БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
  
Международный институт дистанционного образования  
Кафедра «Информационные системы и технологии»

**ОТЧЁТ  
по контрольной работе**

по дисциплине  
«**Вычислительная математика**»

ПРЕПОДАВАТЕЛЬ: Напрасников Владимир Владимирович

ИСПОЛНИТЕЛЬ: студент группы 41703120 Реут Владислав Леонидович

Минск 2023

**Задание № 1**

Определить корни уравнения F(x) = 0 графически и уточнить один из них:

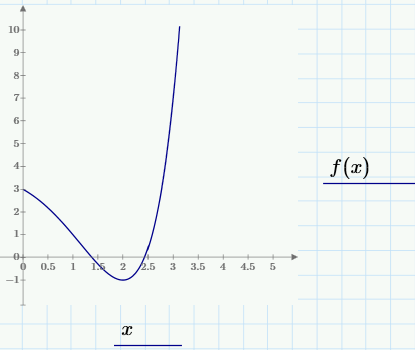
* методом половинного деления;
* методом хорд;
* методом касательных;
* методом секущих.

с точностью ε = 0.001.

Создать функции, реализующие указанные методы, построить графическую иллюстрацию методов, результаты проверить с помощью встроенных функций.



Ниже приведен график функции:



## Теоретические сведения

### Метод бисекции (деления отрезка пополам)

Метод бисекции или метод деления отрезка пополам — простейший численный метод для решения нелинейных уравнений вида F(x)=0. Предполагается только непрерывность функции F(x).

Задача заключается в нахождении корней нелинейного уравнения

 (1.1)

Для начала итераций необходимо знать интервал [xL,xR] значений x, где находится единственный корень. Произведение значений функции на краях этого интервала получится меньше нуля:

 (1.2)

То есть функция меняет знак на данном интервале. Выберем точку внутри интервала

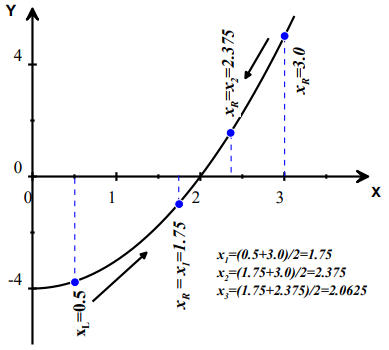
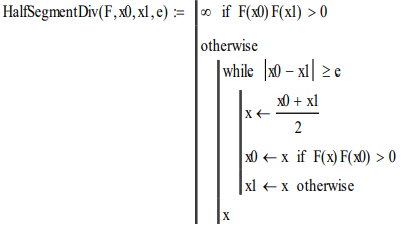


Рис. 1.1 Графическое представление метода бисекций (деления отрезка пополам)

 . (1.3)

Разобьём этот интервал на два [xL,xM] и [xM,xR]. Теперь найдём новый интервал, в котором функция меняет знак. Пусть и соответственно корень находится внутри интервала [xL,xM]. Тогда обозначим xR=xM и повторим описанную процедуру до достижения требуемой точности. За количество итераций N первоначальный отрезок делится в 2N раз. На рисунке 1.1 приведено графическое представление данного метода.

Ниже приведена программная реализация данного численного метода в пакете MathCAD:



Данная функция имеет следующие входные параметры:

* F – нелинейная функция F(x);
* x0 – начало интервала на котором производится поиск решения уравнения F(x)=0;
* x1 – конец интервала на котором производится поиск решения уравнения F(x)=0;
* e – точность решения

В случае если на данном интервале решения не обнаружено, функция возвращает ∞.

Решение нашего уравнения методом бисекции:





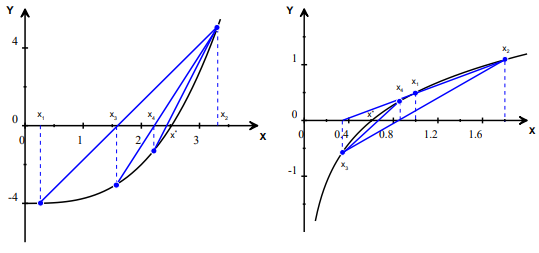
### 

### Метод хорд (метод линейной интерполяции)

Приведём метод, основанный на нахождении xi+1 по двум предыдущим приближениям xi и xi-1 с помощью линейной интерполяции, называемый методом хорд.

