Математическое моделирование систем, процессов и явлений во временно́й области

А.В. Андронов (генеральный директор ООО "Техинспект") andronov@techinspect.ru

Введение

Расчетные задачи, решаемые с помощью *CAE*-систем, становятся всё более сложными. Это связано как с усложнением объектов проектирования, так и с выходом на математическое моделирование запредельных режимов работы объектов. Традиционное математическое моделирование предполагает ряд допущений для облегчения расчетов. Чем больше таких допущений, тем проще (и быстрее) рассчитывается модель, но тем меньше она походит на реальный объект. Как и в случае с квантовой механикой, допущения действуют до тех пор, пока мы не пытаемся получить более точную модель, учитывающую больше свойств объекта. В новых областях – таких, как мехатроника и наноэлектроника – в одну модель объединяются разнородные объекты. Например, при моделировании работы искусственного сустава необходимо учитывать одновременно механические и электронные компоненты. Разброс значений параметров в таких случаях достигает десяти и более порядков. Это существенно повышает требования к математическому пакету: ошибки, накапливаемые за счет допущений и преобразований исходных данных, должны быть сведены к нулю.

Многие системы и объекты описываются дифференциальными уравнениями. Существует две области математического моделирования с помощью численного решения дифференциальных уравнений:

- моделирование в пространственно-временно́й области на основе численного решения дифференциальных уравнений в частных производных (**ДУЧП**);
- моделирование во временно́й области на основе численного решения обыкновенных дифференциальных уравнений (**ОДУ**), которые являются частным случаем дифференциально-алгебраических уравнений (**ДАУ**).

Для решения задач инженерного анализа в пространственно-временно́й области служат программные комплексы *ANSYS* (*ANSYS, Inc.*), *MSC.Nastran* (*MSC.Software Corporation*), *Abaqus* (*Dassault Systèmes*) и др. Во временно́й области используются программные комплексы: *HSPICE* (*Synopsys, Inc.*), *PSPICE* (*Cadence Design Systems*), *MSC.Adams*, *MSC.Easy5* (*MSC.Software Corporation*), *MATLAB* и *Simulink* (*MathWorks, Inc.*), *Maple* и *MapleSim* (*Maplesoft*, подразделение *Waterloo Maple, Inc.*), *Mathcad* (*PTC*), *EULER* (*AutoMechanics, Inc.*), МВТУ (МГТУ им. Н.Э. Баумана), ПА9 (МГТУ им. Н.Э. Баумана) и др.

FMS ПА10

Программы серии **ПА** (Программа Анализа) имеют многолетнюю историю, первая версия была выпущена в 70-х годах. С тех пор вышло уже 9 версий, и ПА9 до сих пор используется в академических кругах для моделирования разнородных систем во временно́й области. Система была реализована как единый продукт на языке *Java*, что резко ограничило скорость выполнения расчетов.

На сегодняшний день в разработке находится новая версия – **ПА10** (см. http://pa10.ru). Она строится на модульной основе, с открытыми форматами обмена между модулями [1]. Такой подход позволяет создавать систему по частям, и в настоящее время разрабатывается ключевой компонент её ядра – библиотека программ-решателей СЛАУ (*систем линейных алгебраических уравнений*) и ДАУ-ОДУ, носящая название ***SADEL*** (*Sets of Algebraic and Differential Equations solvers Library* – Библиотека решателей для систем алгебраических и дифференциальных уравнений).

SADEL содержит набор программ-решателей:

- решатель СЛАУ методом Гаусса с полными матрицами (*LAE-Solver-01*);
- решатель СЛАУ методом прогонки с 3-диагональными матрицами (*LAE-Solver-3diag*);
- решатель СЛАУ методом *LU*-разложения с полными матрицами (*LAE-Solver-02*).

Полная версия библиотеки будет содержать еще три программы-решателя:

- решатель СЛАУ методом *LU*-разложения с разреженными матрицами (*LAE-Solver-03*);
- решатель ДАУ с полными матрицами Якоби (*DAE-Solver-01*);
- решатель ДАУ с разреженными матрицами Якоби (*DAE-Solver-02*).

Существенными особенностями библиотеки являются новый подход к обеспечению точности расчетов и корректности выдаваемых результатов, а также способность решать задачи, выходящие за пределы возможностей других расчетных систем [3].

