

## ПРОГРАММНЫЙ КОМПЛЕКС ПА10 ДЛЯ МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ СИСТЕМ

# 11, ноябрь 2016

Маничев В. Б.<sup>1,\*</sup>

УДК: 519.622.2, 621.039.526

<sup>1</sup>Россия, МГТУ им. Н.Э. Баумана

[\\*manichev@list.ru](mailto:*manichev@list.ru)

### Введение

В настоящее время непрерывно растут требования со стороны организаций, выпускающих конечную продукцию, по снижению стоимости развертывания производства изделий и сокращению сроков выпуска изделий на рынок при одновременном повышении качества и снижении стоимости продукции. Важность этих требований и сложность их удовлетворения значительно увеличивается из-за практически взрывного роста сложности производимых изделий. Одновременно возрастает роль быстрой и точной оценки принимаемых проектных решений. При этом крайне важна именно требуемая точность оценки, поскольку стоимость ошибок в принятии проектных решений на ранних стадиях проектирования чрезвычайно высока. Для решения задач моделирования и анализа проектных решений применяются системы инженерного анализа проектных решений – Computer Aided Engineering (CAE), которые можно разделить на три группы: системы анализа на распределенном уровне, на сосредоточенном уровне и на системном уровне проектирования.

Математическими моделями технических объектов при проведении анализа на распределенном уровне являются системы дифференциальных уравнений в частных производных. Моделирование на распределенном уровне характеризуется высокой степенью точности получаемых результатов, требует значительных затрат времени и вычислительных ресурсов, однако возможности их широкого применения на ранних стадиях проектирования для исследования всего изделия существенно ограничены из-за быстрого роста сложности проектируемых изделий. Среди программных комплексов, выполняющих моделирование на распределенном уровне, следует упомянуть, ANSYS, MSC.Nastran, NX.Nastran, Patran, COMSOL, COSMOS и др.

Моделирование на сосредоточенном уровне связано с решением систем обыкновенных дифференциальных уравнений (ОДУ) и позволяет получить точные результаты за приемлемое время без использования больших вычислительных ресурсов даже для достаточно сложных изделий. Поэтому моделирование на сосредоточенном уровне находит все большее применение при проектировании изделий промышленности. К программным

комплексам моделирования на сосредоточенном уровне относятся программы схемотехнического моделирования электронных схем на основе SPICE решателя, MSC.ADAMS, MATLAB-SIMULINK, MapleSym, Mathematica SystemModeler, отечественные разработки серии ПА, МВТУ, EULER, PRADIS и др. В настоящее время наметилась тенденция к увеличению рынка подобных систем.

Однако, широкое распространение системы моделирования на сосредоточенном уровне получили только при проектировании изделий электроники и вычислительной техники, где применяются специализированные программные средства. В других же областях, в частности, при проектировании машиностроительных изделий указанные системы используются лишь эпизодически. Это связано, в том числе, и с отсутствием на рынке универсальных систем, удовлетворяющих требованиям большого числа проектных организаций [9-11].

К недостаткам имеющихся на рынке систем следует отнести их узкую специализацию, либо на моделирование электронных изделий, либо на моделирование механических систем. Так лидер на рынке MSC.ADAMS ориентирован преимущественно на проектирование изделий автомобильной промышленности. Пополнение библиотеки моделей в программном комплексе MSC.ADAMS крайне затруднено и требует от пользователей навыков программирования. Существующие программные комплексы, за редким исключением, не предоставляют возможностей проведения моделирования систем, состоящих из физически различных подсистем (Multiphysics Simulation). Чаще возможности Multiphysics Simulation на распределенном уровне встраивают в перечисленные выше соответствующие CAE системы. Однако эти программные системы не пригодны, например, для анализа электронных схем во временной и частотной областях.

Между тем возрастает актуальность моделирования изделий мехатроники, например, при проектировании Микро Электро Механических Систем или сокращенно МЭМС (MEMS – MicroElectroMechanical Systems) - это множество микроустройств самых разнообразных конструкций и назначения, производимых сходными методами с использованием модифицированных групповых технологических приемов микроэлектроники. Объединяет их два признака. Первый – это размер, второй – наличие движущихся частей и предназначение к механическим действиям. Технический прогресс движется вперед, и основное направление - микроминиатюризация. МЭМС и особенно НЭМС (Нано Электро Механических Систем) технологии весьма перспективны (особенно в тяжелых условиях эксплуатации). Возможно, это будет новая технологическая революция после революции интегральных схем (ИС, БИС и СБИС). Для решения задач моделирования подобных систем в отдельных случаях также может потребоваться объединение средств анализа динамики на сосредоточенном и распределенном уровне. Такой подход позволит удовлетворить требования по точности даже для сверхсложных изделий.

