CHALMERS

Development of a Statistical Energy Analysis implementation with an emphasis on composite laminates

Master's Thesis in the Master's programme in Sound and Vibration

FREDERIK RIETDIJK

Department of Civil and Environmental Engineering
Division of Applied Acoustics
Vibroacoustics Group
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2013
Master's Thesis 2013:81

MASTER'S THESIS 2013:81

Development of a Statistical Energy Analysis implementation with an emphasis on composite laminates

Frederik Rietdijk

Department of Civil and Environmental Engineering
Division of Applied Acoustics
Vibroacoustics Group
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2013

Development of a Statistical Energy Analysis implementation with an emphasis on composite laminates

Master's Thesis 2013:81

Department of Civil and Environmental Engineering
Division of Applied Acoustics
Vibroacoustics Group
Chalmers University of Technology
SE-41296 Göteborg
Sweden

Tel. +46-(0)31 772 1000

Reproservice / Department of Civil and Environmental Engineering
Göteborg, Sweden 2013

Development of a Statistical Energy Analysis implementation with an emphasis on composite laminates

Master's Thesis in the Master's programme in Sound and Vibration
Frederik Rietdijk

Chalmers University of Technology
Department of Civil and Environmental Engineering
Division of Applied Acoustics
Vibroacoustics Group

Abstract

In structures like for instance an aircraft we would like to predict the sound pressure levels in the cabin due to the engines or aeroacoustically induced vibrations. A nowadays common method within the aerospace and automotive industries is Statistical Energy Analysis (SEA).

Many structures nowadays are (partially) composed of composite materials. Composites are used basically everywhere, and for different purposes. An example would be to improve the damping of a structure by adding a viscoelastic layer to the material of the structure.

Several models exist for the calculation of elastic moduli of composite laminates, and one such model is Classic Laminate Theory. An extension to Classic Laminate Theory makes it possible to calculate the frequency-dependent loss factor of a laminate as well.

Inspired by SEALAB the goal of this project is to write a free and open-source SEA implementation. Additionally the implementation should support the calculation of elastic properties of composite lamina using Classic Laminate Theory and the mentioned extension.

In this report the design of the SEA implementation will be discussed. The successful validation of the composites model is shown as welll. Several severe problems were encountered while writing the SEA implementation. The main issues are discussed and solutions are proposed. In its current state the SEA implementation cannot be used.

Keywords: Statistical Energy Analysis, SEA, Composite Materials, Laminates

Contents

Notations

Roman uppercase letters

A	Area	m^2
B	Bending stiffness	$\mathrm{N\,m}^2$
C	Conductivity	rad^{-1}
C	Stiffness	Pa
D	Flexural rigidity	$\mathrm{N\,m}^2$
E	Energy	J
E	Young's modulus	Pa
G	Shear modulus	Pa
H	Height	m
K	Bulk modulus	Pa
L	Length	m
M	Modal overlap	rad^{-1}
M	Moment resultant	N
N	Force resultant or traction	$\mathrm{N\,m}^{-1}$
N	Mode count	rad^{-1}
P	Power	W
P	Perimeter	m
Q	Reduced stiffness	Pa
S	Compliance	Pa^{-1}
S	Cross-section	m^2
T	Transformation matrix	-

Roman lowercase letters

c	Wave velocity	$\mathrm{m\,s^{-1}}$
c_ϕ	Phase velocity	$\mathrm{m\,s^{-1}}$
c_g	Group velocity	$\mathrm{m\,s^{-1}}$
e	Modal power	W rad
f	Frequency	Hz
j	Imaginary unit	-
k	Wavenumber	$\mathrm{rad\,m^{-1}}$
m	Mass	kg
m	Component of θ	-
n	Component of θ	-
n	Modal density	$\mathrm{s\,rad^{-1}}$
p	Stiffness or elastic modulus	Pa
t	Time	s
x	Direction	m
y	Direction	m
z	Direction	m

Greek uppercase letters

Γ	Correction term	-

Greek lowercase letters

δ	Correction term	-
ϵ	Strain	-
η	Displacement	m
η	Loss factor	$\mathrm{rad^{-1}}$
λ Lambda	m	
ω	Angular frequency	$\mathrm{rad\,s^{-1}}$
σ	Stress	Pa
ν	Poisson's ratio	-
γ	Engineering shear strain	-
ρ	Density	$\mathrm{kg\,m^{-3}}$
τ	Stress when engineering shear strain is used	Pa
θ	Fibre orientation angle of layer	rad

Acronyms

API	Application Programming Interface
CAD	Computer-Aided Design
CLF	Coupling Loss Factor
CPT	Classic Plate Theory
CLT	Classic Laminate Theory
GUI	Graphical User Interface
MVA	Model-View-Adapter
MVC	Model-View-Controller
SEA	Statistical Energy Analysis

1 Introduction

In structures like for instance an aircraft we would like to predict the sound pressure levels in the cabin due to the engines or aeroacoustically induced vibrations. Several techniques exist for predicting the response of a structure to an excitation. A nowadays common method within the aerospace and automotive industries is Statistical Energy Analysis (SEA). SEA consists of procedures for predicting reverberant motion in structures [17]. The dynamical problem is reduced to a set of linear equations relating energetic variables associated with subsystems of the complete structure. Due to this description SEA can be a relatively quick technique for predicting the response of a structure.

The amount of vibrational energy transferred depends on the conductivities and modal overlap factors of the couplings and components along the transfer path as well as their 'acoustic temperatures' [20]. The conductivities and modal overlap factors strongly depend on the losses per cycle in the coupling and the component, values known as respectively the coupling loss factor and damping loss factor. A good estimation of these loss factors is therefore important.

Many structures nowadays are (partially) composed of composite materials. Composites are used basically everywhere, ranging from buildings to gears to the hull of an aircraft, and they can be applied for different purposes. An example would be to improve the damping of a structure. In such case a viscoelastic material is added to the structure.

Several models exist for the calculation of elastic moduli of composite laminates, and one such model is Classic Laminate Theory. An extension to CLT makes it possible to calculate the frequency-dependent loss factor of a laminate.

Inspired by SEALAB [14] the goal of this project is to write a free and open-source SEA implementation. Additionally the implementation should support the calculation of elastic properties of composite laminae. In this report the theory of SEA will briefly be treated as well as a model for the prediction of elastic properties of laminae. The technical design of the SEA implementation will be discussed and a validation will be shown of the composites model. While the focus of the project was on developing an SEA implementation, the design and actual implementation will be discussed only briefly. Instead, the focus in the report is on a model for composite laminae, though limited time and effort was actually spend on this topic.

2 Statistical Energy Analysis

Statistical Energy Analysis (SEA) is a method for predicting reverberant motion in a system. SEA was originally developed during the Apollo program to facilitate predictions of acoustic fatigue occurring during launch [20]. Successful advances were made later enabling noise and vibrations predictions in buildings. Nowadays SEA is used more widely, including the automotive industry.

Statistical Energy Analysis is a method with an energy perspective. It is assumed the system is in steady state. The first part in a Statistical Energy Analysis is to subdivide the structure into subsystems. For each subsystem a power balance equation can be written. For subsystem i the power balance equation would be written as

$$P_{in}^{i} = P_{diss}^{i} + \sum_{j \neq i} P_{c}^{i,j} \tag{2.1}$$

where P_{in}^{i} is the input power to the subsystem from external sources, P_{diss}^{i} is the power dissipated through internal damping, and $P_{c}^{i,j}$ is the power transmitted from subsystem i to a neighbouring subsystem j through a mechanical coupling. All powers are time-averaged.

Power dissipation in a subsystem is given by

$$P_{diss}^{i} = \eta_i \omega E_i \tag{2.2}$$

where η_i is the damping loss factor, ω the angular frequency and E_i the time-averaged energy stored in subsystem i.

The coupling power is assumed to be

$$P_{c}^{i,j} = C^{i,j} \left(\frac{E_i}{n_i} - \frac{E_j}{n_j} \right) \tag{2.3}$$

where $C^{i,j}$ is the vibration conductivity and n_i and n_j the modal densities in respectively subsystems i and j. The fractions are sometimes also called acoustic temperatures. The modal density is the number of modes per unit angular frequency band

$$n = n(\omega) = \frac{\Delta N}{\Delta \omega} \tag{2.4}$$

The modal energy of subsystem i is obtained by dividing the energy stored in the subsystem with its modal density

$$\hat{e}_i = \frac{E_i}{n_i} \tag{2.5}$$

Combining this equation with 2.2 gives

$$P^i_{diss} = \eta_i \omega E_i = \eta_i \omega n_i \hat{e}_i = M_i \hat{e}_i \tag{2.6}$$

where M_i is the modal overlap

$$M_i = \eta_i \omega n_i \tag{2.7}$$

The modal overlap is the parameter that describes dissipation in subsystems of an SEA model. The other parameter describing dissipation in an SEA model is the conductivity. Often equation 2.3 is written as

$$P^{i,j}_c = \omega \left(\eta^{i,j}_c E_i - \eta^{j,i}_c E_j \right) \tag{2.8}$$

where $\eta^{i,j}_c$ and $\eta^{j,i}_c$ are coupling loss factors. The coupling loss factor $\eta^{i,j}_c$ describes the apparent loss factor of subsystem i, because of energy flow to subsystem j, when $\eta_j \gg \eta^{j,i}_c$.

