

АВТОМАТИЗАЦИЯ МОДЕЛИРОВАНИЯ СЛОЖНЫХ СИСТЕМ В УСЛОВИЯХ ВАРЬИРОВАНИЯ СТРУКТУРЫ МОДЕЛИ

Куракин П. В.

Институт проблем управления им. В.А. Трапезникова РАН
pvkurakin@yandex.ru

Аннотация: Рассматривается необходимость варьирования структуры математической модели сложного взаимодействия факторов в экономике и возможности разработки соответствующего программного обеспечения.

Ключевые слова: математическое моделирование, автоматизация моделирования, варьирование структуры математической модели.

Введение

Модели сложного взаимодействия факторов в экономике, в особенности, так называемые рефлексивные модели [1 - 3], во многом оказались математическими объектами качественно новой природы, существенно отличаясь от традиционных задач математического моделирования. Принципиальные отличия появляются в ситуации, когда надо получить семейство решений. Например, если речь идет о задачах численного интегрирования дифференциальных уравнений (один из наиболее распространенных типов эволюционных моделей), то потребителя такой модели (таким потребителем может быть как сам исследователь, так и сторонний заказчик) может интересовать так называемый пучок траекторий. Действительно, одиночное решение может представляться недостаточно информативным. Часто важно понимать поведение исследуемой системы хотя бы в некоторой, пусть небольшой, окрестности пространства переменных. С точки зрения вычислений, нахождение такого пучка может оказаться довольно затратной операцией, но принципиально это простая задача. Мы просто разбиваем область начальных данных или граничных условий на дискретный ряд точек, более или менее плотный. Далее, для каждого начального или граничного условия, т.е. для того или иного подмножества полученной системы точек, мы численно решаем наше дифференциальное уравнение. Таким образом, мы получаем семейство или пучок решений в какой-то области.

Однако существуют модели такого типа, в которых грань между модификацией начальных данных и модификацией самой модели становится заметно более размытой. В рефлексивных

моделях многофакторного взаимодействия рассматривается система из некоторого количества факторов социально – экономической природы (как правило, применительно к некоторой территории), которые характеризуются некоторым взаимно – перекрестным влиянием друг на друга. Это влияние количественно описывается экспертными оценками влияния одних факторов на другие.

Анализ показал, что для полноты понимания роли тех или иных факторов недостаточно варьировать величину связей между ними – необходимо допустить переменную структуру самих моделей. В данном случае это означает наряду с варьированием численных величин воздействия факторов друг на друга необходимо полностью «выключать» одни факторы и добавлять другие.

1 Общая постановка задачи

В работах [1-3] развит математический аппарат описания многофакторного взаимодействия с точки зрения подхода, основанного на экспертных оценках. Пусть состояние системы описывается действительным вектором показателей \mathbf{X} . Начальные приращения показателей Δx_i приводят к первичной реакции $\Delta y = \mathbf{A} \Delta x$ (1), где коэффициенты a_{ij} матрицы \mathbf{A} заданы экспертами, они означают оценку сверху прироста фактора i , вызываемого приращением фактора j .

Рефлексивный подход ставит задачу дальше: матрица \mathbf{A} рассматривается как выразитель лишь первичных возмущений факторов. В реальной жизни мы имеем дело с ситуацией, когда несколько факторов одновременно действуют на другой фактор, а также имеет место развитие процесса во времени и последовательное воздействие факторов друг на друга. В совокупности это означает, что эффективно имеет место некое «полное» воздействие факторов друг на друга, а также, при наличии замкнутых цепочек взаимодействия, воздействие факторов на самих себя. Полное (или «системное») воздействие факторов друг на друга описывается матрицей транзитивных замыканий \mathbf{B} , которая описывается условием $\Delta s = \mathbf{B} \Delta x$, где $\Delta s = \mathbf{A}(\Delta s + \Delta x)$ для любого Δx , то есть Δs обладает устойчивостью по отношению к первичному приращению Δx - это «системная реакция». Матрица \mathbf{B} , как можно показать, может быть представлена в виде разложения:

$$\mathbf{B} = \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \mathbf{A}^4 + \dots$$

