1 **Московский Государственный Технический Университет**

**имени Н.Э.Баумана.**

**Кафедра САПР.**

**Расчетно-пояснительная записка к**

**дипломному проекту.**

**Тема: “Сравнительное тестирование математических ядер программ анализа динамики технических систем”.**

Выполнил

студент группы РК6-Д1

Плакин Д.Е.

Консультант

Маничев В.Б.

СОДЕРЖАНИЕ

АННОТАЦИЯ……………………………………………………………….…..…3

Техническое задание………..………………………………………………….…4

ВВЕДЕНИЕ………………..…………………………………………………..…...5

1. Цели тестирования………………………………………..…………………....6

2. Критерии надежности и эффективности программных

комплексов…………………………………….………………………….…..…7

2.1.Критерии надежности………..…………………………….. …………-

2.2. Критерии эффективности……………………..…..……………………9

3. Основные принципы подбора задач и проверка

программ моделирования с помощью них………...………………………11

4. Численные методы решения дифференциальных уравнений,

применяемые в математических пакетах.………………….………….…12

4.1.Методы Рунге-Кутты………………..…………………….……………-

4.2. Метод трапеций.…………………………………………………...…..13

4.3. Метод Адамса……………………………………………...…………..14

4.4. Метод Адамса-Мултона……………………...………...……………..-

5.Методы решения жестких систем дифференциальных уравнений в

системе DMAN.…………………….…………………………………………15

6. Методы решения жестких систем дифференциальных

уравнений в системе Matlab 5.2………………………………..………………..16

7. Описание тестовых задач………………………………………………..…….17

7.1. Линейные жесткие системы ОДУ………………………...………… 17

7.2. Линейные жесткие системы ОДУ с быстро осциллирующем решением..……………………………………………………………..24

7.3. Интегрирование систем ОДУ в обратном времени……………...… 26

7.4. Задачи с резко меняющимися свойствами функций-решений…….28

7.5. Задачи с разрывами производных функций-решений……………..30

7.6. Нелинейные жесткие задачи………………………………………....31

Результаты тестирования. ……………………………...………………………..32

Список использованной литературы…………………………...……………...33

**АННОТАЦИЯ.**

В пояснительной записке приведена попытка создания новой методики тестирования математических ядер анализа динамики технических систем, которая основана на решении систем обыкновенных дифференциальных уравнений (ОДУ) различного вида. Рассмотрены основные типы тестовых задач для данной методики тестирования – системы ОДУ, большинство из которых имеют известное аналитическое решение: жесткие системы, системы, решениями которых являются функции с разрывом производных, с быстроменяющимися свойствами и сильно осциллирующие решения, хаотические системы; интегрирование систем ОДУ в обратном времени; построение и интегрирование циклических систем. Тестирование и сравнение методов решения систем обыкновенных дифференциальных уравнений проводилось на программах: программ Matlab 5.2, МВТУ 3.0, DMAN, Pspice 8.0, ПА9. Приводятся значения максимальных относительных или абсолютных погрешностей. По результатам решения примеров определяются наиболее эффективные методы для задач соответствующих типов.

**Техническое задание.**

Сравнить методы решения систем обыкновенных дифференциальных уравнений программ Matlab 5.2, МВТУ 3.0, DMAN, Pspice 8.0, ПА9 на тестовых примерах различных типов. Разработать методику тестирования математических ядер программ анализа динамики технических систем на основе решения тестовых систем ОДУ. Составить тестовые задачи и провести их классификацию. Выявить наиболее эффективные методы для решения определенных классов задач.

**ВВЕДЕНИЕ.**

После написания отдельного программного продукта разработчику необходимо провести его тестирование для выявления ошибок и определения, удовлетворяет ли продукт поставленным требованиям. Тестирование программ анализа динамики технических систем проводится в двух направлениях: тестирование на конкретных известных практических задачах (электрические, механические, гидравлические схемы и т.д.) и тестирование математического ядра (системы дифференциально-алгебраических уравнений (ДАУ)). Поскольку основные характеристики (точность и скорость решения) зависят от математического ядра, его тестирование должно показывать надежность и эффективность программ анализа. В настоящее время первое направление развито достаточно широко, а второе применяется крайне редко, несмотря на то, что оно необходимо для дальнейшего усовершенствования существующих программ анализа и разработки новых. Разрабатываемая методика тестирования должна начать закрывать этот пробел.

Математическое ядро программ анализа технических систем представляет собой блок решения систем ДАУ, частным случаем которых являются системы обыкновенных дифференциальных уравнений (ОДУ). Все задачи для тестирования программ анализа условно можно разделить на «простые» и «трудные». С «простыми» задачами большинство программ анализа справляются без особых затруднений. К ним, например, относятся большинство нежестких систем ОДУ, имеющих решениями функции с плавно меняющимися свойствами. К «трудным» относятся задачи, вызывающие определенные сложности при их решении программами анализа. В данной работе практически всегда применялись «трудные» задачи, поскольку «простые» для тестирования представляют малый интерес. В качестве тестовых примеров по возможности выбирались системы ОДУ, имеющие аналитическое решение. В настоящей работе рассматриваются критерии тестирования программ анализа, классификация и набор тестовых задач, результаты тестирования программ Matlab 5.2, МВТУ 3.0, DMAN, Pspice 8.0, ПА9.

**1.ЦЕЛИ ТЕСТИРОВАНИЯ.**

Тестирование обычно проводится при разработке новых программ и при практическом использовании программ для решения задач из определенной прикладной области. При разработке новых программных комплексов тестирование необходимо для выявления неизвестных свойств методов, используемых в данном программном продукте моделирования. Проводя тестирование, разработчик обычно желает выяснить, как проявляются на практике свойства метода, установленные теоретически, проверить работоспособность алгоритмов оценки погрешности, изменения шага и порядка, сравнить эффективность новой и ранее созданных программ (общее время выполнения и т. п.) и т. д.

Во втором случае пользователь может не знать всех деталей метода, но ему известен класс задач и требуемая точность их решения. В этом случае пользователь обычно желает с помощью тестирования выбрать для своего класса одну программу, достаточно надежную и требующую наименьших затрат машинного времени.

**2. КРИТЕРИИ НАДЕЖНОСТИ И ЭФФЕКТИВНОСТИ ПРОГРАММНЫХ КОМПЛЕКСОВ.**

**2.1 Критерии надежности.**

Под достоверностью анализа и надежностью программы понимается ее способность с определенной точностью решить задачу до конца. Не имеет смысла рассматривать эффективность программы, игнорируя степень достоверности даваемого ею приближенного решения. В реальности разработчики не могут гарантировать надежную работу программы во всех возможных случаях. В этих условиях тестирование, хотя и не может создать каких-либо гарантий, оказывается полезным в двух отношениях. Во-первых, оно позволяет идентифицировать частные затруднения программ при решении определенных классов задач. Во-вторых, позволяет на достаточно большом наборе задач установить соотношение между задаваемой и действительно достигнутой точностями приближенного решения. Последнее важно потому, что в различных программах точность интерпретируется по-разному.

В большинстве случаев при тестировании программ под критерием надежности понимается величина наибольшего отклонения приближенного решения от точного на интервале интегрирования при определенной заданной погрешности. При этом важно определить, что брать за точное решение. Часто под точным решением понимается приближенное, полученное некоторой программой интегрирования для определенной заданной погрешности. Однако такой подход не является строгим. Во-первых, применение любой программы не гарантирует получения приближенного решения с погрешностью в заданных пределах. Во-вторых, решение может быть неединственным.

Помимо данных замечаний, необходимо также учитывать затруднения, возникающие при пополнении набора тестовых задач, когда программа, обеспечивающая решение, принимаемое за точное, вообще не справляется с интегрированием новой тестовой системы.

Метод можно считать надежным, если он позволяет получать решение с требуемой точностью во всех случаях, когда это возможно. Надежность можно “доказать” двумя способами: проведя тщательную проверку на исчерпывающем наборе задач и теоретически исследуя как общие вопросы аппроксимации, так и их конкретную реализацию в используемом алгоритме. Некоторые современные методы характеризуются сложной схемой вычислений, и трудно разработать их теорию, однако в этом направлении уже имеются некоторые успехи. Далее, метод должен быть разумен в том смысле, что он должен обеспечивать полученные решения только в случае правильно поставленной задачи.

Самый строгий подход состоит в использовании только таких тестовых задач, точные решения которых известны. Этот подход и реализован в проводимом тестировании программ анализа. При этом использованы два основных критерия надежности – наибольшие абсолютная и относительная погрешности. Учет абсолютной погрешности в дополнение к относительной позволяет получить более полное представление о точности приближенного решения, когда оно переходит через нуль или оказывается близким к нулю на тех отрезках интервала интегрирования, где близкие к нулю значения не являются характерными для решения в целом. В ходе интегрирования систем ДАУ численное решение сравнивалось с аналитическим, и определялась фактическая максимальная абсолютная (или относительная) погрешность интегрирования для всех дифференцируемых переменных. Большинство выбранных тестовых задач относятся к классу жестких систем ОДУ или систем ОДУ с многопериодным решением, которые затруднительно решать существующими стандартными программами интегрирования систем ОДУ.

**2.2. Критерии эффективности.**

Под эффективностью понимается требование по возможности меньших затрат машинного времени на решение задачи. Для пользователя наиболее существенным из обычно используемых критериев эффективности является время процессора, необходимое для решения задачи. Этот критерий характеризует работоспособность программы в целом, в совокупности учитывая все достоинства и недостатки метода, алгоритма и программы. Но этот критерий обладает серьезным недостатком. Он принципиально зависит от ЭВМ, на которой тестируется программа.

Другими критериями, более интересующими разработчика, чем пользователя, являются число шагов интегрирования, число вычислений функций и их якобиана в системе дифференциально-алгебраических уравнений (ДАУ), общее время процессора за вычетом времени, затраченного на вычисление функций и якобиана. Полезным является также критерий, показывающий число вызовов программы решения системы линейных алгебраических уравнений (ЛАУ).

Для того чтобы выяснить, каким методом можно за кратчайшее время решить ту или иную задачу, требуется всесторонняя проверка методов на различных модельных задачах.

