Лабораторная работа 9. Работа с файлами.

Цель: считывание данных с файла. Создание маплета для построения кривой и плоскости по точкам.

## Считывание с файла.

Синтаксис основных команд для работы с файлами очень близок синтаксису языка программирования С.

Перед чтением или записью данных в файл его необходимо открыть. Хотя все команды чтения и записи в Maple открывают файл автоматически, лучше делать это явно с использованием функции fopen.

#### Команда fopen.

Назначение команды: открытие файла для буферизованного чтения или записи.

Формат вызова:

fopen(<имя\_файла>,<режим\_доступа>,<вид>)

Параметры:

<имя\_файла> ‑ имя файла (возможно указание полного пути к файлу);

<режим\_доступа> ‑ режим доступа к файлу ‑ одно из ключевых слов: READ (для чтения файла), WRITE (для чтения и записи в файл с его начала), APPEND (для чтения и записи в конец файла);

<вид> ‑ характер доступа к файлу ‑ одно из ключевых слов: TEXT (текстовый файл — значение по умолчанию) или BINARY (двоичный файл) (необязательный параметр).

Особенности использования команды fopen:

* при указании пути символ обратного слэша \ необходимо дублировать:

\\ (либо можно использовать одинарный прямой слэш /);

* команда возвращает дескриптор файла ‑ целое число;
* если в параметре <вид> указано ключевое слово TEXT, то при чтении и записи в файл символ начала новой строки (\n) будет учитываться в соответствии с правилами соответствующей операционной системы;
* при попытке открыть ранее открытый файл будет выдано сообщение об ошибке;
* если файл открывается для чтения, а его не существует, будет выдано сообщение об ошибке;
* если файл открывается для записи, а его не существует, он будет создан (при ошибке создания будет выдано сообщение об ошибке).

После выполнения всех операций над файлом его следует закрыть, используя ранее созданный дескриптор файла, с помощью команды fclose.

#### Команда fclose.

Назначение команды: закрытие файла.

Формат вызова:

fclose(<список\_файлов>)

Параметр:

<список\_файлов> ‑ список из одного или нескольких дескрипторов или имен файлов.

Особенности использования команды fclose:

* закрывается ранее открытый файл (если файл не был открыт, то при указании имени файла никакого сообщения об ошибке выдано не будет, а при указании дескриптора — будет);
* закрытие файла гарантирует, что все изменения, сделанные в файле, будут записаны на диск;

Пример.

f1:=fopen("d:\\inet\\temp\\data1.txt", READ, TEXT); Открытие текстового файла для чтения (используется двойной обратный слэш)

f2:=fopen("d:/inet/temp/data2.bin",WRITE,BINARY); Открытие бинарного файла для записи (используется одинарный прямой слэш)

f3:=fopen("d:/inet/temp/data2.bin",WRITE,BINARY); Попытка открыть уже открытый файл.

Error, (in fopen) file already open

fclose(f1, f2); Закрытие файлов, описываемых дескрипторами f1 и f2

f4:=fopen("d:/inet/temp/data.ttt",READ); Попытка открыть несуществующий файл

Error, (in fopen) file or directory does not exist

#### Команды для работы с файлами

ﬁlepos ‑ Получение или установка текущей позиции в файле

fprint ‑ Запись в файл по формату

fscanf ‑ Чтение из файла по формату

readdata ‑ Чтение числовых данных из текстового файла

readline ‑ Чтение одной строки из текстового файла

writedata ‑ Запись числовых данных в текстовый файл

writeline ‑ Запись одной строки в текстовый файл

#### Команда fscanf.

Назначение команды: чтение из файла по формату.

Формат вызова:

fscanf(<файл>,<формат>)

Параметры:

<файл> — имя файла или дескриптор;

<формат> — формат (маска) чтения данных.

Команда fscanf возвращает список прочитанных элементов. Если успешно не был прочитан ни один элемент, то при окончании читаемых данных возвращается пустой список NULL. Параметр <формат> представляет собой строку, состоящую из элементов вида

%<необяз><ширина><модификатор><код>

Параметры элемента строки формата:

* <необяз> символ \* (звездочка), указывающий, что объект должен быть прочитан, но не должен присутствовать в возвращаемом результате (необязательный параметр);
* <ширина> — максимальное количество символов в читаемом элементе (необязательный параметр);
* <модификатор> — модификатор типа возвращаемого значения элемента (необязательный параметр):
* l, L — не влияет на результат, оставлены для совместимости с языком C;
* z, Z — чтение комплексного числа (может предшествовать одному из кодов, соответствующих числовому формату: d, o, x, e, f, g);
* <код> — тип элемента для чтения (наиболее важные коды приведены таблице ниже).

