AMPLE Mode ParaboLic Equation

Моделирование трёхмерных акустических полей в океане методом широкоугольных модовых параболических уравнений

25 апреля 2021 г.

1. Начало работы в Windows. Установка

1.1. Шаг 0

Чтобы собрать свою версию приложения, необходимо установить следующие программы:

- Microsoft Visual Studio
- CMake
- Git

При установке Visual Studio необходимо добавить английский языковой пакет и выбрать пункт 'Разработка классических приложений на C++'. Также потребуется добавить CMake и Git в переменную окружения PATH. Об этом вас попросит программа установки.

1.2. Шаг 0.5

Следующие шаги будут выполняться с использованием интерфейса командной строки PowerShell. Чтобы его открыть, нажмите `Win+R`, введите `powershell` и нажмите `enter`. Обратите внимание, что `PS>` обозначает только начало новой строки, и не является частью команды.

1.3. Шаг 1

Выберите рабочую папку из числа тех, к которым у Вас есть доступ (право редактирования), которая в дальнейшем будет обозначаться как `[root]`), и перейдите в неё:

```
PS> cd [root]
```

Теперь установите vcpkg – менеджер пакетов для Visual Studio. Следующие команды создадут папку `vcpkg` и скопируют в неё содержимое репозитория

```
PS> git clone "https://github.com/microsoft/vcpkg"
```

Теперь необходимо завершить установку vcpkg следующими командами. Для них могут потребоваться права администратора

```
PS> cd vcpkg
PS> .\bootstrap-vcpkg.bat
PS> .\vcpkg integrate install
```

Теперь необходимо добавить нужные библиотеки. Это займет примерно 20 минут

```
PS> .\vcpkg install --triplet x64-windows `
    boost-program-options fftw3 nlohmann-json
```

1.4. Шаг 2

Перейдите в корневую папку и скопируйте содержимое репозитория Ample

```
PS> cd ..\
PS> git clone "https://github.com/GoldFeniks/Ample"
```

Перейдите в только что созданную папку и инициализируйте дополнительные модули

```
PS> cd Ample
PS> git submodule update --init --recursive
```

1.5. Шаг 3

Теперь необходимо подготовить решение Visual Studio. Перейдите в `build/cmake` и создайте директорию

```
PS> cd build\cmake
PS> mkdir AMPLE
PS> cd AMPLE
```

Создайте файл решения (исходные данные для сборки программы) при помощи CMake

```
PS> cmake `
    -DCMAKE_TOOLCHAIN_FILE=..\..\..\..\vcpkg\scripts\buildsystems\vcpkg.cmake `
    -DUSE_VCPKG=true -DCMAKE_BUILD_TYPE=Release ..
```

1.6. Шаг 4

Наконец, для того, чтобы собрать Ample, перейдите в папку `[root]/Ample/build/cmake/AMPLE` в Проводнике и откройте `AMPLE.sln` при помощи Visual Studio. В верхней панели инструментов замените тип сборки с `Debug` на `Release` в выпадающем списке. Теперь выберите `Build -> Build Solution` в меню и подождите окончания компиляции. В случае её успешного завершения появится файл `AMPLE.exe` в `[root]/Ample/build/cmake/AMPLE/Release`. Обратите внимание, что все три файла, перечисленные ниже, необходимы для вычислений.

```
AMPLE.exe
boost_program_options-vc*-mt-x*-*.dll
fftw3.dll
```

Символы * в названии второго файла могут обозначать любое число - это зависит от актуальной версии программ и разрядности Вашей ОС.

1.7. Обновление

Программа готова. Для её обновления потребуется перейти в папку с программой и ввести следующую команду

```
cd [root]
git pull --ff-only
```

Последняя команда скачивает изменения с репозитория. Для их применения надо повторить полностью шаг 4. Обновление успешно завершено, и новая версия Ample.exe находится в папке `[root]/Ample/build/cmake/AMPLE/Release`.

2. Вычисления

2.1. Запуск программы

Запуск программы производится через интерфейс командной строки. Для удобства работы рекомендуется создать отдельную папку `[root]` и поместить туда все три файла из 4-го шага установки. Для запуска необходимо открыть 'powershell' и перейти в папку

```
cd [root]
.\Ample.exe task option1 option2...
```

2.1.1. Конфигурирование

Для запуска программы необходимы дополнительные аргументы, вводимые после неё. Первым аргументом идет `task` - собственно, объект расчёта.

- `solution` - вычисление решения широкоугольного модового параболического уравнения (по умолчанию)
- `modes` - вычисление модовых функций и волновых чисел
- `rays` - расчёт акустических лучей
- `impulse` - расчёт распространения импульсного сигнала
- `init` - вычисление начальных условий

Второй и последующие аргументы - параметры `options`. При необходимости они задаются флажком и значением (например, `-v 3`).