Библиотека *SADEL* разрабатывается на языке *C* как самостоятельная математическая библиотека, дополняющая широко известные математические библиотеки стандартных программ: *BLAS, LAPACK, LINPACK, NAG, Intel MKL, IMSL, CUBLAS, Magma* и др. Кроме того, она дополняет программы-решатели СЛАУ и систем ОДУ из широко известных математических пакетов программ *MATLAB, Maple, Mathcad, Mathematica* и др. в отношении решения плохо обусловленных СЛАУ, сверхжестких и сверхколебательных систем ОДУ.

Сверхжесткими мы называем системы ОДУ со степенью жесткости более 10^6, которым соответствуют

СЛАУ с числом обусловленности более 10^6, а сверхколебательными – системы ОДУ, моделирующие колебательные системы с добротностью более 10^6.

Когда необходимо решать дифференциальные уравнения?

Моделирование и инженерный анализ во временнóй области при проектировании технических изделий часто предполагает выполнение сложных расчетов. Расчеты проводятся в *CAE*-системах, зачастую интегрируемых в *CAD/CAM*-системы. Исходными данными для расчета служит некий набор свойств объекта, в том числе и геометрических (рис. 1). Иногда формализация моделируемых объектов занимает бóльшую часть времени проектировщика.

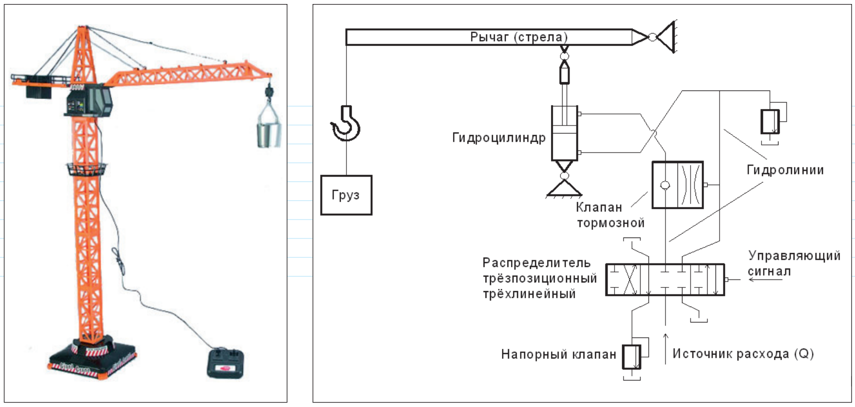

Рис. 1. Башенный кран и его принципиальная схема

Для многих предметных областей структуру объекта можно представить в виде набора простейших взаимосвязанных элементов – эквивалентной схемы (рис. 2). Каждый элемент описывается своими уравнениями, совокупность элементов образует систему. Поскольку уравнения, описывающие компоненты различных систем – электрических, гидравлических и т.д. – содержат производные, то образуется система ДАУ-ОДУ.

Моделирование систем и объектов во временнóй области в *CAE*-системах сводится к решению систем ДАУ-ОДУ, что, в конечном счете, сводится к решению СЛАУ. Результатом расчета, как правило, является график переходных процессов и конечное состояние системы (рис. 2). Расчет крана – достаточно простая задача, с ней справляются все современные пакеты.

