ADVISE

Advanced Inspection of Complex Structures

D1.4
Semi-analytical simulation tools for scattering media suitable for online use

General information	
Grant agreement number	755500
Start date of the project	01/09/2017
Project duration	48 months
Due date of the deliverable	31/08/2019
Actual submission date	29/10/2020
Lead beneficiary	3 – FRAUNHOFER

Keywords

Plugin, scripting, Python, scattering, attenuation, isotropic/transversely-isotropic material, generalized point source synthesis, sound field, wave propagation, stiffness/elasticity matrix, Voigt's notation

Type	Meaning	
R	Document, report	
DEM	Demonstrator, pilot, prototype	
DEC	Websites, patent fillings, videos, etc.	
OTHER	Software, technical diagram, etc.	x

Dissemination Level		
PU	Public	x
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Contents

Table of Figures

Glossary

Abbreviations/Acronym	Description
ANGPPS	Generalized point source synthesis in anisotropic materials
GUI	Graphical user interface
US	Ultra sound
NDT	Non-destructive testing

1 Executive Summary

This document describes the newly implemented plugin mechanism in CEA’s CIVA application at the hand of two example plugins deduced from FRAUNHOFER applications, namely the scattering model and generalized point source synthesis for transversely-isotropic media (ANGPSS). The plugins via Python scripts extend CIVA’s flexibility and range of application considerably for external users in a generic way.

The FRAUNHOFER scattering model derives effective elastic properties, wave propagation parameter and attenuation properties due scattering effects at grain boundaries from simpler, more fundamental material models. Thus, it contributes to the understanding of wave propagation and scattering phenomina in complex materials. ANGPSS provides an easy-to-use, simplified model for sound field calculations/simulations in isotropic and transversely-isotropic materials with good performance. It also contributes to the understanding of wave propagation in the materials under consideration. The output from the scattering model can optionally help enhance the output of ANGPSS by taking the attenuation values into account. The integration of both models into the CIVA scripting environment demonstrates the operability, adds new features and can contribute to improve the setup of practical inspection experiments in complex media.

2 Introduction

In the following we describe the usage and parameterization of software tools that implement FRAUNHOFER’s scattering model [2] and semi-analytical simulation approach for wave propagation (ANGPSS) [3] together with their integration into CIVA by utilizing CEA’s newly developed plugin mechanism [1]. In short, the scattering model derives wave propagation, attenuation parameter and elastic properties in transversely-isotropic materials by accounting for scattering effects in polycrystalline media with textured grain orientations, whereas ANGPSS is dedicated to sound field calculations in these materials with an approximate semi-analytical approach.

The major aim of these efforts is to support, enhance and extend simulation functionalities, which are needed for designing, planning and interpreting results from actual inspection scenarios. The results can provide substantial input for tasks in WP3, in particular T3.1.

The tools described herein can help understanding how US-signals behave and propagate in complex material structures, and give guidance on how to optimize an NDT experiment facing inherent physical limitations.

3 Integration Participants

CEA implemented a Python script engine in CIVA to accommodate third party developments. This is a major step towards enhanced customisability and extensibility of an otherwise relatively self-contained application. It offers the great opportunity for everyone to provide add-ons without touching the kernel of a program, especially when using a commonly-known programming language (Python). It can also serve as a rapid prototyping facility where algorithms are tested and evaluated on a script base and later - when proved to be of value - incorporated into the application's core if performance benefits dictate this. The CIVA plugin mechanism is described in more detail for two examples in the following section.

