МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра механики, мехатроники и робототехники

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Локтионова
2016 г.

РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ЧИСЛЕННЫМИ МЕТОДАМИ (ПРИМЕНИТЕЛЬНО К ЗАДАЧАМ МЕХАНИКИ). МЕТОД ЭЙЛЕРА, МЕТОД РУНГЕ-КУТТА ЧЕТВЁРТОГО ПОРЯДКА, МЕТОД РУНГЕ ДВОЙНОГО ПЕРЕСЧЕТА

Методические указания к выполнению практической и самостоятельной работы
по дисциплине «Моделирование и исследование мехатронных систем и роботов» для студентов направления
15.04.06

Курск

УДК 621

Составители: С.И. Савин

Рецензент
Кандидат технических наук, доцент *Е.Н. Политов*

Решение обыкновенных дифференциальных уравнений численными методами (применительно к задачам механики). Метод Эйлера, метод Рунге-Кутта четвёртого порядка, метод Рунге двойного пересчета: методические указания к выполнению практической и самостоятельной работы по дисциплине «Моделирование и исследование мехатронных систем и роботов» / Юго-Зап. гос. ун-т; сост. С.И. Савин. Курск, 2016. 21 с.: ил. 16, табл. 1. Библиогр.: с.20-21.

Методические указания содержат сведения о применении численных методов при решении обыкновенных дифференциальных уравнений, возникающих при решении задач механики. Рассмотрены метод Эйлера, метод Рунге-Кутта четвёртого порядка, метод Рунге двойного пересчета.

Методические указания соответствуют требованиям программы, утверждённой учебно-методическим объединением (УМО).

Предназначены для студентов направления 15.04.06 – Мехатроника и робототехника всех форм обучения.

Текст печатается в авторской редакции

Подписано в печать . Формат 60х84 1/16
Усл.печ.л. ___. Уч.-изд.л. ___ Тираж 100 экз. Заказ.
Бесплатно.
Юго-Западный государственный университет.
305040 Курск, ул. 50 лет Октября, 94

РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ЧИСЛЕННЫМИ МЕТОДАМИ (ПРИМЕНИТЕЛЬНО К ЗАДАЧАМ МЕХАНИКИ). МЕТОД ЭЙЛЕРА, МЕТОД РУНГЕ-КУТТА ЧЕТВЁРТОГО ПОРЯДКА, МЕТОД РУНГЕ ДВОЙНОГО ПЕРЕСЧЕТА

Цель работы: *изучить применение численных методов при решении обыкновенных дифференциальных уравнений, возникающих при решении задач механики - метод Эйлера, метод Рунге-Кутта четвёртого порядка, метод Рунге двойного пересчета.*

Аппаратные средства: *математический пакет Mathcad.*

1. Краткие теоретические сведения

При решении обыкновенных дифференциальных равнений часто используются численные методы. Одним из базовых численных методов решения ОДУ является ***метод Эйлера*** (также называемый ***методом ломаных***).

Пусть имеется ОДУ, записанное в векторной форме:

$$\frac{d}{dt}\mathbf{y} = \mathbf{f}(\mathbf{y}, t). \qquad (1)$$

где $\mathbf{f}(\mathbf{y}, t)$ - вектор-функция переменного $\mathbf{y}$ и времени *t*. Поставим задачу найти аппроксимацию зависимости $\mathbf{y}(t)$ кусочно-линейной функцией $\mathbf{y}^*(t)$. Найдем значения $\mathbf{y}^*(t)$ в N точках, равномерно распределённых на интервале $t \in [t_0 \quad t_1]$; *i*-ой точке соответствует момент времени $t_i = t_0 + i \cdot \Delta t$.

Для решения этой задачи нужно знать начальное значение $\mathbf{y}_0 = \mathbf{y}(0)$. Тогда искомую последовательность можно найти, используя следующий итерационный алгоритм:

$$\mathbf{y}^*_{i+1} = \mathbf{y}^*_i + \mathbf{f}\left(\mathbf{y}^*_i, t_i\right) \cdot \Delta t, \qquad (2)$$

где $\mathbf{y}^*_i = \mathbf{y}\left(t^*_i\right)$. Данный алгоритм называется ***прямым методом Эйлера***. Для нахождения значения функции $\mathbf{y}^*(t)$ в следующий момент времени данный метод использует значение функции в текущий момент. Это делает его явным методом (в англоязычной литературе такие методы называют ***explicit methods***).

Метод ломаных может быть реализован и другим образом:

$$\mathbf{y}_{i+1}^{*} = \mathbf{y}_{i}^{*} + \mathbf{f}\left(\mathbf{y}_{i+1}^{*}, t_{i+1}\right) \cdot \Delta t, \tag{3}$$

В данной реализации для нахождения значения функции $\mathbf{y}^{*}(t)$ в следующий момент времени требуется знать значение функции $\mathbf{f}$ в этот же момент. Так как значение $\mathbf{f}$ зависит от $\mathbf{y}^{*}$, то в большинстве случаев возможно лишь найти аппроксимацию величины $\mathbf{f}$. Данный метод называется ***обратным методом Эйлера*** и он относится к ***неявным методам*** (в англоязычной литературе такие методы называют ***implicit methods***).

Величина Δt называется шагом интегрирования. Величина шага влияет на точность полученной аппроксимации.

