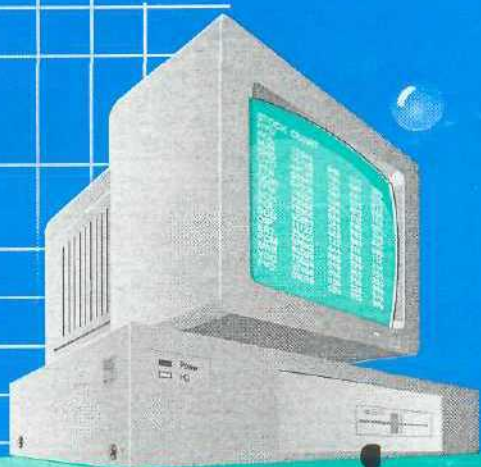


# Информатизация образования

■ 2

■ 2007



*Учредитель –*  
Учреждение «Главный  
информационно-  
аналитический центр  
Министерства образования  
Республики Беларусь»

Свидетельство о регистрации  
№437  
выдано 11.12.2001 г.  
Министерством информации  
Республики Беларусь



220088, г. Минск,  
ул. Захарова, 59

<http://www.giac.unibel.by>

**Ежеквартальный  
научно-методический журнал**  
(приказом ВАК РБ от 18 января 2006 г.  
№ 8 включен в список научных изда-  
ний Республики Беларусь для опубли-  
кования результатов диссертационных  
исследований)

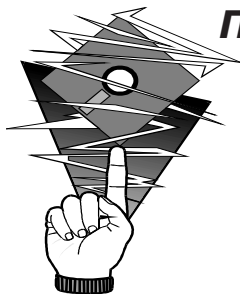
2(47) 2007

# **Информатизация образования**

Издается с IV квартала 1995 года

## *Редакционный совет*

Н.И. Листопад (главный редактор)  
А.Н. Курбацкий (предс. ред. совета)  
С.И. Ладутько (зам. гл. редактора)  
Е.Н. Кишкурно (отв. секретарь)  
М.М. Ковалев  
В.Н. Ярмолик  
М.К. Буза  
А.Н. Морозевич  
Б.Н. Паньшин  
О.И. Тавгень  
И.Ф. Киринович  
В.В. Басько  
М.Г. Зеков  
В.И. Дравица  
С.В. Енин  
Д.И. Пунько  
С.Г. Ершова



## **ПРОГРАММНО-ТЕХНИЧЕСКИЕ СРЕДСТВА**

---

***Д.В. Баровик**, магистрант факультета прикладной математики и информатики Белорусского государственного университета,  
**В.Б. Таранчук**, д.ф-м.н., заведующий кафедрой информационного и программно-математического обеспечения автоматизированных производств Белорусского государственного университета*

### **Библиотека модулей визуализации научных данных в системе Mathematica**

Одним из актуальных направлений реформирования современной системы образования является интеграция информационных и телекоммуникационных технологий в образовательный процесс и в управление образованием. Особая роль при этом отводится принципиально новому конструированию содержания и организации учебного материала, созданию современных учебных сред – баз знаний и соответствующих программных средств, обеспечивающих обучаемых эффективными инструментами познания, самообучения и самоконтроля.

Эффективной программной платформой для решения задач подготовки «электронных уроков и книг с живыми примерами», интерактивных информационно-педагогических сред нового поколения являются системы компьютерной математики [1]. При этом большая нагрузка в создании интерактивных электронных образовательных ресурсов отводится машинной графике, получению необходимых иллюстраций

при разных сценариях работы, и совсем самостоятельную роль имеют задачи визуализации научных данных.

Лидерами систем компьютерной алгебры являются **Mathematica**, **Maple**, **MATLAB**, **MathCAD**, **Simulink** и др. В этих программных комплексах математические выкладки, преобразования и упрощения выражений выполняются после записи уравнений, условий и функций в традиционной математической нотации самими системами. Практика последних лет подтверждает эффективность применения систем интеллектуальных вычислений при решении задач математической физики, механики сплошных сред, экономики и др.

В настоящей работе описаны инструментарий, особенности проектирования и реализации в компьютерной технической системе (КТС) **Mathematica** модулей графической визуализации результатов численных экспериментов. Разработанный графический сервис обеспечивает все традиционные варианты визуализации научных данных.

Анализ и обобщение типичных математических моделей гидродинамики, аэромеханики, тепломассопереноса, геоэкологии позволяют выделить требования и функциональность программного сервиса графической визуализации:

- двумерная и трехмерная графика с поддержкой вывода фронтов и границ (например, внешних и внутренних контуров, границ зон влияния, раздела фаз и слоев, фронтов температуры, горения);
- функции расчета и вывода изолиний цифровых полей рассчитываемых распределений, например, плотности, температуры, давления жидкости или газа, концентраций компонентов примесей, газовой фазы и др.;
- модули формирования и вывода векторных полей скоростей (каждая стрелка в соответствующей точке прорисовывается сонаправленной с вектором скорости потока, а ее длина пропорциональна величине скорости).

