EDtoolbox manual, v 0.1

Peter Svensson, NTNU

January 22, 2018

1 Introduction

2 Program structure for EDmain_convexESIE

The main function EDmain_convexESIE runs through the following blocks, in this given order.

2.1 EDcheckinputstructs

This function checks the input data and assigns default values to input parameters that have not been specified.

2.2 EDreadcad or EDreadgeomatrices

The geometry can either be specified in a separate .cad file, or given as input data matrices. See more on this topic in Section XX. Basically, the geometry is specified as a set of corners (vertices) and planes (faces/polygons). They are stored in a struct called `planedata`.

2.3 EDedgeo

This function identifies all the edges of the polyhedron, and stores data about them in a separate struct called `edgedata`.

2.4 EDSorRgeo

This function is run twice, once to find the visibility data for the source, and the second time to find the visibility data for the receiver. The visibility data tells what edges and planes each source and receiver can see.

2.5 EDfindconvexGApaths

This functions finds all the valid direct sound paths, specular reflection paths, and first-order diffraction paths. This function is specialized on the case of external, convex scattering objects, and for such objects, each source-receiver combination can have maximum one specular reflection. The results are stored in a struct called `firstorderpathdata`.

2.6 EDmakefirstordertfs

Based on the paths specified in the struct `firstorderpathdata`, the function `EDmakefirstordertfs` generates the transfer functions `tfdirect, tfgeom,` and `tfdiff`.

2.7 EDed2geo

This function is run only if difforder $>$ 1. It identifies which edges see which other edges, and stores this information in a struct `edgetoedgedata`. Clearly, this is needed only if the requested diffraction order ¿ 1.

2.8 EDinteg_submatrix

This function is run only if difforder $>$ 1. It identifies the submatrix structure for the subsequent integral equation solving, by the function `EDintegral_convex_tf`.

2.9 EDintegral_convex_tf

This function is run only if difforder $>$ 1. It computes the ackumulation of higher-order diffraction, from order 2 up to a specified diffraction order. The result is stored in the transfer function `tfinteqdiff`.

3 Geometry format

The EDtoolbox handles only polyhedra, including polygonally shaped thin discs/plates. In the EDtoolbox, a polyhedron is defined in terms of 'corners' (vertices) and 'planes' (faces/polygons). These can either be specified directly in the input struct geofiledata (fields `.corners` and `.planecorners`), or in a separate file of the .cad-format, which is a format exported by the CATT-Acoustic software.

3.1 Corners

The .corners field is straightforward: it is a matrix of size [ncorners,3] where row n contains the x-,y- and z-coordinates of corner number n. If the .cad-file had a non-contiguous numbering of the corners, a renumbering will be done for the EDtoolbox, starting with number 1.

3.2 Planes

The .planecorners field is a matrix of size [nplanes,nmaxnumberofcornersperplane] where row n gives the corners that define plane n. The corners must be defined in a counter-clockwise order, as seen from the frontal side of the plane.

4 Input data

The main program, EDmain_convexESIE, is run with 6 structs containing all input parameters:

```
EDmain_convexESIE(geofiledata,Sindata,Rindata,envdata,...
controlparameters,filehandlingparameters)
```

Table 1: Input data struct `geofiledata`

Field name	Required?	Default value	Size
`.geoinputfile`	Alt. A (see below)	—	—
`.corners`	Alt. B (see below)	—	[ncorners,3]
`.planecorners`	Alt. B (see below)	—	[nplanes,nmax][1]
`.firstcornertoskip`	—	1e6 [2]	

Three alternatives exist for the struct `geofiledata`
A. An external `.cad`-file is specified in the field `.geoinputfile`
B. If the field `.geoinputfile` is not specified, then the fields `corners` and `planecorners` can give the geometry data.
C. If neither of the two alternatives above apply (e.g., if the entire struct is left empty), then a file opening window will appear, and a `.cad`-file can be selected.
Priority will be given to the `.geoinputfile` if both alternatives A and B are given.
See section 3 for more information on the geometry format.
[1] The value nmax is the maximum number of corners per plane.
[2] The field `.firstcornertoskip` implies that all edges with at least one corner number having the value of `.firstcornertoskip`, or higher, will be deactivated. This gives the possibility to study cases with a subset of all the edges of a model.