Одним из фундаментальных показательных критериев для временной области программы моделирования является способность получить решение с требуемой точностью за приемлемое время для различных классов задач. Чтобы сравнить производительность различных программ моделирования необходимо предложить стандартный набор тестовых задач. Неудачно, что полученные результаты теста на различных программах моделирования не могут быть непосредственно сравнены друг с другом, т.к. программы моделирования выполнены на различных компьютерах с неизвестным исполнением. В связи с этим был предложен ряд компьютеронезависимых параметров.

Время процессора, например, адекватно только для сравнения программ моделирования, работающих на некоторых компьютерах. Рабочее время процессора зависит не только от производительности программ моделирования, но также и от скорости вычислений. Скорость вычислений обычно представляется скоростью в Мегафлопах(MFLOP).

Неудачно, что скорость MFLOP не находится в соответствие с полным представлением операций в компьютере. Производительность программ моделирования может изменяться даже на компьютерах с одинаковой скоростью MFLOP. Поэтому, скорость MFLOP следует заменить на эффективный параметр скорости Sc, экспериментально определяемый для каждого компьютера. Следует принять в расчет производительность компилятора и относительное исполнение компьютера в операциях с плавающей точкой.

Возможный кандидат для эффективной скорости – исполнение LINPACK, полученное для программного обеспечения различных компьютеров для решения стандартных линейных уравнений. Значения LINPACK для скорости компьютера исполнено как эффективный параметр скорости Sc.

Следующие 3 новых параметра представляют описание производительности во временной области:

Ns=Tcom\*Sc/Ns - эквивалентное число операций с плавающей точкой за один шаг интегрирования;

Nt=Tcom\*Sc/Tf - эквивалентное число операций с плавающей точкой за один интервал времени интегрирования;

Nn=Ns/n - - эквивалентное число операций с плавающей точкой за время одного шага интегрирования на один элемент эквивалентной цепи(Эквивалентная цепь – цепь, полученная из исходной цепи, в которой все компоненты заменяются их моделями.),

где

Tcom – время процессора на моделирование;

Sc – эффективный компонент скорости MFLOP;

Ns – число шагов интегрирования(включая отвергнутые шаги);

Tf – интервал интегрирования;

N – число элементов в эквивалентной цепи, за исключением основных;

Параметр Ns демонстрирует производительность численных процедур в решении нелинейных уравнений, включая все служебные процедуры. Nt описывает качество выбора шага по времени и временных модельных процедур. Третий параметр Nn принимает в счет влияние сложности цепи на рабочее время. Tcom\*Sn представляет некоторый эквивалент длины операций с плавающей точкой, занимающих часть времени прцессора.

**3.Основные принципы подбора задач и проверка программ моделирования с помощью них.**

Строгий контроль поведения погрешности возможен, только если известны точные решения тестовых задач. Это условие затрудняет выполнение другого требования, а именно: подбор задач для тестирования должен быть направлен на то, чтобы охватить по возможности шире все известные свойства систем ОДУ, которые могут существенно влиять на эффективность методов. Это требование важно как для разработчиков, так и для пользователей. У большинства тестов точные решения отсутствуют. Для разработчиков программ полезность таких тестов невелика, поскольку неясно, какое именно свойство или группа свойств вызывают затруднения у некоторых программ.

Крох приводит подробное описание схемы проверки методов. Идея его подхода состоит в том, что метод испытывают в различных предельных ситуациях и изучают его рабочие характеристики, как при этих, так и при промежуточных условиях. Рассмотрим основные особенности этой схемы.

Схема включает набор из четырнадцати задач, относящихся к ОДУ, причем каждая из этих задач выбрана по вполне определенным соображениям. В этом наборе имеются задачи, в которых на одной части отрезка интегрирования размер шага лимитируется достижением требуемой локальной точности, а на остальной части ограничения на шаг диктуются требованиями устойчивости. В одной задаче нужно часто менять шаг интегрирования, в другой длина шага должна оставаться постоянной. В некоторых задачах необходимо следить за абсолютной погрешностью, в других - за относительной.

Каждую задачу следует решать с заданной локальной точностью 1еS, S=1,0,-1,…,-20, на ЭВМ приблизительно с 18 десятичными разрядами. Внешние ограничения на величину шага интегрирования не накладываются, ибо проверяется, в частности, способность алгоритма самостоятельно выбирать шаг интегрирования (в некоторых программах требуется, чтобы максимальное значение шага интегрирования задавалось пользователем, однако во многих случаях пользователю очень трудно сделать правильный выбор). Вывод решения на печать требуется производить в трех точках отрезка интегрирования.

Статистической обработке подвергаются погрешность решения в каждой выдаваемой точке (абсолютная или относительная в соответствии с требованиями) и количество вычислений правых частей дифференциальных уравнений (производных) для каждой задачи и каждой заданной точности. Полученные данные позволяют ответить на ряд важных вопросов, среди которых можно указать следующие:

1.Меньше ли полная погрешность задаваемой пользователем допустимой погрешности?

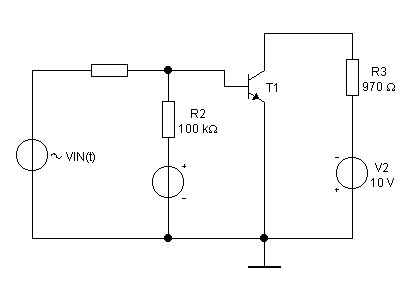
2.Непрерывно ли уменьшается полная погрешность при уменьшении задаваемой пользователем допустимой погрешности? Этот вопрос важен для оценки полной погрешности при практических расчетах, которая часто находится путем интегрирования уравнений с разной заданной точностью.

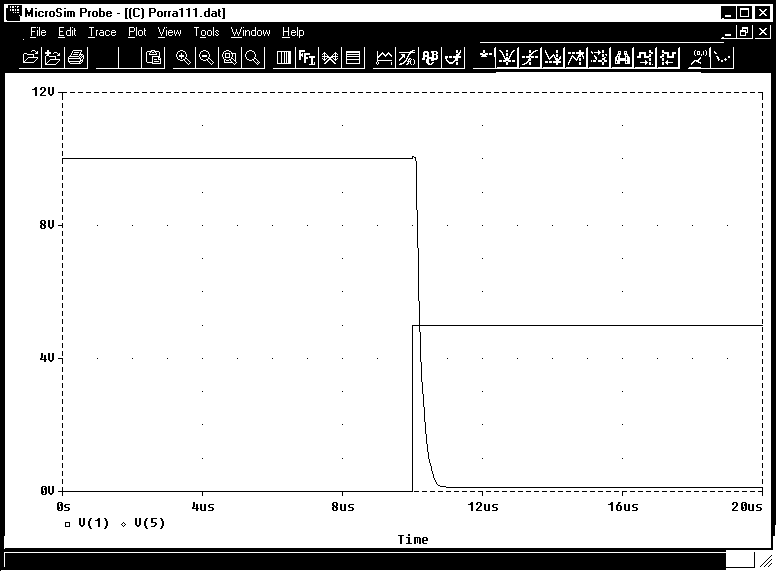
3.Правильно ли работает программа, когда требуется слишком высокая точность?

4.Устойчиво ли работает программа, когда требуется получить решение с очень малой точностью?

Если на указанные выше вопросы получены утвердительные ответы, то в известной мере можно быть уверенным в надежности и разумности метода. Идея Кроха – попытаться создать некий стандартный набор модельных задач и некую стандартную форму представления проверки – оказалась весьма плодотворной.

Большинство программ анализа тестируются на задачах из тех областей, в которых применяются. Например, программа PSpice тестировалась только на решении тестовых электронных схем (усилителей, инверторов и т. п.). Методика тестирования заключалась в следующем. Предлагался набор моделей базовых элементов электронных схем (источник тока p-n перехода, барьерная емкость, диффузионная емкость, диодов, транзисторов) с заданными аналитическими функциями и наборами внутренних параметров. Затем предлагались 36 электронных схем с графиками ожидаемых решений и рекомендуемыми таблицами со сравнительными оценочными параметрами программы. Рассмотрим пример такого теста:

Тест 1. Одиночный инвертор (резистивная нагрузка).

Рис 0.1. Принципиальная схема. Рис. 0.2. Графики результатов.

Результаты анализа.

Таблица 0.1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| V5(0),  В | V5(20),  мВ | V5max,  В | t(V5max)  мкс | t(9),  мкс | t(1),  мкс |
| 10 | 113.193 | 10.0804 | 10.0112 | 10.1107 | 10.478 |

Текст программы:

TEST1

VIN 1 0 PWL(0 0 10E-6 0 10.001E-6 5 10 5)

V1 0 2 10

V2 4 0 10

R1 1 3 10K

R2 2 3 100K

R3 5 4 970

Q1 5 3 0 Q2N696

.LIB D:\PLAKIN\PRAKTIKA\LIB\Bipolar.LIB

.PROBE V(1) V(5)

.OPTIONS LIMPTS=320000 RELTOL=1E-5 NUMDGT=10

.PRINT TRAN V(1) V(5)

.TRAN 0.001us 20us

.END

Но поскольку ядром любой из программ анализа является блок интегрирования систем ДАУ, описывающей моделируемые технические объекты, основной целью тестирования является проверка точности решения программой систем ДАУ.

Основными целями тестирования математических ядер программ анализа динамики технических систем были выделены следующие:

1. Определение методов, наиболее приспособленных для решения определенных классов задач по точности и по времени решения.
2. Определение разброса абсолютных значений переменных для данной программы.
3. Определение трудностей, возникающих у методов при решении определенных классов задач. (Решение не до конца, разрывы и большие изменения производных и т.п.).
4. Определение порядка зависимости времени решения систем ОДУ от числа в них уравнений.
5. Определение чувствительности программ к изменениям шагов интегрирования.

6. Определение правильности фазового портрета (для хаотических систем).

4.Некоторые численные методы решения систем ОДУ, применяемые в программах анализа.

