**Министерство образование и науки Российской Федерации**

**Федеральное государственное бюджетное образовательное**

**учреждение высшего образования**

**«Национальный исследовательский университет «МЭИ»»**

Институт тепловой и атомной энергетики

Кафедра «Технологии воды и топлива»

Направление подготовки 13.03.01

**Отчет**

по учебной (ознакомительной) практике студента первого курса

Автор отчета: Горбунов А.А.

Место прохождение практики: НИУ «МЭИ», кафедра «Технологии воды и топлива»

Сроки прохождение практики: 04.07.16-15.07.16

Москва 2016

Этюд 4. Решатели или Великолепная семерка Mathcad.

«Великолепная семёрка» в пакете Mathcad – это семь встроенных функций, которые используют различные численные методы. С помощью этих функций можно решать уравнения или системы уравнений.

Рассмотрим эти функции на примерах задач.

Задача 1. Велосипедист едет из пункта А в пункт Б (S = 3 км), причём ветер направлен в сторону движения велосипедиста (ветер дует в спину). Затем велосипедист возвращается в исходный пункт и едет против ветра. Весь путь велосипедист прошёл за 1 час 15 минут. Какова скорость ветра, если собственная скорость велосипедиста v = 5 км/ч?

Решение. В нашей задаче время в пути t – это суммарное время, затраченное на поездку в одну сторону (по направлению ветра) и в обратную сторону (против ветра). Поэтому наше уравнение будет иметь вид:



**0. solve**

Решим полученное уравнение с помощью команды **solve** (symbolic math – символьная математика).



Мы получили два корня уравнения. Мы видим, что один из корней уравнения отрицательный, а так как скорость ветра не может быть отрицательной величиной, то берём второй корень.



**1. root + root**

Используем команду **root** для решения уравнения. **Root(f(x),x,[a,b])** возвращает значение переменной **x** , для которого функция **f** принимает нулевое значение. Если указаны параметры **a** и **b** , функция **root** находит значение **x** на этом интервале.



Уравнение движения велосипедиста можно преобразовать в квадратное с помощью оператора **simplify** и оператора **coeffs.** Оператор **simplify** (упростить) приводит левую часть исходного выражения к общему знаменателю, умножает обе части уравнения на полученный знаменатель и переносит все слагаемые в левую часть уравнения. Таким способом выделяется функция, которая приравнена к нулю. Оператор **coeffs** находит коэффициенты этой функции.



**2. polyroots**

Если выражение представляет собой полином (квадратный, например), то можно найти все его нули, использовав ещё одну функцию из «великолепной семёрки» - функцию **polyroots**, имеющую в качестве аргумента вектор коэффициентов полинома и возвращающую его нули.



**3. Find**

Чтобы показать работу ещё одной функции из «великолепной семерки», добавим в нашу задачу ещё одного велосипедиста.

Задача 2. Из двух пунктов А и Б навстречу друг другу одновременно выехали два велосипедиста, причём первый велосипедист едет по направлению ветра, а второй велосипедист едет против ветра. Они встречаются в точке, делящей этот участок дороги в золотом соотношении. Найти скорость второго велосипедиста v₂ и скорость ветра, если известна скорость первого велосипедиста v₁ = 7 км/ч, расстояние между пунктами S = 3 км и время t = 20 мин движения велосипедистов до встречи.

Решение. Напишем уравнение золотого сечения применительно к нашей задаче и решим его аналитически, получив нужную формулу.





Оператор **solve** выдал два решения, из которых нам подходит только второе – 1.146 km, так как первое лежит вне рассматриваемого отрезка.

Решим нашу задачу с помощью функции **Find**. Встроенная функция **Find** меняет значение своих аргументов, начиная от начального приближения так, чтобы уравнения системы превратились в тождества.



**4. lsolve**

Поскольку система двух алгебраических уравнений движения двух велосипедистов навстречу друг другу линейна, то можно применить ещё одну функцию из «великолепной семёрки» - функцию **lsolve**.

Функция **lsolve** имеет два аргумента: матрицу коэффициентов при неизвестных СЛАУ (это у нас **М**) и вектор свободных членов **V**. Функция **lsolve** возвращает вектор найденных значений неизвестных. При решении СЛАУ с помощью функции **lsolve** начальные предположения вводить не нужно.



**5 и 6. Minimize и Maximize**

Покажем работу функций **Minimize** и **Maximize** на примере другой задачи.

Задача 3. Определить скорость производства велосипедов – скорость, при которой затраты на их эксплуатацию будут минимальны.

Решение. Затраты на эксплуатацию велосипедов состоят из двух частей: почасовой оплаты работников, обратно пропорциональной скорости производства велосипедов, и затрат на электричество для станков, пропорциональных квадрату скорости производства велосипедов.

Пусть а – почасовая оплата работников, b – затраты на электричество для станков, v – скорость производства велосипедов.



Функция **Minimize** меняет значение своего второго аргумента, начиная от заданного предполагаемого значения (у нас это 20 шт/ч) так, чтобы значение первого аргумента (целевой функции **Затраты**) приняло минимальное значение. Если бы мы не минимизировали затраты, а максимизировали, например, прибыть работников, то нужно было бы при решении такой задачи использовать функцию **Maximize**.

**7. Minerr**

Теперь используем последнюю функцию «великолепной семёрки» - функцию **Minerr** (Minimal Error – минимальная ошибка). Функция **Minerr** возвращает значения своих аргументов, при которых система уравнений будет максимально приближена к системе тождеств.

Функция **Minerr** является самой главной функцией, поскольку она может заменить практически все функции «великолепной семёрки».



Заключение.

Каждая из рассмотренных функций «великолепной семёрки Mathcad» обладает своими особенностями и ограничениями. Прежде чем приступить к работе, следует подумать, каким из операторов будет лучше всего сделать поставленную задачу. Любому человеку необходимо (а иногда и достаточно) освоить «великолепную семёрку», особенности численных, графических и аналитических методов решения задач, чтобы успешно решать задачи.