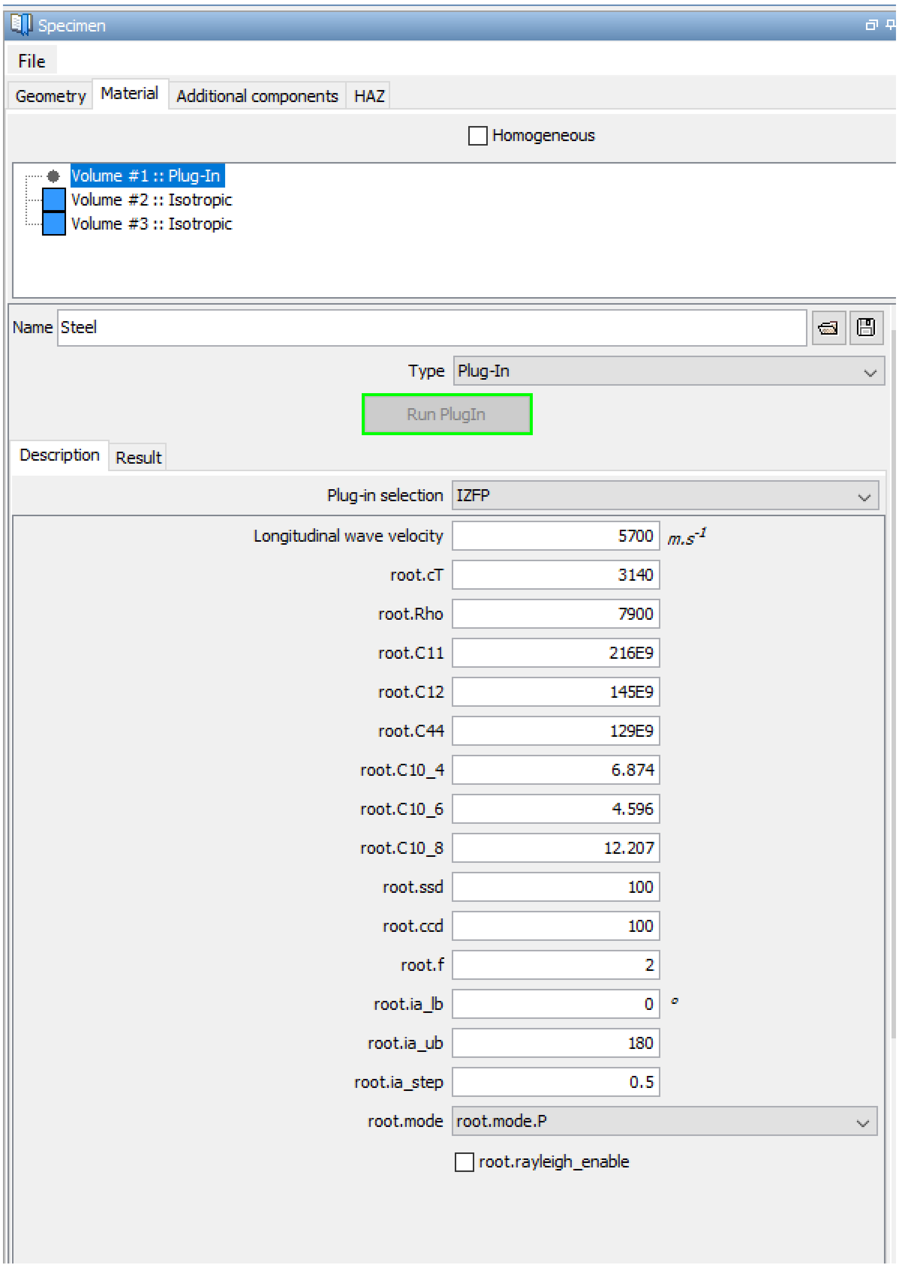
3.1 CIVA Plugin Mechanism

The newly integrated script engine in CIVA allows the execution of external Python routines and modules. In order to have a script binding work properly, some principal prerequisites must be generally met on both the host side (CIVA) and the script side (Python) as well: 1) The host must be aware of the script. This requires some sort of script registration mechanism. 2) The data transfer between host and script must be established. 3) Entry points for script execution must be defined in the host.

One possibility of script registration is file-based: Scripts reside in special subfolders of the CIVA plugin folder, identifying a specific plugin modul. On host's program start these subfolders are searched for valid plugin scripts. Accepted script moduls appear in form of separate items in the host's toolbar. CIVA also offers an alternative registration process, which is completely integrated in CIVA and detailed in [1].

Regarding the data transfer the required input variables for a considered plugin must be defined in a generic CIVA-GUI (see Fig. 1). These registered variables are published to the script at execution time thus being accessible/retrievable with their actual values in the script. The link back from script to host is achieved via import libraries that provide modules/objects with individual functionalities to manipulate the state of the host and especially return data to the host from within a script.