Идея метода состоит в том, что по двум точкам и построить прямую (то есть хорду, соединяющую две точки графика) и взять в качестве следующего приближения  абсциссу точки пересечения этой прямой с осью Ox. Иными словами, приближённо заменить на этом шаге функцию её линейной интерполяцией, найденной по двум значениям x: xi-1 и xi. (Линейной интерполяцией функции F(x) назовём такую линейную функцию L(x), значения которой совпадают со значениями F(x) в двух фиксированных точках, в данном случае – в точках xi-1 и xi.).

В зависимости от того, лежат ли точки xi-1 и xi по разные стороны от корня x\* или же по одну и ту же сторону, получаем следующие графические представления:



|  |
| --- |
| Рис.1.2. Построение последовательного приближения по методу хорд |

Итак, очередное последовательное приближение будет зависеть от двух предыдущих: . Найдём выражение для функции. Интерполяционную линейную функцию L(x) будем искать как функцию с угловым коэффициентом, равным разностному отношению:

 , (1.4)

построенному для отрезка между xi-1 и xi, график которой проходит через точку Mi:

. (1.5)

Решая уравнение L(x) = 0, находим

, (1.6)

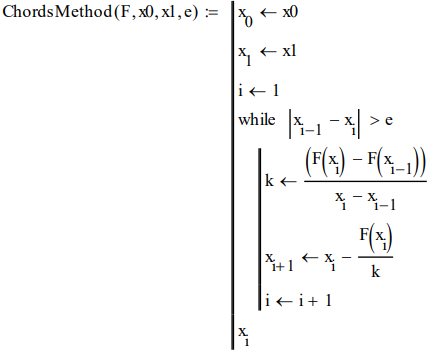
то есть

 (1.7)

Заметим, что величина ki может рассматриваться как разностное приближение для производной в точке xi. Тем самым полученная формула (1.7) – это разностный аналог итерационной формулы метода Ньютона.

Вычисление по формуле вычисления ведутся по формуле (1.7) при i=1,2,3…, начиная с двух приближений x0 и x1, взятых, по возможности, поближе к корню x\*. При этом не предполагается, что x\* лежит между x0 и x1 (и что значения функции F в точках x0 и x1 имеют разные знаки). При этом не гарантируется, что корень попадёт на отрезок между xi-1 и xi на каком-либо следующем шаге (хотя это и не исключено). В таком случае затруднительно дать оценку погрешности, с которой xi+1 приближает истинное значение корня x\*, поэтому довольствуются таким эмпирическим правилом: вычисления прекращают, когда будет выполнено неравенство , где – желаемая точность нахождения корня. При этом полагают приближённое значение корня равным .

Приведем программную реализацию метода хорд в пакете MathCAD:



Данная функция имеет следующие входные параметры:

* F – нелинейная функция F(x);
* x0 – начало интервала на котором производится поиск решения уравнения F(x)=0;
* x1 – конец интервала на котором производится поиск решения уравнения F(x)=0;
* e – точность решения

В случае если на данном интервале решения не обнаружено, функция возвращает ∞.

Решение нашего уравнения методом хорд:





### Метод касательных (метод Ньютона)

Рассмотрение метода одной касательной позволяет предположить, что итерации станут приближаться к корню ещё быстрее, если мы будем выбирать касательную вместо секущей не только на первом, а на каждом шаге. Ясно, что тогда формула итераций будет иметь вид

 (1.8)

Этот метод называется методом касательных, или методом Ньютона. Действительно, последовательные приближения метода Ньютона сходятся гораздо быстрее, чем в общем методе итераций (скорость сходимости приближений в котором, напомним, та же, что у геометрической прогрессии со знаменателем  при ).