Often the SEA consistency relation is valid

$$n_i \eta^{i,j}_c = n_j \eta^{j,i}_c \tag{2.9}$$

in which case

$$C_{i,j} = \eta^{i,j}_c \omega n_i = \eta^{j,i}_c \omega n_j = C_{j,i} \tag{2.10}$$

It can clearly be seen that the expression for the conductivity has a similar shape as that of the modal overlap.

By expressing the conservation of energy in modal powers, modal overlap factors and conductivities, it is possible to assemble the power balance equations of every subsystem in the following matrix form

$$\begin{bmatrix} M_1 + \sum_j C^{1j} & -C^{12} & \cdots & -C_{1n} \\ -C^{12} & M_2 + \sum_j C^{2j} & \cdots & -C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -C_{1n} & -C_{2n} & \cdots & M_n + \sum_j C^{nj} \end{bmatrix} \begin{Bmatrix} \hat{e}_1 \\ \hat{e}_2 \\ \vdots \\ \hat{e}_n \end{Bmatrix} = \begin{Bmatrix} P^1_{in} \\ P^2_{in} \\ \vdots \\ P^n_{in} \end{Bmatrix} \tag{2.11}$$

This formulation allows efficient solution. No numerical difficulties are involved since the matrix is real, symmetric and positive definite.

Next is to specify the modal overlap factors and conductivities for respectively the subsystems and the couplings. These factors (though often expressed as (coupling) loss factors or averaged frequency spacings [17]) can readily be found in literature and will not be repeated here.

Since the modal overlap factors and conductivities depend on the material properties of the structure we shall have a look at elastic material properties and composite laminates in the next two chapters.

3 Solid materials

Solid materials are characterised by their structural rigidity and their resistance to changes of shape. Not all solids behave similarly to for example an applied load; certain solids show linear elastic behaviour while other solids exhibit non-linear phenomenon like hysteresis. This section gives an overview of elastic solids, viscosity and composite materials.

3.1 Elastic materials

Elastic materials are characterised by their elasticity, which is the property to return to the original shape after deformation. When an external force applied to an elastic material would deform the material, then, unlike plastic materials, elastic materials restore to its original state when the external force is removed.

The elasticity of materials is described by a stress-strain curve, as shown in figure 3.1.

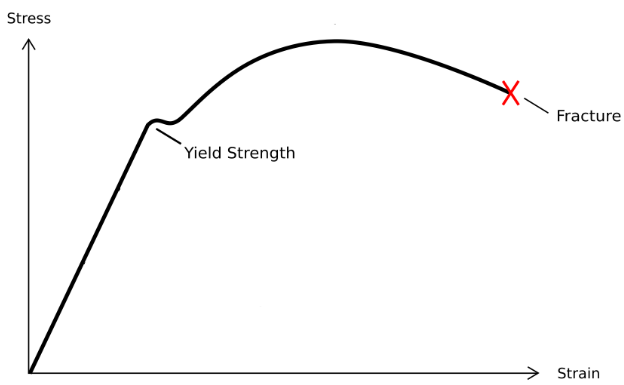

Figure 3.1: Stress as function of strain. Below the yield point this material appears to be linear elastic. Figure adapted from [5].

Below the yield point the material is said to be elastic while above the yield point it becomes non-elastic. If the material is elastic, and the curve is linear, then Hooke's law can be applied.

Different measures of stiffness as a resistance to deformation exist. These measures are known as elastic moduli and each of the moduli applies to a different kind of deformation.

Common moduli are:

- the Young's modulus E, which is defined as the ratio of the stress along an axis over the strain along that axis;
- the shear modulus G, which is defined as the ratio of shear stress to the shear strain;
- the Poisson's ratio ν, which defined as the negative ratio of transverse to axial strain
- the bulk modulus K, which is defined as the ratio of the infinitesimal pressure increase to the resulting relative decrease of the volume.

The first three moduli are mechanical quantities, while the fourth one is in fact a thermodynamic quantity.

Besides these common elastic moduli the flexural rigidity (often denoted D) or bending stiffness B is a common measure of the stiffness of a structure. Stiffness is the rigidity of an object and describes to what extent it resists deformation. The type of stiffness depends on how the stress is caused, and in which direction the strain is measured. It should be emphasised that stiffness is not the same as elastic modulus; an elastic modulus is a material property while a stiffness is a structural property.

Many physical properties of materials are direction-dependent because of the arrangement of the atoms in the crystal lattice. If a material has direction dependent properties, it is called anisotropic. Figure 3.2 illustrates a cubic material with forces acting on it. By dividing the forces with the surface area over which the forces are acting, the stresses on the cube can be obtained. A stress state can be decomposed into nine components. Tensors are used to describe the direction dependency of the direction-dependent physical properties. Stress σ and strain ϵ can be described using the following second-order tensors

$$\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \tag{3.1}$$

$$\epsilon = \begin{bmatrix} \epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{21} & \epsilon_{22} & \epsilon_{23} \\ \epsilon_{31} & \epsilon_{32} & \epsilon_{33} \end{bmatrix} \tag{3.2}$$

The stress tensor is a second-order tensor since it is a quantity associated with two directions. The first number of the subscript indicates the direction of the surface normal upon which the stress acts. The second number indicates the direction of the stress component. The diagonal values represent normal stresses while the values off the diagonal represent tangential or shear stresses. The stress tensor is symmetric as a result of static equilibrium, so $\sigma_{21} = \sigma_{12}$, $\sigma_{31} = \sigma_{13}$, $\sigma_{32} = \sigma_{23}$, and thus describes six stress values. Like the stress tensor a strain tensor is a second-order tensor as well. The strain tensor is also symmetric, representing six strain measures.

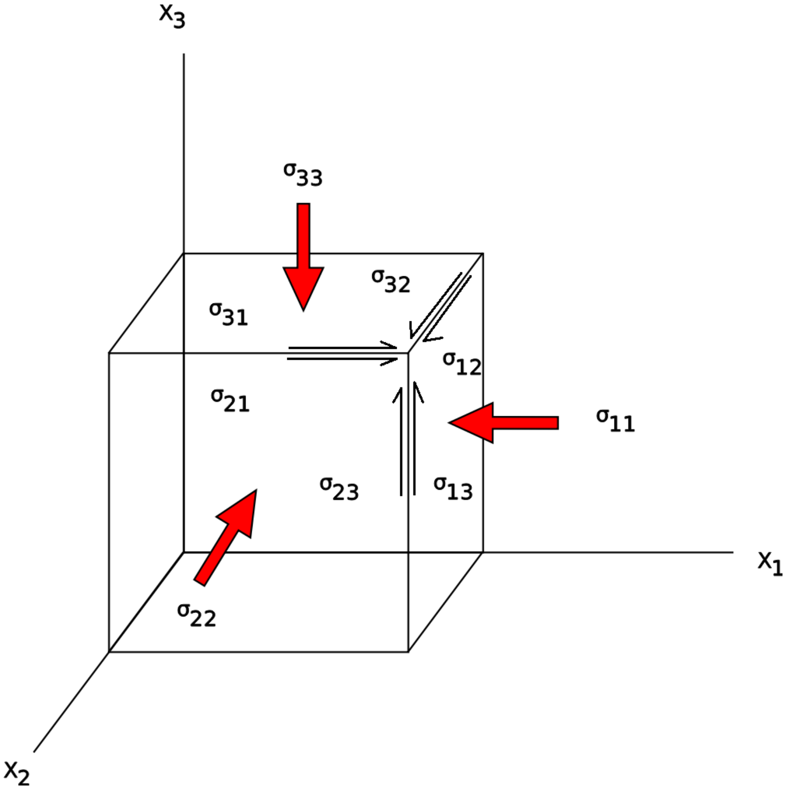

Figure 3.2: Stress directions.

3.2 Linear elastic materials

When Hooke's law applies, a material is linear elastic, and is called Hookean. Hooke's law for elastic materials can be described as

$$\sigma = \mathbf{C}\epsilon \tag{3.3}$$

or

$$\epsilon = \mathbf{S}\sigma \tag{3.4}$$

where σ is the stress, ϵ the strain, $\mathbf{C}$ the stiffness and $\mathbf{S}$ the compliance.

In order to relate the stress and strain, both second-order tensors, a fourth-order tensor is required, describing in total 81 terms. Since both stress and strain are symmetric tensors only 36 terms have to be described. Matrix notation is used instead of tensor notation. Table 3.1 shows the mapping.