**4.1.Методы Рунге-Кутты.**

Пусть требуется найти приближенное решение дифференциального уравнения y'=f(x,y), удовлетворяющее начальному условию y(x0)=y0. Численное решение задачи состоит в построении приближенного значения y1 решения уравнения y(x) в точке x1=x0 + h. Приближенное решение в точке x0+h можно вычислить, используя разложение точного решения в окрестности точки x0 по формуле Тейлора

y(x0+h)=y(x0)+y'(x0)h+...+ y(n-1) (x0)hn-1+O(hn) = y(x0)+hn(x0,y0,h). Расчетные формулы для приближенного решения можно получить, ограничившись первыми членами разложения:

y(x0+h) = y0+hL(x0,y0,h),

где L(x0,y0,h) = f(x0,y0)+...+hn-1f(n-1) /n!.

Эти формулы содержат производные от правых частей уравнения. Методами Рунге-Кутты называют группу одношаговых методов, в которых формулы для вычисления L получены из Тейлоровского разложения, но не содержат производных от правой части f. Наиболее просты для реализации явные методы Рунге- Кутты:

y(x0+h) = y0+hL(x0,y0,h),

где L(x0,y0,h) = c1k1+c2k2+ ... +cmkm,

k1 = f(x0,y0),

kr = f(x0+arh),

y(x0+h) = y0+h(brlkl+...+brr-1kr-1) r = 1, 2, ..., m.

Коэффициенты расчетных формул подбирают так, чтобы

L(0) = L'(0) = ... = L(s)(0) = 0.

Такие методы принято называть Метод Рунге-Кутты m-s. Локальная погрешность такого метода равна O(hs). Методы Рунге-Кутты не требуют вычисления дополнительных начальных точек и позволяют легко менять шаг. Наибольшее распространение получили явные методы Рунге-Кутты четвертого порядка (см. ниже).

Метод Рунге-Кутты 4-го порядка.

Пусть требуется найти приближенное решение дифференциального уравнения y'=f(x,y), удовлетворяющее начальному условию y(x0) = y0. Численное решение задачи состоит в построении приближенного значения y1 решения уравнения y(x) в точке x1 = x0 + h. Методом Рунге-Кутты четвертого порядка называют метод Рунге-Кутты, в котором приближенное решение в точке x0+h вычисляется по формулам:

y(x0+h) = y0+h(k1+2k2+2k3+k4)/6,

k1 = f(x0,y0),

k2 = f(x0+h/2,y0+hk1/2),

k3 = f(x0+h/2,y0+hk2/2),

k4 = f(x0+h/2,y0+hk3).

Локальная погрешность метода равна O(h4).

Метод Рунге-Кутты порядка 5.

Пусть требуется найти приближенное решение дифференциального уравнения y'=f(x,y), удовлетворяющее начальному условию y(x0) = y0. Численное решение задачи состоит в построении приближенного значения y1 решения уравнения y(x) в точке x1 = x0 + h. Методом Рунге-Кутты порядка 5 здесь названа модификация метода Рунге-Кутты 5-го порядка, в которой за счет одного дополнительного (шестого) вычисления правой части можно получить значение погрешности на одном шаге. Расчетные формулы этого метода громоздки и здесь не приводятся. Локальная погрешность - O(h5). Одним из выходных параметров алгоритма является значение погрешности. Метод Рунге-Кутты порядка 5 используется в алгоритме Рунге-Кутты с автоматическим выбором шага (см. ниже).

Метод Рунге-Кутты с автоматическим выбором шага.

Пусть требуется найти приближенное решение дифференциального уравнения y'=f(x,y), удовлетворяющее начальному условию y(x0) = y0. Численное решение задачи состоит в построении приближенного значения y1 решения уравнения y(x) в точке x1 = x0 + h. Алгоритм метода Рунге-Кутты с автоматическим выбором шага состоит в следующем. Задаем некоторое допустимое значение погрешности приближенного решения на одном шаге и весовые коэффициенты для каждого уравнения системы. По этому заданному значению погрешности находим нижние и верхние границы ошибки для каждого решения. Затем вычисляем решение методом Рунге-Кутты c оценкой погрешности. Если вычисленная погрешность меньше нижней границы ошибки, то шаг увеличивается, и решение в очередной точке вычисляется с новым, увеличенным шагом. Если же вычисленная погрешность больше верхней границы ошибки, шаг уменьшается, и вычисления повторяются с новым шагом. Уменьшение шага производится до тех пор, пока не будет достигнуто допустимое значение погрешности, либо пока шаг не станет меньшим заданного допустимого наименьшего значения.

**4.2. Метод трапеций.**

Пусть требуется найти приближенное решение дифференциального уравнения y'=f(x,y), удовлетворяющее начальному условию y(x0) = y0. Численное решение задачи состоит в построении приближенного значения y1 решения уравнения y(x) в точке x1 = x0 + h. Методом трапеций называют метод, в котором приближенное решение в точке x0+h вычисляется по формулам:

y(x0+h) = y0+h(f(x0,y0)+ f(x0+h,y0+h))/2,

Локальная погрешность метода равна O(h2).

4.3. Метод Адамса.

Пусть требуется найти приближенное решение дифференциального уравнения y'=f(x,y), удовлетворяющее начальному условию y(x0) = y0. Численное решение задачи состоит в построении приближенного значения y1 решения уравнения y(x) в точке x1 = x0 + h. Методами Адамса называют группу многошаговых методов, в которых приближенное решение yn+1=y(xn+1) в точке xn+1=x0+h(n+1) вычисляется по формуле, использующей полином P(x) наименьшей степени, интерполирующий правую часть f(x,y) по значениям fn, fn-1, ...,fn-k+1, fr = f(xr,yr). Методы, в которых P(x) = Pkn(x) называют k-шаговыми явными методами Адамса-Башфорта, а методы, в которых P(x) = Pk+1n+1 - (k+1)- шаговыми неявными методами Адамса-Мултона. Методы Адамса k-гопорядка требуют предварительного вычисления решения в k начальных точках. Здесь для вычисления дополнительных начальных значений использован метод Рунге-Кутты 4-4. Локальная погрешность методов Адамса k-го порядка - O(hk). Методы Адамса обладают лучшей, по сравнению с методами Рунге-Кутты устойчивостью.

4.4. Метод Адамса-Мултона

Пусть требуется найти приближенное решение дифференциального уравнения y'=f(x,y), удовлетворяющее начальному условию y(x0) = y0. Численное решение задачи состоит в построении приближенного значения y1 решения уравнения y(x) в точке x1 = x0 + h. В методе Адамса-Мултона 4-го порядка величины yi+1 вычисляют по формуле

yi+1 = yi+h(9f(xi+1,yi+1)-19f(xi,yi) - 5f(xi-1,yi-1)+f(xi-2, yi-2))/24. Локальная погрешность метода равна O(h5). Для вычисления f(xi+1,yi+1) нужно значение yi+1, которое пока неизвестно. Здесь сначала выполняется прогноз методом Адамса-Башфорта 4-го порядка, вычисляем y0i+1,а затем уточняем значение yi+1 по формуле Адамса-Мултона. Итерационный процесс уточнения

yi+1=yi+h(9f(xi+1,yi+1)-19f(xi,yi) -5f(xi-1, yi-1)+f(xi-2,yi-2))/ 24.

Локальная погрешность метода равна O(h5). Для вычисления f(xi+1,yi+1) нужно значение yi+1, которое пока неизвестно. Здесь сначала выполняется прогноз методом Адамса-Башфорта 4-го порядка, вычисляем y0i+1, а затем уточняем значение yi+1 по формуле Адамса-Мултона. Итерационный процесс уточнения заканчивается, когда |yni+1 - yn-1i+1| < ε, ε - заданная погрешность итераций. Для предварительного вычисления решения в 4 начальных точках здесь использован метод Рунге-Кутты 4-4. Неявные методы Адамса, как правило, позволяют производить вычисления с большим шагом, чем явные методы того же порядка.

**5. Краткое описание тестируемых программ.**

**5.1. Универсальная программа DMAN для решения систем дифференциально-алгебраических уравнений.**

FORTRAN-программа и С-программа DMAN предназначены для интегрирования систем ДАУ общего вида, не разрешенных относительно производных:

F(XP,X,Y,t)=0 (1)

где X - вектор дифференцируемых переменных системы (1) размерностью m; XP - вектор производных дифференцируемых переменных по времени размерностью m, т.е. XP = dX/dt; Y - вектор алгебраических переменных системы (1) размерностью l; t - независимая переменная (например, время); F - вектор-функция системы (1) размерностью n, где n=(m+l). Заданы начальные условия для вектора дифференцируемых переменных: X0=X(t0), где t0 - начальный момент времени интегрирования. Начальные значения остальных переменных, т.е. XP0 и Y0 рассчитываются в программе DMAN перед началом интегрирования автоматически. В процессе интегрирования система (1) на каждом шаге интегрирования автоматически дополняется системой:

G(XP,X,h)=0, (2)

где h - шаг интегрирования; G - вектор-функция размерностью m, которая соответствует выбранному методу интегрирования. Например, для неявного метода Эйлера G имеет вид:

G = XP - (1/h) \* (X - XN) = 0,

где XN - значение вектора дифференцируемых переменных в момент времени tn; tn - момент времени в начале шага интегрирования h; X - значение вектора дифференцируемых переменных в текущий момент времени t; h=(t-tn).

В программе DMAN реализованы 4 метода интегрирования систем ДАУ:

М1 - А-устойчивый неявный метод Эйлера первого порядка точности;

М2 - А-устойчивый неявный метод второго порядка точности;

М3 - А-устойчивый неявный метод четвертого порядка точности;

М4 - (экспериментальный) А-устойчивый неявный метод с автоматическим выбором методов М2 или М3 в ходе интегрирования в соответствии с требованиями точности и экономичности.

Во всех методах на каждом шаге интегрирования оценивается относительная, локальная погрешность интегрирования для каждой дифференцируемой переменной (по отношению к максимальному абсолютному значению каждой переменной на заданном отрезке интегрирования). Вычисленная погрешность сравнивается с заданной погрешностью EPS и если она ее превышает, то шаг интегрирования h уменьшается. Максимальное абсолютное значение каждой дифференцируемой переменной определяется в ходе интегрирования автоматически, эти значения можно также задать перед началом интегрирования.

