УДК: 519.876.5, 004.942

ПРОГРАММНЫЙ КОМПЛЕКС ТЕСТИРОВАНИЯ ОДНОШАГОВЫХ МЕТОДОВ ДЛЯ СИСТЕМ ОДУ И ДАУ

Зубанов Алексей Михайлович

Аспирант
ГБОУ ВПО «Международный Университет природы, общества и человека «Дубна»,
Филиал ДИНО, лаборатория супервычислений;
141800, Московская обл., г. Дмитров, мкрн. ДЗФС, 23;
e-mail: azubanov@gmail.com.

В работе описан программный комплекс поддержки работы прикладного математика, занимающегося получением новых численных методов решения систем обыкновенных дифференциальных уравнений (ОДУ). Комплекс имеет распределенную структуру, встроенный решатель (позволяющий тестировать новые методы), поддержку выведения уравнений порядка и устойчивости и позволяет пользователю работать с использованием сервиса «тонкий клиент».

Ключевые слова: программный комплекс, ОДУ, разностные схемы.

PROGRAMMING COMPLEX FOR TESTING ONE-STEP METHODS SOLVING ODE AND DAE SYSTEMS

Zubanov Alexei

PhD student;
Dubna International University of Nature, Society, and Man,
Dmitrov branch;
141800, Dmitrov, Moscow reg., DZFS str., 23;
e-mail: azubanov@gmail.com.

The article presents programming complex for supporting mathematic scientists in creating and testing of new ODE systems solving methods, based on difference schemes. Application has following features: distributed architecture, users operate via thin client, automatic order and stability conditions creation, embedded ODE solver for testing new methods.

Keywords: scientific software, ODE, difference schemes.

Введение

Множество практических задач физики, химии, электротехники и других естественных и технических наук требуют численного решения задачи Коши для систем ОДУ:

$$\frac{d\vec{u}}{dt} = \vec{f}(t,\ \vec{u}),\ \vec{u}(0) = \overrightarrow{u_0}, \tag{1}$$

где $\vec{u} = (u_1,\ \ldots,\ u_n)^T,\ \vec{f} = (f_1,\ \ldots,\ f_n)^T,\ \overrightarrow{u_0} = (u_{01},\ \ldots,\ u_{0n})^T \in R^n$.

Наиболее надежными являются одношаговые методы, к которым относятся методы Рунге-Кутты (2) и их модификация – методы Розенброка (3), подробно описанные в [1].

$$\overrightarrow{K_i} = \vec{f}\left(t + c_i\tau,\ \vec{y}(t) + \tau\sum_{j=1}^{s} a_{ij}\overrightarrow{K_j}\right),\ i = 1,\ \ldots,\ s; \tag{2}$$

$$\vec{y}(t+\tau)=\vec{y}(t)+\tau\sum_{j=1}^{s} b_j \vec{K}_j \ ,$$

где s – количество стадий, τ – локальный шаг сетки численного интегрирования задачи (1), $\vec{y}(t)$ – её разностное решение; c_i, a_{ij} и b_j (веса) – коэффициенты схемы, а $\vec{K}_j$ – неизвестные стадии, геометрический смысл которых – направление движения вдоль траектории решения.

Для уменьшения численных затрат на реализацию неявных схем Розенброк предложил использовать лишь одну Ньютоновскую итерацию. В этом случае в записи схем появляются частные производные (Якобианы) исходной системы:

$$\left[\vec{E}-\tau\gamma_i \frac{\partial \vec{f}}{\partial \vec{u}}\left(t+c_{1,i}\tau,\vec{y}(t)+\tau\sum_{j=1}^{i-1} a_{1,ij}\vec{K_j}\right)\vec{K_i}\right]=$$
$$=\vec{f}\left(t+c_{2,i}\tau,\vec{y}(t)+\tau\sum_{j=1}^{i-1} a_{2,ij}\vec{K_j}\right)+\tau\frac{\partial \vec{f}}{\partial \vec{u}}\left(t+c_{3,i}\tau,\vec{y}(t)+\tau\sum_{j=1}^{i-1} a_{3,ij}\vec{K_j}\right)\sum_{j=1}^{i-1}\gamma_{ij}\vec{K_j}, i=\ 1,\ \ldots, s, \qquad (3)$$