- `-h` - вывод информационного сообщения (без значения)
- `-v` - вывод состояния выполнения расчётов.
  0 - нет вывода
  1 - время выполнения
  2 - время и шкала выполнения
  3 - время, шкала выполнения и конфигурация расчёта
  По умолчанию -v 0
- `-r` - выводит в консоль каждую n-ю строку, значение - номер строки n. Работает, если $-v > 0$ и при задании solution.
  По умолчанию -r 0
- `-c` - путь к конфигурационному файлу config.json.
  По умолчанию он ищется в той же папке
- `-o` - путь к папке, в которую необходимо сохранить результаты расчетов.
  По умолчанию сохранение идёт в папку output
- `-s` - вывод в файл каждой k-й строки.
  По умолчанию -s 100
- `--binary` - сохранение результата в бинарном виде.
  По умолчанию они сохраняются в формате .txt

- `-w` - количество потоков для расчёта. Работает, если задание - solution или impulse.
  По умолчанию -w 1
- `-b` - размер буфера для многопоточных вычислений.
  По умолчанию -b 100

2.2. Конфигурационный файл

Конфигурационный файл задает параметры для вычислений и сохраняется в формате .json. Переменные в нём задаются следующим образом

```
 "var1": value_1,
 "var2": value_2,
 ...
 "varn": value_n,
 "input_data":[
   {"type": "variable1",
   "dimensions": [n1],
   "values": [value_n1]
   },
   {"type": "variable2",
   "dimensions": [n2],
   "values": [value_n2]
   },
   ...,
   {"type": "variable_m",
   "dimensions": [n_m],
   "values": [value_n_m]
   }
 ]
}
```

Первая группа переменных обозначается одним числом, логической переменной или массивом:

- ”n_modes”: количество используемых мод
- ”ppm”: 10 - количество точек, используемых в расчёте моды
- ”mnx”:10 - количество точек по оси X, в которых рассчитываются моды
- ”mny”:5 - количество точек по оси Y, в которых рассчитываются моды

- ”ordRich”: 3 - порядок аппроксимации по Ричардсону
- ”n_layers”:600 - количество водных слоев
- ”complex_modes”: использование комплексных значений мод[0 или 1]
- ”y0”: -2000 - начальная граница волновода по оси Y
- ”y1”: 2000 - конечная граница волновода по оси Y
- ”ny”: 2001 - количество точек по оси Y
- ”x0”: 50 - начальная граница волновода по оси X
- ”x1”: 7500 - конечная граница волновода по оси X
- ”nx”: 7501 - количество точек по оси X
- ”z0”: 0 - начальная граница волновода по оси Z
- ”z1”: 25 - конечная граница волновода по оси Z
- ”nz”: 26 - количество точек по оси Z
- ”z_s”: 4 - глубина источника
- ”betas”: [0, 0.3] - затухание в разных слоях
- ”bottom_rhos”: [1.8] - плотность донного слоя,
- ”bottom_c1s”: [1750] - скорость звука на верхней границе дна
- ”bottom_c2s”: [1750]скорость звука на нижней границе дна
- ”bottom_layers”:[20] - количество слоёв в дне,
- ”init”: - начальные условия, [greene], [gauss], [ray_simple], при выборе последнего необходимо указать минимум 2 следующих параметра,
- ”a0”: -0.3 - апертура источника в отрицательном направлении оси Y,
- ”a1”: 0.3 - апертура источника в положительном направлении оси Y
- ”na”: 90 - сетка для апертуры
- ”sel_range”: [20, 220] - частотный диапазон для расчёта SEL
- ”sel_strict”: - запрет на расчёт, если значение переменной frequency (ниже) лежит вне sel_range [true или false]
- ”reference_index”: 1,

- ”const_modes”: - однородность волновода вдоль оси X [true или false]

Вторая группа переменных обозначается следующим образом:

```
{"type": "variable",
   "dimensions": [n],
   "values": [value_n]
}
```

Первая строчка обозначает название переменной, вторая - её размерность, третья содержит в себе значение, массив либо ссылку на файл со значениями. Для простоты рассмотрим это на примерах. Частоты 150 и 250 Гц:

```
 {
            "type": "frequencies",
            "dimensions": [ 1 ],
            "values": [ 150, 250 ]
        },
```

Координаты трех приёмников в формате [X, Y, Z]

```
   {
            "type": "receivers",
            "dimensions": [ 3 ],
            "values": [
                [0, 0, 4],
                [602, 174, 18.8],
                [6154, 17, 8.7]
            ]
        },
```

Пространственные координаты и количество точек разбиения батиметрии отдельно по осям X и Y, которая хранится в файле bath.txt:

```
    {
            "type": "bathymetry",
            "dimensions": [
                {
                    "n": 151,
                    "bounds": {
                        "a": 0,
                        "b": 7500,
                        "d": 50
                    }
```

```
            },
            {
                "n": 81,
                "bounds": {
                    "a": -2000,
                    "b":  2000,
                    "d": 50
                }
            }
        ],
        "values": "bath.txt"
    }
```

Гидрология, постоянная по оси Z. Две пары значений необходимы для интерполяции по осям X и Y

```
        "type": "hydrology",
        "dimensions": [ 2, 2 ],
        "values": [
            [1500, 1500],
            [1500, 1500]
        ]
```