Таблица. Некоторые коды в описании формата

|  |  |
| --- | --- |
| d | Десятеричное целое со знаком или без знака |
| o | Восьмеричное целое без знака |
| x | Шестнадцатеричное целое без знака |
| e,f,g | Вещественное число с плавающей точкой. Допускается использование символов e или E, указывающих на степень десяти. Допускаются специальные значения inf (бесконечность) и NaN (неопределенность) |
| c | Символ (включая пробел, табуляцию, конец строки). При указании параметра <ширина> читается указанное число символов |
| s | Строка до пробела, табуляции или конца строки |
| [ ] | Символы, указанные между квадратными скобками. Если вначале списка указан символ (крышка), это означает чтение всех символов, кроме указанных.  Символ - (дефис) применяется для указания диапазонов. |
| a | Алгебраическое выражение до пробела, табуляции и символа конца строки |
| И др. |  |

Формат использования команды fscanf совпадает с форматом команды scanf для разбора произвольного текста. Разбор возвращаемого результата может производиться с помощью команды op.

Пример.

Файл d:\inet\temp\data1.txt содержит следующие строки:

Три десятичных числа: 2, -4.23, 0.45e1

Число в шестнадцатеричном формате: 2ef

Два слова: первое второе

Таблица размером 2x3

-1 3 144

2 12 -576

f:=fopen("d:\\inet\\temp\\data1.txt", READ, TEXT);

#Открываем файл для чтения

fscanf(f,"%s%s%s%d,%f,%f"); # Чтение 1-й строки: 6 элементов

[“Три”, “десятичных”, “числа:”, 2, −4.23, 4.5]

fscanf(f,"%s%s%s%s%x"); # Чтение 2-й строки: 5 элементов (2ef в шестнадцатиричной системе - это 751 в десятиричной)

[“Число”, “в”, “шестнадцатиричном”, “формате:”, 751]

fscanf(f,"%s%s%s%s,%s"); # Чтение 3-й строки: 6 элементов

[“Два”, “слова:”, “первое”, “второе”]

s:=fscanf(f,"%\*s%\*s%d%\*c%d"); # Чтение 4-й строки: первые слова и символ "x" не выводим

s := [2, 3]

m:=s[1];n:=s[2]; # Разбираем результат чтения 4-й строки

m := 2

n := 3

fscanf(f,"%{"||m||","||n||"}da"); # Читаем таблицу 2x3 из 5-й и 6-й строк

−1 3 144

2 12 -576

fclose(f); # Закрываем файл

#### Команда fprintf.

Назначение команды: запись в файл по формату.

Формат вызова:

fprintf(<файл>,<формат>,<список\_выражений>)

Параметры:

<файл> — имя файла или дескриптор;

<формат> — формат (маска) записи данных;

<список\_выражений> — список выражений для форматирования и вывода.

Параметр <формат> представляет собой строку, состоящую из элементов вида

%<флаги><ширина><точность><модификаторы><код>

Параметры элемента строки формата:

<флаги> — одно или несколько значений (см. табл. 6.3) (необязательный параметр);

<ширина> — максимальное количество символов в записываемом элементе (необязательный параметр);

<точность> — количество символов для вывода дробной части вещественных чисел (необязательный параметр);

<модификаторы> — модификатор типа аналогично используемому в команде fscanf (необязательный параметр);

<код> — тип элемента для записи аналогично используемому. в команде fscanf.

Таблица. Флаги в описании формата команды fprintf

|  |  |
| --- | --- |
| Код | Описание |
| + | число выводится с принудительным выводом лидирующего знака числа (в т.ч. плюса) |
| - | выравнивание числа по левому краю, если по умолчанию по правому, и наоборот |
| <пробел> | вывод числа с лидирующим знаком «-», если он имеется, и пробелом в противном случае |
| 0 | вывод нуля между знаком и первой цифрой, если происходит выравнивание по правому краю и задан параметр <ширина> |
| {} | опции для вывода прямоугольных таблиц |

Команда fprintf возвращает количество символов, записанных в файл.

При выводе с помощью команды fprintf автоматический переход на следующую строку не происходит. Для перевода строки необходимо указать символы \n. Аналогично команде fprintf в Maple есть команда printf для форматированного вывода на экран.