Во всех методах на каждом шаге интегрирования решается система нелинейных алгебраических уравнений относительно переменных X, XP, Y методом Ньютона, для которого необходимо вычислять матрицу Якоби для системы (1) и системы (2). Матрица Якоби для системы (2) вычисляется в программе DMAN автоматически. Матрица Якоби для системы (1) должна вычисляться в специальной подпрограмме пользователя FCT, наряду с вычислением вектор-функции F. Эта матрица состоит из двух подматриц:

RJ = (RJ1, RJ2 ) ,

где RJ - матрица Якоби размером (n\*n1); n1=m+n. RJ1 - матрица частных производных вектор-функции F по переменным XP размером (n\*m), т.е. матрица RJ1 = [dF/dXP]. RJ2 - матрица частных производных вектор-функции F по переменным Z размером (n\*n), т.е. матрица RJ2 = [dF/dZ], где Z - вектор переменных системы ОДУ размерностью n, который включает дифференцируемые переменные и алгебраические переменные системы (1), т.е.

Z = (X,Y).

В качестве примера приведем подпрограмму для решения тестов 1 и 2:

/\* tests a2 and a3 from paper [3] \*/

#include "dman.c"

/\*variables for task\*/

static double c1,c2,c3,l1,l2,l3,a,r1,r2,r3;

/\*variables for task\*/

void main()

{

static int ii,iii;

f2=fopen("dmga3.rez","wt");

for (ii=1;ii<=1;ii++)

{

if(ii==1) a=0.001;

if(ii==2) a=0.999;

c1=1.;

c2=1.5;

c3=1.;

l1=-1e5;

l2=-1.;

l3=-1e2;

r1=fabs(1./l1);

r2=fabs(1./l2);

r3=fabs(1./l3);

eps=1e-2;

fprintf(f2," relative tolerance - eps=%.4g\n\n",eps);

printf(" relative tolerance - eps=%.4g\n\n",eps);

for (nm=1;nm<=1;nm++)

{

fprintf(f2," number of method - nm=%d\n",nm);

printf(" number of method - nm=%d\n",nm);

t0=0.;

tk=10e9;

hmn=1e-7;

hmx=tk/8.;

ncon=0;

n=3;

m=3;

z[1]=c1+c2+c3;

z[2]=c1\*a+c2-c3;

z[3]=c1\*a+c2+c3;

z1[1]=c1+c2+c3;

z1[2]=c1\*a+c2-c3;

z1[3]=c1\*a+c2+c3;

fprintf(f2,"time x(1) eps x(1) maxeps x(1) x(2) eps x(2) maxeps x(2) x(3) eps x(3) maxeps x(3)\n");

printf("time x(1) eps x(1) maxeps x(1) x(2) eps x(2) maxeps x(2) x(3) eps x(3) maxeps x(3)\n");

dman(z,xp,z1,xp1,f,rj1,rj2,t,t0,tk,h,hmn,hmx,eps,tkv,ar,n,m,nm,ncon,&nbad,&ier,ip);

for(iii=1;iii<=16;iii++)fprintf(f2,"%d ",ip[iii]);

fprintf(f2,"\n");

printf("\n ier= %d \n",ier);

fprintf(f2,"\n ier= %d \n",ier);

fprintf(f2,"\n ii= %d \n",ii);

}

}

fclose (f2);

}

void out(double z[],double xp[],int n,int m,double t,double t0,double tk,double h,double tkv,double ar[],int ncon,int ip[])

{

static double em[5],ea[5],pa[5];

static double zr1,zr2,zr3;

static long io;

printf ("n=%d,m=%d,ncon=%d\n",n,m,ncon);

if(ncon!=0) goto m1;

for(io=1;io<=3;io++) em[io]=0.;

m1:

zr1=exp(l1\*t);

zr2=exp(l2\*t);

zr3=exp(l3\*t);

if((l1\*t)<-70) zr1=0.;

if((l2\*t)<-70) zr2=0.;

if((l3\*t)<-70) zr3=0.;

ea[1]=c1\*zr1+c2\*zr2+c3\*zr3;

ea[2]=c1\*a\*zr1+c2\*zr2-c3\*zr3;

ea[3]=c1\*a\*zr1+c2\*zr2+c3\*zr3;

for (io=1;io<=n;io++)

{

pa[io]=fabs(z[io]-ea[io]);

if(em[io]>pa[io]) goto m3;

em[io]=pa[io];

m3:

continue;

}

fprintf(f2,"\nt=%14.5e\n",t);

for(io=1;io<=n;io++)fprintf(f2,"z[i]=%14.5eea[i]=%14.5e em[i]=%14.5e\n",z[io],ea[io],em[io]);

printf("\nt=%14.5e\n",t);

for(io=1;io<=n;io++)printf("z[i]=%14.5e ea[i]=%14.5e em[i]=%14.5e\n",z[io],ea[io],em[io]);

fprintf(f2,"\n");

printf("\n");

fprintf(f2,"t=%14.5e h=%14.5e tk=%14.5e\n",t,h,tk);

for(io=1;io<=16;io++)fprintf(f2,"%d ",ip[io]);

fprintf(f2,"\n");

printf("t=%14.5e h=%14.5e tk=%14.5e\n",t,h,tk);

for(io=1;io<=16;io++)printf("%d ",ip[io]);

printf("\n");

return;

}

void fct(double z[],double xp[],double f[],double rj1[50][50],double rj2[50][50],int n,int m,double t,double h,int ncon,int \*nbad,int ip[])

{

static double b,g,a1,a11,a12,a13,a21,a22,a23,a31,a32,a33;

b=0.5\*(l2+l3);

g=0.5\*(l2-l3);

a1=1/(1-a);

a11=a1\*(l1-a\*l2);

a12=g;

a13=a1\*(b+a\*g-l1);

a21=a\*a1\*(l1-l2);

a22=b;

a23=a1\*(g+a\*b-a\*l1);

a31=a\*a1\*(l1-l2);

a32=g;

a33=a1\*(b+a\*g-a\*l1);

f[1]=xp[1]-a11\*z[1]-a12\*z[2]-a13\*z[3];

f[2]=xp[2]-a21\*z[1]-a22\*z[2]-a23\*z[3];

f[3]=xp[3]-a31\*z[1]-a32\*z[2]-a33\*z[3];

rj1[1][1]=1;

rj1[2][2]=1;

rj1[3][3]=1;

rj2[1][1]=-a11;

rj2[1][2]=-a12;

rj2[1][3]=-a13;

rj2[2][1]=-a21;

rj2[2][2]=-a22;

rj2[2][3]=-a23;

rj2[3][1]=-a31;

rj2[3][2]=-a32;

rj2[3][3]=-a33;

return;}

**5.2. Программный комплекс «МОДЕЛИРОВАНИЕ В ТЕХНИЧЕСКИХ УСТРОЙСТВАХ»** (ПК «МВТУ, версия 3.0)

Программный комплекс «Моделирование в технических устройствах» («МВТУ») - современная среда интеллектуального САПР, предназначенная для детального исследования и анализа динамических процессов в ядерных и тепловых энергетических установках, в системах автоматического управления (САУ), в следящих приводах и роботах, в любых технических системах, описание динамики которых может быть реализовано методами структурного моделирования.

Может использоваться для моделирования нестационарных процессов в физике, в электротехнике, в динамике машин и механизмов, в астрономии и т.д., а также для решения нестационарных краевых задач (теплопроводность, гидродинамика и др).

Может функционировать в многокомпьютерных моделирующих комплексах, в том числе и в режиме удаленного доступа к технологическим и информационным ресурсам.

Является альтернативой программным продуктам MATRIXx, Simulink, VisSim и др.

Программный комплекс “МВТУ” реализует следующие режимы работы:

МОДЕЛИРОВАНИЕ, обеспечивающий:

* моделирование нестационарных процессов в непрерывных, дискретных и гибридных технических системах, в том числе и при наличии обмена данными (синхронный или асинхронный) с внешними программами и устройствами;
* редактирование параметров структурной схемы и расчета в режиме “on-line”;
* расчет в реальном времени или в режиме масштабирования модельного времени;
* рестарт, архивацию и воспроизведение результатов моделирования.

ОПТИМИЗАЦИЯ, позволяющий решать задачи:

* параметрической оптимизации САУ и идентификации опытных данных;
* cинтеза оптимальных регуляторов и оптимального управления в многокритериальной постановке при наличии ограничений на значения динамических переменных, управляющих воздействий, параметров элементов системы автоматического управления, функционалов качества.

АНАЛИЗ, обеспечивающий:

* расчет амплитудно-фазовых частотных характеристик для любой линейной и большинства нелинейных систем (ЛАХ, ФЧХ, различные годографы и др.);
* расчет коэффициентов, полюсов и нулей передаточных функций.

КОНТРОЛЬ И УПРАВЛЕНИЕ, позволяющий:

* создавать электронные аналоги измерительных приборов и управляющих устройств для оперативного контроля и управления переходными процессами;
* выполнять статистическую обработку сигналов (в том числе и внешних), основанную на быстром преобразовании Фурье.

Программный комплекс «МВТУ» имеет следующие достоинства:

 **открытость** за счет реализации в ПК «МВТУ» нескольких механизмов обмена данными с внешними расчетными программами, а также за счет встроенного в ПК интерпретатора математических функций;

 **принцип вложенности** структур (глубина вложенности неограниченная), что особо актуально при моделировании сложных динамических систем;

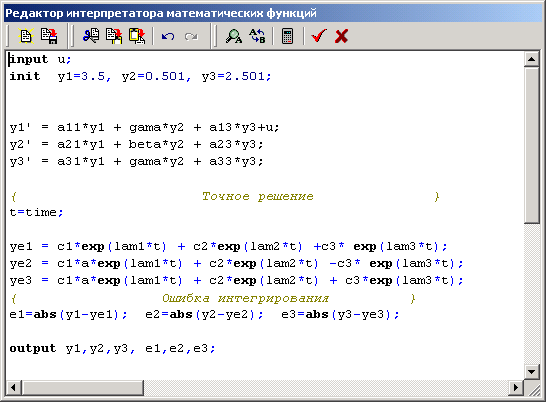
 **векторизация** алгоритмов передачи и обработки данных за счет реализации линий связи типа «шина» данных и векторизации входов/выходов всех типовых блоков;

 наличие **наиболее полной** *Общетехнической* и ряда *Специализированных* библиотек типовых блоков, в т.ч. библиотеки теплофизических свойств основных рабочих тел;

 наличие библиотеки ***Контроль*** *и управление*, что позволяет формировать в ПК «МВТУ» панели (щиты) приборов для отображения и оперативного управления моделируемой системой в процессе расчета;

 **16 алгоритмов интегрирования**, включая 10 новых эффективных алгоритмов (5 явных и 5 неявных) для **жестких** систем дифференциальных уравнений: Эйлера, Рунге-Кутты классический, Рунге-Кутты модифицированный, Мерсона классический, Мерсона модифицированный, Адаптивный 1, Адаптивный 2, Адаптивный 3, Адаптивный 4, Адаптивный 5 (явные адаптивные многошаговые методы переменного порядка (от 1 до 6)), Адаптивный неявный, Диагонально неявный, Гира (многошаговый метод Гира переменного порядка (от 1 до 6), основанный на формулах дифференцирования назад), Эйлера неявный, DIRK33 и DIRK44 (диагонально неявные FSAL‑методы Рунге‑Кутты порядка 3 и 4, соответственно);

 функционирование в **любой** версии WINDOWS, наличие подробной контекстной справочной системы, **эффективность** в отраслевых разработках и учебном процессе.