Существует другой способ задания гидрологии по оси Y от 0 до 21 м, по оси X - для всеё расчётной области, значения хранятся в файле hydr_2.txt.

```
    "type": "hydrology",
        "dimensions": [
            {
                "n": 22,
                "bounds": {
                    "a": 0,
                    "b": 21
                }// "values": [0, 1, 2, 5, 5.01, 7, 7.01, 8, 8.01, 9, 9.01
            },
            {
                "n": 2,
                "values": [0, 1]
            }
        ],
        "values": "hydr_2.txt"
```

```
        },
```

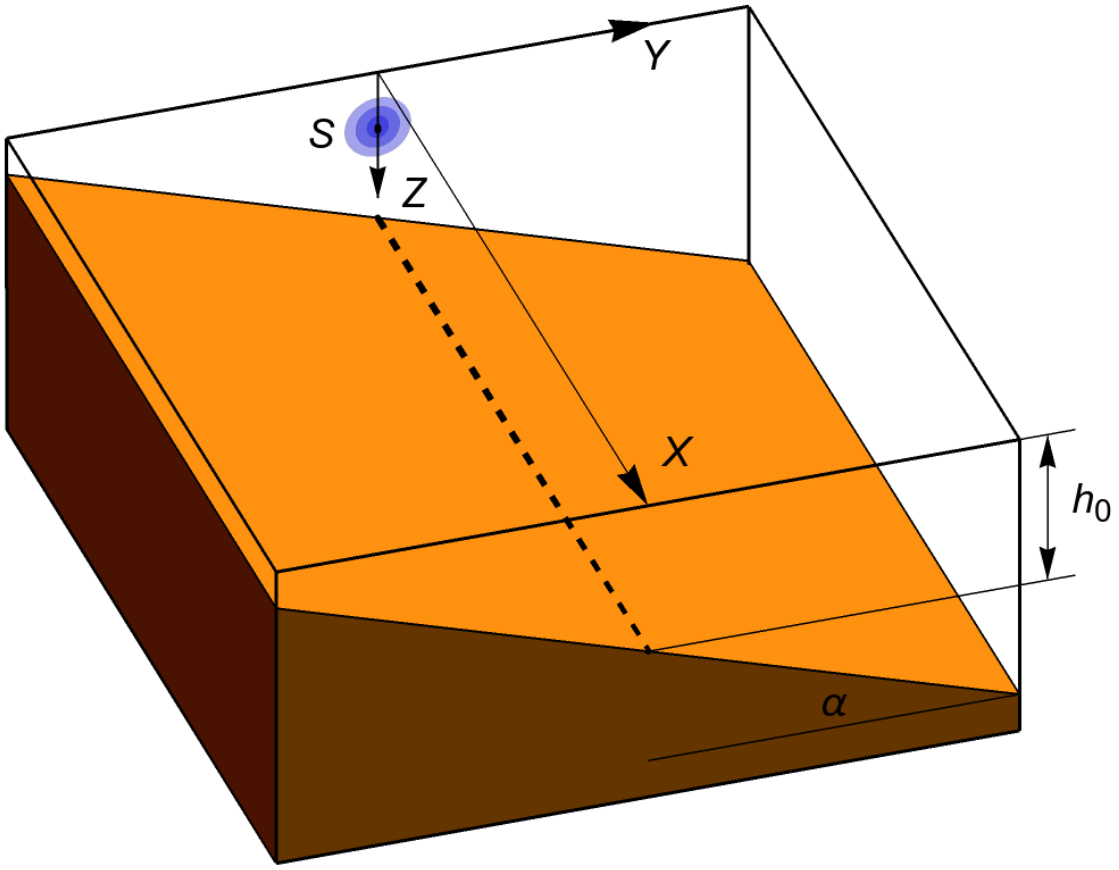
Функция источника (при расчётё SEL и распространения импульса), которая хранится в файле G1.txt

```
    {
            "type": "source_function",
            "dimensions": [
                {
                    "n": 30239,
                    "bounds": {
                        "a": 0,
                        "b": 1
                    }
                }
            ],
            "values": "G1.txt"
        }
```

2.3. Запуск на примерах

2.3.1. Клин

Батиметрия:



Конфигурационный файл:

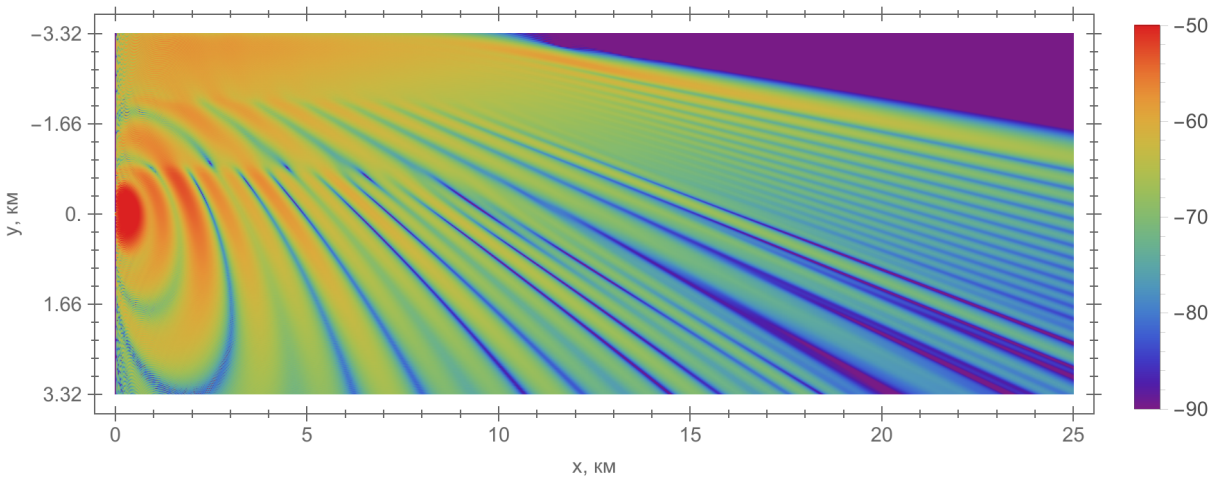
```
    {
    "ppm": 10,
    "y0": -3320,
    "y1":  3320,
    "ny":  6641,
    "x0":  50,
    "x1":  25000,
    "nx":  25001,
    "z0": 30,
    "z1": 30,
    "nz": 1,
    "mny": 1661,
    "z_s": 100,
    "init": "greene",
    "input_data": [
        {
            "type": "frequencies",
            "dimensions": [ 1 ],
            "values": [ 25 ]
```

```
        },
        {
            "type": "bathymetry",
            "dimensions": [
                2,
                {
                    "n": 2,
                    "bounds": {
                        "a": -4000,
                        "b":  4000,
                        "d":  8000
                    }
                }
            ],
            "values": [
                [0, 400],
                [0, 400]
            ]
        },
        {
            "type": "hydrology",
            "dimensions": [ 2, 2 ],
            "values": [
                [1500, 1500],
                [1500, 1500]
            ]
        }
    ]
}
```

Вызов программы для расчёта акустического поля с выводом только времени выполнения и с сохранением результата в папку output1.

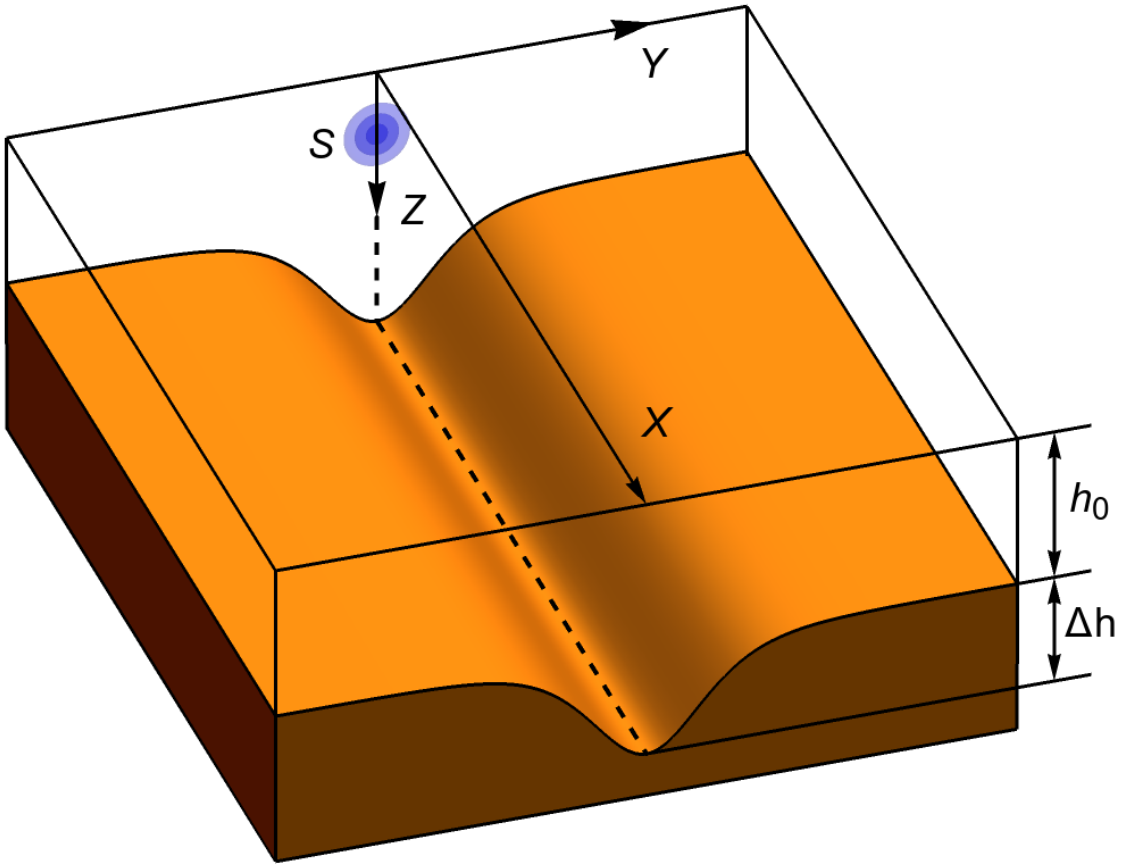
```
    cd [root]
    .\Ample.exe solution -v 1 -o output1
```