Tensor subscript	11	22	33	23,32	13,31	12,21
Matrix subscript	1	2	3	4	5	6

Table 3.1: A mapping was created to write the stress and strain tensors as vectors and the stiffness or compliance as a matrix. Examples of the mapping are $C_{1111} = C_{11}$, $C_{1122} = C_{12}$, $C_{1123} = C_{14}$, $C_{1112} = C_{16}$.

Furthermore, the notation of the stress and strain vectors are contracted. Table 3.2 shows the contracted notation of the stress vector.

Tensor subscript	σ_{11}	σ_{22}	σ_{33}	σ_{23}	σ_{31}	σ_{12}
Contracted subscript	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6

Table 3.2: Tensorial subscript and contracted subscript of the stress vector. Contraction is done similarly for the strain vector.

Hooke's law can then be written in compliance form as

$$\begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} & S_{15} & S_{16} \\ S_{21} & S_{22} & S_{23} & S_{24} & S_{25} & S_{26} \\ S_{31} & S_{32} & S_{33} & S_{34} & S_{35} & S_{36} \\ S_{41} & S_{42} & S_{43} & S_{44} & S_{45} & S_{46} \\ S_{51} & S_{52} & S_{53} & S_{54} & S_{55} & S_{56} \\ S_{61} & S_{62} & S_{63} & S_{64} & S_{65} & S_{66} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{Bmatrix} \tag{3.5}$$

Due to further symmetries only 21 material constants are required for an anisotropic material, resulting in the following symmetric compliance and stiffness matrices

$$\mathbf{S}_{anisotropic} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} & S_{15} & S_{16} \\ S_{12} & S_{22} & S_{23} & S_{24} & S_{25} & S_{26} \\ S_{13} & S_{23} & S_{33} & S_{34} & S_{35} & S_{36} \\ S_{14} & S_{24} & S_{34} & S_{44} & S_{45} & S_{46} \\ S_{15} & S_{25} & S_{35} & S_{45} & S_{55} & S_{56} \\ S_{16} & S_{26} & S_{36} & S_{46} & S_{56} & S_{66} \end{bmatrix} \tag{3.6}$$

$$\mathbf{C}_{anisotropic} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{12} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{13} & C_{23} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{14} & C_{24} & C_{34} & C_{44} & C_{45} & C_{46} \\ C_{15} & C_{25} & C_{35} & C_{45} & C_{55} & C_{56} \\ C_{16} & C_{26} & C_{36} & C_{46} & C_{56} & C_{66} \end{bmatrix} \tag{3.7}$$

When other symmetries are present the amount of required material constants drops further.

3.2.1 Orthotropic

Orthotropic materials have elastic moduli which are different along only three perpendicular directions. Due to this symmetry only nine constants are required to describe an orthotropic material. The compliance matrix of an orthotropic material is given by

$$\mathbf{S}_{orthotropic} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & 0 & 0 & 0 \\ S_{12} & S_{22} & S_{23} & 0 & 0 & 0 \\ S_{13} & S_{23} & S_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & S_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & S_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & S_{66} \end{bmatrix} \tag{3.8}$$

and can be expressed in terms of elastic moduli

$$\mathbf{S}_{orthotropic} = \begin{bmatrix} \frac{1}{E_1} & -\frac{\nu_{21}}{E_2} & -\frac{\nu_{31}}{E_3} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & -\frac{\nu_{32}}{E_3} & 0 & 0 & 0 \\ -\frac{\nu_{13}}{E_1} & -\frac{\nu_{23}}{E_2} & \frac{1}{E_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2G_{23}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2G_{31}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2G_{12}} \end{bmatrix} \tag{3.9}$$

where ν_{ij}, E_i and G_{ij} are respectively Poisson's ratio, Young's moduli and shear moduli.

3.2.2 Isotropic

Isotropic materials require only two independent material constants. Most metallic alloys and certain types of polymers are considered isotropic, where by definition the material properties are independent of direction. The stiffness matrix of an isotropic material is given by

$$\mathbf{C}_{isotropic} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \tag{3.10}$$

The stiffness and compliance are usually expressed in terms of the Young's modulus E and the Poisson's ratio ν.

$$\mathbf{C}_{isotropic} = \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-2\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-2\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1-2\nu \end{bmatrix} \cdot \frac{E}{(1+\nu)(1-2\nu)} \tag{3.11}$$

3.3 Viscoelasticity

Certain materials, like polymers, exhibit both viscous and elastic characteristics when undergoing deformation. The viscosity of a viscoelastic material gives such a material a strain-rate dependent on time. Ideal elastic materials do not dissipate energy when a load is sequentially applied and removed. Viscoelastic materials however lose energy when a load is temporarily applied. This can be seen on a stress-strain curve which shows hysteresis, where the area of the loop represents the energy lost during one loading cycle.

In a linear viscoelastic material the stress and strain time-dependency can be written as

$$\sigma_{ij}^* = \sigma_{ij} e^{j\omega t} \tag{3.12}$$

$$\epsilon_{ij}^* = \epsilon_{ij} e^{j\omega t} \tag{3.13}$$

Several models exist for incorporating damping losses. A common method in dynamic analysis is to use a complex stiffness or modulus p^*

$$p^* = p' + jp'' = p\,(1 + j\eta) \tag{3.14}$$

where η is the damping loss factor, defined as the ratio between lost energy per period $P_{loss}$and reversible energy E_{rev}

$$\eta = \frac{P_{loss}}{2\pi E_{rev}} \tag{3.15}$$

The real part p' of the complex modulus p^* is called the storage modules and represents the elasticity whereas the imaginary part p'' is called the loss modulus and represents viscosity and losses through heat.

Viscoelastic materials are often used to add damping to a structure. A common way of doing so is by adding damping layers to a stiff material, resulting in a composite material.

3.4 Composite materials

Composite materials are materials consisting of two or more distinct materials. The composition can be done at microscopic and macroscopic, which means the composites consist of respectively mixed fibres or several joined layers.

The materials that a composite consists of are divided into two groups, matrix and reinforcement. Matrix materials are used to maintain the relative positions of reinforcement materials. Reinforcement materials are there to enhance the matrix properties by their special properties.

Focus within this thesis is on composite laminate. Composite laminates are structures consisting of several layers. An example of composite laminates are sandwich layers. Figure 3.3 shows a picture of a sandwich structure.

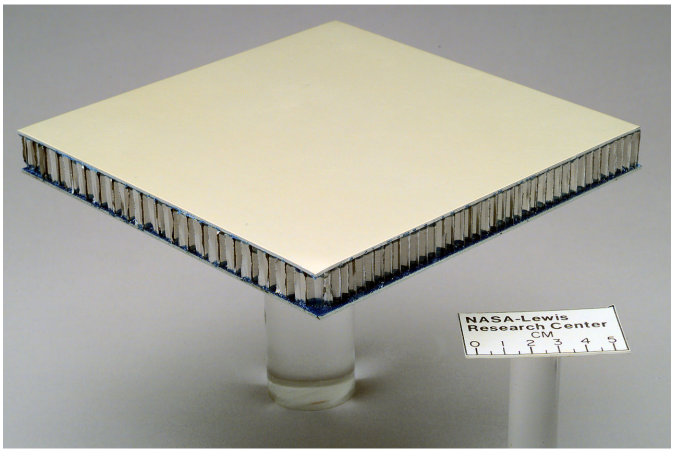

Figure 3.3: Glass Reinforced Aluminum (GLARE) honeycomb composite sandwich structure. Figure taken from [1].

3.4.1 Modelling elastic properties of composites

Different approaches have been used to determine the elastic properties of composites. These approaches can be grouped into micromechanical, macromechanical and structural approaches. Whereas micromechanical methods try to predict material properties based on their microstructure, macromechanical methods are mainly based on continuum mechanics, assuming that materials can be seen as a continuous mass rather than as discrete particles. Predictions on composites are done through homogenisation, i.e., the material is modelled as consisting of several homogeneous materials with a given geometry. Structural approaches take the composite as the fundamental building block and thus offer no possibilities for predicting the properties of the composite.

3.4.2 Plate theory applied to composites

Macromechanical approaches are generally based on beam- and plate-theory. Two well-known plate-theories are Kirchhoff-Love theory and Mindlin-Reissner theory. The Kirchhoff–Love theory of plates, also known as Classic Plate Theory (CPT), is an extension of Euler-Bernoulli beam theory to two dimensions, and can be applied to thin plates. The Mindlin-Reissner theory is an extension of Kirchhoff–Love plate theory that takes into account first-order shear deformations through-the-thickness of a plate and can be applied to thick plates. Depending on the thickness relative to the planar dimensions of the plate the right theory should be chosen.

Based on these two theories are models for predicting the elastic moduli of composite laminates. Classic Laminate Theory (CLT) is based on the former theory. CLT has a range of applicability despite the fact that it suffers from a major deficiency associated with the transverse behaviour of laminates. Crane and Gillespie then extended CLT to model complex moduli [11]. Ghinet presented a model based on Mindlin-Reissner theory [13] for the prediction of complex moduli.