Система ОДУ из теста 1 задается в МВТУ 3.0 следующим образом:

**Рис. 5.1.**

**6.Методы решения жестких систем дифференциальных уравнений в системе Matlab 5.2.**

**Ode45** - одношаговый метод Дорманда-Принса 5-го порядка основан на явных методах Рунге-Кутты 4 и 5 порядков. Применяется для решения нежестких задач.

**Ode23** - одношаговый метод Богацки-Шампайна 3-го порядка основан на явных методах Рунге-Кутты 2 и 3 порядков.

**Ode113** - многошаговый метод переменного порядка (от 1 до 12) основан на многошаговых формулах Адамса.

Вышеупомянутые алгоритмы предназначены для решения нежестких систем .

**Ode15s** - многошаговый метод переменного порядка (от 1 до 5), основанный на формулах численного дифференцирования назад.

**Ode23s** –одношаговый метод, основанный на изменяемой формуле Розенброка 2 порядка .

**Ode23t -** метод Рунге - Кутты 3-го порядка, приспособленный для решения жестких систем.

**Ode23tb** - диагонально неявный метод Рунге-Кутты с первым шагом по формуле трапеций и вторым шагом по формулам дифференцирования назад 2 порядка.

**7. Описание тестовых задач.**

**7.1. Линейные жесткие системы ОДУ.**

Построение тестов из данного пункта приведено на листе 2.

Большинство задач имеют решения, быстро монотонно стремящиеся к 0. Поэтому программы должны очень быстро увеличивать свой шаг интегрирования. В действительности, при задании конечного времени 1010 только программа DMAN и 5 методов программы МВТУ (диагонально неявный, Гира, Эйлера неявный, DIRK33 и DIRK44) смогла получить решение за приемлемое время.

**Тест 1.**

Линейная система ОДУ А2 из /1/ c постоянными коэффициентами и постоянной диагональной матрицей.

Исходная система ОДУ:

с1=с3=1

с2=1.5

a=0.001

l1=-105

l2=-102

l3=-1

b=(l2+l3)/2

g=(l2-l3)/2

a1=1/(1-a)

a11=a1\*(l1-a\*l2)

a12=g

a13=a1\*(b+a\*g-l1)

a21=a\*a1\*(l1-l2)

a22=b

a23=a1\*(g+a\*b-a\*l1)

a31=a\*a1\*(l1-l2)

a32=g

a33=a1\*(b+a\*g-a\*l1)

y1’=a11\*y1+a12\*y2+a13\*y3

y2’=a21\*y1+a22\*y2+a23\*y3

y3’=a31\*y1+a32\*y2+a33\*y3

Начальные условия:

y1=c1+c2+c3

y2 =c1\*a+c2-c3

y3=c1\*a+c2+c3

Аналитическое решение:

y1=с1\*el1\*x+ с2\*el2\*x+ с3\*el3\*x

y2= с1\*a\*el1\*x+ с2\*el2\*x -с3\*el3\*x

y3= с1\*a\*el1\*x+ с2\*el2\*x+ с3\*el3\*x

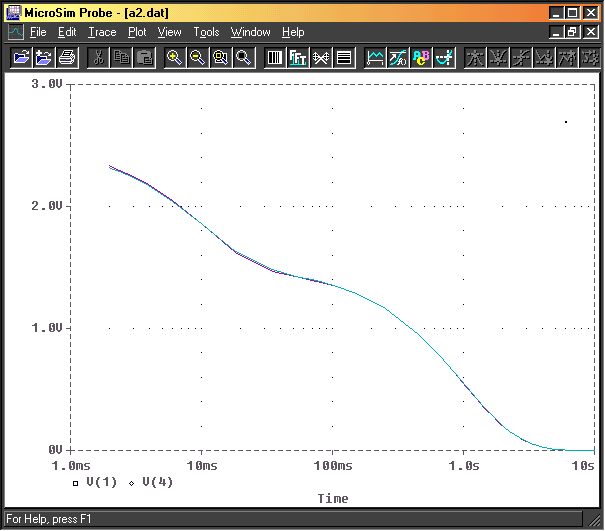
Интервал интегрирования: [0,10].

Рис. 1.1 График численного и аналитического решения решения y1 (Pspice 8.0).

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой Matlab 5.2.

### Таблица 1.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | ode23s | ode23 | ode23t | ode23tb | ode45 | ode15s | ode113 |
| 10-3 | 8.076\*10-4 | - | 6.186\*10-4 | 5.78\*10-4 | - | 1.03\*10-3 | - |
| 10-4 | 1.89\*10-4 | - | 1.5\*10-4 | 1.27\*10-4 | - | 8.15\*10-5 | - |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой МВТУ 3.0.

##### Таблица 1.2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | Эйлера | Рунге-Кутты 4-5  классич. | Рунге-Кутты 4-5  модифиц. | Мерсона  классич. | Мерсона  модифиц. | Адаптив-  ный 1 | Адаптив-  ный 2 | Адаптив-  ный 3 |
| 10-3 | - | 2000 | 0.029 | 1.5\*10-4 | 6.8\*10-4 | 0.006 | 0.0067 | 7\*10-4 |
| 10-4 | - | 0.041 | 0.0015 | 1.3\*10-5 | 8\*10-5 | 0.0016 | 0.0019 | 1.9\*10-4 |
|  | Адаптив-  ный 4 | Адаптив-  ный 5 | Адаптив-  ный неявный | Диаго-  нально-  неявный | Гира | Эйлера  неявный | DIRK 33 | DIRK 44 |
| 10-3 | 4.4\*10-4 | 0.0015 | 6\*10-4 | 0.0015 | 0.0028 | 0.015 | 3\*10-4 | 3.8\*10-5 |
| 10-4 | 2.9\*10-5 | 2.5\*10-4 | 8.5\*10-5 | 2.6\*10-4 | 5\*10-4 | 0.0041 | 6\*10-5 | 5\*10-6 |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой DMAN.

##### Таблица 1.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | М2 | | М3 | |
| FORTRAN | C | FORTRAN | C |
| 10-3 | 2.38\*10-4 | 2.44\*10-4 | 1.24\*10-6 | 1.20\*10-6 |
| 10-4 | 5.75\*10-5 | 5.79\*10-5 | 2.84\*10-7 | 2.17\*10-7 |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами систем Pspice 8.0., ПА9, ПА7.

Таблица 1.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | Pspice 8.0 | ПА9 | | ПА7 |
| Euler | Trapec | MAN 12s |
| 10-3 | 0.019 | 0,0176 | 0,28 |  |
| 10-4 | 0.019 | 0,004 | 0,13 |  |

**Тест 2.**

Линейная система ОДУ А3 из /1/ c постоянными коэффициентами и сильно связанными компонентами. Система отличается от предыдущей значением параметра a=0.999. Матрица системы близка к вырожденной, поскольку ее собственные вектора, соответствующие собственным значениям l1 и l2, отличаются незначительно. Все компоненты решения связаны между собой одинаково. Степень связи компонент, зависящая от собственных векторов матрицы системы, оказывает существенное влияние на надежность и эффективность явных нелинейных методов интегрирования.

Все остальные условия такие, как в тесте 1.

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой Matlab 5.2.

### Таблица 2.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | ode23s | ode23 | ode23t | ode23tb | ode45 | ode15s | ode113 |
| 10-3 | 1.478 | - | 5.9\*10-4 | 5.6\*10-4 | - | 9.83\*10-4 | - |
| 10-4 | 1.23 | - | 1.41\*10-4 | 1.22\*10-4 | - | 8.52\*10-5 | - |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой МВТУ 3.0.

##### Таблица 2.2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | Эйлера | Рунге-Кутты 4-5  классич. | Рунге-Кутты 4-5  модифиц. | Мерсона  классич. | Мерсона  модифиц. | Адаптив-  ный 1 | Адаптив-  ный 2 | Адаптив-  ный 3 |
| 10-3 | - | 780 | 0.0175 | 5.7\*10-5 | 3.5\*10-5 | 0.0055 | 0.0061 | 0.0017 |
| 10-4 | - | 0.083 | 0.0017 | 2.2\*10-5 | 3.2\*10-5 | 0.0019 | 0.0024 | 3.8\*10-4 |
|  | Адаптив-  ный 4 | Адаптив-  ный 5 | Адаптив-  ный неявный | Диаго-  нально-  неявный | Гира | Эйлера  неявный | DIRK 33 | DIRK 44 |
| 10-3 | 3\*10-4 | 0.008 | 0.013 | 0.0045 | 0.0028 | 0.0124 | 3.5\*10-4 | 1.7\*10-4 |
| 10-4 | 3.8\*10-5 | 0.026 | 0.0014 | 5.9\*10-4 | 5\*10-4 | 0.004 | 5.7\*10-5 | 3.9\*10-6 |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой DMAN.

##### Таблица 2.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | М2 | | М3 | |
| FORTRAN | C | FORTRAN | C |
| 10-3 | 1.59\*10-4 | 1.49\*10-4 | 1.03\*10-6 | 8.84\*10-7 |
| 10-4 | 3.8\*10-5 | 3.58\*10-5 | 1.03\*10-6 | 9.22\*10-7 |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами систем Pspice 8.0., ПА9, ПА7.