Для решения задачи отыскания некоторой аппроксимации $\mathbf{y}^{*}(t)$ искомого решения $\mathbf{y}(t)$ используются и другие методы. Наиболее распространенным является ***метод Рунге-Кутта четвёртого порядка***. Его реализация может быть представлена в виде следующего итерационного алгоритма:

$$k_{1,i} = \mathbf{f}\left(\mathbf{y}_{i}^{*}, t_{i}\right), \tag{4}$$

$$k_{2,i} = \mathbf{f}\left(\mathbf{y}_{i}^{*} + k_{1,i} \cdot \frac{1}{2}\Delta t, \quad t_{i} + \frac{1}{2}\Delta t\right) \tag{5}$$

$$k_{3,i} = \mathbf{f}\left(\mathbf{y}_{i}^{*} + k_{2,i} \cdot \frac{1}{2}\Delta t, \quad t_{i} + \frac{1}{2}\Delta t\right) \tag{6}$$

$$k_{4,i} = \mathbf{f}\left(\mathbf{y}_{i}^{*} + k_{3,i} \cdot \Delta t, \quad t_{i} + \Delta t\right) \tag{7}$$

$$\mathbf{y}_{i+1}^{*} = \mathbf{y}_{i}^{*} + \left(k_{1,i} + 2k_{2,i} + 2k_{3,i} + k_{4,i}\right) \cdot \frac{\Delta t}{6} \tag{8}$$

Данный метод является явным, так как для нахождения следующего значения $\mathbf{y}^{*}$ используются данные, известные на текущем шаге.

Для обеспечения заданной точности вычислений используются методы интегрирования с переменным шагом. К таким относится ***метод Рунге двойного пересчета***, реализуемый следующим итерационным алгоритмом:

$$k_{1,i} = \mathbf{f}\left(\mathbf{y}_i^*, t_i\right), \tag{9}$$

$$k_{2,i} = \mathbf{f}\left(\mathbf{y}_i^* + k_{1,i} \cdot \frac{1}{3}\Delta t, \quad t_i + \frac{1}{3}\Delta t\right), \tag{10}$$

$$k_{3,i} = \mathbf{f}\left(\mathbf{y}_i^* + k_{1,i} \cdot \frac{1}{6}\Delta t + k_{2,i} \cdot \frac{1}{6}\Delta t, \quad t_i + \frac{1}{3}\Delta t\right), \tag{11}$$

$$k_{4,i} = \mathbf{f}\left(\mathbf{y}_i^* + k_{1,i} \cdot \frac{1}{8}\Delta t + k_{2,i} \cdot \frac{3}{8}\Delta t, \quad t_i + \frac{1}{2}\Delta t\right), \tag{12}$$

$$\mathbf{g}_i = \mathbf{y}_i^* + \left(\frac{1}{2}k_{1,i} - \frac{3}{2}k_{3,i} + 2k_{4,i}\right) \cdot \frac{\Delta t}{2}, \tag{13}$$

$$k_{5,i} = \mathbf{f}\left(\mathbf{g}_i, \quad t_i + \Delta t\right), \tag{14}$$

$$\mathbf{y}_{i+1}^* = \mathbf{y}_i^* + \left(k_{1,i} + 4k_{4,i} + k_{5,i}\right) \cdot \frac{\Delta t}{6}, \tag{15}$$

$$\Delta t := \begin{cases} \frac{1}{2}\Delta t & \text{если} \quad \left|\mathbf{g}_i - \mathbf{y}_{i+1}^*\right| > 5\delta \\ 2\Delta t & \text{если} \quad \left|\mathbf{g}_i - \mathbf{y}_{i+1}^*\right| \le \frac{1}{64}\delta, \\ \Delta t & \text{иначе} \end{cases} \tag{16}$$

где δ определяет точность вычислений.

3. Методика выполнения лабораторной работы

Рассмотрим обыкновенное линейное дифференциальное уравнение второго порядка, описывающее движение гармонического осциллятора с затуханием:

$$m\ddot{y} + \mu\dot{y} + ky = 0. \tag{17}$$

где $m = 1$ *кг* - масса, $\mu = 5 \ \frac{кг \cdot м}{с}$ - коэффициент вязкого сопротивления, $k = 5 \ Н/м$ - жесткость упругого элемента. Зададим начальные условия $y(0) = 2$ *м*, $\dot{y}(0) = 1$ *м/с*.

Приведем уравнение (17) к системе ОДУ первого порядка, введя следующие замены:

$$y_0 = y, \qquad y_1 = \dot{y}. \tag{18}$$

Тогда полученная система ОДУ будет иметь вид:

$$\begin{cases} \dot{y}_0 = y_1 \\ \dot{y}_1 = -\dfrac{\mu}{m} y_1 - \dfrac{k}{m} y_0 \end{cases}. \tag{19}$$

Переменные y_0, y_1 называют ***фазовыми***. Программа в среде Mathcad, реализующая описанные действия, приведена ниже:

Листинг 1 Исходное уравнение, система ОДУ первого порядка

Исходное уравнение (линейное однородное ОДУ 2ого порядка):

$$\mathrm{ddy}(y, dy, m, \mu, k) := -\frac{1}{m}\cdot(\mu\cdot dy + k\cdot y)$$

Приведение к системе ОДУ первого порядка:

$$\mathrm{ds}(s, m, \mu, k) := \begin{bmatrix} s_1 \\ -\dfrac{1}{m}\cdot\left(\mu\cdot s_1 + k\cdot s_0\right) \end{bmatrix}$$

Введем вектор начальных условий $IC = [y(0) \quad \dot{y}(0)]^{\mathrm{T}}$. Подставим числа вместо использованных констант.