где $\vec{E}$ – единичная матрица, а $\dfrac{\partial \vec{f}}{\partial \vec{u}}$ – *матрица Якоби* исходной системы (1). Однако такой подход резко увеличивает количество свободных коэффициентов (которыми можно распорядиться). Например, 5-ти стадийный метод имеет 65 коэффициентов. Кроме того, при такой записи схемы можно использовать комплексные значения (это направление, в частности, развивалось в работах [2, 3]).

Был построен программный комплекс решения систем дифференциальных уравнений описанными методами. Схематично архитектура комплекса может быть представлена в виде взаимосвязанных блоков (рис. 1)

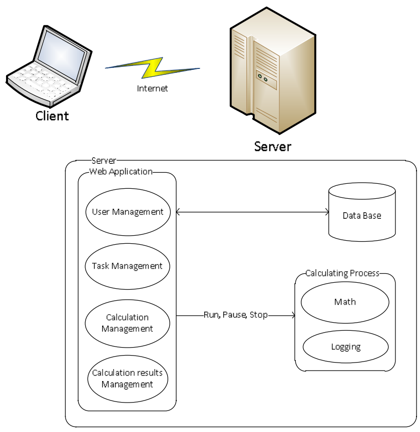

Рис. 1. Схема системы

Web-приложение реализовано на ASP.NET C# 4.0 с использованием MS Ajax Toolkit и JQuery, слой доступа к данным Entity Framework. Ядро математических расчетов реализовано как отдельное C++ приложение с использование библиотек stl, boost, mpfr, mpir. Обмен данными между подсистемами осуществляется через специальные XML файлы описания методов, задач, запусков. Информация о пользователях системы, библиотека задач и методов хранится в базе данных (MS SQL Server).

На рис. 2-4 представлены снимки некоторых экранов Web-приложения.

