
PLANES Documentation

Release 1.0

Olivier Dazel, Mathieu Gaborit

Oct 24, 2016

1	General information	3
1.1	Methods of discretisation	3
1.2	Resolution strategies	3
1.3	Material models	3
2	Getting Started	5
2.1	Get the code	5
2.2	Understanding architecture	5
3	Creating a new project	7
3.1	Project file	7
4	Identification of Method, Material & Boundary	9
4.1	Types of numerical models	9
4.2	Types of media	9
4.3	Types of boundaries	10
5	Example and Benchmark : Kundt tube	11
6	Using the PLANES Plane Wave Solver	13
6.1	The multilayer structure	14

PLANES is tool developped at the [LAUM](#) to simulate and compute the behaviour of acoustic systems involving porous materials.

PLANES also provides a context to test hybrid methods for acoustic simulation.

Keywords : porous material, numerical simulation, FEM, DGM, hybrid methods

General information

PLANES relies on a set of Matlab scripts and is completely open-source.

This tool provides several methods of discretisation for fields and several types of resolution strategies which are listed below.

1.1 Methods of discretisation

- 3 Nodes Triangles (TR3)
- 6 Nodes Triangles (TR6)
- Triangle with Hermite interpolation (H12)

1.2 Resolution strategies

- Finite Element Method (FEM)
- Discontinuous Galerkin Method (DGM)
- Plane Waves for multilayer systems

1.3 Material models

As the main goal of this software is to provide an easy-to-configure, reliable way to model and resolve complex systems involving porous materials it's possible to identifying all those different materials/aproximation :

- air
- equivalent elastic solid
- equivalent fluid
- limp model
- Biot model (1998 formulation)
- Biot model (2001 formulation)
- PML

It is, of course, possible to specify periodicity conditions (for code-simplicity sake, periodicity must be defined on compatible boundaries).

1.3.1 Methods Coupling

Through its quite simple code base and its script-based architecture, PLANES is really flexible and is sometimes used to experiment new techniques. The experimentation on FEM & DGM coupling led to the introduction of method coupling abilities.

It's therefore possible to couple FEM & DGM methods, using compatible or incompatible meshes.

Getting Started

2.1 Get the code

As PLANES is in continuous development, the only available version today is on [Github](#).

You can either download the latest version as a [zip file here](#) or get it through [Git](#) to stay on the cutting edge

```
$ git clone https://github.com/OlivierDAZEL/PLANES
```

2.2 Understanding architecture

In the `src/` folder, you'll find all the scripts, the folders inside `src/` have self-explanatory names :

FEM/ & DGM/ & PW/ Implementation of Finite Elements & Discontinuous Galerkin Methods, Plane wave based methods (multilayer)

Material/ & Physics/ Material properties & modelling

Mesh/ Mesh management routines (reading, modification, tagging, etc...)

Plots/ Output functions

Polynomials/ Function to combine, derive, integrate, etc.. polynomials

Problems/ & Solutions/ Directory for m-files describing configurations & analytical solutions of problems (legacy)

The most important for you will be `src/Main`. This folder contains the main routines, `PLANES_main.m` for legacy projects & `PLANES.m`¹ for newer, separate-directory projects.

Other files are initialization, logging and postprocessing routines.

¹ Which contains the `PLANES()` function.

Creating a new project

Note: The following section only discuss separate-directory projects.

To start a new project, create a directory separated from the `src/` directory of PLANES (your dir may be anywhere on the file system) and create a `.m` file inside

```
$ mkdir /home/you/MyPLANESProject/  
$ cd /home/you/MyPLANESProject  
$ touch MyProject.m
```

Some other directories will be useful too (such as output for info-files, graphs, FreeFEM programs & data files)

```
$ mkdir -p out Profiles m m/Materials FF
```

The files are organised as follows:

- The root m-file is used to start the experiment (its contents are details hereafter)
- Each subproject has a m-file in `m/` and if needed a EDP (FreeFEM) file in `FF/`. They are named : `<project name>_<num of subproject>_data.m` and `<project name>_<num of subproject>.m`
- The materials related m-files used for the project are stored in `m/Materials/`
- Output (text) files from runs will be sent to `out/` and graphs to `Profiles/`
- A `m/<project name>_<num of subproject>_postprocess.m` can be used to define a post-processing routine and it will be triggered after the resolution.

Tip: You can consider a project as a global experiment. The next step is therefore to specify the different configurations for the experiment. If one wants for example to test reflection on a porous material set behind a membrane, it may be interesting to try different parameters for the membrane : all the configurations are about the same experiment, thus in the same project dir. Configurations are identified by appending an underscore (`_`) and a number at the end of m-file and `.edp`.

3.1 Project file

The main file must define some structures, see the example below for more information :

```
clear all
close all
current_dir=pwd;
PLANES_dir='.././src/Main/';

frequency.nb=5; % number of frequencies (negative : log spaced)
frequency.min=100;
frequency.max=20000;

% Information about the output graphs
data_model.profiles.mesh=1; % plot the mesh
data_model.profiles.x=0; % profiles along x/y axis
data_model.profiles.y=0;
data_model.profiles.map=0; % plot a 2D map of the solution
data_model.profiles.custom=0; % custom plots activation
data_model.profiles.custom_plots = {}; % cellarray of strings, scripts stored in m/

% export data ?
data_model.export.profiles=0;
data_model.export.nrj=1;
data_model.export.reset=0; % delete old files before export

% log level (0 -> 2)
data_model.verbosity = 1;

% DGM related (number of waves of the reconstruction basis & initial tilt)
data_model.theta_DGM.nb=4;
data_model.tilt=0;

% Fully separated projects
name.project_directory=current_dir;

% Call to PLANES
cd(PLANES_dir)
PLANES('Project Name', num_of_subproject, data_model, frequency, name)
cd(current_dir);
```

Identification of Method, Material & Boundary

The different elements of the model are specified using labels. Different structures keep track of those labels:

- `elem.model(n)` type of modelling for the n-th element
- `edge_msh(n, 3)` type of boundary for the n-th edge
- `elem.label(n)` identifier for the material (can be set in the FreeFEM++ mesher)

4.1 Types of numerical models

Label	Solving Method
1	FEM on TR6 (6 nodes triangles)
2	FEM on H12 (3 nodes triangles with Hermite splines interpolation)
3	FEM on TR3 (3 nodes triangles)
10	DGM on Triangles
11	DGM on Rectangles

Tip: The meshes between FEM and DGM can be incompatible.

4.2 Types of media

Label	Media
0	Air
1XXX	Elastic medium
2XXX	Equivalent fluid (rigid frame) material
3XXX	Limp model
4XXX	Poroelectric Material (or FEM with 1998 formulation)
5XXX	Biot (or FEM with 2001 formulation)
80xy	PML (with x and y boolean direction)

4.3 Types of boundaries

Label	Type of Boundary
1	Rigid wall
2	Unit pressure (fluid)
3	Unit normal velocity
4	Unit tangential velocity
5	Sliding (PEM)
6	Bonded (PEM) or clamped (elastic)
7	Unit pressure (PEM)
8	Unit normal velocity (PEM)
9	Unit tangential velocity (PEM)
10	Incident air plane wave on acoustic/Biot98 element
11	Incident air plane wave on elastic element
12	Incident air plane wave on Biot 2001 element
13	DtN plate (Dirichlet to Neumann)
20	Transmitted air plane wave on acoustic/Biot 98 element
21	Transmitted air plane wave on elastic element
21	Transmitted air plane wave on Biot 2001 element
60	Unit normal velocity on H12 with flux application
61	DGM radiative boundary
98	Periodicity left
99	Periodicity right
4xx	ZOD even/odd
400	FSI (Fluid-Structure interaction)
1xyz	Excitation wave 1 angle xyz in degree (PEM)
2xyz	Excitation wave 2 angle xyz in degree (PEM)
3xyz	Excitation wave 3 angle xyz in degree (PEM)
500	Velocity diffraction cylindre FEM

Example and Benchmark : Kundt tube

PLANES is bundled with an example project (also used as a benchmark) : a simple Kundt tube of unit length.

The different subprojects implement different methods and couplings :

Subproject	Method(s)
0	FEM on TR6
1	FEM on H12
2	FEM on TR6 coupled to FEM on H12
3	DGM on TR
4	DGM on H
5	DGM on TR coupled to DGM on H
6	FEM on H12 coupled to DGM on H
7	FEM on TR6 coupled to DGM on H

These examples can show you how to organized a multi-subprojects project directory as well as how to use [FreeFEM++](#) and PLANES together.

Using the PLANES Plane Wave Solver

The Plane Wave solver relies on a method close to the Transfer Matrix Method. It can be used to model a wide range of layered systems, even including 3D fully anisotropic PEMs.

Todo

add the paper (2013)

The solver relies on project similar to those for the FEM/DGM solver. The function to call is the following

```
PLANES_Multilayer(name, number, data_model, multilayer, frequency)
```

With:

- `name` is a string associated to the name of the project. This string is the same than the folder in the project area.
- `number` is an integer. This is the number of the subproject.
- `data_model` is a structure that contains the data of the model. In the present case, it only contains the angle of incidence.
- `multilayer` is a structure array which contains one or several multilayer structures (see below).
- `data_model` is a structure that contains the data of the model. In the present case, it only contains the angle of incidence. It can be either a real number or an array of two real numbers.
- `frequency` is the structure associated to frequency (see below).

At the end of the resolution, the solver outputs a PW file named `out/<project name>_<num of subproject>.PW`.

It's a plain text file with $6n+1$ columns. The first column corresponds to the frequency axis, the others are the following (for each multilayer structure):

- Absorption coefficient
- Real part of the reflexion coefficient
- Imaginary part of the reflexion coefficient
- Transmission loss
- Real part of the transmission coefficient
- Imaginary part of the transmission coefficient

6.1 The multilayer structure

The different multilayer structures are gathered in a single object containing several arrays:

- `multilayer.nb(l,m)` correspond to the number of layer of multilayer structure `m`.
- `multilayer.termination(l,m)` correspond to the termination condition of multilayer structure `m`. The value is 0 for rigid backing and 1 for a radiation condition
- `multilayer.d(l,m)` correspond to the thickness of the layer `l` of multilayer structure `m`. It is a real number.
- `multilayer.mat(l,m)` correspond to the material of the layer `l` of multilayer structure `m`. The label is an integer associated to the convention presented here.