4 Composite laminae

We shall now have a look an analytic, macromechanical model presented by Crane and Gillespie in 1992 [11]. This model is based on the elastic viscoelastic correspondence principle and is an extension to Classic Laminate Theory. It should be emphasised that this model applies only to thin composite laminates as it is .

The correspondence principle states that if the elastic solution for any dependent variable having a time-varying component exists, then the viscoelastic problem can be solved by replacing the equations of the elastic material by the equations that describe the viscoelastic material. The model is capable of predicting the elastic moduli and their respective loss factors of composite laminates.

Several assumptions were made in the model. The first assumption is that the composite can be approximated as a homogeneous orthotropic material, reducing the number of elastic constants required to nine, as explained in 3.2.1. The second assumption is that the material is in a state of plane stress, where the stresses normal to the plane of the plate are assumed to be zero. Composite laminae are used in structural shapes like beams and plates which have at least one characteristic geometric dimension an order of magnitude less than the other two dimensions. The stresses in the direction of this one characteristic dimension are generally also much smaller and can be considered to be zero. The third assumption made is that the composite is linearly viscoelastic, adding a time-dependency as shown in section 3.3. Note that in this chapter $e^{j\omega t}$ is omitted.

4.1 Stress-strain relations

Following the previously mentioned assumptions the stress-strain relations become

$$\begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & 0 & 0 & 0 \\ S_{12} & S_{22} & S_{23} & 0 & 0 & 0 \\ S_{13} & S_{23} & S_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & S_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & S_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & S_{66} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ 0 \\ 0 \\ 0 \\ \tau_{12} \end{Bmatrix} \tag{4.1}$$

Note that engineering shear strain[1] instead of shear strain is used from here on. Since three equations result in zero these can now be removed for the system, resulting in the reduced system of equations.

$$\begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & 0 \\ S_{12} & S_{22} & 0 \\ 0 & 0 & S_{66} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} \tag{4.2}$$

While the perpendicular stresses are assumed to be zero, it does not yet mean that there is no strain in the 3-direction. They are simply left out of the analysis, though strain does occur since

$$\epsilon_3 = S_{13}\sigma_1 + S_{23}\sigma_2 \tag{4.3}$$

The 3x3 matrix of compliances is called the reduced compliance matrix. Writing in inverse form the reduced stiffnesses are obtained

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \gamma_{12} \end{Bmatrix} = \mathbf{Q} \begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \gamma_{12} \end{Bmatrix} \tag{4.4}$$

The reduced stiffnesses can be written in the form of a reduced stiffness matrix $\mathbf{Q}$. For orthotropic layers the reduced stiffnesses are

$$Q_{11} = C_{11} - \frac{C_{13}^2}{C_{33}} = \frac{E_1}{1-\nu_{12}\nu_{21}} \tag{4.5}$$

$$Q_{12} = C_{12} - \frac{C_{13}C_{23}}{C_{33}} = \frac{\nu_{12}E_1}{1-\nu_{12}\nu_{21}} = \frac{\nu_{21}E_1}{1-\nu_{12}\nu_{21}} \tag{4.6}$$

$$Q_{22} = C_{22} - \frac{C_{23}^2}{C_{33}} = \frac{E_2}{1-\nu_{12}\nu_{21}} \tag{4.7}$$

$$Q_{66} = C_{66} = G_{12} \tag{4.8}$$

and for isotropic layers the reduced stiffnesses are

$$Q_{11} = \frac{E}{1-\nu^2} \tag{4.9}$$

$$Q_{12} = \frac{\nu E}{1-\nu^2} \tag{4.10}$$

$$Q_{22} = Q_{11} \tag{4.11}$$

$$Q_{66} = G = \frac{E}{2\,(1+\nu)} \tag{4.12}$$

[1]Engineering shear strain γ_{xy} is a total measure of shear strain in the xy-plane. Shear strain ϵ_{xy} is the average of the shear strain on the x face along the y-direction, and on the y face along the x-direction. They can be related through $\gamma_{xy} = 2\epsilon_{xy}$. When engineering shear strain is used the stress is written as τ_{xy} though $\tau_{xy} = \sigma_{xy}$.

4.2 Coordinate system transformation

Usually, the coordinate system used to analyse a structure is based on the shape of the structure rather than the direction of the fibres of a particular lamina. Therefore it is necessary to perform a coordinate transformation. A transformation matrix $\mathbf{T}$ is defined as

$$\mathbf{T} = \begin{bmatrix} m^2 & n^2 & 2mn \\ n^2 & m^2 & -2mn \\ -mn & mn & m^2 - n^2 \end{bmatrix} \tag{4.13}$$

with

$$m = \cos\theta \tag{4.14}$$

$$n = \sin\theta \tag{4.15}$$

where θ is the fibre orientation angle of the layer, relative to a chosen reference which is identical for all layers. The transformation has to be done for both the stress and strain vectors.

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} = \mathbf{T}^{-1}\mathbf{S}\mathbf{T} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \overline{\mathbf{S}} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} \overline{S}_{11} & \overline{S}_{12} & \overline{S}_{16} \\ \overline{S}_{12} & \overline{S}_{22} & \overline{S}_{26} \\ \overline{S}_{16} & \overline{S}_{26} & \overline{S}_{66} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \tag{4.16}$$

where $\overline{\mathbf{S}}$ is the transformed reduced compliance matrix. The transformed reduced stiffness matrix of a layer is given by

$$\overline{\mathbf{Q}} = \mathbf{T}^{-1}\mathbf{Q}\mathbf{T} \tag{4.17}$$

and consists of nine transformed reduced stiffnesses

$$\overline{\mathbf{Q}} = \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{bmatrix} \tag{4.18}$$

4.3 Forces and moments

In order to continue it is necessary to define the forces and moments that act on the composite. The force resultants or tractions are defined as

$$N_x = \int_{-\frac{H}{2}}^{\frac{H}{2}} \sigma_x \mathrm{d}z \tag{4.19}$$

$$N_y = \int_{-\frac{H}{2}}^{\frac{H}{2}} \sigma_x \mathrm{d}z \tag{4.20}$$

$$N_{xy} = \int_{-\frac{H}{2}}^{\frac{H}{2}} \tau_{xy} \mathrm{d}z \tag{4.21}$$

where H is the laminate thickness and z the position in z-direction. The resultants N_x and N_y are normal force resultants and N_{xy} is a shear force resultant. The moment resultants are

$$M_x = \int_{-\frac{H}{2}}^{\frac{H}{2}} \sigma_x z \mathrm{d}z \tag{4.22}$$

$$M_y = \int_{-\frac{H}{2}}^{\frac{H}{2}} \sigma_x z \mathrm{d}z \tag{4.23}$$

$$M_{xy} = \int_{-\frac{H}{2}}^{\frac{H}{2}} \tau_{xy} z \mathrm{d}z \tag{4.24}$$

where M_x and M_y are bending moment resultants and M_{xy} is a twisting moment resultant. Stress and strain were related through the transformed reduces stiffnesses as

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} \tag{4.25}$$

The total strain can be written in terms of extensional strain ϵ_i^0 and curvature κ_i^0 of the reference surface

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_x^0 + z\kappa_x^0 \\ \epsilon_y^0 + z\kappa_y^0 \\ \gamma_{xy}^0 + z\kappa_{xy}^0 \end{Bmatrix} \tag{4.26}$$

Combining the definition of the force resultants with the stress-strain relations, one obtains the following relation for the force resultant N_x

$$N_x = \int_{-\frac{H}{2}}^{\frac{H}{2}} \left[\overline{Q}_{11}\left(\epsilon_x^0 + z\kappa_x^0\right) + \overline{Q}_{12}\left(\epsilon_y^0 + z\kappa_y^0\right) + \overline{Q}_{16}\left(\gamma_{xy}^0 + z\kappa_{xy}^0\right)\right] \mathrm{d}z \tag{4.27}$$

The strains and curvatures can be taken outside the integral since they are not functions of position z. Simplifications can be made by considering each integral separately. The first integral in the previous equation for example is given as

$$\epsilon_x^0 \int_{-\frac{H}{2}}^{\frac{H}{2}} \overline{Q}_{11} \mathrm{d}z \tag{4.28}$$

The reduced stiffnesses are material properties that vary from layer to layer but are constant within any given layer. Since the reduced stiffnesses are piece-wise constant, the integral can be expanded through the thickness resulting in the following summation

$$\int_{-\frac{H}{2}}^{\frac{H}{2}} \overline{Q}_{11} \mathrm{d}z = \sum_{k=1}^{N} \overline{Q}_{11_k} \left(z_k - z_{k-1}\right) \tag{4.29}$$

where k represents the layer number and z_k represents the height of the composite up to that layer. This value is typically denoted as A_{11}.