Листинг 2 Подстановка чисел в полученные выражения, упрощение их записи.

$$m := 1 \qquad \mu := 5 \qquad k := 5 \qquad IC := \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$\mathrm{ds}(s) := \mathrm{ds}(s, m, \mu, k)$$

Без доказательства примем, что аналитическое решение описанного уравнения имеет вид:

Листинг 3 Общее решение системы ОДУ

$$\mathrm{Solution}(t) := \begin{bmatrix} 3.68\cdot e^{-1.38\cdot t} + -1.68\cdot e^{-3.62\cdot t} \\ -0.5\cdot e^{-3.62\cdot t}\cdot\left(10.2\cdot e^{2.24\cdot t} - 12.2\right) \end{bmatrix}$$

Запишем код, реализующий прямой метод Эйлера.

Листинг 4 Прямой метод Эйлера

```
NormalEuler(Time, Δt) := | Count ← Time/Δt
                         | s ← IC
                         | for i ∈ 0..Count
                         |   | t ← i·Δt
                         |   | ε ← ds(s)
                         |   | s ← s + ε·Δt
                         |   | Error ← Solution(t) − s
                         |   | Out[i,0] ← t
                         |   | for j ∈ 0..1
                         |   |   | Out[i,j+1] ← s[j]
                         |   |   | Out[i,j+3] ← Error[j]
                         | Out
```

В описанном коде параметр *Time* определяется отрезок времени, на котором будет находиться аппроксимация решения. Параметр Δt определяет шаг интегрирования, переменная *Count* - число итераций, s - вектор фазовых переменных, ε - вектор производных от фазовых переменных, i - счетчик, t - текущее время.

Вызов написанной функции осуществляется следующим образом:

Листинг 5 Вызов функции, реализующей прямой метод Эйлера

$$\mathrm{Res} := \mathrm{NormalEuler}\left(10, 10^{-3}\right)$$

Написанная программа возвращает временную зависимость $y(t)$, её производную - $\dot{y}(t)$, а также величину отклонения каждой из этих зависимостей от аналитических (см. рисунки 1 и 2). Обозначим последние как $e(t)$ и $\dot{e}(t)$.

Напомним, что $y(t)=y_0(t)$, $\dot{y}(t)=y_1(t)$, т.е. полученные зависимости также отражают изменение во времени фазовых переменных.

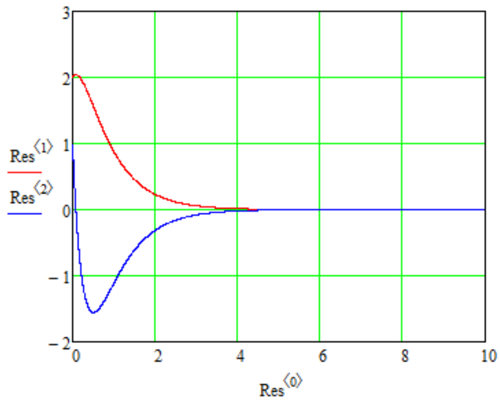

Рисунок 1 Временные зависимости $y(t)$ и $\dot{y}(t)$, аппроксимирующие решение системы ОДУ

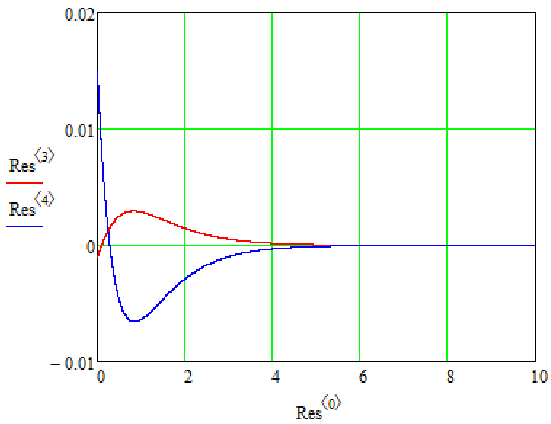

Рисунок 2 Временные зависимости $e(t)$ и $\dot{e}(t)$, показывающие разницу между точным решением системы ОДУ и его аппроксимациями.

Запишем код, реализующий метод Рунге-Кутта четвёртого порядка.

Листинг 6 Метод Рунге-Кутта четвёртого порядка.

$$
\mathrm{NormalRungeKutta}(Time, \Delta t) :=
\begin{array}{|l}
Count \leftarrow \dfrac{Time}{\Delta t} \\
s \leftarrow IC \\
\text{for } i \in 0..Count \\
\quad \begin{array}{|l}
t \leftarrow i \cdot \Delta t \\
k1 \leftarrow ds(s) \\
k2 \leftarrow ds\left(s + k1 \cdot \dfrac{\Delta t}{2}\right) \\
k3 \leftarrow ds\left(s + k2 \cdot \dfrac{\Delta t}{2}\right) \\
k4 \leftarrow ds(s + k3 \cdot \Delta t) \\
s \leftarrow s + (k1 + 2k2 + 2k3 + k4) \cdot \dfrac{\Delta t}{6} \\
Error \leftarrow Solution(t) - s \\
Out_{i,0} \leftarrow t \\
\text{for } j \in 0..1 \\
\quad \begin{array}{|l}
Out_{i,j+1} \leftarrow s_j \\
Out_{i,j+3} \leftarrow Error_j
\end{array}
\end{array} \\
Out
\end{array}
$$

Вызов записанной функции осуществляется следующим образом:

Листинг 7 Вызов функции, реализующей метод Рунге-Кутта четвёртого порядка

$$Res := \mathrm{NormalRungeKutta}\left(10, 10^{-3}\right)$$

Написанная программа, как и предыдущая, возвращает временную зависимость $y(t)$, её производную - $\dot{y}(t)$, а также величину отклонения каждой из этих зависимостей от аналитических $e(t)$ и $\dot{e}(t)$ (см. рисунки 3 и 4).

