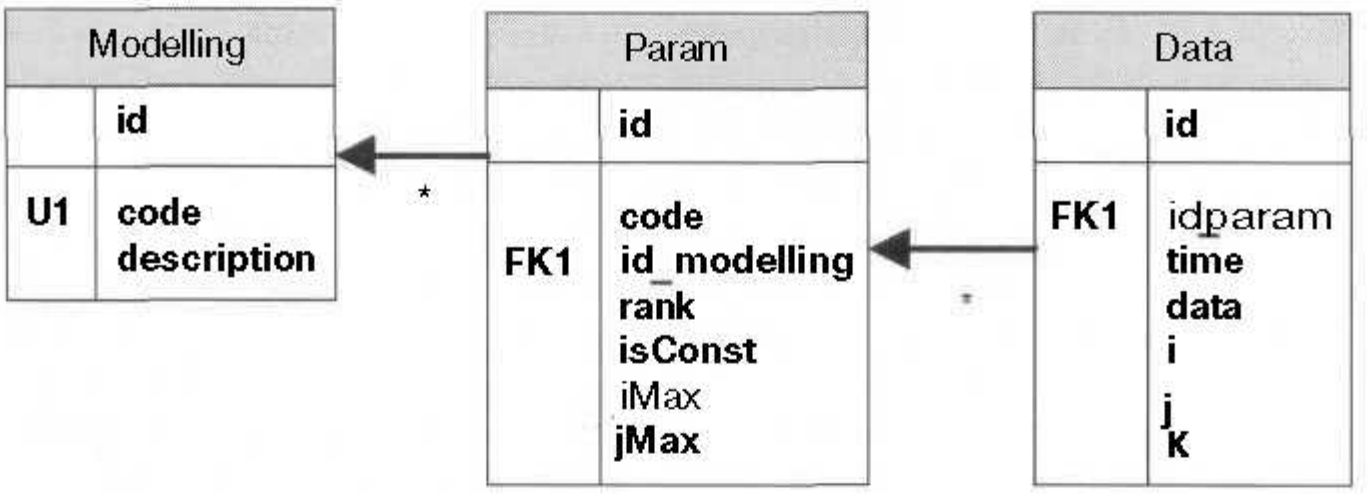

Figure 2: Simplified Entity - Relationship Diagram of the Database for Mathematical Modelling Results.

setValue [connection, idParam, time, value], setValue [connection, idModelling, paramName, isConst, time, value] save in a database the *value* of parameter of modelling at the moment of *time* (or with layer number *time).* Parameter is identified by a name *paramName* or the numerical identifier *id param.* Let's mention that the value is not limited to just a number, it can be a vector or a two or three dimensional table.

Request from database of values at the certain moment of time is carried out by function *getValue [connection, idModelling, paramName, time].*

It is necessary to note universality of the offered scheme of knowledge base organization, the possibility of its application for storage of results of mathematical modelling of various problems. Advantages of the given library are independency of a database type (Oracle, Microsoft SQL Server, Access, MySQL or any other relational database), and also simplicity in use.

4 Library of Specialized Modules for Visualization of Results of Numerical Modelling in System Mathematica

The analysis and generalization of typical mathematical models of hydrodynamics, aeromechanics, heat and mass transfer, geoecology allows us mention requirements and functionality of program service of graphic visualization:

- the two-dimensional and three-dimensional graphics with support of visualization of fronts and borders (for example, external and internal contours, borders of zones of influence, boundaries of phases and layers, fronts of temperature, combustion fronts);
- functions of generating and a drawing of isolines of digital fields of calculated distributions, for example, density, temperatures, pressure of a liquid or gas, concentration of components of impurity, a gas phase, etc.;

- modules of formation and a visualization of vector fields of velocities (each arrow at a corresponding point appears co-directed with a vector of velocity of a stream, and its length is proportional to the strength of velocity).