The plugin execution itself is done at specific entry points by assigning appropriate values to the previously defined input variable and running the plugin's main script, which may embrace other scripts/routines or call external stand-alone applications. After having finished its job the host is signalled and can show the script returns as far as standard host functionality is concerned (see Fig. 2). Otherwise the script itself can visualize its results if necessary.



□ Homogeneous

Volume #1 :: Plug-In
Volume #2 :: Isotropic
Volume #3 :: Isotropic

Name Steel

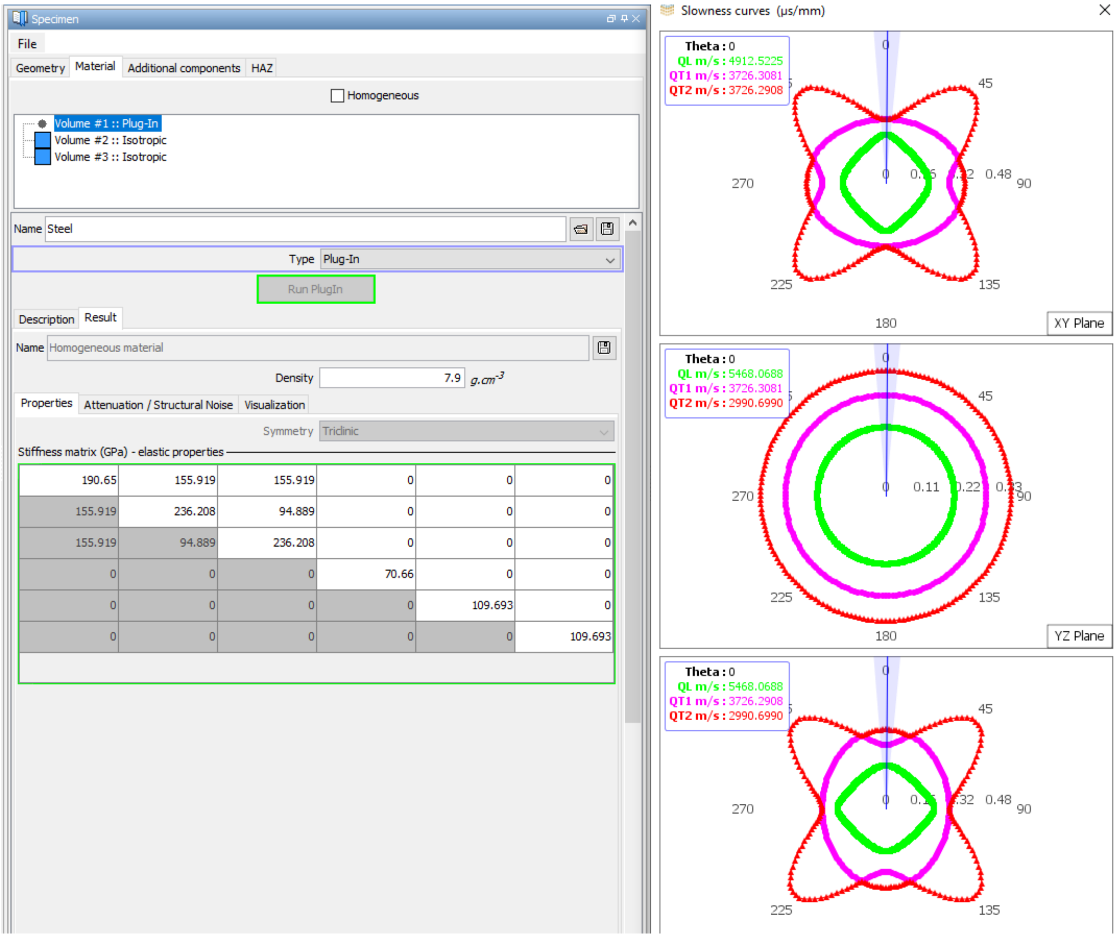
Type Plug-In

Plug-in selection IZFP

Longitudinal wave velocity 5700 $m.s^{-1}$
root.cT 3140
root.Rho 7900
root.C11 216E9
root.C12 145E9
root.C44 129E9
root.C10_4 6.874
root.C10_6 4.596
root.C10_8 12.207
root.ssd 100
root.ccd 100
root.f 2
root.ia_lb 0 °
root.ia_ub 180
root.ia_step 0.5
root.mode root.mode.P

