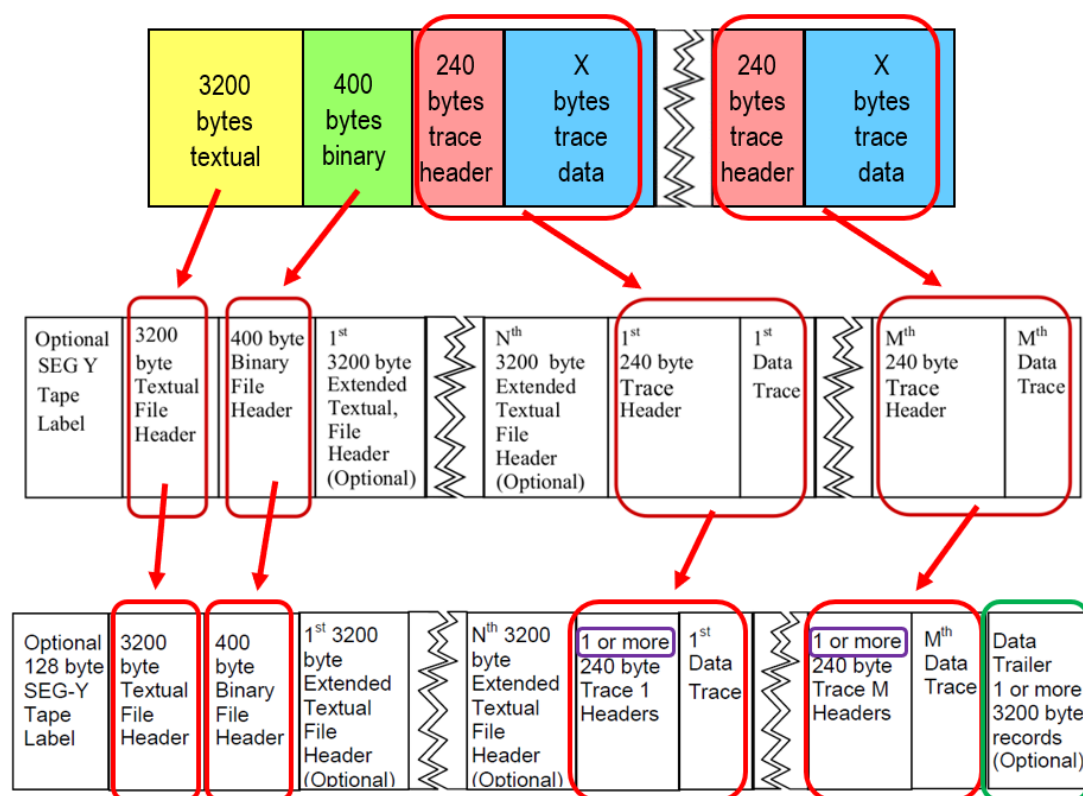


SEG-Y HEADERS WRITER: ИНСТРУКЦИЯ ПОЛЬЗОВАНИЯ

ГОРЕЛИК Г.Д. & МЕЛЬНИКОВ А.О.



САНКТ-ПЕТЕРБУРГ 2025

*Посвящается всем любителям алгоритмизации в
геофизике.*

1. Ввод данных

Для начала зададим входную папку, откуда будем брать файлы для записи заголовков; затем – выходную, куда будут записываться новые файлы.

```
1 clc; clear workspace;
2
3 input_folder = 'C: '; % Folder with raw .sgy files
4 output_folder = 'C: '; % Folder to save the modified
   files
```

Далее зададим цикл, в котором организован перебор всех файлов в папке в соответствии с именем каждого, в нашем случае 50 (рис. 1.1).