The listed types of graphic data presentation (described by functions or as tabular data) in **Mathematica** system can be realised, but only in rectangular areas (for 3D - in parallelepipeds). The technique, algorithms, the concrete program realizations are described below, allowing to generate images in the areas limited to curvilinear borders (surfaces). The technique is given further for 2D graphics (on a plane). Generalization on a 3D case (Cartesian coordinates) becomes similar since in **Mathematica** almost all options for 2D and 3D graphics are identical ([4]).

Mathematica system consists of a set of standard visualization tools of a two-dimensional graphics in following approaches: planimetric plots (isolines), density (zones) plots, vector fields, evolution demonstration (animation). Disadvantage of a drawing in **Mathematica** is that displaying occurs in rectangular area. Thus there is no possibility of visualization of *subregions* and *inclusions*. Lets explain these terms.

Subregion - a region (a part of plane), limited with a contour (having no self-crossings) with specially given attributes. The basic sense of introduction of concept of a subregion is caused by features of digital field function approximation, processing in it (namely, rules of integration, differentiation, arithmetic and logic operations with several functions in points of a grid inside a subregion). If input map contains subregions than during calculation and visualization of a digital field, values of functions calculated only in internal points of a grid (except the points which belong to inclusions), and in each subregion calculation is carried out on all set of points of a database having values of the processed parameter. It gives the chance, while approximating of any digital field, to divide all set of initial data into parts / regions, in particular, if there is aprioristic information on independent behavior of restored functions on separate regions.

Inclusion - a region (a part of plane), limited with a contour not having self-crossings, entirely laying inside of a subregion. Any inclusion can have enclosed subregions, which have no crossings with inclusion itself). Inside inclusions calculation of a digital field, formation of isolines, maps of density, vector fields is not made.

Visualization of maps in subregions with curvilinear border. The following technique for visualization of contour and density maps in the one-coherent areas, which are not rectangular, is offered. The user inputs the border region of visualization in the form of the list of coordinates of vertexes of the enclosed polygon (restrictions on convexity of a polygon are not imposed). The roundabout way direction should be such that the visualization area remained on the right. Lets call these coordinates through (xO, yO), (xl, yl), ..., (xn, yn), where xO = xn, yO = yn. Width and height (xmax and ymax accordingly) of the rectangular map generated by standard functions of **Mathematica,** such as ContourGraphics or DensityGraphics are also inputted. Graphics generation in a subregion is made by the algorithm contourCut which returns coordinates of vertexes of two enclosed polygons "framing" area of visualization. After imposing of these polygons by background color

above the usual image of isolines or a density map, formed by standard functions of **Mathematica,** the effect of visualization of the map in a demanded area is reached. Advantage of such technique is that it can be applied to the images (maps) created be standard graphics objects of system **Mathematica.** Thus, it is possible to use any options (properties, settings) of these graphics: scaling, coloring, drawing of coordinate axes, grid lines, their calibration, headings, legends, etc.

The idea of *contourCut* algorithm is in finding of such two points (xstart, ystart) and (xend, yend) on a set of vertexes (xi, yi), i =0, n, that two segments [(0,0), (xstart, ystart)] and [(xmax, ymax), (xend, yend)] are not crossed with the border of area of visualization (obviously, except of the mutual points). Coordinates of vertexes of the first required polygon are the following: (xstart, ystart), (xstart+1, ystart+1), ..., (xend-1, yend-1), (xend, yend), (xmax, ymax), (0, ymax), (0,0), (xstart, ystart). Coordinates of vertexes of the second: (xend, yend), (xend+1, yend+1), ..., (xstart-1, ystart-1), (xstart, ystart), (0,0), (xmax, 0), (xmax, ymax), (xend, yend).

Drawing polygons with background colors above the rectangular map is made directly by function Show or through its option Epilog. Function Line can be used for drawing the area border with a specified color and style. Also using function Show (or its option Epilog) contours, borders, marking lines can be drawn on the generated map in different colors and styles. If in such a way user draws a subregion using background color, he receives a subregion of type "hole" (visualization in multi coherent areas).

In Fig. 3 the example of visualization of isolines and zones in a *subregion* (subregion border is a contour of Brest region of the Republic of Belarus, the legend is not deduced since the digital field is not concretized) is resulted, and except isolines administrative area borders are shown.