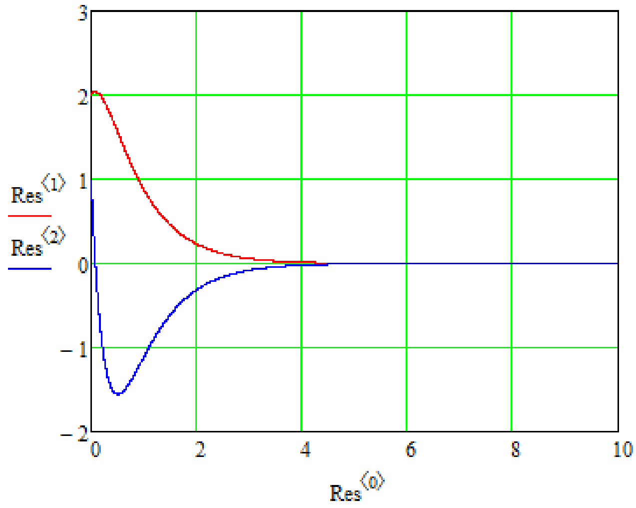

Рисунок 3 Временные зависимости $y(t)$ и $\dot{y}(t)$, аппроксимирующие решение системы ОДУ

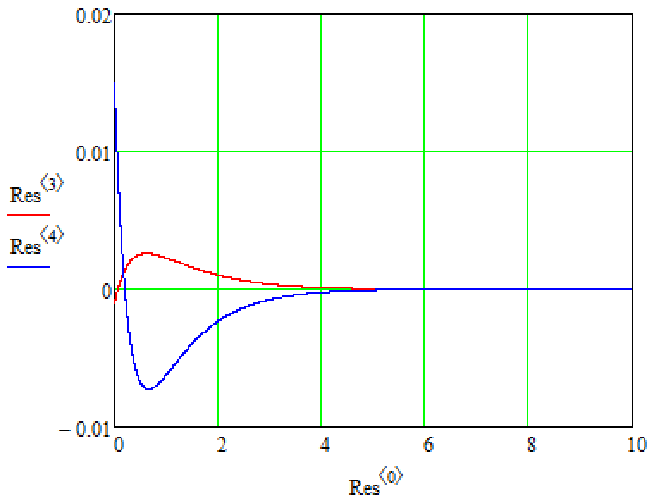

Рисунок 4 Временные зависимости $e(t)$ и $\dot{e}(t)$, показывающие разницу между точным решением системы ОДУ и его аппроксимациями.

Можем заметить, что существенных различий между аппроксимациями решений, полученных методами Эйлера и Рунге-Кутта, с точки зрения точности, не наблюдается. Это связано с выбором величины шага интегрирования $\Delta t = 10^{-3}$, который достаточно мал для получения точного решения любым из указанных методов.

Запишем код, реализующий метод Рунге двойного пересчета.

Листинг 8 Метод Рунге двойного пересчета

$$
\mathrm{VariableStep}(\mathrm{Time},\Delta t,\delta,\max\Delta t) :=
\left|
\begin{array}{l}
s \leftarrow IC \\
t \leftarrow 0 \\
\text{while } t < \mathrm{Time} \\
\quad \left|
\begin{array}{l}
t \leftarrow t + \Delta t \\
k1 \leftarrow ds(s) \\
k2 \leftarrow ds\left(s + \frac{1}{3}\cdot k1\cdot \Delta t\right) \\
k3 \leftarrow ds\left(s + \frac{1}{6}\cdot k1\cdot \Delta t + \frac{1}{6}\cdot k2\cdot \Delta t\right) \\
k4 \leftarrow ds\left(s + \frac{1}{8}\cdot k1\cdot \Delta t + \frac{3}{8}\cdot k2\cdot \Delta t\right) \\
s1 \leftarrow s + \frac{1}{2}\cdot\left(\frac{1}{2}\cdot k1 - \frac{3}{2}\cdot k3 + 2\cdot k4\right)\cdot \Delta t \\
k5 \leftarrow ds(s1) \\
s \leftarrow s + (k1 + 4k4 + k5)\cdot\frac{\Delta t}{6} \\
\mathrm{Error} \leftarrow \mathrm{Solution}(t) - s \\
\Delta \leftarrow \sqrt{(s1 - s)\cdot(s1 - s)} \\
Out_{i,0} \leftarrow t \\
\text{for } j \in 0..1 \\
\quad \left|
\begin{array}{l}
Out_{i,j+1} \leftarrow s_j \\
Out_{i,j+3} \leftarrow \mathrm{Error}_j
\end{array}
\right. \\
Out_{i,5} \leftarrow \Delta t \\
Out_{i,6} \leftarrow \Delta \\
\Delta t \leftarrow \mathrm{if}\left(\frac{\Delta}{5} > \delta, 0.5\cdot\Delta t, \Delta t\right) \\
\Delta t \leftarrow \mathrm{if}\left(\Delta \le \frac{\delta}{64}, 2\Delta t, \Delta t\right) \\
\Delta t \leftarrow \mathrm{if}(\Delta t > \max\Delta t, \max\Delta t, \Delta t) \\
i \leftarrow i + 1
\end{array}
\right. \\
Out
\end{array}
\right.
$$

Вызов записанной функции осуществляется следующим образом:

Листинг 9 Вызов функции, реализующей метод Рунге двойного пересчета

$$\mathrm{Res} := \mathrm{VariableStep}\left(20, 10^{-3}, 0.01, 1\right)$$

$$\text{длина}\left(\mathrm{Res}^{\langle 0 \rangle}\right) = 2.92 \times 10^{3}$$

Обратим внимание на то, что использование метода Рунге двойного пересчета может позволить сократить число итераций алгоритма. Так в нашем случае число итераций при использовании метода Эйлера составило бы 20 000, тогда как метод Рунге двойного пересчета решил задачу используя лишь 2920 итераций (Здесь и далее, говоря «число итераций при использовании метода Эйлера составило бы» подразумеваем, что использовалась та же величина шага интегрирования, которая была задана как стартовая для метода с переменным шагом).