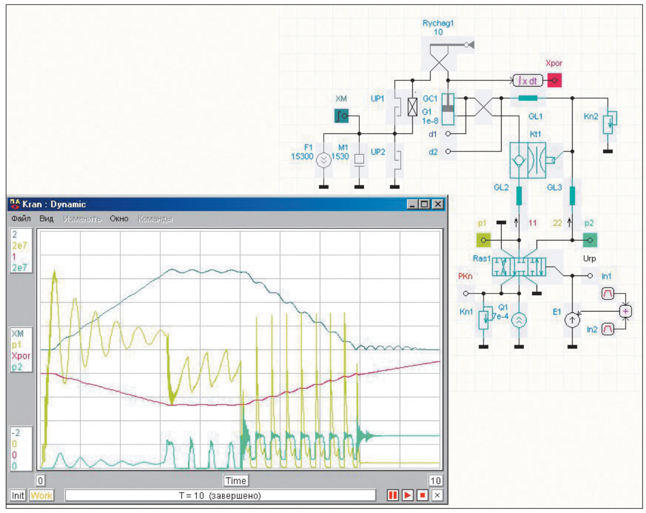

Рис. 2. Эквивалентная схема башенного крана и результаты расчета в ПА9

Объединение в одной модели разнородных элементов (например, “медленных” механических и “сверхбыстрых” электронных в мехатронике) существенно усложняет расчеты. На основе таких моделей появляются сверхжесткие системы ДАУ-ОДУ, решение которых сводится к решению плохо обусловленных СЛАУ. Традиционно считается, что путем преобразований можно получить “хорошую” СЛАУ, которую просто решить. Но это не совсем верно. Например, метод регуляризации Тихонова справедлив только при аналитических вычислениях, тогда как преобразования матриц коэффициентов СЛАУ по этому методу при ограниченной разрядности компьютера не будут эквивалентными [2]. Для сверхжестких систем ДАУ-ОДУ, в которых приходится решать тысячи СЛАУ, из-за такого допущения накапливаются ошибки, и результатом является неверное решение.

Поскольку исходными данными для расчета служат построенная в *CAD*-системе геометрическая модель и задаваемые пользователем параметры, правильное решение заранее неизвестно. О корректности расчета можно судить только по косвенным признакам, так что опыт расчетчика играет тут практически главную роль. В новых и сверхсложных задачах, для которых пока нет базы “похожих” решений, приходится полагаться на *CAE*-систему, которая должна выдавать только корректные результаты.

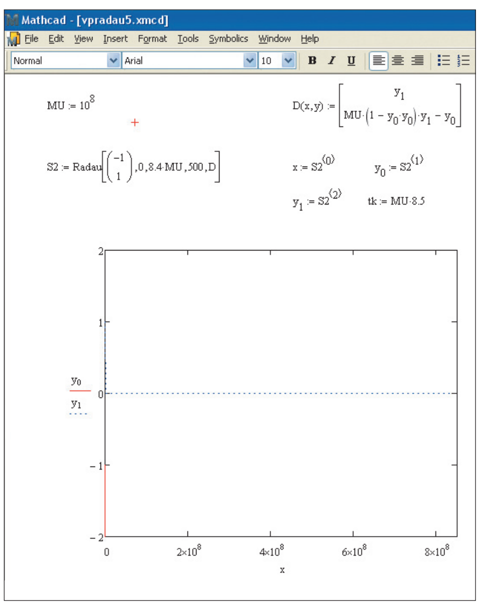

Рис. 3. Расчет генератора Ван Дер Поля в среде Mathcad (жесткость – 10^8)

Для примера рассмотрим задачу Ван Дер Поля [4]. Это нелинейная задача с жесткой системой ОДУ 2-го порядка, для которой известно точное решение (генератор Ван дер Поля) и можно задать параметр жесткости *MU*. На практике обычно используются значения этого параметра от 10^6 до 10^9, а бóльшие значения пока встречаются редко.

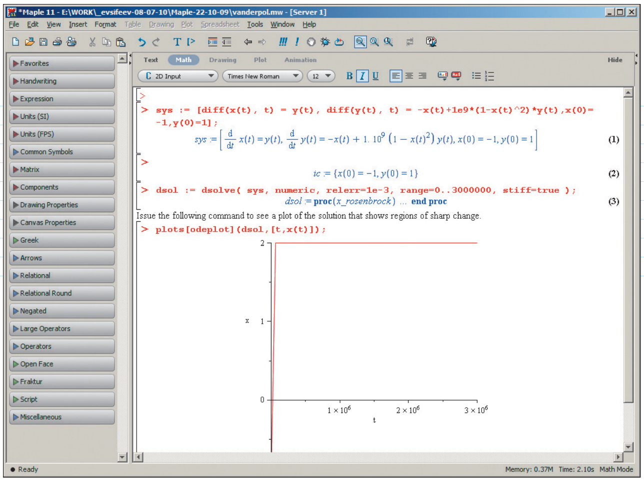

Рис. 4. Расчет генератора Ван Дер Поля в среде Maple (жесткость – 10^9)