Visualization of vector fields. To display a vector fields in areas of a difficult configuration the following method is used. The vector field is formed by one of the standard functions (for example, ListPlotVectorField), further in created object Graphics all vectors (objects Arrows) which do not lay inside of the modelling area (i.e. laying outside of a subregion) are deleted.

For realization of such algorithms of processing of the standard image library *Modelling, m* is developed. Below is the description of the main and optional functions used in the realization of the package, having also independent significance for some other problems of visualization of various types of vector fields.

showBorder [gRaphics, borderList, toShow, oPts] visualizes a map *gRaphics* in area *borderList.* Here: *gRaphics* the graphic object similar to the first parameter of function **Show;** *borderList* the list of coordinates of vertexes of the polygon border of two-dimensional area of visualization which has a format {{x0, y0}, {xl, yl}, ..., {xn, yn}}, where xO = xn, yO = yn; color of a background, color and style of border of area are defined by global variables *backgroundColor, mamB order Color* and *mainBorderStyle; toShow* the optional parameter (can be **True** (by default) or **False);** in case of value **False** visualization of the received map on the screen does not occur, and the output occurs to option **DisplayFunction—Identity;** *oPts* optional parameters options of display which are transferred to function **Show**

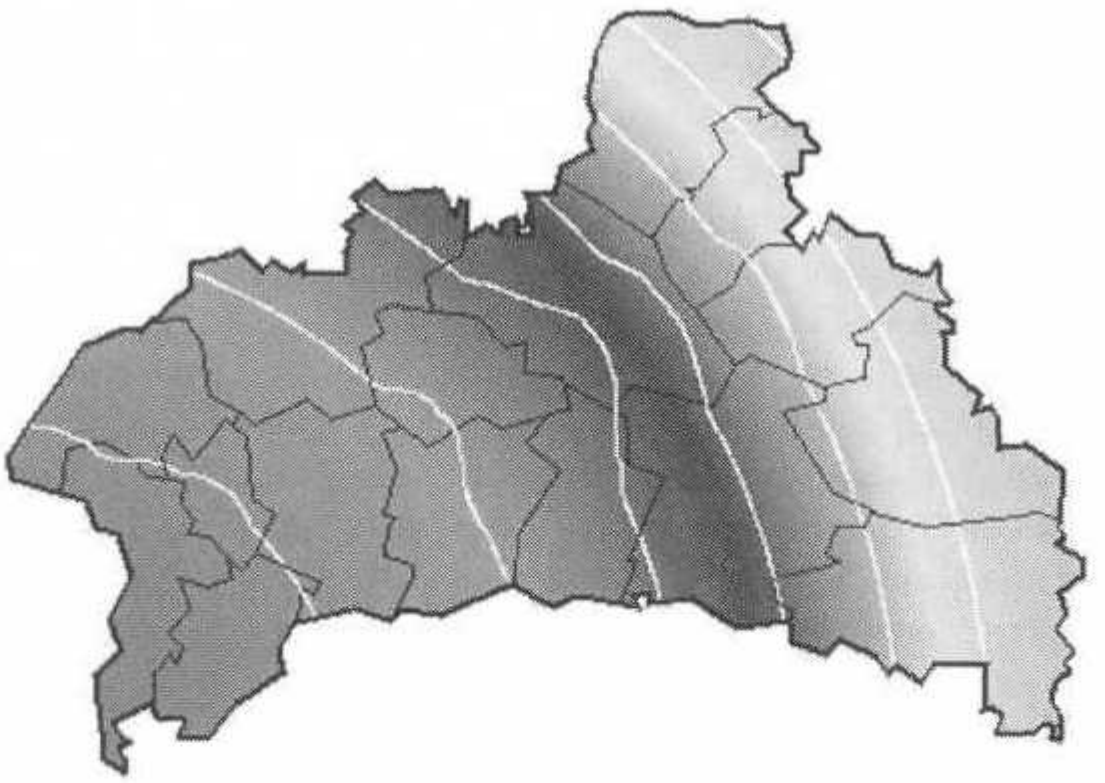

Figure 3: An example of isolines and a density map in a subregion.