Написанная программа, как и предыдущие, возвращает временную зависимость $y(t)$, её производную - $\dot{y}(t)$, а также величину отклонения каждой из этих зависимостей от аналитических $e(t)$ и $\dot{e}(t)$ (см. рисунки 5 и 6). Также программа возвращает текущую величину шага интегрирования (см. рисунок 7) и текущее значение величины $\Delta = \left|\mathbf{g} - \mathbf{y}^*\right|$ - аппроксимацию ошибки численного метода (см. рисунок 8).

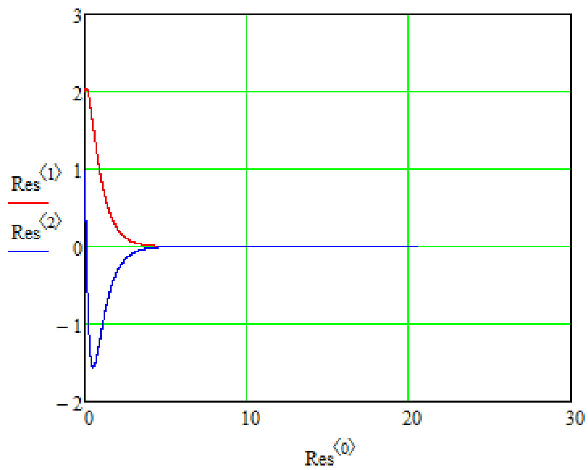

Рисунок 5 Временные зависимости $y(t)$ и $\dot{y}(t)$, аппроксимирующие решение системы ОДУ

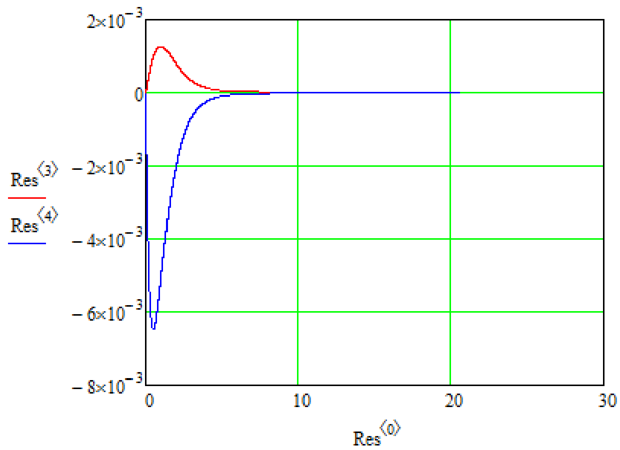

Рисунок 6 Временные зависимости $e(t)$ и $\dot{e}(t)$, показывающие разницу между точным решением системы ОДУ и его аппроксимациями.

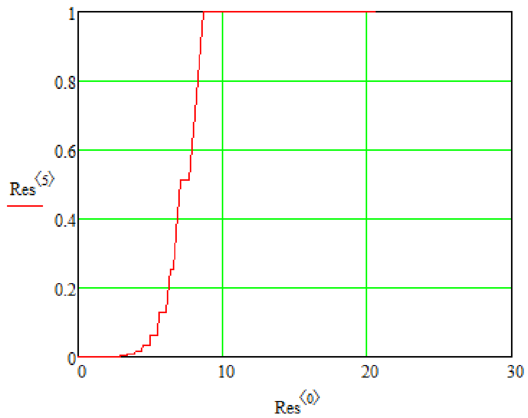

Рисунок 7 Временная зависимость величины шага интегрирования Δt .

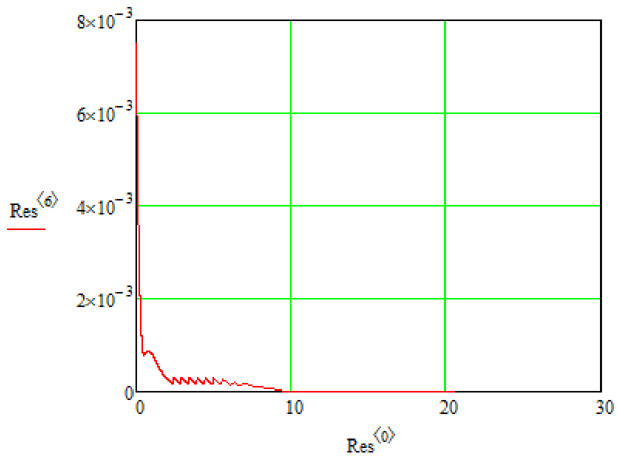

Рисунок 8 Временная зависимость аппроксимации ошибки численного метода Δ

Отметим, что метод Рунге двойного пересчета на каждой итерации рассчитывает аппроксимацию решения ОДУ дважды. Это делается для получения оценки точности решения. Вместе с тем это усложняет структуру алгоритма и повышает объем необходимых вычислений. В некоторых случаях в качестве косвенной оценки точности вычислений можно использовать величину прироста значения функции $y(t)$ на данном шаге. На этом методе основана следующая программа, реализующая численное решение системы ОДУ с переменным шагом интегрирования.