Рис.1.3. Последовательные приближения метода Ньютона (касательных)

Геометрический смысл метода Ньютона состоит в том, что на каждом шаге мы строим касательную к графику точке очередного последовательного приближения xi, а за следующее приближение xi+1 берём точку пересечения этой касательной с осью Ox. Тем самым наклон прямой подстраивается на каждом шаге наилучшим образом (ведь кривизну графика, связанную со второй производной, мы не учитываем, и поэтому неизвестно, в какую сторону от касательной отклонится график).

Заметим, что по-другому идею метода Ньютона мы можем описать так: на каждом шаге вместо исходного уравнения F(x)=0 мы решаем приближённое, линеаризованное в точке xi уравнение:

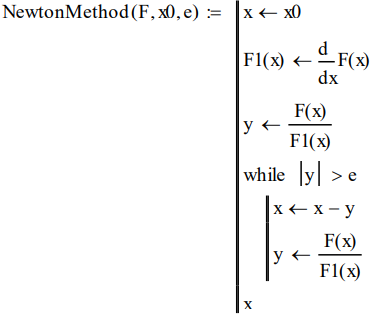
 (1.9)

в котором левая часть – это многочлен Тейлора первого порядка для функции F(x) в точке xi, то есть линейная функция

 (1.10)

Решением линеаризованного уравнения служит следующее приближение xi+1, в то время как решением исходного точного уравнения F(x) служит искомый корень x\*.

Приведем программную реализацию метода касательных или метода Ньютона в пакете MathCAD:



Данная функция имеет следующие входные параметры:

* F – нелинейная функция F(x);
* x0 – начальное приближение поиск решения уравнения F(x)=0;
* e – точность решения

Решение нашего уравнения методом Ньютона:





### Метод секущих

В качестве функции берут любую постоянную , знак которой совпадает со знаком производной в окрестности E (и, в частности, на отрезке, соединяющем x0 и x\*). Постоянная не зависит также и от номера шага i. Тогда формула итераций оказывается очень проста:



и на каждой итерации нужно один раз вычислить значение функции F(x).

Выясним смысл этой формулы, а также смысл условия о совпадении знаков и . Рассмотрим прямую, проходящую через точку на графике с угловым коэффициентом . Тогда уравнением этой прямой будет:

.

Найдём точку пересечения этой прямой с осью $ Ox$из уравнения

,

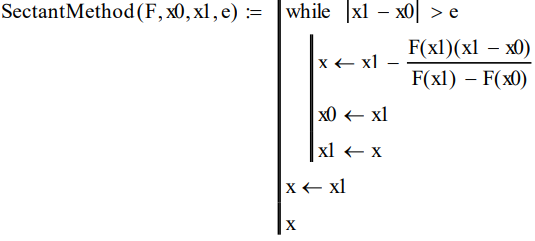
откуда. Следовательно, эта прямая пересекает ось Ox как раз в точке следующего приближения. Тем самым получаем следующую геометрическую интерпретацию последовательных приближений. Начиная с точки x0, через соответствующие точки графика  проводятся секущие с угловым коэффициентом  того же знака, что производная . (Заметим, что, во-первых, значение производной вычислять не обязательно, достаточно лишь знать, убывает функция  или возрастает; во-вторых, что прямые, проводимые при разных xi, имеют один и тот же угловой коэффициент k и, следовательно, параллельны друг другу.) В качестве следующего приближения к корню берётся точка пересечения построенной прямой с осью Ox.



Рис. 1.4. Графическая интерпретация метода секущих

На рисунке 1.4 изображены итерации при, в случае . Мы видим, что последовательные точки xi приближаются к корню, оставаясь всё время с одной стороны от него.