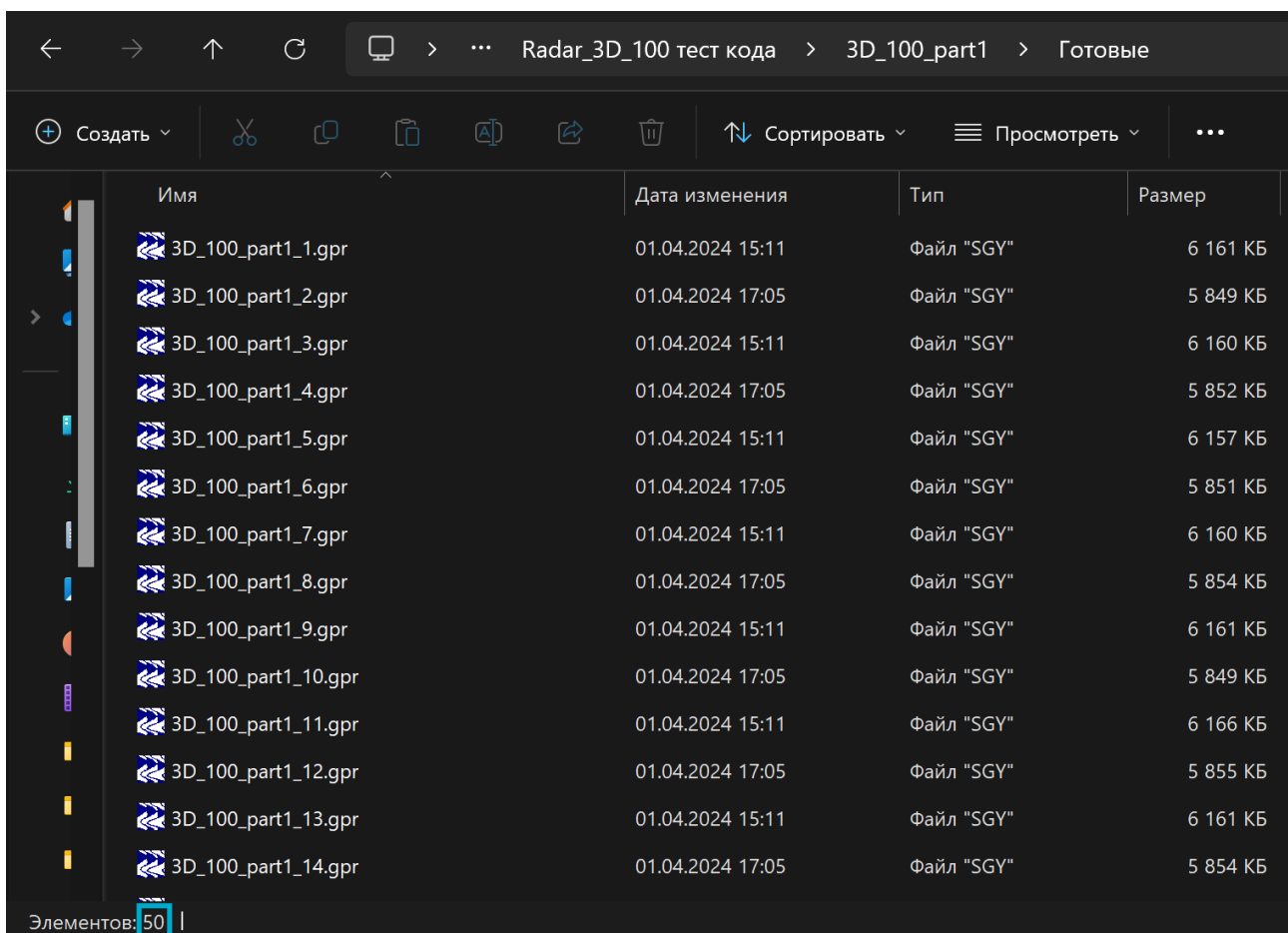


Рис. 1.1. Папка с файлами на вход

```
1 % File search depending on the number of files in the
   folder
2 for j = 1:50
3     % Generate input filename
4     filename = sprintf('3D_100_part1_%01d.gpr.sgy', j);
5     fullpath = fullfile(input_folder, filename); % Full
       path to file
```

```

6 % Generating the name to save
7 new_filename = sprintf('3D_100_part1_modified_%02d.
  sgy', j);
8 new_fullpath = fullfile(output_folder, new_filename
  );
9 copyfile(fullpath, new_fullpath); % copy the
  original file into the new one, so as not to
  overwrite the information in the original one

```

Представленный выше кусок кода перебирает файлы по номеру из входной папки и сохраняет копии с новым именем в новой папке.

У каждого файла своё количество трасс, его можно считать с помощью функции из стороннего пакета [1]. После установки, пакет необходимо подгрузить в MATLAB для дальнейшего использования содержащихся в нём функций (рис. 1.2)