Таблица 2.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | Pspice 8.0 | ПА9 | | ПА7 |
| Euler | Trapec | MAN 12s |
| 10-3 | 0.019 | 0,01 | 0,13 |  |
| 10-4 | 0.019 | 0,003 | 0,13 |  |

**Тест 3.**

Линейная система ОДУ А5 из /1/ c переменными коэффициентами, одно собственное значение системы в начале интервала интегрирования положительно.

Исходная система ОДУ:

l=-0.9531

s=1

p=-0.1

l1=(s-p)\*(t\*l+1)\*el\*t-s.

l2=-102

l3=-1

с1=с3=1

с2=1.5

a=0.001

b=(l2+l3)/2

g=(l2-l3)/2

a1=1/(1-a)

a11=a1\*(l1-a\*l2)

a12=g

a13=a1\*(b+a\*g-l1)

a21=a\*a1\*(l1-l2)

a22=b

a23=a1\*(g+a\*b-a\*l1)

a31=a\*a1\*(l1-l2)

a32=g

a33=a1\*(b+a\*g-a\*l1)

y1’=a11\*y1+a12\*y2+a13\*y3

y2’=a21\*y1+a22\*y2+a23\*y3

y3’=a31\*y1+a32\*y2+a33\*y3

Начальные условия:

y1=c1+c2+c3

y2 =c1\*a+c2-c3

y3=c1\*a+c2+c3

Аналитическое решение:

y1=с1\*el1\*x+ с2\*el2\*x+ с3\*el3\*x

y2= с1\*a\*el1\*x+ с2\*el2\*x -с3\*el3\*x

y3= с1\*a\*el1\*x+ с2\*el2\*x+ с3\*el3\*x

Интервал интегрирования: [0,10].

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой Matlab 5.2.

### Таблица 3.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | ode23s | ode23 | ode23t | ode23tb | ode45 | ode15s | ode113 |
| 10-3 | 8.08\*10-4 | - | 6.19\*10-4 | 5.84\*10-4 | - | 7.43\*10-4 | - |
| 10-4 | 1.9\*10-4 | - | 1.5\*10-4 | 1.27\*10-4 | - | 9.41\*10-5 | - |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой МВТУ 3.0.

##### Таблица 3.2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | Эйлера | Рунге-Кутты 4-5  классич. | Рунге-Кутты 4-5  модифиц. | Мерсона  классич. | Мерсона  модифиц. | Адаптив-  ный 1 | Адаптив-  ный 2 | Адаптив-  ный 3 |
| 10-3 | - | 4.9 | 0.0011 | 5\*10-5 | 2\*10-4 | 0.0045 | 0.0017 | 0.0012 |
| 10-4 | - | 0.085 | 3.7\*10-4 | 4.1\*10-5 | 10-5 | 0.0018 | 6.1\*10-4 | 2.1\*10-4 |
|  | Адаптив-  ный 4 | Адаптив-  ный 5 | Адаптив-  ный неявный | Диаго-  нально-  неявный | Гира | Эйлера  неявный | DIRK 33 | DIRK 44 |
| 10-3 | 1.7\*10-4 | 0.0023 | 0.0012 | 0.0019 | 0.002 | 0.015 | 3.1\*10-4 | 3.1\*10-5 |
| 10-4 | 2.7\*10-5 | 6\*10-4 | 2.7\*10-6 | 3.7\*10-4 | 7.8\*10-4 | 0.0041 | 5.3\*10-5 | 10-5 |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой DMAN.

##### Таблица 3.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | М2 | | М3 | |
| FORTRAN | C | FORTRAN | C |
| 10-3 | 3.91\*10-4 | 5.93\*10-4 | 8.89\*10-7 | 1.26\*10-6 |
| 10-4 | 9.61\*10-5 | 8.87\*10-5 | 4.05\*10-7 | 3.74\*10-7 |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами систем Pspice 8.0., ПА9, ПА7.

Таблица 3.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | Pspice 8.0 | ПА9 | | ПА7 |
| Euler | Trapec | MAN 12s |
| 10-3 | 0.3 | 1 | 1 |  |
| 10-4 | 0.3 | 1 | 1 |  |

**Тест 4.**

Линейная система ОДУ А6 из /1/ c переменными коэффициентами, одно собственное значение системы по модулю возрастает. Тест отличается от предыдущего значениями параметров

l=-102

s=104

p=10-4

l2=-1

l3=-102

Одно собственное значение возрастает по модулю с –10е-4 до –10е4. Все компоненты решения положительны.

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой Matlab 5.2.

### Таблица 4.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | ode23s | ode23 | ode23t | ode23tb | ode45 | ode15s | ode113 |
| 10-3 | 1.21\*10-3 | - | 1.35\*10-3 | 1.15\*10-3 | - | 5.89\*10-4 | - |
| 10-4 | 2.41\*10-4 | - | 2.95\*10-4 | 2.19\*10-4 | - | 1.21\*10-4 | - |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой МВТУ 3.0.

##### Таблица 4.2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | Эйлера | Рунге-Кутты 4-5  классич. | Рунге-Кутты 4-5  модифиц. | Мерсона  классич. | Мерсона  модифиц. | Адаптив-  ный 1 | Адаптив-  ный 2 | Адаптив-  ный 3 |
| 10-3 | - | 1650 | 0.011 | 1.8\*10-4 | 3.3\*10-4 | 0.0048 | 0.0055 | 0.0035 |
| 10-4 | - | 0.02 | 0.0031 | 2.1\*10-5 | 2.3\*10-5 | 0.0014 | 0.0018 | 2.4\*10-4 |
|  | Адаптив-  ный 4 | Адаптив-  ный 5 | Адаптив-  ный неявный | Диаго-  нально-  неявный | Гира | Эйлера  неявный | DIRK 33 | DIRK 44 |
| 10-3 | 7\*10-4 | 0.0014 | 0.002 | 0.0038 | 0.0078 | 0.02 | 8\*10-4 | 2.2\*10-4 |
| 10-4 | 5.6\*10-5 | 2.7\*10-4 | 6\*10-5 | 8.1\*10-4 | 0.0014 | 0.0071 | 1.5\*10-4 | 1.9\*10-5 |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой DMAN.

##### Таблица 4.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | М2 | | М3 | |
| FORTRAN | C | FORTRAN | C |
| 10-3 | 9.55\*10-4 | 7.4\*10-4 | 2.05\*10-6 | 2.63\*10-6 |
| 10-4 | 2.42\*10-4 | 1.87\*10-4 | 5.36\*10-7 | 3.45\*10-7 |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами систем Pspice 8.0., ПА9, ПА7.

Таблица 4.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | Pspice 8.0 | ПА9 | | ПА7 |
| Euler | Trapec | MAN 12s |
| 10-3 | 0.118 | 1 | 1 |  |
| 10-4 | 0.118 | 1 | 1 |  |

**Тест 5.**

Линейная система ОДУ А7 из /1/ c переменными коэффициентами, одно собственное значение системы по модулю убывает. Пример отличается от предыдущего значениями параметров

l=-50

s=1

p=104

l2=-104

l3=-1

Одно собственное значение убывает по модулю с –10-4 до –104.

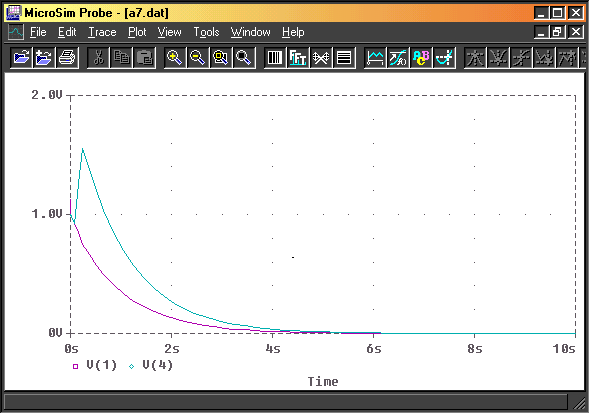


Рис. 5.1 График численного и аналитического решения решения y1 (Pspice 8.0).

Ни один из методов системы Matlab 5.2 не смот решить данную систему (по крайней мере за приемлемое время)

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой МВТУ 3.0.

##### Таблица 5.1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | Эйлера | Рунге-Кутты 4-5  классич. | Рунге-Кутты 4-5  модифиц. | Мерсона  классич. | Мерсона  модифиц. | Адаптив-  ный 1 | Адаптив-  ный 2 | Адаптив-  ный 3 |
| 10-3 | - | 3,5\*1016 | 7,4\*1025 | 2,6\*1016 | 6\*1024 | 1,2\*1022 | 2,5\*1027 | 3,7\*1024 |
| 10-4 | - | 4,6\*1016 | 1,7\*1025 | 1,8\*1016 | 1,2\*1021 | 4,4\*1020 | 2,4\*1026 | 5,1\*1020 |
|  | Адаптив-  ный 4 | Адаптив-  ный 5 | Адаптив-  ный неявный | Диаго-  нально-  неявный | Гира | Эйлера  неявный | DIRK 33 | DIRK 44 |
| 10-3 | 5\*1027 | 220 | 0,65 | 0,7 | - | 0,75 | - | 0,65 |
| 10-4 | 1,9\*1027 | 8\*1024 | 0,75 | 0,75 | 0,75 | 0,75 | 1,6\*1028 | 0,7 |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой DMAN.

##### Таблица 5.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | М2 | | М3 | |
| FORTRAN | C | FORTRAN | C |
| 10-3 | 0,767 | 0,77 | 0,77 | 0,77 |
| 10-4 | 0,775 | 0,775 | 0,771 | 0,76 |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами систем Pspice 8.0., ПА9, ПА7.

Таблица 5.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | Pspice 8.0 | ПА9 | | ПА7 |
| Euler | Trapec | MAN 12s |
| 10-3 | 0.75 | 0,75 | 0,77 |  |
| 10-4 | 0.75 | 0,75 | 0,77 |  |

Сравнение методов решения ОДУ тестируемых программ показало, что программа Pspice плохо пригодна для решения линейных жестких систем ОДУ. Для решения таких задач лучше всего подходят программы: DMAN(методы М2,М3), Matlab 5.2 (методы ode23s, ode23t, ode23tb,ode15s) и МВТУ(методы Мерсона классический и модифицированный, Адаптивный 4, DIRK 33 и DIRK 44).