4.4 ABD-matrix

Continuing this process for the moment resultants, and by rewriting the other integrals the following general summations are obtained

$$A_{ij} = \sum_{k=1}^{N} \overline{Q}_{ij_k} \left(z_k - z_{k-1}\right) \tag{4.30}$$

$$B_{ij} = {}^1\!/_2 \sum_{k=1}^{N} \overline{Q}_{ij_k} \left(z_k^2 - z_{k-1}^2\right) \tag{4.31}$$

$$D_{ij} = {}^1\!/_3 \sum_{k=1}^{N} \overline{Q}_{ij_k} \left(z_k^3 - z_{k-1}^3\right) \tag{4.32}$$

Clearly the order of layers has no influence on the A_{ij} terms. By writing the forces and moments in terms of these A_{ij}, B_{ij} and D_{ij} terms the ABD-matrix is obtained

$$\left\{\begin{array}{c} N_x \\ N_y \\ N_{xy} \\ \hline M_x \\ M_y \\ M_{xy} \end{array}\right\} = \left[\begin{array}{ccc|ccc} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ \hline B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{array}\right] \left\{\begin{array}{c} \epsilon_x^0 \\ \epsilon_y^0 \\ \gamma_{xy}^0 \\ \hline \kappa_x \\ \kappa_y \\ \kappa_z \end{array}\right\} \tag{4.33}$$

Each of the 3x3 **A**, **B** and **D** matrices has a distinct function:

- **A** is a extensional stiffness matrix,
- **B** is a extension-bending coupling matrix,
- **D** is a bending stiffness matrix.

4.5 Elastic moduli and stiffnesses

The ABD-matrix relates the forces and moments to the strains in the laminate. The next step is to determine the elastic moduli and loss factors. The effective moduli for a composite can be obtained from the inverse ABD matrix

$$\left[\begin{array}{ccc|ccc} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ \hline B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{array}\right]^{-1} \tag{4.34}$$

The ABD inverse matrix takes into account the stress couplings that may occur from the various orientations of the fibres used in the laminate. Using the ABD inverse matrix elastic moduli and loss factors can be calculated. The calculation of several of these values will now be shown.

The complex Young's modulus in x-direction is given by

$$E_x = \frac{1}{hA_{11}^{-1}} \tag{4.35}$$

where A_{11}^{-1} is the element A_{11} from the inverse ABD-matrix. Damping losses were included through a complex stiffness in equation 3.14 The loss factor along x-direction is then given by

$$\eta_x = \frac{A_{11}''^{-1}}{A_{11}'^{-1}} \tag{4.36}$$

The complex Young's modulus in y-direction is given by

$$E_y = \frac{1}{hA_{22}^{-1}} \tag{4.37}$$

and the loss factor along the same orientation by

$$\eta_y = \frac{A_{22}''^{-1}}{A_{22}'^{-1}} \tag{4.38}$$

The Poisson's ratio in xy-direction is given by

$$\nu_{xy} = -\frac{A_{12}^{-1}}{A_{11}^{-1}} \tag{4.39}$$

and the Poisson's ratio in yx-direction is given by

$$\nu_{yx} = -\frac{A_{12}^{-1}}{A_{22}^{-1}} \tag{4.40}$$

As mentioned before the matrix $\mathbf{D}$ is also called the bending stiffness matrix. The complex bending stiffness in x-direction is given by

$$B_x = \frac{12}{h^3 D_{11}^{-1}} \tag{4.41}$$

and the loss factor along the same orientation by

$$\eta_x = \frac{D_{11}^{\prime\prime -1}}{D_{11}^{\prime -1}} \tag{4.42}$$

5 Modelling composites in SEA

Chapter 2 gave a brief overview of Statistical Energy Analysis and showed that the power balance equations depend on the modal overlap of the subsystems considered. The modal overlap depends on structural and material properties of these subsystems. The previous chapter showed how elastic moduli or stiffnesses and loss factors can be calculated for composite laminates. In this chapter it is shown how composite laminates can be modelled in an SEA system. One goal is to obtain an expression for the modal overlap M of a subsystem describing a composite laminate.

5.1 General result for a plate

Consider a composite laminate consisting of several layers as was shown in figure 6.4. Depending on the dimensions in relation to the wavelengths this structure can be considered a 2-dimensional plate. The wavenumber of a mode on a plate is given by

$$k_{m,n} = \sqrt{\left[(m - \delta_1)\frac{\pi}{L_1}\right]^2 + \left[(n - \delta_2)\frac{\pi}{L_2}\right]^2} \tag{5.1}$$

where m and n are the mode numbers along the L_1 and L_2 edges and δ_1 and δ_2 are correction terms that depend on the boundary conditions at the edges. The amount of modes with a wavenumber less than a certain value of k is the mode count function

$$N(k)^{2D} \simeq \frac{Ak^2}{4\pi} + \Gamma_{BC}Pk \tag{5.2}$$

where $A = L_1L_2$ is the area of the plate, $P = 2(L_1 + L_2)$ the perimeter and Γ_{BC} a quantity that depends on the boundary conditions. Differentiating the mode count function N with respect to the angular frequency ω results in the modal density

$$n(\omega)^{2D} = \frac{\mathrm{d}N}{\mathrm{d}\omega} = \frac{\mathrm{d}N}{\mathrm{d}k}\frac{\mathrm{d}k}{\mathrm{d}\omega} = \frac{A\omega}{2\pi c_g c_\phi} + \Gamma^{'}_{BC}P \tag{5.3}$$

where c_g and c_ϕ are respectively group and phase velocities of the wave and $\Gamma^{'}_{BC}$ is a quantity depending on boundary conditions and wave type. For connected subsystems it is best to assume that this value is zero according to Lyon[17].

The modal overlap of the plate is given by [20]

$$M = \eta\omega n(\omega)^{2D} \tag{5.4}$$

where η is the damping loss factor of the subsystem. The expressions shown in this section are the general result. We shall now have a look at specific wave types in such a two-dimensional system.

5.2 Flexural waves

The wave equation for free bending waves on a plate according to Kirchhoff-Love theory is given by

$$B_x \frac{\partial^4 \eta}{\partial x^4} + 2B_{xy} \frac{\partial^2 \eta}{\partial x^2} \frac{\partial^2 \eta}{\partial y^2} + B_y \frac{\partial^4 \eta}{\partial y^4} + m \frac{\partial^2 \eta}{\partial t^2} = 0 \tag{5.5}$$

where B_x, B_y and B_{xy} are bending stiffness in respectively the x, y and xy-direction and m the mass of the plate. The displacement η of the plate field at a time t given by

$$\eta = \eta_A \left(e^{-jk_x x} + e^{-jk_y y} \right) e^{j\omega t} \tag{5.6}$$

Combining these relations results in the dispersion relation

$$k_i^4 = \frac{\omega^2 \rho S_i}{B_i} \tag{5.7}$$

where ρ is the density of the plate and S_i and B_i are respectively the cross-section and bending stiffness of the plate in the i-direction. Using this relation together with

$$f = \lambda c \tag{5.8}$$

results in the phase speed of flexural waves

$$c_{\phi_i} = \sqrt{\omega} \sqrt[4]{\frac{B_i}{\rho S_i}} \tag{5.9}$$

The group speed is given by

$$c_{g_i} = \frac{\mathrm{d}\omega}{\mathrm{d}k_i} = 2c_{\phi_i} \tag{5.10}$$

In Classic Laminate Theory and the model by Crane and Gillespie it was assumed that a composite laminate can be considered homogeneous and orthotropic. For flexural waves in orthotropic plates an approximation of the mode count can be obtained through the geometric mean of the wave speeds in the two principal directions. The modal density is then given by

$$n(\omega)^{2D} = \frac{A\omega}{2\pi \sqrt{c_{g_x} c_{g_y} c_{\phi_x} c_{\phi_y}}} \tag{5.11}$$

Lyon assumed the thickness was small compared to the wavelength, and that the plate was homogeneous. While the second assumption is not the case, this assumption is made as well in CLT. Both theories also make the plane-stress assumption.

The damping loss factor describes the losses per cycle. The total loss factor for flexural waves can also be obtained through the geometric mean of the loss factors

$$\eta_B = \sqrt{\eta_{B_x}} \sqrt{\eta_{B_y}} \tag{5.12}$$

The modal overlap of a subsystem representing bending waves in a thin composite laminate is then given by

$$M = \eta_B \omega n(\omega)^{2D} \tag{5.13}$$

5.3 Longitudinal and shear waves

The previous section showed how to calculate the modal overlap for bending waves. Calculating the modal overlap for longitudinal and shear waves in a composite laminate gets more troublesome. According to Bosmans [10] the equations of motion cannot be uncoupled for longitudinal and shear waves in an orthotropic material. This is because in a directions different from the principal material directions, compression or tension always induces in-plane shear. Because no expressions can be obtained for the wavenumber it is not possible to calculate the modal overlap of longitudinal and/or shear waves in a composite laminate using this theory.

5.4 Limitations

The above given method and equations can only be used when the wavelength is much larger than four times the thickness of the plate [17].

$$\lambda \gg 4H \tag{5.14}$$

When the thickness of a layer becomes relatively large compared to the wavelength the differences in wave speed between the layers becomes large and the layers start moving independently. In this case the layers should be modelled as an individual subsystem.