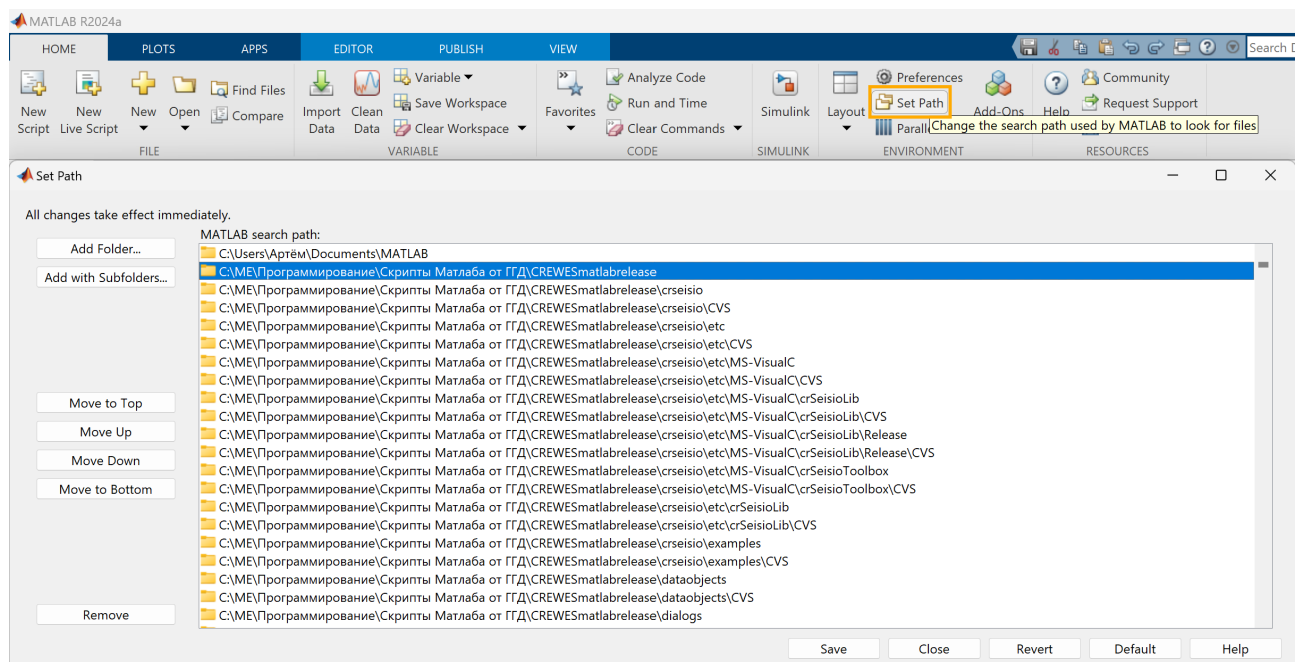


Рис. 1.2. Выбор папки для подгрузки внешних функций в MATLAB

```

1 SEG=altreadseggy(fullpath);
2 [tr_smpls, num_tr]=size(SEG); % read file size

```

После выполнения кода, в переменную `tr_smpls` запишется количество отсчётов по каждой трассе, а в переменную `num_tr` – количество трасс в файле. Например, для файла «3D_100_part1_1.gpr.sgy» `num_tr = 512`; `tr_smpls = 4988` (см. рис. 2.1).

2. Бинарный вид. Открываем SEG-Y

Прежде, чем записывать данные в заголовки трасс, необходимо открыть файлы в бинарном виде. Посмотреть, с каким типом кодировки и порядком байт имеем дело, можно в ПО SeiSee [2].