**7.2. Линейные жесткие системы ОДУ с быстро осциллирующем решением.**

**Тест 6.**

Линейная система ОДУ с колебательным решением из работы /1/ - задача B3. Система c сильно осциллирующем решением.

Исходная система ОДУ:

a=-1

b=100

y1’=a\*y1+b\*y2

y2’=-b\*y1+a\*y2

Начальные условия:

y1=3

y2 =1

Аналитическое решение:

y1=(3\*cos(b\*x)+sin(b\*x))\* ea\*x

y2=(cos(b\*x)-3\*sin(b\*x))\* ea\*x

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой Matlab 5.2.

### Таблица 6.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | ode23s | ode23 | ode23t | ode23tb | ode45 | ode15s | ode113 |
| 10-3 | 5.08\*10-2 | 1.95\*10-2 | 4.63\*10-2 | 3.85\*10-2 | 3.85\*10-3 | 1.25\*10-2 | 3.83\*10-3 |
| 10-4 | 1.04\*10-2 | 1.87\*10-3 | 9.58\*10-3 | 7.62\*10-3 | 2.85\*10-4 | 3.91\*10-3 | 4.17\*10-4 |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой МВТУ 3.0.

##### Таблица 6.2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | Эйлера | Рунге-Кутты 4-5  классич. | Рунге-Кутты 4-5  модифиц. | Мерсона  классич. | Мерсона  модифиц. | Адаптив-  ный 1 | Адаптив-  ный 2 | Адаптив-  ный 3 |
| 10-3 | 2.2\*1020 | 0.026 | 0.012 | 0.0075 | 0.0075 | 0.05 | 0.026 | 0.024 |
| 10-4 | 2.9\*1020 | 0.0018 | 0.0027 | 9\*10-4 | 0.0023 | 0.007 | 0.005 | 0.0028 |
|  | Адаптив-  ный 4 | Адаптив-  ный 5 | Адаптив-  ный неявный | Диаго-  нально-  неявный | Гира | Эйлера  неявный | DIRK 33 | DIRK 44 |
| 10-3 | 0.0022 | 2\*10-13 | 5.4\*10-4 | 0.18 | 0.22 | 1.3 | 0.069 | 0.006 |
| 10-4 | 1.3\*10-4 | 2\*10-13 | 2.6\*10-5 | 0.034 | 0.026 | 0.66 | 0.012 | 5\*10-4 |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой DMAN.

##### Таблица 6.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | М2 | | М3 | |
| FORTRAN | C | FORTRAN | C |
| 10-3 | 0.0492 | 0.0491 | 2.83\*10-4 | 2.85\*10-4 |
| 10-4 | 0.0109 | 0.0109 | 4.7\*10-5 | 4.7\*10-5 |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами систем Pspice 8.0., ПА9, ПА7.

Таблица 6.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | Pspice 8.0 | ПА9 | | ПА7 |
| Euler | Trapec | MAN 12s |
| 10-3 | 3.8 | 6 | 6 |  |
| 10-4 | 0.8 | 6 | 6 |  |

**Тест 7.**

Линейная система:

y1’=-y2

y2’=y1

Начальные условия:

y1=1

y2 =0

Аналитическое решение:

y1=cos(x)

y2=sin(x)

###### Время интегрирования: [0;700]

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой Matlab 5.2.

### Таблица 7.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | ode23s | ode23 | ode23t | ode23tb | ode45 | ode15s | ode113 |
| 10-3 | - | - | - | - | - | 0.241 | 6.17\*10-2 |
| 10-4 | 0.193 | 3.43\*10-2 | 0.179 | 0.145 | 5.32\*10-3 | 6.82\*10-2 | 5.99\*10-3 |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой МВТУ 3.0.

##### Таблица 7.2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | Эйлера | Рунге-Кутты 4-5  классич. | Рунге-Кутты 4-5  модифиц. | Мерсона  классич. | Мерсона  модифиц. | Адаптив-  ный 1 | Адаптив-  ный 2 | Адаптив-  ный 3 |
| 10-3 | - | 0.164 | 0.07 | 0.047 | 0.051 | 0.294 | 0.136 | 0.147 |
| 10-4 | - | 0.011 | 0.0364 | 0.0057 | 0.0146 | 0.042 | 0.029 | 0.022 |
|  | Адаптив-  ный 4 | Адаптив-  ный 5 | Адаптив-  ный неявный | Диаго-  нально-  неявный | Гира | Эйлера  неявный | DIRK 33 | DIRK 44 |
| 10-3 | 0.0075 | 8.9\*10-12 | 0.0032 | 1.03 | 0.857 | 0.99 | 0.557 | 0.0423 |
| 10-4 | 0.0008 | 2.28\*10-12 | 1.55\*10-4 | 0.208 | 0.164 | 0.99 | 0.0755 | 0.00325 |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой DMAN.

##### Таблица 7.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | М2 | | М3 | |
| FORTRAN | C | FORTRAN | C |
| 10-3 | 0.183 | 0.185 | 8.658\*10-4 | 1.189\*10-3 |
| 10-4 | 4.142\*10-2 | 4.21\*10-2 | 1.64\*10-4 | 1.55\*10-4 |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой Pspice 8.0.

Таблица 7.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | Pspice 8.0 | ПА9 | | ПА7 |
| Euler | Trapec | MAN 12s |
| 10-3 | 2 | 1 | 1,01 |  |
| 10-4 | 2 | 1 | 0,23 |  |

По результатам проведенных тестов наиболее подходящими программами для решения систем с быстроосциллирующим решением являются МВТУ(методы адаптивный 4 и 5) и DMAN (метод М3).

**7.3. Интегрирование систем ОДУ в обратном времени.**

Много процессов в технике являются обратимыми, т.е. могут выполняться как в прямом, так и в обратном направлении. Поэтому программы анализа технических систем должны уметь с заданной точностью интегрировать системы ОДУ в разных направлениях. Является целесообразным интегрировать каждый тест в двух направлениях: прямом и обратном. Рассмотрим формирование так называемой "обратной" системы ОДУ.

Пусть дана задача Коши:

Y'=F(T,Y), Y(T0)=Y0 , (1)

имеющая аналитическое решение:

# Yан=Yа(T) (2)

# Интервал интегрирования: [T0 ; TK].

# Тогда задача Коши для интегрирования в обратном направлении от TK до T0 системы (1) имеет вид:

Y'=-F(T,Y), Y(T0)=Yа(TK) , (3)

Аналитическое решение ее имеет вид:

Yан=Yа(TK-T) (4)

**Тест 8.**

Данный тест представляет собой тест 1(система А2) с временем интегрирования [10 0]. Особенностью данного и последующих тестов является достаточно плавное поведение функции-решения в начале интервала интегрирования и скачкообразное к концу. В связи с этим у всех методов, которые смогли решить данный тест, наблюдаются очень большие погрешности.

Программа DMAN не смогла решить данный тест.

Программа МВТУ 3.0 смогла решить данный тест только следующими методами: диагонально неявным, Гира, DIRK 33, DIRK 44.Абсолютная погрешность решения для всех методов превосходит или равна 1.5. Аналогичную погрешность дает при решении программа ПА9.

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой Matlab 5.2.

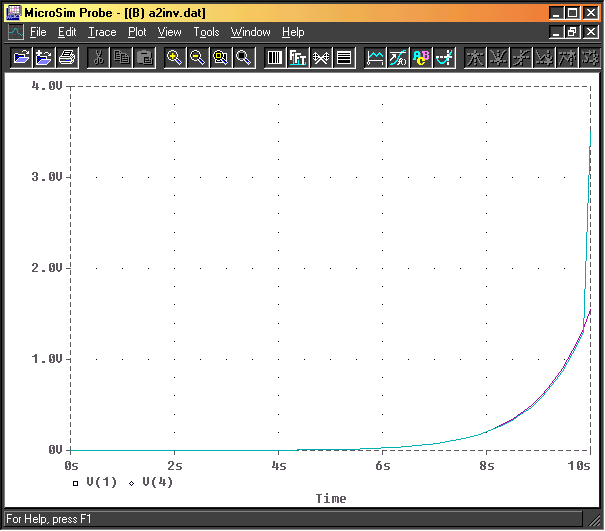
### Таблица 8.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | ode23s | ode23 | ode23t | ode23tb | ode45 | ode15s | ode113 |
| 10-3 | 0,738 | - | 0,789 | 0,731 | - | 0,847 | - |
| 10-4 | - | - | - | - | - | 0,793 | - |

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой Pspice 8.0.

Таблица 8,2

|  |  |
| --- | --- |
| Заданная максимальная погрешность | Полученная  погрешность |
| 10-3 | 1,9 |
| 10-4 | 1,9 |

Рис. 8.1. График численного и аналитического решения решения y1 (Pspice 8.0).

**Тест 9.**

Тест 3(система А5) с временем интегрирования [10 0].

Данный тест за приемлемое время не смог решить ни один из методов программы Matlab 5.2. Программа DMAN решила тест методами М2 и М3 с одинаковой абсолютной погрешностью-2.5, как и Pspice 8.0. Программа ПА9 смогла решить этот тест методом Эйлера с заданной точностью10-4, полученная погрешность – 2. Программа МВТУ данный тест решить не смогла.

Выходит, далеко не всегда программа, проинтегрировавшая систему в прямом направлении, сможет также решить ее и в обратном, а это имеет огромное значение, ведб в технике есть много обратимых процессов.