Пример.

f:=fopen("d:\\inet\\temp\\data2.txt",WRITE);

fprintf(f,"Матрица %s состоит из %d строк и %d столбцов\n","A", 2, 2); # Возвращаемое значение 41 говорит, что было записано 42 символа, \n - перевод строки

fprintf(f,"%.2f %.2f\n%.2f %.2f",2.129,3.3,-2.23,2.4503); # Будут записаны вещественные числа с двумя знаками после запятой

fclose(f);

f := 0

42

20

Файл d:\inet\temp\data2.txt содержит следующие строки:

Матрица A состоит из 2 строк и 2 столбцов

2.13 3.30

-2.23 2.45

#### Команда readline.

Назначение команды: чтение строки из файла.

Формат вызова:

readline(<файл>)

Параметр:

<файл> — имя файла или дескриптор.

Особенности использования команды readline:

* команда читает строку до символа окончания строки (т.е. символа \n);
* символ окончания строки функция не возвращает;
* если строки для чтения отсутствуют (достигнут конец файла), то функция возвращает 0.

Пример.

f := fopen("d:\\inet\\temp\\data1.txt", READ);

line := readline(f): # На экран содержимое строки пока не выводим

while line <> 0 do

printf("%s\n", line);

line := readline(f);

end do:

fclose(f);

f := 0

Три десятичных числа: 2, -4.23, 0.45e1

Число в шестнадцатиричном формате: 2ef

Два слова: первое второе

Таблица размером 2x3

-1 3 144

2 12 -576

#### Команда writeline.

Назначение команды: запись строки в файл.

Формат вызова:

writeline(<файл>,<список\_строк>)

Параметры:

<файл> — имя файла или дескриптор;

<список\_строк> — одна или несколько строк.

Команда writeline возвращает количество символов, записанных в файл.

Каждая строка в списке начинается с новой строки.

Пример.

f := fopen("d:\\inet\\temp\\data3.txt", WRITE);

writeline(f, "Строка N1", "Строка N2");

fclose(f);

f := 0

20

Для чтения и записи больших объемов числовых данных можно использовать команды readdata, writedata.

#### Команда readdata.

Назначение команды: чтение числовых данных из файла.

Формат вызова 1:

readdata(<файл>,<формат>)

Формат вызова 2:

readdata(<файл>,<число\_колонок>)

Формат вызова 3:

readdata(<файл>,<формат>,<число\_колонок>)

Параметры:

<файл> — имя файла или дескриптор;

<формат> — ключевое слово integer, ﬂoat или string или список этих слов, задающих формат файла;

<число\_колонок> — положительное целое число, указывающее количество колонок данных в файле.

Особенности команды readdata:

* файл, из которого читаются данные, должен состоять из числовых данных, организованных в несколько колонок;
* фактически readdata — это применение команды readline и затем команды sscanf для разбора прочитанной строки по формату;
* при использовании формата 3 считается, что в файле находятся вещественные числа;
* если в файле одна колонка, то возвращается список значений, в противном случае — список списков.

Пример.

Файл d:\inet\temp\matrix.dat содержит следующие 4 колонки данных:

1 2.34 4.15 3.54

2 3.74 -1.06 7.22

3 -0.45 10.93 6.34

4 7.03 -0.88 3.43

5 8.0 4.93 0.75

f := fopen("d:\\inet\\temp\\matrix.dat", READ);

readdata(f, 4); # Чтение 4 колонок как вещественных чисел

fclose(f);

f := 0

[[1., 2.34, 4.15, 3.54], [2., 3.74, −1.06, 7.22], [3., −0.45, 10.93, 6.34],

[4., 7.03, −0.88, 3.43], [5., 8.0, 4.93, 0.75]]

f := fopen("d:\\inet\\temp\\matrix.dat", READ);

readdata(f, [integer, float, float, float]); # Чтение 1-й колонки как целых, а остальных – как вещественных чисел

fclose(f);

f := 0

[[1, 2.34, 4.15, 3.54], [2, 3.74, −1.06, 7.22], [3, −0.45, 10.93, 6.34],

[4, 7.03, −0.88, 3.43], [5, 8.0, 4.93, 0.75]]

f := fopen("d:\\inet\\temp\\matrix.dat", READ);

readdata(f, integer); # Чтение только 1-й колонки как целых чисел

fclose(f);

f := 0

[1, 2, 3, 4, 5]

#### Команда writedata.

Назначение команды: запись числовых данных в файл.