Листинг 10 Упрощенный численный метод решения ОДУ с переменным шагом интегрирования

```
Res

VariableStepSimplified(Time, Δt, δ, maxΔt) := | s ← IC
                                              | t ← 0
                                              | while t < Time
                                              |   | t ← t + Δt
                                              |   | ε ← ds(s)
                                              |   | s ← s + ε·Δt
                                              |   | Error ← Solution(t) − s
                                              |   | Δ ← max(ε·Δt)
                                              |   | Out[i,0] ← t
                                              |   | for j ∈ 0..1
                                              |   |   | Out[i,j+1] ← s[j]
                                              |   |   | Out[i,j+3] ← Error[j]
                                              |   | Out[i,5] ← Δt
                                              |   | Out[i,6] ← Δ
                                              |   | Δt ← if(Δ > (1/10)·δ, 0.5·Δt, Δt)
                                              |   | Δt ← if(Δ ≤ δ/100, 2Δt, Δt)
                                              |   | Δt ← if(Δt > maxΔt, maxΔt, Δt)
                                              |   | i ← i + 1
                                              | Out
```

Обратим внимание на то, что предложенный код позволяет определять максимально допустимый шаг интегрирования.

Вызов записанной функции осуществляется следующим образом:

Листинг 11 Вызов функции, реализующей упрощенный численный метод решения ОДУ с переменным шагом интегрирования

$$Res := VariableStepSimplified\left(10, 10^{-3}, 0.001, 1\right)$$

$$длина\left(Res^{\langle 0 \rangle}\right) = 9.511 \times 10^{3}$$

Обратим внимание на то, что написанная функция нашла аппроксимацию решения системы ОДУ используя 9 511 итераций, тогда как метод Эйлера сделал бы тоже за 10 000 итераций.

Написанная программа, как и предыдущая, возвращает временную зависимость $y(t)$, её производную - $\dot{y}(t)$, а также величину отклонения каждой из этих зависимостей от аналитических $e(t)$ и $\dot{e}(t)$ (см. рисунки 9 и 10). Также программа возвращает текущую величину шага интегрирования (см. рисунок 11) и текущее значение величины $\Delta_i = \mathbf{y}^*_{i+1} - \mathbf{y}^*_i$ - косвенную оценку ошибки численного метода (см. рисунок 12).

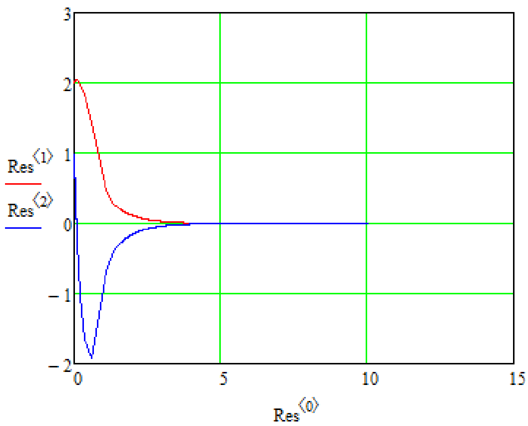

Рисунок 9 Временные зависимости $y(t)$ и $\dot{y}(t)$, аппроксимирующие решение системы ОДУ

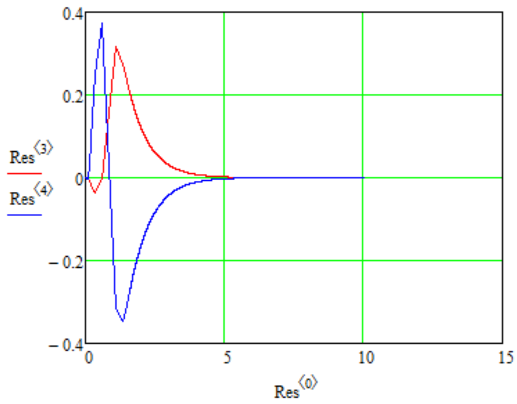

Рисунок 10 Временные зависимости $e(t)$ и $\dot{e}(t)$, показывающие разницу между точным решением системы ОДУ и его аппроксимациями.

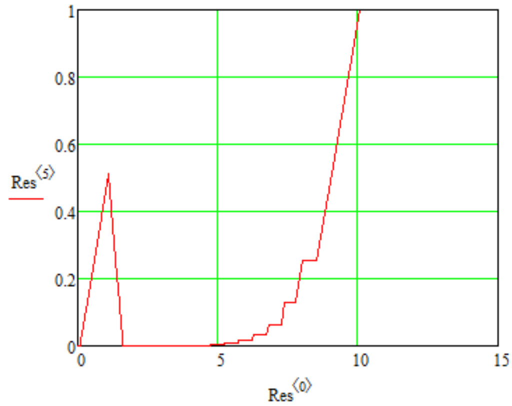

Рисунок 11 Временная зависимость величины шага интегрирования Δt.

