

## БАЗЫ ДАННЫХ РЕЗУЛЬТАТОВ ЧИСЛЕННОГО МОДЕЛИРОВАНИЯ (НА ПРИМЕРЕ ЗАДАЧИ РАСПРОСТРАНЕНИЯ ЛЕСНЫХ ПОЖАРОВ)

We offer an organization scheme of mathematical modelling numerical results database. Technological aspects of programming Wolfram Mathematica's module for database use are described. We discuss the features and scenarios of using the module at the example of computer simulation of forest fire spread dynamics.

Математическое моделирование является основным инструментом для исследования, прогнозирования и управления сложными нестационарными технологическими, гидродинамическими, биологическими, экологическими, финансовыми и другими процессами, происходящими в природе и обществе. Вычислительный эксперимент предполагает, что после построения математической модели проводится ее численное исследование, позволяющее «проиграть» поведение объекта в различных условиях или в различных модификациях [1].

Для вычислительного эксперимента на ЭВМ математическая модель преобразовывается из языка математических терминов (уравнений) в компьютерную программу. При исследовании математических задач, которые не имеют аналитических решений, применяются численные методы. Многие математические модели не имеют эталонного решения, по которому можно было бы определить точность (сходимость, аппроксимацию) применяемых численных схем. Одним из способов проверки является расчет модельных задач, результаты которых интуитивно понятны, а также сопоставление решений, полученных на различных сетках.

При обработке результатов вычислительных экспериментов (и для отладки применяемой численной схемы) требуется сравнение результатов расчетов, полученных с использованием различных аппроксимаций, алгоритмов, шагов сетки, определяющих параметров задачи. Для некоторых классов задач на получение каждого результата моделирования затрачивается достаточно продолжительное время даже на компьютерах с высоким быстродействием. Расчеты по таким моделям в масштабе реального времени невозможны. Другой подход заключается в формировании базы данных результатов математического моделирования и создания удобного инструментария для работы с ней. Тогда проведение повторных расчетов по трудоемким с вычислительной точки зрения алгоритмам в реальном времени не требуется – результаты извлекаются из уже наполненной базы данных для последующего сопоставления, обработки и предоставления пользователям [2].

Следует отметить, что у предлагаемого способа решения имеются существенные ограничения, связанные с невозможностью создания базы данных абсолютно всех вариантов моделирования задачи при большом количестве входных переменных. Первоначально эксперты в конкретной предметной области должны указать характерные, наиболее важные для конечных пользователей параметры, а также требуемые сценарии работы с полученными данными. Понятно, что в режиме реального времени пользователи смогут работать только с содержащимися в базе результатами. В дальнейшем база данных может пополняться на основании анализа запросов пользователей, для которых в базе отсутствовал готовый (рассчитанный) результат. Такой подход обеспечивает обратную связь от пользователей к экспертам-предметникам для наполнения базы наиболее востребованными случаями. Значительно снижаются требования к мощности компьютеров конечных пользователей и скорости моделирования, так как задача сводится к созданию механизма поиска и вывода уже накопленных данных [3, 4].

Практика показывает эффективность применения систем компьютерной алгебры (*Mathematica*, *Maple*, *MATLAB* и др.) при математическом моделировании. В данной работе предлагается обобщенная схема базы данных. Для удобства работы с ней используется модуль, разработанный на основе технологии *DatabaseConnectivity* системы *Wolfram Mathematica 7*. Описываются особенности и типовые сценарии на примере компьютерного моделирования динамики распространения лесного пожара.

Схема базы данных результатов математического моделирования. В контексте настоящей работы рассматриваются базы данных результатов математического моделирования. Для сохранения получаемых массивов чисел предлагается схема (рис. 1), состоящая из трех таблиц: **Modelling**, **Param**, **Data** и перечня названий столбцов каждой из них. В строках таблицы **Modelling** записываются уникальный

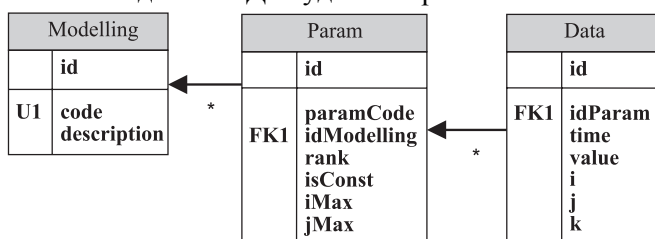


Рис. 1. Обобщенная схема базы данных для хранения численных результатов вычислительных экспериментов

Для сохранения получаемых массивов чисел предлагается схема (рис. 1), состоящая из трех таблиц: **Modelling**, **Param**, **Data** и перечня названий столбцов каждой из них. В строках таблицы **Modelling** записываются уникальный

идентификатор (*id*), словесный код (*code*) и описание (*description*) конкретного рассчитанного результата моделирования. Таким образом, каждому конкретному вычислительному эксперименту будет сопоставлена одна строка таблицы **Modelling**, а следовательно, и уникальный идентификатор-число. Этот идентификатор используется при заполнении столбца *idModelling* таблицы **Param**.