Перечисленные виды графического представления данных (описываемых функциями или таблично) в КТС **Mathematica** могут быть реализованы, но в прямоугольных областях (для 3D – в параллелепипедах). Ниже описаны методика, алгоритмы, конкретные программные реализации, позволяющие формировать изображения в областях,

ограниченных криволинейными границами (поверхностями). Изложение дается для 2D графики (на плоскости), обобщение на случай 3D (пространство, декартовы координаты) делается аналогично, т.к. в КТС почти все опции для 2D и 3D графики одинаковы [2].

Система **Mathematica** содержит набор стандартных средств визуализации двухмерной графики в следующих подходах: контурные карты (изолинии), карты плотностей (зоны), векторные поля, демонстрация эволюции (анимация). Особенностью графики в КТС является то, что отображение происходит в прямоугольной области. При этом нет возможности выделения *подобластей* и *включений*. Поясним эти термины.

*Подобласть* – участок (часть плоскости), ограниченный заданным во входной карте (схеме) не имеющим самопересечений контуром (ломаной) со специально для него назначенными атрибутами. Основным смыслом введения понятия подобласти обусловлены особенностями аппроксимации, обработки в ней сеточных функций (а именно, правилами интегрирования, дифференцирования, арифметических и логических операций с несколькими функциями в узлах сетки внутри подобласти). Если входная карта содержит подобласти, то при получении и визуализации цифрового поля значения сеточной функции рассчитываются только во внутренних узлах сетки (кроме узлов, попадающих во включения), причем в каждой подобласти расчет выполняется по всей совокупности опорных точек базы данных со значением обрабатываемого параметра. Это дает возможность при аппроксимации любого цифрового поля разделять на части/участки всю совокупность исходных данных, в частности, если есть априорная информация о независимом поведении восстанавливаемых функций на отдельных участках.

*Включение* – участок (часть плоскости), ограниченный заданным не имеющим самопересечений контуром (ломаной), целиком лежащий внутри подобласти. Внутри любого включения могут находиться вложенные подобласти, которые не должны иметь пересечений с их содержащими включениями. Во включениях расчет цифрового поля, формирование изолиний, карт плотностей, векторных полей не производится.

## Визуализация карт в подобластях с криволинейной границей

Предлагается следующая методика для визуализации контурных карт и карт плотностей в односвязных областях, не являющихся прямоугольными. Пользователь задает границы области визуализации в виде списка координат вершин замкнутого многоугольника (ограничения на выпуклость многоугольника не накладываются). Направление обхода должно быть таким, чтобы при обходе область визуализации оставалась справа. Обозначим эти координаты через  $(x_0, y_0)$ ,  $(x_1, y_1), \dots, (x_n, y_n)$ , где  $x_0 = x_n$ ,  $y_0 = y_n$ . Так же описываются ширина и высота  $(x_{\max}$  и  $y_{\max}$  соответственно) прямоугольной карты, построенной стандартными средствами **Mathematica**, такими как **ContourGraphics** или **DensityGraphics**. Формирование изображения в подобласти обеспечивает алгоритм и соответствующий модуль *contourCut*, который возвращает координаты вершин двух замкнутых многоугольников, «обрамляющих» область визуализации. При наложении этих многоугольников цветом фона на обычное изображение изолиний или карту плотностей, формируемых функциями КТС, достигается эффект визуализации заданной карты в требуемой области. Достоинством такой методики является то, что ее можно применять к изображениям (картам), задаваемым стандартными графическими объектами системы **Mathematica**. При этом, можно использовать любые опции (свойства, настройки) этих графиков: масштабирование, окраска, вывод координатных осей, линий сетки, ее калибровки, надписей, заголовков, легенды и т. п.