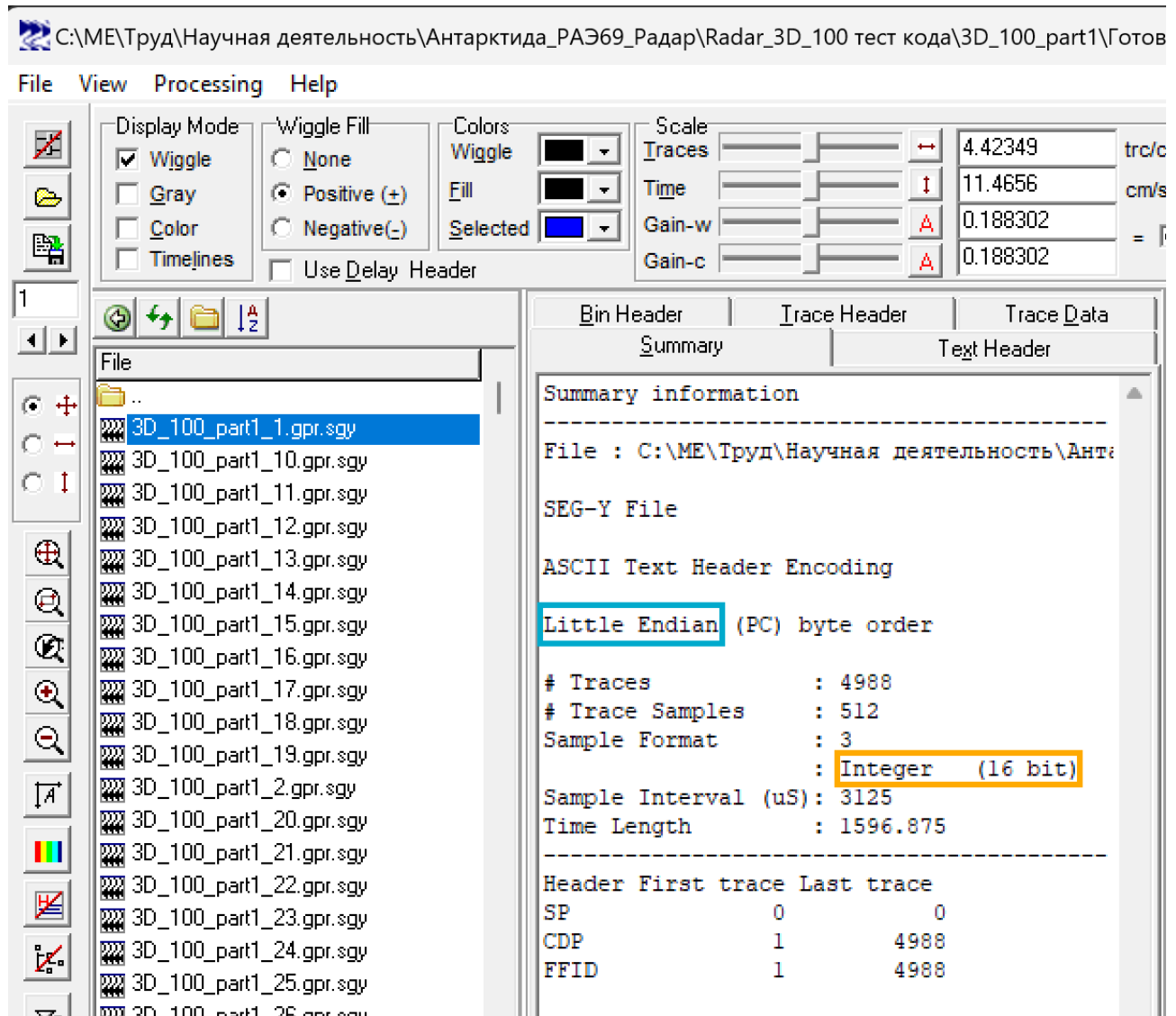


Рис. 2.1. Смотрим порядок байт (выделено синим) и тип кодировки (выделено оранжевым)

Поскольку формат записи 16-тибитовый, информация по каждому заголовку в каждой трассе хранится в ($\frac{16}{8} =$) **2-х** байтах. Порядок записи байт также важен, указываем его при открытии файла 'l' (little-endian), соответственно 'b' (big-endian). 'r+b' – чтение и запись в бинарном виде.

```
1 % Open the file in binary mode to read 'l' - little
  -endian
2 fid = fopen(new_fullpath, 'r+b', 'l');
3
4 if fid == -1
```

```

5 |         error('Unable to open file: %s', new_fullpath);
6 |     end

```

Следующий шаг – организовать перебор по всем трассам внутри каждого файла для записи данных геометрии в каждую трассу.

3. Расчёт геометрии

Каждая сейсмограмма представляет собой набор трасс, записанных с определённым шагом дискретизации. Зная параметр **num_tr** и длину профиля **100 м**, найдём **dX** как $\frac{100}{num_tr-1}$ (рис. 3.1).

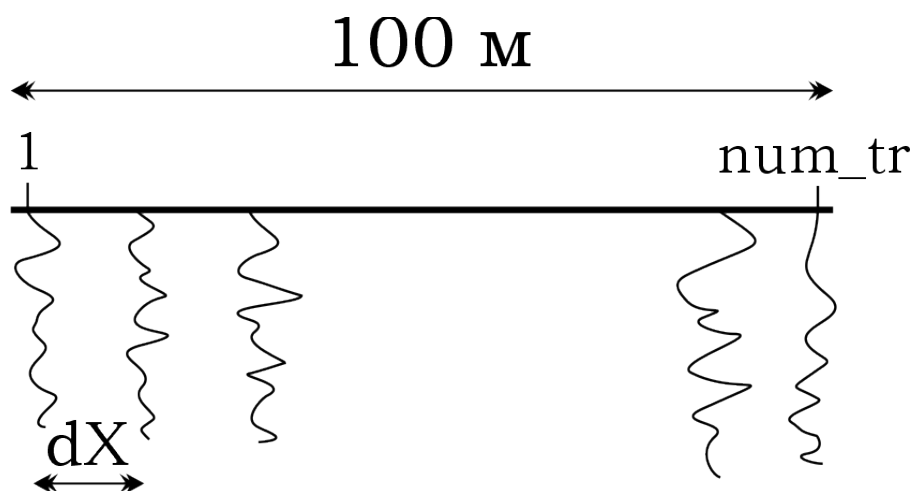


Рис. 3.1. Анализ шага дискретизации (он же – шаг по оси X)

Рассчитав **dX**, нашли шаг координат по оси X.