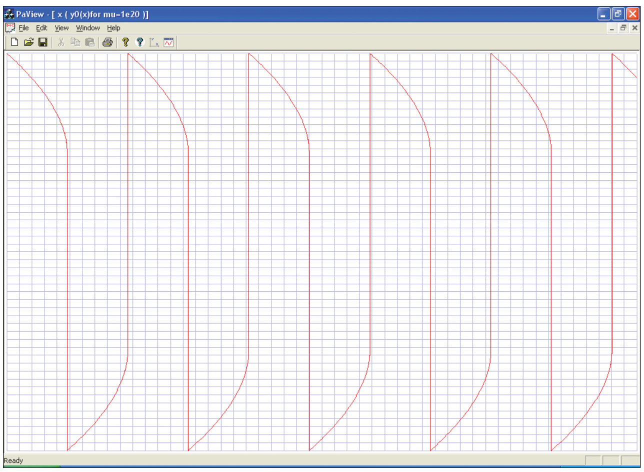

Рис. 5. Расчет генератора Ван Дер Поля в среде SADEL (жесткость – 10^{20})

На рис. 3, 4 приведены результаты решения этой задачи в системах *Mathcad* и *Maple*. При значениях жесткости менее 10^8 *Mathcad* дает верное решение. *Maple* при значении 10^6 предупреждает об ошибке; при увеличении жесткости до 10^9 сообщение об ошибке пропадает, результат перестает быть корректным. Расчет с помощью *SADEL* дает верный результат до значения жесткости 10^{20} (рис. 5). *Mathcad* и *Maple* при таком значении параметра никаких сообщений не выдают, хотя результат получается неверный.

Чтобы быть уверенным в корректности расчетов, пользователь должен оценивать степень "хорошести" (жесткости, добротности, обусловленности) своей задачи. Сейчас это делается перед началом моделирования, но для сложных вычислительных задач надо делать такую оценку и в ходе расчета.

Ошибаться могут все

Сложные вычислительные задачи, решение которых неочевидно, приходится рассчитывать в разных *CAE*-системах и затем сравнивать результаты. Из-за различий в представлении исходных данных в *CAE*-системах расчетные модели не всегда идентичны, поэтому результаты тоже получаются различные.

Некоторые *CAE*-системы позволяют пользователю разрабатывать свои собственные алгоритмы и обрабатывать массивы исходных данных, представленные в табличном виде. В этом случае инженер опирается на базовые алгоритмы, предоставляемые системой, для построения более сложного алгоритма обработки своих исходных данных. О настоящей гибкости можно говорить, когда пользователь оперирует отдельными расчетными модулями и использует их при разработке приложения на языке высокого уровня. Часто проектировщики сами пишут программы-решатели (как правило, на языке *C*), считая это простой задачей. За счет удачных интерфейсов такие программы иногда становятся популярными в своей области (например, система анализа ветровой устойчивости зданий). Однако в них изначально не учитываются все особенности расчета, и поэтому корректно они работают в ограниченном пространстве решений, а за его пределами может произойти что угодно – от фатального сбоя до выдачи неверных данных без каких-либо предупреждений. Схожая проблема наблюдается и у крупных математических пакетов.

Для проверки корректности работы *CAE*-систем используются "эталонные" задачи, имеющие известное решение. Сравнение с ним результатов расчетов позволяет оценить, насколько этой *CAE*-системе можно доверять при решении определенного класса задач, и в каких пределах она выдает верный результат.

Проверка решателей ОДУ, входящих в состав популярных математических пакетов, на сложных задачах выявила проблемы при решении тестовых жестких систем ОДУ с многопериодным решением (табл. 1). Тестирование было проведено для параметров интегрирования, рекомендуемых для этих решателей по умолчанию. Так как для базового инженерного расчета достаточной является точность в три значащих цифры для всех переменных систем ДАУ-ОДУ, сравнение решателей ДАУ-ОДУ проводилось при заданной относительной точности интегрирования 0.001 (в программном комплексе *MATLAB* такая точность задается по умолчанию).

Табл. 1. Тесты для жестких систем ОДУ с многопериодным решением

Тесты \ Решатели ОДУ	*SADEL*	*Mathcad*	*MATLAB*	*Maple*
Тест 1. Уравнения Ван дер Поля, $MU=10^6$	+	+	+	–
Тест 1. Уравнения Ван дер Поля, $MU=10^9$	+	–	–	–
Тест 2. Высокодобротный фильтр $kt=1$, $ki=1$, $ku=0.01$	+	–	–	+
Тест 2. Высокодобротный фильтр $kt=10^{-104}$, $ku=1$, $ki=1$	+	–	–	–
Тест 3. Локально-неустойчивая система ОДУ, $m=10^6$	+	+	–	–
Тест 4. Моделирование свечения лазера	+	+	+	–