Программная реализация данного метода в пакете MathCAD выглядит так:



Данная функция принимает следующие параметры:

* F – нелинейная функция F(x);
* x0 – начало интервала на котором производится поиск решения уравнения F(x)=0;
* 1 – конец интервала на котором производится поиск решения уравнения F(x)=0;
* e – точность решения

Решение нашего уравнения методом секущих:





Проверка результатов с помощью встроенных функций:





# Задание № 2

Решить систему уравнений с тремя неизвестными методом Гаусса-Жордана

Составить функции, реализующие метод, проверить решение с помощью встроенных функций MathCAD.

## Теоретические сведения

### Метод Гаусса (метод последовательного исключения неизвестных)

Пусть дана система m линейных уравнений с n неизвестными . Требуется найти ее общее решение, если она совместна, или установить ее несовместность. Выпишем расширенную матрицу системы



Назовем элементарными операциями следующие действия с матрицами:

1. перестановка строк;
2. умножение строки на число, отличное от нуля;
3. сложение строки с другой строкой, умноженной на число.

Отметим, что при решении системы уравнений, в отличие от вычисления определителя и нахождения ранга, нельзя оперировать со столбцами.

Легко проверить, что если по матрице, полученной из A\* выполнением элементарной операции, восстановить систему уравнений, то новая система будет равносильна исходной.

Цель алгоритма – с помощью применения последовательности элементарных операций к матрице A\* добиться, чтобы каждая строка, кроме, быть может, первой, начиналась с нулей, и число нулей до первого ненулевого элемента в каждой следующей строке было больше, чем в предыдущей.

Шаг алгоритма заключается в следующем. Находим первый ненулевой столбец в матрице A\*. Пусть это будет столбец с номером i. Находим в нем ненулевой элемент и строку с этим элементом меняем местами с первой строкой. Чтобы не нагромождать дополнительных обозначений, будем считать, что такая смена строк в матрице A\* уже произведена, то есть . Тогда ко второй строке прибавим первую, умноженную на число , к третьей строке прибавим первую, умноженную на число , и т.д. В результате получим матрицу



(Первые нулевые столбцы, как правило, отсутствуют.)

Если в матрице встретилась строка с номером k, в которой все элементы равны нулю, а , то выполнение алгоритма останавливаем и делаем вывод, что система несовместна. Действительно, восстанавливая систему уравнений по расширенной матрице, получим, что k-е уравнение будет иметь вид



Этому уравнению не удовлетворяет ни один набор чисел .

Матрицу  можно записать в виде:

,

где

.

По отношению к матрице выполняем описанный шаг алгоритма. Получаем матрицу

,

где ,. Эту матрицу снова можно записать в виде

,

и к матрице C\* снова применим описанный выше шаг алгоритма.

Процесс останавливается, если после выполнения очередного шага новая уменьшенная матрица состоит из одних нулей или если исчерпаны все строки. Заметим, что заключение о несовместности системы могло остановить процесс и ранее.

Если бы мы не уменьшали матрицу, то в итоге пришли бы к матрице вида

.

Далее выполняется так называемый обратный ход метода Гаусса. По матрице составляем систему уравнений. В левой части оставляем неизвестные с номерами, соответствующими первым ненулевым элементам в каждой строке, то есть . Заметим, что ранг  равен рангу. Остальные неизвестные переносим в правую часть. Считая неизвестные в правой части некоторыми фиксированными величинами, несложно выразить через них неизвестные левой части.

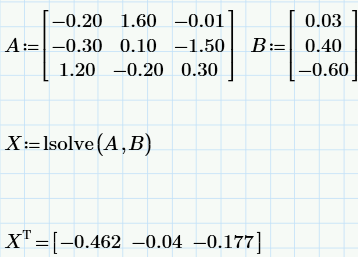
Теперь, придавая неизвестным в правой части произвольные значения и вычисляя значения переменных левой части, мы будем находить различные решения исходной системы . Чтобы записать общее решение, нужно неизвестные в правой части обозначить в каком-либо порядке буквами, включая и те неизвестные, которые явно не выписаны в правой части из-за нулевых коэффициентов, и тогда столбец неизвестных можно записать в виде столбца, где каждый элемент будет линейной комбинацией произвольных величин  (в частности, просто произвольной величиной Ck). Эта запись и будет общим решением системы. Здесь  .