□ root.rayleigh_enable

Figure 1: CIVA generic GUI for input variables in FRAUNHOFER scattering model



Specimen

File

Geometry | Material | Additional components | HAZ

□ Homogeneous

Volume #1 :: Plug-In
Volume #2 :: Isotropic
Volume #3 :: Isotropic

Name Steel

Type Plug-In

Run PlugIn

Description | Result

Name Homogeneous material

Density 7.9 $g.cm^{-3}$

Properties | Attenuation / Structural Noise | Visualization

Symmetry Triclinic

Stiffness matrix (GPa) - elastic properties

190.65	155.919	155.919	0	0	0
155.919	236.208	94.889	0	0	0
155.919	94.889	236.208	0	0	0
0	0	0	70.66	0	0
0	0	0	0	109.693	0
0	0	0	0	0	109.693

Slowness curves (µs/mm)

Theta : 0
QL m/s : 4912.5225
QT1 m/s : 3726.3081
QT2 m/s : 3726.2908

Theta : 0
QL m/s : 5468.0688
QT1 m/s : 3726.3081
QT2 m/s : 2990.6990

Theta : 0
QL m/s : 5468.0688
QT1 m/s : 3726.2908
QT2 m/s : 2990.6990

Figure 2: Example output scattering model in CIVA

3.2 Integration of Scattering Model

In order to fit into the CIVA plugin mechanism the GUI was stripped off of the original code base. The program receives its input via a plain text file and produces its output also in plain text files. Note: Since the scattering model is a Matlab application, it is absolutely mandatory to have installed the same Matlab runtime environment, which the app was built with and which can be obtained by Mathworks for free. Usually this pre-condition is guaranteed, when CIVA is installed correctly. The implementation as a CIVA plugin comes along in two versions that differ in their output and are partially equivalent. They are discussed subsequently.

3.2.1 Batch version for complex wavenumber

The batch version has the same functionality as the GUI version, but instead of the Materialliste.txt a txt-file for all input parameters is used.

Files to Package for Standalone

- CalcComplexK_App.exe
- readme.txt
- Input.txt
- splash.png

Execution and Functionality

To execute the standalone application correctly, an Input.txt with the structure as described in the following section is necessary. The input file is required to be in the same folder. Then the executable can be run. When executed, a splash display is briefly shown to indicate the calculation process. A short time later, the result files are saved as *.txt in the folder of the app. Two separate files, one file with the real part and one file with the imaginary part of the calculated complex solution, are stored. The solution is a column vector, divided into the individual angular steps that were entered. All values are saved with the unit $\frac{1}{mm}$. In the plugin version the input file is generated in the main-script by using the published input variables from CIVA and writing them to file. Similarly the script reads in the produced output files and calculates the expected stiffness matrix.

Input File

The input file must contain the following parameters:

- isotropic longitudinal wave velocity: c_L in $\frac{m}{s}$
- isotropic shear wave velocity: c_T in $\frac{m}{s}$
- density of the material: ρ in $\frac{kg}{m^3}$
- the single crystal constants for cubic crystals: C_{11}^0, C_{12}^0, C_{44}^0 in $\frac{N}{m^2}$
- grain diameter in isotropic case: ssd in micrometer
- columnar crystal diameter for calculation of anisotropic case: ccd in micrometer
- measurement frequency: f in MHz
- angles of incident wave in relation to the columnar crystal axis:

  from ia_lb in degree
  to ia_ub in degree
  with a stepwith ia_step in degree
- one can choose the orientation of columnar crystals between:

  vertical: orientation = 1
  horizontal: orientation = 2
- one can choose the mode of transmitted wave:

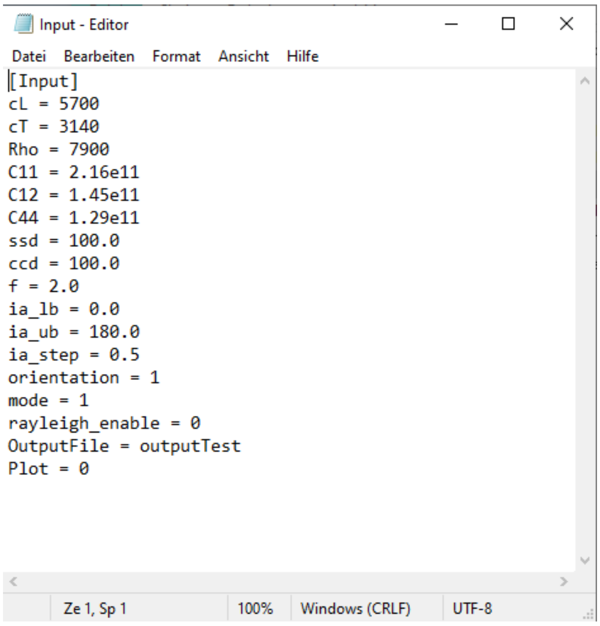
  longitudinal: mode = 1
  shear: SV mode = 2
  SH mode = 3
- one can choose, if Rayleigh approximation should be used:

  Rayleigh off: rayleigh_enable = 0
  Rayleigh on: rayleigh_enable = 1
- one can choose a name for output file, which will be stored in the same folder as the application file:

  OutputFile
- for testing one can choose whether a plot should be opened in addition to the output file:

  no plot: Plot = 0
  plot: Plot = 1

An example is given in Figure 3.



```
[Input]
cL = 5700
cT = 3140
Rho = 7900
C11 = 2.16e11
C12 = 1.45e11
C44 = 1.29e11
ssd = 100.0
ccd = 100.0
f = 2.0
ia_lb = 0.0
ia_ub = 180.0
ia_step = 0.5
orientation = 1
mode = 1
rayleigh_enable = 0
OutputFile = outputTest
Plot = 0
```

Figure 3: Example of a scattering model input file

3.2.2 Batch version for complex stiffness matrix

The batch version for a calculation of the complex stiffness matrix is realized in the same manner as the batch version for the complex wavenumber, but in addition with the calculation described in chapter **Error! Reference source not found.** of Del1.2 [2].

Files to Package for Standalone

- CalcComplexStiffness.exe
- readme.txt
- Input.txt
- splash.png

The execution, functionality and input follow the same scheme as in 3.2.1. The output file contains now the calculated complex stiffness matrix in Voigt's notation in units of Pa.

Example of results

An example of a calculated complex stiffness matrix is given in Figure 4. The same data were used as in the example in section **Error! Reference source not found.** for a vertical weld made of X6CrNi18 11 with the material data from **Error! Reference source not found.** in Del 1.2 [2] and a frequency of 2MHz. The Rayleigh approximation, which assumes scattering to be inversely proportional to the fourth power of the wavelength, was not used, a discussion of which can be found in D1.2.

```
2.36342649E+11 9.49776763E+10 1.53525938E+11 0.00000000E+00 0.00000000E+00 0.00000000E+00
9.49776763E+10 2.36342649E+11 1.53525938E+11 0.00000000E+00 0.00000000E+00 0.00000000E+00
1.53525938E+11 1.53525938E+11 1.91175165E+11 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 1.09560848E+11 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 1.09560961E+11 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 7.06824865E+10
+i
6.26685570E+06 -1.33483319E+08 -4.21169835E+07 0.00000000E+00 0.00000000E+00 0.00000000E+00
-1.33483319E+08 6.26685570E+06 -4.21169835E+07 0.00000000E+00 0.00000000E+00 0.00000000E+00
-4.21169835E+07 -4.21169835E+07 3.35558040E+02 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 2.75609977E+02 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 6.38388911E+01 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 6.98750871E+07
```

Figure 4: Calculated stiffness matrix for vertical oriented austenitic weld X6 CrNi 18 11

3.3 Integration of ANGPSS

The integration of ANGPSS follows similar rules as for the scattering model. Since ANGPSS is a statically linked native C++ application, it has no further dependencies. It takes an input file, does its calculation/simulation (i.e. sound field) and writes its results to an output file. The theoretical background will be detailed in an extra-report [3].