Таблица **Param** позволяет унифицировать и классифицировать все контролируемые, сохраняемые в базе параметры моделирования, которые могут быть как константами, так и изменяемыми во времени величинами и задаваться как одним числом, так и массивом, двух- и трехмерной таблицей. Опишем столбцы таблицы **Param** на примере задачи распространения верхового лесного пожара, укажем какие числовые данные, массивы следует хранить, извлекать, обрабатывать, визуализировать.

При построении компьютерной модели процесса предполагается, что известны скорость и направление ветра, температура окружающей среды, геометрические, структурные и реакционные свойства полога леса, очаги воспламенения. Лес рассматривается как пористо-дисперсная пространственно неоднородная пятифазная среда, включающая сухое органическое вещество, воду в жидко-капельном состоянии, коксик, золу и газовую фазу. Газовая фаза состоит из кислорода, горючих компонентов продуктов пиролиза, инертных компонентов воздуха, а также водяного пара и инертных продуктов горения. Однотемпературная математическая модель верхового пожара может быть записана в виде системы дифференциальных уравнений в частных производных [5–7]. В таблице приведены несколько строк из **Param**.

Параметры модели лесных пожаров, хранящиеся в таблице **Param**

id	paramCode	idModelling	rank	isConst	iMax	jMax
1	Tijk	1	3	False	N	M
2	E1	1	0	True	NULL	NULL
3	spruce_humidity	1	2	True	12	NULL
4	Xi_forest_gran	1	1	True	NULL	NULL
5	Yj_forest_gran	1	1	True	NULL	NULL
...	...	...	...	...	...	...

Согласно записям таблицы, в базе сохраняется информация о температуре ( $id = 1$ ), которой присвоен код *Tijk*. Видно, что это динамическая (изменяемая во времени) величина ( $isConst = False$ ), представляющая собой трехмерный ( $rank = 3$ ) куб данных, ширина и высота которого равны  $iMax = N$  и  $jMax = M$  соответственно.

Во второй строке описывается константа ( $paramCode = E1$ ) – энергия активации физико-химической реакции пиролиза древесины. В третьей строке приводятся статистические данные о влагосодержании произрастающей на территории Беларуси сосны обыкновенной в зависимости от времени года ( $iMax = 12$  месяцев) и возраста дерева. Четвертая и пятая строки описывают векторы значений  $x$  и  $y$  – координат границы области лесного массива.

Следует отметить универсальность предложенной схемы организации базы данных, возможность ее применения для хранения результатов компьютерного моделирования любых задач математической физики. Так, если будет решаться задача распространения лесного пожара в двухмерном случае, то единственное изменение в приведенных записях коснется значения размерности температуры –  $rank$  станет равен двум.

Таблица **Param** логически связана (и эта связь контролируется СУБД) с кратностью «один ко многим» с таблицей **Modelling**, так как в различных вычислительных экспериментах могут контролироваться разные параметры (при крупномасштабных пожарах, например, могут вообще не учитываться некоторые микропроцессы из-за их незначительного влияния на пожар в целом).

Наконец, таблица **Data** содержит конкретные числовые значения параметров  $idParam$ , описанных в таблице **Param**, колонка *time* – момент времени или временного слоя, а колонки  $i, j, k$  – индексы (координаты) элементов многомерных величин (сеточных функций). Непосредственно сами числовые значения хранятся в столбце *value*.

Контроль за правильностью индексов при заполнении таблицы **Data** является важным вопросом с точки зрения целостности базы данных. К примеру, нельзя допустить присутствия в таблице **Data** двух записей об одной и той же величине, имеющих одинаковые индексы  $i, j, k$  и *time*. Однако проведение такого контроля при каждом изменении данных в таблице **Data** средствами СУБД может привести к снижению производительности, особенно при больших объемах числовых данных. Поэтому корректность заполнения индексов реализована в функциях по наполнению базы данных

описанного далее модуля. Предполагается, что изменение данных другими средствами не осуществляется.