Далее, линейная модель заменяется на нелинейную, которая использует правила дискретного типа, которые можно отнести к правилам многозначной логики. Взаимодействия факторов предлагается оценивать положительными и отрицательными числами, в балльной системе на некотором интервале $[a_{\min}, a_{\max}]$. Будем считать, что в этом же диапазоне находятся допустимые значения самих показателей (факторов). Результат независимого примитивного воздействия фактора i на фактор j описывается формулой специфического логического умножения:

$$y_{ij} = a_{ij} \otimes x_j = \min(|a_{ij}|, |x_j|) \text{ sign}(a_{ij} x_j). \quad (1)$$

В этой формуле под x_j следует понимать текущее (на данном шаге моделирования) балльное значение показателя x , a_{ij} - балльное значение одиночного воздействия фактора j на фактор i , y_{ij} - результат воздействия, то есть, новое значение фактора x .

Однако в реальности мы всегда имеем ситуацию, когда несколько факторов влияют на данный. В этом случае необходимо различать ситуации так называемых *взаимно дополнительного* и *взаимно компенсирующего* действия факторов.

При взаимно дополнительном воздействии группы факторов g на фактор i для результата требуется воздействие всех факторов группы. Тогда результат определяется как минимальный из всех возможных одиночных воздействий, входящих в группу:

$$y_{ig} = \min_j \{y_{ij}\} \min_{i,j} (\text{sign}(y_{ij})), j \in g. \quad (2)$$

При взаимно компенсирующем воздействии факторов j и k на i , когда для результата достаточно любого из действующих факторов, представляется формулой логической суммы

$$y_{ij} \oplus y_{ik} = \max(|y_{ij}|, |y_{ik}|) \text{ sign}(y_{ij} + y_{ik} \pm \varepsilon) \quad (3),$$

где $0 < \varepsilon < 1$ – добавка, позволяющая оценить разброс результатов в результате вычислительной неоднозначности операции sign .

В случае нелинейной (дискретной) арифметики так же, как и в линейном случае, ставится задача нахождения полных (системных) связей между факторами, т.е. вычисления матрицы транзитивных замыканий \mathbf{B} . Алгоритмические процедуры для этого разработаны в [1 - 3].

В данной работе рассматривается другой подход к исследованию моделей описанного класса. Если в [1-3] исследовались свойства модели с точки зрения получения матрицы транзитивных замыканий. Предполагается, что необходимо непосредственно вычислять и, по возможности, визуализировать разными способами результат эволюции описанной системы в модельном времени (связь модельного и физического времени рассматривается отдельно). Иными словами, уравнения (1) – (3) рассматриваются как динамическая система специального вида.

В любом случае, получение матрицы транзитивных замыканий или эволюционных траекторий системы (1) – (3) не завершает работу с моделью. Далее требуется проведение процедур *верификации* модели, которые включают:

- анализ чувствительности оценки конкретной системной связи к задаваемым вариациям оценок примитивных связей;
- поиск системной связей, зависящих от конкретной примитивной связи для заданного числа рефлексий;
- поиск примитивных связей, влияющих на данную системную связь за заданное число рефлексий.

Технически, перечисленные методы верификации требуют вариации заданных оценок взаимного первичного влияния факторов. Однако, в каждом случае эти вариации желательно проводить не бессистемно. Исследователь, скорее всего, будет придерживаться некоторой стратегии, и при этом ему необходимо как-то фиксировать результаты своих вычислительных экспериментов, чтобы впоследствии иметь возможность удобно работать со всей совокупностью получаемых результатов, в частности - получить некоторый инструмент навигации между экземплярами этих результатов.

Желательно поставить задачу еще шире. Представляется крайне полезным проведение экспериментов, связанных не только с вариацией исходных первичных оценок влияния, но и с изменением самой *структуры модели* – о чем было заявлено выше. Изменение структуры в нашем случае означает добавление или удаление той или иной (одной, реже – нескольких) примитивной связи. Это как раз тот случай, когда стирается грань между модификацией начальных условий и модификацией самой модели. Дело в том, что смысл и роль вариации того или иного показателя становится максимально ясным и исчерпывающим как раз при «шевелении» модели, то есть – при добавлении и\или удалении связей. Иными словами, роль каждого фактора проявляется лишь в силу наличия других факторов, существуют не отдельные факторы, а «экосистемы» из факторов (Рис. 1).