during the map generation.

getRegions {{regionlBorder, ...}, {regionlType, ...}, {regionlBackgroundColor, ...}, {regionlBorderStyle, ...}} returns the object of type **Graphics** representing a set of subregions of certain type, color of a background, style and color of border. Here: *{region IB order, region2B order, ...}* the list of subregions; every of which is set by coordinates of vertexes, is similar to parameter *borderList* of function *showBorder; {regionlType, region2Type, ...}* the optional list of types of subregions; if less variables are provided than subregions of the default type *regionTypeSolid* is used; *{regionlBackgroundColor, region2BackgroundColor, ...}* the optional list of colors by which corresponding areas are painted over; if there's less variables than areas than by default global variable *regionBackgroundColor* is used; depending on value *regionNType* the specified color can be ignored; *{regionlBorderStyle, regionEBorderStyle, ...}* the optional list of colors and styles of borders of areas; if there's less variables than areas default values are set by the global variables *regionBorderColor* and *regionBorderStyle.*

Types of subregions (possible values of the second parameter): *regionTypeSolid* (by default) the subregion is filled in with the specified color; *regionTypeHole* the subregion is filled in with color of a background *backgroundColor* even if other color is explicitly specified; *regionTypeNone* the subregion is transparent, i.e. only the border is shown.

In all parameters of this function instructions of braces necessarily, even in case the list consists of one variable.

borderListPlot [listPlot, borderList, toShow, opts] carries out visualization of **DensityListPlot** or **ContourListPlot** in the region of visualization *borderList.* Except for the first parameter, this function is similar to function *showBorder.*