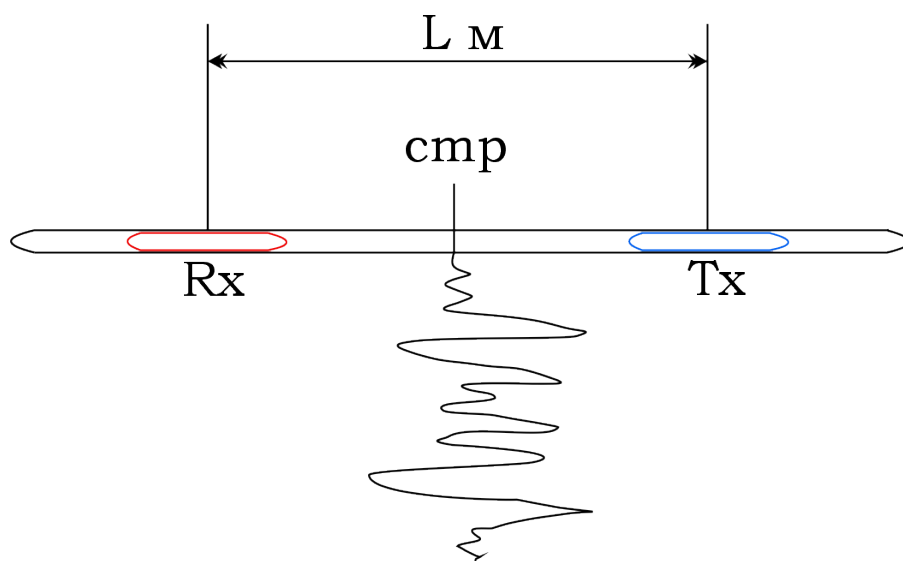


Рис. 3.2. Анализ расстояния источник-приёмник и положения срединной точки

Рассояние источник-приёмник L находим на основании параметров конкретного георадара (можно посмотреть в документации аппаратуры). За основу шага в заданной сетке полигона 0-100 по каждой из осей берём координаты смр, тогда координаты источника в каждой точке смр по оси X будут на $\frac{L}{2}$ м больше, а приёмника на $\frac{L}{2}$ м меньше. По оси Y координаты смр, источника (sou или T на рис. 3.2) и приёмника (rec или R 3.2) совпадают, поскольку радар волокни вдоль именно этой оси.

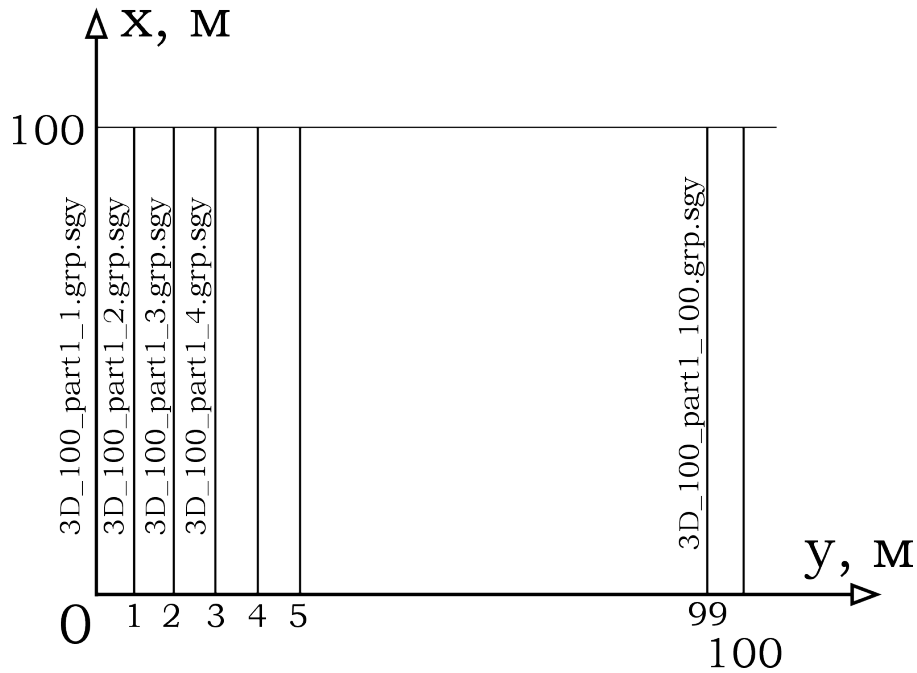


Рис. 3.3. Анализ общего плана и шага по оси Y

Координата по оси Y совпадает с названием файла, поскольку шаг между профилями 1 м.

```

1   for i = 1:num_tr
2       L=1; % distance between source and receiver
3       dx = 100/(num_tr-1);
4       cmp_x = (i-1) * dx + 1000;
5       sou_x = cmp_x + L./2;
6       rec_x = cmp_x - L./2;
7
8       cdp_y = (j-1) + 1000;
9       sou_y = cmp_y;
10      rec_y = cmp_y;

```

Всю описанную геометрию заносим в формульном виде в начале вложенного цикла с перебором по трассам, поскольку запись координат геометрии уникальная для каждой трассы. Дополнительно зададим начало координат на 1000 м больше, поскольку имеем значения в отрицательной области (координата смр_x начинается с 0-вой отметки, а координата rec_x с $-0.5 \left[\text{cmp}_x - \frac{L}{2} \right]$). Данная поправка может быть любой превышающей значение

самой большой координаты по модулю в отрицательной области. 1000 берём из соображений, что профиля по 100 м, а значит визуально после добавления 1000 картина изменится не значительно.