Input File

The input file is a plain text file and is logically (not formally) divided into several sections. The first section contains two filenames (comma-separated) representing the output from the scattering model (see 3.2) for attenuation purposes, the polarization mode to be calculated (i.e. SH, qP or qSV) and an “incidence” angle range. For all practical use cases, this angle range should be the same as given in the scattering model and fixed to the range of 0° - 180 °. The second section contains the excitation signal, which describes the type of excitation and its parameters. This is currently restricted to simple monochromatic signals with frequency [MHz], Amplitude [mm] and direction [unit vector] as parameter. (An alternative signal type handles measured data from a transducer surface scan and will not be detailed here.) The third section specifies a material, defined by type identifier (1=isotropic, 2= transversely-isotropic) followed by the type-dependant material parameter list. For isotropic materials the latter consists of the mass density [g/cm^3] and the longitudinal and transversal sound velocity [km/s]. For transversely-isotropic media the parameter list includes the mass density, the entries C_{11}, C_{33}, C_{44}, C_{66}, C_{13} [GPa] of the stiffness matrix in Voigt’s notation with the anisotropy axis **a** in x-direction and a 3-element unit vector for the preferential anisotropy axis **a**. Note that the given stiffness parameters are internally transformed into the coordinate-invariant Lamé constants. The following two sections define the excitation area, i.e. the area where the excitation signal is applied to, and the inspection area, i.e. that area where the sound field is calculated in. Areas are typically described by their geometric properties, i.e. boundaries and discretization, and their kinematic state, i.e. position **P**=(x,y,z)[mm] and orientation (α,β,γ)[°]. Note that the geometry defines – if not stated otherwise – a point set in a virtual x-y-plane. These points are rotated and translated according to $\boldsymbol{x} \to \boldsymbol{P} + \boldsymbol{O} \cdot \boldsymbol{x}$**,** with $\boldsymbol{O} \coloneqq \boldsymbol{O}_z(\gamma) \cdot \boldsymbol{O}_y(\beta) \cdot \boldsymbol{O}_x(\alpha)$ and rotation matrices $O_s, (s = x, y, z)$ about coordinate axes. Also note as a rule of thumb that the discretization of the excitation area should have a resolution better than half of the minimal wavelength in the system. The last section of the input file contains the name of the output file and some internal control parameter. In future expansions the sections with the material-area combination could be repeated in order to account for discrete inhomogenities or multi-material setups. More formally the input file looks like

<scatteringfiles>

<mode> ϵ {0 = SH, 1 = qP, 2 = qSV }

<anglerange>

<signal>
<material>
<excitationarea>
<inspectionarea>
<outputfile>
<controlparameter>

<anglerange> = <startangle>,<endangle>,<anglestep> (don’t touch)

<signal> = <stype>,<frequency>,<amplitude>,<direction>

<material> = <mtype> ϵ {1 = isotropic, 2 = transversely-isotropic}

<density>

<mparameter>

<excitationarea> = <atype> ϵ {4 = circle, 5 = rectangle }

<geometry>

<kin. State> = <position>,<orientation>

<mparameter> = <c_L>, <c_T> | <C_{11}>,<C_{33}>,<C_{44}>,<C_{66}>,<C_{13}>

<a_x>,<a_y>,<a_z>

<geometry> = <radius>,<NPoints>,0 | <Lbound X>, <Ubound X>, <NPointsX>,<Lbound Y>, <Ubound Y>, <NPointsY>

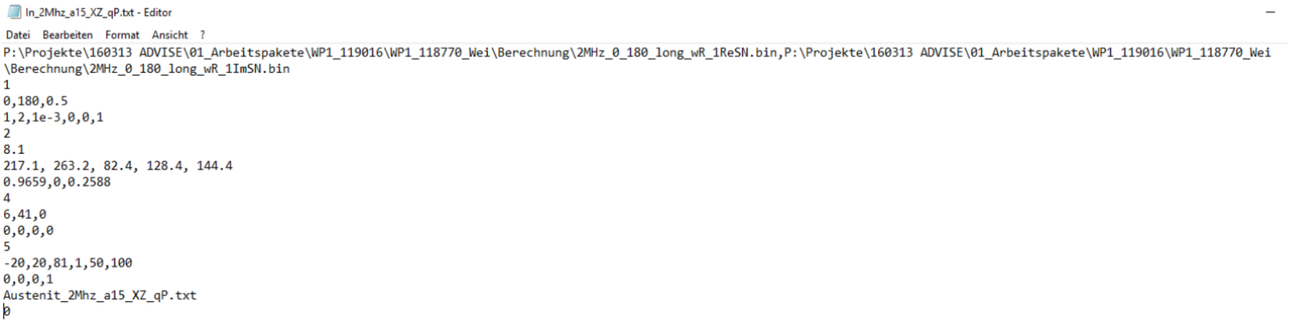
<position> = <x>,<y>,<z>

<orientation> = <otype> | <alpha>,<beta>,<gamma>

<controlparameter> = 0 (don't touch)

Terminal entities, i.e. those not further specified, represent simple numerical values.

Note: This format is not carved in stone; it presently works fine, but may be subject to changes in future releases.


In_2Mhz_a15_XZ_qP.txt - Editor
Datei Bearbeiten Format Ansicht ?

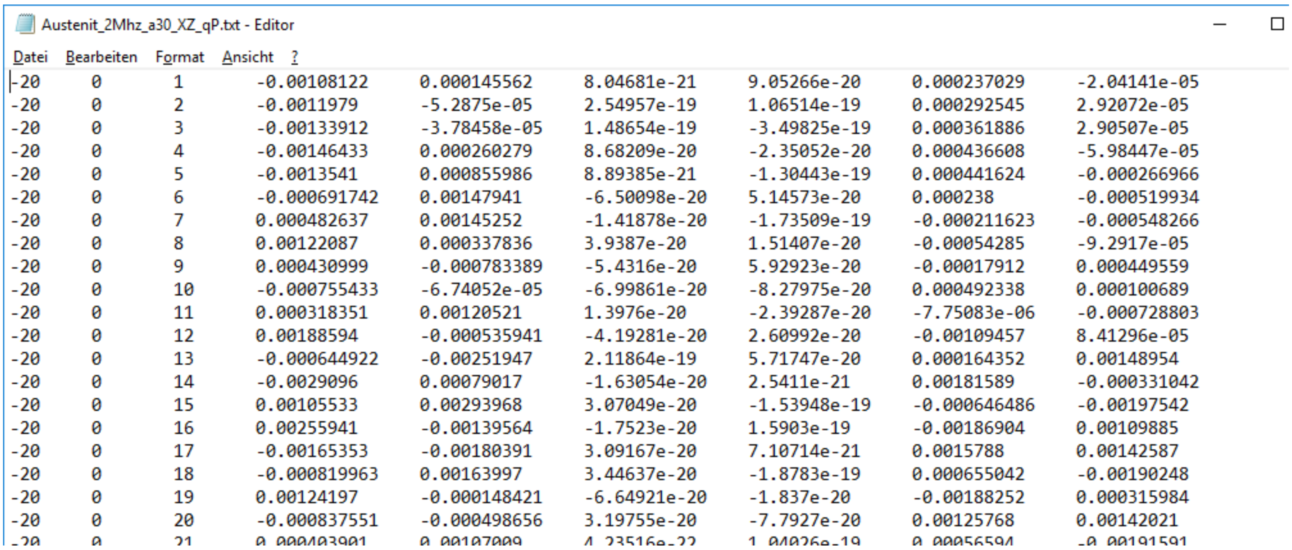
```
P:\Projekte\160313 ADVISE\01_Arbeitspakete\WP1_119016\WP1_118770_Wei\Berechnung\2MHz_0_180_long_wR_1ReSN.bin,P:\Projekte\160313 ADVISE\01_Arbeitspakete\WP1_119016\WP1_118770_Wei
\Berechnung\2MHz_0_180_long_wR_1ImSN.bin
1
0,180,0.5
1,2,1e-3,0,0,1
2
8.1
217.1, 263.2, 82.4, 128.4, 144.4
0.9659,0,0.2588
4
6,41,0
0,0,0,0
5
-20,20,81,1,50,100
0,0,0,1
Austenit_2Mhz_a15_XZ_qP.txt
0
```