6 Software implementation

The goal of the thesis includes writing an easy to use and extend SEA implementation. The following chapter provides an overview of implementation motivation and requirements, implementation choices as well as the technical design. The actual implementation is discussed as well as future improvements.

6.1 Requirements

The following requirements to the implementation were set in advance:

- A low barrier for contributing to the implementation, and especially for those with specific knowledge/expertise.
- The physics model should be separated from the rest of the program logic for the above given reason among others. This makes it possible to implement a physics model describing for instance a subsystem, without knowing anything about how for example the results can be presented or the input data will be obtained.
- There should be a clear distinction between the different principal components of an SEA model.
- It should be as simple as possible to quickly implement models without having any deeper programming knowledge.
- Where possible the model should update values immediately upon a change, thus always showing the current values instead of old and possibly misleading data.
- The implementation should have access to a powerful geometry-kernel. This eases the process of designing the geometry and gives the possibility of detecting connections and retrieving their respective shapes

The next step was to conduct a study to possible tools for implementing this.

6.2 Motivation for new implementation

In 2010 Johansson and Comnell[14] wrote a free and open-source SEA implementation called SEALAB. A possibility would be to extend SEALAB. Instead of extending SEALAB the author deems a new implementation necessary because of several reasons.

SEALAB requires MATLAB[18] or GNU Octave[16]. Both languages are very specific languages i.e., they are meant to be used for mathematics. These two languages offer few constructs for general programming and are as such very limiting. The graphical user interface (GUI) is only available in MATLAB due to MATLAB-specific code and thus requiring a MATLAB license, limiting the availability.

SEALAB does not have access to a geometry-kernel. This means it is necessary to create all couplings manually. It also means that it is not possible to get a graphical overview of the structure that is modelled.

Finally, SEALAB is limited through its design. The implementation was written for specific cases lacking an abstract design.

6.3 Implementation choices

Writing a geometry-kernel is outside the scope of this project, as well as of the capabilities of the author, and therefore it was decided to utilise an existing geometry-kernel. A powerful, free and open-source geometry kernel can be found in Open CASADE [3]. Open CASCADE is a software development platform which includes components for 3D surface and solid modelling. Open CASCADE is mainly used in CAD programs and can also be found in finite element analysis software.

Open CASCADE is however found too low-level to work with straight-away. Since kernels like these are mainly used in CAD programmes, a short study on several free and open-source 3D CAD programmes was conducted, resulting in the decision to write a module for FreeCAD.

FreeCAD is a free and open source (LGPL license[8]) 3D CAD modeller [19]. FreeCAD's execution model determines which objects are affected by a change and updates them accordingly. This means FreeCAD's execution model supports updating/recomputing values directly when a changed, if this is wished. When a material density is changed one might want to see immediately how this influences the modal overlap, and thus the object responsible for calculating the modal overlap should be informed of the change. However, the modal powers should likely not be updated directly due to their computation time. The calculation of these will therefore be excluded from the execution model.

With FreeCAD it is possible to programmatically create a geometry, which can be used as a base for the SEA model. FreeCAD features a Python API[1] offering easy access to the model. It is also possible to extend FreeCAD through this API. Figure 6.1 shows a screenshot of FreeCAD.

Python is a general-purpose, high-level programming language whose design philosophy emphasises code readability [4]. Python's syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C. As a high-level programming language it features automatic memory management. Python has a large and comprehensive standard library. Many third-party modules are available as well, including NumPy [2] for scientific computing. NumPy is comparable in features and constructs with GNU Octave [16] and MATLAB [18].

As the syntax of Python and NumPy is comparable with that of GNU Octave and MATLAB, it is expected that the barrier for extending (and contributing to) the implementation is relatively low, since the language should not be an obstruction to a fast implementation of models.

A motivation has been given for the tools used to construct the implementation. The next section will be devoted to the technical design of the implementation.

[1] API is short for application programming interface. An API is a protocol intended to be used as an interface by software components to communicate with each other.

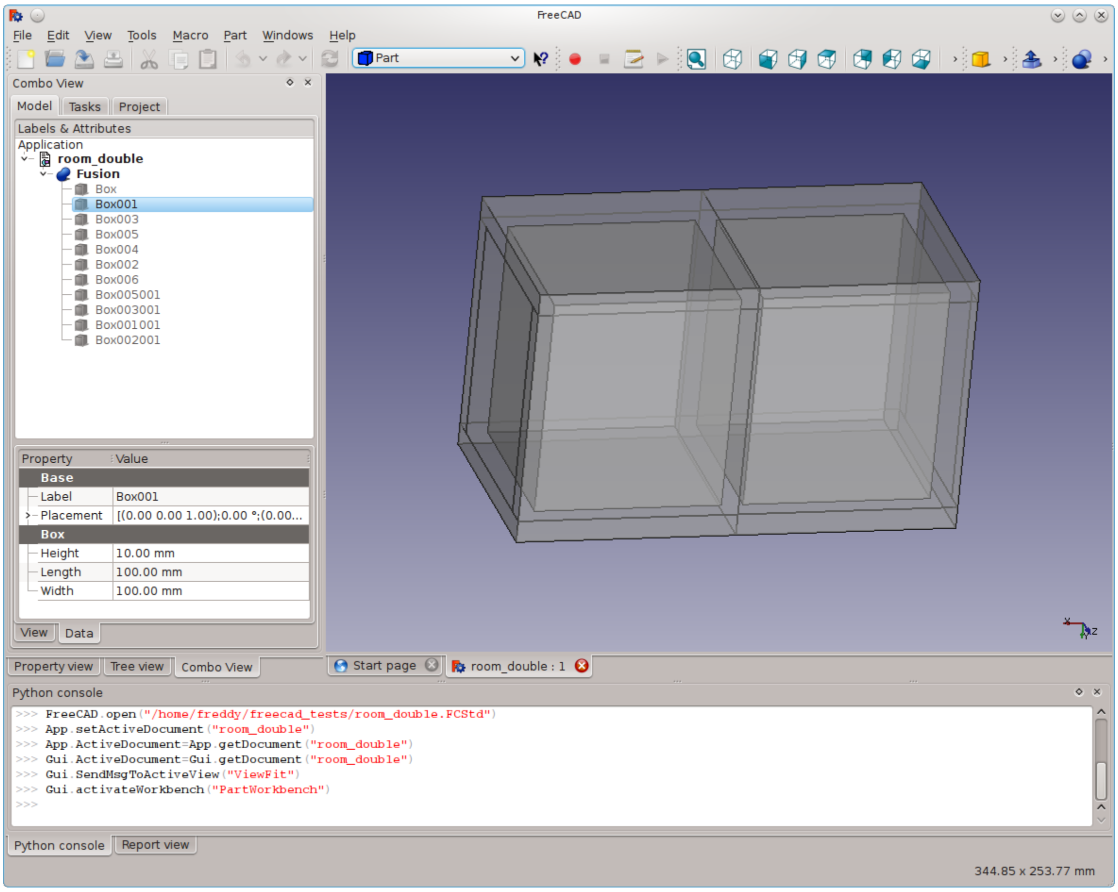

Figure 6.1: Screenshot of FreeCAD showing the geometry of two cavities enclosed by plates resembling two connected rooms. Left of the document window (showing the 3D-model) is the object tree and property editor. Below the document window a Python console can be found.

6.4 Technical design

Many different programming paradigms[2] exist of which Python support several. Since an SEA model can be described using several types of objects which all have their respective characteristic properties it was decided to use object-oriented programming as the main paradigm. In object-oriented programming objects interact with one another. These objects are instances of classes. Classes are a type of object which have attributes and methods, that is, functions that are part of and interact with the class.

[2] A programming paradigm is a fundamental style of computer programming.

6.4.1 Classes

In the implementation the different objects constituting an SEA model are represented by the following abstract base classes[3] written in *italic*:

- Static properties of a physical object are described by a *component*.
- Components can be described using a geometry and a *material*.
- Dynamic properties of a component are described in a *subsystem*.
- Two or more components are connected to each other through a *connection*.
- A pair of subsystems are connected through a *coupling*.

Using the descendants of these abstract base classes an SEA model can be created. The modal powers are obtained through solving a system of equations which is done by an instance of the *system* class. This class basically encompasses the entire SEA model. Figure 6.2 shows how the classes are related.