Если система была однородной, то получим общее решение однородной системы. Коэффициенты при C1, взятые в каждом элементе столбца общего решения, составят первое решение из фундаментальной системы решений, коэффициенты при C2 – второе решение и т.д.

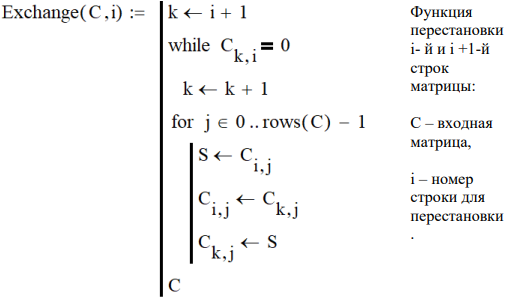
Фундаментальную систему решений однородной системы можно получить и другим способом. Для этого одной переменной, перенесенной в правую часть, нужно присвоить значение 1, а остальным – нули. Вычислив значения переменных в левой части, получим одно решение из фундаментальной системы. Присвоив другой переменной в правой части значение 1, а остальным – нули, получим второе решение из фундаментальной системы и так далее.



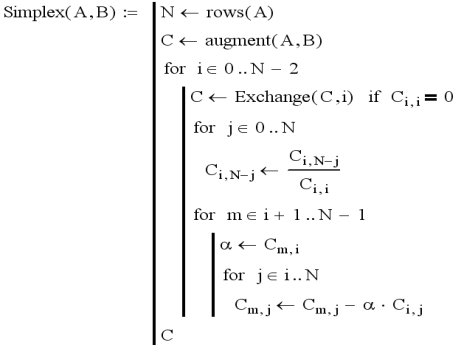
Решим данную систему уравнений средствами MathCAD:



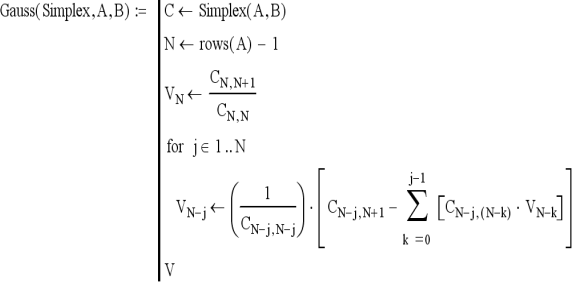
А теперь напишем функции реализующие решение системы уравнений методом Гаусса-Жордана:



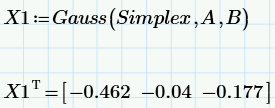
Следующая функция формирует расширенную матрицу системы С, которая является результатом склеивания матриц A и B, и приводит матрицу С к ступенчатой форме.



Следующая функция осуществляет обратный ход метода Гаусса после приведения матрицы С к ступенчатой форме.



Результаты работы с данными функциями:



Как видим результаты вычислений встроенными средствами MathCAD и результаты, полученные на основе представленной реализации метода Гаусса-Жордана, совпали.

# Задание № 3

Решить задачу Коши для дифференциального уравнения

 (3.1)

на отрезке при заданном начальном условии

 (3.2)

и шаге интегрирования h:

* методом Эйлера;
* методом Рунге-Кутта 4-го порядка точности.

## Теоретические сведения

### Метод Эйлера

Метод Эйлера занимает в теории численных методов решения ОДУ ключевую позицию. При этом будем считать, что вычисления проводятся с **расчетным шагом**

****

расчетными точками (узлами) служат точки  промежутка и целью является нахождение функции в виде таблицы приближенных значений yi решения у=у(х) задачи в расчетных точках хi

|  |  |
| --- | --- |
| X | Y |
|  |  |
|  |  |
| … | … |
|  |  |

Пользуясь тем, что в точке x0 известно значение решения y(x0)=y0, и значение его производной  (согласно (3.1)), можно записать уравнение касательной к графику искомой функции у=у(х) в точке (х0;у0):