Акустическое поле:


−3.32
−1.66
0.
1.66
3.32
y, км
0
5
10
15
20
25
x, км
−50
−60
−70
−80
−90

2.3.2. Каньон

Батиметрия:



Конфигурационный файл:

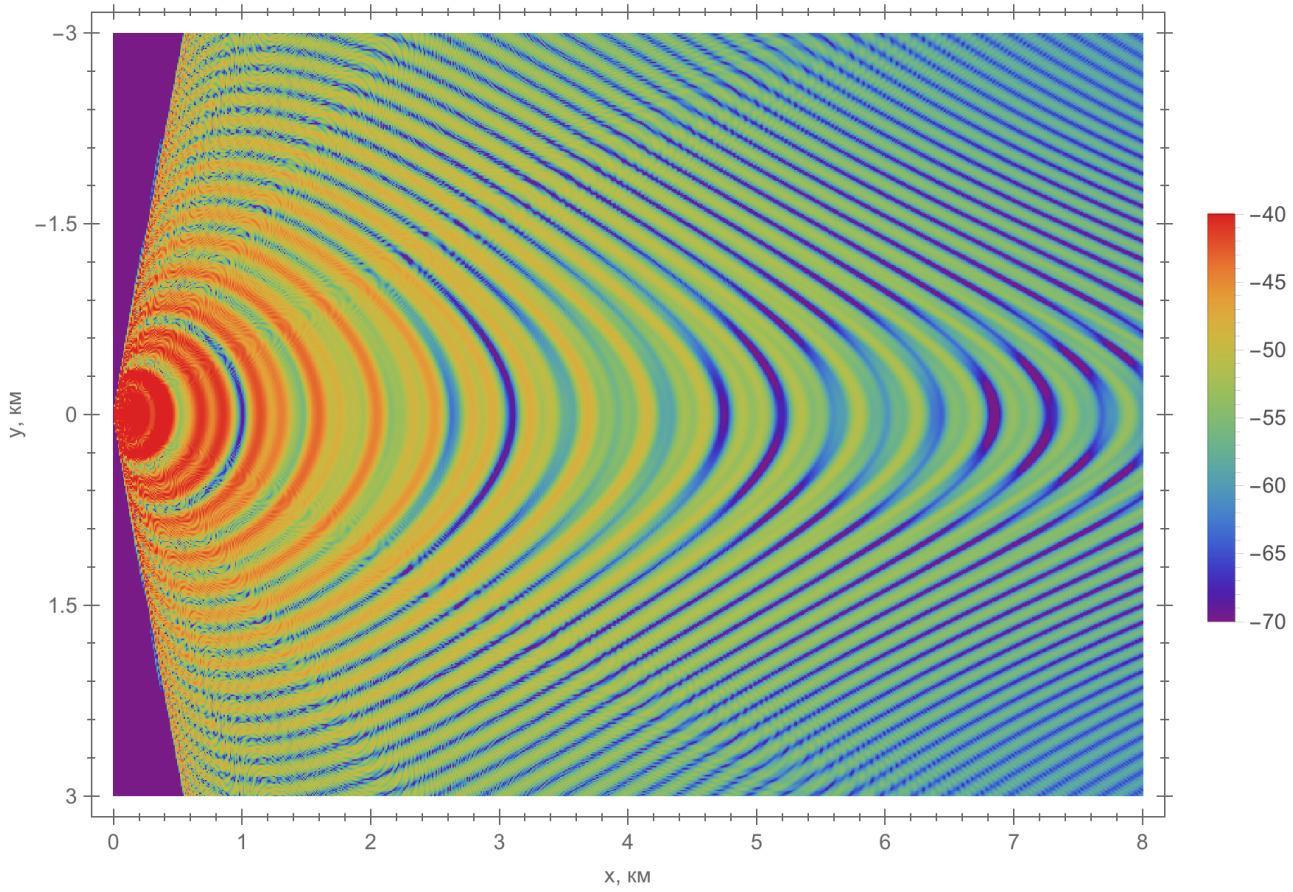
```
    {
    "ppm": 10,
    "y0": -3000,
    "y1": 3000,
    "ny": 6001,
    "x0": 50,
    "x1": 8000,
    "nx": 8001,
    "z0": 10,
    "z1": 10,
    "nz": 1,
    "z_s": 10,
    "bottom_rho": [2],
    "bottom_c1s": [1800],
    "bottom_c2s": [1800],
    "init": "greene",
```

```
    "input_data": [
        {
            "type": "frequencies",
            "dimensions": [ 1 ],
            "values": [ 150 ]
        },
        {
            "type": "bathymetry",
            "dimensions": [
                2,
                {
                    "n": 6001,
                    "bounds": {
                        "a": -3000,
                        "b":  3000,
                        "d":  1
                    }
                }
            ],
            "values": "bathymetry.txt"
        },
        {
            "type": "hydrology",
            "dimensions": [ 2, 2 ],
            "values": [
                [1500, 1500],
                [1500, 1500]
            ]
        }
    ]
}
```

Вызов программы для расчёта акустического поля с выводом времени, полоски выполнения и записью каждой 25-й строки в файл.