4. Насыщение заголовков геометрией

Для записи собственных значений координат в заголовки необходимо знать положение байт в бинарном виде файла SEG-Y (рис. 4.1).

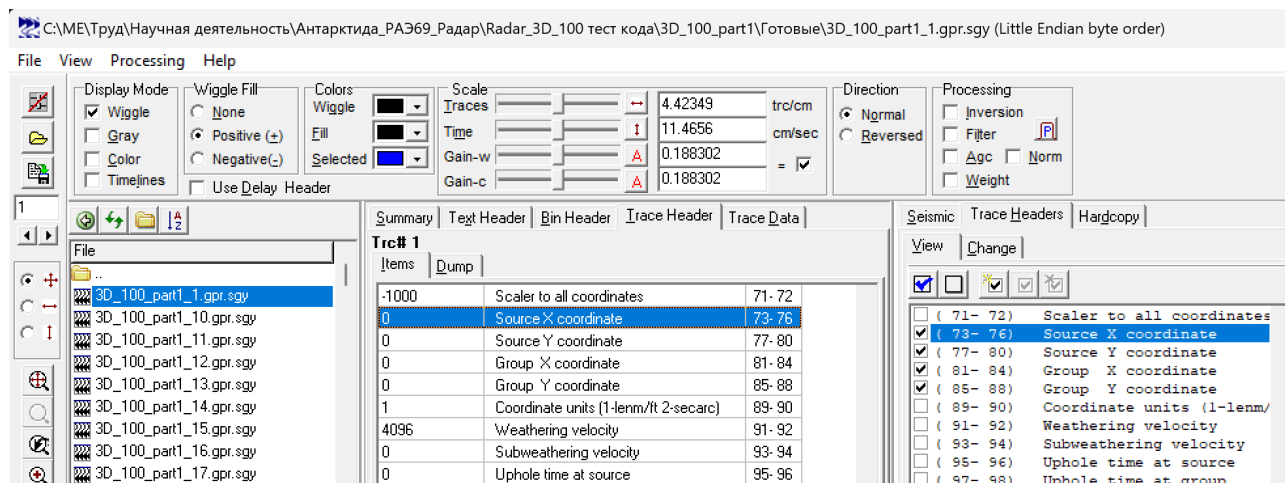


Рис. 4.1. Находим положение информации о заголовке `sou_x` в бинарном виде

Далее нам необходимо «прыгнуть» в указанные байты, чтобы записать в них информацию. Для этого пропустим 3200 байт, в которых содержится текстовая информация о файле, и 400 байт с описанием заголовков файла. Таким образом, байтовое положение для координаты `sou_x` для 1-ой трассы **3673**.

Теперь необходимо заполнить все остальные трассы в файле, для этого и используется цикл от 1 до `num_tr` (количество трасс в файле). Чтобы записать информацию в трассу 2, «прыгнем» сперва к её началу, то есть $3200 + 400 + 240 + 2 \cdot 512$, а затем ещё на +73, в которые хотим записать информацию. Полный алгоритм выглядит следующим образом:

$$\text{position_cmp_x} = 3200 + 400 + (i - 1) \cdot (240 + 2 \cdot \text{tr_smpls}) + 73 - 1, \quad (4.1)$$

где 3200 – количество байт на запись текстовой информации;

400 – количество байт на описание заголовков файла;

`i` – счётчик номера трассы (1 : `num_tr`);

240 – количество байт на описание заголовков трассы;

2 – рассчитанное в главе 2 количество байт на запись заголовка в трассе;

`tr_smpls` – количество отсчётов в каждой трассе (см. гл. 1 и рис. 2.1);

73 – номер хранения информации по необходимому заголовку;

–1 – «встаём» перед желаемым байтом, чтобы запись началась именно в него.

```
1      position_cmp_x = 3200 + 400 + (i-1)*(240 + 2*
2          tr_smpls) + 181 -1;
3      fseek(fid, position_cmp_x, 'bof'); % bof -
4          write from the beginning of the file
5      fwrite(fid, uint64(cmp_x.*1000), 'uint64');
6
7      position_cmp_y = position_cdp_x + 4;
8      fseek(fid, position_cmp_y, 'bof'); % bof -
9          write from the beginning of the file
10     fwrite(fid, uint64(cmp_y.*1000), 'uint64');
11
12     position_sou_x = 3200 + 400 + (i-1)*(240 + 2*
13         tr_smpls) + 73 -1;
14     fseek(fid, position_sou_x, 'bof'); % bof -
15         write from the beginning of the file
16     fwrite(fid, uint64(sou_x.*1000), 'uint64');
17
18     position_rec_x = 3200 + 400 + (i-1)*(240 + 2*
19         tr_smpls) + 81 -1;
20     fseek(fid, position_rec_x, 'bof'); % bof -
21         write from the beginning of the file
22     fwrite(fid, uint64(rec_x.*1000), 'uint64');
23
24     position_rec_y = position_rec_x + 4;
25     fseek(fid, position_rec_y, 'bof'); % bof -
26         write from the beginning of the file
27     fwrite(fid, uint64(rec_y.*1000), 'uint64');
28
29     end
```

'bof' – «прыжок» осуществляется с начала файла. 'uint64' – перекодирование числа в формат для записи в бинарном виде (64 используется, поскольку работаем с числами 6-ого порядка, для меньших порядков подойдёт 'uint32' и так далее). И каждую координату домножаем на 1000 для корректного отображения координат в SeiSee и SeisPro (поскольку там значения берутся в 1000 раз меньше).

5. Завершение работы

После записи информации во все заголовки каждого файла, необходимо закрыть циклы и сам файл.

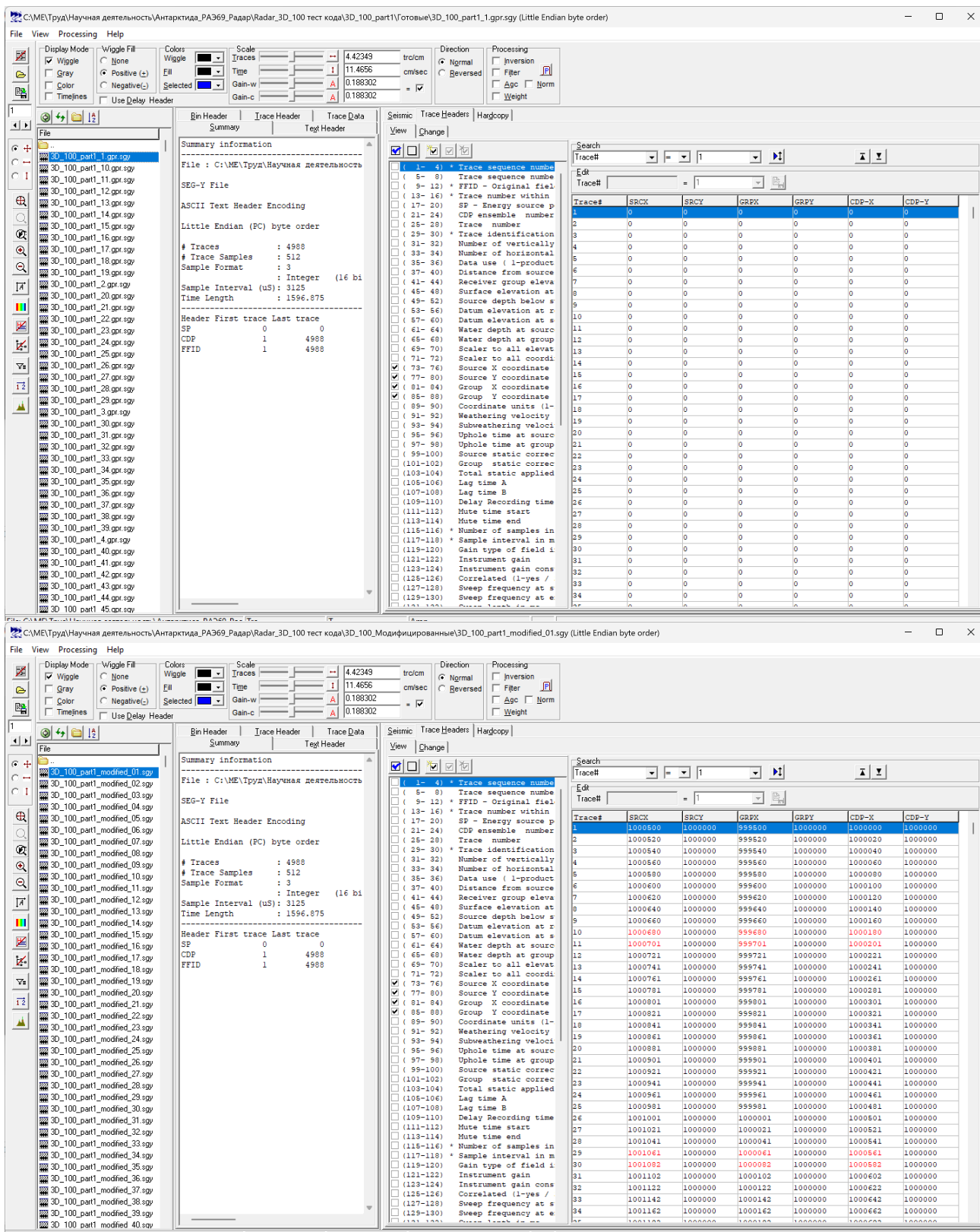


Рис. 5.1. Заголовки и файлы в папках до/после (сверху/снизу) работы кода

```

1     end
2
3     % Close the file
4     fclose(fid);
5 end
6
7 disp('Files were successfully modified and saved as new
    .');

```

Цельный код

```

1 clc; clear workspace;
2
3 % Specify the path to the folder with the source files.
4 input_folder = 'C: '; % Folder with source .sgy files
5 % Specify the path to the folder where the new files
   will be written.
6 output_folder = 'C: '; % Folder to save the modified
   files to
7
8
9 % File search depending on the number of files in the
   folder
10 for j = 1:50
11     % Generate input filename
12     filename = sprintf('3D_100_part1_%01d.gpr.sgy', j);
13     fullpath = fullfile(input_folder, filename); % Full
       path to file
14     % Generating the name to save
15     new_filename = sprintf('3D_100_part1_modified_%02d.
       sgy', j);
16     new_fullpath = fullfile(output_folder, new_filename
       );
17     copyfile(fullpath, new_fullpath); % copy the
       original file into the new one, so as not to
       overwrite the information in the original one
18
19     SEG=altreadsegy(fullpath);
20     [tr_smpls, num_tr]=size(SEG); % read file size
21

```

```

22 % Open the file in binary mode to read 'l' - little
    -endian
23 fid = fopen(new_fullpath, 'r+b', 'l');
24
25 if fid == -1
26     error('Unable to open file: %s', new_fullpath);
27 end
28
29 % Cycle of writing the necessary data to the
    headers
30 for i = 1:num_tr
31     L=1; % distance between source and receiver
32     dx = 100/(num_tr-1);
33     cdp_x = (i-1) * dx + 1000;
34     sou_x = cdp_x + L./2;
35     rec_x = cdp_x - L./2;
36
37     cdp_y = (j-1) + 1000;
38     sou_y = cdp_y;
39     rec_y = cdp_y;
40
41     position_cdp_x = 3200 + 400 + (i-1)*(240 + 2*
        tr_smpls) + 181 -1;
42     fseek(fid, position_cdp_x, 'bof'); % bof -
        write from the beginning of the file
43     fwrite(fid, uint64(cdp_x.*1000), 'uint64');
44
45     position_cdp_y = position_cdp_x + 4;
46     fseek(fid, position_cdp_y, 'bof'); % bof -
        write from the beginning of the file
47     fwrite(fid, uint64(cdp_y.*1000), 'uint64');
48
49     position_sou_x = 3200 + 400 + (i-1)*(240 + 2*
        tr_smpls) + 73 -1;
50     fseek(fid, position_sou_x, 'bof'); % bof -
        write from the beginning of the file
51     fwrite(fid, uint64(sou_x.*1000), 'uint64');
52
53     position_sou_y = position_sou_x + 4;
54     fseek(fid, position_sou_y, 'bof'); % bof -
        write from the beginning of the file
55     fwrite(fid, uint64(sou_y.*1000), 'uint64');
56

```

```

57         position_rec_x = 3200 + 400 + (i-1)*(240 + 2*
           tr_smpls) + 81 -1;
58         fseek(fid, position_rec_x, 'bof'); % bof -
           write from the beginning of the file
59         fwrite(fid, uint64(rec_x.*1000), 'uint64');
60
61         position_rec_y = position_rec_x + 4;
62         fseek(fid, position_rec_y, 'bof'); % bof -
           write from the beginning of the file
63         fwrite(fid, uint64(rec_y.*1000), 'uint64');
64     end
65
66     % Close the file
67     fclose(fid);
68 end
69
70 disp('Files were successfully modified and saved as new
    .');

```

Содержание

| | | |
|---|---------------------------------|----|
| 1 | Ввод данных | 2 |
| 2 | Бинарный вид. Открываем SEG-Y | 4 |
| 3 | Расчёт геометрии | 5 |
| 4 | Насыщение заголовков геометрией | 7 |
| 5 | Завершение работы | 9 |
| | Цельный код | 10 |
| | Список литературы | 14 |

Список литературы

- [1] [altreadsegy.m](#) — ссылка на GitHub с необходимой функцией.
- [2] [SeiSee 1](#), [SeiSee 2](#) — ссылки на установку.