В связи с этим могут быть сформулированы требования, которым должны удовлетворять разрабатываемые в настоящее время системы моделирования на сосредоточенном уровне:

- комплексное моделирование технических объектов, состоящих из различных физических подсистем - электрических, механических, гидравлических, тепловых и т.д.;
- надежность и высокая точность математических ядер таких программ, возможность легкой замены и расширения математического ядра;
- возможность интеграции с системами геометрического моделирования;
- возможность интеграции с системами моделирования на распределенном уровне для многоуровневого анализа;
- наличие интерактивных средств создания моделей пользователями, при этом от пользователя не должно требоваться специальных знаний в области программирования и/или численных методов;
- наличие средств параллельной работы разработчиков моделей, возможность выполнения вычислений на специально выделенном сервере или кластере.

Для решения перечисленных задач необходимо создание нового поколения систем моделирования динамических процессов в технических системах. Учитывая приведенные выше требования, а также опыт, накопленный при разработке программ ПА1, ПА-4, ПА-6, ПА-7, ПА-9, предполагается разработать систему моделирования нового поколения ПА-10 ([www.pa10.ru](http://www.pa10.ru)).

### **Математическое ядро ПА10 - решатель систем ОДУ-ДАУ dmanzhuk**

Поскольку явные методы решения систем ОДУ и ДАУ не являются универсальными и не отвечают требованиям устойчивости при решении с переменным шагом интегрирования жестких систем ОДУ, то в качестве базовых методов интегрирования для моделирования непрерывных динамических систем и объектов рекомендуется выбирать неявные методы интегрирования [12-14]. Программа-решатель систем дифференциально-алгебраических уравнений (ДАУ) dmanzhuk ориентирована, в первую очередь, на решение жестких и сверхжестких систем ОДУ-ДАУ с многопериодным характером решения, которые часто сводятся к решению плохо обусловленных систем линейных алгебраических уравнений (ЛАУ). Все переменные рассчитываются с гарантированной достоверностью и точностью результатов решения и с одинаковой математической и компьютерной точностью для всех переменных расширенного координатного пространства переменных на каждом шаге интегрирования. Программа dmanzhuk рассматривает системы ДАУ в **расширенном координатном пространстве переменных**, в котором к дифференциально-алгебраическим переменным добавлены производные дифференциальных переменных [7]:

$$G(PX, X, Y, t) = 0, \quad (1)$$

где  $X$  - вектор дифференциальных переменных, размерностью  $m$ ;

$PX = dX/dt$  - вектор производных дифференциальных переменных по времени размерностью  $m$ ;

$Y$  - вектор алгебраических переменных размерностью  $k$ ;

$G$  - вектор-функция размерностью  $m + k$ ;

$t$  – время. Заданы начальные условия  $X_0=X(0)$  и отрезок интегрирования  $t=[T_0,TK]$ .

Алгоритмы для решения системы (1) основаны на совместном решении систем ДАУ вида (1), сформированных на соответствующих стадиях разработанных одношаговых, неявных методов интегрирования, относительно дифференциально-алгебраических переменных  $X$ ,  $Y$  и производных дифференциальных переменных  $PX$ , дополненных системой ЛАУ вида  $H(X,PX)=0$  для соответствующих методов интегрирования. Был разработан конкретный вид систем ЛАУ  $H(X,PX)=0$  для  $S$ -стадийных одношаговых, неявных методов интегрирования систем ДАУ вида (1) в расширенном координатном пространстве дифференциально-алгебраических переменных и производных дифференциальных переменных  $X$ ,  $Y$  и  $PX$ . Предложенный подход для решения систем ОДУ-ДАУ не требуют приведения систем ОДУ к нормальной форме Коши и получения «правой» части для этих систем (общепринятое представление в известных формулах численного решения систем ОДУ). Для этих систем ЛАУ были получены соответствующие параметры, разработаны и реализованы три одношаговых, неявных метода интегрирования [4]:

- М1 - А-устойчивый первого порядка точности (совпадает после пересчета параметров с неявным методом Эйлера);
- М2 – АL-устойчивый неявный метод второго порядка точности (совпадает после пересчета параметров с неявным методом трапеций);
- М3 - АL-устойчивый неявный метод четвертого порядка точности (совпадает после пересчета параметров с неявным методом Лобатто IIIA).