 (3.3)

При достаточно малом  шаге h ордината , полученная подстановкой в правую часть (3.3) значения , по непрерывности  должна мало отличаться от ординаты y(x1) решения y(x) задачи (3.1). Следовательно, точка (x1,y1) пересечения касательной (3.3) с прямой х=х1 может быть приближенно принята за новую начальную точку. Через эту точку снова проведем прямую которая уже приближенно отражает поведение касательной к у=у(х) в точке (х1;у(х1)). Подставляя сюда х = х2, х2= х1+h, иначе, пересекая эту «касательную» прямой х=х2, получим приближение значения у(х2) значением , и т.д. В итоге этого процесса, определяемого формулой

,

и называемого методом Эйлера, получим график решения у=у(х) данной задачи Коши (3.1)-(3.2) в виде приближенной ломанной, составленной из отрезков приближенных касательных (Рис. 3.1) откуда происходит другое название – метод ломаных.



Рис.3.1. Геометрическая интерпретация метода Эйлера.

Анализ метода Эйлера показывает, что локальная ошибка дискретизации на одном щаге равна O(h). Это обычно выражают утверждением, что метод Эйлера имеет **первый порядок**. Практическим следствием этого факта является ожидание того, что при уменьшении h приближённое решение будет всё более точным и при стремлении h к нулю будет сходиться к точному решению с линейной скоростью по h; т.е. мы ожидаем, что при уменьшении шага h вдвое ошибка уменьшится примерно в два раза.

Очень медленная сходимость при уменьшении h характерна для методов первого порядка и служит препятствием для их использования.

### Методы типа Рунге-Кутта

Как пример одного из подходов к построению методов, погрешность которых при стремлении h к нулю убывает с более высокой скоростью, мы рассмотрим **Метод Хьюна**, определяемый формулой



Обратите внимание, что мы просто заменили f(xk ,yk ) в методе Эйлера на среднее значение функции f, вычисленных в двух различных точках. Метод Хьюна известен также как модифицированный метод Эйлера или **метод Рунге-Кутта** второго порядка и, имеет локальную ошибку дискретизации O(h2).

Наиболее используемым из методов Рунге-Кутта является классический метод четвёртого порядка, называемый «методом одной шестой», и задаваемый формулой

,

где



Здесь f(xk ,yk ), использованное в методе Эйлера, заменено на среднее взвешенное значение f, вычисленных в четырёх различных точках.

Метод Рунге-Кутта четвёртого порядка является одношаговым, также как и метод Эйлера, который иногда называют методом Рунге-Кутта первого порядка. Все такие методы могут быть представлены в общем виде как



с соответствующей функцией g. В случае метода Эйлера функцией g является функцией f, в то время как для метода Хьюна

.

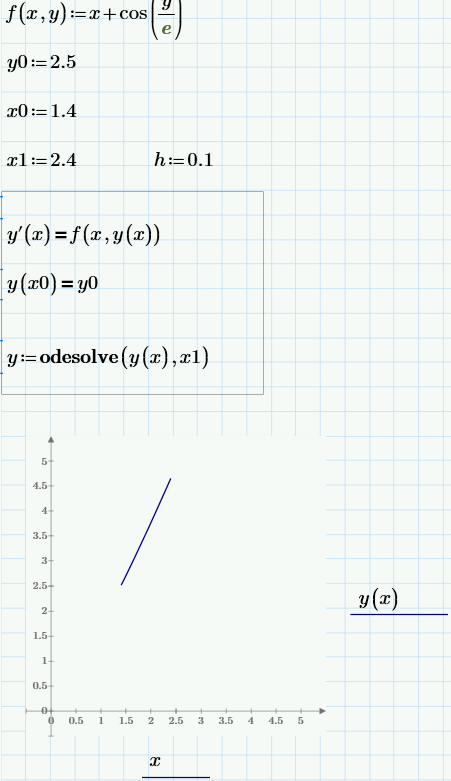
Соответствующая функция для метода Рунге-Кутта четвёртого порядка может быть записана в аналогичном виде.