Формат вызова:

writedata(<файл>,<данные>,<формат>,<проц\_нечисл>)

Параметры:

<файл> — имя файла или дескриптор;

<данные> — список, вектор, матрица или список списков значений;

<формат> — формат данных аналогично команде readdata (необязательный параметр);

<проц\_нечисл> — процедура, указывающая, как записывать нечисловые данные (необязательный параметр).

Особенности команды readdata:

* обыкновенные дроби перед записью преобразуются к формату чисел с плавающей запятой;
* параметр <проц\_нечисл> обычно используется для записи комплексных чисел.

Пример.

with(LinearAlgebra):

a := RandomMatrix(3, 5); # Создаем матрицу 3x5 из случайных элементов



f := fopen("d:\\inet\\temp\\data4.txt", WRITE);

writedata(f, a);

fclose(f);

f := 0

Файл d:\inet\temp\data4.txt содержит следующие строки:

-76 -32 27 99 92

-72 -74 8 29 -31

-2 -4 69 44 67

Для считывания больших объемов данных также есть команды ImportData и ImportMatrix

## Задание 1.

Создайте с помощью команд Maple в любом доступном каталоге файл «test.dat». В первую строку выведите свою фамилию, имя и отчество, разделяя их запятыми. Во вторую строку запишите дату своего рождения в формате «день.месяц(числом).год». Далее прочитайте записанные данные из «test.dat» и выведите на экран сообщение формата

<число> <месяц (прописью)> <год> года родился

<Имя> <Отчество> <Фамилия>.

## Задание 2.

Создайте матрицу A размеров 5×7 со случайными элементами. Сохраните в файле «matrix.dat» информацию о размерах матрицы, а также ее элементы. Загрузите из этого же файла матрицу B и сравните ее с исходной матрицей A.

## Задание 3.

Создайте матрицу размера 5×4, состоящую из случайных чисел из отрезка [−10; 10] с одним знаком после запятой. Сохраните ее в файле с помощью команды writedata. Загрузите эту матрицу из файла с помощью команды readdata.

## Задание 4.

Считать данные из файла lab9.Txt:

Первый столбец – нумерация, он же время t

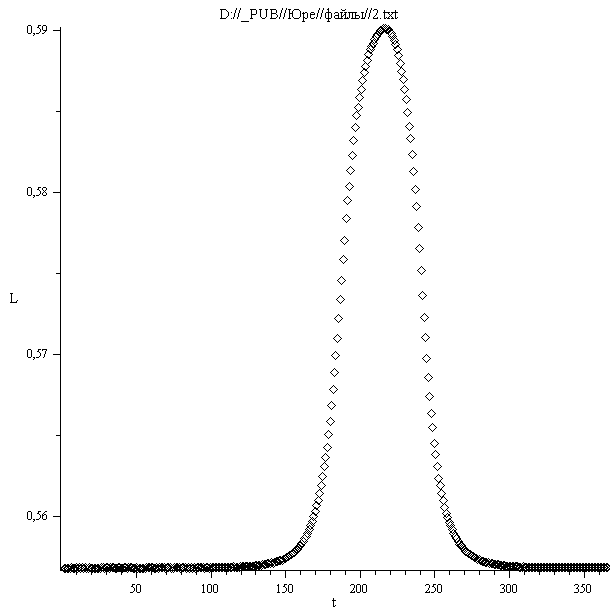
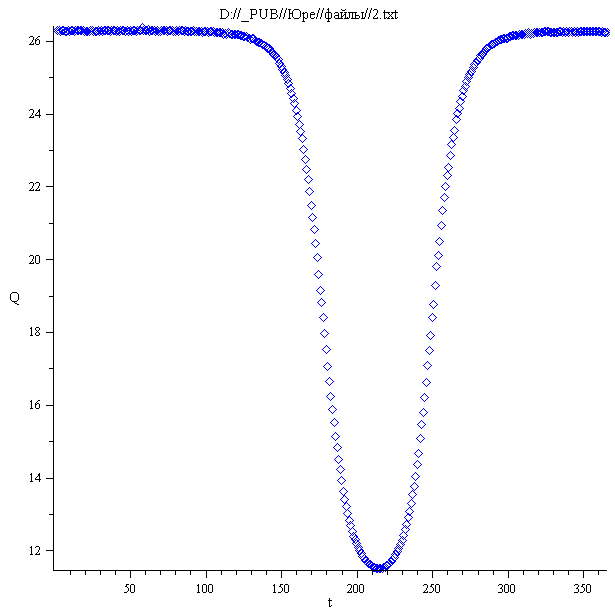
Второй столбец чисел L