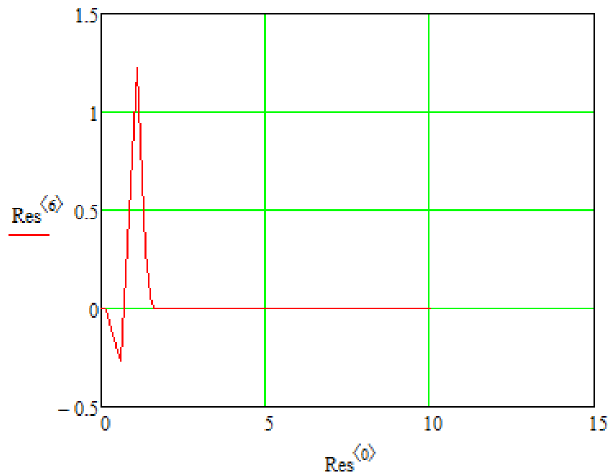

Рисунок 12 Временная зависимость величины Δ - косвенной оценки ошибки численного метода

Для того, чтобы продемонстрировать возможности указанного метода найдем аппроксимацию решения системы ОДУ на в 10 раз большем временном интервале:

Листинг 11 Вызов функции, реализующей упрощенный численный метод решения ОДУ с переменным шагом интегрирования

$$Res := VariableStepSimplified\left(100, 10^{-3}, 0.001, 1\right)$$

$$длина\left(Res^{\langle 0\rangle}\right) = 9.629 \times 10^{3}$$

Заметим, что число итераций выросло лишь на 128. Таким образом, решение на последних 90 секунд было найдено используя 128 итераций, тогда как метод Эйлера использовал бы 90 000 итераций. Причина столь значительного сокращения числа итераций состоит в том, что среднее значение величины шага интегрирования на временном отрезке $t \in [10 \quad 100]$ было близко к максимально допустимому, в данном случае 1 секунде.

На рисунках 13-16 показаны полученные временные зависимости.

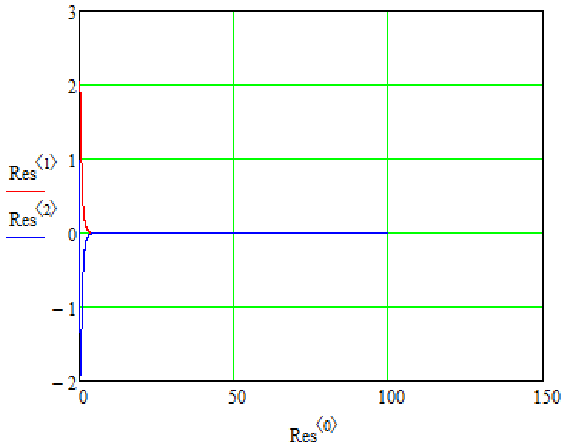

Рисунок 13 Временные зависимости $y(t)$ и $\dot{y}(t)$, аппроксимирующие решение системы ОДУ

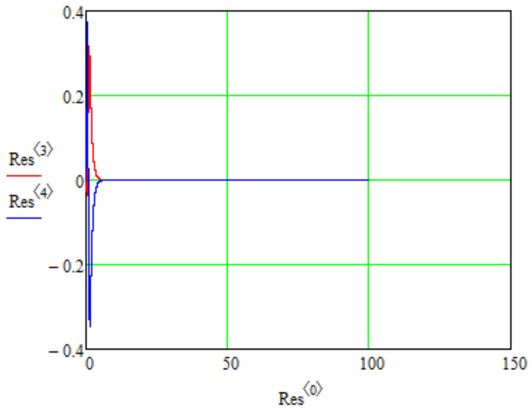

Рисунок 14 Временные зависимости $e(t)$ и $\dot{e}(t)$, показывающие разницу между точным решением системы ОДУ и его аппроксимациями.

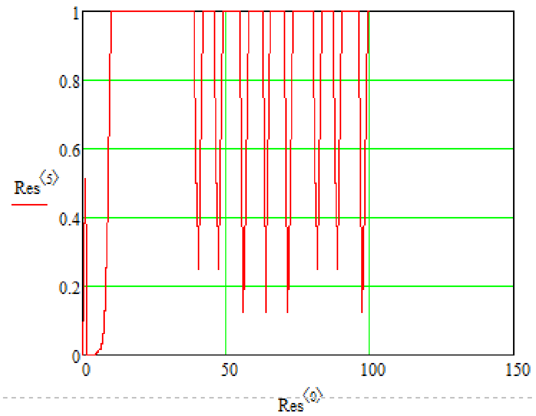

Рисунок 15 Временная зависимость величины шага интегрирования Δt.

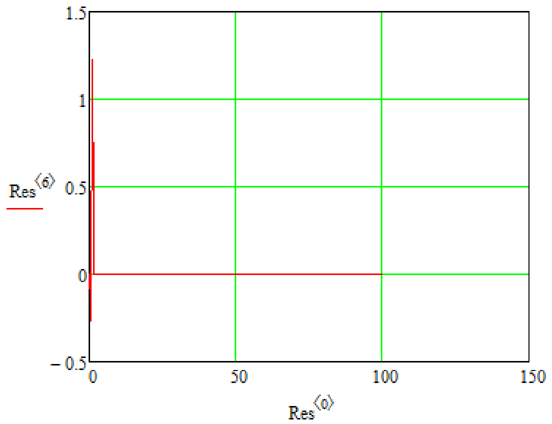

Рисунок 16 Временная зависимость величины Δ - косвенной оценки ошибки численного метода

Задание на выполнение лабораторной работы состоит в нахождении численного решения уравнения:

$$m\ddot{y}+\mu\dot{y}+ky=0 \qquad (16)$$

для набора параметров, данных в таблице, всеми описанными в методическом пособии методами. Значения констант выбираются в соответствии с номером студента в списке группы.