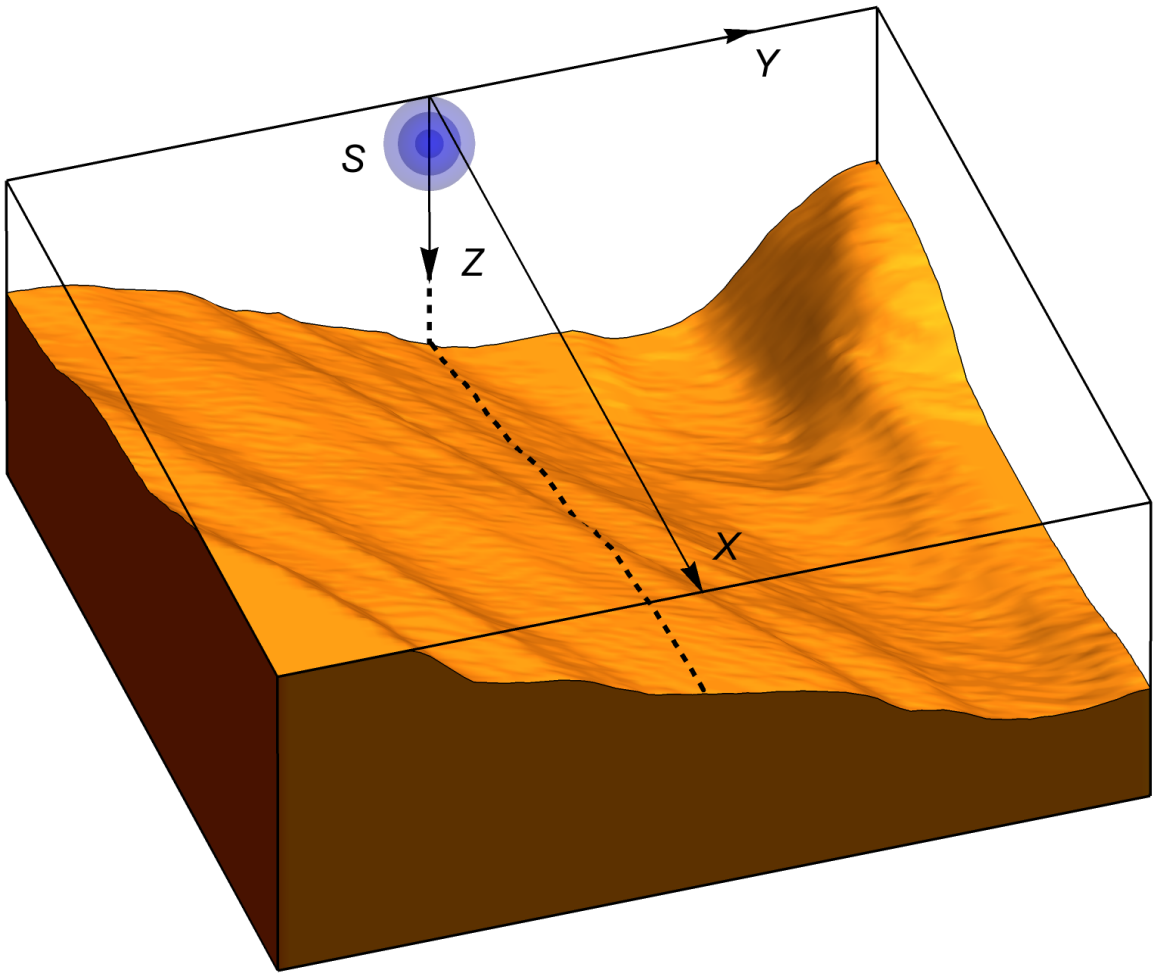
```
    cd [root]
    .\Ample.exe solution -v 2 -s 25
```

Акустическое поле:


−3
−1.5
0
1.5
3
y, км
0
1
2
3
4
5
6
7
8
x, км
−40
−45
−50
−55
−60
−65
−70

2.3.3. Реальный пример

Батиметрия:



Конфигурационный файл:

```
    {
    "ppm": 10,
    "y0": -2000,
    "y1": 2000,
    "ny": 2001,
    "x0": 50,
    "x1": 7500,
    "nx": 7501,
    "z0": 0,
    "z1": 25,
    "nz": 26,
    "z_s": 4,
    "n_layers": 5,
```

```
    "betas": [0, 0, 0, 0, 0, 0.3],
    "bottom_rhos": [1.8],
    "bottom_c1s": [1750],
    "bottom_c2s": [1750],
    "bottom_layers": [600],
    "init": "greene",
    "a0": -0.3,
    "a1":  0.3,
    "sel_range": [20, 220],
    "sel_strict": false,
    "reference_index": 1,
    "const_modes": false,
    "input_data": [
       {
            "type": "frequencies",
            "dimensions": [ 1 ],
            "values": [ 150 ]
        },

         {
            "type": "hydrology",
            "dimensions": [
                {
                    "n": 22,
                    "bounds": {
                        "a": 0,
                        "b": 21
                    }// "values": [0, 1, 2, 5, 5.01, 7, 7.01, 8, 8.01, 9, 9.01
                },
                {
                    "n": 2,
                    "values": [0, 1]
                }
            ],
            "values": "hydr_2.txt"
        },
        {
            "type": "receivers",
            "dimensions": [ 3 ],
            "values": [
```