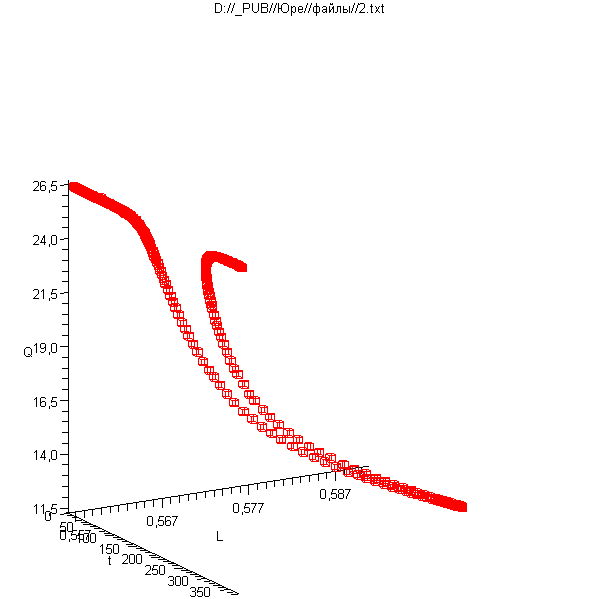
Третий столбец чисел Q

Построить графики:

* L от t
* Q от t
* И трехмерный L,Q,t

Сделать маплет, позволяющий выбрать файл такого типа. И реализующий построение графика по данным из файла.



## Задание 5.

**Метод наименьших квадратов.**

Пусть  ‑ совокупность  точек с различными абсциссами .

Линия построенная, методом наименьших квадратов



коэффициенты находятся из условия минимума расстояния

 (находится минимум функции)

Например, для параболы, построенной методом наименьших квадратов

,

коэффициенты находятся из условия

.

Замечание 1.

Данный метод распространяется и на трехмерный случай:

Заданы  точек  и совокупность  линейно-независимых функций . Требуется найти  таких коэффициентов , чтобы функция , заданная в виде

,

минимизировала сумму квадратов ошибок



**Задание:**

Считать данные из файла **lab9.m**, в котором записаны координаты точек x, y, z. Файл может содержать другое количество цифр.

Получим списки







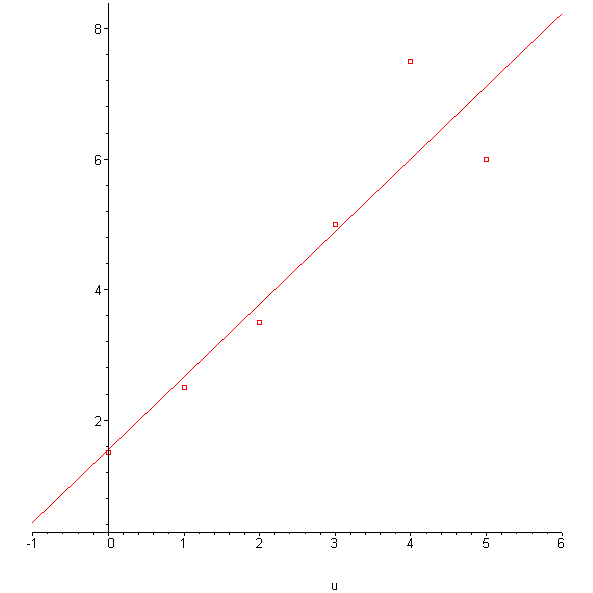
1) методом наименьших квадратов построить линию .

Координаты А, В определяются из условия минимума суммы:





Подставляем найденные коэффициенты в уравнение  и строим графики точек и прямой:



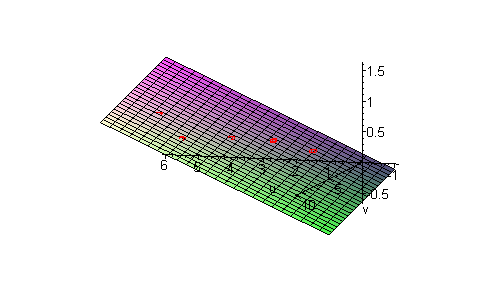
2) методом наименьших квадратов построить плоскость .

Координаты А, В, С определяются из условия минимума суммы:





Подставляем найденные коэффициенты в уравнение  и строим графики точек и плоскости:



3) методом наименьших квадратов построить поверхность .

Координаты А, В, С определяются из условия минимума суммы:





Подставляем найденные коэффициенты в уравнение  и строим графики точек и поверхности:

