

# Planar3D-LL [ФЦП 1.2]

Программа расчета геометрии трещины ГРП с использованием функции Грина для слоистой среды и универсальных асимптотик для отслеживания фронта трещины ГРП для плоской трехмерной трещины ГРП с учетом контрастов упругих модулей и напряжений.

## Сборка программы

После успешной сборки используйте команду `./planar3D --version`, чтобы вывести информацию об используемой версии, компиляторе, целевой платформе и дате сборки программы.

### Windows

Для автоматической сборки запустите командную строку `Visual Studio` и вызовите команду `nmake` в корневой папке репозитория.

Доступны следующие команды:

- `nmake` или `nmake all` – собрать программу, скопировать папки и файлы, необходимые для работы собранной программы.
- `nmake openmp` – собрать программу с поддержкой OpenMP, скопировать папки и файлы, необходимые для работы собранной программы.
- `nmake clear` – удалить все генерируемые файлы и папки;
- `nmake create_dirs` – создать сборочную папку, в неё скопировать папки и файлы, необходимые для работы собранной программы;
- `nmake planar3D` – пересобрать программу (сборочная папка должна быть создана).
- `nmake planar3D.openmp` – пересобрать программу с поддержкой OpenMP (сборочная папка должна быть создана).

### Unix

Для автоматической сборки вызовите команду `make` в корневой папке репозитория.

Доступны следующие команды:

- `make` или `make all` – собрать программу, скопировать папки и файлы, необходимые для работы собранной программы.
- `make openmp` – собрать программу с поддержкой OpenMP, скопировать папки и файлы, необходимые для работы собранной программы.
- `make clear` – удалить все генерируемые файлы и папки;
- `make create_dirs` – создать сборочную папку, в неё скопировать папки и файлы, необходимые для работы собранной программы;
- `make planar3D` – пересобрать программу (сборочная папка должна быть создана).
- `make planar3D.openmp` – пересобрать программу с поддержкой OpenMP (сборочная папка должна быть создана).
- `make depend` – пересобрать программу с поддержкой OpenMP (сборочная папка должна быть создана).

### Ручная сборка

Для успешной сборки требуется компилятор `C++` с поддержкой стандарта 2011 года (проверено для `g++`, `clang++`, `msc`). Рекомендуется использовать оптимизацию третьего уровня.

Для запуска программы потребуется создать папки `InitialConditions` (в неё скопировать текстовые файлы из `sources/data`) и `Results`, а также подпапки `Results/Concentration`, `Results/Opening`, `Results/Pressure`.

Пример правильной структуры:



Пример команды компиляции:

```
clang++ -std=c++11 -march=native -O3 -Weffc++ -Wall -pedantic -o ./planar3D ./sources/*.cc
```

## Добавление программы в состав модулей FractureGUI

Для имплементации расчётного модуля скопируйте собранный экземпляр программы (только исполнительный файл) в папку `Modules` . Подключение произойдёт автоматически при следующем запуске графического интерфейса.

## Входные данные

Данная версия использует значения модуля Юнга, коэффициентов Пуассона и Картера только из центрального слоя.

### Перед первым запуском

Убедитесь, что созданы все необходимые файлы и папки (см. п. [Сборка программы -> Ручная сборка](#))

### Список входных параметров

Для вызова полного списка входных параметров используйте команду `./planar3D --help` .

### Формат пользовательских входных файлов

Данные о слоях задаются текстовым файлом в виде матрицы Nx6, где N – число слоёв. В качестве разделителя используется символ табуляции.

Указываемые параметры:

- *Start* – координата верхней границы слоя в метрах;
- *Finish* – координата нижней границы слоя в метрах;
- *Stress* – величина пластовых напряжений в МПа;
- *Young's modulus* – модуль Юнга в ГПа;
- *Poisson's ratio* – коэффициент Пуассона;
- *Carter's coefficient* – коэффициент утечек Картера в мкм/с^0.5.

В качестве нулевой координаты выбирается высота точки закачки (положение точечного источника).

Для правильной работы обязательно наличие слоя, границы которого имеют координаты разных знаков.

Пример содержания стандартного файла слоёв `InitialConditions/layers.txt` :

```
#####
#                                     #
#   Start       Finish       Stress   Young's   Poisson's   Carter's   #
```

```
# modulus ratio coefficient #
# [m] [m] [MPa] [GPa] [n/d] [um/s^0.5] #
# -----
#
# 25.0 1000.0 32.1 26.0 0.23 6.0
# 15.0 25.0 32.0 23.0 0.25 4.0
# -15.0 15.0 30.0 25.0 0.25 2.0
# -25.0 -15 32.2 27.0 0.24 3.0
# -1000.0 -25 32.4 28.0 0.25 2.0
#
# =====
#
#####
```

План закачки задаётся текстовым файлом в виде матрицы Nx4, где N – число стадий закачки. В качестве разделителя используется символ табуляции.

Указываемые параметры:

- *Start* – время начала стадии в минутах;
- *Finish* – время окончания стадии в минутах;
- *Fluid injection* – скорость закачки жидкости в м<sup>3</sup>/мин;
- *Bulk proppant* – плотность пропанта (удельная масса в жидкости).