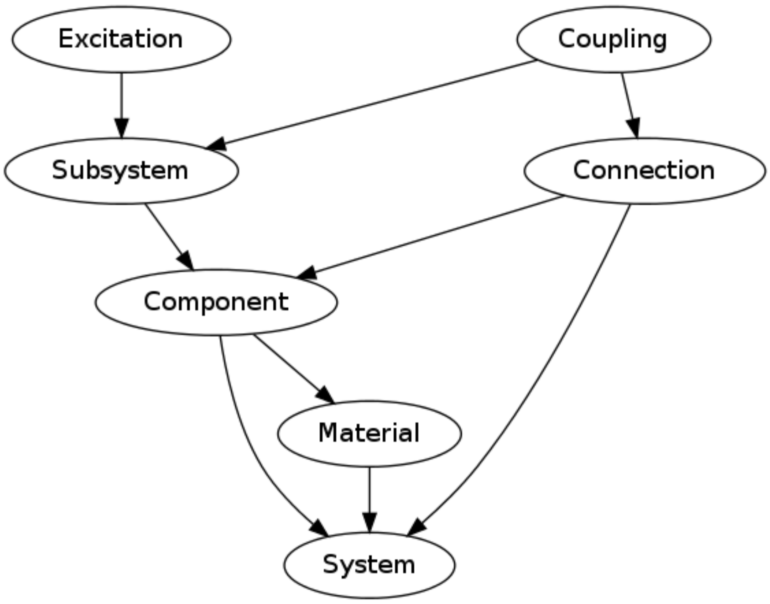

Figure 6.2: Directed acyclic graph (DAG) showing the relation between the classes. A DAG does not contain any directed cycles. FreeCAD's execution model requires a DAG, since if there would be such a cycle, the execution model could become recursive and get stuck in an infinite loop.

[3]Abstract classes are classes that cannot be instantiated themselves. Descendants of the abstract class that have implemented the attributes and methods required by the abstract class can be instantiated.

6.4.2 Architectural pattern

The Model-View-Adapter[4] (MVA) architectural pattern was adopted in order to separate the physics model. In MVA the Model and the View oblivious of each other and can only interact with each other through the Adapter. An example would be separating the physics model in a simulation (Model) from the user interface (View). An Adapter is created to mediate between the Model and the View, and potentially to allow for multiple views (a graphical user interface and a console interface).

In the implementation a clear separation exists between classes describing the physics, classes describing the GUI and classes that are mediating between the former two, the Adapter classes. Using such a separation results in cleaner code and allows the model and view to diverge from each other while not affecting one another. Because the model is isolated and doesn't contain anything besides equations describing the physics, it should also lower the barrier of implementing models for those with less programming expertise.

Figure 6.3: Communication between Model and View has to happen through the Adapter.

Adapter classes and Model classes shall exist for every object in figure 6.2. Instances of the Adapter classes point to each other using the directions of the arrows as shown in that figure.

The Modal and Adapter classes still need to connect to one another. In general two techniques exist for this, encapsulation and inheritance. Inheritance means that a descendent of a class takes over properties (attributes and methods) of the parent class. The Adapter class could for instance inherit the physics model from the Model class, and then extend upon that. Encapsulation can be achieved by assigning one class as attribute to the other class. For connecting the Adapter and Model classes encapsulation was chosen. Every Adapter class has a Model class as attribute. Encapsulation was chosen because in this way a clear separation remains between the Adapter and respective Model classes, without any risk of attribute/method name clashing.

6.4.3 Connection between SEA model and geometry

Adapter instances are objects capable of directly communicating with other FreeCAD objects. In FreeCAD a geometry can be designed with *part* objects. These *part* objects have a *shape* attribute describing the shape of the geometry using solids or shells.

[4] Model-View-Adapter (MVA) is also known as mediating-controller MVC where MVC stands for Model-View-Controller [9].

Through FreeCAD's linking system it is for example possible to link from a *component* Adapter instance to a *part* instance, thus connecting the SEA model to the geometry. Through a set of rules it should also be possible to create a certain type of *component* based on a certain *part*. FreeCAD gives the possibility to detect whether objects are connected. This gives the possibility of detecting a *connection* as well as determining its size.

6.4.4 Support of composites in implementation

Consider a laminate consisting of three layers as shown in figure 6.4. The geometry of such a laminate can be constructed in FreeCAD using three *part* objects.

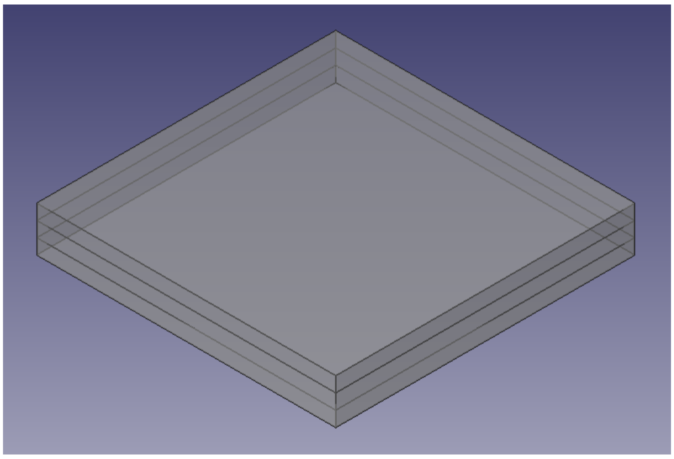

Figure 6.4: Composite laminate consisting of three layers.

Currently the *component* classes have a *material* attribute. The idea is to give the *part* classes a *material* attribute as well, and link the *component* material attribute to the *part* attribute. Using FreeCAD it is possible to unite objects into a *fusion*, while still keeping the original *part* objects with their respective solids. The *fusion* class is itself also a *part*, having a *material* attribute.

If each layer of a laminate would be represented as a *part* and if a *fusion* would represent the laminate, then by assigning the laminate a special *material* object capable of calculating the elastic moduli according to the model described in chapter 4, it would be possible to add support for composites.

6.4.5 Design conclusion

In this section the design of an SEA implementation has been described. The design meets all the requirements that were initially set. Based on this design the implementation is written.

6.5 Current implementation

Based on the technical design a SEA implementation called FreeCAD-SEA was written. The source code along with documentation can be found at [12] and is released under the 3-clause BSD license [7]. This section focuses on what is implemented and how that was done, as well as what is and what is not possible with this implementation.

6.5.1 Design

Initially the design as outlined in the previous section was followed. Unfortunately several problems were encountered, which resulted in a change of the design. Encapsulation was initially the method for connecting the Adapter and Model classes. A significant problem was encountered when saving the SEA model to file. When saving the SEA model a text representation of Adapter objects has to be saved. This is done by encoding all attributes to text. When loading a model, the attributes are decoded and fitted back into an Adapter instance.

The Adapter objects all had a Model object as an attribute, which could not easily be encoded as JSON [5]. Simple attributes like for example strings or floats are no problem to encode. While it is technically possible to encode such a complex object as text it proved to be difficult. Therefore encapsulation was abandoned and inheritance was adopted instead, solving this problem.

Despite the fact that Model objects now do not have to be stored anymore, there were still complications with saving the SEA model. Every Model object contains references to other Model objects that it requires. As an example, in order to calculate the mass of a *component*, the *component* Model object requires a density described in a *material* Model object. When saving the *material* Adapter object, the *material* object is saved. When saving the *component* Adapter object, not just the reference to the *material* object is saved but also a copy of the *material* object itself. Apparently the objects have become disconnected and the *material* is saved twice. It is therefore not possible to restore the original SEA model. A solution to this problem will be presented in section 6.6.

[5] JSON is a text format for storing objects.

6.5.2 Support of composites

Initially it was thought that adding a *material* attribute to *part* classes would also benefit other FreeCAD modules, e.g. a Finite Element Method module which is currently in development. The *part* module is however solely intended for geometric modelling. The required material parameters also depend on the type of analysis and on the geometry that is considered. For example, in an SEA model certain portions of geometry might be put together in order to obtain a high enough modal overlap. This piece of geometry might then require a different set of material parameters, as was explained in the previous chapter.

For these reasons composites were eventually not supported in the SEA implementation. Nevertheless a simple implementation was made of the model described in chapter 4, of which a validation is shown in chapter 7.

6.5.3 Features

As mentioned before it is not possible to save and restore a SEA model. A significant redesign is necessary to make that possible. The next section will explain what changes are required.

It is currently possible to create a SEA model through both the Python API as well as the graphical user interface (GUI). Solving the modal powers works as well, though no validation has been performed. Results are not (yet) shown in the GUI, but can be retrieved through the Python API.

Plate-room and room-plate couplings have been implemented but were validated neither. Prediction of structural coupling loss factors is not included. An implementation was written for the calculation of CLF's for line junctions but gave wrong values and is therefore not included. The user is currently supposed to manually enter the CLF's.

6.6 Future improvements

In order to make the implementation usable a significant change is needed. First of all, encapsulation shall be adopted again as the method for connecting Adapter and Model classes. Next, only the *system* Adapter shall store a Model instance. The other Adapter classes will be changed so that they will not store a Model object anymore. Instead, they control their Model object through the Model object of the system. Figure 6.5 illustrates the new concept.

