

# Применение Matlab для обработки данных, полученных от детекторов космических излучений

Романенко В. С., Шамбин А .И.

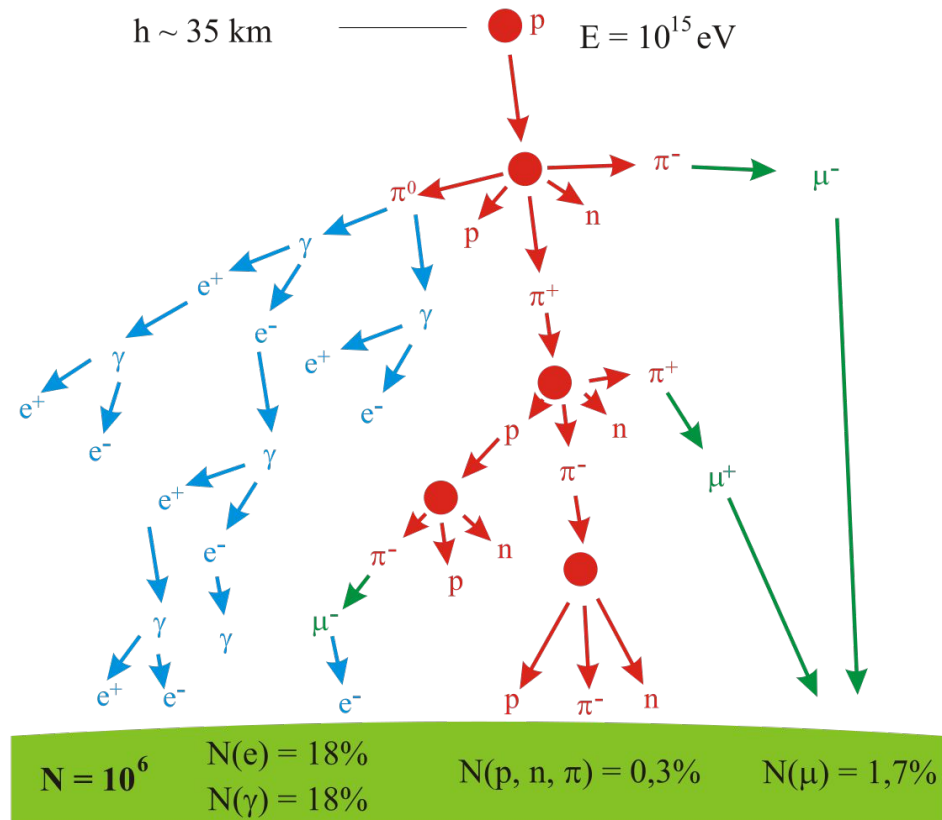
# Постановка задачи

Основной задачей является поиск космического гамма-излучения сверхвысоких энергий. Регистрация осуществляется методом Широких Атмосферных Ливней (ШАЛ)

# Что такое ШАЛ?

ШАЛ представляет собой каскад вторичных частиц, которые образуются в результате взаимодействия первичной частицы с атмосферой Земли

# Формирование воздушного душа в атмосфере. Первый протон сталкивается с частицей в воздухе, создавая пионы, протоны и нейтроны



# Как восстановить параметры первичной частицы?

Для восстановления параметров  
первичной частицы (направление  
прихода и энергия) строятся  
специальные ливневые установки.