Решим задачу Коши для дифференциального уравнения:

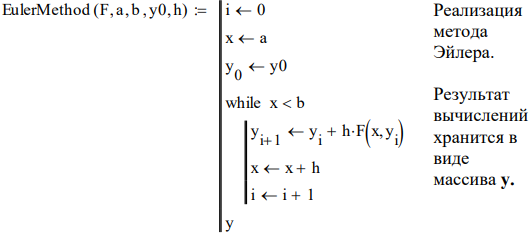


* методом Эйлера;
* методом Рунге-Кутта 4-го порядка точности;
* при помощи встроенных средств пакета MathCAD.

Сначала решим данное дифференциальное уравнение средствами пакета MathCAD с помощью встроенной функции Odesolve:



Теперь реализуем методы Эйлера и Рунге-Кутта 4-го порядка точности для решения уравнений и сравним полученные результаты:



Функция для решения дифференциального уравнения имеет следующие входные параметры:

F – функция двух переменных F(x,y);

a – начальная точка интегрирования;

b  – конечная точка интегрирования;

y0 – начальное условие (значение функции y(a));

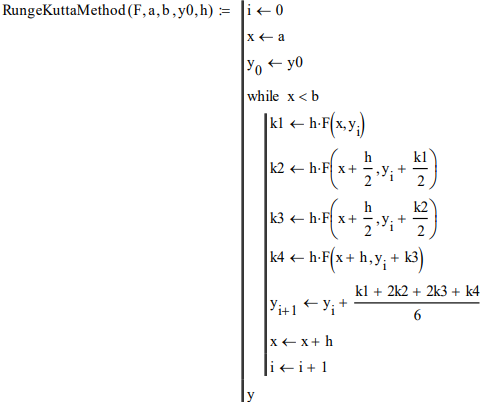
h  – шаг интегрирования.

Решение задачи методом Эйлера:



Аналогичным образом реализуем и вызываем метод Рунге-Кутта 4-го порядка точности.

Результат вычислений хранится в виде массива **y.**



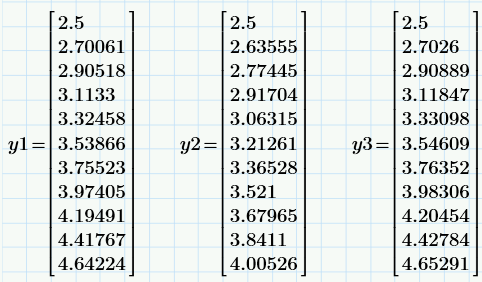
Вызываем метод Рунге-Кутта с параметрами заданными в условии задачи:



Теперь в массиве **y1** хранятся результаты вычислений в расчетных точках, полученные методом Эйлера, в массиве **y2** результаты, полученные при помощи метода Рунге-Кутта 4-го порядка точности.

В массиве **y3** для сравнения представлены значения, полученные при помощи встроенных средств пакета MathCAD:





Вывод: в ходе контрольной работы все задания выполнялись в программе MathCAD, ознакомились с различными методами для определения корней уравнения, с методом Гаусса-Жордана для решения системы уравнения с тремя неизвестными и решением Коши для диф-го уравнения методом Эйлера и методом Рунге-Кутта 4-го порядка точности.

# 

# Литература

1. Б.П. Демидович, И.А. Марон Основы вычислительной математики. – М.: Наука, 1966. с. 276-278.
2. Макаров Е.Г. Инженерные расчеты в MathCAD . - СПб.: Питер, 2005-448 с.
3. С.В. Поршнев, И.В. Беленкова Численные методы на базе MathCAD. СПб.: БХВ-Петербург, 2005-456с.
4. Математика для экономистов на базе MathCAD/Черняк А.А. и др. –СПб.: БХВ-Петербург, 2003-496 с.
5. Решение инженерных задач на ЭВМ/:Практическое руководство /Шуп Терри Е. - М.: Мир, 1982-235 с.