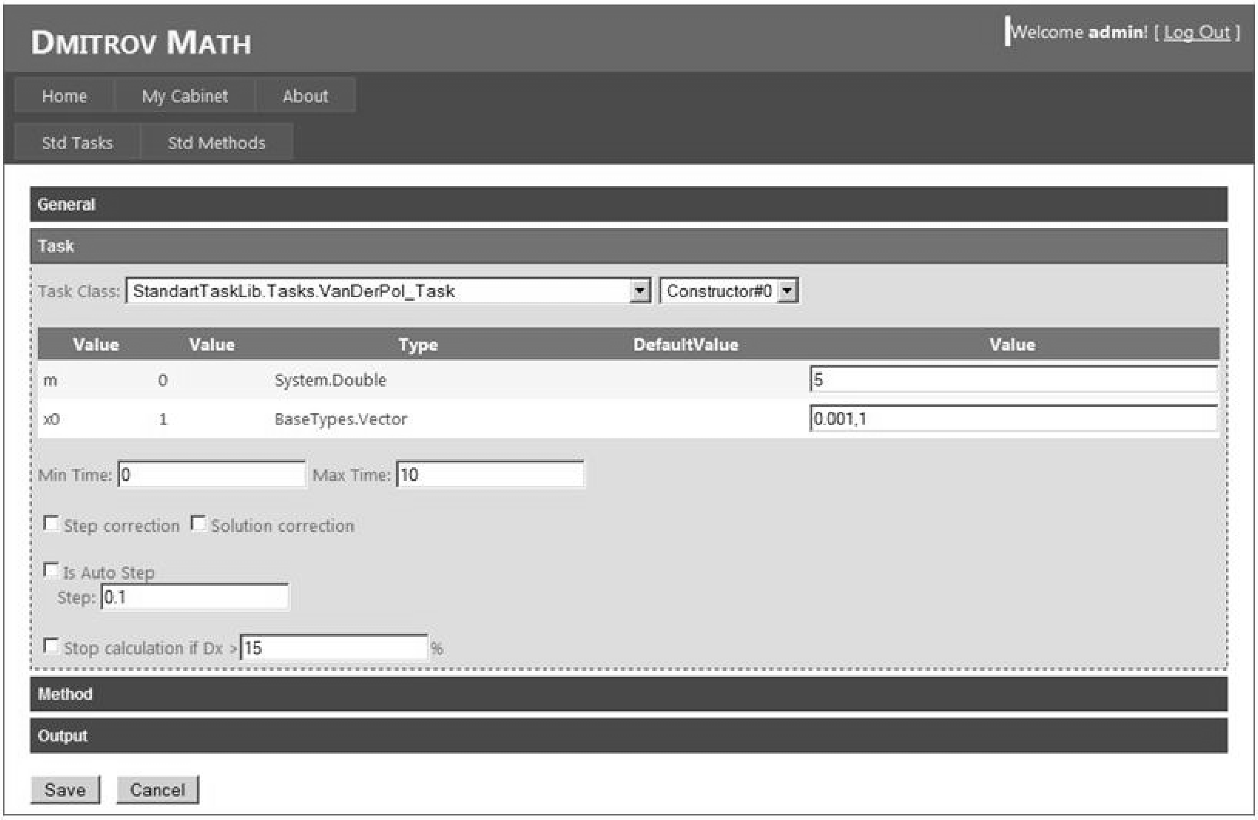

Рис. 2. Экран создания новой задачи

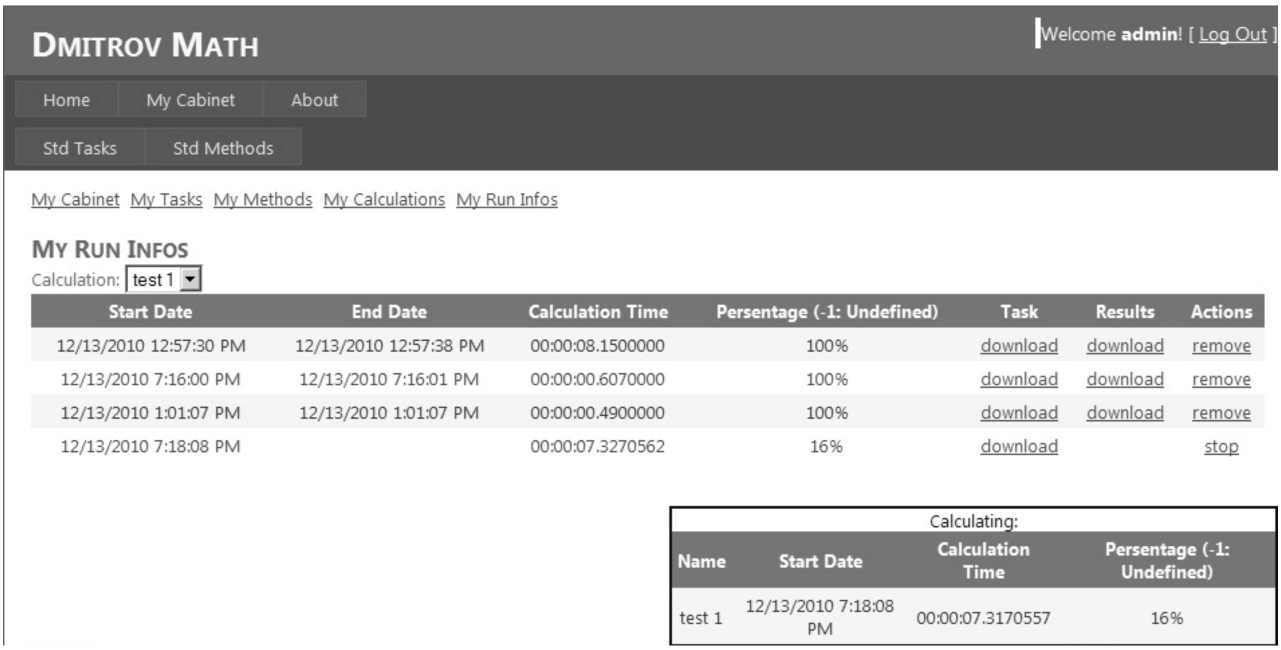

Рис. 3. Экран списка задач

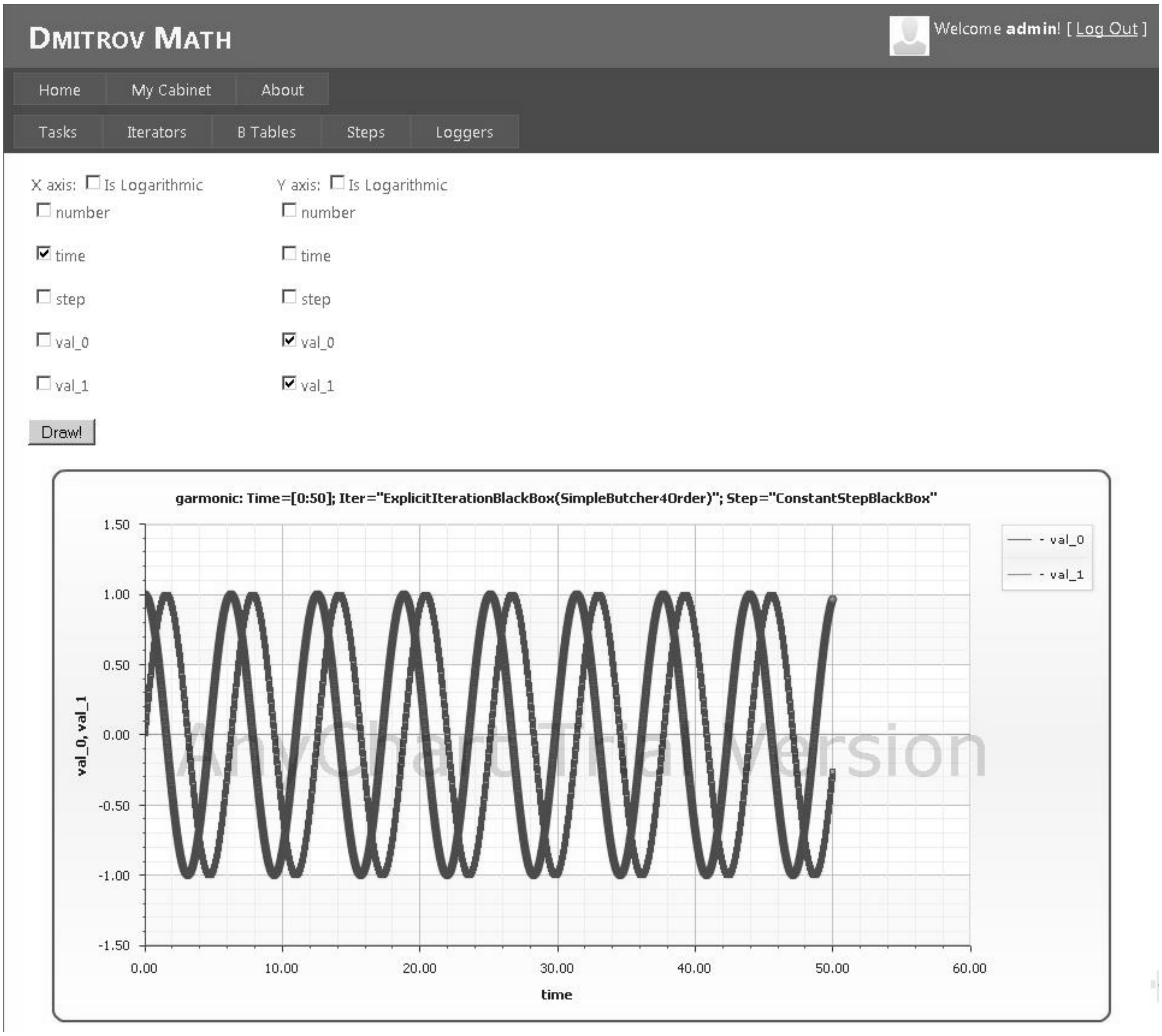

Рис. 4. Экран визуализации результатов счета

Математическая подсистема

Подсистема использует как внешние библиотеки (Boost, STL, MPFR, MPIR, MPFR C++), так и собственные (линейной алгебры, методов интегрирования систем ОДУ и ДАУ, тестовых задач). На рис. 5 изображена диаграмма классов математического ядра.