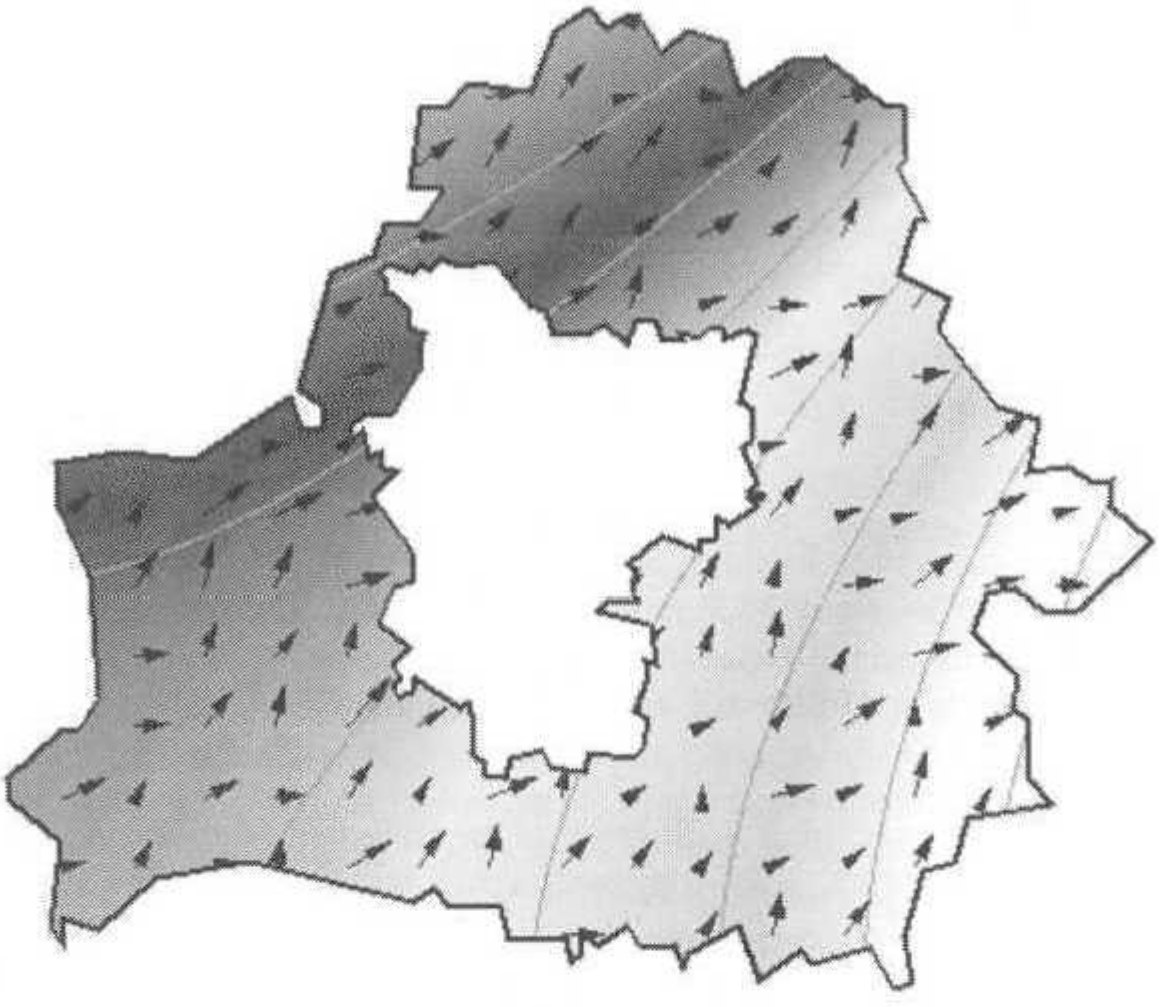

Figure 4: An example of a map of a vector field on a subregion with inclusion.

deleteVectorsOutsideBorder [vectorFieldPlot, borderList] deletes from graphic object *vectorFieldPlot}* all vectors which do not belong to visualization area *borderList.*

The optional functions of library having independent value: *getLineNumber [xO, yO, xl, yl, x, y]* function shows how the point (x, y) is located relatively to a straight line set by points (xO, yO) and (xl, yl). Returned values: equals 0 if the point lays on a straight line; greater than 0 if the point is located on the right; less than 0 if the point is located at the left. *doesSegmentsCross fxO, yO, xl, yl, x2, y2, x3, ySj* function returns whether two segments [(xO, yO), (xl, yl)] and [(x2, y2), (x3, **y3)]** are crossed. Returned values: False if pieces have no mutual points, True, otherwise.

isOutsideOfPolygon [gran, xi, yi, isConvex] function checks, whether the point lays (xi, yi) strictly inside of the enclosed polygon; returns **True** or **False,** *gran* coordinates of vertexes of the enclosed polygon {{xO, yO}, {xl, yl}, ... {xn, yn}}, where xO = xn, yO = yn; *xi, yi* - coordinates of a point to check; *isConvex* the optional parameter accepting values **True** or **False** (by default), whether specified polygon is convex; if **True** specified faster algorithm of checking is used.

The following image shows some possibilities of library *Modelling.m.* In Fig. 4 the example of visualization of a vector field, isolines and zones in a *subregion* with *inclusion* is presented. In the example a subregion schematically represented area of Belarus, inclusion the Minsk region.

References

[1] *Djakonov V.P.* Computer mathematics. J. Sorosovsky educational magazine 7 (2001) 116-121

[2] *Grishin A.M.* General mathematical models of forest and peat fires and their applications. J. Successes of mechanics 4 (2002) 41-89

[3] *Kuleshov A.A.* Mathematical modelling in problems of industrial safety and ecology. J. Information technologies and computing systems 4 (2003) 56-70

[4] *Morozov A.A., Taranchuk V.B.* Programming of problems of the numerical analysis in system Mathematica. BSPU, Minsk (2005)

[5] *Barovik D. V., Korzjuk V.I, Taranchuk V.B.* About computer modelling of fires in client-server architecture of calculations, processings and visualisation of results (Part 1). In: Proceedings of Network Computer Technologies NTECH2007, BSU (2007) 170-176

[6] *Barovik D. V., Taranchuk V.B.* Library of modules of visualisation of scientific data in system Mathematica. J. Informatisation of education 2 (2007) 24-31