**7.4. Задачи с резко меняющимися свойствами функций-решений.**

Одними из самых трудных для решения программами являются задачи с резко меняющимися свойствами функций-решений. Например, в течение большого интервала времени функция монотонно медленно возрастала и вдруг скачок. Если программа не сможет правильно его отследить, то может пойти по неверной траектории решения. Это очень большая проблема для программ с изменяющимся шагом интегрирования. Идеальной должна быть программа, которая когда необходимо будет увеличивать шаг интегрирования, а когда нужно уменьшать соответствующим образом. Увы, идеальных программ пока нет, поскольку они должны обладать интеллектом. Все программы при определенных параметрах пропускают такие скачки из-за большого наращенного шага интегрирования. Отследит программа скачок или нет зависит от случая. При тестировании программ чрезвычайно важно учитывать этот момент. Но в большинстве случаев при моделировании технических объектов известно, когда будут происходить подобные скачки. Поэтому эти моменты времени необходимо "сообщать" программе. В большинстве программ (например, Pspice, МВТУ) такого механизма пока нет.

**Тест 10.**

"Гладкий пик", не имеющий разрывов производных:

c

b

T

0

Y

d

a

Данная функция состоит из 2 прямых и 3 парабол и является гладкой. Значения параметров:

a – расстояние до вершины "пика";

b – высота "пика";

c – ширина "пика".

Задача Коши для данной функции имеет вид:

y(0):=d

Полученные значения относительных погрешностей в таблицах приведены для значений: a=1, b=1000, c=0.5.

Программа МВТУ отслеживала "пик" случайным образом и во многих случаях с большой погрешностью. Часто программа получала более правильное решение при низкой заданной погрешности, но не получала его при более высокой. При задании b достаточно высоким (порядка 10250) наблюдаются непонятные небольшие скачки (порядка 1015) на нулевом участке. Поэтому МВТУ не приспособлена для решения таких задач и требует в этом аспекте тщательной доработки.

Таблица полученных максимальных относительных погрешностей, полученных при решении методами системой МВТУ 3.0.

##### Таблица 10.1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Заданная максимальная погрешность | Эйлера | Рунге-Кутты 4-5  классич. | Рунге-Кутты 4-5  модифиц. | Мерсона  классич. | Мерсона  модифиц. | Адаптив-  ный 1 | Адаптив-  ный 2 | Адаптив-  ный 3 |
| 10-3 | - | 0.05 | 0.05 | 2.7\*10-4 | 2.7\*10-4 | - | 0.028 | 0.025 |
| 10-4 | - | 0.15 | 0.15 | 0.05 | 0.05 | - | 0.0024 | 0.025 |
|  | Адаптив-  ный 4 | Адаптив-  ный 5 | Адаптив-  ный неявный | Диаго-  нально-  неявный | Гира | Эйлера  неявный | DIRK 33 | DIRK 44 |
| 10-3 | - | 0.15 | 0.01 | 4.8\*10-4 | - | - | 6.9\*10-4 | 0.01 |
| 10-4 | - | 0.15 | 0.12 | 0.001 | - | - | 1.4\*10-4 | 0.2 |

В программе DMAN введен специальный параметр TKV, с помощью которого можно точно стробировать задаваемые пользователем моменты времени в ходе интегрирования. Задавая TKV в примере равным a-c/2, мы заставляем выбрать программу такой шаг, при котором она попадет точно в начало "пика".

Таблица полученных максимальных относительных погрешностей, полученных при решении методами системой DMAN.

##### Таблица 10.2

|  |  |  |
| --- | --- | --- |
| Заданная максимальная погрешность | М2 | М3 |
| 10-3 | 1,186\*10-3 | 2,26\*10-5 |
| 10-4 | 1,378\*10-5 | 2,27\*10-5 |

Таблица полученных максимальных относительных погрешностей, полученных при решении методами системой Pspice 8.0.

Таблица 10.3

|  |  |
| --- | --- |
| Заданная максимальная погрешность | Полученная  погрешность |
| 10-3 | 0.15 |
| 10-4 | 0.02 |

Большие погрешности при решении подобных задач большинством программ получаются из-за того, что в программах нет проверки на разницу производных в двух соседних вычисленных точках функции-решения. Необходимо программе делать проверку |x’k+1-x’k|<ε. И если условие не выполняется точка xk+1 не принимается и проверяется точка xk+h/2, где h= xk+1-xk.

**7.5. Задачи с разрывами производных функций-решений.**

**Тест 11.**

Кусочно-линейная функция: Задача Коши:

y=|x-2|+|x-4|-|2\*x-6| dy/dx=|x-2|/(x-2)+|x-4|/(x-4)-2\*|2\*x-6|/(2\*x-6)

y(0)=0.

Программы МВТУ 3.0 и Matlab 5.2 не могут работать с разрывами производных (попадение в точку, близкую к точке разрыва сразу вызывает ошибки).

Программа Pspice 8.0 хоть и не вызывает ошибок при не очень высоких погрешностях, но зато допускает значительные погрешности из-за отсутствия в ней механизма работы с большими изменениями производной. Основными погрешностями решения таких задач на PSpice являются “срез” или “сглаживание” углов и пиков и в связи с этим большая потеря точности из-за попадания на неверную траекторию решения.

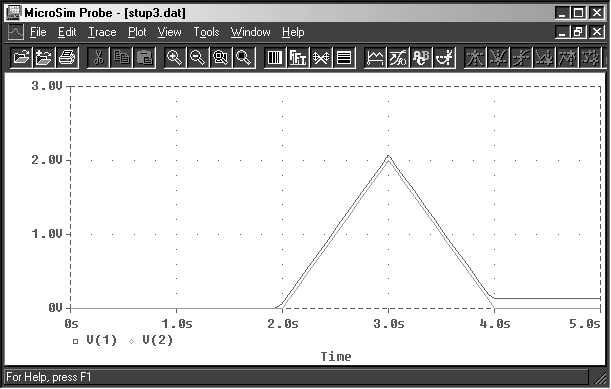
 Рис. 11.1**.** Графики аналитического V(2) и численного V(1) решения (RELTOL=1E-6).

Таблица полученных максимальных абсолютных погрешностей, полученных при решении методами системой Pspice 8.0 и программой DMAN.

Таблица 8.2

|  |  |  |  |
| --- | --- | --- | --- |
| Заданная максимальная погрешность | Полученная  погрешность | М2 | М3 |
| 10-3 | 0.07 | 9.77\*10-4 | 4.88\*10-4 |
| 10-4 | 0.17 | 6.1\*10-5 | 6.1\*10-5 |

Программа DMAN лучше всех предыдущих программ решила этот тест. Кроме того, в программе введен параметр NBAD, позволяющий точно определить программе точки разрыва производных.

**7.6. Нелинейные жесткие задачи.**

Тест 12.

Чувствительность программ к изменениям шагов интегрирования проверяется для системы дифференциальных уравнений Ван дер Поля, взятых из нелинейной механики

Уравнение Ван-дер-Поля:

y1’=y2

y2’=m\*(1-y1\*y1)\*y2-y1

Начальные условия:

y1(0)=2

y2 (0)=0

Интервал интегрирования: [0 4.2\*m].

Определяются максимальные значения коэффициента m>102, при, котором может быть получено решение за приемлемое время.

Методы системы Matlab 5.2. Таблица 10.2

|  |  |
| --- | --- |
| Метод | Максимальное значение m |
| ode23s | 108 |
| ode15s | 108 |
| ode23t | 106 |
| ode23tb | 106 |

Система Matlab5.2 находит решение с большими временными затратами.

Методы системы МВТУ 3.0 Таблица 10.2

|  |  |
| --- | --- |
| Метод | Максимальное значение m |
| Эйлера | - |
| Рунге-Кутты классический | - |
| Рунге-Кутты модифицированный | 103 |
| Мерсона классический | - |
| Мерсона модифицированный | 103 |
| Адаптивный 1 | 1014 |
| Адаптивный 2 | 1011 |
| Адаптивный 3 | 106 |
| Адаптивный 4 | 109 |
| Адаптивный 5 | 106 |
| Адаптивный неявный | 1012 |
| Диагонально-неявный | 1014 |
| Гира | 103 |
| Эйлера неявный | 109 |
| DIRK 33 | 107 |
| DIRK 44 | 106 |

С помощью программы DMAN удалось получить правильное решение этой системы вплоть до значения m=1020. С помощью программы PSPICE правильное решение было получено только до значения m=104.

**Результаты тестирования.**

Обзорная таблица сравнения методов решения систем ОДУ программ анализа технических систем приведена на листе 5.

Среди методов пакета Matlab 5.2 наибольшую эффективность показали методы ode23t и ode23tb. Методы ode15s и ode 45 имеет большую погрешность по сравнению с ними. Методы не могут работать с разрывами производных. В целом же эти 4 метода по своей пригодности к применению для решения различных классов задач равнозначны. Их можно применять лишь при небольших требуемых точностях решения.Методы для нежестких систем смогли решить лишь очень малую часть предложенных тестов.

Среди методов программы МВТУ лучшими являются (см. таблицу): адаптивные(2-5, неявный), диагонально-неявный, DIRK33, DIRK44. Недостатком программы является неумение работать с разрывами производных.

Программа DMAN в целом по всем тестам является лучшей из тестируемых (метод М3). В программу заложен механизм работы с разрывами производных.

Программа Pspice 8.0 большинство тестов не смогла решить с приемлемой точностью из-за того, что тестирование её математического ядра не проводилось ее производителями

Недостатком всех программ является неумение выбирать шаг интегрирования в местах, где функции-решения резко изменяют производную. Именно поэтому ни одна из программ не смогла точно решить систему А7 из/1/. Для устранения этого недостатка необходимо в программах проводить проверку на изменение производной в двух соседних точках вычисленной функции-решения и соответствующим образом выбирать шаг.

**Список использованной литературы.**

1.Заворин А.Н. Тестирование программ решения жестких систем обыкновенных дифференциальных уравнений. Материал информационного фонда РФАП Латвии. Инв.No ИМ0020, ВЦ ЛГУ им. П.Стучки, Рига, 1984, 42 с.

2.Хайрер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и алгебро-дифференциальные задачи: Пер. с англ.-М.: Мир, 1999.-612с

3. O. Vityaz, V.Porra. Testing of Time Domain Simulators for Nonlinear Electronic Circuits. Helsinki University of Technology, Faculty of Electrical Engineering, Electronic Circuit Design Laboratory, Report 4, Finland, July 1988.

4. Разевиг В.Д. Применение программ P-CAD и PSPICE для схемотех нического моделирования на ПЭВМ. В 4-х вып. Вып. 3: Моделирование аналоговых устройств. - М.: Радио и связь, 1992. - 71 с.