Математическое ядро собирается в виде серии консольных приложений с разными характеристиками вычислительного ядра (способ представления и хранения вещественных чисел с указанием длины мантиссы). Задания на счет описываются в виде специально сформированного XML-файла, который программа принимает в качестве параметра. Ядро, проводя счет, периодически скидывает на диск данные и свой статус. Web-приложение по команде пользователя формирует XML-файл задания и запускает приложение ядра на счет. После чего периодически опрашивает файлы, в которые приложение пишет свой статус. Web-приложение запоминает идентификационный номер процесса ядра и по команде пользователя может в любой момент принудительно завершить счет. В любой момент времени пользователь системы может получить вывод математического ядра и построить графики по полученным данным.

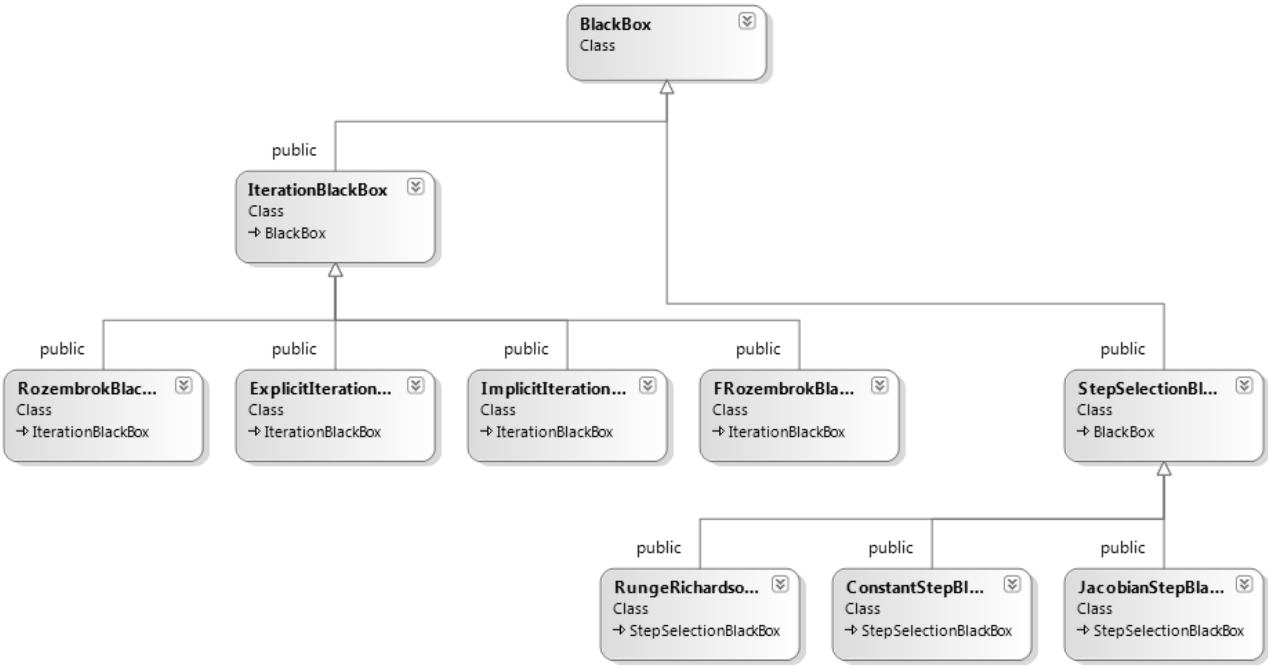

Рис. 5. Диаграмма классов математического ядра

Ядро может быть использовано независимо от Web-приложения – как отдельная система счета, загружаемая через любую вычислительную мощность (например, удаленную). Для этого достаточно сформировать файл описания задания, который может быть выгружен по команде пользователя с сайта. Пробные тестирования ядра в удаленном режиме, в частности, проводились на кластере Института математического моделирования в 2010 году.