### Программная реализация

Как уже было сказано выше, программный комплекс ПА10 ориентирован, прежде всего, на задачи, связанные с решением систем дифференциально-алгебраических уравнений (ДАУ). Описание таких систем может производиться многочисленными способами, например, с помощью эквивалентных электрических схем или просто в виде естественной символьной записи.

Язык описания объекта моделирования и самой задачи должен обеспечивать разработчику необходимую гибкость и широту возможностей. Кроме того, такой язык должен быть легок в освоении и в достаточной степени документирован. В предыдущих версиях ПА (ПА7, ПА9) для разработки моделей были применены специализированные языки, описание на которых в дальнейшем конвертировались в код С или Java соответственно, компилировались и подключались в пакет. Очевидные недостатки подобного решения:

- необходимость изучения как базового языка, так и его расширения;
- малое количество документации.

В ПА10 решено реализовывать несколько иной подход. Разработка моделей ведется непосредственно на языке программирования С++.

Немаловажно, что применение объектно-ориентированных языков программирования для описания объектов моделирования позволяет легко перенести ООП в разработку самих моделей. Возможность установления отношений наследования и агрегации между

классами, содержащими описание моделей объектов, позволяет легко реализовать, например, объединение нескольких моделей в одну или создание частных случаев моделей, не занимаясь разработкой с нуля, а лишь частично перегрузив функциональность базовой модели.

Кроме разработки моделей непосредственно на языке программирования, планируется реализовать возможность их создания в визуальной среде, что не требует умения программировать вообще. В этом случае код классов будет генерироваться средой разработки, а затем компилироваться и подключаться в пакет.

Такой подход, должен значительно упростить разработку новых моделей и значительно расширить возможности пакета по сравнению с прошлыми версиями.

Рассмотрим далее непосредственно реализацию описанной выше концепции математического ядра ПА10.

## Классы-компоненты

*Компонент* – элементарная единица вычислительного процесса. Каждый компонент описывает часть вычислительной задачи и символизирует собой одно или несколько уравнений в общей системе ДАУ. Можно заметить, что в случае эквивалентных электрических схем ими являются не только элементы (сопротивления, источники, конденсаторы и так далее), но и узлы. Для того, чтобы какой-то класс стал компонентом, он должен реализовывать интерфейс, например, **IComponent**.

Компонент далеко не всегда имеет информацию о том, в каких вычислениях ему приходится участвовать. В большом числе случаев он лишь несет описательную информацию, такую как свойства или, например, дифференциальное уравнение в текстовом виде. В компоненте могут производиться вычисления, но не всегда результаты этих вычислений могут быть без дополнительной обработки, применены при решении системы уравнений.

Для перехода от данных и вычислений компонента к непосредственному заполнению матрицы Якоби и вектора значений функций (что собственно и требуется для решения системы ДАУ низкоуровневым математическим ядром) нужен класс **ComponentController**. Объекты именно этого класса (или его дочерние) производят необходимые действия для заполнения матриц и другие сопутствующие вычисления.

Удобство такого подхода значительно облегчает разработку компонентов. Большинство сложных вычислений и операций с матрицами перекладывается на классы-контроллеры, все основные варианты которых можно написать и отладить один раз.

Стоит отметить, что было бы неверным позволять контроллерам компонент иметь полный и неограниченный доступ к матрицам, участвующим в вычислениях. Это не только повышает возможность непреднамеренных ошибок, но и допускает специальную “порчу” данных. Именно поэтому все обращения к матрицам должны быть реализованы через специальный класс **DataAccessor**. Его задача – предоставить возможность доступа только к тем элементам матрицы, в которых компонент действительно нуждается.

## Классы для управления процессом вычисления

Рассмотрим возможные классы, которые управляют процессом моделирования.

Во-первых, к таким классам относится «сборщик» модели, реализованный в классе **Linker**. Как уже упоминалось выше, решаемая система уравнений представлена в виде списка отдельных компонент. Класс **Linker** призван организовывать связь между этим списком и классом **DmanzhukWrapper**, отвечающим за взаимодействие с низкоуровневым математическим ядром.