Таблица 1 Задания на выполнение лабораторной работы

№	Задание
1	Параметры осциллятора: $m=3$ *кг*, $\mu=5\ \frac{кг\cdot м}{c}$, $k=4\ H/_{м}$. Начальные условия: $y(0)=$ 1 *м*, $\dot{y}(0)=$ 2 *м/с*.
2	Параметры осциллятора: $m=5$ *кг*, $\mu=4\ \frac{кг\cdot м}{c}$, $k=3\ H/_{м}$. Начальные условия: $y(0)=$ 1 *м*, $\dot{y}(0)=$ 0 *м/с*.
3	Параметры осциллятора: $m=7$ *кг*, $\mu=5\ \frac{кг\cdot м}{c}$, $k=2\ H/_{м}$. Начальные условия: $y(0)=$ 2 *м*, $\dot{y}(0)=$ 2 *м/с*.
4	Параметры осциллятора: $m=2$ *кг*, $\mu=7\ \frac{кг\cdot м}{c}$, $k=3\ H/_{м}$. Начальные условия: $y(0)=$ 3 *м*, $\dot{y}(0)=$ 2 *м/с*.
5	Параметры осциллятора: $m=10$ *кг*, $\mu=8\ \frac{кг\cdot м}{c}$, $k=4\ H/_{м}$. Начальные условия: $y(0)=$ 0.2 *м*, $\dot{y}(0)=$ 0.4 *м/с*.
6	Параметры осциллятора: $m=12$ *кг*, $\mu=2\ \frac{кг\cdot м}{c}$, $k=2\ H/_{м}$. Начальные условия: $y(0)=$ 0.4 *м*, $\dot{y}(0)=$ 2 *м/с*.
7	Параметры осциллятора: $m=20$ *кг*, $\mu=10\ \frac{кг\cdot м}{c}$, $k=\ 7\,H/_{м}$.

	Начальные условия: $y(0) = 0.4\ м,\ \dot{y}(0) = 0.4\ м/с$.
8	Параметры осциллятора: $m = 15\ кг,\ \mu = 11\ \frac{кг \cdot м}{с},\ k = 7\ Н/м$. Начальные условия: $y(0) = 4\ м,\ \dot{y}(0) = 2\ м/с$.
9	Параметры осциллятора: $m = 25\ кг,\ \mu = 8\ \frac{кг \cdot м}{с},\ k = 4\ Н/м$. Начальные условия: $y(0) = 1\ м,\ \dot{y}(0) = 0.5\ м/с$.
10	Параметры осциллятора: $m = 3\ кг,\ \mu = 4\ \frac{кг \cdot м}{с},\ k = 3\ Н/м$. Начальные условия: $y(0) = 0.5\ м,\ \dot{y}(0) = 0.5\ м/с$.

Рекомендуемая литература

1. Самарский А.А., Введение в численные методы, Учебники для вузов. Специальная литература. Лань, М., с 288, 2009 г.
2. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М., Численные методы, Классический университетский учебник. Бином. Лаборатория знаний, ЛКИ, М., с 640, 2011 г.
3. Калиткин Н.Н., Численные методы, Учебная литература для вузов. БХВ-Петербург, с 592, 2011 г.
4. Дж. Холл, Дж. Уатт, Современные численные методы решения обыкновенных дифференциальных уравнений, Книга по Требованию, с 312, 2012 г.
5. Агафонов С.А., Герман А.Д., Муратова Т.В. Математика в техническом университете. Выпуск 8. Дифференциальные уравнения, М.: МГТУ им. Н.Э. Баумана, 2001. – 336с.
6. Немыцкий В.В., Степанов В.В. Качественная теория дифференциальных уравнений, М.: Регулярная и хаотическая динамика, 2004. – 456с.

7. Арнольд В.И. Обыкновенные дифференциальные уравнения, М.: МЦНМО, 2012. – 352с.
8. Арнольд В.И. Геометрические методы в теории обыкновенных дифференциальных уравнений, М.: МЦНМО, 2012. – 384с.
9. Трикоми Ф. Дифференциальные уравнения, М.: Едиториал УРСС, 2010. – 352с.
10. Канатников А.Н., Крищенко А.П., Четвериков В.Н. Дифференциальное исчисление функций многих переменных, М.: МГТУ им. Н.Э. Баумана, 2007. – 456с.
11. Канатников А.Н., Крищенко А.П. Математика в техническом университете. Выпуск 4. Линейная алгебра, М.: МГТУ им. Н.Э. Баумана, 1999. – 336с.
12. Ильин В.А., Позняк Э.Г. Линейная алгебра, М.: ФИЗМАТЛИТ, 2014. – 280с.
13. Александров П.С. Курс аналитической геометрии и линейной алгебры, М.: Лань, 2009. – 512с.
14. Курош А.Г. Курс вышей алгебры, М.: Лань, Физматкнига, 2007. – 432с.
15. Хорн Р., Джонсон Ч. Матричный анализ, М.: Книга по Требованию, 2012. – 667с.
16. Воднев В.Г., Наумович А.Ф., Наумович Н.Ф. Математический словарь высшей школы, М.: Издательство МПИ, 1989. – 527с.
17. Свешников А.Г., Тихонов А.Н. Теория функций комплексной переменной, М.: ФИЗМАТЛИТ, 2001. – 336с.
18. Половинкин Е.С. Курс лекций по теории функций комплексного переменного, М.: Физматкнига, МФТИ, 2003. – 208с.
19. В.А. Ильин, Математический анализ / В.А. Садовничий, Б.Х. Сендов // Главная редакция физико-математической литературы издательства "Наука", 720 стр., 1979 г.
20. В.С. Зарубин Математическое моделирование в технике / Серия «Математика в техническом университете» (Выпуск XXI), МГТУ им. Н. Э. Баумана, 2010, 496 стр.