Подсистема символьных вычислений

Неотъемлемой частью процесса построения нового метода является выведение уравнений порядка и уравнений затухания для методов из некоторого общего класса, используемых для численного решения систем ОДУ или ДАУ. Этот процесс требует значительного внимания математика и довольно трудоемок, т.к. подразумевает большую и кропотливую работу по разложению в ряды Тейлора точного и численного решения и их сопоставления. Например, даже в случае автономной правой части исходной системы (1) количество различных дифференциалов до 4-го порядка включительно равно 17.

На вход подсистема получает от пользователя необходимые для построения уравнений порядка данные.

- Выбранный пользователем тип семейства схем. На данный момент поддерживаются следующие схемы:
    - Методы Розенброка в наиболее общей форме записи (включая случай с комплексными коэффициентами).
    - Методы типа Розенброка с факторизуемым оператором (в виде полинома от Якобиана системы) – они описаны в [4].
- Указания порядка метода и количество стадий.
- Указания конкретного подкласса методов, для которого необходимо получить уравнения порядка и затухания (какие переменные обращены в ноль для конкретной схемы из выбранного подмножества).
- Параметры форматирования итогового вывода.

Результатом работы подсистемы является серия текстовых записей системы уравнений с различной степенью упрощения системы. Особенности реализации перечислены ниже:

- Интеграция с web-комплексом численного интегрирования.

- Поддержка комплексных коэффициентов.
- Интеграция с пакетом Mathematica (http://www.wolfram.com/mathematica/) для финального упрощения системы уравнений в автоматическом режиме.
- Оптимизация аналитических произведений и разложений, основанных на данных полученных от пользователя:
  - Исключаются операции, приводящие к получению членов с вкладом менее O(τ).
  - Исключаются члены, содержащие нулевые коэффициенты.

Пример вывода уравнений порядка до 4-й степени включительно, полученных системой для факторизованных схем [4], приведен ниже:

```
O(tau^0)   1 = b0+b1;
O(tau^1)   1/2 = b1*(a10+g11+ro10);
O(tau^2)   1/6 = (b1*(a10^2+2*d10*g11+2*h10*ro10))/2;
           1/6 = b1*(-g12+g11*(a10+g11+ro10));
O(tau^3)  1/24 = b1*(a10^3+3*d10^2*g11+3*h10^2*ro10)/6;
          1/8 = b1*d10*(-g12+g11*(a10+g11+ro10));
          1/24 = b1*(a10^2*g11+2*d10*g11^2-2*d10*g12+2*g11*h10*ro10))/2;
         1/24=b1*(g11^3+a10*(g11^2-g12)-2*g11*g12+g11^2*ro10-g12*ro10).
```

Развернутое приложение доступно по адресу http://math.zubanov.com.

Список литературы

1. Хайрер Э., Норсетт С., Ваннер Г., Решение обыкновенных дифференциальных уравнений. Нежесткие задачи. – М.: Мир, 1990.
2. Ширков П.Д. Одношаговые методы интегрирования систем ОДУ с точной оценкой локальной погрешности // Системный анализ в науке и образовании: электрон. науч. журнал. – Дубна, 2008. – №2. – [Электронный ресурс]. URL: http:/www.sanse.ru/archive/4.
3. Зубанов А.М., Ширков П.Д., Двухстадийные однократные ROW методы с комплексными коэффициентами для автономных систем ОДУ// Компьютерные исследования и моделирование. – М., 2010. – №1. – Вып. 2. – С. 19-32.
4. Зубанов А.М., Ширков П.Д., Методы типа Розенброка, *L*-эквивалентные неявным методам Рунге-Кутты // Сборник трудов 2-й Международной Конференции «Моделирование нелинейных процессов и систем». – М.: МГТУ СТАНКИН, 2012. – С. 10. (*принята в печать*).