Заметим, что используемая схема базы данных может быть реализована в любой реляционной СУБД (*MS Access, MySQL, MS SQL Server, Oracle* и др.).

**Модуль работы с базами данных.** Разработан подключаемый к системе *Wolfram Mathematica* модуль, обеспечивающий удобную работу с базами данных, построенными на основе предложенной нами схемы.

Опишем интерфейс и назначение основных функций.

*connect[databaseName]* выполняет подключение к источнику ODBC с именем *databaseName*; для отключения от базы данных используется функция *disconnect[]*. В теле данных функций ключевыми командами являются функции **OpenSQLConnection** и **CloseSQLConnection** стандартного пакета **DatabaseLink** системы *Wolfram Mathematica 7*.

*resetModelling[connection, code, description]* создает или выполняет подключение к рассчитанному результату моделирования со словесным кодом *code* (из таблицы **Modelling**), здесь *connection* – это результат выполнения описанной функции *connect*. Она возвращает уникальный идентификатор *id* из таблицы **Modelling** по введенному коду *code*. Если такой записи не существует, то она будет создана.

*deleteModelling[connection, idModelling]* удаляет результат моделирования.

*setValue[connection, idModelling, paramCode, isConst, time, value]* и *setValue[connection, idParam, time, value]* сохраняют в базе данных значение *value* параметра моделирования в момент времени (или с номером временного слоя) *time*, задаваемого именем *paramCode* или числовым идентификатором *idParam*. Заметим, что вид *value* не ограничивается только числом, а может быть и вектором, и двух- и трехмерной таблицей. Для примера приводим фрагмент программного кода данной процедуры.

```
setValue[connection_, idParam_, time_, value_] :=
Block[{arrayDepth = ArrayDepth[value], iMax, jMax, kMax },
(*... сохранение значения двухмерной сеточной функции ...*)
If[2 == arrayDepth, iMax = Length[value]; jMax = Length[value][[1]]];
For [j = 1, j <= jMax, j++, For[i = 1, i <= iMax, i++,
SQLInsert[connection, "Data", {"idParam", "[time]", "value", "i", "j"}, {ToString[idParam], time,
N[value[[i, j]]], ToString[i], ToString[j]}] (* ... *)];
```

Отметим некоторые аспекты реализации данной процедуры. Функция **ArrayDepth** ядра *Mathematica* позволяет определить размерность переданной величины. Сохранение значений в базе данных выполняется командой **SQLInsert** посредством SQL-запроса. Аналогичного эффекта можно добиться, используя более универсальную команду **SQLExecute**.

*getValue[connection, idModelling, paramCode, time]* или *getValue[connection, idParam, time]* извлекают из базы данных значение переменной, описываемой кодом *paramCode* или идентификатором *idParam*, в момент времени *time*. Заметим, что интерфейс данной функции не зависит от того, значение какой величины требуется получить. Данная функция находит необходимую информацию о размерности запрашиваемой величины в таблице **Param** и возвращает данные в соответствующем виде. Для поиска и извлечения информации из базы данных используются функции **SQLSelect** и **SQLExecute**.

Кроме уже описанных часто употребляемых функций реализованный модуль содержит и специфические, такие как, например, функция *getValueSectionJ[connection, idModelling, paramName, time, jSection]*, возвращающая не полный набор данных запрашиваемой многомерной величины, а только некоторое сечение. Данная функция оказывается полезной для сравнения двух решений, например, задачи распространения верхового лесного пожара по направлению ветра в двухмерном и одномерном случаях. Используя эту функцию, из двумерного случая можно извлечь одномерные сечения профилей решения, располагающиеся на разных расстояниях от очага пожара, визуализировать их на одном и том же графике и изучать зависимость влияния моделируемых неоднородностей на скорость и характеристики распространения пожара.

Отметим достоинства реализованного модуля. Его пользователям не требуются знания в области баз данных и экспертных систем. Более того, способ хранения данных вообще скрыт от пользовате-

ля. Все необходимые операции пополнения, поиска, извлечения из базы данных выполняют функции данного модуля. Для подключения к базе достаточно указать название источника данных и словесный код моделирования. Для сохранения и извлечения данных используются простые функции, в которые передаются требуемые переменные моделирования и моменты времени (в случае необходимости), при этом автоматически определяются размерности переменных, выполняются указанные пользователем действия.