Суть алгоритма *contourCut* заключается в нахождении таких двух точек  $(x_{\text{start}}, y_{\text{start}})$  и  $(x_{\text{end}}, y_{\text{end}})$  на множестве вершин  $(x_i, y_i)$ ,  $i=0, 1, \dots, n$ , что два отрезка  $[(0,0), (x_{\text{start}}, y_{\text{start}})]$  и  $[(x_{\max}, y_{\max}), (x_{\text{end}}, y_{\text{end}})]$  не пересекаются с границей области визуализации (очевидно, за исключением общих точек). Координаты вершин первого искомого многоугольника будут следующие:  $(x_{\text{start}}, y_{\text{start}})$ ,  $(x_{\text{start}+1}, y_{\text{start}+1})$ ,  $\dots$ ,  $(x_{\text{end}-1}, y_{\text{end}-1})$ ,  $(x_{\text{end}}, y_{\text{end}})$ ,  $(x_{\max}, y_{\max})$ ,  $(0, y_{\max})$ ,  $(0, 0)$ ,  $(x_{\text{start}}, y_{\text{start}})$ . Координаты вершин второго:  $(x_{\text{end}}, y_{\text{end}})$ ,  $(x_{\text{end}+1}, y_{\text{end}+1})$ ,  $\dots$ ,  $(x_{\text{start}-1}, y_{\text{start}-1})$ ,  $(x_{\text{start}}, y_{\text{start}})$ ,  $(0, 0)$ ,  $(x_{\max}, 0)$ ,  $(x_{\max}, y_{\max})$ ,  $(x_{\text{end}}, y_{\text{end}})$ .

Наложение многоугольников на прямоугольный график цветом фона производится непосредственно функцией **Show** или через ее опцию **Epilog**. Функцией **Line** может быть выведена указываемым стилем и граница области. Также функцией **Show** (или ее опцией **Epilog**) на полученную карту выводятся задаваемыми цветами и стилями всевозможные контура, границы, линии разметки, а если так добавленную подобласть закрасить цветом фона, получим подобласть типа «дырка» (визуализация в многосвязных областях).

На рисунке приведен пример визуализации изолиний и зон в *подобласти* (граница подобласти – контур Брестской области РБ, легенда не выводится, т.к. цифровое поле не конкретизируется), причем кроме изолиний на схеме показаны нанесенные внутренние контуры-границы (в данном примере границы административных районов).

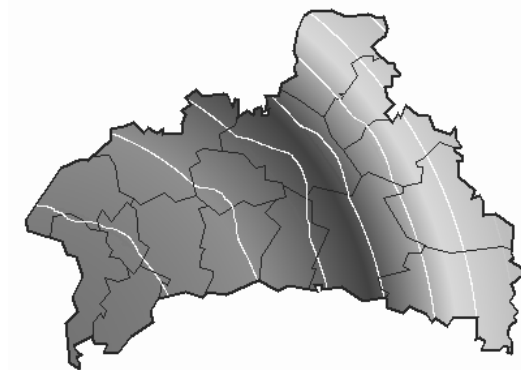


Рис. 1. Пример изолиний и карты плотности по подобласти

### Визуализация векторных полей

Для отображения векторных полей в областях сложной конфигурации используется следующий прием. Векторное поле формируется одной из стандартных функций (например, **ListPlotVectorField**), далее в полученном объекте **Graphics** удаляются все векторы (объекты **Arrows**), которые не попадают в область моделирования (т.е. удаляются все векторы вне подобласти).

Для реализации такого алгоритма обработки стандартного изображения запрограммирована библиотека *Modelling.m*. Приведем описание основных и вспомогательных функций в реализации пакета, имеющих также самостоятельное значение для некоторых других задач визуализации различного типа векторных полей.

Функция *showBorder[gGraphics, borderList, toShow, oPts]* выполняет визуализацию графиков *gGraphics* в области *borderList*. Здесь:

- *gGraphics* – графический объект, аналогичный первому параметру функции *Show*;
- *borderList* – список координат вершин многоугольника, задающего границу (контур) двумерной области визуализации, который имеет формат  $\{\{x_0, y_0\}, \{x_1, y_1\}, \dots, \{x_n, y_n\}\}$ , где  $x_0=x_n$ ,  $y_0=y_n$ ; цвет фона, цвет и стиль границы области определяются глобальными переменными *backgroundColor*, *mainBorderColor* и *mainBorderStyle*;
- *toShow* – необязательный параметр, принимающий значения **True** (по умолчанию) или **False**; в случае значения **False** визуализация полученной карты на экране не происходит, а вывод происходит с опцией **DisplayFunction** → **Identity**;
- *oPts* – необязательные параметры – опции отображения, которые передаются в функцию **Show** при отображении графиков.

Функция *getRegions[{region1Border, ...}, {region1Type, ...}, {region1BackgroundColor, ...}, {region1BorderStyle, ...}]* возвращает объект типа **Graphics**, представляющий собой совокупность подобластей определенного типа, цвета фона, стиля и цвета границы. Здесь:

- $\{region1Border, region2Border, \dots\}$  – список подобластей; каждая задается координатами вершин, аналогично параметру *borderList* функции *showBorder*;
- $\{region1Type, region2Type, \dots\}$  – необязательный список типов подобластей. Если переменных в этом списке меньше, чем подобластей, то они будут иметь тип по умолчанию *regionTypeSolid*;
- $\{region1BackgroundColor, region2BackgroundColor, \dots\}$  – необязательный список цветов, которыми закрашиваются соответствующие области. Если переменных в этом списке меньше,



чем областей, то они будут иметь значение по умолчанию, определяемое глобальной переменной *regionBackgroundColor*. В зависимости от значения *regionNType* указанный цвет может игнорироваться;

- {*region1BorderStyle*, *region2BorderStyle*, ...} – необязательный список цветов и стилей границ областей. Если переменных в этом списке меньше, чем областей, то они будут иметь значения по умолчанию, задаваемые глобальными переменными *regionBorderColor* и *regionBorderStyle*.