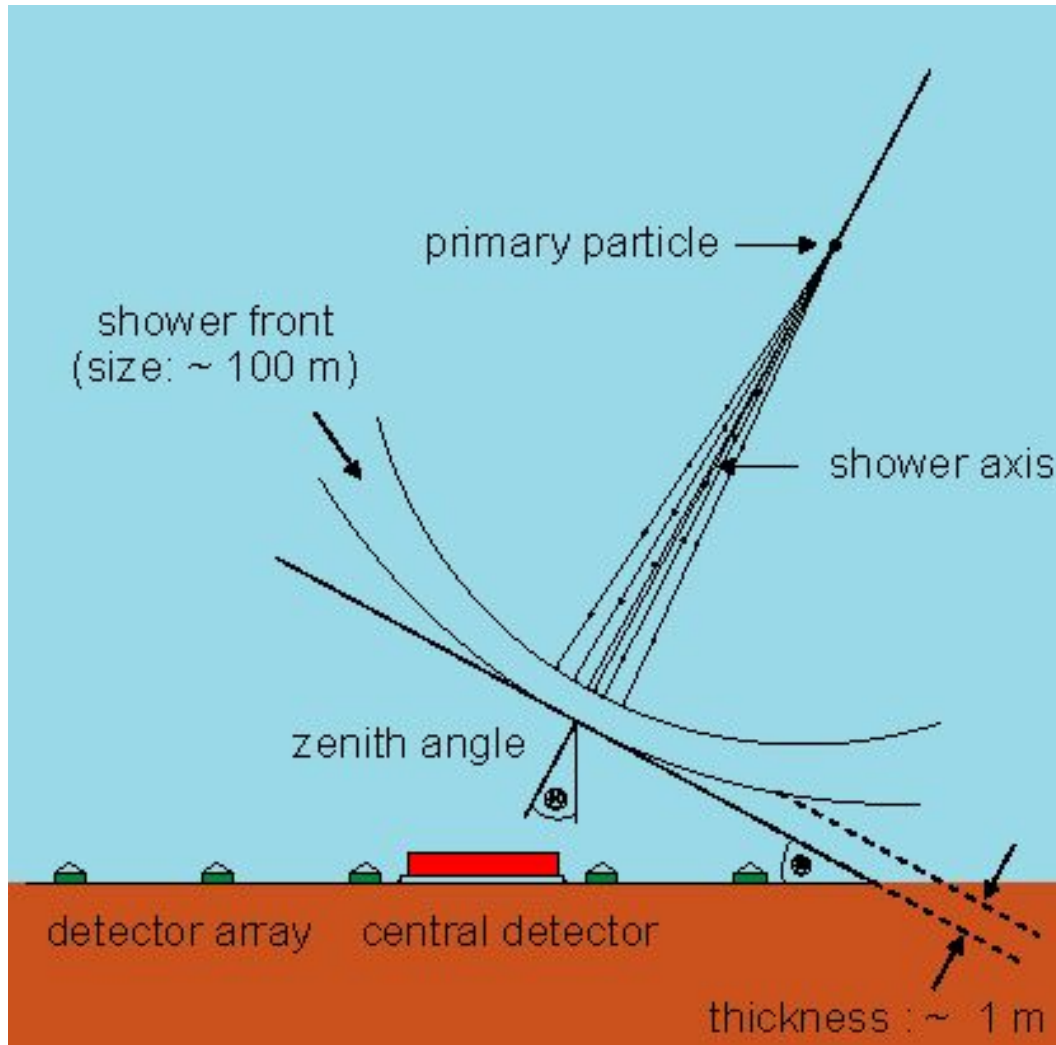
# Как восстановить параметры первичной частицы?

Ливневая установка регистрирует:

1. время прихода фронта ливня для каждого модуля
2. общее энерговыведение в нем.

Из этих данных в последствии  
восстанавливается направление  
прихода и энергия первичной частицы.

# Принцип работы ливневой установки



Первичная частица (primary partice) сталкиваясь с атмосферой образует каскад вторичных частиц (ШАЛ).

Фронт ливня (shower front) проходя через массив детекторов (detector array) записывает информацию о времени прохождения фронта ливня через конкретный детектор и энергосодержание в нем, которое используется для восстановления направления прихода ливня и размера ливня. Центральный детектор используется для определения

# Зачем нужен Matlab?

Все регистрируемые установкой события сохраняются в виде бинарных файлов, которые могут быть открыты для их обработки. В нашем случае такой средой будет являться MATLAB.



# Функции Matlab для работы с бинарными файлами

`fwrite` – записывает данные в файл

`fopen` – открывает файл для чтения

`fread` – считывает данные из файла

`fclose` – закрывает файл

# Функция fwrite

```
fwrite(<идентификатор  
файла>, <переменная>, <тип  
данных>);
```

<идентификатор файла> -  
указатель на файл, с  
которым придётся работать

# Функция fopen

<идентификатор файла> =  
fopen(<имя файла>, <режим  
работы>)

# Режимы работы функции fopen

Параметр <режим работы>	Описание
'r'	чтение
'w'	запись (стирает предыдущее содержимое файла)
'a'	добавление (создает файл, если его нет)
'r+'	чтение и запись (не создает файл, если его нет)
'w+'	чтение и запись (очищает прежнее содержимое или создает файл, если его нет)
'a+'	чтение и добавление (создает файл, если его нет)
'b'	дополнительный параметр, означающий работу с бинарными файлами, например, 'wb', 'rb', 'rb+', 'ab' и т.п.