Table 2: Input data struct `Sindata`

Field name	Required?	Default value	Size/value
`.coordinates`	Yes	—	[nsources,3]
`.doaddsources`	—	0	0 or 1[1]
`.sourceamplitudes`	—	ones(nsources,1)	[nsources,1]

[1] If this value is set to 1, the contributions from all sources will be added and saved in a single transfer function, after being multiplied by the values in the vector `.sourceamplitudes`. This is a straightforward way to simulate extended sources, or vibration patterns. See section 5 for a description of the scale values.

5 Output data

The main program, EDmain_convexESIE, is run with 6 structs containing all input parameters:

Table 3: Input data struct `Rindata`

Field name	Required?	Default value	Size
`.coordinates`	Yes	—	[nreceivers,3]

Table 4: Input data struct `envdata`

Field name	Required?	Default value	Size
`.cair`	—	344	—
`.rhoair`	—	1.21	—

Table 5: Input data struct `controlparameters`

Field name	Required?	Default value	Size, or possible values
`.docalctf`	—	1	0 or 1 [1]
`.docalcir`	Irrelevant [2]	1	0 or 1
`.frequencies`	Yes [3]	—	[1,nfrequencies]
`.fs`	Irrelevant [2]	44100	—
`.directsound`	—	1	0 or 1
`.difforder`	—	15	integer >= 0
`.nedgepoints_visibility`	Irrelevant [4]	2 [5]	—
`.Rstart`	—	0 [6]	—
`.discretizationtype`	—	2	0 or 2 [7]
`.ngauss`	—	16 [8]	even integer >= 2

[1] If the field `.docalctf` is set to 0, edges will be derived and source/receiver visibility will be computed. Please note that to have any use for these calculations, you must specify in the struct `filehandlingparameters` that the proper geometry information is saved.
[2] The sampling frequency, `fs`, and the parameter `.docalcir`, is used in upcoming time-domain calculation functions, but is not read/used by EDmain_convexESIE.
[3] A list of frequencies must be specified for the main function EDmain_convexESIE, and other upcoming frequency-domain versions (unless `.docalctf` is 0). It is not needed for time-domain versions.
[4] This parameter specifies how many points along each edge will be tested for visibility. This is irrelevant for convex scattering bodies since either the whole edge or no part of an edge is visible. It is relevant for upcoming calculation alternatives for non-convex geometries.
[5] The default value of 2 implies that the two end points of each edge will be tested for visbility.
[6] The parameter `.Rstart` determines the phase of the final transfer function (or the definition of time zero in upcoming time-domain calculation alternatives). To simulate an incoming plane wave with amplitude 1, and phase zero, at the origo, then `.Rstart` should be set to the distance to the far-away point source.
[7] The value 0 implies a uniform discretization of the edges. The value 2 gives a Gauss-Legendre discretization. The value 1 is obsolete/not used.
[8] The value `.ngauss` specifies the number of quadrature points along the longest edge. It will be scaled down linearly based on the length of each edge, and an even number of quadrature points will always be chosen.

Table 6: Input data struct `filehandlingparameters`

Field name	Required?	Default value	Possible values
`.outputdirectory`	Yes [1]	Same as geoinputfile [2]	—
`.filestem`	Yes[1]	Name of cad-file	—
`.savesetupfile`	—	1	—
`.savecadgeofile`	—	0	—
`.saveSRdatafiles`	—	0	—
`.saveeddatafile`	—	0	—
`.savesubmatrixdata`	—	0	—
`.saveinteqsousigs`	—	0	—
`.loadinteqsousigs`	—	0	—
`.savepathsfile`	—	0	—
`.saveISEStree`	—	0	—
`.savelogfile`	—	1	—
`.savediff2result`	—	0	—

[1] If the geometry is given in the form of the input fields `.corners` and `.planecorners`, then the fields `.outputdirectory` and `.filestem` must be specified.

[2] Note that a folder called "results" will be generated in the output directory (if it doesn't already exist). All result files will be saved in that results folder.