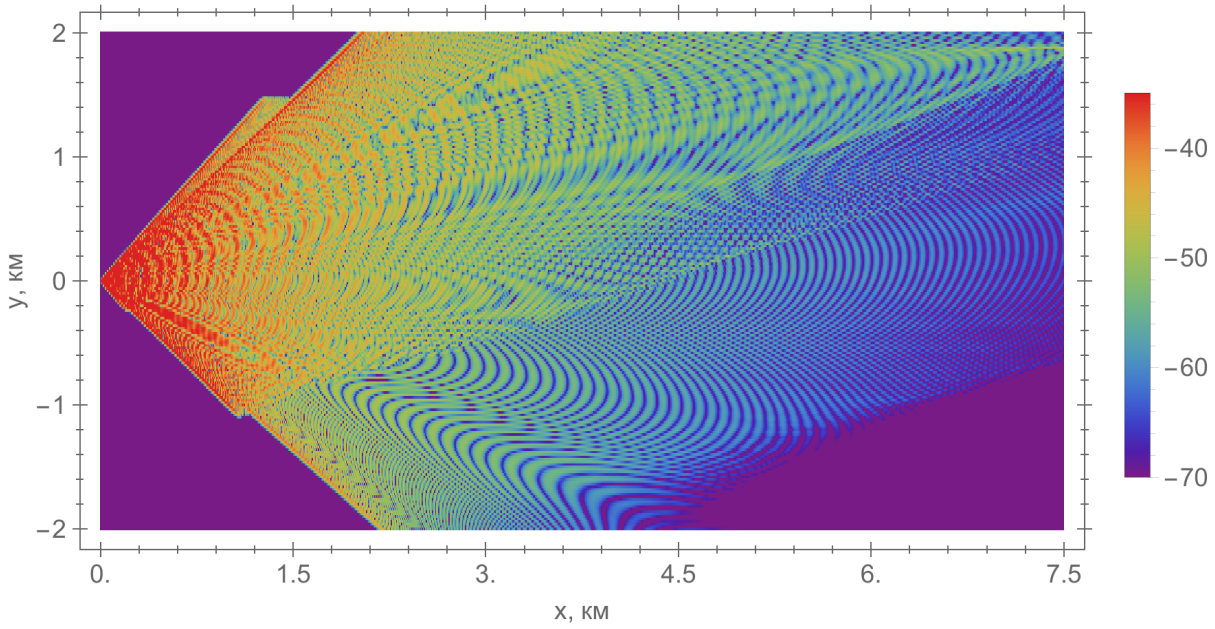
```
            [0, 0, 4],
            [602, 174, 18.8],
            [6154, -17, 8.7]
        ]
    },
    {
        "type": "bathymetry",
        "dimensions": [
            {
                "n": 151,
                "bounds": {
                    "a": 0,
                    "b": 7500,
                    "d": 50
                }
            },
            {
                "n": 81,
                "bounds": {
                    "a": -2000,
                    "b":  2000,
                    "d": 50
                }
            }
        ],
        "values": "bath.txt"
    },
    {
        "type": "source_function",
        "dimensions": [
            {
                "n": 30239,
                "bounds": {
                    "a": 0,
                    "b": 1
                }
            }
        ],
        "values": "G1.txt"
    }
```

```
    ]
}
```

Запуск программы для расчёта акустического поля, модовых функций с выводом всей конфигурационной информации в консоль, отображением времени, полоски заполнения, записью каждой 10-й строки в бинарный файл

```
    cd [root]
    .\Ample.exe solution -v 3 -s 10 --binary
```

Акустическое поле:



3. Интерпретация результатов расчёта в MatLab

Результаты расчётов по умолчанию сохраняются в одной из дочерних папок в папке output (например, solution или impulse), название файла - частота расчёта (например, 150.txt или 250.bin). Визуализировать их можно при помощи предложенных ниже скриптов в системе компьютерной математики MatLab версии не ниже R2017b. Файл скрипта должен лежать в той же папке, что и результат расчёта. Там же должен содержаться файл meta.json, содержащий информацию о расчётах.

3.1. Акустическое поле

Для отрисовки акустического поля нам понадобится запустить скрипт ReadDrawField с параметрами, перечисленными через запятую, через командную строку в MatLab

```
    ReadDrawField(filename, depth)
```

Параметры:

- filename - название файла, в котором сохранены результаты расчётов
- depth - требуемая глубина в метрах, для которой рассчитано акустическое поле. Если поле рассчитано для одного горизонта, то можно указать любое число

Рассмотрим работу скрипта подробнее.

```
close all
clc

set(0, 'DefaultAxesFontSize', 14, 'DefaultAxesFontName', 'Arial');
set(0, 'DefaultTextFontSize', 16, 'DefaultTextFontName', 'Arial');
```

Тут происходит очистка командной строки, закрытие сторонних окон MatLab и настройка шрифтов.

```
c = jsondecode(fileread('meta.json'));
c1 = c.dimensions;
c2 = c1{2};
x1 = c2.bounds.a;
x2 = c2.bounds.b;
nx = c2.n;

c3 = c1{3};
y1 = c3.bounds.a;
y2 = c3.bounds.b;
ny = c3.n;

c4 = c1{4};
z1 = c4.bounds.a;
z2 = c4.bounds.b;
nz = c4.n;
```

Данные команды считывают границы расчётной области и сетку, по которой она разбивается, напрямую из файла config.json. Именно с этим связано ограничение на версию MatLab.

```
if (c.binary==1)
y = fopen(filename);
a = fread(y, 'double');
else
```

```
a = dlmread(filename);
end
```

Тут содержится условный оператор, выбирающий, как открывать файл - как бинарный, либо как текстовый.

```
b = a(:,1:2:end) + 1i*a(:,2:2:end);
```

Приведение поля в комплекснозначный вид, такой шаг требуется из-за особенностей записи в файл.

```
if ((z1 - z2) == 0)
 field = b;
else
  field = b(:, depth + 1);
end
```

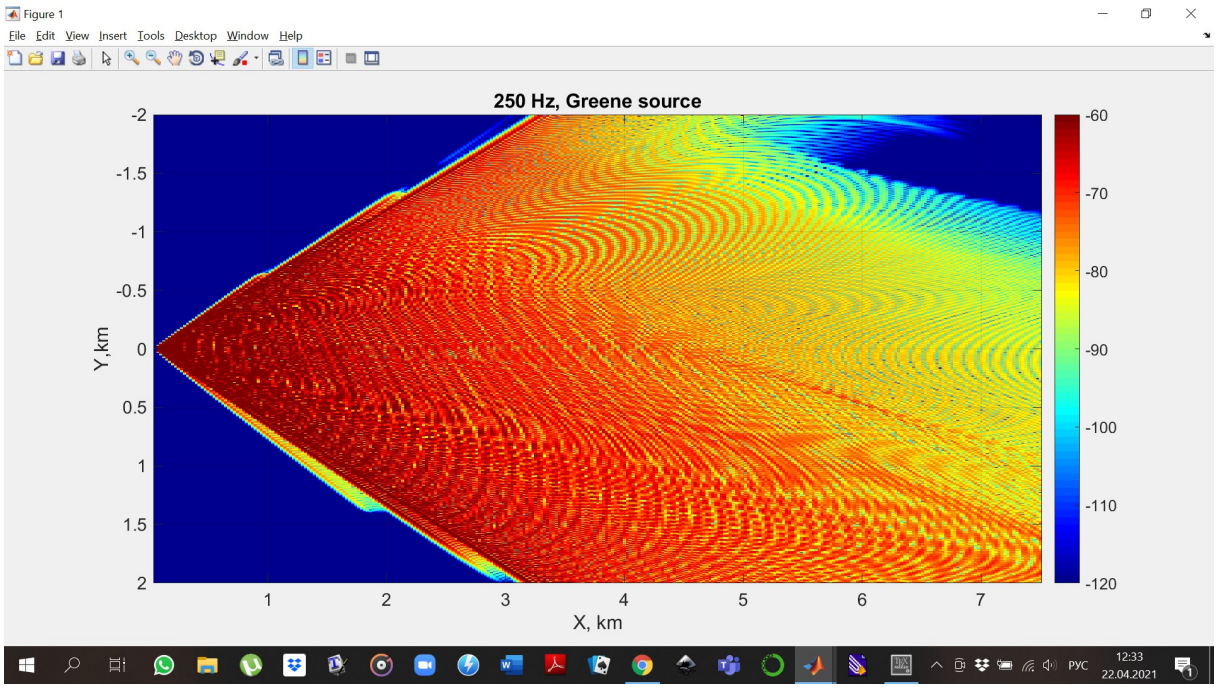
При расчёте поля более, чем на одном горизонте, тут выбирается нужная матрица

```
field1 = reshape(field, ny, []);
new_x = linspace(x1, x2, size(field1,1));
new_y = linspace(y1, y2, size(field1,2));
```

Формирование сетки для отрисовки поля.

```
figure;
imagesc(new_x/1000,new_y/1000,20*log10(abs(field1)));
hold on;
grid on;
xlabel('X, km');
ylabel('Y,km');
colorbar;
colormap(jet);
caxis([-120 -60]);
```

Отрисовка поля $TL = 20log_{10}U$. Через некоторое время MatLab выведет акустическое поле.



3.2. SEL

Для отрисовки акустического поля нам понадобится запустить скрипт ReadDrawSEL с аналогичными параметрами

```
ReadDrawSEL(filename, depth)
```

Через некоторое время MatLab выведет результаты расчётов.