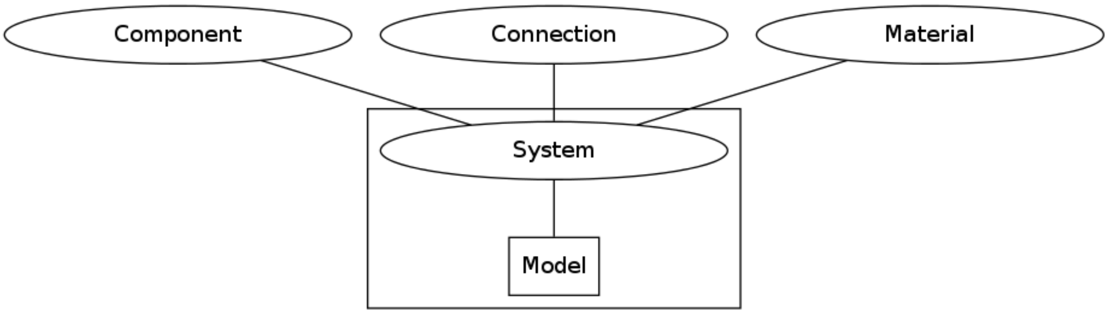

Figure 6.5: An overview of the new design. Only FreeCAD Adapter classes shall exist for *component*, *connection*, *material* and *system*. The Model is entirely contained by the *system*. The other Adapter classes communicate with the Model through the *system* Adapter.

Using this design means that only the Model object of the *system* has to be saved. This is still a quite complex object to encode as JSON. The tool jsonpickle [15] might be capable of encoding and decoding such a complex object. This has not been tested yet though.

7 Validation

Due to the problems encountered with developing the SEA implementation only the validation of the composite laminate model implementation will be shown. The implementation was validated against data provided in [11]. Three validations were conducted.

7.1 Material data

In all three validations the laminae consisted of a single material, S-2 Glass/3501-6. The ply thickness, 2.54 centimetre, was also constant. The Young's modulus in longitudinal direction E_1 was 57.85 GPa and the loss factor

$$\eta_1 = 5.15 \cdot 10^{-7} f + 5.99 \cdot 10^{-3}$$

where f is the frequency of analysis. The Young's modulus in transverse direction E_2 was 19.86 GPa and the loss factor

$$\eta_2 = 8.37 \cdot 10^{-4} \log f + 4.07 \cdot 10^{-3}$$

The shear modulus in the 12-direction G_{12} was 6.1 GPa and the loss factor

$$\eta_{12} = 25.5 \cdot \log f - 50.37 \cdot 10^{-4}$$

Finally the Poisson's ratio in two directions was required

$$\nu_{12} = 0.0913$$
$$\nu_{21} = 0.346$$

7.2 Parametric studies of flexural damping loss factor

The flexural damping loss factor was determined as function of ply orientation for two different laminates, each entirely consisting of the previously mentioned glass. Fibre orientations ranged from 0° to 90° in increments of 15°. The first laminate was symmetric angle-ply and consisted of 16 ply. The second laminate was off-axis and also consisted of 16 ply.

Figure 7.1 shows the flexural loss factor as function of frequency for the angle-ply configuration. The model seems to be in agreement with figure 4 shown in [11].

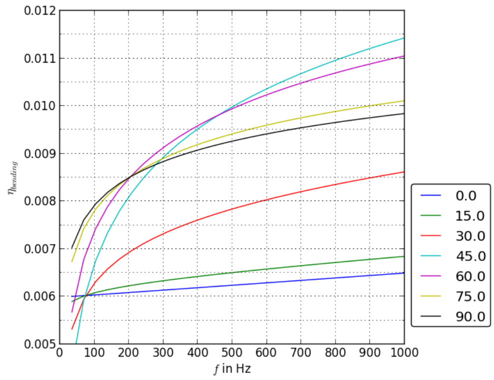

Figure 7.1: Flexural loss factor for different fibre orientations using angle-ply configuration.

Figure 7.2 shows the flexural loss factor as function of frequency for the off-axis configuration. The model seems to be in agreement with figure 5 shown in [11].

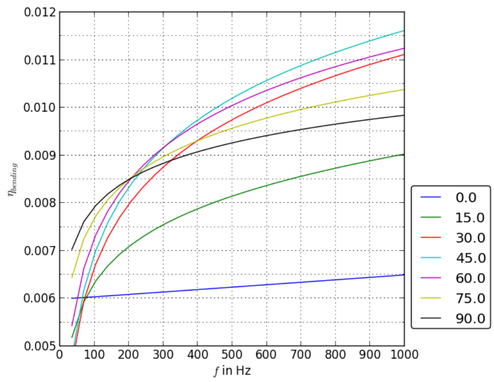

Figure 7.2: Flexural loss factor for different fibre orientations using off-axis configuration.

A final validation was made for three more complex configurations. The three configurations are denoted as

$$(90/0/-45/45)_{2S} \tag{7.1}$$
$$(45/-45/90/0)_{2S} \tag{7.2}$$
$$(0/90/45/-45)_{2S} \tag{7.3}$$

The numbers between parenthesis indicate fibre orientation angles. The number 2 in the subscript indicates that the pattern is repeated twice and the S that it is a symmetric laminate. The three ply considered thus consist each of 16 layers with orientations as shown in table 7.1.

Layer	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
7.1	90	0	-45	45	90	0	-45	45	45	-45	0	90	45	-45	0	90
7.2	45	-45	90	0	45	-45	90	0	0	90	-45	45	0	90	-45	45
7.2	0	90	45	-45	0	90	45	-45	-45	45	90	0	-45	45	90	0

Table 7.1: The orientations of the three laminates.

Figure 7.3 shows the flexural loss factor as function of frequency for the three configurations.

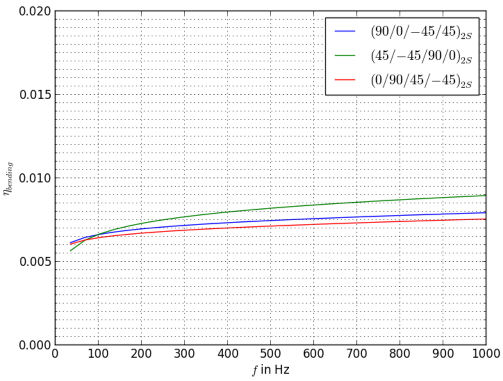

Figure 7.3: Flexural loss factor for three different configurations.

The result matches with figure 7 in [11], in which the results were also compared with experimental data.

8 Conclusion

An SEA implementation was designed and implemented as well as a model for the prediction of complex elastic moduli of composite laminae. The goal was to include the composites model in the SEA implementation. Several problems were encountered with the SEA implementation of which some were significant rendering the implementation practically unusable. A significant change has been motivated which should fix the main issue that was encountered. The current plan is to fix this issue and continue development of the implementation.

Due to several reasons the implementation of the prediction of complex elastic moduli was separated from the SEA code. Both implementations now exist in separate code bases. The composites implementation was successfully validated against literature values.

Bibliography

[1] Glare honeycomb. http://en.wikipedia.org/wiki/File:Glare_honeycomb.jpg.

[2] NumPy. http://www.numpy.org/.

[3] Open CASCADE. http://opencascade.org/.

[4] Python Programming Language. http://python.org.

[5] Stress Strain Ductile Material. http://upload.wikimedia.org/wikipedia/commons/8/84/Stress_Strain_Ductile_Material.png.

[6] Tensors: Stress, strain and elasticity. http://serc.carleton.edu/NAGTWorkshops/mineralogy/mineral_physics/tensors.html.

[7] The BSD 3-Clause License. http://opensource.org/licenses/BSD-3-Clause.

[8] The GNU Lesser General Public License, version 2.1. http://opensource.org/licenses/LGPL-2.1.

[9] Apple Inc. Model-View-Controller. http://developer.apple.com/library/mac/#documentation/General/Conceptual/CocoaEncyclopedia/Model-View-Controller/Model-View-Controller.html, 2012.

[10] I. Bosmans, P. Mees, and G. Vermeir. STRUCTURE-BORNE SOUND TRANSMISSION BETWEEN THIN ORTHOTROPIC PLATES: ANALYTICAL SOLUTIONS. 1995.

[11] Roger M. Crane and John. W. Gillespie. Analytical model for the prediction of damping loss factor of composite materials. *Polymer composites*, 13(3), 1992.

[12] Frederik Rietdijk. FreeCAD-SEA. https://github.com/FreeCAD-SEA/Sea.

[13] Sebastian Ghinet and Noureddine Atalla. Modeling of general laminate composite structures with viscoelastic layer. *Canadian Acoustics*, 34(3):203, 2006.

[14] Daniel Johansson and Peter Comnell. Statistical Energy Analysis software. Master's thesis, 2010.

[15] John Paulett. jsonpickle. http://jsonpickle.github.io/.

[16] John W. Eaton. GNU Octave. http://www.gnu.org/software/octave/.

[17] Richard H. Lyon and Richard G. DeJong. *Theory and Application of Statistical Energy Analysis*. RH Lyon Corp, second edition edition, 1988.

[18] MathWorks, Inc. MATLAB. http://www.mathworks.in/products/matlab/.

[19] Juergen Riegel, Werner Mayer, and Yorik van Havre. FreeCAD. http://freecadweb.org.

[20] S. Finnveden. Lecture Notes for the course Energy Methods. Technical report.