Для правильной работы обязательно, чтобы первая стадия начиналась с 0 минут и имела положительную скорость закачки.

Пример содержания стандартного файла слоёв InitialConditions/injection.txt :

```
#####
#
# Start Finish Fluid Bulk
# injection proppant
# [min] [min] [m^3/min] [kg/m^3]
# -----
#
# 0.0 40.0 3.6 0
# 40.0 60.0 3.2 200
#
# =====
#
#####
```

## Выходные данные

### Список начальных параметров

Начальные параметры сохраняются в формате JSON в текстовый файл Results/parameters.json и могут быть затем считаны сторонними программами, либо импортированы в модуль Planar3D для воспроизведения результатов.

Пример содержания файла параметров Results/parameters.json :

```
{
  "model": "Planar3D",
  "time": 10,
  "mesh": {
    "height": 41,
    "length": 21,
    "cell": {
      "height": 2.7,
      "length": 2.7
    }
  },
}
```

```

"injection": {
  "0": {
    "start time": 0,
    "stop time": 40,
    "fluid injection": 3.6,
    "bulk proppant": 0,
    "rheology index": 1,
    "dynamic viscosity": 0.4
  },
  "1": {
    "start time": 40,
    "stop time": 60,
    "fluid injection": 3.2,
    "bulk proppant": 200,
    "rheology index": 1,
    "dynamic viscosity": 0.4
  }
},
"layers": {
  "0": {
    "y1": 1000,
    "y2": 25,
    "stress": 3.21e+7,
    "Young's modulus": 2.6e+10,
    "Poisson's ratio": 0.23,
    "Carter's coefficient": 6e-06
  },
  "1": {
    "y1": 25,
    "y2": 15,
    "stress": 3.2e+7,
    "Young's modulus": 2.3e+10,
    "Poisson's ratio": 0.25,
    "Carter's coefficient": 4e-06
  },
  "2": {
    "y1": 15,
    "y2": -15,
    "stress": 3.0e+7,
    "Young's modulus": 2.5e+10,
    "Poisson's ratio": 0.25,
    "Carter's coefficient": 0
  },
  "3": {
    "y1": -15,
    "y2": -25,
    "stress": 3.22e+7,
    "Young's modulus": 2.7e+10,
    "Poisson's ratio": 0.24,
    "Carter's coefficient": 3e-06
  },
  "4": {
    "y1": -25,
    "y2": -1000,
    "stress": 32.4,
    "Young's modulus": 2.8e+10,
    "Poisson's ratio": 0.25,
    "Carter's coefficient": 2e-06
  }
}
}

```

Единицы измерения данных в файле `parameters.json` :

```

time = мин

mesh:
  height = шт
  length = шт

```

```
cell:
    height = м
    length = м

injection:
    start time = мин
    stop time = мин
    fluid injection = м^3/мин
    bulk proppant = кг/м^3
    rheology index = б/р
    dynamic viscosity = Па*с

layers:
    y1 = м
    y2 = м
    stress = Па
    Young's modulus = Па
    Poisson's ratio = б/р
    Carter's coefficient = м/с^0.5
```

## Данные об эволюции трещине

Краткая сводка данных об эволюции трещины сохраняется в текстовый файл `Results/fracture_m.txt` .  
Записываются следующие характеристики:

- *Time* – время в минутах;
- *Opening* – призабойное раскрытие в миллиметрах;
- *Pressure* – призабойное давление в Па;
- *Length* – длина трещины в метрах;
- *Height* – высота трещины в метрах;
- *Aspect ratio* – соотношение сторон (длина к высоте).

## Концентрация пропанта в трещине

Значение концентрации пропанта на момент окончания расчётов сохраняется в текстовый файл `Results/concentration_m.txt` в виде матрицы, каждое значение соответствует значению в одной из ячеек расчётной области.

Если требуется, промежуточные значения концентрации сохраняются аналогичным образом в подпапку `Results/Concentration` с указанием моментов времени в названиях текстовых файлов.

## Давление в трещине

Значение давления на момент окончания расчётов сохраняется в текстовый файл `Results/pressure_m.txt` в виде матрицы, каждое значение соответствует значению в одной из ячеек расчётной области.

Если требуется, промежуточные значения давления сохраняются аналогичным образом в подпапку `Results/Pressure` с указанием моментов времени в названиях текстовых файлов.

## Раскрытие трещины

Значение раскрытия на момент окончания расчётов сохраняется в текстовый файл `Results/opening_m.txt` в виде матрицы, каждое значение соответствует значению в одной из ячеек расчётной области.

Если требуется, промежуточные значения раскрытия сохраняются аналогичным образом в подпапку `Results/Opening` с указанием моментов времени в названиях текстовых файлов.

## Возможные ошибки

Для вызова списка возможных ошибок используйте команду `./planar3D --list-errors` .

## Особенности текущей версии программы

Следующий функционал требует подключения дополнительных расчётных модулей (интеграция планируется в ближайшем обновлении):

- учёт слоистости по упругим модулям;
- учёт слоистости для коэффициента Картера;
- расчёт поля температуры.