Табл. 2. Сравнение решателей СЛАУ

Решатель СЛАУ	Тестовая задача с матрицей Уилкинсона (*Wilkinson*), обусловленность системы *Cond(A)*=10^6	Тестовая задача с матрицей Гильберта (*Hilbert*), обусловленность системы *Cond(A)*=1.5e13
LAE-Solver-01	+ (15)	+ (15)
LAE-Solver-02	+ (15)	+ (15)
MATLAB	- (10)	- (4)
Mathcad	- (10)	- (4)
Mathematica	- (10)	- (4)
Maple	- (10)	- (4)
NAG-LAPACK	- (10)	- (4)
IMSL	- (10)	- (4)
Intel MKL	- (10)	- (4)
CUBLAS+Magma	- (10)	- (4)

Сравнивались только методы интегрирования соответствующих решателей ОДУ, рекомендуемые для решения жестких систем ОДУ. Знак минус означает невозможность получения решения или (в большинстве случаев) качественно неверный результат, выдаваемый безо всякого предупреждения о возможных ошибках. Решение системы ОДУ для этих тестовых задач (табл. 2) сводилось к решению СЛАУ с плохо обусловленными матрицами коэффициентов – такими, как матрицы Уилкинсона и Гильберта [5].

Основные требования к решателю СЛАУ при тестировании – задавать исходные данные в типе *double* и выдавать результаты расчета в *double* с 15-ю верными значащими цифрами для всех неизвестных значений решения соответствующей СЛАУ. Знак минус перед цифрой в табл. 2 означает, что решатель СЛАУ с параметрами по умолчанию не обеспечил точность в 15 верных значащих цифр для всех неизвестных значений соответствующей тестовой задачи и не выдал никакого предупреждающего сообщения об этом. В скобках указано полученное количество верных значащих цифр.

С деталями тестирования и полным описанием тестов можно ознакомиться на сайте *http://pa10.ru*.

Сверхжесткие и высокодобротные системы

Рассмотрим соответствующие примеры при решении реальных задач:

• расчет высокодобротного фильтра низких частот при увеличенной жесткости задачи (рис. 6÷9);

• расчет свечения простейшего лазера – нелинейная задача с умеренными жесткостью и добротностью (рис. 10, 11).

Математические пакеты прекрасно решают задачи с низкой жесткостью и добротностью при помощи любых имеющихся в их комплекте программ решения ОДУ. Однако, при повышении жесткости, рекомендуемые программы для решения жестких ОДУ (на рисунках приведены результаты для математических пакетов 2007 года выпуска на платформе *x*32) давали неверные результаты безо всякого предупреждения.

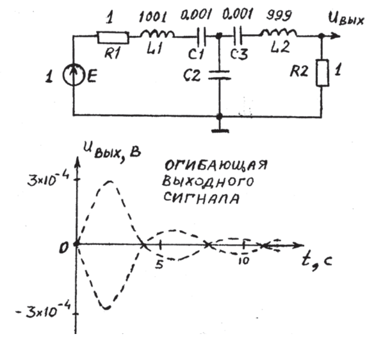

Рис. 6. Высокодобротный фильтр низких частот и аналитическое решение для выходного напряжения фильтра при единичном скачке напряжения на входе

Опыт многих инженеров подсказывает, что использование для одной задачи разных методов расчета (при условии, что они корректны) должно давать одинаковые результаты. Расхождение говорит об ошибке в исходных данных или о невозможности решения задачи выбранными средствами. Применение нескольких разных алгоритмов и сравнение результатов до представления их пользователю обеспечивает бóльшую достоверность результата, но существенно увеличивает время расчета. К счастью, новые алгоритмы позволяют определить, достижим ли результат при заданной