Ранее нами описана схема хранения данных на основе стандартных функций СУБД. Сценарии работы с данными и механизмы их вывода пользователю зависят от конкретной задачи. Опишем применение модуля при исследовании предельных условий распространения лесных пожаров в неоднородных средах.

**О сценариях использования модуля.** Модуль позволяет автоматизировать проведение серий вычислительных экспериментов с изменением параметров моделирования в заданных диапазонах. Появляется возможность прерывать длительные по времени процессы моделирования и возобновлять их, начиная с произвольного сохраненного в базе временного слоя. Существует возможность уточнения результата моделирования некоторого интервала (между двумя сохраненными временными слоями) без необходимости пересчета всей задачи.

Модуль оказывается полезным при установлении фактов сходимости, аппроксимации и устойчивости численных схем. С его помощью нетрудно произвести расчеты при грубом и мелком шагах сетки, извлечь полученные результаты, сравнить их, визуализировать в динамике процессы одновременно на одном и том же графике [2, 8].

Для иллюстрации некоторых возможностей модуля приведем фрагменты анализа и визуализации результатов вычислительных экспериментов, когда рассматривается упомянутый процесс распространения верхового пожара в неоднородной среде в двухмерном случае. Фронт пожара распространяется по направлению ветра в виде «стены огня». На его пути создается лесопожарный разрыв (участок леса, очищенный от горючих материалов) шириной 5 м (прямоугольная область, отмеченная штриховой линией на рис. 2). В результате моделирования фронт пожара разрывается, и далее по направлению ветра распространяется только его фрагмент. Расчеты для экономии машинного времени проводились на двух разных сетках. Грубая сетка (по времени) использовалась до и после прохождения участка лесопожарного разрыва. На участке с неоднородными лесными горючими материалами, влияющими на условия устойчивости численной схемы, использовался более мелкий шаг [7, 9].

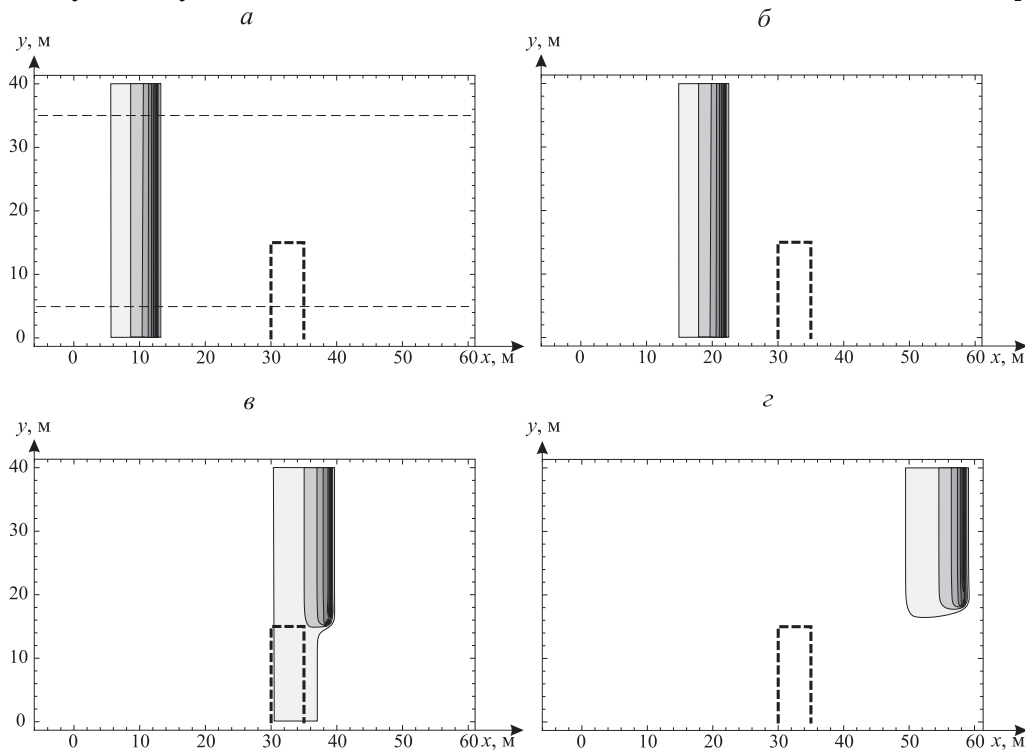


Рис. 2. Карты изолиний температуры во фронте верхового пожара (вид сверху) в случае неоднородного распределения лесных горючих материалов в моменты времени: *a* – 1,5, *б* – 3, *в* – 6, *г* – 9 с