# Некорректное открытие файла

Если функция `open()` не может корректно открыть бинарный файл, то она возвращает значение `-1`

# Функция fread

<переменная>=fread(<идентификатор  
файла>, <размер>, <точность>)

# Функция fclose

fclose (<идентификатор файла>)

# Пример 1

Требуется создать файл `my_file.dat`, записать в него значения вектор-строки  $A=(1, 2, 3, 4, 5)$ , затем открыть файл и вывести его содержимое на экран.



# M-функция

```
A = [1 2 3 4 5];
```

```
fid = fopen('my_file.dat', 'wb'); % открытие файла на запись  
if fid == -1 % проверка корректности открытия  
    error('File is not opened');  
end
```

```
fwrite(fid, A, 'double'); % запись матрицы в файл (40 байт)  
fclose(fid); % закрытие файла
```

```
fid = fopen('my_file.dat', 'rb'); % открытие файла на чтение  
if fid == -1 % проверка корректности открытия  
    error('File is not opened');  
end
```

```
B = fread(fid, 5, 'double'); % чтение 5 значений double  
disp(B); % отображение на экране  
fclose(fid); % закрытие файла
```

# Результат работы M-функции

В результате работы функции `fwrite` в рабочем каталоге создан бинарный файл `my_file.dat`, имеющий размером 40 байт, содержащий 5 значений типа `double`, которые записаны в виде последовательности байт (по 8 байт на каждое значение). Затем функция `fread()` последовательно считывает все сохраненные байты и автоматически преобразовывает их к типу `double` (каждые 8 байт интерпретируются как одно значение типа `double`).

# Случай неизвестного числа элементов файла

В приведенном примере явно указывалось число элементов (пять) для считывания из файла. Однако часто общее количество элементов бывает наперед неизвестным, либо оно меняется в процессе работы программы. В этом случае было бы лучше считывать данные из файла до тех пор, пока не будет достигнут его конец.

# Функция feof

Используется для проверки конца файла.

Формат:

`feof(<идентификатор файла>)`

Функция `feof` возвращает 1 при достижении конца файла и 0, если конец файла ещё не достигнут.

# Пример 2

Требуется считать данные из файла, содержащего произвольное число элементов.

# M-функция

```
fid = fopen('my_file.dat', 'rb'); % открытие файла на чтение
if fid == -1
    error('File is not opened');
end

B=0;          % инициализация переменной
cnt=1;       % инициализация счетчика
while ~feof(fid) % цикл, пока не достигнут конец файла
    [V,N] = fread(fid, 1, 'double'); % считывание одного
    % значения double (V содержит значение
    % элемента, N – число считанных элементов)
    if N > 0 % если элемент был прочитан успешно, то
        B(cnt)=V; % формируем вектор-строку из значений V
        cnt=cnt+1; % увеличиваем счетчик на 1
    end
end
disp(B);     % отображение результата на экран
fclose(fid); % закрытие файла
```

# Результат работы

В данной программе динамически формируется вектор-строка по мере считывания элементов из входного файла. MatLab автоматически увеличивает размерность векторов, если индекс следующего элемента на 1 больше максимального.

# Параметры функции fread

Функция `fread()` записана с двумя выходными параметрами `V` и `N`. Первый параметр содержит значение считанного элемента, а второй – число считанных элементов. В данном случае значение `N` будет равно 1 каждый раз при корректном считывании информации из файла, и 0 при считывании служебного символа EOF, означающий конец файла.



# Недостаток данных с НЕИЗВЕСТНЫМ ЧИСЛОМ ЭЛЕМЕНТОВ

На такую процедуру тратится много машин-ного времени и программа начинает работать заметно медленнее, чем если бы размерность вектора  $V$  с самого начала была определена равным 5 элементам, например, так

```
V = zeros(5,1);
```

# Задание 1

1. Изучить работу функций *fopen()* и *fread()*, 2. Открыть бинарный файл *18d313r01.DAT*, расположенный в папке *data*, в среде MATLAB, зная, что используемый тип данных в предложенном файле *int32*. После открытия файла он будет сохранен в поле *Variables*, с названием которое вы указали при выполнении функции *fread()* и представлять собой массив состоящий из одного столбца и *n* строк.