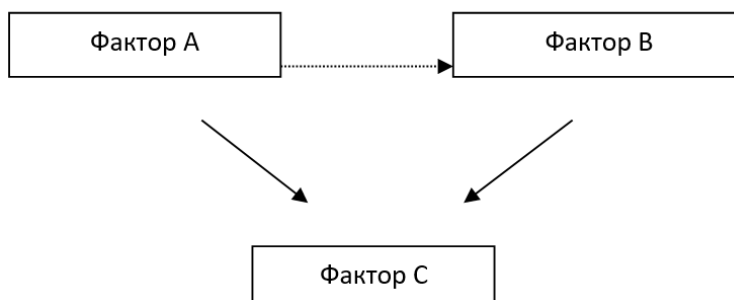


Рис. 1. В большой «экосистеме» факторов влияние фактора А на фактор С может косвенно зависеть от фактора В из-за многочисленности связей

2 Тесная связь верификации и варьирования рефлексивных моделей

Рассмотрим подробнее указанные выше процедуры верификации и рассмотрим, какая совокупность вычислительных экспериментов необходима в каждом случае. Также, проанализируем для каждого сценария необходимость и возможности изменения структуры модели.

1. Анализ чувствительности оценки конкретной системной связи к задаваемым вариациям оценок примитивных связей.

Здесь можно говорить о функциональной зависимости: мы изменяем один или несколько «аргументов» (оценки «примитивных» связей) и получаем (по крайней мере, в идеале), как при этом изменяются вычисленные оценки конкретной «системной» связи.

Важно понять, что провести исчерпывающие вычисления для такой функциональной зависимости практически невозможно. То, что в наших силах, и что имеет смысл – небольшое

количество вариаций некоторых «примитивных» связей. В этом случае пользователю желательно получить не столько график функции, сколько практичный интерфейс, позволяющий в удобной форме задавать вариации одной величины и наблюдать отклик другой величины.

Если говорить о варьировании структуры модели, то формулировку данной процедуры следует уточнить: анализ чувствительности оценки данной системной связи к задаваемым вариациям, а также добавлениям и удалениям примитивных связей. Ясно, что чувствительность оценки выбранной системной связи от выбранной примитивной связи может зависеть от наличия или отсутствия дополнительных связей. Эти «дополнительные» операции (добавление и удаление связей) необходимы при углубленном анализе.

Уже на этапе обсуждения самой первой процедуры верификации рефлексивных моделей становится ясно, что было бы очень полезно неким образом сохранять историю производимых вариаций. По мере дальнейшего рассмотрения эта необходимость предстанет еще более отчетливо.

2. Поиск системных связей, зависящих от конкретной примитивной связи для заданного числа рефлексий.

В этом сценарии вычислительного эксперимента мы варьируем только одну примитивную связь и отслеживаем изменение всех вычисленных (для заданного числа рефлексий) оценочных значений для системных связей.

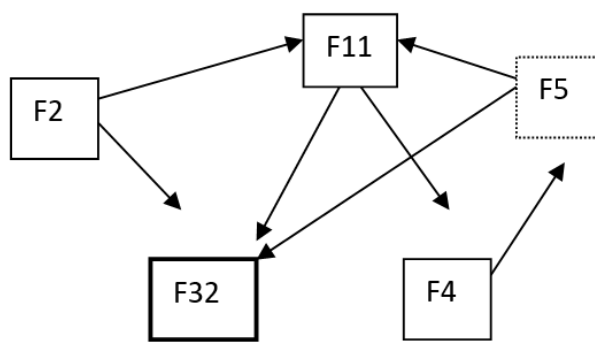


Рис. 2 Оперативная модификация структуры модели: фактор F5 будет удален, фактор F3 добавлен

Имеет ли смысл в этом сценарии менять структуру модели, то есть добавлять и/или удалять примитивные связи? Автору представляется, что имеет, во всяком случае, при углубленном анализе. Та или иная системная связь может зависеть или не зависеть от заданной примитивной связи не в безусловном смысле, а при наличии (или наоборот - отсутствии) дополнительных факторов (разумеется, это верно и для предыдущего сценария). Надо дать исследователю возможность добавлять и удалять (можно сказать – «включать и выключать») такие факторы. Разумеется, это нужно делать не случайным «подбрасыванием монеты», а на основе той или иной гипотезы о совместном действии факторов.

То есть, инструмент изменения структуры модели «на ходу» нужен тому исследователю, у которого есть такие гипотезы, он их оперативно выдвигает. Снова легко видеть, что инструмент должен позволять не просто менять какие-то показатели и добавлять и удалять первичные связи факторов, но и сохранять историю этих изменений. Кроме сохранения самих параметров и настроек модели, необходимо сохранять поясняющие комментарии.

3. Поиск примитивных связей, влияющих на данную системную связь за заданное число рефлексий.

Для этого сценария верификации, как и для двух предыдущих, можно с тем же основанием утверждать, что учет и понимание влияния примитивных связей на системные связи становится более полным (и объемным), если мы понимаем, как это влияние зависит от наличия или отсутствия дополнительных факторов, «подключающих» свое действие к той или иной примитивной связи.

Полагаем, что в данном сценарии добавление или удаление примитивных связей представляется максимально естественным, можно сказать – практически очевидным - инструментом поиска.

Видим, что в последнем сценарии снова вполне уместно поставить задачу предоставить исследователю специальный программный инструмент для работы с вариациями графа

примитивных, оснащенный некоторым хранилищем. В этом хранилище необходимо сохранять как структуру графа (конфигурацию связей), так получаемые результаты – это можно назвать *протоколом* эксперимента. Сохраненные протоколы должны позволять проводить впоследствии удобную навигацию между сохраненными данными.

3 «Графический редактор»: возможности, наработки и перспективы

Можно утверждать, что на данный момент уже имеются наработки, представляющие собой программную платформу для практической реализации задач, описанных выше. Речь идет о так называемом «специализированном графическом редакторе». Этот редактор изначально был частью более сложного программно-технического комплекса, разработанного с целью автоматизировать оценочные расчеты при планировании пилотируемых космических миссий. Архитектура, организация и функционирование комплекса описаны в [4 – 6].

Основная идея указанного комплекса – реализовать его в виде достаточно стандартной и давно известной архитектуры «клиент - сервер» (Рис. 3). Основное отличие, или просто особенность – состоит в том, что в качестве так называемого *back – end* (серверное приложение, выполняющее вычисления, обеспечивающие основные функции всего комплекса) выступает не СУБД (система управления базами данных), как в обычных коммерческих приложениях, а профессиональный пакет математических вычислений. В нашем случае это были MATLAB или Octave [7, 8].

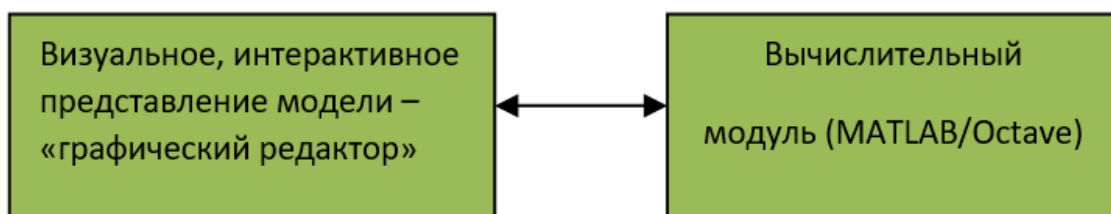


Рис. 3. Система моделирования, реализованная в виде архитектуры «клиент - сервер»

Пользовательский интерфейс представляет собой специализированный *графический редактор*, реализованный в среде веб – браузера и написанный на языке программирования JavaScript. Крупномасштабная структура математических моделей, на которые ориентирована первоначальная версия этого редактора, основана на абстракции *математического графа*. Поэтому и сам редактор построен как инструмент последовательного визуального построения самим пользователем графа, соответствующего экземпляру задачи пользователя. Узлы такого графа могут соответствовать, например, этапам полета космического аппарата, а ребра – переходным процессам между этими этапами (в этом случае мы имеем дело с ориентированным графом).

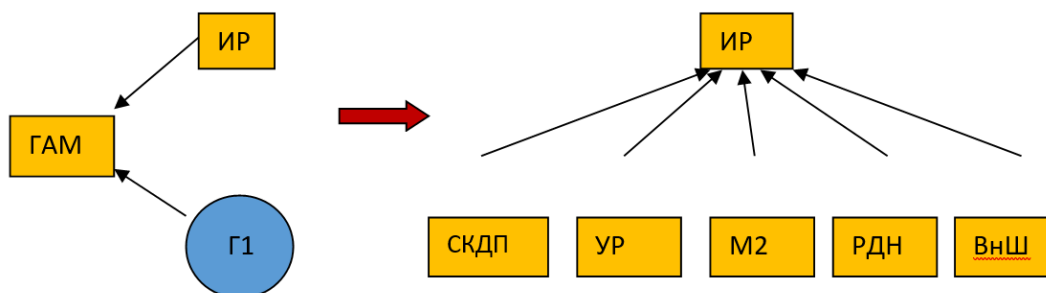


Рис. 4. Переход от локального графа, описывающего воздействие на фактор «Государственные антикризисные меры» (со стороны фактора «Инфляционные риски» и и некоторой группы факторов «Г1»), к локальному графу, описывающему воздействие на фактор «ИР» (со стороны факторов «Смягчение кредитно – денежной политики», «Укрепление рубля», «Денежная масса М2», «Рублевая долговая нагрузка» и «Внешние шоки»).

Язык программирования JavaScript встроен в браузеры практически всех *вендоров* (производителей программного обеспечения) и является фактическим стандартом браузерного программирования. Кроме этого, в настоящее время сообществом веб - программирования разработано огромное количество программных библиотек, написанных на JavaScript, служащих для автоматизации типовых задач программирования в разных областях, включая, например,

графические библиотеки и библиотеки типовых элементов пользовательского интерфейса. Мы активно использовали такие библиотеки при разработке своего графического редактора, о чем подробно рассказано в [4 – 6].

В случае рефлексивных моделей сложного взаимодействия факторов в экономике мы так же имеем дело с (ориентированным) графом. В этом случае имеется собственный смысл у вершин и ребер графа: узлы соответствуют отдельным экономическим факторам, направленные ребра соответствуют примитивным связям влияния. Этот граф можно отрисовывать целиком, но автор придерживается мнения, что в настоящее время в этом нет большой необходимости. Вопреки первичным и интуитивным посылам такой способ представления не делает работу с моделью более удобной.

Представляется гораздо более удобным и эргономичным выводить на экран так называемые *локальные графы*, описывающие все воздействия на выбранный фактор со стороны других факторов (входящих в модель), а также, наоборот, все воздействия выбранного фактора на другие факторы, имеющиеся в модели. Этот граф можно отображать непосредственно в виде небольшого графа (Рис. 4), либо в виде таблицы (Рис. 5). Эти две формы представления взаимно дополняют друг друга, причем локальные графы удобны тем, что можно последовательно переходить от графа для одного фактора к графу для некоторого другого фактора, связанного с данным.

Естественным образом возникает вопрос – где и как проводить соответствующие рассматриваемой модели математические вычисления. Выше уже было отмечено: в системе, разработанной ранее автором с коллегами [4 - 6], вычисления выполняются отдельным специализированным программным продуктом, но вовсе не обязательно придерживаться этого стандарта в любых ситуациях. Профессиональный математический пакет в качестве back – end всего программного комплекса нужен далеко не всегда. Как отмечено выше, возможности самого JavaScript как встроенного языка браузерного программирования постоянно растут за счет все большего числа специализированных библиотек расширения, написанных на JavaScript, которые можно подключать к проектам. В частности, в последние годы интенсивно развиваются библиотеки математических расчетов [9].

GAM	
factor	interaction
IR	35
group_GAM_1	---

group_GAM_1	
factor	interaction
VnSh	#7
VeK	#-5

Save as: tr34
(current)
Save

Рис. 5. Воздействия на фактор «Государственные антикризисные меры» изображено в виде набора таблиц

Разумеется, эти библиотеки, как и сама программная среда веб - браузера, заметно ограничены в возможностях, в первую очередь – в вычислительной мощности, по сравнению со специализированным программным обеспечением. Но мы должны отталкиваться от реально имеющейся в конкретной задаче сложности вычислений. Не слишком трудоемкие вычисления вполне по силам веб – браузеру и соответствующим библиотекам JavaScript. В таких случаях мы

имеем замечательную возможность реализовать весь программный пакет на одном языке программирования JavaScript и запустить его на исполнение в браузере.

Итак, не слишком объемные и сложные математические вычисления в принципе можно проводить в среде браузера, не привлекая дополнительного ПО. Стандартные реализации веб – браузера, практически от всех вендоров, реализуют некоторые функции веб – сервера; в частности, можно загружать и выгружать данные в виде файлов. Это необходимо для длительного (дольше, чем в рамках одного сеанса работы с приложением) хранения разветвленной сети модификаций задачи – ведь мы говорим о варьировании модели. Наконец, как было указано в самом начале данного раздела, визуальные возможности для создания пользовательского интерфейса как графического редактора также предоставляются готовыми (и бесплатными) библиотеками JavaScript. Мы использовали библиотеку Raphael [10] и автор намерен использовать ее в дальнейшем.

Заключение

Таким образом, имеются все технические возможности реализовать в браузере программное обеспечение, функционально позволяющее работать с рефлексивными моделями многофакторного взаимодействия в экономике, в частности: вести и сохранять некоторый протокол вычислительных экспериментов при варьировании структуры модели. Разумеется, этот протокол сам по себе еще подлежит разработке и отладке в сериях таких вычислительных экспериментов.

Программное обеспечение подобного рода является системой в значительной степени нового поколения: это система интерактивного моделирования, в которой не только параметры, но сама структура численного эксперимента оперативно управляется исследователем. Автор рассчитывает, что системы подобного уровня позволят на качественно новом уровне подойти к процессу численного анализа сложных многофакторных задач.

Литература

1. Гусев В.Б., Исаева Н.А. Рефлексивные процедуры анализа экспертных данных // Информационные технологии и вычислительные системы. 2016. № 2. С. 31-35.
2. Гусев В.Б., Павельев В.В. Использование непрерывных шкал при оценивании и принятии решений в сложных проблемных ситуациях: научное издание / В.Б. Гусев, В.В. Павельев. – М.: ИПУ РАН, 2013. – 118 с.
3. Гусев В.Б., Исаева Н.А. Анализ моделей управления на основе экспертных данных. Москва: ИПУ РАН, 2017. – 116 с.
4. Зухба Р.Д., Куракин П.В., Малинецкий Г. Г., Махов С. А., Митин Н. А., Торопыгина С.А. Система моделирования «КОСКОН» как инструмент поддержки принятия решений в космической отрасли. Препринты ИПМ им. М. В. Келдыша. — 2015. — № 113. — 36 с. — URL: <http://library.keldysh.ru/preprint.asp?id=2015-113>.
5. П. В. Куракин. «Новая программная архитектура для специализированных систем математических расчетов». Информационные технологии и вычислительные системы, 2016 г, № 2 стр. 66 – 74.
6. Куракин П. В. «Специализированные системы математических расчетов нового поколения». Программные системы и вычислительные методы – 2016 г, №1(14), стр. 80 – 94. DOI: 10.7256/2305-6061.2016.1.17997.
7. Octave (статья в Wikipedia), URL: https://ru.wikipedia.org/wiki/GNU_Octave.
8. Octave (официальная страница проекта) , URL: <https://www.gnu.org/software/octave/>.
9. John Mueller, Four Serious Math Libraries for JavaScript (<https://blog.smartbear.com/testing/four-serious-math-libraries-for-javascript/>).
10. Официальный веб-сайт проекта Raphael, URL: <http://raphaeljs.com/>.