Его основные задачи:

- определение размерности задачи для правильной инициализации **Wrapper**-а;
- участие в создании **DataAccessor**-ов компонент для обеспечения безопасного доступа к данным (матрице Якоби, значениям переменных и так далее);
- управление процессом заполнения матриц в процессе вычисления.

Кроме «сборщика» в процессе составления и решения системы участвует «разборщик», задача которого управлять получаемыми в процессе моделирования данными. Применение AL-устойчивых методов требует периодического взаимодействия с компонентом, предоставляющим уравнение. Разрывы производных и недоопределенность функций могут быть успешно выявлены именно в том случае, когда программная модель компонента информирована о ходе вычисления и промежуточных результатах.

## Заключение

Таким образом, решается задача построения математического ядра, независимого от представления компонент и вида уравнений. Соответствующие реализации перечисленных интерфейсов позволяют запрограммировать большинство методов автоматического составления сосредоточенных моделей. Также на их основе возможно построение анализаторов и компоновщиков систем для распределенных моделей.

Система, разработанная в соответствии с приведенной архитектурой, должна решать набор тестовых задач, специально разработанных и отобранных для проверки математических ядер самых популярных САЕ-систем.

В настоящее время разработан простейший прототип программного комплекса ПА10 – программа PA10mini, которую можно скачать с сайта [www.pa10.ru](http://www.pa10.ru) [8].

## Список литературы

- [1]. Жук Д.М., Маничев В.Б., Папсуев А.Ю. Обобщенный метод моделирования динамики технических систем // Информационные технологии. 2004. № 8. С. 6-14.
- [2]. Мацяшек Л.А. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML. М.: Издательский дом "Вильямс". 2002. 432 с.
- [3]. Страуструп Б. Язык программирования C++. М.: Бином. 2011. 1136 с.

- [4]. Жук Д.М., Маничев В.Б., Ильницкий А.О. Методы и алгоритмы решения дифференциально-алгебраических уравнений для моделирования систем и объектов во временной области. Часть 1. // Информационные технологии. 2010. №7. С. 16-24.
- [5]. Жук Д.М., Маничев В.Б., Ильницкий А.О. Методы и алгоритмы решения дифференциально-алгебраических уравнений для моделирования систем и объектов во временной области. Часть 2. // Информационные технологии. 2010. №8. С. 23-26.
- [6]. Маничев В.Б., Жук Д.М., Витюков Ф.А. Метод математического тестирования программ анализа переходных процессов в САПР электронных схем // Проблемы разработки перспективных микро- и наноэлектронных систем. 2014. Сборник трудов / под общ. ред. академика РАН А.Л. Стемпковского. М.: ИППМ РАН. 2014. Ч. I. С. 83-88. Режим доступа: <http://www.mes-conference.ru/data/year2014/pdf/D005.pdf> (дата обращения: 28.10.2016).
- [7]. Маничев В.Б., Жук Д.М. Программа DMANZHUK для решения систем дифференциально-алгебраических уравнений общего вида. Свидетельство о государственной регистрации программы для ЭВМ. № 2013615287 от 20 августа 2013. М.: Роспатент. 2013.
- [8]. Маничев В.Б. RA10mini. Описание. Режим доступа: <http://rk6.bmstu.ru/pub/Manichev/> (дата обращения: 28.10.2016).
- [9]. Норенков И.П., Трудоношин В.А., Федорук В.Г. Метод формирования математических моделей для адаптируемых программных комплексов анализа радиоэлектронных схем // Радиотехника. 1986. № 9. С.67-72.
- [10]. Трудоношин В.А. Эволюция средств моделирования динамических систем // Информационные технологии. 2012. № 10. С. 4-7.
- [11]. Трудоношин В.А., Федорук В.Г. Решение обратной задачи динамики с помощью универсальных систем моделирования // Вестник Московского государственного технического университета им. Н.Э. Баумана. Серия: Приборостроение. 2014. № 1 (94). С.94-100.
- [12]. Хайпер Э., Нерсетт С., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи / Пер. с англ. М.: Мир. 1990. 512 с.
- [13]. Хайпер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи / Пер. с англ. М.: Мир. 1999. 685с.
- [14]. Butcher J.C. Numerical methods for ordinary differential equations. 2nd ed. John Wiley & Sons Ltd. 2008. 484 p.