Приведем фрагмент извлечения из базы данных нескольких временных слоев распределений температуры и их визуализации в динамике.

```
conn = connect["forest_fires_KB"]; idModelling = resetModelling[conn, "razriv_5m"];
gran_x = getValue[conn, idModelling, "gran_razriv_x", 0];
gran_y = getValue[conn, idModelling, "gran_razriv_y", 0];
gran_len = Length[gran_razriv_x];
gran_razriv = Table[{gran_x[[i]], gran_y[[i]]}, {i, 1, gran_len}];
Table[Rasterize[ListContourPlot[getValue[conn, idModelling, "Tij", i],
Epilog[{{Dashed, Thick, Black, Line[gran_razriv]}}, {i, 1, stepN}]; ListAnimate[%]
```

В данном случае происходит подключение функциями *connect* и *resetModelling* к конкретному результату моделирования распространения верхового пожара. Извлекается граница области лесопожарного разрыва функцией *getValue* в виде списков  $x$  и  $y$  координат узлов многоугольника. Строится карта изолиний (с использованием стандартной функции **ListContourPlot**) распределения температуры в различные моменты времени. Изображение растеризуется (командой **Rasterize**) для вывода на растровое устройство. Граница лесопожарного разрыва выводится поверх полученного графика опцией **Epilog**. Анимация графиков визуализируется функцией **ListAnimate** (см. рис. 2).

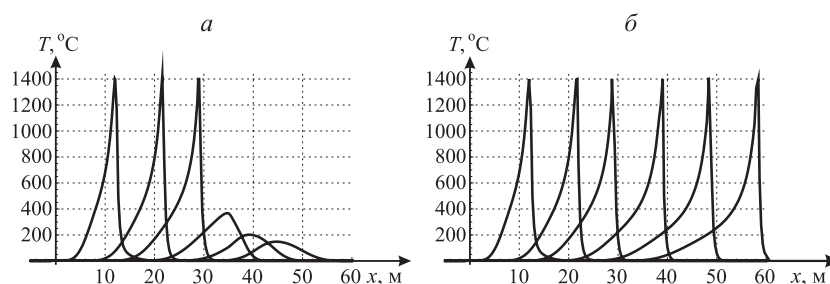


Рис. 3. Профили распределения температуры в моменты времени: 1,5; 3; 4,5; 6; 7,5 и 9 с. *a* – сечение сделано при  $y = 5$  м, *б* – при  $y = 35$  м

Использование базы данных позволяет изучать результаты вычислительных экспериментов, получать различные выборки. Например, на рис. 3 показаны извлеченные из базы данных графики решений на выбранных профилях распределения температуры (места сечений указаны горизонтальными штриховыми линиями на двумерной карте контуров рис. 2).

1. Самарский А. А. // Вестн. АН СССР. 1979. № 5. С. 38.
2. Barovik D. V., Taranchuk V. B. // The Fifth International Workshop: Computer Algebra Systems in Teaching and Research (CASTR). Poland. 2009. Vol. Mathematical Modeling and Mathematical Physics. P. 7.
3. Джарратано Дж., Райли Г. Экспертные системы. Принципы разработки и программирование. М., 2007.
4. Огнев И. В., Борисов В. В. Ассоциативные среды. М., 2000.
5. Гришин А. М. Математическое моделирование лесных пожаров и новые способы борьбы с ними. Новосибирск, 1992.
6. Баровик Д. В., Корзюк В. И., Таранчук В. Б. // Сетевые компьютерные технологии: Сб. тр. III Междунар. науч. конф. Мн., 2007. С. 170.
7. Баровик Д. В., Таранчук В. Б. // Вестн. БГУ. Сер. 1. 2010. № 1. С. 138.
8. Баровик Д. В., Таранчук В. Б. // Информатизация образования. 2007. № 2. С. 24.
9. Barovik D. V., Taranchuk V. B. // Math. Model. Anal. 2010. Vol. 15. № 2. P. 161.

Поступила в редакцию 15.01.10.

**Дмитрий Валентинович Баровик** – аспирант кафедры информационного и программно-математического обеспечения автоматизированных производств. Научный руководитель – доктор физико-математических наук, профессор, заведующий кафедрой информационного и программно-математического обеспечения автоматизированных производств В. Б. Таранчук.