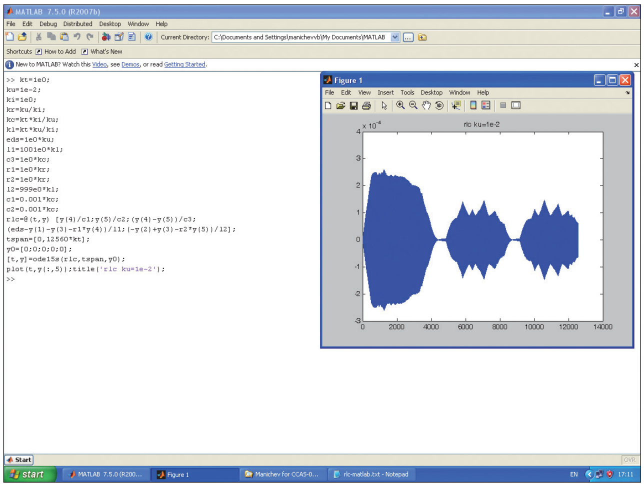

Рис. 7. Расчет фильтра в среде MATLAB 7.5.0 дает неверное решение без предупреждения об ошибках

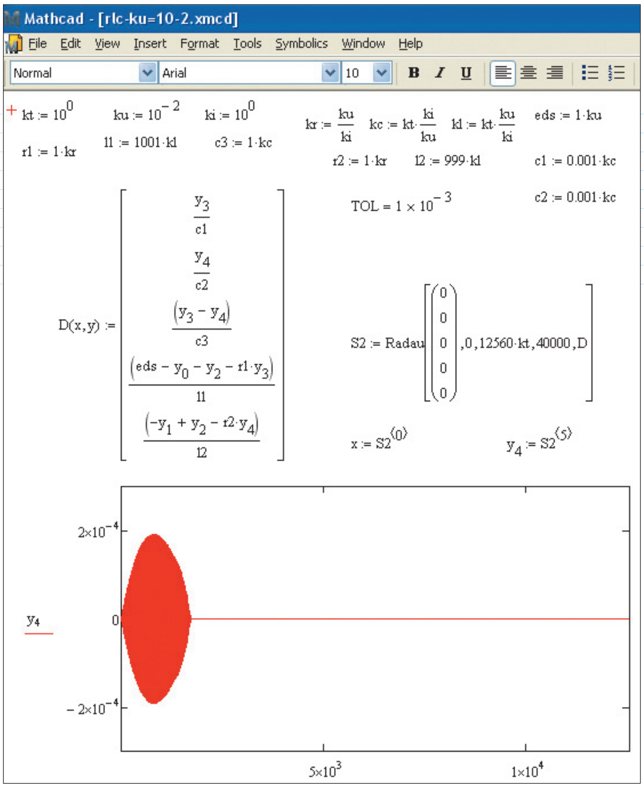

Рис. 8. Расчет фильтра в среде Mathcad дает неверное решение без предупреждения об ошибках

точности, будет ли решение устойчивым. Для многих исследовательских задач погрешность в 50÷100% является приемлемой, если известно, что решение существует. Намного хуже, когда решение на порядки отличается от того, что выдает система, и нет возможности определить, действительно ли оно существует…

Многие ошибки в расчетах связаны с аппаратными ограничениями вычислительной техники, выходом за пределы возможностей стандартных компьютеров и некорректной обработкой этих ситуаций. Возможным выходом является использование суперкомпьютеров, которые позволяют существенно расширить круг решаемых задачи.