Figure 5: Example ANGPSS input file for an anisotropic material

Output File

The output file is also a plain text file and contains the sound field displacement **u** $\in \mathbb{C}^3$ [mm] together with space points **P**=(x,y,z) [mm], where **u** is calculated, in table form. Each row corresponds to a tab-separated ("\t") **P**-**u** combination with following column arrangement:

<x>\t<y>\t<z>\t Re<u_x>\tIm<u_x>\tRe<u_y>\tIm<u_y>\tRe<u_z>\tIm<u_z>



```
-20     0       1       -0.00108122     0.000145562     8.04681e-21     9.05266e-20     0.000237029     -2.04141e-05
-20     0       2       -0.0011979      -5.2875e-05     2.54957e-19     1.06514e-19     0.000292545     2.92072e-05
-20     0       3       -0.00133912     -3.78458e-05    1.48654e-19     -3.49825e-19    0.000361886     2.90507e-05
-20     0       4       -0.00146433     0.000260279     8.68209e-20     -2.35052e-20    0.000436608     -5.98447e-05
-20     0       5       -0.0013541      0.000855986     8.89385e-21     -1.30443e-19    0.000441624     -0.000266966
-20     0       6       -0.000691742    0.00147941      -6.50098e-20    5.14573e-20     0.000238        -0.000519934
-20     0       7       0.000482637     0.00145252      -1.41878e-20    -1.73509e-19    -0.000211623    -0.000548266
-20     0       8       0.00122087      0.000337836     3.9387e-20      1.51407e-20     -0.00054285     -9.2917e-05
-20     0       9       0.000430999     -0.000783389    -5.4316e-20     5.92923e-20     -0.00017912     0.000449559
-20     0       10      -0.000755433    -6.74052e-05    -6.99861e-20    -8.27975e-20    0.000492338     0.000100689
-20     0       11      0.000318351     0.00120521      1.3976e-20      -2.39287e-20    -7.75083e-06    -0.000728803
-20     0       12      0.00188594      -0.000535941    -4.19281e-20    2.60992e-20     -0.00109457     8.41296e-05
-20     0       13      -0.000644922    -0.00251947     2.11864e-19     5.71747e-20     0.000164352     0.00148954
-20     0       14      -0.0029096      0.00079017      -1.63054e-20    2.5411e-21      0.00181589      -0.000331042
-20     0       15      0.00105533      0.00293968      3.07049e-20     -1.53948e-19    -0.000646486    -0.00197542
-20     0       16      0.00255941      -0.00139564     -1.7523e-20     1.5903e-19      -0.00186904     0.00109885
-20     0       17      -0.00165353     -0.00180391     3.09167e-20     7.10714e-21     0.0015788       0.00142587
-20     0       18      -0.000819963    0.00163997      3.44637e-20     -1.8783e-19     0.000655042     -0.00190248
-20     0       19      0.00124197      -0.000148421    -6.64921e-20    -1.837e-20      -0.00188252     0.000315984
-20     0       20      -0.000837551    -0.000498656    3.19755e-20     -7.7927e-20     0.00125768      0.00142021
-20     0       21      0.000403901     0.00107009      4.23516e-22     1.04026e-19     0.00056594      -0.00191591
```

Figure 6: ANGPSS output file example

4 Conclusion

In summary two new software moduls for scattering and wave propagation phenomena in complex materials were developed. The integrability of these moduls into CIVA's new plugin mechanism was demonstrated and showed the power in the scripting approach. The ongoing evalution of the individual software components currently exhibits some weaknesses as to quantitative results and handling, which don't compromise their principal operability, though. Moreover, they are capable of extending the scope of CIVA application scenarios considerably and individually. Their actual contributions and usability guidelines will be addressed in Del1.5.

5 Bibliography

[1] S. Leberre, "CIVA PYTHON PLUGIN DEVELOPMENT", CIVA_plugin_training_ADVISE.pdf (internal document)

[2] M. Weikert-Müller, "DEL1.2, Analytical scattering theory"

[3] D. Weber, "Green's function and generalized point source synthesis in transversely-isotropic media"