Типы подобластей (возможные значения второго параметра):

- *regionTypeSolid* (по умолчанию) – подобласть заливается указанным цветом;
- *regionTypeHole* – подобласть заливается цветом фона *backgroundColor*, даже если явно указан другой цвет;
- *regionTypeNone* – подобласть является прозрачной, т.е. выводится только граница.

Во всех параметрах этой функции указание фигурных скобок обязательно, даже в случае, если список состоит из одной переменной.

Функция *borderListPlot*[*listPlot*, *borderList*, *toShow*, *opts*] выполняет визуализацию графиков **DensityListPlot** или **ContourListPlot** в области визуализации *borderList*. За исключением первого параметра, эта функция аналогична функции *showBorder*.

Функция *deleteVectorsOutsideBorder*[*vectorFieldPlot*, *borderList*] удаляет из графического объекта *vectorFieldPlot*, все векторы, которые не попадают в область визуализации *borderList*.

Вспомогательные функции библиотеки, имеющие самостоятельное значение:

- *GetLineNumber*[ $x_0$ ,  $y_0$ ,  $x_1$ ,  $y_1$ ,  $x$ ,  $y$ ] – функция показывает, как расположена точка  $(x, y)$  относительно прямой, задаваемой точками  $(x_0, y_0)$  и  $(x_1, y_1)$ . Возвращаемые значения: 0, если точка лежит на прямой; > 0, если точка расположена справа; < 0, если точка расположена слева.

- *DoesSegmentsCross*[ $x_0$ ,  $y_0$ ,  $x_1$ ,  $y_1$ ,  $x_2$ ,  $y_2$ ,  $x_3$ ,  $y_3$ ] – функция, которая используется, чтобы показать, пересекаются ли два отрезка:  $[(x_0, y_0), (x_1, y_1)]$  и  $[(x_2, y_2), (x_3, y_3)]$ . Возвращаемые значения: **False**, если отрезки не имеют общих точек, **True**, в противном случае.

- $IsOutsideOfPolygon[gran, x_p, y_p, isConvex]$  – функция проверяет, лежит ли точка  $(x_i, y_i)$  строго внутри замкнутого многоугольника; возвращает **True** или **False**.  $gran$  – координаты вершин замкнутого многоугольника  $\{\{x_0, y_0\}, \{x_1, y_1\}, \dots, \{x_n, y_n\}\}$ , где  $x_0 = x_n, y_0 = y_n$ ;  $x_i, y_i$  – координаты проверяемой точки;  $isConvex$  – необязательный параметр, принимающий значения **True** или **False** (по умолчанию), указывающий является ли многоугольник выпуклым. Если указано **True**, то используется более быстрый алгоритм проверки.

Следующее изображение демонстрирует некоторые возможности библиотеки *Modelling.m*. На рисунке приведен пример визуализации векторного поля, изолиний и зон в подобласти с включением. В примере подобласть – схематично изображенная площадь Республики Беларусь, включение – Минская область.

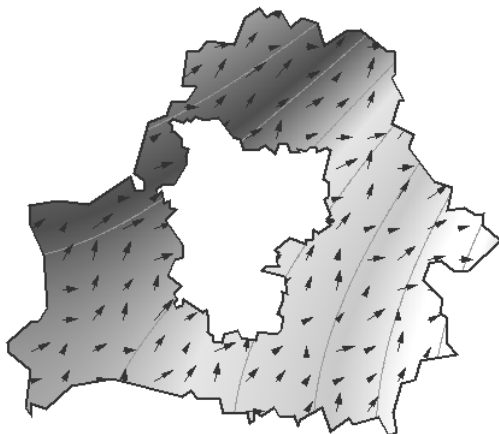


Рис. 2. Пример карты векторного поля по подобласти с включением

## Литература

1. Дьяконов В.П. Компьютерная математика // Соросовский образовательный журнал. – 2001. – том 7, № 1. – С. 116-121.
2. Морозов А.А., Таранчук В.Б. Программирование задач численного анализа в системе Mathematica: Учеб. пособие. – Мн.: БГПУ, 2005. – 145 с.