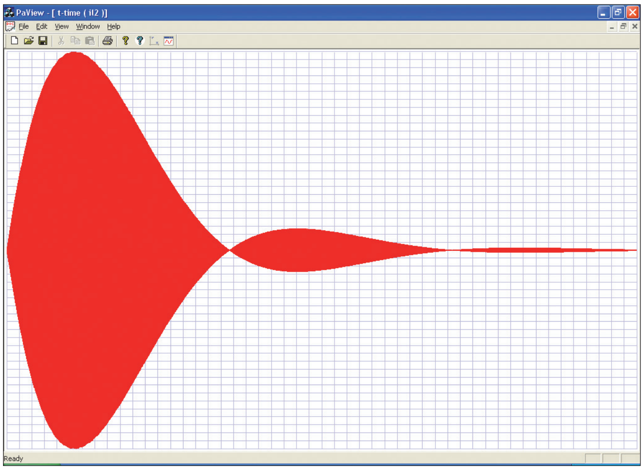

Рис. 9. Расчет фильтра в среде SADEL дает верное решение

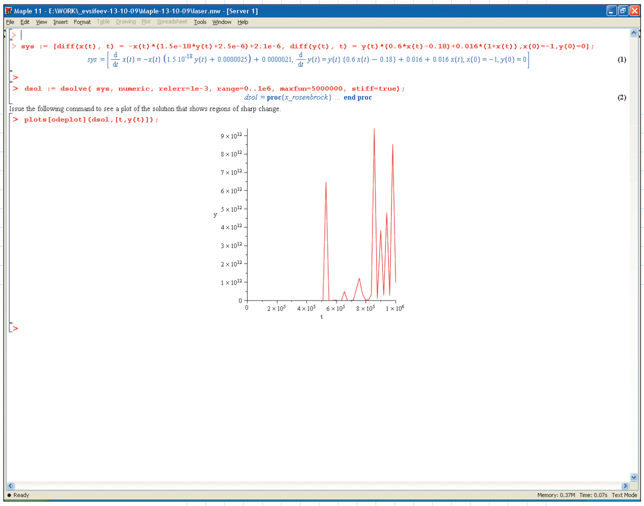

Рис. 10. Расчет лазера в среде Maple дает неверное решение без предупреждения об ошибках

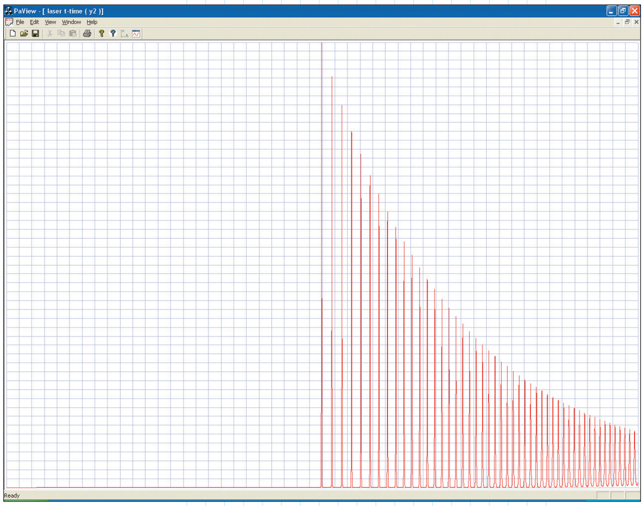

Рис. 11. Расчет лазера в среде SADEL дает верное решение

Сверхсложные задачи и суперкомпьютеры

Современные суперкомпьютеры намного дружественнее к пользователю, чем их предшественники. Можно со своего рабочего места подключиться к ним методом удаленного доступа, загрузить и скомпилировать приложение, скачать результат. Правда, для этого нужно получить доступ и оплатить машинное время, или же установить такой суперкомпьютер у себя. В России оба варианта реальны скорее для государственных, образовательных и научно-исследовательских учреждений, чем для частных компаний, даже проводящих научные исследования за свой счет.

Новое слово в развитии суперкомпьютеров сказала компания *NVIDIA*, выпустившая персональный суперкомпьютер. Графические процессоры *Tesla C2050/C2070* с 448 ядрами устанавливаются в обычный слот *PCIe* и обеспечивают пиковую производительность до 1.03 *TFLOPS* при операциях одинарной точности (по информации с официального сайта www.nvidia.ru).

МАШИНОСТРОЕНИЕ И СМЕЖНЫЕ ОТРАСЛИ

Такая производительность достигается за счет распараллеливания расчетов на множество ядер *CUDA* (*Compute Unified Device Architecture*), которые одновременно обрабатывают большой массив данных. Естественно, повышение скорости возможно только после создания новых алгоритмов расчета, учитывающих параллельную модель *CUDA*. Некоторые задачи распараллеливаются хорошо, и уже появились библиотеки и компоненты, позволяющие в разы увеличить скорость их решения.

К сожалению, однозначно хороших параллельных алгоритмов для решения СЛАУ и ДАУ-ОДУ не существует. Разработчики ПА10 работают над адаптацией отдельных алгоритмов решения ДАУ-ОДУ для платформы *NVIDIA CUDA*. Основное преимущество этой платформы – возможность одновременного выполнения команды десятками ядер, производящими операции над смежными блоками данных, которые находятся в быстрой видеопамяти. Одновременное решение не всегда приемлемо, поэтому основная задача – группировка данных в условно независимые блоки. Кроме того, на параллельную платформу переводятся вспомогательные вычисления, связанные с оценкой корректности решения. Отделение их от основного блока само по себе дает существенный прирост производительности.

Поиск верных результатов

Как же получать только верные результаты? К сожалению, на этот вопрос нет однозначного ответа. Можно сформулировать несколько рекомендаций, которые послужат ориентирами при оценке решения, выдаваемого системой.

Эффективнее всего оценивать обусловленность решаемых СЛАУ. Если она больше 1, то результат необходимо перепроверять в различных системах (или применять системы, решающие плохо обусловленные СЛАУ). При решении дифференциальных уравнений могут решаться сотни и тысячи СЛАУ, поэтому такая проверка подойдет только для расчетов с использованием библиотек, когда пользователь может контролировать процесс расчета.

Традиционно, хорошим способом является проведение расчета хотя бы в двух разных системах. Путь этот – трудоемкий и ресурсоемкий, но это единственный выход, когда ошибаться нельзя ни в коем случае.

В научной литературе часто публикуются задачи с известным решением, в том числе "трудные" для математических пакетов. Многие тесты содержат параметр, определяющий степень сложности расчета (такой, как жесткость в задаче Ван Дер Поля). Постепенное увеличение сложности дает возможность определить границу, за которой пакет выдает неверные решения. Дальнейшее увеличение позволит найти предел, за которым система "замечает", что расчет не верен.

Задачи, описанные выше, пока не очень распространены. Расчетные системы постоянно развиваются и дополняются. Велика вероятность, что в их состав войдет библиотека *SADEL* или другие подобные разработки. Пока же для уверенности в корректности решения приходится проверять, правильно ли система решает плохо обусловленные СЛАУ.

Заключение

CAE-системы не стоя́т на месте. Всё теснее становится их интеграция с *CAD*, пользователь освобождается от многих рутинных операций, получение результатов расчетов сводится к нескольким нажатиям кнопок мыши. Но на новых задачах такой подход проявляет себя не с лучшей стороны: потеря гибкости и привычное доверие к системе провоцируют появление ошибок в расчетах. Ошибки накапливаются, и известны случаи, когда они приводили к авариям и катастрофам [2].

Современный математический пакет программ представляет собой сложный набор модулей, созданных различными группами разработчиков и объединенных общим интерфейсом. Как известно, эффективные расчетные алгоритмы пишутся на языках *C* и *FORTRAN*, тогда как современные приложения разрабатываются в *RAD*-среде, такой как *.NET*. Разработанные много лет назад модули подключают к новым интерфейсам, дописывают и перерабатывают, но в рамках крупного пакета сложно контролировать все чисто математические аспекты…

Наука движется вперед, создаются новые алгоритмы и методы расчетов. Их качественная реализация позволяет ускорить расчеты, повысить достоверность и корректность. При разработке ПА10 учтены описанные проблемы. В первую очередь, эта система предназначена для пользователей, которым нужна максимальная гибкость расчетов при стабильном результате. Рассмотренная в статье библиотека *SADEL* всесторонне тестируется – в том числе, независимыми учеными. Возможность прямого обращения к библиотекам в рамках ПА10 останется и в дальнейшем, что необходимо для сохранения максимальной гибкости, а также для проведения новых независимых тестов.

Скачать демонстрационную версию библиотеки и больше узнать о проекте ПА10 можно на сайте *http://pa10.ru*.

Литература

1. Жук Д.М., Маничев В.Б., Андронов А.В. Платформа математического моделирования во временно́й области разнородных технических систем и объектов *FMS PA10* // Всероссийская научно-техническая конференция "Проблемы разработки перспективных микро- и наноэлектронных систем – МЭС-2010", Истра, 4–8 октября 2010 года // *www.mes-conference.ru*.

2. Петров Ю.В. Обеспечение достоверности и надежности компьютерных расчетов. Санкт-Петербург: БХВ-Петербург, 2008.

3. Жук Д.М., Маничев В.Б., Ильницкий А.О. Методы и алгоритмы решения дифференциально-алгебраических уравнений для моделирования систем и объектов во временно́й области. Часть 1, 2 // Информационные технологии (*www.novtex.ru/IT*), 2010, №7, с. 16–24, №8, с. 23–26.

4. Хайрер Э., Нерсетт С., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи.– Москва: Мир, 1990.

5. Тыртышников Е.Е. Методы численного анализа.– Москва: Издательский центр "Академия", 2007, 320 с. (Университетский учебник. Серия "Прикладная математика и информатика").